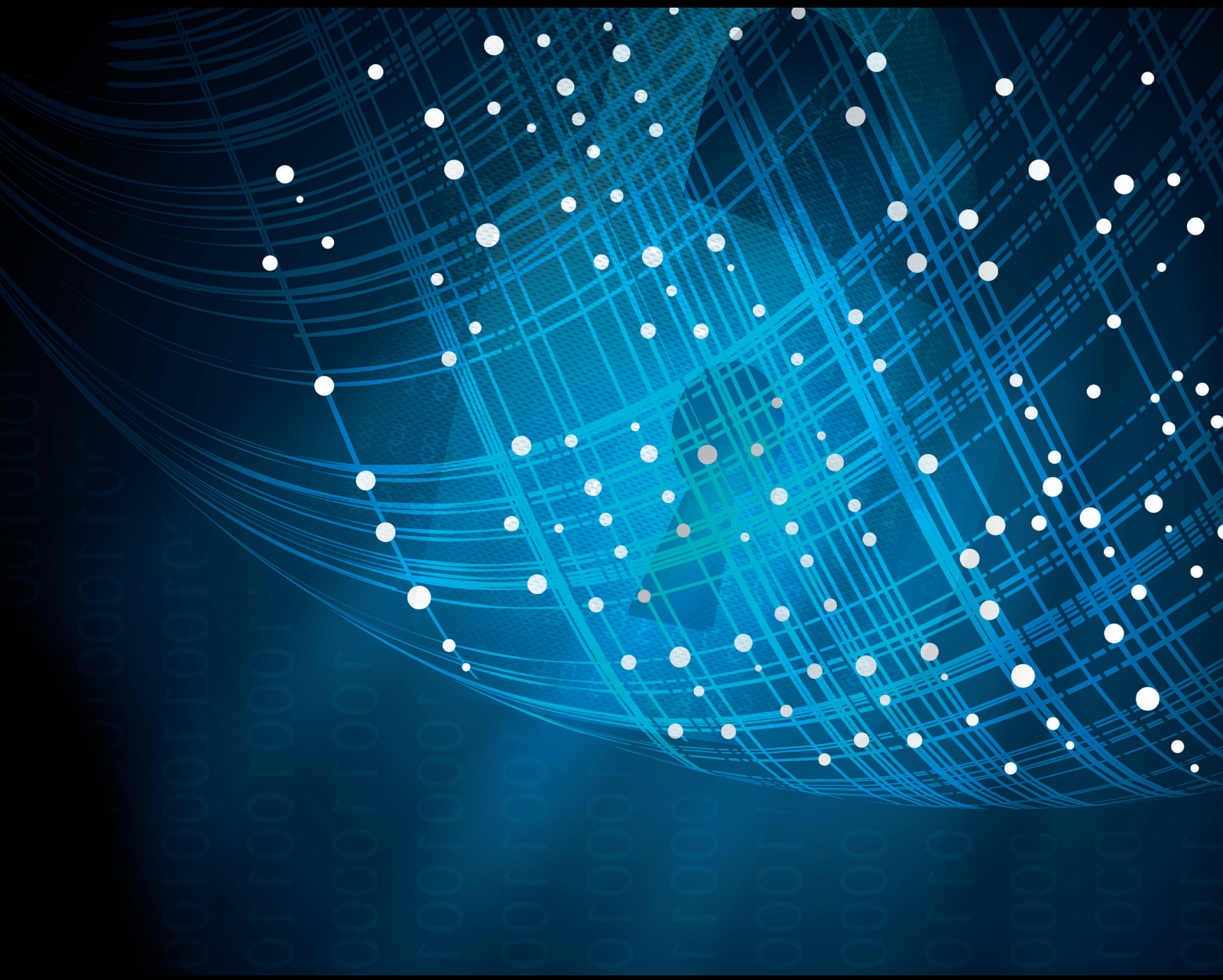
Security and Communication Networks

Blockchain Technologies for Decentralization and Forensics of Outsourcing Services

Lead Guest Editor: Yinghui Zhang
Guest Editors: Chunhua Su, Qi Li, Jin Cao, and Jianting Ning



Blockchain Technologies for Decentralization and Forensics of Outsourcing Services

Security and Communication Networks

Blockchain Technologies for Decentralization and Forensics of Outsourcing Services

Lead Guest Editor: Yinghui Zhang
Guest Editors: Chunhua Su, Qi Li, Jin Cao, and Jianting Ning

Contents

Hindawi
Security and Communication Networks
Volume 2021, Article ID 9910078, 19 pages
https://doi.org/10.1155/2021/9910078

Review Article

The Practicality of Adopting Blockchain-Based Distributed Identity Management in Organisations: A Meta-Synthesis

Sarah S. M. Mulaji and Sumarie S. Roodt

Department of Information Systems, University of Cape Town, Rondebosch 7700, Cape Town, South Africa

Correspondence should be addressed to Sarah S. M. Mulaji; mljsar001@myuct.ac.za

Received 18 March 2021; Revised 24 July 2021; Accepted 25 October 2021; Published 28 November 2021

Academic Editor: Yinghui Zhang

Blockchain has become an irresistible disruptive technology with the potential to innovate businesses. Ignoring it may in itself result in a competitive disadvantage for organisations. Except for its original financial application of cryptocurrency, more applications are being proposed, the most common being supply chain management and e-voting systems. However, less focus is made on information and cybersecurity applications of blockchain, especially from the enterprise perspective. This paper addresses this knowledge gap by exploring blockchain as a use case for identity management in the context of an organisation. The paper gives a comprehensive background aiming at understanding the topic, including understanding whether claims made around it, especially blockchain's potential to address identity management challenges, are based on facts or just a result of hype. Meta-synthesis was used as a research methodology to summarise the 69 papers selected qualitatively from reputed academic sources. The general trend shows theoretical evidence supporting some of the claims made but not necessarily friendly to the enterprise context. The study reveals a promising but immature state of blockchain, consequently questioning whether adopting blockchain-based distributed identity management in organisations is fully practical. A research model called TOE-BDIDM is proposed to guide further investigation.

1. Introduction

"Issues related to data integrity are most acute, as data tampering can have a huge impact on mission-critical services that depend upon reliable data" [1]. One of the fundamental steps in enforcing data integrity is safeguarding the digital system (such as a network, a website, a database, and an application) using the data through effective identification and authentication management. In this way, only authorised people can access the system and potentially use the data. Yet data breaches and their consequences are still occurring, making current IDM systems to some extend questionable [2]. For example, a Serianu report revealed that Africa has one of the highest cybercrimes and financial losses [3]. The IBM 2019 Cost of a Data Breach Study reported an increase in the average cost of a data breach in South Africa, by 12% from 2018 to 2019 [4].

Meanwhile, several claims are increasingly made about the potential of blockchain to provide a way forward in managing digital identities. Some studies claim that (i) "Blockchain solutions for cybersecurity could represent a paradigm shift in how data manipulation will be defended by creating a trusted system in a trustless environment" and that (ii) "Blockchain could address cybersecurity challenges such as Identity management" [1]. Others claim that (iii) blockchain systems have "arguably no single point of failure vulnerability" [5] and that (iv) blockchain identities are privacy-preserving and (v) "give back to users their power over their data" [6]. Further claims suggest that (vi) centralised IDM systems are "subject to different problems and threats such as data breaches" [7], hence should (vii) evolve to possess distributed, disintermediated and secure capabilities [1]. Therefore, it was worthwhile to explore blockchain as a use case for IDM in organisations.

This study explores how practical adopting blockchain-based distributed identity management (BDIDM) is from the organisational perspective, providing a comprehensive background to understand the topic. This includes understanding whether claims about blockchain concerning IDM, especially blockchain potential to address IDM challenges, are based on facts or merely a result of hype. Because there is so much ambiguity around blockchain topics, "their true nature is often obscured by marketing and hype" [8]. Before reporting the review results, the following section will discuss the methodology followed to execute the research.

2. Methodology

This explorative study followed a "qualitative meta-aggregation and meta-summary" research methodology called meta-synthesis. The latter seeks to summarise and "distil information to draw conclusions" [9] while creating "refined meanings, exploratory theories and new concepts." It is rooted in an interpretive approach and aims to "rigorously synthesize qualitative research findings" to produce generalisable knowledge [10].

This study opted for a realist meta-synthesis by combining positive and interpretive approaches to overcome their respective limitations, including all types of studies: quantitative, qualitative, empirical, conceptual, and review. This realist meta-synthesis shared some similarities with a systematic review, predefining most of the rules followed during the review process [11]. The main difference with a systematic review was that the review process was repeated several times to mature the review scope and satisfy the richness requirement of a qualitative study. Meta-analysis was not suitable because it is linear, typically analyses findings across quantitative studies "to identify statistically significant results" [9], and tends to prioritise objectivity over richness [10]. The predefined rules in this review were the review scope, data location (databases), search terms, selection criteria, exclusion criteria, and techniques and procedures of analysis and synthesis. The initial phase consisted of framing the review exercise, determining the scope of the review.

2.1. Framing the Review Exercise. Scoping meta-synthesis is still a debate, with some views advocating for "a narrower, more precise approach" and the others advocating for "a broader, more inclusive stance" [10]. Since this review follows the realism philosophy, it considered a pragmatic approach by having the scope dictated by the themes that made up the topic and having it refined as needed to mature. After several refinements, the final scope retained four main themes (MT) that were further broken down into subthemes. Two main themes represent the fundamental concepts of the topic (MT1: "identity management" and MT2: "blockchain technology"), and the two represent the interrelationships between them (MT3: "enterprise perspective of BDIDM and implementation proposals" and MT4: "related theories").

2.2. Phases of the Review Exercise. Figure 1 shows that the review exercise consisted of five phases repeated four times over a year as new papers were published: December 2019, March 2020, June 2020, and September 2020. The review did so to allow the maturity of the scope and accommodate the topic's relative newness at the time of writing. There was not much written on the topic at the beginning of the research process. The review ended when the topic was saturated: there was a repetition of what was already lent. The main requirements throughout the review process were to achieve *diversity* when locating papers, *inclusion* when deciding what to include, *fairness* when appraising studies, *genuineness* when analysing studies, and *richness and simplicity* when synthesising them.

Diversity in information sources was achieved by including unusual sources such as reports, standards, and theses, often inaccessible from common databases. Therefore, in addition to those recommended for information system studies (the five databases included in EBSCOhost), the review considered other databases to accommodate the technical side of the topic (IEEE and ACM) and generic ones such as Google Scholar to boost diversity. Given the topic complexity and high variance rate of its concepts, the search terms were intentionally exhaustive to capture as much information as necessary to cover the scope of the review. As shown in Table 1 below, the search terms were derived from the four main themes and used one at a time in each predefined database. This data retrieval technique is also called "berrypicking of information" [10].

Inclusion was achieved by considering different types of papers, from books to unpublished theses, as well as considering studies with "different methodological approaches" since meta-synthesis embraces the challenging idea that "multiple approaches can be synthesized" [10]. The remaining selection criteria were simply based on common sense.

The fairness of the results was ensured by assessing the quality of individual studies using the ten basic claims by Ngwenyama [12] as part of the appraisal phase. Some studies often bypassed the appraisal stage, assuming that "the rigour of individual studies is less important than the attempt to be as inclusive as possible" [10]. After all, the review adopted a centric approach that values both studies' inclusion and results' fairness. In addition, the review assessed the validity of the claims made about the topic using related theories.

The originality of the findings was ensured by trying to preserve the original meaning of the text of individual studies while resisting, as much as possible, "the temptation to force a fit in the interests of illustrating homogeneity," since "the links between studies may be reciprocal, complementary or conflicting." Originality also partially justified the intense use of direct quotes. The selected studies were seriously reviewed to identify key ideas to aggregate and draw common themes and concepts. These were then "juxtaposed to identify homogeneity to note discordance and dissonance" [10].

The richness of the account was achieved by opting for a narrative synthesis that "reflects the tension between contradictory or alternative explanations if reciprocal translations suggest a lack of congruence." In this way, the synthesis provides a comprehensive background necessary to understand the links between concepts and the underlying debate

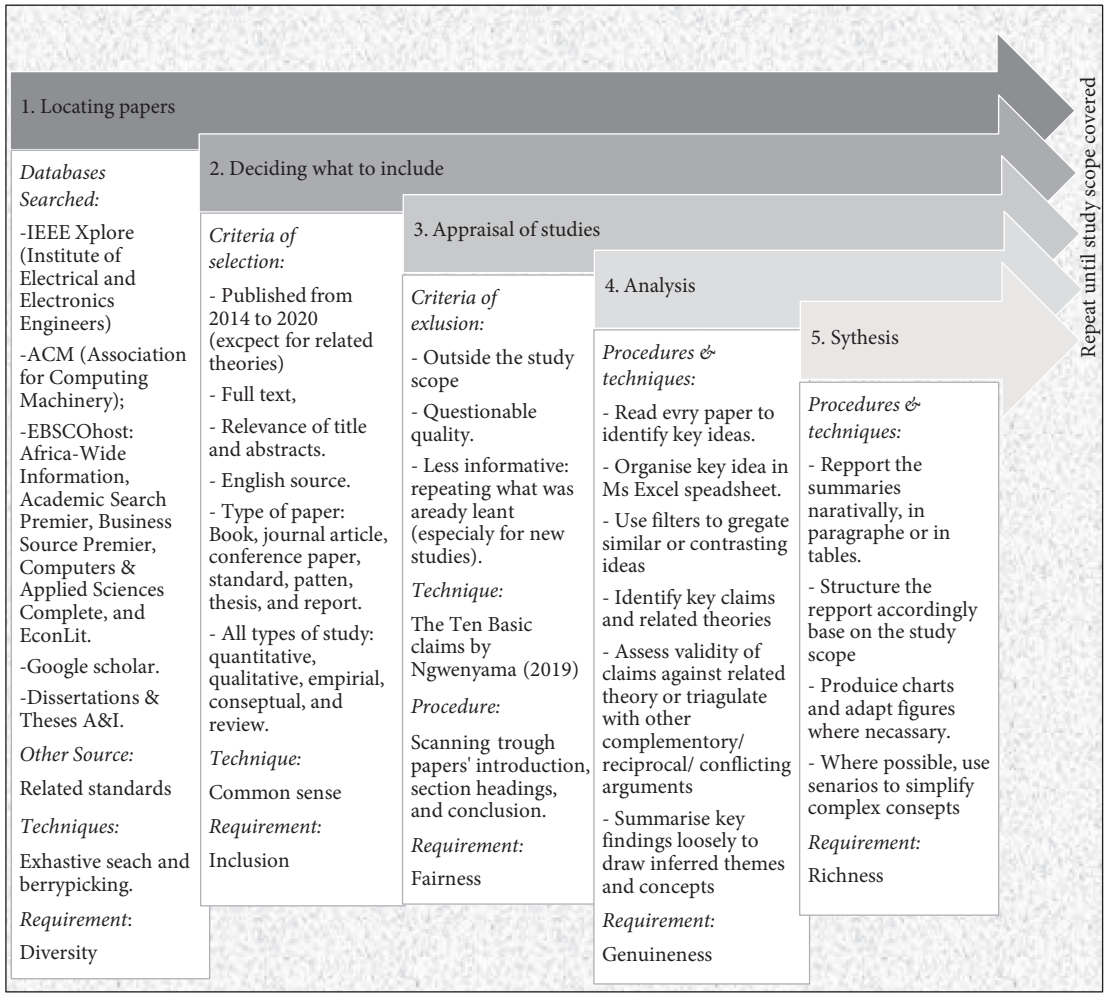


Figure 1: Summary of the five phases of the review exercise.

Table 1: List of search terms.

Search terms
(i) ("Identity Management" OR "ÍDM" OR "Identity and Access Control" OR "IAM") AND (issues OR challenges OR problems OR vulnerabilities OR implementation)
(ii) (Blockchain OR distributed) AND (OR "Identity Management" OR "Identity Authentication" OR "Identity Proofing" OR IDM)
(iii) [Blockchain AND (identity OR ID)] AND (issues OR challenges OR weaknesses OR problem OR vulnerabilities)
(iv) [(Permissioned OR Permissionless) AND "Blockchain"] OR ("Public Blockchain" OR "Private Blockchain" OR "Open blockchain" OR "federated blockchain")
(v) "Adoption of blockchain" OR "blockchain adoption" OR "Blockchain ID adoption" OR "Distributed ID adoption"
(vi) ("Sigle point of failure" AND "Identity management" AND blockchain) OR [(central ∗ OR distribut ∗) AND (architecture OR system)]

around "enterprise BDIDM." Eventually, the synthesis as a "whole is greater than the sum of the constituent parts." To achieve simplicity while increasing comprehensibility, the review used illustrations, images, and scenarios to simplify complex concepts while using tables to summarise ideas involving a considerable amount of information [10].

2.3. Description of the Sample. After completing several iterations of the five phases of the review exercise and saturating the topic, the final number of selected papers came to 69 (excluding those supporting the research methodology). Descriptive statistics (numbers, percentages, and charts) summarised the sample based on the type of studies and year of publication. The pie chart on the left-hand side of Figure 2 indicates the type of distribution of the sample in percentage, mainly made of 32 conference papers (46.4%), 25 journal articles (36.2%), and 6 books (8.7%). The scatter chart on the right-hand side of Figure 2 indicates that approximately 84% (59) of the 69 papers were published between 2017 and 2020.

Qualitative methods (thematical analysis) described the sample from the perspective of the review scope. Figure 3 shows how each selected paper relates to the review scope of

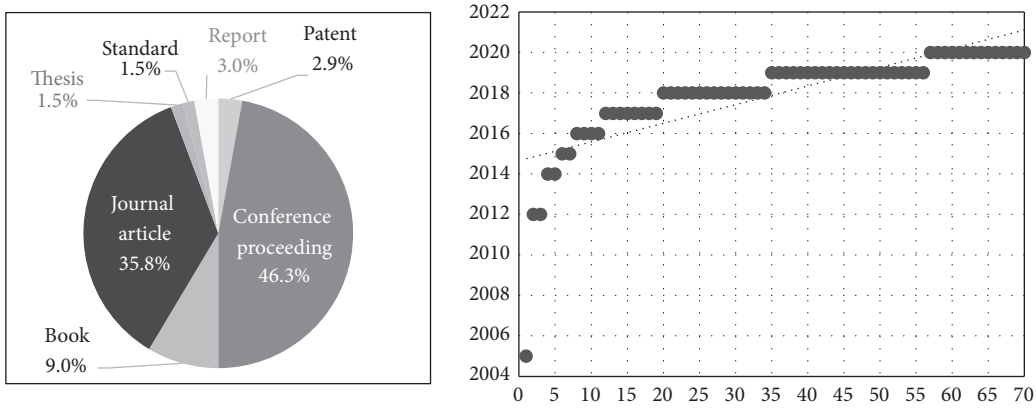


Figure 2: Description of the sample from the perspective of type and year of publication.

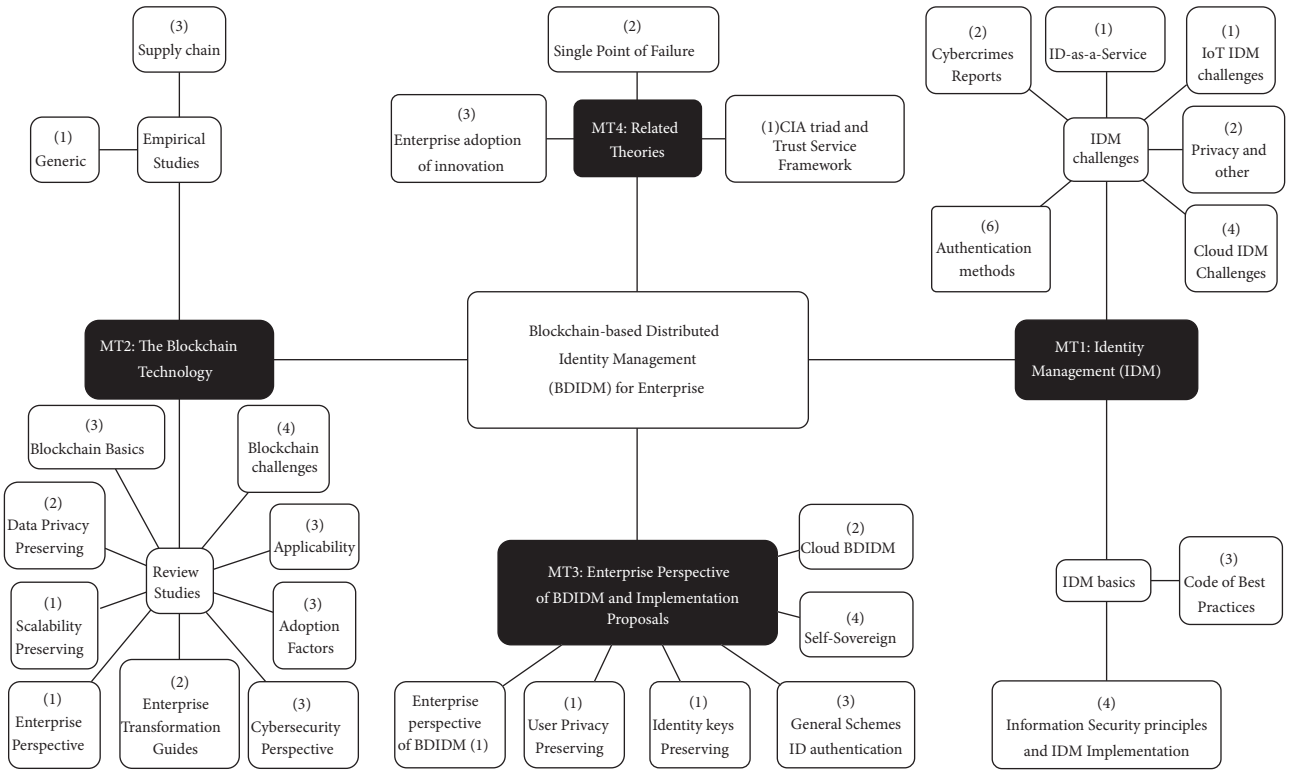


Figure 3: Thematical distribution of sources.

the 4 main themes broken down into subthemes (and leaves themes where possible). It also reports the number of papers retrieved per theme in bracket (n). In total, 26 papers felt under MT2: "the blockchain technology" (22 for "review studies" and 4 for "empirical studies" subthemes), 23 papers under MT1: "identity management" (16 for "IDM challenges" and 7 for "IDM basics" subthemes), 14 papers under MT3: "BDIDM implementation proposals" and "enterprise perspective of BDIDM," and 6 papers under MT4: "related theories."

3. Results and Discussion

This section reports the review findings narratively. The review is structured in such a way to cover the main themes within the review scope, as shown in Figure 3. MT1 relates to IDM fundamentals, IDM challenges that need to be addressed and the evolution of IDM models to address IDM challenges. MT2 concerns blockchain fundamentals, including blockchain promoting and constraining factors. MT3 discusses the practicality of BDIDM in organisations

from different angles: concept, IDM model, blockchain implementation, and ability to address IDM challenges. MT4 assesses the validity of claims made about BDIDM throughout the review and explains factors that impact BDIDM adoption in organisations based on the technology-organisation-environment theory.

The following sections of the review gives the fundamentals of IDM and highlights some critical IDM challenges needing to be addressed.

3.1. Identity Management (IDM). A *digital identity* is "a set of claims made by one digital subject about itself or another digital subject." A *digital subject* is the digital illustration of the defined individual, often referred to as an *entity*. A *claim* is an assertion of propriety about a subject [13].

Technically, IDM consists of managing matters related to two fundamental information security principles: *identification* and *authentication*. Identification and authentication are vital first steps in controlling access to a digital system, such as a corporate website, an application, a database, and so on. On the one hand, identification proves that a user is who they claim to be. As illustrated below, this is imperative because access should only be granted to legitimate users (authorisation). On the other hand, authentication proves that a user acted on a system (accountability). Likewise, a user should not be able to deny what they have done (nonrepudiation or nondenial) [14].

> Identification: "*I am a user of this system*"—here is my username: "Alice"
>
> Authentication: "*I can prove I'm a user of this system*"—here is my password: "All#125gef"
>
> Authorisation: "*Here's what I can do with the system*"—I can view and edit "Client_file.mdb"
>
> Accountability: "*You can track and monitor my use of the system*"—I cannot deny my actions [14]

An *IDM system* labels each entity with an identifier (usually in a human-friendly format, for instance, a meaningful string), providing a way for the entity to authenticate (often by proving knowledge of some private information, e.g., a password, phone number, PIN, biometrics, etc.) and stores its relevant identity information on a dedicated component (generally a server) [2].

3.2. The Criticality of Addressing IDM Challenges in Organisations. IDM is a fundamental security control that mitigates security breaches in organisations [14]. However, IDM faces many challenges. The most common are vulnerabilities in authentication methods, vulnerabilities in system architecture, the imbalance between security and privacy, credential reuse and weak credential, and the pressure to achieve "secure cloud" and "secure IoT."

3.2.1. Vulnerabilities in Authentication Methods. Authentication is a principle of information security that challenges the user to provide information that formally proves that they are known by the system and thus may officially log onto it. That information, also called user credentials, can take various forms, from passwords to biometrics, and can be implemented as an authentication method [14].

Unfortunately, every authentication method has known vulnerabilities and can be compromised. Knowledge-based methods like passwords and PIN are vulnerable to guessing attacks such as dictionary, rainbow table, bruteforce, and so on [14]. Moreover, users may experience difficulties in matching their passwords to different accounts [15]. Smart/magnetic cards can be lost or stolen. Hard biometrics, such as finger/palm prints and retina/iris scans, are relatively expensive to implement and invasive for users. In addition, their effectiveness depends on their false-positive and false-negative rates [16, 17]. Soft biometrics methods such as signatures and typing patterns, as well as location-based methods such as the Global Positioning System (GPS) and Indoor Positioning System (IPS), are only secondary to continuously verifying an authenticated user [18].

When users' credentials are compromised, the security of every system relying on them to authorise access is also breached. "Strong authentication requires a minimum of two authentication mechanisms drawn from two different authentication factors" [14]. Therefore, codes of best practices in information security, including the ISO/EIC and NIST, recommend the use of multifactor authentication (MFA) to establish "strong authentication and identity verification" [19, 20]. However, despite the use of MFA, organisations are still facing data breaches. The literature increasingly emphasises that another vital issue weakening IDM systems might be their traditional centralised architecture [21, 22].

3.2.2. Vulnerabilities in the IDM System Architecture. Centralised IDM embeds a critical vulnerability of single point of failure (SPOF), as they use a central server to store the identity data. When the server is compromised, identity data is exposed, and the server may no longer be available [22]. SPOF is a well-known theory in security risk management. It suggests that when a system's overall functionality depends on a single node, there is a high risk for the whole system to collapse when that particular node fails. Some studies suggest that "multicopy redundancy technology" [23] would mitigate the SPOF vulnerability and achieve reliability and resilience in digital systems [24]. Redundancy involves having a duplicate copy of the database on every node, generally known as distribution [25]. That is why distributed systems, such as blockchains, have "arguably no single point of failure vulnerability" [5].

In Figure 4, the left-hand side illustrates a distributed system where all nodes are equal and play the provider and consumer of services. If one node fails, the others can still take over. The right side illustrates a centralised system, such as the client-server, where the server provides services for clients to consume [25]. The failure of the server knocks the whole system down [22]. In a distributed system like blockchain, "more than 50%" of nodes must be compromised first to bring the entire system down, which is extremely difficult to achieve [5].

3.2.3. Balance between Security and Privacy. The ongoing data breaches in organisations indicate the need to ensure effective identity and access management systems [26]. Sometimes, organisations undermine privacy, since security managers face a dilemma about user identity data. On the one hand, organisations need to comply with their business strategy seeking "user ownership," which involves having direct contact with and getting much information as possible about their (potential) customers. On the other hand, security managers must protect users' privacy in compliance with government regulations such as POPIA in South Africa. Users, of course, "want good services offered in convenient ways" yet are very "concerned about infringements to their privacy" [27].

An example of a "security and privacy conflicting" business requirement is the Know Your Customer regulation to verify clients' identities in the banking industry. This mitigates the risks posed by malicious customers and "is part of Anti Money Laundering initiatives" [28]. In this case, centralised IDM might be dangerous for customers' privacy as it endorses total control of customers' identity data to banks. Customers must trust banks not to exploit this data and "effectively protect it from external attacks" [2]. This issue verifies the theory of "the CIA triad," an acronym for three fundamental objectives of information security: *confidentiality*, *integrity*, and *availability*.

Whitman and Mattord indicate that the CIA triad "has been the standard for computer security in both industry and government since the mainframe development" [14], apparently formally established by Donn Parker in 1998. This theory suggests that the security and reliability of a computer system depend on a balance between confidentiality, integrity, and availability. Confidentiality prevents unauthorised access to information; integrity prevents unauthorised modification of information; and availability ensures the information is always available to authorised users [14]. However, another underlying requirement for a digital system is privacy. Privacy prevents unauthorised access to the personal data of employees, clients, partners, and so on. Figure 5 illustrates a typical application of this extended CIA as the Trust Service Framework (TSF), developed by Romney et al. [29] to guide the field of accounting information systems. Just as a four-legged table cannot balance if one leg is missing, the TSF suggests that security without privacy is problematic.

3.2.4. Credential Reuse and Weak Credentials. The Internet has grown significantly. As a result, numerous online services have forced users to have dozens of accounts with specific online services they subscribe to, causing the burden of matching every account with its credentials [14]. Users have been reusing the same credentials on different services, creating redundant security data [30]. In this way, when one service is compromised, the security of all substantial services relying on the same credential to authorise access is also breached. Others use weak passwords, so they are easy to remember, making it easier for imposters to guess.

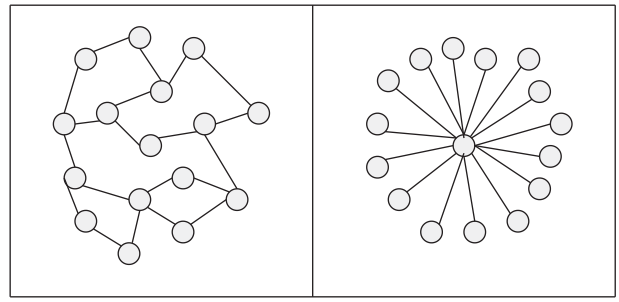


Figure 4: Distributed versus centralised system architecture (adapted from [25]).

Meanwhile, guessing engines known as bruteforce attacks are getting more sophisticated, using high computation power. In 2019, a hacker under the pseudonym "Tinker" announced on Twitter that an open-source password recovery tool could crack an 8-character Windows NTLM password hash in less than 2.5 hours.

3.2.5. "Secure Cloud" and "Secure IoT". Initially, IDM systems were used to identify a living individual in a digital system and involved authenticating them as a legitimate user of the system [2]. Today, IDM systems need to identify and authenticate not only individuals but also "things" such as software, smartphone, robot, automobile, appliances, entertainment devices, and so on—hence the origin of the so-called IoT, an acronym for internet of things [31]. IoT has made IDM management even more complex than before due to the many interconnected smart devices interacting with computers and humans today. Since "the security of these devices has not always been a primary concern" of their vendors, IoT increases the possibility of security breaches [14].

Furthermore, secure and reliable IDM appears to be "the greatest challenge facing cloud computing today" [32]. Although "accountability is the main construct and key enabler of trust" in the cloud [33], "secure and reliable management of identities" is proven "the greatest challenges facing cloud computing today" [34]. Effective IDM in the cloud is a "key area of cloud security" and is vital for its wide adoption [35, 36], Still, traditional cloud-based identity and access control systems follow a centralised approach, where a cloud server acts as the central authority controlling access to data in the cloud [37].

The following subsection discusses the development of IDM models and their attempts to address the above IDM challenges over time.

3.3. Evolvement of IDM Models in Addressing IDM Challenges in Organisations. Traditional IDM systems implement a service-centric approach, also seen as an organisation-centric approach, principally including centralised and federated IDM models. A new approach to IDM tends to be user-centric, including the so-called self-sovereign identity (SSI) and some types of federated identity [2]. Figure 6 illustrates the contrast between the two approaches.

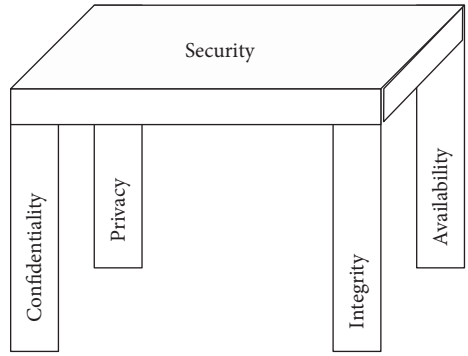


Figure 5: The CIA triad and the TSF.

3.3.1. Centralised IDM. Traditional IDM systems are "based on central authorities" usually isolated from each other, setting up silos of trust in such a way users "cannot sign on across different domains" [7]. As a result, "users are forced to rely on a different central service to manage their identity data in each different domain" [2]. A user has an account (username and password or biometrics) for every isolated service. Although this is virtually perfect from the enterprise perspective (since it gives an organisation complete control over the use of "its" digital assets), it is "inefficient and cumbersome for users (forcing them to remember many different private authentication information)" [2]. Centralised IDM systems use protocols such as RADIUS and Kerberos, providing authentication of both individuals and applications on a dedicated server [38].

3.3.2. ID-as-a-Service. The centralised cloud model of IDM is also called ID-as-a-service. In this model, the organisation transfers its responsibility of managing the identities of its digital systems, including related costs, to a trusted third party. However, most organisations would prefer to manage identities themselves rather than outsourcing it as a service, mainly due to privacy issues and the legal responsibilities involved, especially in data breaches. ID-as-a-service utilises cloud-based services protocols, usually vendor-based products, such as OKTA or AWS-IAM, providing authentication of both individuals and applications on a dedicated server in the cloud [7, 39].

3.3.3. Federated IDM. Federated IDM is a model of trust that helps mitigate partially the problems posed by centralised IDM by "enabling Single Sign-On (SSO)," a kind of server-centric system that "enables users to adopt the same identity system across different domains" [38]. When signing on a trusted third-party system, "the user is redirected for authentication and user identity data retrieval to his home *identity provider*" [7]. In this way, the third-party's system, known as *identity consumer*, is granted some privilege on the user's identity data stored on their home central authority over the Internet [14]. In other words, if services A and B trust mutually, a user registered with service A can access service B without creating an account with it, and vis-versa. A typical example of a federated IDM is when a given online shopping website can be accessed using a Google account. Federation uses protocols such as OpenID, SAMUAL, and Auth [40].

3.3.4. User-Centric IDM. Even though federated IDM "eases the burden on users, it still gives them no control over their identity data that remain centralized for each domain as before" [2]. That is where user-centric IDM comes into play. It partially addresses privacy issues by putting the user in charge of some aspects of their own identity data, limiting the privileges of third parties [27].

The system asks users for their consent on how much of their identity information will be "released in the federation from their home identity provider (the data controller) to the service provider (data processor)." However, the user's information is still subject to a potential data breach as their "identity are still held on the server-side, and authentication is validated on the server" [7].

3.3.5. Self-Sovereign Identity (SSI). A typical user-centric IDM uses blockchain to obtain SSI systems [41]. In this model, the decentralized identity provider system is not owned by a single entity. Thus, it "does not represent a trusted third party and allows digital identities that are under full control of the associated subject" [42]. That is why a growing tendency portraits SSI as the most "privacy-respectful solution" for IDM systems [7]. Identity data is stored on the user side, technically on their individual block, using a software wallet installed on their device (like a smartphone) [43]. "Users can register, retrieve and even revoke the data if they do not want to use them anymore" [5].

Figure 7 below illustrates the evolvement of IDM models above discussed from the perspective of their privacy-preserving capabilities.

The following section discusses the fundamentals of blockchain and its impacting and challenging factors from the perspectives of enterprise implementation.

3.4. The Blockchain Technology. Blockchain is a constantly growing distributed record of updates about a specific matter among a group of participants. A consensus protocol

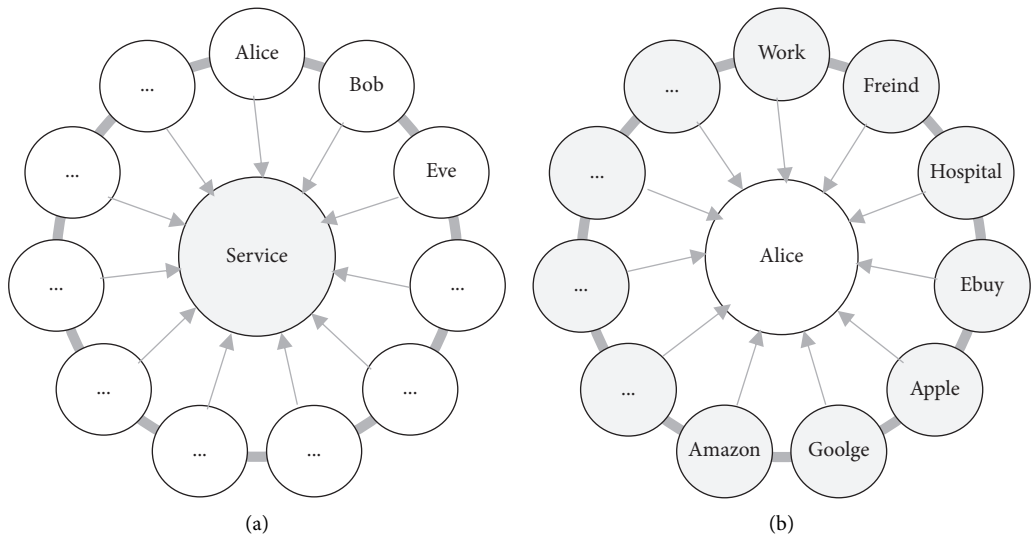


Figure 6: Traditional centralised IDM (a) versus self-sovereign identity (b) models (adapted from [2]).

regulates interactions among participants, and cryptographic technologies, namely digital signature and hash algorithm, maintain security [44, 45]. Table 2 shows that blockchain implementation involves determining three fundamental needs: who can join the network, whether a validator will be needed, and what type of consensus protocol will regulate interactions between participants. Combining these needs results in three types of blockchain implementation: public permissionless, public permissioned, and private permissioned [46, 47].

3.4.1. Enterprise Blockchain (EB). The concept of EB refers to a "permissioned blockchain utilized by any organisation" [48]. However, ambiguities on the applicability of EB in the real world are perhaps one of the reasons for delays in its adoption. "Technology professionals are knowledgeable, yet not enough substantial business problems have been solved with Blockchains" [49]. Demir et al. proposed the Blockchain Technology Transformation Framework (BTTF) to guide executives and managers in evaluating blockchain-based solutions to innovate their industry. Likewise, Labazova [47] proposed the framework for assessing blockchain implementations in organisations, regardless of its use case. However, despite its potential impact on business that could promote its adoption, EB is still subject to various constraints.

3.4.2. Promoting and Constraining Factors ofEB. There are eight important architectural properties of blockchain, paired in a mutual influence relation, that could promote its adoption: decentralisation and disintermediation, programmability and automation, transparency and auditability, and immutability and verifiability [50]. Additional blockchain's impacting features include integrity, origin authentication, and trust. Table 3 below discusses these architectural features of blockchain from the perspective of their business impact.

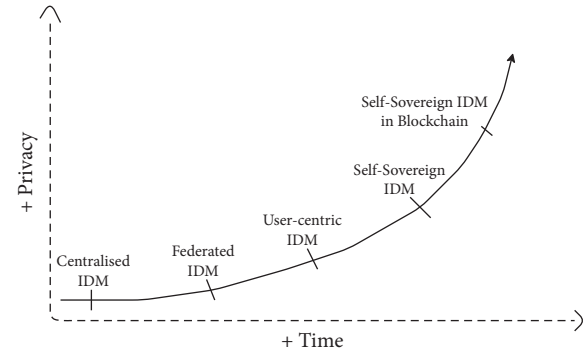


Figure 7: IDM models evolvement over time from the user privacy perspective (adapted from [7]).

Blockchain is a relatively new technology that is still suffering from immaturity [49]. Table 4 discusses the fundamental challenges ahead of its implementation that might prevent or delay its adoption in organisations.

These challenges tend to question the practicality of adopting blockchain-related technologies such as BDIDM.

3.5. The Practicality of Adopting BDIDM in Organisations. This subsection focuses on the pragmatism of BDIDM in the context of an organisation. Among other things, the section discusses the SSI flavour of BDIDM, which was initially intended for individual use on the Internet, evaluating its practicality for the enterprise context, especially the so-advertised potential to address IDM challenges in organisations.

3.5.1. The Practicality of the Concept. The following scenario set up the context of BDIDM in organisations:

> Alice has just joined company B. The company's system administrator, Bob, needs to create a corporate account for the newly recruited employee, Alice. A username, password, biometrics, and other personal information (such as name, physical address, phone number,

Table 2: Blockchain implementation types.

Blockchain implementation					
Consensus protocol	Raft consensus	Prof. of authority (PoA)	Federated consensus	Prof of work (PoW)	Prof of stake (PoS)
Who can join/ validator trust	Private/permissioned	Public/permissioned		Public/permissionless	
Description	"access authorization does not entail validation permissions, which require additional authorization rights given to several nodes." Only trustful nodes enforce consensus.	"only authenticated and predefined users can read and write transactions. All nodes participate in the finding of the consensus. Identifiable nodes determine consensus mechanisms.".		"everyone can read, write, and validate the information. Consensus is enforced by proof-of-work or proof-of-stake. Users are usually anonymous and pseudonymous."	
Application	Enterprise projects (Hyperledger)	Organisational consortia (Ripple, R3)		Cryptocurrencies (Bitcoin)	
References	[46, 47]				

Table 3: Blockchain promoting factors.

Blockchain features and business impacts	
Decentralization and disintermediation	Blockchain eliminates system dependencies and intermediaries [1]. It enables direct interactions between participants without the need for a trusted third party [50, 51].
Programmability and automation.	Smart contracts allow for automated execution of predefined codes "once certain conditions have been met," though arbitrary code may increase bugs [50]. Automation "simplifies complex business processes by alleviating the need for manual interventions" [49].
Transparency and auditability	Each user of the blockchain can track how blocks have been added over time [52]. However, a permissioned blockchain might reduce transparency due to the privacy requirement [53].
Immutability and verifiability	Blockchain keeps temper-evident historical records of all transactions happening on the network [49]. "The information stored in the blocks cannot be changed unless an attacker can gather more than 51% of the computational power network" [52, 54].
Integrity, authentication of origin, and trust	Cryptographic methods ensure that information is protected from unauthorised modifications, improving trust [52, 53].

national identification number, age, e-mail address, etc.) need to be captured in the system. However, Alice already has a digital identity stored on a blockchain. Therefore, she authorises her new employer to access it without viewing her personal data. Alice can now access corporate digital resources using her blockchain-based ID. Bob has no control over Alice's digital identity, as it is stored on an independent system. Alice has complete control over her digital identity and can authorise whatever online service she wants to create an account with, from a hospital to an online shopping website. As a result, Alice only has a single account and thus fewer passwords to recall.

The scenario seems troublesome from the enterprise perspective of IDM for the following reasons: (i) an organisation would tend not to trust Alice's ID because it is external, (ii) it would tend to know whether the participants in that blockchain are trustworthy, (iii) it would not want to lose control over Alice's account since she has access to the company's confidential information, (iv) it would be concerned about what would happen when Alice's ID gets hacked or whether someone is behind Alice's ID to spy the company's business. Yet this is what BDIDM for enterprise, especially in its SSI flavour, is all about.

SSI is a paradigm focusing on a user-centric approach, an IDM model that emerged with blockchain. It "strives to place the user in full control of their digital identity" [1, 42]. SSI is a result, on the one hand, of the decrease in users' trust in major corporations. Users are increasingly concerned about their privacy that they disapprove of the misuse of their personal data. On the other hand, "the awareness of the commercial worth of user data ownership by service providers and networking" advocates for giving back the user their power over their data [6].

3.5.2. The Practicality of the BDIDM-SSI Model. Nearly the entire sample of the papers retrieved on BDIDM implementation proposals, regardless of whether they included the enterprise context, tended to converge toward the SSI as the ideal BDIDM model. They claim that SSI is decentralised and distributed [62]. Decentralisation refers to the removal of the IDM central authority (server). In contrast, distribution refers to utilising the exact copy of a user's ID across all components of the IDM system (redundancy) [2].

Table 4: Blockchain constraining factors.

	Technology challenges
Software and sustainability issues	Software used to ensure transactions among active participants on a blockchain network are open-source, thus subject to frequent updates [49]. Recurrent updates make the blockchain system "highly volatile" [55].
Technical integration challenges	Due to its decentralised architecture, blockchain may make it difficult to connect with legacy systems [49]. A poorly designed blockchain can result in a system incompatible with existing systems, such as "a fine-grained identity" [55] and role-based access control [56].
Scalability and performance	Blockchain requires a careful design to "ensure sufficient scalability without sacrificing decentralisation" [1]. Scalability is generally measured in throughput, latency, bootstrap time, storage, cost of confirmed transactions, fairness, and network utilization [8].
Security	It is possible to breach the security of a blockchain "when a "miner" controls more than 51% of the computing power" [54, 57]. Although this is still thought very difficult to achieve, it may not be impossible with quantum computing [1, 58].
Skill shortage	"Blockchain-focused technical skills are not yet taught in standard higher education curricula" [59]. As a result, the industry is suffering from a deficit of expertise. Meanwhile, the demand for blockchain skills is growing [49, 59].
Complexity	Blockchain is considered both "user and developer unfriendly." It is thought complex to implement and difficult for a user to adapt [60].
	Business challenges
Cost-benefit analysis	Blockchain ecosystems were initially designed as "an investment rather than a traditional business use with an expected return on investment." Its upfront implementation cost is high, as it includes new infrastructure and a highly skilled team, which rather negatively impact existing revenues [49]
Governance	"The governance of a blockchain concerning updating its fundamental rules is problematic" [50]. "The whole network relies on a consensus mechanism" that involves all the nodes, "which can be any device" [61]. Therefore, there are issues of accountability and management [56].
Uncertain regulatory status/lack of standards	The lack of firm regulatory guidelines and policy standardisation is "the most concerning challenge for bringing blockchain into many fields daily," as "laws tend to catch up slowly with new technology" [49, 59].
Cultural adaptation and reluctance to change	The blockchain distributed fashion of sharing information "not only distributes power but also reduces the control of former authorities" and "fear of unknown technology and its possible shortcomings can cause concern" [49].
Awareness	The widespread adoption of blockchain is also potentially restricted by the lack of adequate knowledge and awareness [56].

> Technically, SSI allows individuals to "create immutable identity records represented as identity containers capable of accepting attributes or credentials from any number of organisations. Each organisation can decide whether to trust credentials in the container based on which organisation verified or attested to them" [2].

Figure 8 illustrates that the SSI identification process involves three parties: (i) the *subject* of the identity (user: an individual or a thing), (ii) the *certifier* or *insurance* to notarise the documents (usually "a government agency, an accounting firm or a credit referencing agency"), and (iii) the *inquisitor or verifier*, which is the service provider that "inquires into the identity of the subject" [5]. The user obtains a distributed identity (DID) with verifiable claims and credentials from the issuer authority, in a user-centric way using their devices such as a smartphone. The latter hosts a software wallet that keeps keys secure [43]. SSI's privacy-preserving capabilities can enable the user "to present Zero-Knowledge crypto proofs against a Service Provider acting as verifier that checks in the blockchain attestations and signatures" [7].

The principles of SSI include existence, control, access, transparency, persistence, portability, interoperability, consent, minimalisation, and protection [2]. These principles could be summarised in "three characteristics usually required by any IDM system: "*Security*, the identity information must be kept secure; *controllability*, users must have control of who can access their data; and *portability*, the user must be able to use their identity data wherever they want and not be tied to a single provider" [2]. The main contrast with traditional IDM systems is the control given to the user rather than to the identity provider.

However, as shown in Figure 8, a smartphone can be considered as a token authentication method, so there are still security concerns when the wallet is compromised, for example, in the event of a lost or stolen smartphone [14]. Beyond this, the long-term challenge for SSI is to be resilient to the rule of 51%: a severe security breach that happens "when a "miner" controls more than 51% of the computing power" [54, 57]. This cyberattack on blockchains may still be though difficult to achieve but may not be impossible with quantum computing [58, 60].

3.5.3. The Practicality of the Ideal Blockchain Implementation . Figure 9 shows that public permissionless blockchains, on the one hand, tend to be decentralized, transparent, and scalable but inefficient in computing power and, thus, are

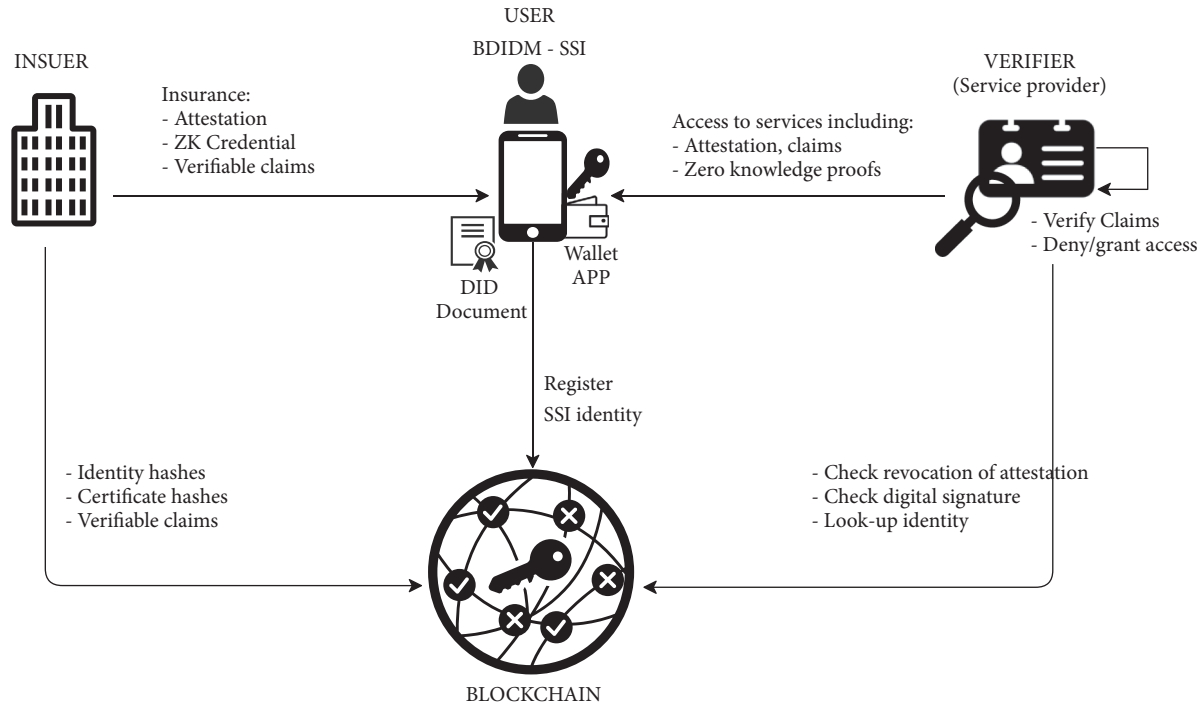


Figure 8: SSI model (adapted from [7]).

slow. On the other hand, private permissioned blockchains tend to be more centralised, less transparent, and not scalable but efficient in computation power consumption and, thus, are fast. The challenge of blockchain is that consensus algorithms, especially PoW, used to create a trustful system in a trustless environment are technically expensive to achieve. For "more efficient and simpler consensus algorithms," it is necessary to relax trust assumptions in the system, balancing between decentralisation and transparency. "The more trust a system places on nodes," "the more efficient the system gets, but often also the more centralised" [2].

Public permissioned blockchains, also known as federated blockchains, are more balanced versions of blockchains [63]. They tend to fit the concept of federated IDM discussed earlier and are claimed to be more decentralised, scalable, and efficient [57] and ensure "privacy protection and high transparency" [62]. A public permissioned blockchain seems the ideal implementation for BDIDM. Indeed, Sovereign Foundation, a firm that advocates for SSI on the Internet, claims to create "blockchain instances that are open for all to use," but whose network of nodes performing consensus is permissioned [7].

Still, one would argue that private permissioned blockchain may be the ideal implementation for "enterprise BDIDM" because it endorses a service-centric approach by giving total control of the system to the identity provider called "Trust Anchor." But a service-centric approach to BDIDM would not differ from the traditional centralised IDM, from which one would want to move. "A Trust Anchor defines who represents the highest authority of a given system that has the authority to grant and revoke, read, and write access." A node with the "read" privilege can only view some aspects of the identity, while a node with the "write" privilege has full access to the identity data and can modify or even block it [37].

Wüst and Gervais [53] proposed a structured methodology to determine the appropriate blockchain implementation to address the choice of blockchain implementation ambiguities. The methodology suggests that the choice should depend on trust assumptions. From the outsider-threat perspective of cybersecurity theory supporting traditional implicit trust [14], this means that BDIDM would be unnecessary for *trusted users* (staff members accessing the system from the intranet). That permissioned BDIDM would make sense for *semi-trusted users* (clients, suppliers, partners, etc., accessing the system from the extranet) and permissionless BDIDM for *untrusted users* (visitors or any unknown user accessing the system from the Internet).

However, with the rise of the insider-threat perspective of cybersecurity, there is a growing tendency to shift from the traditional implicit trust to a "zero trust" (ZT) security architecture, as recently proposed by NIST. ZT recommends that there should be "no implicit trust granted to assets or user accounts based solely on their physical or network location (i.e., local area networks versus the internet) or based on asset ownership (enterprise or personally owned)" [64]. Every entity should, by default, be restricted access to the system and must accurately identify and authenticate to access it because any user is a potential threat to a digital system. In this way, ZT might endorse radical BDIDM for any user. After all, "blockchains assume the presence of adversaries in the network by making compromise

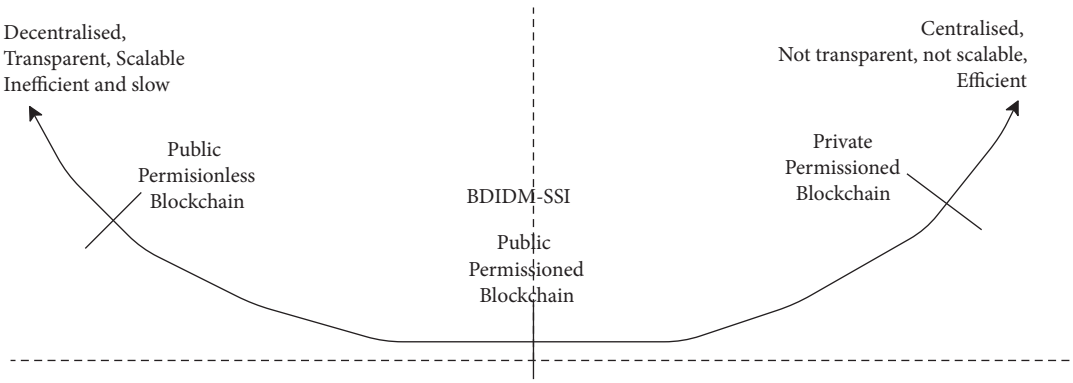


Figure 9: Balancing decentralisation and transparency to achieve efficient blockchains.

significantly expensive," which is why it is claimed to create a trusted system in an untrusted environment [1].

3.5.4. The Practicality of BDIDM in Addressing IDM Challenges in Organisations. SSI critics maintain its impracticality in organisations by highlighting the weakness of the blockchain that dwells at its endpoints [51]. The anonymity of a given blockchain not only means that there is no central authority to block an account in case of identity theft or misbehaviour but also that "each user must themselves safeguard against forgetting (or losing) the private key" [6]. "Blockchain could practically introduce novel issues for users" because they would be the only one "in charge of managing all the cryptographic keys to protect their identity information" [2]. Some researchers even question whether further adoption of blockchain-based solutions should be encouraged and whether the overall potential for change "could be net positive" [65].

However, "reluctance to adopt disruptive technologies may be a significant competitive disadvantage for an organisation, whereas proactive planning can be a significant advantage" [49]. Blockchain represents an opportunity for "a paradigm shift in the development of next-generation cyber defence strategies": first, because blockchain ensures data integrity "as tampering of blockchains is extremely challenging due to the use of a cryptographic data structure and lack of reliance on secrets," second, because "Blockchains assume the presence of adversaries in the network, making a compromise by adversaries significantly expensive," and third, because blockchain "is resilient to single point of failure" [1].

Indeed, those advocating for BDIDM highlight that identity self-management could be beneficial from the privacy-preserving perspective since users have direct control of their own data. Di Francesco Maesa and Mori argue that identity self-management could actually "lead to the practical advantage of reduced expenses" for both users and organisations: users because of "the potential costs of identity theft and private data leaking of traditional centralised solutions" and organisations and external services because they "would not have to store and protect any more private information, nor replicate it among the interested services with the related costs and privacy issues" [2].

The cost savings in password management alone could range in the millions. A Canadian study estimated that "$572 million are lost annually to call centre password management services and lost productive hours" in the country [66]. However, critics might refute cost-saving arguments. They might suggest that the potential cost of data breaches and password management is insufficient to make a case for BDIDM in organisations, assuming that organisations would still prefer to pay those costs than the cost of losing control over users.

Elsewhere, research suggests that "blockchain-based identity and access management systems can address some of the key challenges" associated with the secure cloud [5]. Since the IoT relies on the cloud, the "current centralised cloud model of IoT security" is problematic because "IoT devices are identified, authenticated, and connected through cloud servers" that often perform processing and storage via the Internet. Operations passing through the Internet are subject to manipulation. "Blockchain sovereign identity solutions" can help solve these issues, and some projects and experiments that focus on IoT identity problems are undergoing [31].

A pragmatic point of view would argue that the disruptive capabilities of BDIDM may be beneficial "only in those scenarios where the advantages outweigh the drawbacks" [2]. In other words, when considering a benefit of BDIDM, such as privacy-preserving, one "should question whether it would add value, eliminate a weakness, provide an advantage, or preclude a threat from competitors" [49].

Still, an objective viewpoint would add that more empirical evidence is needed to prove the prevailing argument, since there is more that could impact the likelihood of an organisation to adopt such innovation. The literature suggests some theories that could holistically explain the adoption phenomenon. These theoretical considerations are key in anticipating factors that might predict BDIDM adoption, in this way reconcile views around whether to adopt this innovation in organisations while providinglenses that could be used to further investigate thisphenomenon.

3.6. Theoretical Considerations about the Adoption of BDIDM in Organisation. This subsection analyses how related

theories would shape the adoption of BDIDM in organisations. The section identifies the technology-organisation-environment (TOE) theory as more suitable for explaining this matter than other competing theories. The section ends by proposing a revised version of the TOE theoretical framework, called TOE-BDIDM, as a research model for future empirical studies.

3.6.1. Learning from Related Empirical Studies. Some studies have recently studied the adoption of blockchain technology, mainly in its use case of supply chain management. Unlike the studies of Kamble et al. [67] and Queiroz and Fosso Wamba [68] that were based on individual blockchain adoption, this study considers the enterprise perspective of blockchain adoption like those by Clohessy and Acton [69] and Karamchandani et al. [48]. Nevertheless, all of these studies used one or a combination of the Technology Acceptance Model (TAM), the Theory of Planned Behaviour (TPB), the Unified Theory of Acceptance and Use of Technology (UTAUT), and the Technology Readiness Index (TRI) frameworks.

Since this study focuses on a single blockchain's use case of IDM in the context of an enterprise, the TOE theory seemed appropriate. Initially described by Tornatzky and Fleischer in 1990 as part of "The Processes of Technological Innovation" and lately updated by Jeff Baker in 2011, TOE is a framework that defines enterprise-level theory, explaining how the firm context impacts the adoption of innovation [70].

Unlike some studies limiting the framework to the organisational element only, considering it "the most significant determinant of IT innovation adoption in organisations" [69], this study considers the entire TOE framework. Karamchandani et al. [48] recommended introducing a technological perspective. In addition, the three elements of technology, organisation, and environment constitute a full context of an enterprise. They have been shown to impact, by constraining or promoting, how an organisation "identifies the need, searches, and adopts new technologies" [70].

3.6.2. Technological Context. The technological context consists of an organisation's technologies in use and those existing in the marketplace but not yet adopted. Technologies in use impact the organisation's adoption decision by determining the scope boundaries and the extent to which technological change is needed. Innovations that exist but have not yet been adopted impact the adoption decision-making of the organisation by setting the limits of what is possible and illustrating how technology can enable the organisation to evolve and adapt [70]. Existing technologies such as centralised access control may play a key role in adopting BDIDM as they may not be compatible with a distributed architecture [55]. However, some BDIDM product vendors (such as IBM, KYC-Chain, UniquID, Microsoft, Oracle, etc.) are now available on the market. Organisations can gain some insight into what it could be possible to achieve and what it could not. Baker adds that the innovation's characteristics, that is, the extent of the change it brings, also impact its adoption decision-making. BDIDM is disruptive, a kind of "radical" innovation, as it may render existing IDM and related competencies obsolete. In contrast to innovations that bring incremental or synthetic change, BDIDM does not "introduce new versions of existing technologies" but tends to replace existing centralised IDM systems by "combining existing technologies" in a radically different manner of distributed computing [70]. Blockchain tends to shift the security paradigm by assuming "the presence of adversaries in the network" [1]. Therefore, as part of what Baker describes as "innovations that produce discontinuous change," BDIDM has a high adoption risk. Still, it may have the potential to "enhance competitive standing in an organisation" (232).

From an information security perspective, Hameed and Arachchilage [71] identified additional technology characteristics that impact the adoption of innovation in enterprises, which are also relevant to the adoption of BDIDM: trialability (ease with which the user would adopt/appreciate BDIDM), observability (degree of controllability and monitoring of BDIDM by an organisation), compatibility (ease with which the BDIDM system would interoperate with other systems), and complexity (ease with which an organisation would implement BDIDM). In addition to these, another relevant technological construct is "technical know-how" [72], which includes the availability of skills, consultants, vendors, and so on. However, Baker [70] identifies these items under external environment instead.

3.6.3. Organisational Context. The organisational context consists of firm characteristics and resources that can impact adoption in different ways.

The first is the organisation structure: formal mechanisms linking different units of the organisation (internal boundaries) may promote innovation. Virtually, organisations with an organic and decentralised organisational structure may be suited for the BDIDM adoption phase. Those with formal reporting relationships, centralised decision-making, and clearly defined roles for employees may be the best in the implementation phase [70].

The second is the organisational communication processes, which may either promote or constrain adoption. Support from top management is key to preparing a corporate culture that welcomes change. The support includes describing the role of innovation within the organisation's overall strategy, indicating its importance to subordinates, rewarding initiatives, and building "a skilled executive team" that can cast a compelling firm vision [70]. Regarding BDIDM, since organisations tend to be hostile to privacy, "top management support and organisational readiness are enablers for the adoption of Blockchain" [69].

The third is the organisation's size, considered minor requirements as there have not been many empirical studies that confirm their link to innovation adoption [70]. Instead,

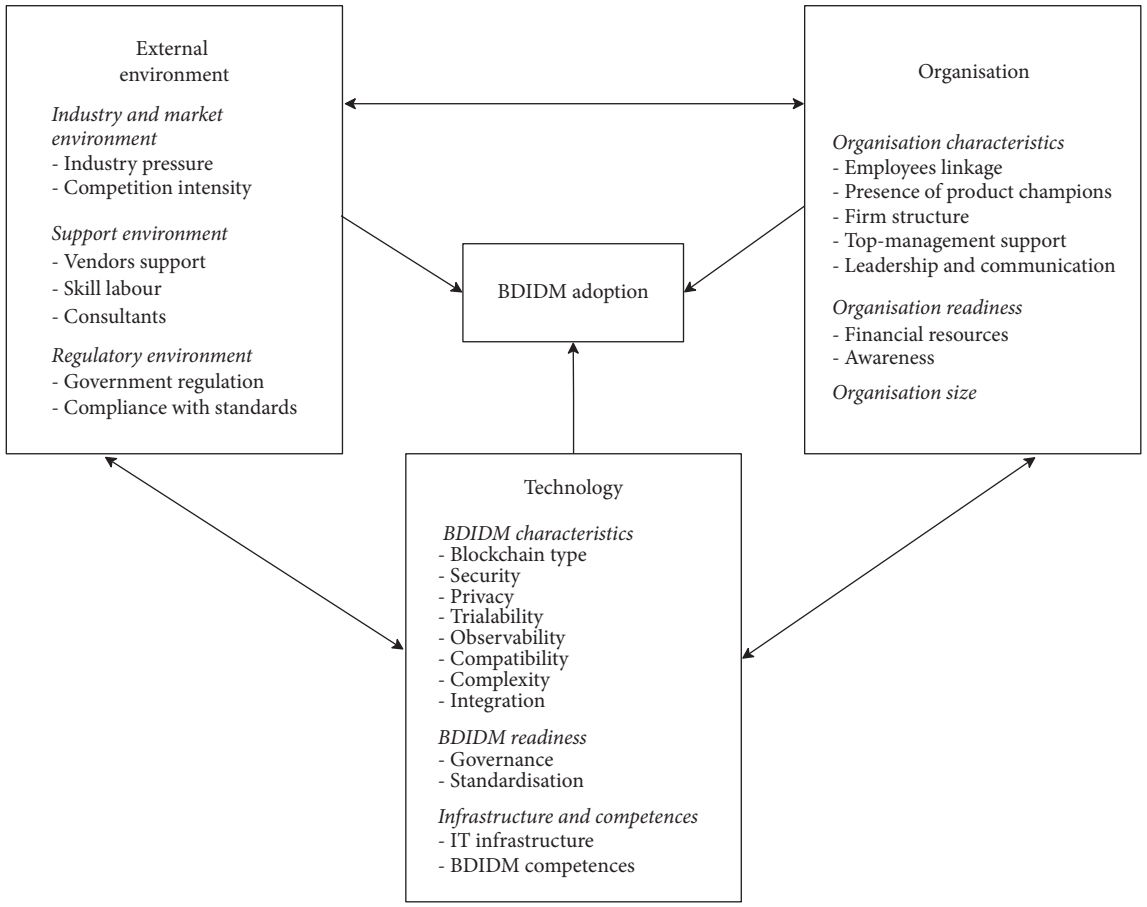


Figure 10: TOE-BDIDM.

the financial cost is reported to have a significant impact. This may be relevant for BDIDM adoption, as BDIDM is perceived to be relatively expensive to implement [49], both in terms of finance and human competencies. However, some studies on blockchain show that large enterprises would be more likely to adopt BDIDM than SMEs [69]. Besides, cultural adaption, awareness, and reluctance to change may also impact the adoption of BDIDM [56].

3.6.4. Environmental Context. The environmental context is all about the industry's structure (such as competition, dominant firms, etc.), whether technology service providers and the regulatory environment (such as government regulations) exist. For instance, the industry life cycle impacts innovation adoption: firms in rapidly growing industries tend to innovate more quickly than those in mature or declining industries. Similarly, the support infrastructure for technology; the availability of skills, labour, and consultants; and government regulation impact adoption [70].

Concerning BDIDM, government regulations in the field of IDM (such as the legal requirement for organisations to protect user privacy, case of POPIA in South Africa), standards (such as codes of best practices, like ISO/IEC [20] and NIST [19]), and cyber-threat landscape could impact BDIDM adoption in organisations [22, 73]. However, blockchain still lacks firm regulatory guidelines and policies for standardisation [49, 59].

3.6.5. The TOE-BDIDM Research Model. Figure 10 illustrates TOE-BDIDM, the proposed research model to empirically investigate the TOE factors impacting the adoption of BDIDM in organisations. TOE-BDIDM is rooted in the TOE theory as described above, a revision of the original model proposed by Baker [70]. The revision aimed to adapt the TOE model to the information security and blockchain contexts. For example, the items "readiness" and "awareness" were added due to the relative newness of the blockchain [49, 56]. Governance and standardisation of the blockchain would also impact the decision to adopt BDIDM in organisations [50]. The literature shaped additional items, including security, privacy, competencies, and skill labour. The BDIDM Type variable was added under BDIDM characteristics to measure the type of blockchain implementation an organisation would prefer for BDIDM adoption.

4. Conclusions

This section synthesises the findings considering the study's objectives and scope introduced earlier. The section also

highlights several knowledge gaps identified in the literature as hints for further research and ends by giving key study's limitations.

This study sought to explore the literature to provide background on the BDIDM as a use case of blockchain. The aim was to understand the topic, mostly how practical the adoption of BDIDM was from an organisational perspective. The study tacitly demonstrated whether the claims made about blockchain, including its potential to address IDM challenges in organisations, were factual. Moreover, the study implicitly showed whether BDIDM was as disruptive for organisations (compared to traditional IDM systems) as assumed.

4.1. Summary of Findings. The main findings could be synthesized as follows:

First, IDM consists of managing matters related to two fundamental information security principles: identification and authentication. Identification labels each entity with an identifier, while authentication allows it to prove they are who they claim to be. IDM is essential because a system should grant access only to legitimate users. IDM can be implemented in two traditional approaches: centralised or federated IDs. A new approach to IDM implementation is distributed IDs (which include the SSI model). The critical challenges of IDM to be addressed include: (i) vulnerabilities in authentication methods, (ii) vulnerabilities in IDM architecture, (iii) the balance between security and privacy, (iv) credential reuse and weak credentials, and (v) secure cloud and secure IoT.

Second, a blockchain is a continuously growing distributed record of updates about a specific matter, such as IDM. A consensus protocol regulates interactions among participants, and the security of data is maintained using cryptography. A blockchain can be implemented in three fundamental ways: public permissionless, public permissioned, and private permissioned. The literature suggests two guidelines to help an enterprise leverage blockchain: Blockchain Technology Transformation Framework and Framework for Evaluation of Blockchain Implementations. When doing so, enterprises should consider, on the one hand, 5 business-promoting factors linked to its features: (i) decentralisation and disintermediation, (ii) programmability and automation, (iii) transparency and auditability, (iv) immutability and verifiability, and (v) integrity, authentication of origin, and trust. On the other hand, 11 business and technological challenges linked to its implementation: (i) software and sustainability, (ii) technical integration, (iii) scalability and efficiency, (iv) security, (v) skill shortage, (vi) complexity, (vii) cost-benefit analysis, (viii) governance, (ix) uncertain regulatory status and lack of standard, (x) cultural adaption and awareness, and (xi) reluctance to change.

Third, blockchain is the underlying technology used to implement a typical distributed IDM system known as SSI. Blockchain does not eliminate vulnerabilities in authentication methods or prevent users from reusing credentials or using weak ones. However, blockchain mitigates the risks linked to vulnerabilities of authentication methods due to cryptography, providing an extra security layer in addition to MFA. Moreover, thanks to its distributed architecture, its decentralized and disintermediated proprieties, blockchain may not have SPOF vulnerability as traditional centralised systems do. BDIDM might also mitigate credential reuse as it allows for ID interoperability among different services, thus significantly reducing the number of accounts per user. Additionally, BDIDM-SSI might better preserve user privacy as it enables them to self-manage their identity data, thus mitigating risks linked to data breaches. Lastly, BDIDM could potentially help achieve secure cloud and secure IoT.

Fourth, an enterprise might implement BDIDM using a public permissioned blockchain to take advantage of blockchain disruption. It turned out that that public permissioned blockchain tends to be ideal for SSI implementation. SSI follows three fundamental principles: (i) security, identity data must be kept secure; (ii) controllability, users must control who can access their data; and (ii) portability, the user must be able to use their identity data wherever they want to. Although a private permissioned blockchain would fit the current enterprise IDM context, it would not differ from the traditional centralised IDs from which one might want to move. A traditional cyber threat theory suggests that the choice of BDIDM implementation should depend on the trust assumptions. NIST highlights the new tendency to shift from this traditional implicit trust to zero-trust security architecture. If widely adopted in organisations, zero trust could enable BDIDM diffusion because it assumes that all users are untrusted, exactly what BDIDM-SSI advocates for. In the meantime, when adopting BDIDM to manage identities in an enterprise, one should consider doing a strength-weaknesses-opportunity-threat analysis according to their business context.

Last, on the debate on whether to adopt BDIDM in organisations, supporters argue that user privacy matters even in an organisational context, which often prioritises security over privacy. Adopting BDIDM-SSI would eliminate the need for organisations to host personal identifiable information on their servers, and in this way, a data breach can be mitigated when the server is compromised. Supporters see the potential of blockchain to mitigate other IDM challenges, including cost-saving on the daily IDM maintenance due to the SSI's identity self-management feature. However, critics of BDIDM would refute this, arguing that organisations would still prefer to pay the cost of corporate IDM than lose control over users. Since empirical evidence is crucial to prove the prevailing argument, the review identified the TOE as more suitable to empirically investigate this matter. The TOE explains how the firm context, in terms of technological, organisational, and environmental contexts, impacts the adoption of innovation such as BDIDM. The TOE model was revised to adapt it to the BDIDM context. Hence, the TOE-BDIDM research model is proposed for further empirical studies.

In summary, most of the claims about blockchain and BDIDM discussed in the study appeared to have some theoretical foundation. This verifies that claims about blockchain, including its potential to address IDM challenges in organisations, are factual rather than just a result of hype.

Therefore, one could conclude that a carefully designed and implemented BDIDM will potentially mitigate IDM challenges, probably reduce the cost related to daily identity maintenance, and possibly decrease data breaches in organisations. Although BDIDM-SSI might not fully make sense to organisations yet, as apparent through the literature discussion, proactive planning instead of ignorance or resistance could avoid potential competitive disadvantages in the future. Ultimately, more research is needed to get blockchain to move from theory to practice by solving real-world issues such as IDM challenges. Hence, the proposed TOE-BDIDM research model is suggested for further studies.

4.2. Gaps in the Literature and Future Research. While reviewing the selected papers, the researchers observed some knowledge gaps at different levels that might inspire future research.

First, there is a lack of blockchain standards, regulations, and guidelines. Some studies [47, 49] have partially addressed the guidelines aspects. However, more studies are needed to fill in the gap of blockchain standardisation, as it seems to be one of the potential precursors of its adoption and diffusion in organisations.

Second, most papers retrieved about nonfinancial blockchain are either generic or mainly focused on the supply chain use case. The few materials dedicated to blockchain IDM specifically discussed the topic from the perspective of IoT (identification and authentication of smart devices on the Internet), cloud computing perspective (ID-as-a-service), or the individual adoption (adoption of blockchain ID by individuals for Internet use). Very few included or were about the enterprise perspective.

Third, most of the retrieved papers about the IDM use case of blockchain are conceptual than empirical. Empirical studies on blockchains are still rare, partially justified by the newness of blockchain. Although conceptual works are equally important, more should be done, including investigating BDIDM through empirical studies.

Last, of the empirical studies on blockchain retrieved, none was about blockchain-based identity management. In addition, they all used one or a combination of TAM, TPB, UTAUT, and TRI. Researchers found only one study that included only one construct of the TOE theory. Additionally, none of them had tested the TOE theory quantitatively. Some used TOE with qualitative methods [69], while others used quantitative methods with different theories [68].

4.3. Limitations. This literature review is not perfect. The principal limitation was that not all potential papers were included in the sample. First, because of the diversity in blockchain applications and the high interest resulting in hundreds of articles published mainly in the last free years from the time of writing. There review needed to stay as focused on the topic as possible. Second, because the topic involves various concepts from both IDM and blockchain, the study tried to limit the sample strictly to the scope of the review. Hence, some papers were excluded though they were satisfactory to some selection criteria. However, researchers were confident they saturated the topic because there was a repetition of what had already been lent.

This literature review may not, on its own, be sufficient to make a case for BDIDM adoption in organisations. As far as its objective is concerned, it gives the background to understand the topic while inspiring further empirical investigations.

Data Availability

This research used secondary data: journal articles, conference papers, books, reports, patents, and standards. These are listed in the reference section, and most of them are accessible on common academic databases, including EBSCOhost and Google Scholar.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The authors would like to acknowledge Professor Michael Kyobe, Department of Information Systems at the University of Cape Town, for his guidance at the earlier stage of the drafting of this work. The authors also appreciate their families and friends' support during the drafting process.

References

[1] S. Shetty, C. A. Kamhoua, and L. L. Njilla, *Blockchain for Distributed Systems Security*, John Wiley & Sons, Hoboken, New Jersey, United States, 2019.

[2] D. Di Francesco Maesa and P. Mori, "Blockchain 3.0 applications survey," *Journal of Parallel and Distributed Computing*, vol. 138, pp. 99–114, 2020.

[3] P. Musuva-Kigen, F. Mueni, and D. Ndegwa, *Africa Cyber Security Report 2016*, Serianu Cyber Threat Intelligence Team, Nairobi, Kenya, 2016.

[4] IBM-Security, "IBM: cost of a data breach report," *Computer Fraud & Security*, vol. 2019, no. 8, p. 4, 2019.

[5] N. Kshetri, "Blockchain's roles in strengthening cybersecurity and protecting privacy," *Telecommunications Policy*, vol. 41, no. 10, pp. 1027–1038, 2017.

[6] M. Kuperberg, "Blockchain-based identity management: a survey from the enterprise and ecosystem perspective," *IEEE Transactions on Engineering Management*, vol. 67, no. 4, pp. 1008–1027, 2019.

[7] J. Bernal Bernabe, J. L. Canovas, J. L. Hernandez-Ramos, R. Torres Moreno, and A. Skarmeta, "Privacy-preserving solutions for blockchain: review and challenges," *IEEE Access*, vol. 7, pp. 164908–164940, 2019.

[8] J. Kolb, M. AbdelBaky, R. H. Katz, and D. E. Culler, "Core concepts, challenges, and future directions in blockchain," *ACM Computing Surveys*, vol. 53, no. 1, pp. 1–39, 2020.

[9] D. Finfgeld-Connett, *A Guide to Qualitative Meta-Synthesis*, Routledge, New York, NY, 2018.

[10] D. Walsh and S. Downe, "Meta-synthesis method for qualitative research: a literature review," *Journal of Advanced Nursing*, vol. 50, no. 2, pp. 204–211, 2005.

[11] G. Oosterwyk, I. Brown, and S. Geeling, "A synthesis of literature review guidelines from information systems journals," *Proceedings of 4th International Conference on the*, vol. 12, pp. 250–260, 2019.

[12] O. Ngwenyama, "The ten basic claims of information systems research: an approach to interrogating validity claims in scientific argumentation," *SSRN Electronic Journal*, pp. 1–40, 2019.

[13] D. Chakravarty and T. Deshpande, "Blockchain-enhanced identities for secure interaction," in *Proceedings of the 2018 IEEE International Symposium on Technologies for Homeland Security (HST)*, pp. 1–4, IEEE, Crystal City, VA, USA, May 2018.

[14] M. E. Whitman and H. J. Mattord, *Principles of Information Security*, Cengage Learning, Boston, Massachusetts, US, 2018.

[15] K. Marky, P. Mayer, N. Gerber, and V. Zimmermann, "Assistance in daily password generation tasks," in *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*, pp. 786–793, Singapore, Singapore, October 2018.

[16] M. A. Kiran, P. Yogeshwari, K. V. Bhavani, and T. Ramya, "Biometric authentication: a holistic review," in *Proceedings of the 2018 2nd International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC) I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*, pp. 428–433, IEEE, Palladam, India, August 2018.

[17] T. Seitz, F. Mathis, and H. Hussmann, "The bird is the word: a usability evaluation of emojis inside text passwords," in *Proceedings of the 29th Australian Conference on Computer-Human Interaction*, pp. 10–20, Brisbane, Queensland, Australia, November 2017.

[18] L. Xiaofeng, Z. Shengfei, and Y. Shengwei, "Continuous authentication by free-text keystroke based on CNN plus RNN," *Procedia computer science*, vol. 147, pp. 314–318, 2019.

[19] W. A. Hufstetler, M. J. H. Ramos, and S. Wang, "Nfc unlock: secure two-factor computer authentication using nfc," in *Proceedings of the 2017 IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, pp. 507–510, IEEE, Orlando, FL, USA, October 2017.

[20] *South African National Standard: Information Technology — Security Techniques — Code of Practice for Information Security Controls*, ISO/IEC, Switzerland, 2014.

[21] S. Pranata and H. T. Nugroho, "2FYSH: two-factor authentication you should have for password replacement," *Telkomnika*, vol. 17, no. 2, pp. 693–702, 2019.

[22] Y. Liu, G. Sun, and S. Schuckers, "Enabling secure and privacy preserving identity management via smart contract," in *Proceedings of the 2019 IEEE Conference on Communications and Network Security (CNS)*, pp. 1–8, IEEE, Washington, D.C, USA, June 2019.

[23] T. G. Rauscher, "Raid system with multiple controllers and proof against any single point of failure," Google Patents, 2005.

[24] B. Feng, C. Huang, and X. Gong, "Distributed storage method, apparatus, and system for reducing a data loss that may result from a single-point failure," Google Patents, 2014.

[25] D. Dresher, *Blockchain Basics*, Apress, Frankfurt, 2017.

[26] E. Karanja and M. A. Rosso, "The chief information security officer: an exploratory study," *Journal of International Technology and Information Management*, vol. 26, no. 2, pp. 23–47, 2017.

[27] J. Breuer, H. Ranaivoson, U. Buchinger, and P. Ballon, "Who manages the manager? Identity management and user ownership in the age of data," in *Proceedings of the 2015 13th Annual Conference on Privacy, Security and Trust (PST)*, pp. 22–27, IEEE, Izmir, Turkey, July 2015.

[28] D. Baars, *Towards Self-Sovereign Identity Using Blockchain Technology*, University of Twente, Enschede, Netherlands, 2016.

[29] M. Romney, P. Steinbart, J. Mula, R. McNamara, and T. Tonkin, *Accounting Information Systems Australasian Edition*, Pearson Higher Education AU, Australia, 2012.

[30] A.-S. Shehu, A. Pinto, and M. E. Correia, "Privacy preservation and mandate representation in identity management systems," in *Proceedings of the 2019 14th Iberian Conference on Information Systems and Technologies (CISTI)*, pp. 1–6, IEEE, Coimbra, Portugal, June 2019.

[31] X. Zhu and Y. Badr, "A survey on blockchain-based identity management systems for the Internet of Things," in *Proceedings of the 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pp. 1568–1573, IEEE, Halifax, NS, Canada, July 2018.

[32] R. Charanya and M. Aramudhan, "Survey on access control issues in cloud computing," in *Proceedings of the 2016 International Conference on Emerging Trends in Engineering, Technology and Science (ICETETS)*, pp. 1–4, IEEE, Pudukkottai, India, February 2016.

[33] J. K. Mwenya and I. Brown, "Cloud privacy and security issues beyond technology: championing the cause of accountability," in *Proceedings of the The 30th Australiasian Conference on Information Systems (ACIS)*, Perth, Western Australia, December 2019.

[34] K. Bendiab, N. Kolokotronis, S. Shiaeles, and S. Boucherkha, "WiP: a novel blockchain-based trust model for cloud identity management," in *Proceedings of the 2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech)*, pp. 724–729, IEEE, Athens, August 2018.

[35] X. Ma, "Managing Identities in Cloud Computing Environments," in *Proceedings of the 2015 2nd International Conference on Information Science and Control Engineering*, pp. 290–292, IEEE, Shanghai, China, April 2015.

[36] F. F. Moghaddam, P. Wieder, and R. Yahyapour, "A policy-based identity management schema for managing accesses in clouds," in *Proceedings of the 2017 8th International Conference on the Network of the Future (NOF)*, pp. 91–98, IEEE, London, UK, November 2017.

[37] N. Sohrabi, X. Yi, Z. Tari, and I. Khalil, "BACC: blockchain-based access control for cloud data," in *Proceedings of the Australasian Computer Science Week Multiconference*, pp. 1–10, Melbourne, VIC, Australia, February 2020.

[38] D. Alexander, A. Finch, D. Sutton, and A. Taylor, *Information Security Management Principles*, Third Edition ed. edition, 2020.

[39] N. Mpofu and W. J. van Staden, "Evaluating the severity of trust to identity-management-as-a-service," in *Proceedings of the 2017 Information Security for South Africa (ISSA)*, pp. 83–89, IEEE, 54 on Bath Hotel, Rosebank, Johannesburg, South Africa, August 2017.

[40] S. Michael and Z. J. Anna, "An identity provider as a service platform for the edugain research and education community," in *Proceedings of the 2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, pp. 739-740, IEEE, Washington, DC, USA, April 2019.

[41] M. A. Bouras, Q. Lu, F. Zhang, Y. Wan, T. Zhang, and H. Ning, "Distributed ledger technology for eHealth identity privacy: state of the art and future perspective," *Sensors*, vol. 20, no. 2, p. 483, 2020.

[42] A. Grüner, A. Mühle, and C. Meinel, "An integration architecture to enable service providers for self-sovereign identity," in *Proceedings of the 2019 IEEE 18th International Symposium on Network Computing and Applications (NCA)*, pp. 1–5, IEEE, Cambridge, MA, USA, September 2019.

[43] A. R. Thota, P. Upadhyay, S. Kulkarni, P. Selvam, and B. Viswanathan, "Software wallet based secure participation in hyperledger fabric networks," in *Proceedings of the 2020 International Conference on COMmunication Systems & NETworkS (COMSNETS)*, pp. 1–6, IEEE, Bengaluru, India, January 2020.

[44] H. Kim, S.-H. Kim, J. Y. Hwang, and C. Seo, "Efficient privacy-preserving machine learning for blockchain network," *IEEE Access*, vol. 7, pp. 136481–136495, 2019.

[45] R. Post, K. Smit, and M. Zoet, "Identifying factors affecting blockchain technology diffusion," in *Proceedings of the Americas Conference on Information Systems (AMCIS 2018)*, New Orleans LA, USA, August 2018.

[46] E. Politou, F. Casino, E. Alepis, and C. Patsakis, "Blockchain mutability: challenges and proposed solutions," *IEEE Transactions on Emerging Topics in Computing*, p. 1, 2019.

[47] O. Labazova, "Towards a framework for evaluation of blockchain implementations," in *Proceedings of the International Conference on Information Systems*, ICIS, Munich, Germany, December 2019.

[48] A. Karamchandani, S. K. Srivastava, and R. K. Srivastava, "Perception-based model for analyzing the impact of enterprise blockchain adoption on SCM in the Indian service industry," *International Journal of Information Management*, vol. 52, Article ID 102019, 2020.

[49] M. Demir, O. Turetken, and A. Mashatan, "An enterprise transformation guide for the inevitable blockchain disruption," *Computer*, vol. 53, no. 6, pp. 34–43, 2020.

[50] B.-J. Butijn, D. A. Tamburri, and W.-J. v. d. Heuvel, "Blockchains," *ACM Computing Surveys*, vol. 53, no. 3, pp. 1–37, 2020.

[51] P. Helebrandt, M. Bellus, M. Ries, I. Kotuliak, and V. Khilenko, "Blockchain adoption for monitoring and management of enterprise networks," in *Proceedings of the 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pp. 1221–1225, IEEE, UBC, Vancouver , BC , Canada, 2018.

[52] N. El Madhoun, J. Hatin, and E. Bertin, "Going beyond the blockchain hype: in which cases are blockchains useful for it applications?" in *Proceedings of the 2019 3rd Cyber Security in Networking Conference (CSNet)*, pp. 21–27, IEEE, November 2019.

[53] K. Wüst and A. Gervais, "Do you need a blockchain?" in *Proceedings of the 2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*, pp. 45–54, IEEE, Zug, Switzerland, June 2018.

[54] M. Ahmed, I. Elahi, M. Abrar, U. Aslam, I. Khalid, and M. A. Habib, "Understanding blockchain: platforms, applications and implementation challenges," in *Proceedings of the 3rd International Conference on Future Networks and Distributed Systems*, pp. 1–8, Paris France, July 2019.

[55] A. Marsalek, C. Kollmann, T. Zefferer, and P. Teufl, "Unleashing the full potential of blockchain technology for security-sensitive business applications," in *Proceedings of the 2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pp. 394–402, IEEE, Seoul, South Korea, May 2019.

[56] N. Upadhyay, "Demystifying blockchain: a critical analysis of challenges, applications and opportunities," *International Journal of Information Management*, vol. 54, Article ID 102120, 2020.

[57] Q. T. Thai, J.-C. Yim, and S.-M. Kim, "A scalable semi-permissionless blockchain framework," in *Proceedings of the 2019 International Conference on Information and Communication Technology Convergence (ICTC)*, pp. 990–995, IEEE, Jeju Island, South Korea, October 2019.

[58] E. Fernando, "Essential blockchain technology adoption factors in pharmaceutical industry," in *Proceedings of the 2019 4th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, pp. 523–526, IEEE, Yogyakarta, Indonesia, November 2019.

[59] P. T. Duy, D. T. T. Hien, D. H. Hien, and V.-H. Pham, "A survey on opportunities and challenges of Blockchain technology adoption for revolutionary innovation," in *Proceedings of the Ninth International Symposium on Information and Communication Technology*, pp. 200–207, Danang City, Viet Nam, December 2018.

[60] P. G. Lopez, A. Montresor, and A. Datta, "Please, do not decentralize the Internet with (permissionless) blockchains," in *Proceedings of the 2019 IEEE 39th International Conference on Distributed Computing Systems*, pp. 1901–1911, IEEE, Dallas, TX, USA, July 2019.

[61] Z. Cui, F. Xue, S. Zhang et al., "A hybrid BlockChain-based identity authentication scheme for multi-WSN," *IEEE Transactions on Services Computing*, vol. 13, no. 2, pp. 241–251, 2020.

[62] T. Mitani and A. Otsuka, "Traceability in permissioned blockchain," *IEEE Access*, vol. 8, pp. 21573–21588, 2020.

[63] F. Buccafurri, G. Lax, A. Russo, and G. Zunino, "Integrating digital identity and blockchain," in *Proceedings of the OTM Confederated International Conferences on the Move to Meaningful Internet Systems*, pp. 568–585, Springer, Valletta, Malta, October 2018.

[64] V. Stafford, "Zero Trust Architecture," *NIST Special Publication*, vol. 800, p. 207, 2020.

[65] A. Rot and B. Blaicke, "Blockchain's future role in cybersecurity. analysis of defensive and offensive potential leveraging blockchain-based platforms," in *Proceedings of the 2019 9th International Conference on Advanced Computer Information Technologies (ACIT)*, pp. 447–451, IEEE, Ceske Budejovice, Czech Republic, June 2019.

[66] G. Wolfond, "A blockchain ecosystem for digital identity: improving service delivery in Canada's public and private sectors," *Technology Innovation Management Review*, vol. 7, no. 10, 2017.

[67] S. Kamble, A. Gunasekaran, and H. Arha, "Understanding the Blockchain technology adoption in supply chains-Indian context," *International Journal of Production Research*, vol. 57, no. 7, pp. 2009–2033, 2019.

[68] M. M. Queiroz and S. Fosso Wamba, "Blockchain adoption challenges in supply chain: an empirical investigation of the main drivers in India and the USA," *International Journal of Information Management*, vol. 46, pp. 70–82, 2019.

[69] T. Clohessy and T. Acton, "Investigating the influence of organizational factors on blockchain adoption: an innovation theory perspective," *Industrial Management & Data Systems*, vol. 119, no. 7, pp. 1457–1491, 2019.

[70] J. Baker, "The technology-organization-environment framework," *Information Systems Theory*, vol. 28, pp. 231–245, 2012.

[71] M. A. Hameed and N. A. G. Arachchilage, "A conceptual model for the organizational adoption of information system security innovations," in *Security, Privacy, and Forensics Issues in Big Data*, pp. 317–339, IGI Global, Pennsylvania, United States, 2020.

[72] H. O. Awa, O. Ukoha, and B. C. Emecheta, "Using T-O-E theoretical framework to study the adoption of ERP solution," *Cogent Business & Management*, vol. 3, no. 1, Article ID 1196571, 2016.

[73] P. Grassi, *Digital Identity Guidelines: Enrollment and Identity Proofing*, National Institute of Standards and Technology, Gaithersburg, MD, US, 2017.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 9501572, 20 pages
https://doi.org/10.1155/2021/9501572

Research Article

Smart Grid Nontechnical Loss Detection Based on Power Gateway Consortium Blockchain

Xudong He, Jian Wang, Jiqiang Liu, Enze Yuan, Kailun Wang, and Zhen Han

Beijing Jiaotong University, Beijing, China

Correspondence should be addressed to Jian Wang; wangjian@bjtu.edu.cn

Received 17 June 2021; Revised 5 August 2021; Accepted 27 September 2021; Published 14 October 2021

Academic Editor: Yinghui Zhang

The rapid development of the smart grid brings convenience to human beings. It enables users to know the real-time power supply capacity, the power quality, and the electricity price fluctuation of the grid. However, there are still some threats in the smart grid, which increase all kinds of expenses in the grid and cause great trouble to energy distribution. Among them, the man-made nontechnical loss (NTL) problem is particularly prominent. Recently, there are also some NTL detection programs. However, most of the schemes need huge amounts of supporting data and high labor costs. As a result, the NTL problem has not been well solved. In order to better avoid these risks, problems such as tampering of smart meter energy data, bypassing the smart meter directly connected to the grid, and imbalance between revenue and expenditure of the smart grid are tackled, and the threat scene of NTL is constructed. A hierarchical grid gateway blockchain is proposed and designed, and a new decentralized management MDMS system is constructed. The intelligent contract combined with the elliptic curve encryption technology is used to detect the storage and the acquisition of power data, and the detection of NTL problems is realized. At the same time, it has a certain ability to resist attacks such as replay, monitoring, and tampering. We tested the time consumption and throughput of this method on Hyperledger Fabric. At the same time, eight indexes of other methods proposed in the literature are compared. This method has a good effect.

1. Introduction

The concept of smart grid was put forward in 2003, and the "Smart Grid Technology Forum" was established by the European Union in 2005. The smart grid is essentially a modern transmission network. It uses information and communication technology to adjust the production, transmission, and distribution of electric power [1], to achieve the purpose of saving energy, reducing loss, and enhancing the reliability of the power grid. The smart grid can realize the two-way communication of information the services [2, 3]. The smart meter in the smart grid not only has the basic measurement function but also has more abundant functions, such as communication function. In order to adapt to the use of modern smart grid and new energy, it is also equipped with a storage module and a calculation module, which can store electricity consumption information and the two-way ladder rate metering function, and also provides a control interface that can be remotely controlled, as well as intelligent functions such as electricity theft prevention. In the smart grid, Advanced Metering Infrastructure (AMI) system is used for intelligent management. AMI system is mainly composed of smart meter, communication system and equipment, and Meter Database Management System (MDMS).

While the smart grid brings advantages, for example, intelligent power grid management, it is also faced with extremely serious threats, which are mainly divided into natural threats and man-made threats. Among the many threats, the most common is that power thieves or power users deceive power companies through a series of ways and then bring nontechnical loss to the entire smart grid. NTL refers to the remaining part of the loss of power transmission and distribution that cannot be explained by technology after excluding TL. Abnormal electricity consumption behaviors such as electricity theft are the main cause of NTL

[4]. According to statistics, in countries such as India, Brazil, China, and the United States, the loss of power supply caused by power theft is more than 25%. In recent years, not only is the phenomenon of electricity theft becoming more and more serious, but also the electricity theft methods used by electricity theft users are more and more various, and means of electricity theft are becoming more and more sophisticated. In addition to the traditional power theft methods, such as the undervoltage method and undercurrent method [5], there are also high-tech methods of electricity thefts, such as strong magnetic interferences, power thefts from high-frequency power supply, and network attacks on intelligent meters or data centers [6]. The behavior of electricity thefts is becoming more and more technically sophisticated. It can be seen that, in the past, the means that users relied on to steal electricity, such as destroying traditional electricity meters or private power lines, have been transformed into attacks on smart meters through digital storage technology and network communication technology [7]. The attack is to reduce the corresponding time power consumption or directly return it to zero through data tampering, in order to reduce the electricity bill payable.

In the operation of the power grid, nontechnical losses will cause a large number of energy and economic losses, and the uncertainty of power theft behavior will directly affect the load supply and demand balance of the power grid and interfere with the stability of the power system. Therefore, it is of great practical significance to analyze power consumption data and to detect electricity theft behavior [8]. In response to the aforementioned nontechnical power loss problem, much related work has been done which can be divided into the following three categories: (1) Physical detection solutions include the use of physical solutions to prevent and detect electricity theft. These physical solutions include routine inspections, sensor monitoring, camera monitoring, and drone monitoring. (2) The NTL fraud detector based on machine learning algorithms mainly uses machine learning technology to establish a detection model to identity electricity theft. However, the training dataset of the nontechnical power loss detection model requires power experts to mark the attack data in the power dataset; thus, the cost is high. In addition, because the power theft against smart grids will bring huge economic benefits to attackers, the diversity of related attack behaviors increases. The feature extraction becomes more and more difficult, and the inaccuracy of features directly leads to the high accuracy of detection models. The reduction in magnitude has led to huge economic losses in the power system. (3) Based on the comparison method, this kind of scheme usually adopts a safe and reliable central instrument to measure the abnormal situation and compare it with other suspicious instruments. These schemes are usually lightweight and flexible, but existing schemes can only detect NTL fraud with small datasets.

Therefore, even if there are some detection schemes for NTL attacks, we still need to explore other more effective solutions. The study is aimed at the NTL problem in the smart grid and develop a detection plan from the MDMS in the AMI system. We designed a smart grid NTL problem protection scheme based on the power gateway consortium blockchain. The scheme can solve the problems such as the difficulty of state detection of smart meters, the difficulty of smart meter access authentication, and the insecurity of hierarchical management of power transactions. We use power data and meter status data to detect NTL. It has a good detection effect on smart meter data tampering and power theft caused by users directly connected to the power grid. It is used to solve the problems caused by NTL in the smart grid.

The main contributions of this paper are as follows:

(1) The scheme proposed in this paper can effectively resist replay attacks, surveillance attacks, man-in-the-middle attacks, and witch attacks.

(2) This paper stores the electric energy information and the state of the smart meter in the MDMS system, and adopts the storage mode of the edge network blockchain to store the user's smart meter status and the user payment information, which is used for NTL audit and accountability.

(3) This paper proposes the NTL threat scenario, which detects NTL based on the edge network blockchain, and uses the blockchain technology to ensure that the data cannot be tampered with. The detection method does not rely on a large amount of data to train the model but on smaller user power consumption data.

The rest of the paper consists of the following sections. Section 2 introduces the related research work of blockchain technology and the NTL detection technology. Section 3 proposes a smart grid NTL detection scheme based on the power network association chain, including the overall structure, client registration, and data encryption and decryption transmission. Section 4 demonstrates the experiment and the experimental results as well as the comparison. Section 5 analyzes the security and threat scenarios of the overall scheme. Section 6 gives the research results and discussion.

2. Related Work

This section will summarize the existing work; we first summarize the related work of NTL detection in smart grid, then investigate the important role of blockchain technology in the smart grid, and finally summarize the related detection technology of blockchain to illustrate the feasibility of smart grid NTL detection scheme based on the gateway blockchain.

2.1. Smart Grid NTL Detection. Nowadays, with the development of smart, integrated, and interconnected power grids, to achieve the goal of reliability, security, and cost-effectiveness of the power grid and to prevent the occurrence of power theft incidents, the NTL detection technology and related research are gradually developing. Leite et al. [9] proposed a strategy for detecting nontechnical losses using a multivariate control chart, which establishes a reliable area

to monitor the measured variance. After detecting the nontechnical loss, the pathfinding program based on the algorithm can find the consumption point of the nontechnical loss. Jeyaraj et al. [10] put forward a multidimensional deep learning algorithm to learn and classify nonperiodical electricity and then can detect user theft of electricity from the periodic load curve. The weekly load pattern and daily load pattern are both processed as 2D power data samples. Saeed et al. [11] suggested an efficient classification method based on the BoostingC5.0 decision tree to detect nontechnical losses in electric utilities. First, extract data features from the dataset to distinguish honest from fraudulent customers. Afterward, Pearson's chi-square feature selection algorithm is used to select the most relevant feature among the extracted features. Finally, use the BoostedC5.0 decision tree (DT) algorithm to classify honest consumers and fraudsters based on the results of the selected functions. Viegas et al. [12] mentioned a clustering-based method to detect power theft. By clustering the collected data, typical consumer behavior prototypes can be extracted. If the distance between a new data sample and a typical consumer prototype is too large, the distance-based novelty detection framework will classify it as vicious data. Okino Otuoze et al. [13] put forward a power theft detection framework based on a general predictive algorithm. The framework uses universal anomaly detection (UAD) based on the Lempel-Ziv universal compression algorithm, which can realize real-time detection in the smart grid environment. It detects anomalies by monitoring many network parameters, including monitoring energy consumption data, the change rate of energy consumption data, and date stamps as well as time stamps. Blazakis et al. [14] introduced an adaptive neuro-fuzzy inference system (ANFIS) for power theft detection. The results show that if the technology is correctly applied, it can achieve a high detection success rate in the case of fraudulent activities caused by unauthorized energy use.

Given the NTL problem in the smart grid, the above detection methods have played a certain role, but a few of them require a large amount of data, and the calculation method is complex. It poses a serious threat to the privacy and security of power-related data. We explore new technologies to solve the NTL problem by investigating the application of blockchain in the smart grid.

2.2. Application of Blockchain in Smart Grid. In the smart grid system, various network transpositions require a large amount of data sharing and exchanges between gateways. At the same time, information exchanges between power suppliers and individual consumers are also very frequent; therefore if the power system encounters network security threats, it will cause huge losses. Blockchain technology has the characteristics of decentralization, openness, transparency, and nontamperability; realizes the collaborative trust and concerted actions between multiple subjects; and is widely used in the construction of smart grids. Gai et al. [15] suggested an alliance blockchain method to solve the privacy leakage problem of energy transaction users in smart grids without restricting transaction functions. This method also can detect the relationship between it and other information (such as physical location and energy usage) by mining various energy transaction volumes. Guan et al. [16] put forward a blockchain-based smart grid data aggregation privacy protection scheme, which divides users into different groups, and each group has a private blockchain to record the data of its members. The scheme uses pseudonyms to hide the identity of users. Each user can create multiple pseudonyms and associate their data with different pseudonyms. However, this scheme also only conducts a single-dimensional data collection, and the user power data in the same area is transmitted in plain text, posing a great security risk. Pop et al. [17] used blockchain technology to design a demand-side response model for distributed management of energy networks. The model uses tamper-proof blockchain technology to store energy consumption data collected from the IoT smart meter. At the same time, the automatically executed smart contract defines the expected energy loss of each producer and each consumer in a programmatic way and then realizes it. In order to match the production and demand of the smart grid. Gao et al. [18] put forward a smart grid monitoring method based on a secure sovereign blockchain and also implemented a smart contract. The contract executes the established procedures and then provides a network-based trusted system. The system proved to be very effective because users can monitor how the electricity is used, and it also provides a platform that no one needs to manipulate.

Through the investigation of related work, there are many applications of blockchain technology in the smart grid, less research working on NTL detection and, some problems such as information sharing; thus, we also investigate the scheme of abnormal problem detection of blockchain in our paper.

2.3. Smart Grid Combined with Blockchain-Related Work. Blockchain technology is also used in the industrial Internet of things scenarios [19]. In response to the problem of abnormality detection in the smart grid, the blockchain can realize the cooperative trust between different information interaction parts through "smart contracts" and efficiently detect abnormal situations.

Li et al. [20] mentioned a blockchain-based method for detecting abnormal electricity consumption in smart grids, aiming to use sensor processing, smart meter readings, machine learning, and blockchain to accurately and timely detect electricity consumption abnormality.

Signorini et al. [21] proposed a blockchain-based anomaly detection method (BAD). BAD is a complete framework that relies on several components that utilize its core blockchain metadata to collect potentially malicious activities. BAD avoids any central point of failure and can prevent malware from deleting or changing its own traces.

Golomb et al. [22] mentioned a lightweight framework CIoTA, which uses the concept of blockchain to perform distributed and collaborative anomaly detection on devices

with limited resources. Through the consensus between proof and IoT devices, CIoTA uses the blockchain to gradually update the reliable anomaly detection model.

Casado-Vara et al. [23] suggested a new system for detecting fraud based on blockchain. The blockchain is used to store the data of the distribution network monitored by the WSN and apply the created clustering algorithm to detect fraud. Whenever the blockchain grows, the stored data is more secure. Therefore, the power company can check the stored blockchain data. It is proved that blockchain technology has a certain effect on abnormal problem detection.

Through the above research and analysis, it is found that, with the development of the smart grid, the interaction between power suppliers and users becomes more convenient. At the same time, due to the application of various intelligent devices and the generation of corresponding massive data and information, problems such as Internet security and power theft continue to appear in the power grid system. Aiming for the problem of NTL, several scholars have also proposed a detection scheme, but the scheme has some problems, such as the large demand for data and the need for data concentration. Moreover, data privacy and security cannot be guaranteed and are high labor costs. Therefore, combined with the blockchain technology, this paper proposes a smart grid power theft detection model based on the power network association chain, which gives full play to the dispersion, openness, transparency, and tamper-proof of the blockchain technology, and applies it to the smart grid NTL detection problem.

3. Smart Grid NTL Detection Based on Power Gateway Consortium Blockchain

Through the investigation of related work, we found that the smart grid has problems of NTL caused by the tampering of the electricity data of the smart meter at the home network layer, NTL caused by bypassing the smart meter and directly connected to the grid network, and difficulty in detecting the imbalance of smart grid revenue and expenditure. Based on the edge of the smart grid network, we designed a smart grid NTL problem protection program based on the power gateway blockchain. We first introduced the smart grid gateway consortium blockchain structure and described the threat model scenarios of NTL in the smart grid. Finally, a smart grid NTL detection model and detection method based on the power gateway consortium blockchain are proposed in Section3.3. In the detection method, the smart meter registration, online data storage and query, data structure, consensus, and detection process are introduced in detail.

3.1. Smart Grid Gateway Consortium Blockchain Structure. The smart grid gateway consortium blockchain structure consists of two parts, including the power infrastructure network and the power communication network. The power communication network includes three levels: wide-area network (WAN), local area network (LAN), and home network (HAN). The WAN consortium blockchain network consists of LAN power gateways, and each LAN power gateway node includes multiple LAN consortium blockchain networks. The LAN consortium blockchain network is composed of HAN power gateways, and each HAN power gateway node includes multiple HAN networks. The specific structure is shown in Figure 1.

Definition 1. Power infrastructure network.

The basic network of power facilities includes the basic equipment in the traditional power grid, such as power generation facility, power transmission stations, and substation/distribution stations. After generating electricity from the power generation facility, the process of voltage boosting, transmission, and the step-down is carried out, and finally, the electricity is sold to the users by the distribution station. It provides a guarantee for the production, transmission, and use of electric energy.

Definition 2. Electric power communication network.

The electric power communication network is composed of three types of network structures, including HAN, LAN, and WAN. Each layer of the network structure includes power gateway equipment for data aggregation and network communication as shown in Table 1.

The blockchain structure of HAN, LAN and WAN, grid gateway, and smart meter in the power communication network is shown in Figure 2.

The electric power communication network is divided into HAN, LAN, and WAN according to the communication range from small to large. The three are inclusive ($\mathrm{HAN} \subset \mathrm{LAN} \subset \mathrm{WAN}$). Among them, the HAN network includes HAN power gateways, smart meters, and various home electrical equipment. Electrical equipment gathers power consumption information in smart meters, which are connected to the HAN power gateway. Here, we define multiple HAN networks as $\mathrm{HAN}_1, \mathrm{HAN}_2 \ldots \mathrm{HAN}_N$. LAN network is composed of multiple HAN networks, namely, $\mathrm{LAN} = \{\mathrm{HAN}_1 \cup \mathrm{HAN}_2 \ldots \mathrm{HAN}_N\}$. In the LAN network, the HAN power gateway is used as a node to form a LAN network consortium blockchain. Similarly, the WAN network consists of multiple LAN networks, namely, $\mathrm{WAN} = \{\mathrm{LAN}_1 \cup \mathrm{LAN}_2 \ldots \mathrm{LAN}_N\}$. In the WAN network, the LAN power gateway is used as a node to form a WAN network consortium blockchain.

3.2. Threat Scenario. The user is the smallest unit in the smart grid scenario and is divided into malicious users and normal users. The malicious user is the core threat that causes nontechnical power loss in the smart grid. Based on the behavior and distribution characteristics of malicious users, this paper divides the threats of malicious users into three categories: active malicious user threats, passive malicious user threats, and group malicious user threats. The specific scenarios of the three different threats will be introduced one by one as follows:

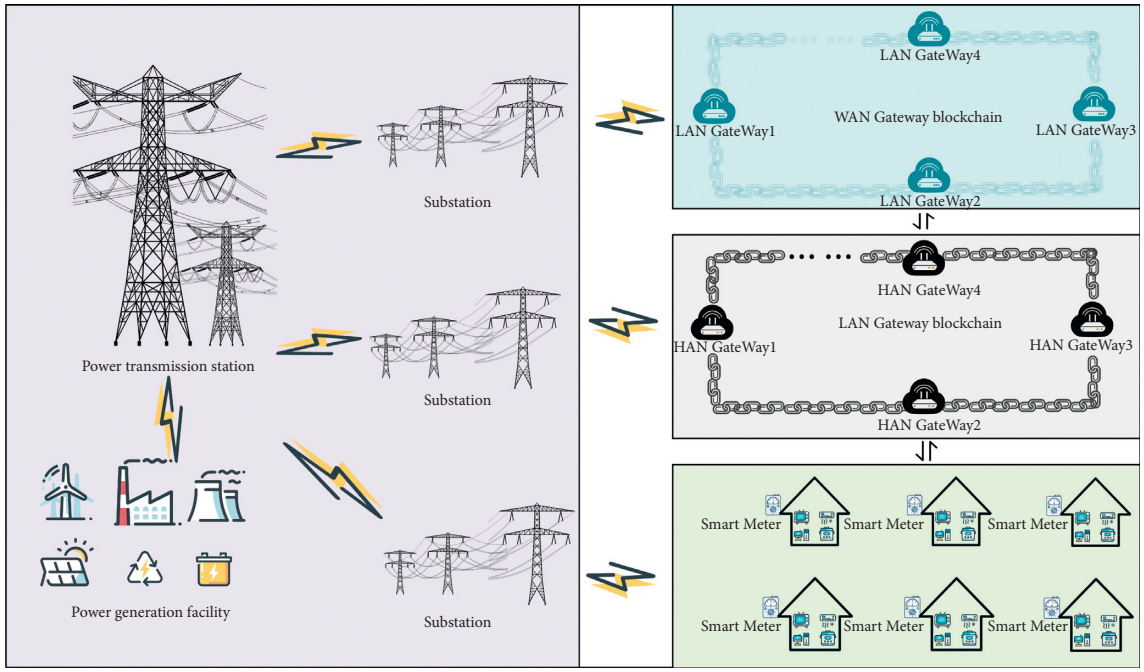


Figure 1: Smart grid gateway consortium blockchain structure.

Table 1: Interpretation of key nouns.

Name	Description
Power gateway	There are different types of gateways in different network structures. The HAN network includes HAN power gateway equipment and smart meter equipment; the LAN network includes the LAN power gateway equipment; and the WAN network includes the WAN power gateway equipment.
HAN network	Devices in a home area network (HAN) share resources through public communication networks (such as Ethernet) or wireless connections (such as WIFI, Bluetooth low energy, ZigBee, and IEEE 802.15.4). The smart meter of each home local area network is used as the entrance and exit of electric energy control, and the electricity consumption in the home network is collected and controlled through the smart meter.
LAN network	The local area network (LAN) is larger than the HAN network communication range from the perspective of network information communication. The LAN network is an alliance blockchain composed of HAN power gateways, which can store data. In the LAN consortium blockchain network, the HAN power gateway node collects and stores information from the smart meters in HAN.
WAN network	From the perspective of network communication, the wide-area network (WAN) has a larger communication range than LAN. In the wide-area network, the power gateway in the LAN is used as a node to form an alliance blockchain. The LAN power gateway in the WAN consortium blockchain network completes data collection and storage in the LAN network.

Active Malicious User Threat. Active malicious users are malicious users with intermittent power theft from the perspective of behavior characteristics. This type of user will perform normal charging behaviors and also conduct power theft behavior. From the perspective of distribution characteristics, this type of user does not have obvious geographic clustering and is usually mixed with normal users.

Passive Malicious User Threat. The distribution characteristics of passive malicious users and active malicious users are the same, but the behavior characteristics are different, which is mainly reflected in the passive malicious users not performing charging behavior.

Threats of Group Malicious Users. The harm of group malicious users to the smart grid is extremely serious. The most distinctive feature is that malicious users gather in the same area, and the behaviors of malicious users are complex and diverse, for example, active malicious users are mixed with passive malicious users.

3.3. Smart Grid NTL Detection Model Based on Power Gateway Consortium Blockchain. In the proposed detection method, the overall structure and concept, intelligent meter registration, online data storage and query, data structure, consensus, and detection process are introduced in detail in the following subsections.

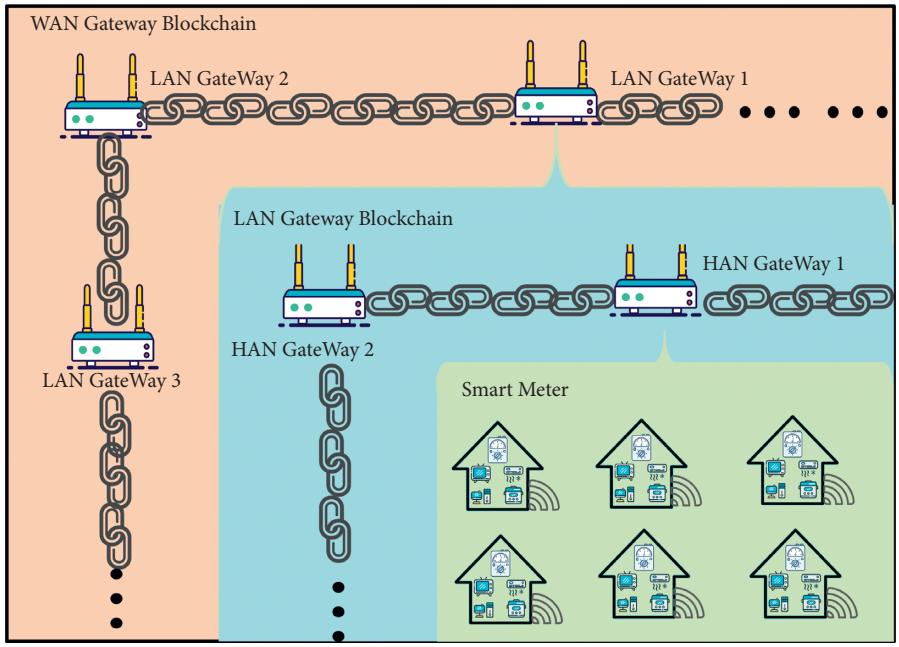


Figure 2: Blockchain structure of power communication network.

3.3.1. Overall structure and Concept of the Detection Method. Logically speaking, each layer of the power communication network contains the MDMS system. Based on the MDMS storage and detection mechanism, a smart grid NTL detection model based on the power gateway consortium blockchain is proposed. The structure of the detection model is shown in Figure 3.

The overall power communication network model includes three parts: the blockchain network, the power gateway, and the smart meter. The communication network includes three network domains, home network, local area network, and wide-area network. The local area network and the wide-area network contain alliance blockchains, which are, respectively, LAN network consortium blockchain and WAN network consortium blockchain. The WAN network consortium blockchain and the LAN network consortium blockchain combine MDMS to manage and control the data of power gateway devices and smart meters, including two parts: device information data and hierarchical power information data.

Equipment Information Collection Task Business. The WAN network consortium blockchain and the LAN network consortium blockchain are combined with the MDMS system to store and manage device information on the chain. The LAN network consortium blockchain forms the MDMS system through the HAN power gateway node to provide device information data query and storage services. The LAN network consortium blockchain collects the state information of the smart meter through the power gateway and stores it in the LAN network consortium blockchain. Similarly, the LAN power gateway is a node of the WAN network consortium blockchain and stores the device status information of the LAN gateway in the WAN network consortium blockchain.

Hierarchical Power Information Collection Task. WAN network consortium blockchain and the LAN network consortium blockchain combine with the MDMS system to store and manage hierarchical power information on the chain. The hierarchical power information includes user payment information, smart meter power information, HAN power gateway power information, and LAN power gateway power information. Among them, the user payment information and power information are uploaded to the LAN network alliance blockchain storage management through the smart meter and the HAN power gateway node power information through the HAN power gateway node. The LAN power gateway power information is stored and managed in the WAN network consortium blockchain through the LAN power gateway node.

Block Structure. The block structure includes the block head and the block body. The block header includes a block identification number, a block size, a timestamp, an address number, and a Merkle root. The block includes equipment information, power information, and source address (smart meter ID, power gateway ID). The specific block structure is shown in Figure 4.

Data Content. The data in the WAN network consortium blockchain includes WAN network layer input power, LAN power gateway ID, timestamp, LAN power gateway equipment power consumption, and device status. The data in the LAN network consortium blockchain includes HAN power gateway output power, HAN power gateway ID, timestamp (including power purchase time, transaction processing time, and power start reading time), smart meter ID, household name, remaining power, purchase power and purchase time, smart meter public, and private key pairs.

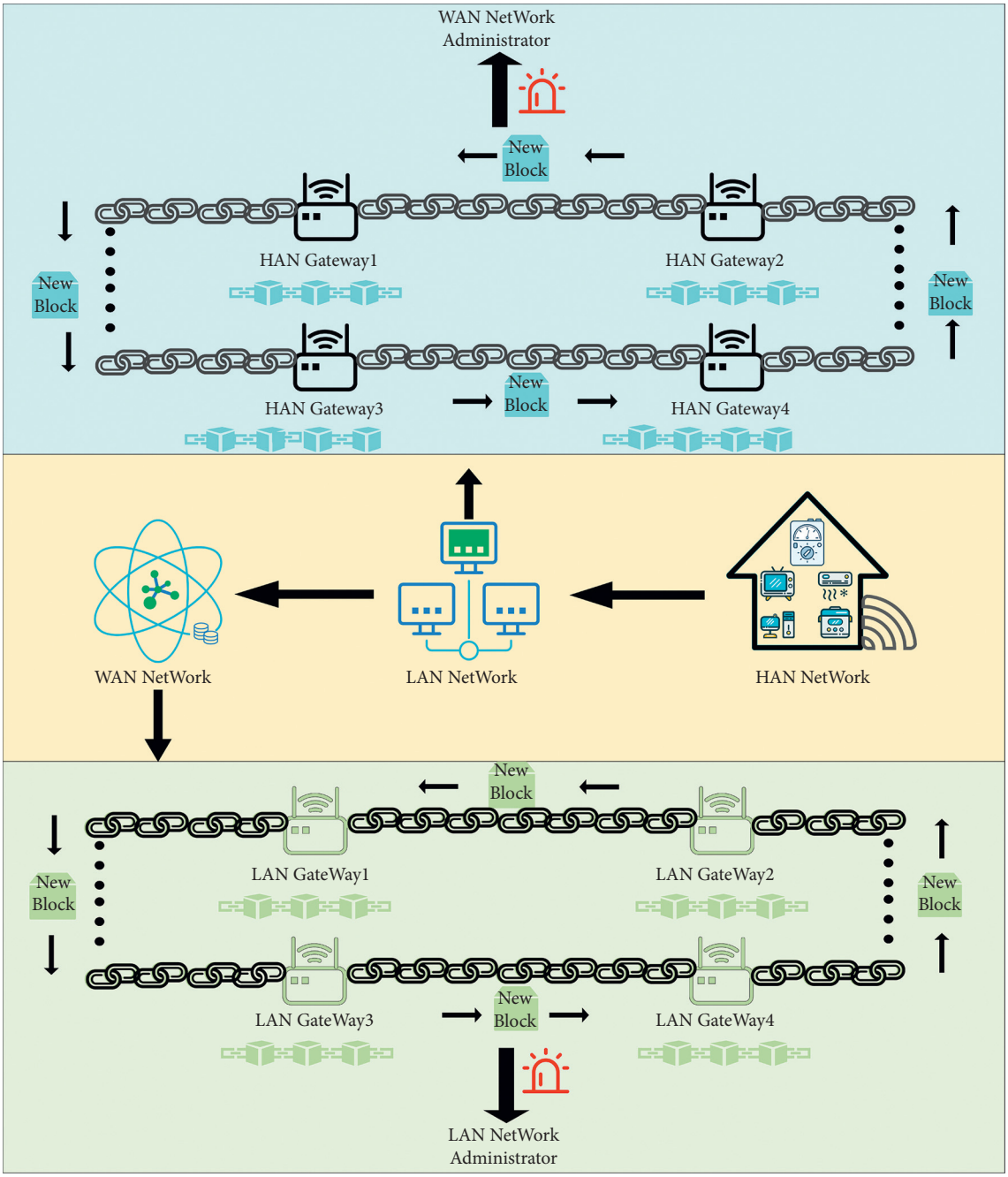


Figure 3: Smart grid NTL detection model based on power gateway consortium blockchain.

Raft Consensus. In Fabric, the orderer service based on Raft replaces the previous Kafka orderer service. Generally, a Raft cluster includes $2N+1$ orderer nodes, allowing N faulty serves in the network. In raft, each node can only be in one of three states [24, 25]:

Follower: in the initial situation, all nodes are followers
Leader: responsible for processing client requests and ensuring that all followers have the same data records
Candidate: candidates will initiate elections to compete for leaders

Under certain conditions, the state of a node can be transformed. In the initial situation, all nodes are followers. Since there is no message from the leader within a period of time, the follower will automatically transform into a candidate and initiate a vote. After receiving votes from most nodes, the node will transform into a leader, accept and respond to requests from clients. For example, when the leader receives an information storage request from the client (HAN Gateways) in the LAN alliance chain, the leader will broadcast this request to the followers. A response will be sent if the follower receives the request successfully. When the leader receives responses from more than half of

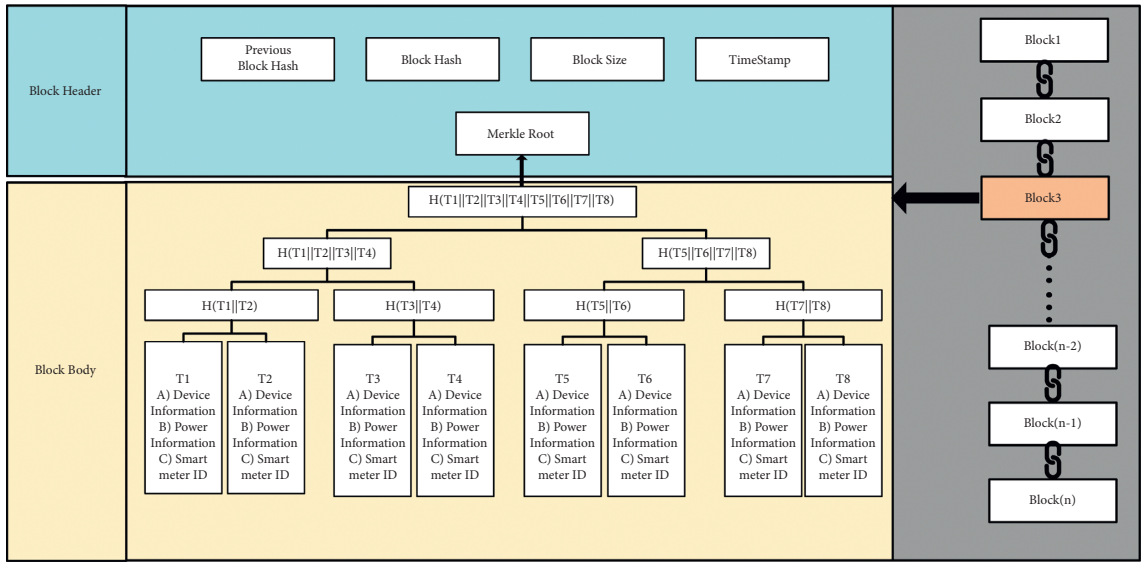


Figure 4: Structure of block data.

the nodes, it will submit the request locally and broadcast all followers to execute the request. The follower accepts and verifies whether the request is legal; after that, the request will be packaged to generate a block, broadcast to all HAN Gateways, and written into the local ledger.

The overall structure takes the form of an alliance blockchain, which is a special blockchain, based on a certain number of preselected authentication nodes. The consensus algorithm of the blockchain is performed by these preselected nodes instead of all the nodes in the whole network, which can greatly reduce the network overhead. In the power grid system, different regions can be regarded as different alliances, so that they can be autonomously managed, and the information can be shared within the scope. The power consumption statistics equipment (smart meter) in the power grid is detected by HAN and LAN power gateway, and the monitoring data are collected and stored. As the real-time detection and audit consume much calculation and storage, a conditional trigger is used to detect the behavior trigger. The introduction of the threat model triggers the detection mechanism when the following methods are employed in the NTL problem detection process of the smart grid.

3.3.2. Initialization and Registration

Assumption 1. The power blockchain gateway is trusted. The audit terminals in the MDMS system deployed by the alliance chain are also trusted.

Assumption 2. The smart meter is semitrusted, and the user is not trusted. The communication channel between the intelligent meter and the power gateway is not completely secure.

Assumption 1 specifies that the gateway of the power blockchain is trustworthy. The power gateway generates certificates and private keys for the intelligent watt-hour meter. This information is stored in the power gateway to ensure that the information is secure and will not be stolen or tampered with. As the audit client in the federation chain MDMS, the audit terminal is also credible, which makes the audit results accurate.

For Assumption 2, the smart meter is a semitrusted entity; it will not actively tamper with and steal information but will be subject to passive attacks. Users are untrusted by default, and such entities are highly aggressive.

The symbols and descriptions used in the whole process are shown in Table 2. The low-power encryption scheme is very important in the Internet of things [26, 27]. The key process of the model is as follows:

System Initialization. The symbol definitions used in the detection method are shown in Table 2.

The system selects an elliptic curve E: $y^2 = x^3 + ax + b \pmod{n}$. The generator is P, and the following three hash function operations are selected. H_1: $\{0,1\}* \longrightarrow G$; H_2: $\{0,1\}* \longrightarrow \{0,1\}*$; H_3: $G \longrightarrow \{0,1\}*$. The private key of the power gateway is α and its public key is $p_a = \alpha * P$.

Smart Meter Information Registration Process. HAN network users request access to smart meters from HAN power gateway nodes through the communication network. Access is allowed if authentication is passed, and access is denied if the authentication fails. The HAN power gateway combines the information of smart meter and house number to generate the unique identification number in the current HAN network. All the smart meter identification number information in the HAN network is stored in the HAN gateway. When the NTL occurs, the HAN gateway can be responsible for the smart meter with NTL problems according to the identification number information. Since the smart meter as a client needs to sign when it needs to submit to blockchain request to the HAN gateway, the HAN gateway needs to generate a public and private key pair for the smart meter and send

Table 2: Notations used in this paper.

Symbol	Description
P	Generator
E	Elliptic curve
(α, p_a)	Private key and public key of power gateway
SM_{id}	ID of smart meter
Q_{SM}	Certificate of smart meter
$PSK_{SM_{id}}$	Private key of smart meter
(r_a, y_a)	One-time password power gateway private key, public key
(r_s, Y_s)	One-time password smart meter private key, public key
M	Data uploaded by smart meter
T_i	Time stamp
H_1, H_2, H_3	Hash operation
(r_b, y_b)	Audit client's private key, public key

the private key to the smart meter for signature. The specific process is shown in Figure 5.

The smart meter has a unique ID for SMid, for the power gateway to issue a certificate for it, as follows:

Step 1: smart meter generates random number k_i as its private key, $k_i \in [1, n-1]$

Step 2: smart meter sends (k_i, SM_{id}) to power gateway for the later generation of certificates

Step 3: the power gateway calculates its certificate $Q_{sm} = \alpha * k_i * P$, to further update its private key to $PSK_{SM_{id}} = \alpha * k_i + H_3(Q_{sm}) * \alpha$

Step 4: the power gateway will return $(Q_{sm}, PSK_{SM_{id}}, t_i)$ to the smart meter is $(certificate, privatekey, timestamp)$

3.3.3. Data Storage and Query Process. The nodes of the LAN network consortium blockchain and WAN network consortium blockchain are in the HAN power gateway and the LAN power gateway, respectively, and they are responsible for the client to submit data information to the blockchain. The process is shown in Figure 6.

The smart meter signs and uploads the data, and the process is mainly divided into four steps: one-time password generation, message signature, identity verification, and message verification.

In order to ensure the security of the data, the one-time password is used every time the smart meter uploads the data, and the generation process is as follows:

Step 1: the power gateway generates a random number r_a and sends it to the smart meter

Step 2: the smart meter randomly selects r_s as its private key and calculates its public key as $y_s = r_a * r_s * Q_{SM}$

Step 3: the power gateway uses its private key α to generate a public key of $y_a = H_2(r_a) \oplus H_3(\alpha * y_s)$

As the smart meter is a semitrusted entity, when generating the public key, the public key value is determined by both the power gateway and the smart meter.

The smart meter signature process for uploading data:

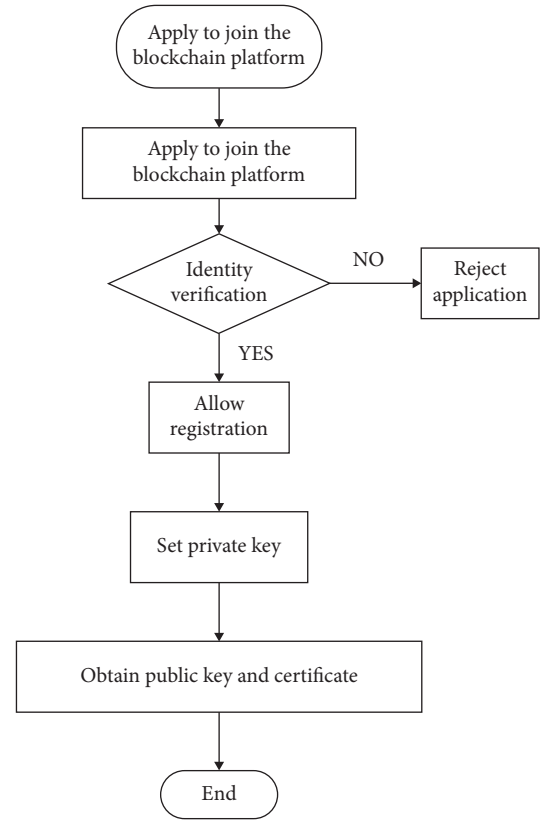


Figure 5: Smart meter information registration process.

Step 1: firstly, the private key $PSK_{SM_{id}}$ issued by the power gateway node is used to sign the uploaded data: $sign(M) = H_2(M, SM_{id}, y_s, t_i) * r_s + PSK_{SM_{id}}$.

Step 2: the smart meter will upload the data $msg = (SM_{id}, sign(M), y_s, M, t_i, Q_{sm}, \backslash\backslash H_2(r_a))$ to the power gateway. It is easy to verify the identity of the smart meter. If the transmission channel is eavesdropped or tampered with, the power gateway can determine whether the message has been tampered with according to the signature $\text{sign}(M)$.

The authentication process of the power gateway to the smart meter is as follows:

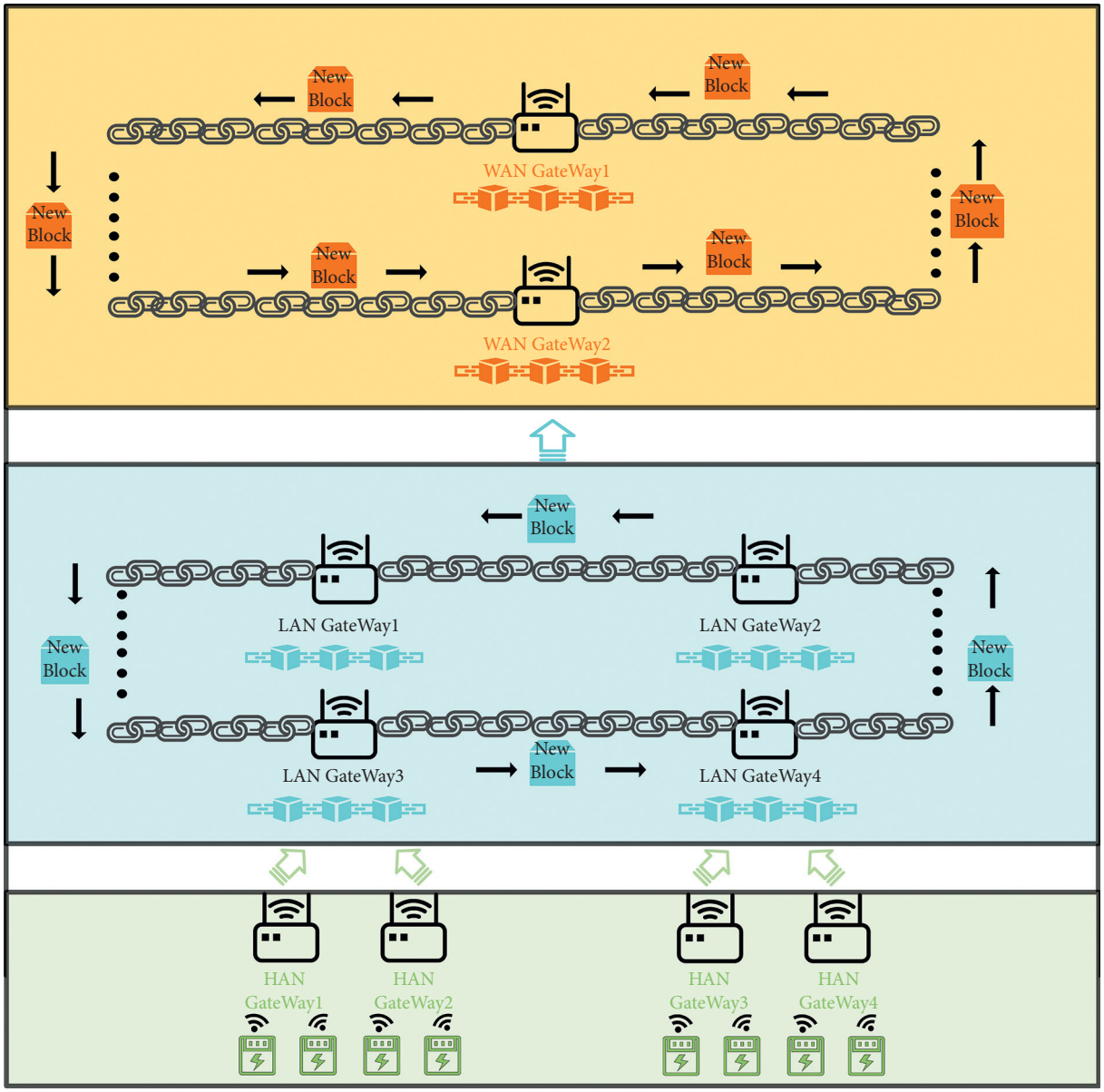


Figure 6: Data block generation process.

In the case that the Q_{sm} verification is passed, to prevent the certificate from being eavesdropped on by an adversary, further judge $H_2(r_a) = y_a \oplus H_3(\alpha * y_s)$. It ensures that the message cannot be tampered with.

The power gateway verifies the message sent by the intelligent meter as follows:

First, calculate the $h_1 = H_2(M, SM_{id}, y_s, t_i) * r_s$; $h_2 = H_3(Q_{sm})$. Determine whether the equation $sign(M) * P = h_1 * y_s + Q_{sm} + h_2 * p_a$ is true, and if so, receive the message.

The verification principle is as follows:

$$\begin{aligned} sign(M) * P &= \left(H_2(M, SM_{id}, y_s, t_i) * r_s + PSK_{SM_{id}}\right) * P \\ &= h_1 * y_s + \alpha * k_i * P + h_2 * P \\ &= h_1 * y_s + Q_{sm} + h_2 * p_a. \end{aligned} \tag{1}$$

The above is a single message authentication process. If batch message processing is carried out and the number of messages is assumed to be n, the verification process is as follows:

$$\begin{aligned} SP &= \sum_{i=1}^{n} sign_i(M_i) * P \\ &= \sum_{i=1}^{n} \left(h_1 + PSK_{SM_{id}}\right) * P \\ &= \sum_{i=1}^{n} h_1 * P + \sum_{i=1}^{n} [\alpha * k_i + H_3(Q_{SM} * \alpha)] * P \\ &= \sum_{i=1}^{n} h_1 * P + \sum_{i=1}^{n} Q_{sm} + \sum_{i=1}^{n} h_2 * p_a. \end{aligned} \tag{2}$$

The HAN power gateway node stores the collected smart meter data (HAN network layer data) on the LAN network consortium blockchain, and the LAN power gateway node stores the aggregated LAN network layer data on the WAN network consortium blockchain. The data is stored in an encrypted manner, and the way the data is stored on the blockchain and obtained is shown in Figure 7.

After obtaining the data, the power gateway node encrypts the data through the encryption algorithm, stores it on the chain, and decrypts the query in the process of detection and audit.

The audit client audits the data uploaded by the smart meter, and the process is as follows: as the audit client and the power gateway are trusted entities, both parties can use the original elliptic curve encryption algorithm when transmitting data:

Step 1: the audit client chooses the private key as r_b; then, its public key is $y_b = r_b * P$

Step 2: the power gateway hashes the data m to be audited: $M = H_1(m)$, randomly generates r, and calculates the point $R = r * P$

Step 3: the power gateway calculates $C = M + r * y_b$ and returns the (C, R) to the auditor

Step 4: fter the audit client gets the ciphertext C, calculate the plaintext $M = C - r * y_b = C - r * r_b * P = C - R * r_b$ and audit it

3.3.4. NTL Detection Method. The HAN user initiates the power purchase on the platform, and the user sends the verification information $HANPurchaseInfo = \{UserID, SMID, Purchaseamount, TimeStamp\}$ to the platform for verification. After the verification is passed, the audit contract of the detection mechanism is triggered, as shown in Figure 8.

The steps for the audit contract are as follows and the process is shown in Algorithm 1:

Step 1: HAN tests the connectivity of the smart meter (obtaining the meter status data), performs Step 2 if the test is successful, and issue an alarm to the auditor if the test fails.

Step 2: the HAN gateway node sends a request for information collection to the smart meter of the power buyer.

Step 3: if the smart meter receives the request information, it responds to the request of the HAN gateway node and transmits the $HANsm = \{SMID, UserID, Remaining\ Electricity\}$ information to the HAN power gateway node.

Step 4: the HAN power gateway node obtains the $HAN\ gw = \{SMID, UserID, CurrentTime,\ the user's\ last\ power\ purchase\ time\ (Tlast), after\ the\ electricity\ purchase\ (Elast)\}$ information, which is compared and fused with the HANsm information. We calculate the difference between the (*Elast*) and the remaining power of the watt-hour meter after the last power purchase and compare it with the output electricity of the HAN gateway (the electricity information between the last purchase time and the current purchase time). We judge whether the charging users and other users under the current HAN power gateway node have abnormal power consumption.

Step 5: After the verification is passed, $HANgw1 = \{SMID, UserID, CurrentTime, PurchaseTime,\ after\ purchased\ electricity\ (ATE)\}$ is packaged and uploaded. At the same time, the platform will send the purchased electricity to the smart meter of the family.

(1) NTL Detection Method for HAN Network. Aiming for the problem of passive malicious user detection, a HAN network NTL detection method is proposed based on The NTL detection method. Every once in a while, the HAN gateway will query the data on the chain, request the data of the smart meter, then calculate the theoretical power consumption of each smart meter under the current HAN network, and after that compare it with the actual output power EOutput of each user's HAN gateway. If the actual output power is greater than the theoretical power consumption, the user is considered to be a passive malicious user. The process is shown in Algorithm 2.

(2) NTL Detection Method for LAN Network. The WAN network layer regularly audits LAN users following the audit rules. The WAN network initiates a regular audit of the power output of the LAN power gateway to audit whether the WAN input and the LAN output are balanced. According to the audit results, it is to judge whether the LAN group users have NTL problems. The process is shown in Algorithm 3.

After the WAN network carries out the connectivity test to the gateway node (obtains the equipment state data), every interval T triggers the audit contract; in other words, it carries out the query about the WAN gateway node information stored in the LAN consortium blockchain. The input power data of the WAN network is obtained and compared with the LAN node data to determine whether there is a problem with LAN group user NTL. If there is a problem, the auditor is alerted.

4. Experimental Simulation

We have carried out experiments on the proposed smart grid NTL detection scheme based on the power network association chain and simulated the data winding and the detection process of the LAN alliance chain, including HAN users (smart meter), the alliance chain composed of the HAN gateway, and the detection client. The structure of the experiment is shown in Figure 9.

4.1. Experimental Environment. The Docker is used to simulate peers on the blockchain to verify our scheme. The OS used is Ubuntu 18.04, and the version of the Hyperledger Fabric is 2.3.0. More details for the experimental environment are listed in Table 3.

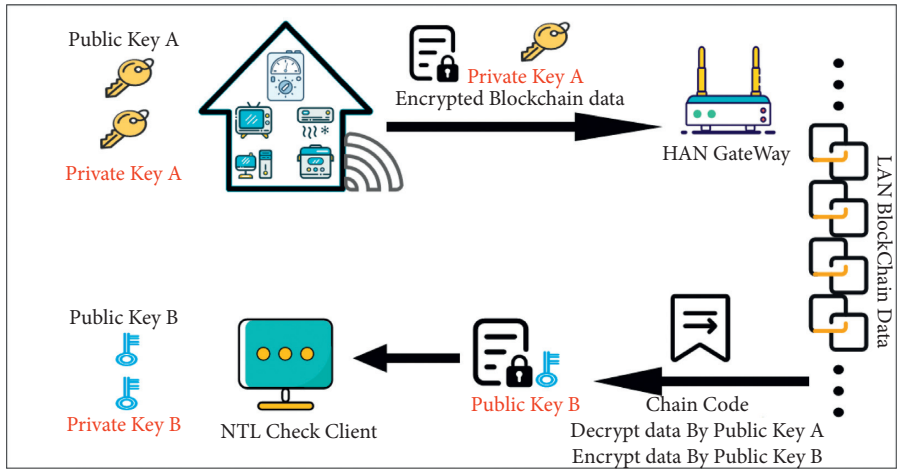


Figure 7: Data storage and acquisition process.

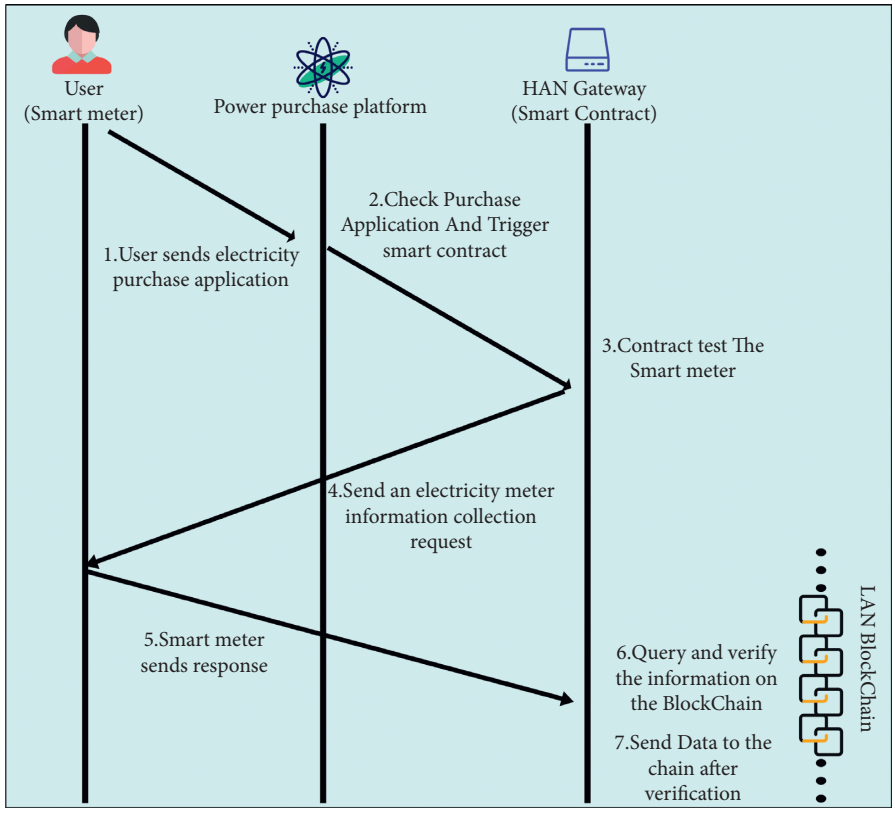


Figure 8: NTL detection method.

For the LAN alliance chain, the blockchain network consists of two Orgs, each of which has fifteen peers (HAN Gateways). The peer0 of each Org serves as the anchor node of its own Org and is responsible for the communication between organizations. There is one channel in the network; all peers will install the chain code and join the channel.

4.2. Experimental Result. The main steps of the experiment include the creation and maintenance of the channel, the development, and the use of the chain code. The administrator is responsible for adding HAN Gateways and LAN Gateways to their corresponding channels, developing and deploying chain code, and fulfilling other requirements. The blockchain network function test and the Smart Grid data interaction function test are shown in Tables 4 and 5, which mainly include storing and querying the gateway power date.

We tested a network with two Orgs, and four HAN Gateways per Org. The test results are shown in Figures 10 and 11. The results show that the processing capacity of the LAN blockchain network reaches the peak when four HAN Gateways initiate transactions at the same time.

Input: HANpurchaseInfo = {UserID, SMID, Purchase amount, TimeStamp}
Output: Audit Result
(1) **function** NTL(HANpurchaseInfo)
(2) get the state of the Smart Meter
(3) **if** State = offline **then**
(4) **return** Send warning to Auditors
(5) **else**
(6) Send request to the corresponding Smart Meter
(7) get HANsm = {SMID, UserID, SOC} sent from Smart Meter
(8) get HANgw = {SMID, UserID, Current Time, Tlast,Elast} from HAN
(9) **if** (Elast-SOC)-E_Output > threshold **then**
(10) **return** Send warning to Auditors
(11) **else**
(12) Purchasing Time = TimeStamp
(13) SOC = SOC + Purchase amount
(14) send HANgw1 = {SMID, UserID, Current Time, Purchasing Time, ATE} to BlockChain
(15) send update to user's smart meter
(16) **return** Normal
(17) **end if**
(18) **end if**
(19) **end function**

Algorithm 1: Contract for audit.

Input: TIME INTERVAL
Output: Analysis Result
(1) **function** NTL FOR HAN
(2) **if** Current Time-Last Time = TIME INTERVAL **then**
(3) **for** $i = 0 \longrightarrow n$ **do**
(4) get HANsm = {SMID, UserID, SOC} sent from $User_i$'s Smart Meter
(5) get HANgw = {SMID, UserID, Current Time, Tlast,Elast} from HAN
(6) E_Theoretical_Consumption = Elast-SOC
(7) get E_Output from HAN
(8) **if** E_Output > E_Theoretical_Consumption **then**
(9) Send warning to Auditors
(10) **end if**
(11) **end for**
(12) **return** over
(13) **end if**
(14) **return** waiting
(15) **end function**

Algorithm 2: NTL for HAN.

Instead of controlling the peers, join the channel one by one; we join all the peers into the channel at the same time and then control the number of peers that initiate transactions at the same time.

In the LAN alliance chain, different numbers of HAN Gateways initiate transactions at the same time for different total transactions. The results include the time consumption and throughput referring to the number of transactions that can be processed per second. Figure 12 shows the relationship between the time required to complete the transaction and the number of HAN Gateways needed to initiate the transaction. Figure 13 shows the relationship between the throughput and the number of HAN Gateways needed to initiate the transaction at the same time. It can be seen that, with the increase in the number of HAN Gateways participating in the transaction, the processing capacity of the LAN alliance chain network continues to increase and eventually stabilizes. When three HAN Gateways initiate transactions at the same time, the maximum processing capacity of the LAN network is achieved. It can be seen that, in application, we only need a small number of nodes to make full use of the blockchain network; thus, we can save our costs.

It is worth noting that the throughput of the blockchain is affected by many factors, including but not limited to system architecture, hardware, and consensus algorithm.

Input: TIME INTERVAL
Output: Analysis Result
(1) **function** NTL FOR LAN
(2) get the state of LAN
(3) **if** State = offline **then**
(4) **return** Send warning to Auditors
(5) **else**
(6) **if** Current Time-Last Time = TIME INTERVAL **then**
(7) **for** $i = 0 \longrightarrow n$ **do**
(8) get LAN1 = {LANID, SOC} sent from LAN_i
(9) get LAN2 = {LANID, Elast} from WAN
(10) E_Theoretical_Consumption = Elast-ATE
(11) get E_Output from WAN
(12) **if** E_Output > E_Theoretical_Consumption **then**
(13) Send warning to Auditors
(14) **end if**
(15) **end for**
(16) **return** over
(17) **end if**
(18) **return** waiting
(19) **end if**
(20) **end function**

Algorithm 3: NTL for LAN.

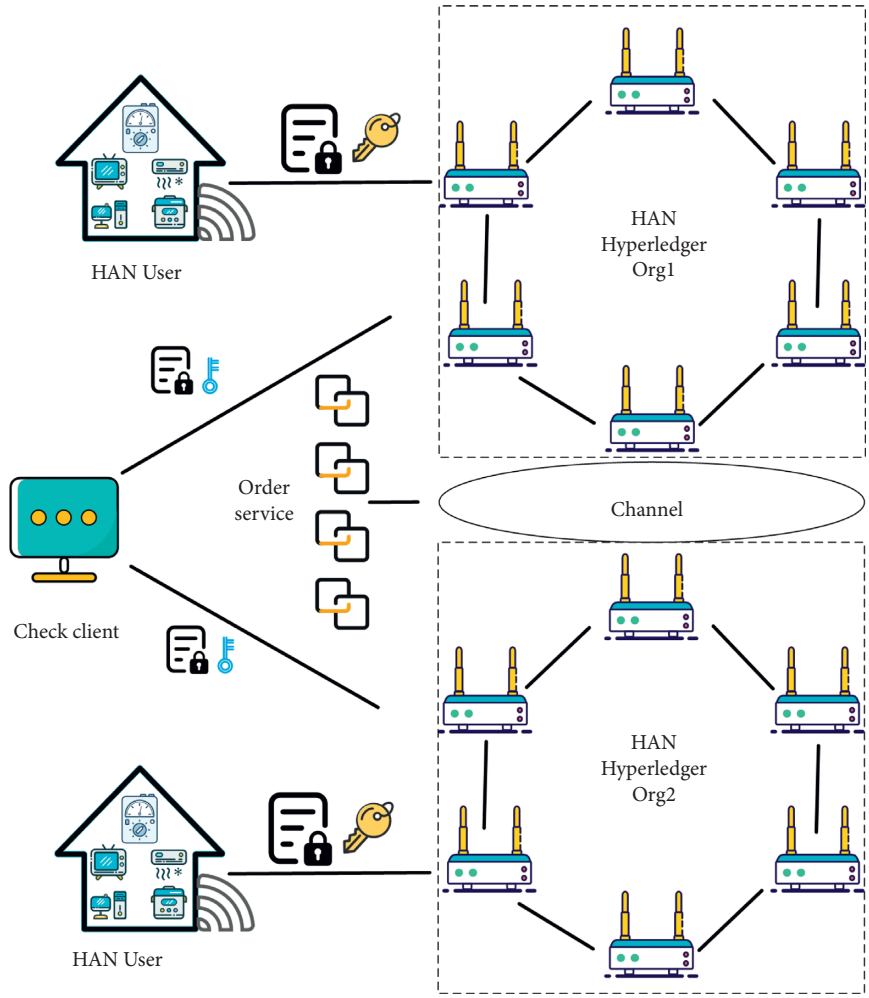


Figure 9: Experimental environment.

Table 3: Experimental environment.

Tools	Version	Function
Ubuntu	18.04	The operating system
Hyperledger Fabric	2.3.0	An open-source alliance chain framework for generating blockchain network
Docker	19.03.6	Used to simulate peers in blockchain
Docker-compose	1.17.1	Manage container
Go	1.15.7	Develop chain code (smart contract)

Table 4: Blockchain network function test.

Function	Explanation	Result
Create channel	Create channels for LAN or WAN alliance chain	Success
Join channel	Add HAN Gateways and LAN Gateways to their corresponding channels	Success
Deploy chain code	Install chain code on the channel	Success
Invoke chain code	Execute the function defined on the chain code	Success

Table 5: Blockchain network function test.

Function	Explanation	Result
Power data storage	Upload power data to blockchain	Success
Power data query	Query the power data from blockchain	Success

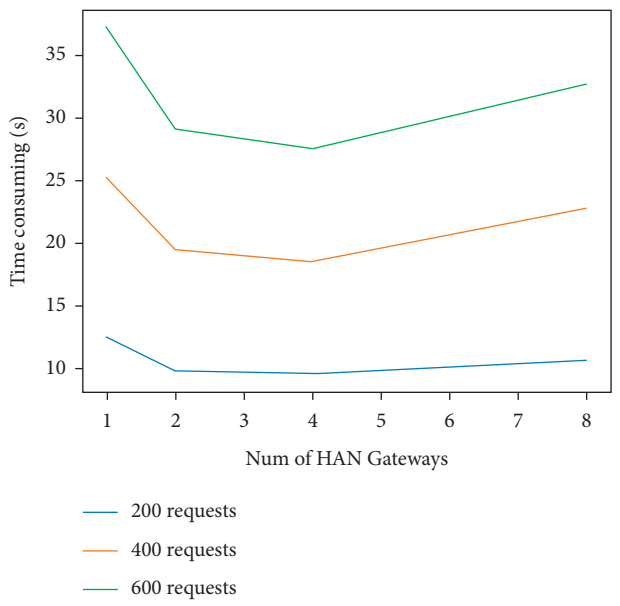


Figure 10: Time consumption for different numbers of transactions and HAN Gateways (four nodes).

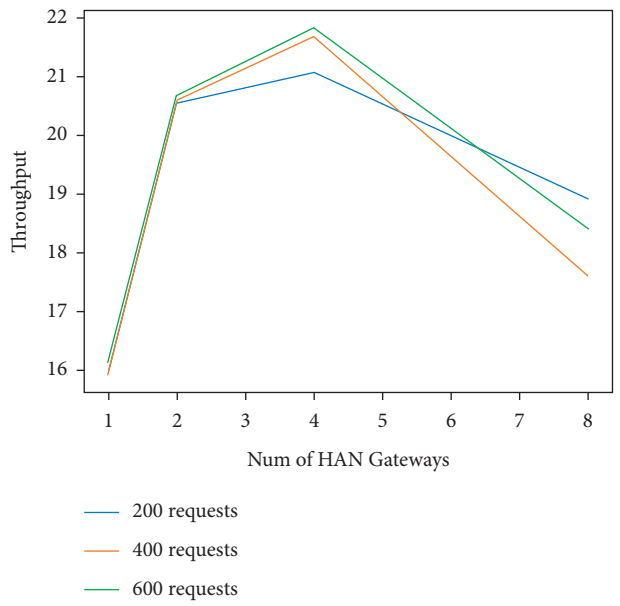


Figure 11: Throughput for different numbers of transactions and HAN Gateways (four nodes).

The number of peers needs to be appropriately set according to the application of the scenario.

To further verify the feasibility of the scheme, the dataset [28] from Smart Energy Informatics Lab was selected. The dataset consists of electricity consumption data (December 2016 to January 2018) from a high-rise residential building inside the IIT Bombay campus. Each apartment is instrumented with a smart meter. For privacy reasons, the name of apartments are kept anonymous and are replaced by numbers. The date is downsampled at 1-hour granularity. It includes apartment ID, timestamp, voltage, and energy consumption.

The results are as shown in Figures 14 and 15, similar to previous results, the processing capacity of the LAN alliance chain network continues to increase and eventually stabilizes with the increase in the number of HAN Gateways participating in the transaction.

To our best, only one similar paper is found. Khalid et al. [29] tried to combine the IoT device with blockchain to eliminate nontechnical loss. The IoT devices are deployed at

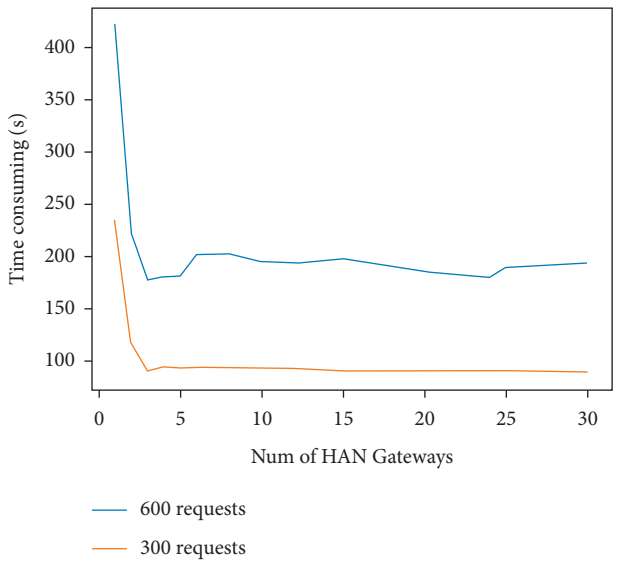


Figure 12: Time consumed for different numbers of transactions and HAN Gateways (fifteen nodes).

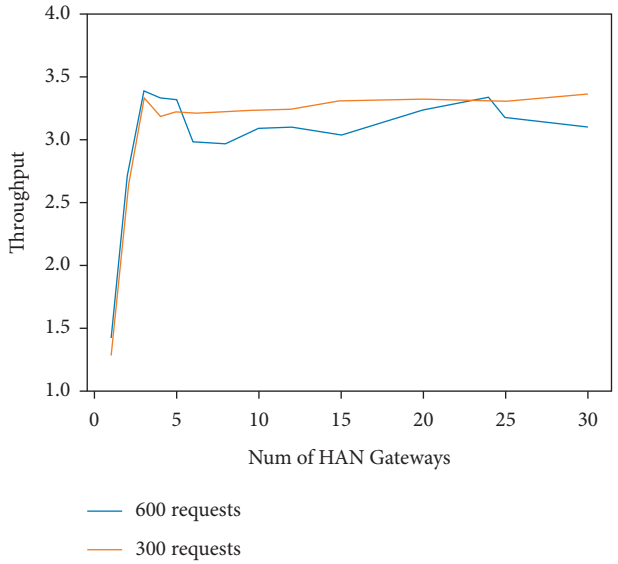


Figure 13: Throughput for different numbers of transactions and HAN Gateways (fifteen nodes).

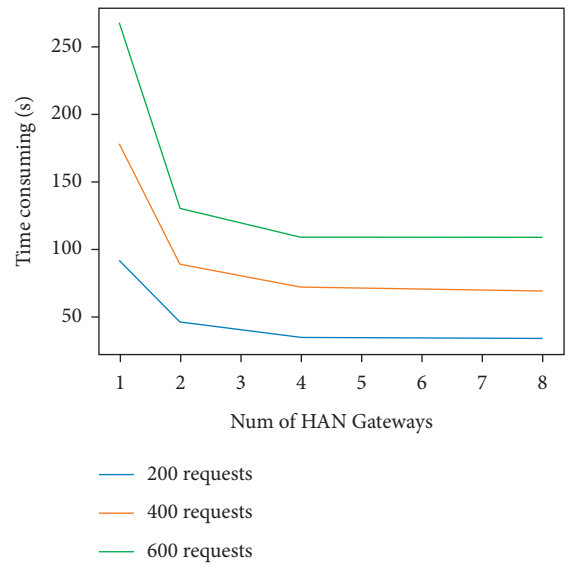


Figure 14: Time consumed for different numbers of transactions and HAN Gateways (four nodes).

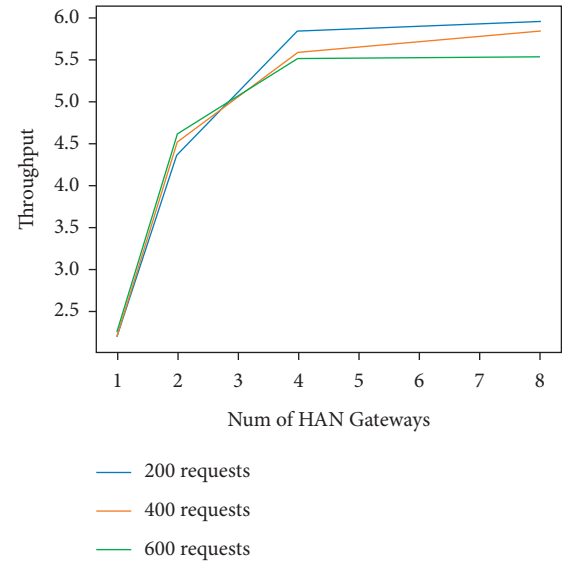


Figure 15: Throughput for different numbers of transactions and HAN Gateways (four nodes).

the key point of the power system to detect electricity production and consumption. The nontechnical loss is detected by calculating the difference between production and consumption. Ethereum is used to verify this scheme finally.

For consumers of different sizes, Sana designed different solutions. Private chains, alliance chains, and public chains are used to target large-scale, medium-scale, and small-scale consumers, respectively. This indeed improves the throughput of the blockchain, but there is little improvement in NTL detection. The scheme proposed in our work is based on Fabric which also is known as alliance chain. Although the private chain has a high throughput, the peers in the private chain are required to be mutually trustworthy, which is impossible in the actual situation. Compared with the private chain, the alliance chain is more in line with the actual situation. This is because the nodes in the alliance chain only need to be semitrusted between others. However, this paper only offers the results of the successful execution of smart contracts and blockchain; it does not offer the performance results.

What is more, large-scale IoT devices are needed to be installed to find specific users who stole electricity which will result in high costs. However, the hierarchical structure proposed in our paper allow us to locate users who stole electricity more conveniently and flexibly based on existing power supply equipment. By analyzing the data from different HANs and LANs in their respective

blockchain networks, we can effectively solve the problems of single-user power theft and group user power theft.

4.3. Qualitative Analysis of the Results. In this section, we will discuss the differences between the design scheme of this paper and other existing schemes in each index dimension, which mainly includes the following eight index dimensions. The first dimension is the detection effectiveness of NTL, and the second dimension is to judge whether it has the characteristics of decentralization, which can avoid the single point of failure and other problems. The third dimension is the data tamper-proof; because this paper uses the blockchain structure, it has the antitamper ability of data. The fourth dimension is the intelligent detection capability, which mainly examines whether the detection scheme can be carried out without the need for manual table lookup, to reduce the labor cost. All the detection processes in this paper can be automatically carried out by the intelligent contract; thus, there is no need for any manual table lookup. The fifth dimension is the ability of information sharing, which mainly refers to the ability of data sharing between nodes. In this paper, due to the use of blockchain mechanism, different nodes achieve data consistency through the consensus mechanism. The sixth dimension is the confidentiality of the data. All the upper-chain information in this paper is ciphertext so that the data can be effectively protected. The seventh dimension is the traceability and auditability of the data; because all the power equipment information and the power purchase information are stored on the chain, the power purchase behavior can be traced back and audited. The eighth dimension is independent of audit data; because the detection process in this paper is to trigger intelligent contracts for detection, there is no need to train datasets for learning; thus, it is not dependent on large audit data.

Based on the eight indicators previously pointed out earlier, our work is compared with other existing works, and the results are shown in Table 6.

5. Security Analysis

In this section, the security of the proposed method is analyzed from the aspects of smart meter information initialization, data authentication, block verification, and threat scenario.

5.1. Smart Meter Information Initialization. The smart meter, as a client, needs to be signed when submitting a chain request to the HAN gateway; therefore, the HAN gateway needs to generate a public-private key pair for the smart meter and send the private key to the smart meter as a signature. The HAN gateway uses a hash algorithm and a random number generator to generate public and private key pairs. Although the random number generator is built randomly by man, it can be exploited by attackers. The hash algorithm provides a more secure method. The SM public key information is a blockchain created based on the Merkle tree and timestamp using a hash function and is stored in the HAN gateway to keep it safe during the initialization phase of the smart meter.

5.2. Data Authentication Security. The data is stored on the permissioned blockchain through encryption. After the HAN gateway obtains the smart meter data, it encrypts the user's meter data through an encryption algorithm and stores it on the blockchain. When SM communicates with the HAN gateway, they create a secure session and update the private key pairs at intervals of time t. When the HAN gateway initiates a request and receives a message encrypted with the private key by the SM, as the leader, it uses the SM public key to verify the signature of the encrypted data. The authentication security and the integrity of data transmission are ensured by means of private key pairs verification.

5.3. Block Verification Security. The security of block verification in the scheme is guaranteed by the Raft algorithm. The MDMS in the designed smart grid is a distributed system, in which the failure of a single gateway is an independent event. Assume that there are n HAN Gateways in a LAN alliance chain, where the number of faulty nodes is f. As the election of the leader is based on voting in raft, we need to ensure that the number of normal nodes is greater than the faulty nodes to guarantee the voting process. Therefore, we need $n - f > f$, which leads to $n > 2f$. Then, we need to ensure that there are at least $2f + 1$ nodes in the system to ensure the security of the distributed system.

5.4. Security Analysis of Threat Scenario

(1) Active Malicious User Threat Analysis. As the active malicious user will carry out the charging behavior, it will trigger the NTL detection method mentioned above. After the request of the active malicious user passes the platform verification and the active malicious user's electricity meter passes the subsequent connectivity test, the HAN gateway will collect the information from the user's smart meter HAMsm: smart meter ID, UserID, remaining power (ERemain), and then the HAN gateway will query the information on the chain to obtain the user's last charging information HANgw: $\{smartmeterID,\ UserID,\ currenttime,\ user's\ last\ purchase\ time\ (Tlast),\ after\ the\ meter\ (ELast)\}$. Combined with HANsm, the difference between the quantity of (ELast) and the remaining power of the meter after the last purchase is calculated to get the theoretical power consumption E_Theoretical_Consumption = (ELast-ERemain) and compared with the actual output of the HAN gateway (electricity information between the last purchase time and the current purchase time).

Since the active malicious user has the behavior of stealing electricity, the E_Output is greater than E_Theoretical_Consumption, and the difference between the two

Table 6: Comparison between the proposed method and other related methods.

Metric	[9]	[6]	[30]	[11]	[31]	[20]	Our method
Detection of NTL	Yes	Yes	Yes	Yes	No	Yes	Yes
Decentralization	No	No	No	No	Yes	Yes	Yes
Data tamperproof	No	No	No	No	Yes	Yes	Yes
Intelligent detection	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Information sharing	No	No	No	No	Yes	Yes	Yes
Data confidentiality	No	No	No	No	No	No	Yes
Data traceability and audit	No	No	No	No	Yes	Yes	Yes
Nonaudit data reliance	Yes	Yes	No	No	Yes	Yes	Yes

represents the malicious degree of the malicious user. The more the number of power theft, the higher the malicious degree. At this time, an alarm of electricity theft will be made to the LAN power network administrator and be dealt with according to the degree of malice.

(2) Passive Malicious User Threat Analysis. Passive malicious users do not charge, so regular NTL detection methods cannot be triggered. However, this paper introduces the NTL detection method of the HAN network, and every once in a while, the HAN gateway will query the data on the chain, request the data of the smart meter, then calculate the theoretical power consumption E_Theoretical_Consumption, of each smart meter under the current HAN network, and compare it with the actual output power E_Output of each user's HAN gateway. If the actual output power is greater than the theoretical power consumption, the user is considered to be a passive malicious user. And because the HAN network node in this paper separately maintains the blockchain data structure for each user, compared with maintaining a blockchain data structure, the query data amount of this scheme is smaller; thus, it is more efficient.

(3) Group Malicious User Threat Analysis. At present, the LAN network NTL detection method and the HAN network NTL detection method proposed in this paper can effectively solve the threat of malicious users of this group. The difference between the LAN network NTL detection method and the HAN network NTL detection method mainly lies in the different content of the blockchain data. The block data of the LAN network chain records the power purchase information in the unit of the user, while the block data of the WAN network chain records the power purchase information in the unit of the region, and each LAN power gateway represents an area. Therefore, the group of malicious users can be classified as the different malicious area.

6. Conclusion

In this paper, we propose a smart grid NTL problem detection scheme based on the power gateway blockchain to solve the NTL problem in the smart grid system. Our scheme divides the communication network domains such as HAN, LAN, and WAN in the smart grid. A hierarchical power grid gateway blockchain is proposed and designed, and a decentralized management MDMS system is constructed. Without the support of a large amount of data, the intelligent contract combined with encryption technology is used to store and query the power data, and the detection of NTL problems is realized. First of all, the overall structure of the consortium blockchain of the smart grid gateway is described. Secondly, the threat scenarios of NTL problems in the smart grid are analyzed. Finally, a smart grid NTL detection model based on the power grid association consortium blockchain is proposed. The model uses the edge network blockchain to store the state information of smart meter, power gateway, and related power data. In the model, the data situation in the smart grid, and the data winding and query process in the smart grid are described in detail. The trigger mechanism and the detailed detection flow of the NTL detection method are introduced, and a smart contract is written to ensure the safe and reliable operation of the detection scheme. It has a certain ability to resist attacks such as replay, monitoring, and tampering. It is worth noting that the throughput and the consumed time of the blockchain are affected by many factors. The number of peers needs to be reasonably set according to the application in the scenario. After testing the performance of the scheme, it is proved that it is theoretically feasible. In the future, we will expand our work to optimize the efficiency of the consensus algorithm and to refine the trigger conditions of the detection mechanism to improve the practical feasibility of the scheme.

Data Availability

The dataset used in this manuscript belongs to synthetic data. The synthetic data used to support the findings of this study are available from the corresponding author upon request. There are no restrictions on access to the synthetic datasets.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the Fundamental Research Funds for the Central Universities (2019YJS033) in China.

References

[1] K. Yu, K. Shibata, T. Tokutake et al., "A lightweight ledger-based points transfer system for application-oriented LPWAN," in *Proceedings of the 2020 IEEE 6th International Conference on Computer and Communications (ICCC)*, pp. 1972–1978, IEEE, Chengdu, China, December 2020.

[2] L. Chen, S. Suo, X. Kuang, Y. Cao, and W. Tao, "Secure ubiquitous wireless communication solution for power distribution internet of things in smart grid," in *Proceedings of the IEEE International Conference on Consumer Electronics and Computer Engineering (ICCECE)*, pp. 780–784, IEEE, Guangzhou, China, December 2021.

[3] Y. Zhang, J. Zou, and R. Guo, "Efficient privacy-preserving authentication for V2G networks," *Peer-to-Peer Networking and Applications*, vol. 14, no. 3, pp. 1366–1378, 2021.

[4] J. L. Viegas, P. R. Esteves, R. Melício, V. M. F. Mendes, and S. M. Vieira, "Solutions for detection of non-technical losses in the electricity grid: A review," *Renewable and Sustainable Energy Reviews*, vol. 80, pp. 1256–1268, 2017.

[5] Y. Li, Q. Wang, D. Zhang, X. Sun, and X. Xu, "Research and application of electricity anti-stealing system based on neural network," in *Proceedings of the 2016 3rd International Conference on Information Science and Control Engineering (ICISCE)*, pp. 1039–1043, 2016.

[6] V. B. Krishna, K. Lee, G. A. Weaver, R. K. Iyer, and W. H. Sanders, "F-DETA: a framework for detecting electricity theft attacks in smart grids," in *Proceedings of the 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2016*, pp. 407–418, IEEE Computer Society, Toulouse, France, June 2016.

[7] S.E. McLaughlin, D. Podkuiko, and P.D. McDaniel, "Energy Theft in the Advanced Metering Infrastructure," in *Critical Information Infrastructures Security, 4th International Workshop, CRITIS 2009, Bonn, Germany, September 30–October 2, 2009. Revised Papers; Rome (Lecture Notes in Computer Science)*, R. E. Bloomfield, Ed., vol. 6027, pp. 176–187, Springer, 2009.

[8] J. Dou, X. Liu, J. Lu, D. Wu, and X. Wang, "Research on electricity anti-stealing method based on power consumption information acquisition and big data," *Electrical Measurement and Instrumentation*, pp. 60–67, 2018.

[9] J. B. Leite and J. R. S. Mantovani, "Detecting and Locating Non-Technical Losses in Modern Distribution Networks," *IEEE Transactions on Smart Grid*, vol. 9, no. 2, pp. 1023–1032, 2018.

[10] P.R. Jeyaraj, E. Nadar, A.C. Kathiresan, and S.P. Asokan, "Smart grid security enhancement by detection and classification of non-technical losses employing deep learning algorithm," *International Transactions on Electrical Energy Systems*, 2020.

[11] M.S. Saeed, M.W. Mustafa, U.U. Sheikh, T.A. Jumani, I. Khan, and S. Atawne, "An efficient boosted C5.0 decision-tree-based classification approach for detecting non-technical losses in power utilities," *Energies*, vol. 13, 2020.

[12] J. L. Viegas, P. R. Esteves, and S. M. Vieira, "Clustering-based novelty detection for identification of non-technical losses," *International Journal of Electrical Power & Energy Systems*, vol. 101, pp. 301–310, 2018.

[13] A. Okino Otuoze, M. Wazir Mustafa, I. Ebianga Sofimieari et al., "Electricity theft detection framework based on universal prediction algorithm," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 15, no. 2, pp. 758–768, 2019.

[14] K. V. Blazakis, T. N. Kapetanakis, and G. S. Stavrakakis, "Effective Electricity Theft Detection in Power Distribution Grids Using an Adaptive Neuro Fuzzy Inference System," *Energies*, vol. 13, no. 12, p. 3110, 2020.

[15] K. Gai, Y. Wu, L. Zhu, M. Qiu, and M. Shen, "Privacy-Preserving Energy Trading Using Consortium Blockchain in Smart Grid," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3548–3558, 2019.

[16] Z. Guan, G. Si, X. Zhang et al., "Privacy-Preserving and Efficient Aggregation Based on Blockchain for Power Grid Communications in Smart Communities," *IEEE Communications Magazine*, vol. 56, no. 7, pp. 82–88, 2018.

[17] C. Pop, T. Cioara, M. Antal, I. Anghel, I. Salomie, and M. Bertoncini, "Blockchain Based Decentralized Management of Demand Response Programs in Smart Energy Grids," *Sensors*, vol. 18, no. 2, p. 162, 2018.

[18] J. Gao, K. O. Asamoah, E. B. Sifah et al., "GridMonitoring: Secured Sovereign Blockchain Based Monitoring on Smart Grid," *IEEE Access*, vol. 6, pp. 9917–9925, 2018.

[19] K.P. Yu, L. Tan, M. Aloqaily, H. Yang, and Y. Jararweh, "Blockchain-enhanced data sharing with traceable and direct revocation in IIoT," *IEEE transactions on industrial informatics*, 2021.

[20] M. Li, K. Zhang, J. Liu, H. Gong, and Z. Zhang, "Blockchain-based anomaly detection of electricity consumption in smart grids," *Pattern Recognition Letters*, vol. 138, pp. 476–482, 2020.

[21] M. Signorini, M. Pontecorvi, W. Kanoun, and R. Di Pietro, "BAD: A Blockchain Anomaly Detection Solution," *IEEE Access*, vol. 8, pp. 173481–173490, 2020.

[22] T. Golomb, Y. Mirsky, and Y. Elovici, "CIoTA: Collaborative IoT Anomaly Detection via Blockchain," 2018, https://arxiv.org/pdf/1803.03807.pdf.

[23] R. Casado-Vara, J. Prieto, J. M. Corchado et al., "How Blockchain Could Improve Fraud Detection in Power Distribution Grid," in *International Joint Conference SOCO'18-CISIS'18-ICEUTE'18-San Sebastián, Spain, June 6-8, 2018, Proceedings, Advances in Intelligent Systems and Computing*, J.A. Sáez, H. Quintián, and E. Corchado, Eds., vol. 771, pp. 67–76, Springer, Berlin, Germany, 2018.

[24] F. Jamil, N. Iqbal, Imran, S. Ahmad, and D. Kim, "Peer-to-Peer Energy Trading Mechanism Based on Blockchain and Machine Learning for Sustainable Electrical Power Supply in Smart Grid," *IEEE Access*, vol. 9, pp. 39193–39217, 2021.

[25] J. Abdella, Z. Tari, A. Anwar, A. Mahmood, and F. Han, "An Architecture and Performance Evaluation of Blockchain-based Peer-to-Peer Energy Trading," *IEEE Transactions on Smart Grid*, 2021.

[26] F. Khan, H. Li, Y. Zhang, H. Abbas, and T. Yaqoob, "Efficient attribute-based encryption with repeated attributes optimization," *International Journal of Information Security*, vol. 20, no. 3, pp. 431–444, 2021.

[27] W. Wang, H. Huang, L. Zhang, and C. Su, "Secure and efficient mutual authentication protocol for smart grid under blockchain," *Peer-to-Peer Networking and Applications*, pp. 1–13, 2020.

[28] P.M. Mammen, H. Kumar, K. Ramamritham, and H. Rashid, "Want to reduce energy consumption, whom should we call?" *Proceedings of the Ninth International Conference on Future Energy Systems*, pp. 12–20, 2018.

[29] S. Khalid, A. Maqbool, T. Rana, and A. Naheed, "A Blockchain-Based Solution to Control Power Losses in Pakistan," *Arabian Journal for Science & Engineering*, p. 45, Springer Science & Business Media BV, 2020.

[30] Z. A. Khan, M. Adil, N. Javaid, M. N. Saqib, M. Shafiq, and J.-G. Choi, "Electricity theft detection using supervised learning techniques on smart meter data," *Sustainability*, vol. 12, no. 19, p. 8023, 2020.

[31] G. Liang, S. R. Weller, F. Luo, J. Zhao, and Z. Y. Dong, "Distributed Blockchain-Based Data Protection Framework for Modern Power Systems Against Cyber Attacks," *IEEE Transactions on Smart Grid*, vol. 10, no. 3, pp. 3162–3173, 2019.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 8338616, 14 pages
https://doi.org/10.1155/2021/8338616

Research Article

Towards Achieving Personal Privacy Protection and Data Security on Integrated E-Voting Model of Blockchain and Message Queue

Siriboon Chaisawat and Chalee Vorakulpipat

Information Security Research Team, National Electronics and Computer Technology Center, Pathumthani 12120, Thailand

Correspondence should be addressed to Chalee Vorakulpipat; chalee.vorakulpipat@nectec.or.th

Received 25 May 2021; Revised 2 August 2021; Accepted 8 September 2021; Published 29 September 2021

Academic Editor: Jianting Ning

The growing number of e-voting applications indicates the need in resolving issues that exist in the traditional election model. By integrating with blockchain technology, we could extend the model's capabilities by presenting transparency in logic execution and integrity in data storage. Despite these advantages, blockchain brings in new challenges regarding system performance and data privacy. Due to distributed nature of blockchain, any new updating request needs to be reflected in all network's peers before proceeding to the subsequence requests. This process produces delay and possibility in request rejection due to update conflict. In addition, data removal is no longer feasible since each record is protected by immutable hashed link. To overcome these limitations, the integration model of blockchain and message queue is proposed in this paper. The design addresses security concerns in data exchanging patterns, voter anonymization, and proof of system actor's legitimacy. Performance tests are conducted on system prototypes which were deployed on two different settings. The result shows that the system can perform well in production environment, and introduction of message queue handling scheme can cope with blockchain's errors in unexpected scenarios.

1. Introduction

Voting is an act of delegating one's decision-making power. Traditional election relies on marking and counting ballot papers. Even though this model is still widely used in many nations, the overall procedure is time consuming, inefficient, and prone to error and electoral frauds [1]. Online voting is introduced to overcome the limitations, achieve better efficiency as well as provide convenience to the users. The simplest implementation started from a single server where authentication and vote processing are performed. Despite the presence of data encrypting schemes, all cryptographic operations and key storing are done at server side. In sum, overall system operation remains hidden from the users.

Decentralized Application (DApp) is a new programming approach that allows application to operate on the distributed computer system or trusted P2P network like blockchain. Execution of application's logic is moved to the client side without central authority governing. Also, data directly traverse among only trusted app's clients. Every transaction must be validated against the consensus and pre-agreed rulesets. No malicious action beyond logic agreement shall be carried out. With blockchain, data integrity is preserved by block hash which represents the entire chain's state up to that current point and can be computed by taking previous block hash as an input. Merkle tree [2], illustrated in Figure 1, is a data structure for representing structure of the chain. To validate whether a specific transaction i exists, inclusion can be proved by checking if the tree root (R) is equal to hash of the transaction i that concatenates with its sibling and sequences of sibling of all i's ancestors (π). Let ϕ denote position of i node in Merkle tree, verification can be computed within time complexity of $O(\log n)$, and the equation is defined as follows:

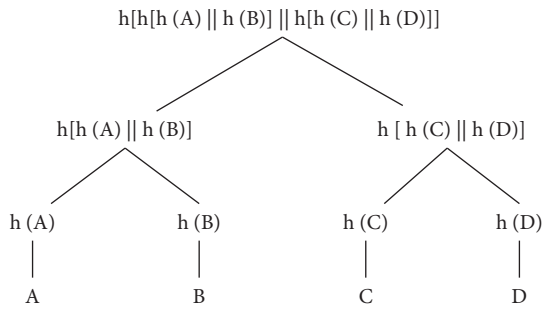


Figure 1: Example of Merkle tree data structure.

$$R = \begin{cases} \text{hash}\left(i \left\| \pi_1 \ \pi_2 \right\| \ldots \pi_{|\pi|}\right), & \text{if } \varnothing < \text{node position of } \pi_1, \\ \text{hash}\left(\pi_1 \, i \left\| \pi_2 \right\| \ldots \pi_{|\pi|}\right), & \text{otherwise.} \end{cases} \tag{1}$$

Since operations on distributed ledger rely largely on an underlying consensus mechanism, a number of consensus mechanisms have been proposed, e.g., Proof of Work (PoW), Proof of Stake (PoS), and Practical Byzantine Fault Tolerance (PBFT). Differences in mechanisms directly influence security and performance of blockchain. To avoid possibility of double spending attack [3] and damages from block reversion due to chain fork, transaction ordering service based on Raft consensus [4] was chosen for this proposed design. With Raft implemented, transactions processing is ensured to be linearizable as there is only a single leader per term. Its responsibilities are to ensure that new updating requests are committed to replicate log and all followers maintain exactly the same order of log entries. In addition, the system can tolerate up to n servers failure given that there are $2n + 1$ servers in total.

Previous research work on implementation of fault-tolerant e-voting systems [5] addresses system design based on the assumption that each district function requires different control mechanisms and encounters different amount of traffic load. The design enables the system to be scaled at functional level, which promotes efficient use of resources and suitable to serve a large-scale election. Nevertheless, there is still a need for refinement in several issues, such as weakness in anonymization schemes, security protection for data transmission over the public Internet, and handling schemes to cope with unexpected circumstances.

Presenting the queuing mechanism that supports reliable data delivery, message queue is found to be an attractive solution as it could be integrated as a middleware for transaction buffering, error handling, and blockchain's event messages listener. Extending the prior study, this paper proposes an integrated model of e-voting by leveraging a messaging protocol. The goal is to overcome technical challenges appeared in previous work and other existing e-vote models, which are data privacy, security, and a need for performance improvement. Refined architecture design and setups of network components are presented, along with error handling schemes to ensure delivery of data.

The subsequent section presents related studies on the e-voting system and different solution schemes to overcome limitations in blockchain, followed by the proposed system design and operation workflow. To elaborate on the introduced concepts, implementation details are presented along with performance evaluation and security assessment on the developed prototypes. The final section summarizes the research findings and provides recommendations for future study.

2. Related Works

Helios [6] is one of the earliest implementations of e-voting systems in which system transparency is promoted by publicly displaying all votes in encrypted form. Even though the design can eliminate external parties' intervention, voter's privacy cannot be guaranteed since all ballots need to be decrypted by the authority who can access to all voter's private keys during the tallying process. Thus, it is possible to sneak into individual's information. Online voting has begun to adopt in many countries [7–11]. For example, in Estonia [12], the system is developed upon national ID infrastructure. Eligible voters must authenticate themselves by dipping ID card and installing a voting application. Once the vote is casted, personal data will be removed and only the candidate selection data will be encrypted using with Estonian National Electoral Committee (ENEC)'s public key. Nevertheless, the analysis study [13] suggests that the system requires major fixes as multiple security loopholes have been found (e.g., server-side malware injection and client-side vote data sniffing). With an advancement in cryptography, some cryptosystems exhibit homomorphic properties in which the mathematical operation $\varnothing$ on a set of encrypted payloads shall be equivalent to encryption of the result from performing operation θ on plaintexts $E(a) \varnothing E(b) \equiv E(a \theta b)$. Such a scheme is beneficial in ballot tallying without requiring prior payload decryption. The study [14] proposes implementation e-voting based on ElGamal encryption scheme which possesses homomorphic properties. During the voting period, voter must construct table of $R * C$ where R represents a selection array in which each cell could either be 0 or 1 and C represents the list of candidates. Vote tallying can be conducted by performing algebraic multiplication on encrypted ballots. The similar concept applies to an implementation of voting scheme based on Paillier cryptosystem [15]. However, proving whether the ballot format is valid (each cell is marked by only 0 or 1) requires generation of all voting possibilities. Thus, the scheme is applicable to the election where voting options are predefined and unchanged. Even though many studies have put efforts on voting scheme design, overall operation remains a black box from user's perspective. This leads to an introduction of decentralized application (DApp) where operation execution is shifted to the client side. All business logic and permission ruleset must be pre-agreed prior to initialization of system process. In addition, an underlying consensus mechanism ensures that distributed ledgers are synchronized and integrity of each record entry is preserved by the cryptographic algorithm. With the presence of immutable

audit trails, it attracts adoption in various business fields. Served as a trusted verification source, blockchain is applied to improve effectiveness in current banking systems [16–18], health insurance [19], supply chain management [20], and right management on digital content sharing platforms [21, 22]. Despite the aforementioned benefits, blockchain encounters two major challenges which are privacy and performance.

Due to immutable property that applies to all record entries, deletion of data is no longer possible once it has been stored on the chain. As stated in the article [23] regarding storing and processing personal data, individual has rights to withdraw consent and request for personal data erasure. Thus, confidential data are recommended to store off-chain as to comply with general data protection regulations (GDPRs) [24]. Nevertheless, many research studies have sought a way to attain on-chain privacy protection for sensitive data [25]. Mixing is one of the approaches, which suggests aggregation of multiple users' transactions into a single transaction in order to prevent attacker from analysing victims' actions. Several implementations of mixing include Mixcoin [26] and CoinJoin [27]. Anonymous signature is another alternative method that presents the concept of using a representative signature for transaction signing instead of the actual performer. Utilizing ring signatures for concealing identity, the scheme [28] ensures that all involved parties can validate the transactions' authenticity without gaining further knowledge of the originating source. Nevertheless, the odds of correct guessing is $1/n$, where n denotes the number of network participants. The probability will be incremented as the number of participants is smaller to 1. Adopting a variant of the TOR and similar to the concept of Mixnet implementation [29], a study [30] introduces an implementation of the Garlic Routing (GOR) on a sidechain in which all transactions created on the main chain are required to route through a sidechain's smart contract mesh in order to conceal creators' identities. The more complexity in sidechain topology, the more often the sender's address is encapsulated and concealed. Nevertheless, the greater complexity of the blockchain logic, the more computational resources are required for the blockchain to validate transactions.

Another key issue is performance. Indeed, there are several factors that contribute to execution competency [31] such as network latency, consensus algorithm, number of participating nodes, and smart contract complexity which are in proportion to the number of read/write operations needed to be executed. Despite the distributed design of blockchain that offers high availability, there exists a limitation in handling large volume of transaction. Common problems that blockchain will encounter in production setup is transaction rejection due to disagreement in transaction generation and processing capacity as well as read-write operation conflicts. To resolve such issues, transaction queuing and error handling mechanism are required.

With lightweight and P2P communication of messaging protocol, it enables emergence of distributed system models, such as device communication in IoT ecosystems. Several protocols serve to standardize message exchange patterns such as MQTT [32], AQMP [33], and ZMTP [34]. With the presence of queueing mechanism, delivery of data is ensured to be in order and take place exactly once. Due to its asynchronous nature and publish-subscribe communication pattern, a message queue has become a popular middleware for synchronizing states among dispersed services. A study [35] leverages message queues for providing consistent updates across databases located in heterogeneous environments. Any change to the database will trigger generation of event messages for acknowledging relevant parties to perform local updates corresponding to the new changes. Apart from data synchronization, message queue is introduced for improving reliability and delivery in data transmission, especially when data production and consumption rate are inharmonious. A study [36] presents use of message queue in replacement of relational database in a mailing system. Traditionally, mail queuing pipeline relies largely on altering rows in relational databases. To prevent the occurrence of operational conflict or bottleneck, the system must avoid large load generation by limiting the number of concurrent active users. With introduction of Apache Kafka [37] as a queuing middleware, tasks beyond capacity limits are added to a queue and held to be processed later without interrupting core operation pipeline.

Multiple studies have proposed integration models of blockchain and message queues. Nowadays, many applications rely heavily on event-driven processing. To avoid alteration or insertion of falsified events into the message stream, blockchain has been introduced for validating authenticity of data exchanged over messaging protocols [38, 39]. On the other way round, message-oriented middleware services have been deployed as blockchain's event listener. Eventeum [40] is one of the implementations for Ethereum Blockchain [41] in which all the blocks and transaction events will be propagated to message bus, and the bus then exposes REST api to application for further processing. Other products of message streaming middleware are OCI Streaming service [42] and Amazon Simple Queue Service (SQS) [43, 44] which offer blockchain event collection and integration with a number of business services, such as user notification (SMS and e-mail) or streaming events directly to business intelligence or analytics engines. However, these products only facilitate the outgoing messages from blockchain which have low traffic density, small chance of bottlenecks, and low error conflicts in contrast to an inward direction which is one of the concerns stressed in this paper.

3. Proposed Integration Design of Blockchain-Based E-Voting Systems

To promote ease of adoption to real-world settings and enhance overall system resilience, the proposed model emphasizes design towards generality while ensuring data protection from end to end. The first part presents system topology design and setup of key components. The later part introduces system operational procedure comprising voter authentication, ballot data transmission, and ballot verification and storing process.

3.1. System Overview. In the architecture design (Figure 2), we assume that voters are not technical experts who can host or run full blockchain nodes. Voters are assumed to reside off-chain and possess personal devices (i.e., smartphone or PC) with Internet connectivity. The authentication process is expected to be performed off the chain with local personnel database managed by the responsible authorities. After a voter is authenticated, he/she is permitted to submit a voting request. To follow the principle of directness stated in system design guidelines [45, 46], point-to-point with no broker messaging protocol is leveraged to ensure that data are transmitted to the chain without passing any intermediaries. In the case that peers are not discoverable within the network (i.e., peers might be located in different networks and do not own public IPs), then messaging brokers are required to be configured as a directory service only for the purpose of facilitating peer discovery [47]. Once data reach the on-chain node, integrity of data and authenticity of the sender will be validated against sets of information records stored on blockchain states. The following display the records which consist of (a) list of eligible token seeds, (b) list of used voting tokens, (c) list of ballots, and (d) lists of nodes' public keys.

(a) Eligible Token Seeds
TS_1
TS_2
....
TS_n

(c) Ballots	
Transaction ID	Selected Candidate
Tx_ID_1	$c_1 \in \{Candidates\}$
Tx_ID_2	$c_2 \in \{Candidates\}$
....	....
Tx_ID_n	$c_n \in \{Candidates\}$

(b) Used Voting Token
TE_1
TE_2
....
TE_n

(d) Nodes' Public Keys	
NODE ID	Public Key
NID_1	PK_{NID1}
NID_2	PK_{NID2}
....	....
NID_n	PK_{NIDn}

Permissioned blockchain is leveraged in order to prevent unnecessary flow of data to irrelevant parties. To ensure that the system can operate with high availability, we leverage the design of a fault-tolerant blockchain network as proposed in previous research work [5].

3.2. Key Components. This proposed design introduces three key actors as defined below. These actors are blockchain client nodes with additional setup of supplementary services. Prior to chain initiation, each node will be assigned to a specific role. A public key for each node ($PK_{<NODE\_ID>}$) is required to be published to the blockchain's shared records to enable key lookup among network components. Two-level role-based permissions (displayed in Table 1) are introduced for defining nodes' accessibility to system resources.

3.2.1. Authenticator Node. This node comprises of two functional components. The first part is an authenticating service, which locally connects to the personal database in order to provide high-security protection on the data. Another part is the blockchain interface, which stores the node's cryptographic credentials and connects authenticator to the running blockchain network.

3.2.2. Proxy Voter Nodes. In order to provide anonymity to the voter, voting token (TE) is proposed to represent voting eligibility instead of referencing to an actual performer. The main tasks of this node are to perform token validation, initiate peer-to-peer connection with trustworthy clients (voters), and trigger submission of voting transactions on behalf of the actual voters. Two main services are implemented in this node: messaging sockets and blockchain interface service. This type of node can be set up as a cluster for load-balancing incoming data packets from clients.

3.2.3. Validator Node. The validator node is responsible for verifying the election results by ensuring that the number of created ballots and used tokens always matches. Also, it facilitates transaction querying in case voters wish to verify their ballots. As displayed in Table 1, the node is permitted only to inspect and query blockchain resources for the purposes of validation. Thus, the node plays no role in modifying the data due to restrictions of the consensus rules.

3.3. System Flow

3.3.1. Authentication and Token Generation. In the beginning, voters are required to authenticate themselves with an authenticating service. To access to this service, authenticator node must configure a private connection channel and provide the configuration to all intended voters. Authentication mechanism and strictness level can vary according to election regulations, which are usually defined by the election commission of each campaign. Once a voter is authenticated, the node then invokes a smart contract for adding a new voter. The returned transaction ID will be used as a token seed (TS). The seed will be recorded to a blockchain's list of eligible token seeds and will also be used for constructing a voting token (TE). Leveraging asymmetric signature JSON Web Token (JWT) [48] format, TE's structure can be divided into three parts, as displayed in Figure 3. Payload contains two types of data. The first type is system data, for example, token expiration date and time. These data help system prescreen packets in order to reduce unnecessary load. Another type is custom information. This part contains an encrypted TS with proxy voter's public key (PK_{PX}). Extending asymmetric key encryption, authenticator's private key (SKAUTH) is used for signature signing. In order to validate token authenticity, one must decrypt the signature part with an authenticator's public key (PK_{AUTH}), which is retrievable from the blockchain state.

3.3.2. Ballot Data Transmission. Once voters are authenticated and obtain TE from an authenticator, they need to establish secure connections to the proxy voter prior to exchanging confidential data. According to Figure 2, a client

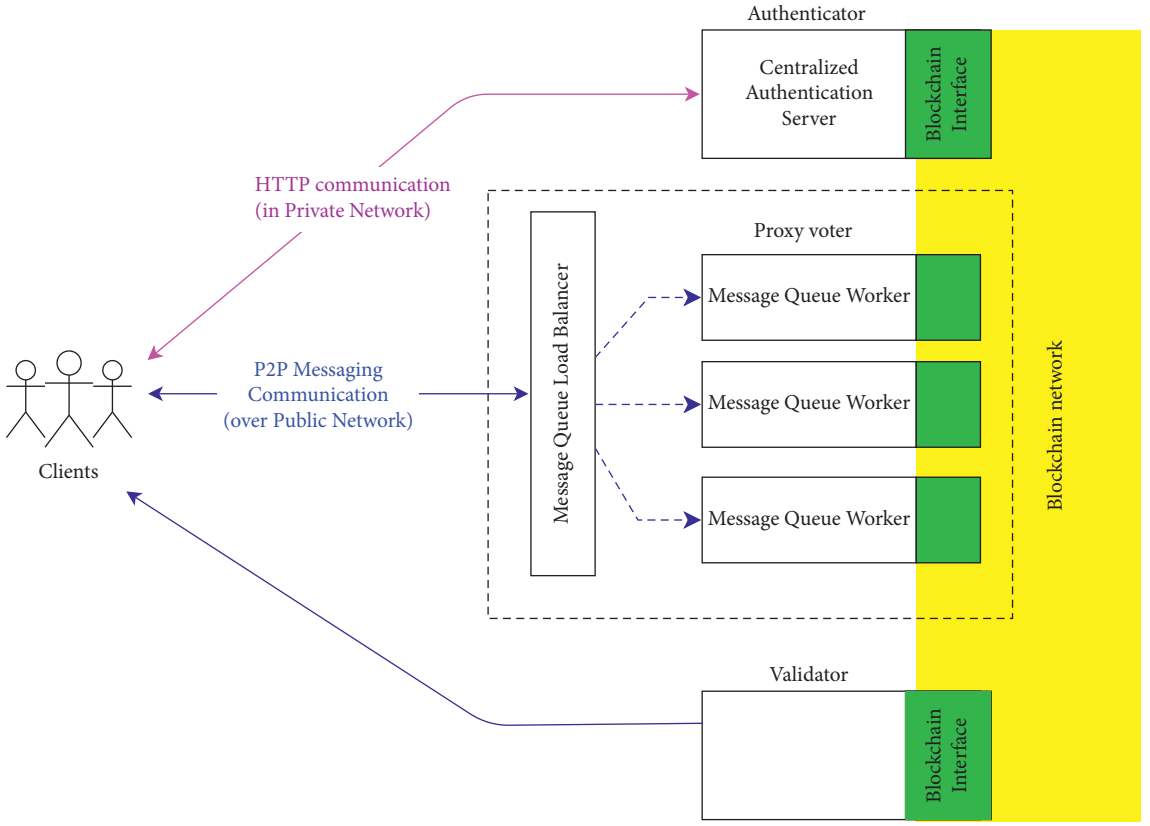


Figure 2: System design and components setup.

Table 1: Blockchain permissions categorized by node roles and permission types.

Permission types node name	Permission on network resources	Permission on logic			
		List of eligible tokens	List of used token	Ballot list	Nodes' public key
Authenticator	Create and retrieve transaction	*R/W*	—	—	*R*
Proxy voter	Create and retrieve transaction	*R*	*R/W*	*R/W*	*R*
Validator	Retrieve transaction	*R*	*R*	*R*	*R*

∗*R* means Read Permission; *W* means Write Permission.

Section	Content
Header	Base64Encoded({ "alg": <hashing algorithm>, "typ" : <type of token>})

Payload	Base64Encoded({
	//example of system data
	"iss": <issuer>,
	"exp": <expiration time>,
	
	//example of custom data
	"identifier_key" : ENC(PK_{PX}, *TS*) })

Signature	HashAlg(Base64Encoded(Header)+ " . " + Base64Encoded(Payload), SK_{AUTH})

Figure 3: Structure of voting token.

can be seen as a remote peer in distributed network. Since authentication support in messaging protocols is limited to device and service level, a voting token is introduced to offer user-level authentication by embedding in the data frame along with the message stream. To authenticate connecting peers at the device level, we leverage handshake mechanism in CurveCP protocol [49] for exchanging 2 sets of key pairs. The first is the permanent (long-term) key pair, which is used for identifying the data source and for generation of the transient key pair. The second is the transient (short-term) key pair, which is used for encrypting exchanged messages. To prevent any kind of intercepting attacks, the transient keys will be destroyed and recreated every time and a connection session is reestablished. For simplicity in explanation, proxy voter nodes and voters are represented as server and client, respectively. Let permanent public keys of client and server be denoted as C and S, while private keys are denoted as c and s. For transient key pairs, let (S', s') and (C', c') denote pairs of short-term public and private keys of server and client, respectively. All voters in the same campaign are assumed to share a common (C, c) and have initial knowledge of server's public key, S. The server is assumed to know voter common public key, C, and possess its initial key pair (S, s). The communication scheme is illustrated in Figure 4.

(1) Connection Validation. Following the CurveCP protocol, upon connection establishment, each voter must generate his/her own transient key $(C', '')$, encrypted C' with C and send to target end. If the receiver can decrypt the packet, it can then be certain that the connection is from legitimate peer and returns encrypted S' in exchange. The connection is now established.

(2) Eligibility Verification and Data Transmission. Goal of this step (2) is to further assure that connection is initiated from valid entity in an election campaign. Goal of this step is to further perform authentication at user-level to ensure that the connecting clients are eligible to vote. Different messaging patterns are introduced for serving the requirement. Figure 5 displays a socket setup for each end. Server (proxy voter) must implement 1 REP socket, 1 PUSH socket, and a pool of PULL sockets while a client (voter) is required to implement 1 REQ, 1 PUSH, and 1 PULL socket. Prior to ballot submission, client must provide proof of voting eligibility by sending REQ's message that contains TE along with client's PULL socket configuration. On receiving the data, the server then validates integrity of the message and authenticity of TE by verifying TE's signature as well as checking if TE is not expired. If TE is valid, encrypted token seed will be decrypted using its private key (SK_{PX}). The result (TS) will be compared against the blockchain's list of eligible token seeds and made sure that TE itself is not in the list of used tokens. If all conditions are met, the server will find and reserve available address listening by its PULL sockets. The address will be returned to the client for further communication. For ballot data submission, the PUSH-PULL pattern is leveraged since this process requires altering blockchain states which often takes some time for data to be processed. With PUSH-PULL type configured, the message queue ensures that any late responses will be captured and persistently maintained until an intended recipient obtains the data. To submit voting data, the client constructs a message containing a selected candidate choice and encrypts it with the transient key. The message is then pushed to the address listening by the server's provided PULL socket.

3.3.3. Ballot Verification and Storing. On receiving a message, the server decrypts data and passes it to a blockchain interface service to convert into blockchain-compatible transaction format and sign with SK_{PX}. Once the transaction is published to the network, it will be validated against permissions at blockchain network layer and business logic layer. At the network layer (lower level), permission is defined for limiting activities that affect system resources or configurations such as adding new members to the chain or submitting new transactions to the network. Permission at business logic layer (higher level) governs individual rights on invoking specific functions on smart contract. Table 1 displays permissions classified by the node's roles.

Proxy voter is the only type of node that is allowed to append ballot transactions to blockchain state. Once the transaction is recorded to the chain, the corresponding TE will be added to the used token list, and the transaction ID will be directly sent via PUSH socket to the client as verifiable evidence. Until the terminating request is fired, PUSH/PULL sockets ensure that a late blockchain's response of transaction ID is successfully delivered to the client's hands.

4. Implementation

To affirm that the design can satisfy all functional requirements, prototypes were developed by utilizing Hyperledger Fabric [50], an open-source framework for developing permissioned blockchains, and ZeroMQ [51], a messaging library that relies on ZMTP messaging protocol, for facilitating client-to-node communication.

4.1. Blockchain Network Implementation. Hyperledger version 1.4 with Raft ordering service was deployed for development of blockchain network. The network was set up on 2 different environments: a single-host setting and multihost setting. For single-host setup, the network was deployed using Docker [52] which is installed on Ubuntu 16.04 LTS machine with 2 CPU cores 2.80 GHz and 12 GB of RAM. An architecture composes of three Raft ordering nodes and two organizations with single peer and single CA each. LevelDB is set up as peer's state database. For multihost setup (Figure 6), 4 virtual machines (Ubuntu 18.04, 2 CPU cores, and 4 GB of RAM) are set up as Kubernetes [53] cluster (1 master node and 3 worker nodes). An architecture is composed of three Raft ordering nodes and two organizations with two peers and single CA each. Each node is deployed as a pod along with its corresponding Cluster IP service. CouchDB is deployed as peer's state database, and pod affinity was configured to ensure that the database is co-located with its corresponding peer node. Hyperledger

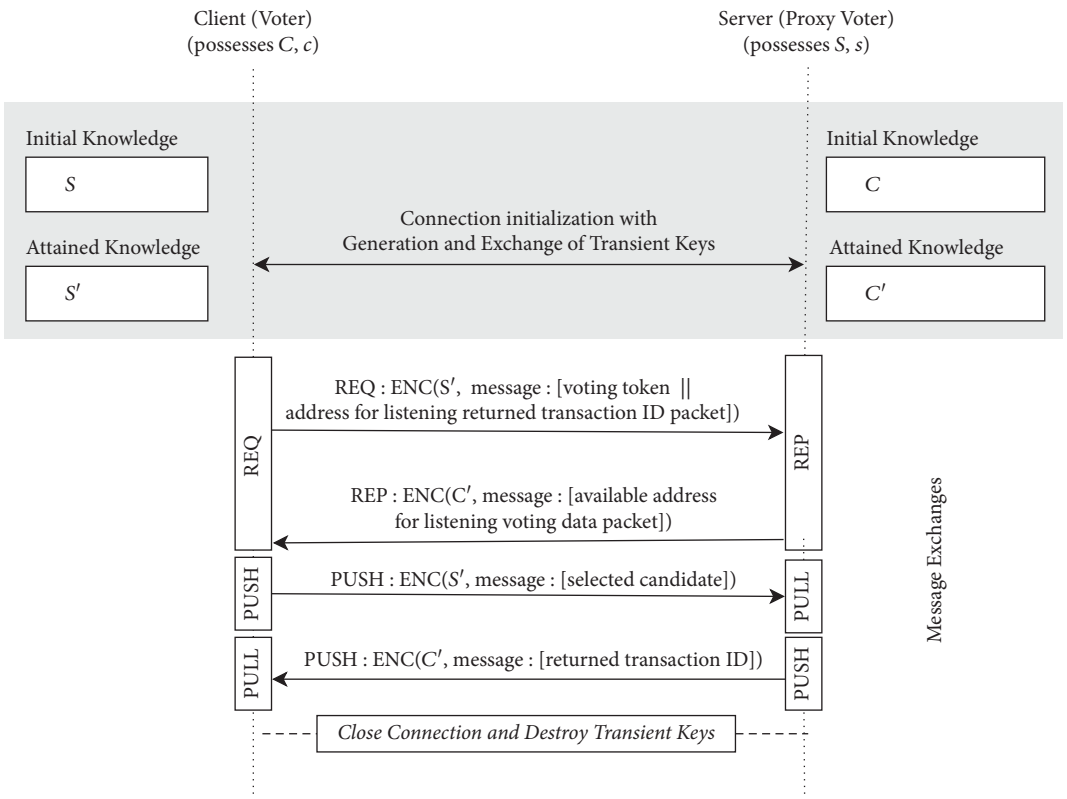


Figure 4: Message exchanges over messaging protocol.

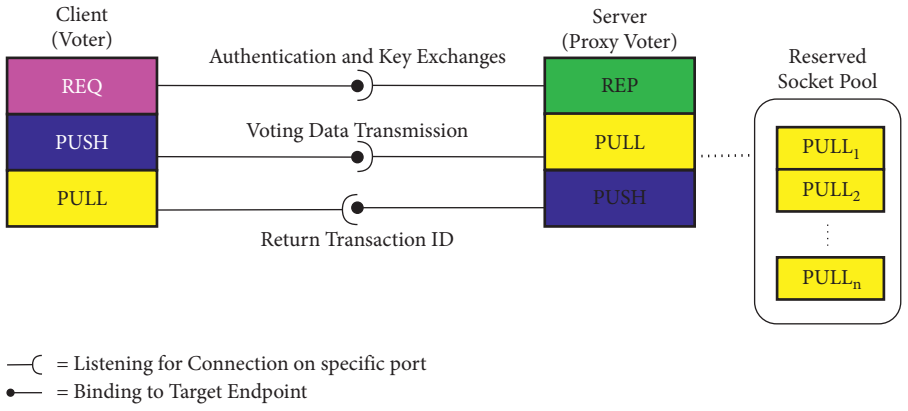


Figure 5: Sockets setup.

Fabric SDK is deployed as a separated service to enable external interaction with the blockchain. The results after conducting a performance test on both environment settings are displayed in the evaluation section.

To restrict the capability of each node's role in interacting with network resources, we override default configuration of Hyperledger Fabric's Access Control List (ACL) in order to customize the policy. The following is a code snippet that defines permission on creating (write) and querying (read) transactions in the form of reader/writer set:

```
Policies:
  Writers:
    Type: Signature
    Rule: "OR("Authenticator.admin," "ProxyVoter.client")"
  Readers:
    Type: Signature
    Rule: "OR("Verifier.admin, "Verifier.peer," " Verifier.client")"
```

To ensure that business logics on smart contract are invoked by the authorized entities, built-in attribute-based access control (ABAC) is used for checking each performer's credentials and validating against predefined conditions. For the purpose of implementation, we import "ClientIdentity" class from "fabric-shim" library in order to retrieve and

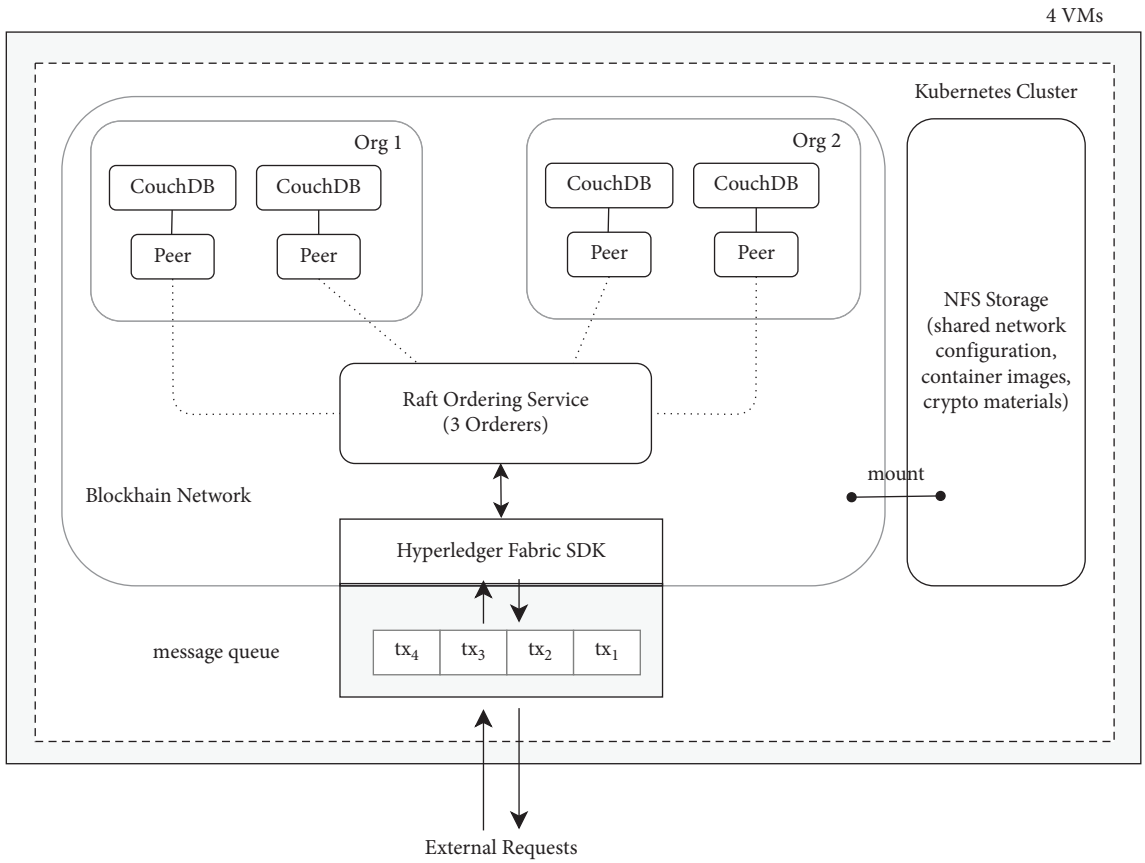


Figure 6: Blockchain network setup on multihost setting.

inspect the credentials of transaction actors. The following is a code snippet of the smart contract's token generation function. Prior to executing the logic, we perform credential checking if the requesting entity is an "authenticator."

```
const ClientIdentity = require("fabric-
shim").ClientIdentity;
function addToken(arguments) {
  let cid = new ClientIdentity(stub);
  if (cid.assertAttributeValue("node_name", "authen-
ticator"){
    //add new eligibility token.
}
  else{ throw new Error("Invoking Entity is Unau-
thorized"); }
}
```

4.2. Message Queue Implementation. ZeroMQ, a brokerless message-oriented networking library based on ZMTP protocol, is leveraged for enabling P2P data exchanges. To make sure that connections are established from/to the legitimated peers, ZeroMQ authentication protocol (ZAP) [54] is used for authenticating connections against a set of known peers' public keys. In order to obtain the keys, CurveZMQ [55], a protocol based on CurveCP and the NaCl cryptographic library, is used for generating 256 bit elliptic curve Curve25519 key pairs.

5. Performance Evaluation and Security Analysis

This section is divided into three parts. The first part presents security analysis on voting tokens, data exchanging schemes over messaging protocol, and state alteration in the blockchain. The second part provides comparative analysis between different designs of blockchain-integrated e-voting models. The final part presents the performance results of overall system operation.

5.1. Security Analysis

5.1.1. Security on Voting Tokens. Design of voting tokens extends the asymmetric signature JWT token format. To verify authenticity and integrity of the token, one must have permission according to ACL and ABAC to access the list of public keys stored on the blockchain. To ensure that voting claim (TS) can be retrieved only by a designated node, the data are encrypted with PK_{PX} to prevent nodes other than proxy voter from accessing. In order to limit the token's usage duration and to reduce unnecessary processing load, "exp" field is used for prescreening packets.

5.1.2. Security on Data Exchanges over Message Queue. Off-chain data exchange leverages CurveZMQ and ZAP which present secure data encryption scheme and authentication mechanism. Introduction of 2 key pairs (permanent and transient) allows each peer to verify if the connection is initiated by a known source and to be certain that exchanging data remain original. Since the transient key is destroyed and recreated for every new connection, analysing user behaviour through intercepting and monitoring packets is no longer feasible. Furthermore, nonce, a random array of arbitrary numbers, is embedded in every outgoing packet. Therefore, messages are ensured to be non-replayable. Results from conducting packet analysis with Wireshark show that the tool cannot extract any information from captured packets.

5.1.3. Security on Blockchain Network. Supported in Hyperledger Fabric, Access Control List (ACL) is utilized for defining permissions on network entities at the system level. The test was conducted by switching between different roles and performing transaction submission as well as attempting to query data from blockchain state. The result of mismatch entities to the ACL policy (defined in Table 1) appears as the following system error.

Error: failed evaluating policy on signed data during check policy [signature set did not satisfy policy].

In order to validate permission at business logic level, Attribute-Based Access Control (ABAC) is evaluated with scenario-based testing, such as letting the proxy voter, who is allowed to submit transactions according to ACL, and invokes a smart contract function token generation (code snippet is displayed in the implementation section). The result displays a custom error, as follows, since calling this function is restricted to the authenticator.

Error: Invoking Entity is Unauthorized.

To confirm that Raft ordering service presents linearizability to blockchain's state transitions, we deployed Hyperledger Explorer [56] for monitoring activities in network (e.g., number of block/transaction creation and size of local ledger maintained by each peer node). By observing through each peer's historical records, the results confirm that transactions are processed in consecutive manner, there is no evidence of cyclic behaviour, and the same copy of ledger is maintained in all peers.

5.2. Comparative Analysis on Blockchain-Integrated E-Voting Models. An integrated model of blockchain and web service is found to be the most popular e-voting model at present. One of the major concerns is an inconsistency in speed of data production and consumption. Message queues and HTTP web services are capable of generating requests at high frequency. By contrast, the capability and speed of processing transactions by blockchain are limited by the consensus mechanism. Comparative analysis was conducted on three different scenarios in order to observe data handling mechanism presented in each design.

Assume that Joe and Mary are eligible voters. Joe uses an e-voting system that connects to web services, while Mary's system is connected to a message queue.

Situation 1. Once users had submitted their votes, the requests were rejected due to unavailability of proxy voter.

In Joe's case, proxy voter is a web server while Mary's is implemented as message queue server. Joe will receive a response notifying of server's unavailability. Sometime later, Joe needs to retry his submission. As a result, Joe will unexpectedly obtain privilege in reconsidering his voting choice. For Mary, her message will remain in the queue on the client side. Once the server becomes available, her message will be pulled out from queue for processing. She will receive an acknowledgement once her message has already been processed.

Situation 2. Clients are unavailable after requests have been submitted.

In this case, Joe and Mary may lose their Internet connection or encounter unexpected system failure after they have already submitted their votes. Since the response from the blockchain is directly sent to Joe's web browser, the message will be disregarded due to unavailability of the web client. Without catching mechanism presented, Joe will not receive any notification indicating transaction success or failure. On the contrary, Mary's response remains in the queue on the server side. As soon as she becomes available, her socket will automatically retrieve data (transaction response) from the binding socket.

Situation 3. A large number of users are using the system and generating a large amount of transactions that are beyond blockchain's capacity limit. Due to network congestion, two possible cases can occur.

Case 1. Transaction is processed with an unpredictable delay in returning response.

Due to the synchronous nature of web services, once request timeout is reached, the web server will terminate connection with the blockchain and disregard further messages even though later the blockchain might return a successful status (Figure 7). From Joe's perspective, he has no idea whether his data have been recorded. As a result, he may retry submission without noticing that his transactions are now doubling in record. In addition, there is a possibility that his second request gains an opportunity to be processed and recorded before his previous request. For Mary, a message queue acts as data buffer in order to release data to the chain at a steady rate and in chronological order. By relieving the amount of blockchain workload, processing delay and odds of transaction rejection (Case 2) are expected to be reduced.

Case 2. Transaction is rejected due to blockchain's processing limitation or conflict in altering data state (multiple transactions try to update on same key address).

In this case, web server will notify Joe with an error message (Figure 8). In order for his data to be recorded, Joe needs to compete with others by retrying the submission until he receives a success response. For Mary in this case, a message queue will act as a cache which will automatically resubmit the request until transaction ID that indicates success is returned (Figure 9). Otherwise, the request message will not be removed out from queue.

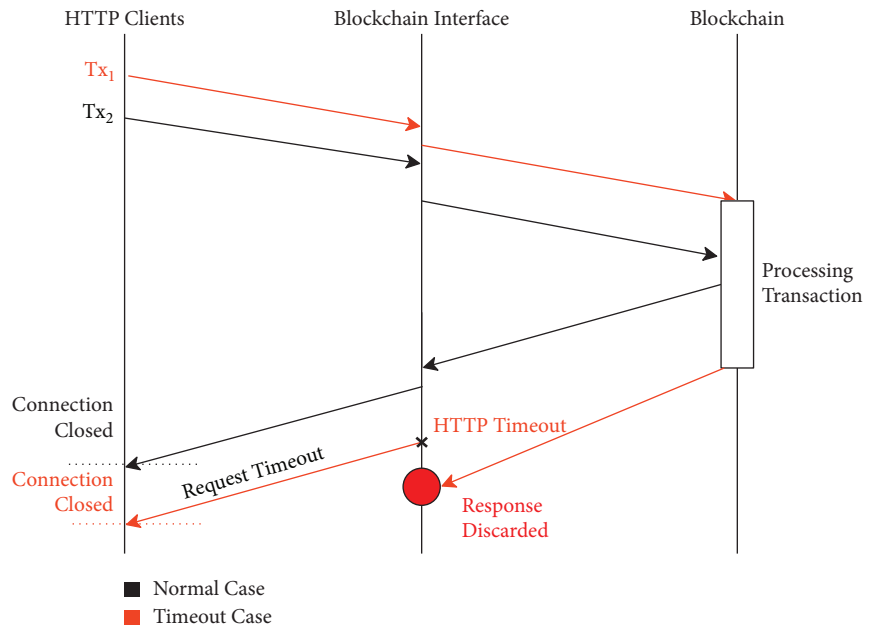


Figure 7: Http request handling scheme in the case of connection timeout.

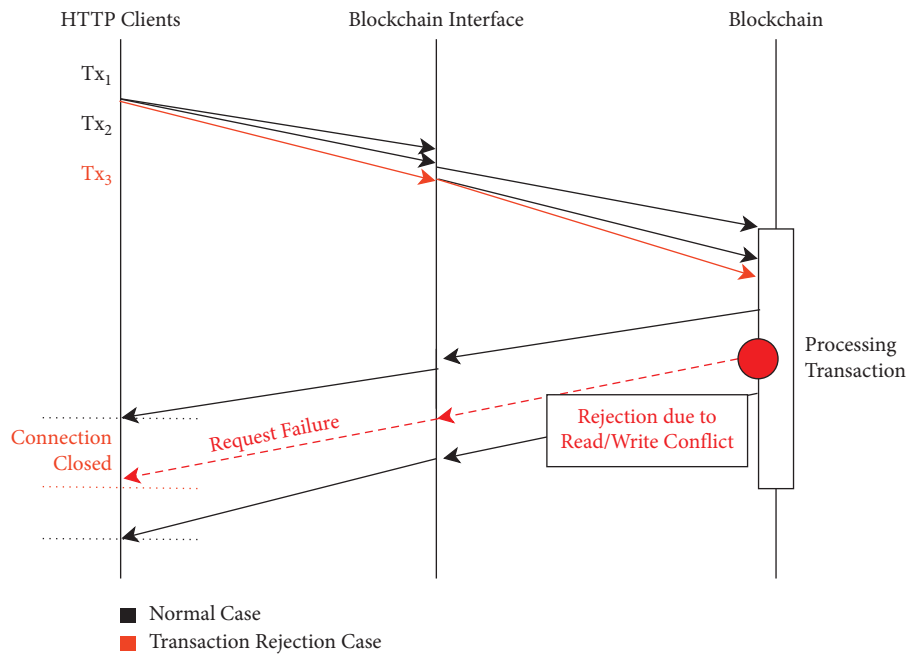


Figure 8: Http request handling scheme in the case of transaction rejection.

Enhancing capabilities of blockchain by providing it with an error handling mechanism, an integrated model of message queue guarantees that all data packets are delivered even in the unexpected circumstances.

5.3. Performance Evaluation

5.3.1. Message Queue. Latency and throughput testing was conducted on a single machine of 4 CPU cores 2.90 GHz with 16 GB of memory. Different communication patterns and socket configurations are deployed to fit different operational requirements. For voting token validation, we leverage a Request-Reply pattern in which REQ and REP sockets were configured at the client and server, respectively. Displayed as follows, the multipart message is constructed and sent over tcp://127.0.0.1 : 3000, which is reserved for the socket's listening address.

Frame1	Voting Token (T_E)
Frame2	Client's PULL Socket Listening Address

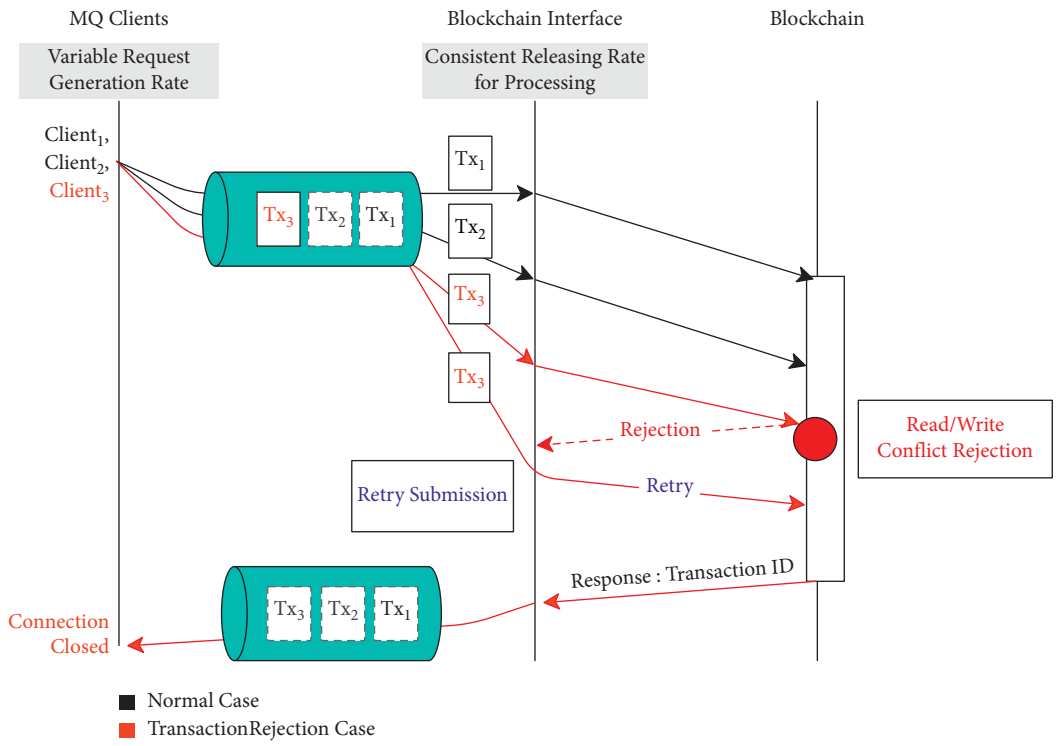


Figure 9: Data handling scheme in message queue (proposed model) in the case of transaction rejection.

For ballot data submission, we leverage a Push-Pull pattern in which PUSH and PULL sockets are configured on the client and server, respectively. A single-framed message containing voter's selected candidate ID is sent over the pre-agreed TCP port.

Frame1	Selected Candidate

For the REQ-REP socket pair, delay is measured in terms of Round-Trip Time (RTT). The equation can be written as $RTT = t_1 + t_2 + P$, where t_1 and t_2 are the message transmission time from sender to receiver and from receiver back to sender, respectively. P denotes the server processing time, which in this test is set to be close to zero. The measuring result shows an average round-trip time of approximately 168.2917 milliseconds. For the PUSH-PULL socket pair, delay is measured in terms of latency (L). The equation can be written as $L_n = I_n - O_n$, where O_n denotes the time when n-th message is emitted from PUSH socket and I_n denotes the time when the receiving peer obtains the complete message data. Average latency in data transmission between this socket pair is 6.642857 milliseconds.

Message throughput can be measured by counting the number of messages processed within a unit of time. Figure 10 displays the results after conducting 10 rounds of test on each socket pair. An average throughput over the REQ-REP socket is 4,297 messages per second, while the average of the PUSH-PULL socket is 5,589 messages per second.

5.3.2. Blockchain Network. To measure performance of Raft in comparison with other types of ordering services supported in Hyperledger Fabric 1.4, an architecture composing

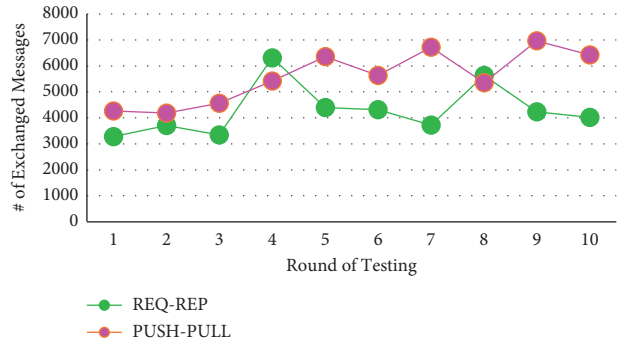


Figure 10: Throughput results on message queue data exchanges over REQ-REP and PUSH-PULL socket pairs.

of 2 organizations with a single peer each was set up. LevelDB is used as peer's state database. The experiment was conducted on Ubuntu 16.04 LTS with 2 CPU cores 2.80 GHz and 12,288 MB of memory. Hyperledger Caliper [57] is deployed as a performance measuring tool. As displayed in Table 2, latency in transaction querying of Raft is close to Solo, a single node ordering service recommended using in only development environment. For production setting, the crash fault-tolerant (CFT) multiordering services such as Kafka and Raft are recommended. According to the results, Kafka exhibits higher latency due to overhead resulted from complex Zookeeper ensembles setup. Raft, on the other hand, outperforms Kafka in both transaction types.

As mentioned in Section 4, two prototyped systems were developed and deployed on two different settings: on single-host docker network machine and on multihost Kubernetes cluster. Hyperledger Caliper is deployed in a separated

Table 2: Average latency in transaction processing.

Transaction types	Ordering service type (total transactions submitted = 300; rate = 100 TPS*)		
	Solo (s)	Kafka (s)	Raft (s)
Query	0.01	0.05	0.01
Invoke	0.33	0.75	0.64

*TPS means transactions per second.

Table 3: Performance of Raft network on different settings.

Deployment settings	Query transaction latency			Invoke transaction latency		
	Min (s)	Max (s)	Avg. (s)	Min (s)	Max (s)	Avg. (s)
Single host (docker)	0.01	0.05	0.01	0.59	0.88	0.68
Multihosts (kubernetes)	0.08	0.36	0.18	0.61	1.96	1.24

container/pod located within the blockchain network. Table 3 displays transaction processing latency of Raft network in both settings. The results show that multihost setting exhibits larger delay on both transaction types. Several external factors contribute to the delay in multihost include location of each server in the cluster, network transmission delay, differences in hardware of underlying physical machines, and so on. Nevertheless, overall performance is in acceptable level and can be improved by adjusting network configuration and/or hardware spec.

6. Conclusion

Exploiting distinct properties of blockchain and message queue, an integrated model for e-voting is proposed with the goal to protect voter's privacy and present transparency and efficiency in overall procedure. In order to preserve voter anonymity, a single-use token is introduced for representing one's eligibility to vote within a specified time period. In order to distribute the tokens, generation of short-term and long-term key pairs following CurveCP protocol allows off-chained data to be transferred securely to the designated receivers. Once ballot packets reach blockchain nodes, transactional actions will proceed corresponding with the pre-agreed consensus. Role-based permissions are defined to restrict nodes' capabilities in accessing and altering blockchain states.

Results after performing scenario-based analysis and performance testing on the prototypes show that the system can perform well in production environment. In addition, introduction of message queue as a data buffering and error handler exhibits competitive advantages over other blockchain-integrated patterns. Recently, many countries start digitalizing their core business processes, and global-wide development of digital identity infrastructure is expected to be put into practice and officially accepted as identity representation. With the use of digital ID in replacement of central authentication server, reliability in voter authentication processes will be enhanced as inputting data from corrupted sources is no longer permitted.

Data Availability

The data used to support this study are available from the first author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] F. Lehoucq, "Electoral fraud: causes, types, and consequences," *Annual Review of Political Science*, vol. 6, pp. 233–256, 2003.

[2] B. Laurie, A. Langley, and E. Kasper, "Certificate transparency," *Certificate Transparency*, 2013.

[3] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008, http://bitcoin.org/bitcoin.pdf.

[4] D. Ongaro and J. Ousterhout, "In search of an understandable consensus algorithm," in *Proceedings of the USENIX Annual Technical Conference'14*, pp. 305–320, June 2014, https://raft.github.io/raft.pdf.

[5] S. Chaisawat and C. Vorakulpipat, "fault-tolerant architecture design for blockchain-based electronics voting system," in *Proceedings of the 17th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, pp. 116–121, Bangkok, Thailand, November 2020.

[6] B. Adida, "Helios: web-based open-audit voting," in *Proceedings of the 17th Conference on Security Symposium*, pp. 335–348, Berkeley, CA, USA, July 2008.

[7] B. Ian, C. Jordi, G. David, and S. Guasch, "An overview of the ivote 2015 voting system," 2015, https://www.elections.nsw.gov.au/about_us/plans_and_reports/ivote_reports.

[8] M. Epp, "Towards remote e-voting: estonian case," in *Proceedings of the Electronic Voting in Europe-Technology, Law, Politics and Society, Workshop of the ESF TED Programme together with GI and OCG*, pp. 83–100, Schloß Hofen/Bregenz, Lake of Constance, Austria, July 2004.

[9] G. Kristian, "The Norwegian Internet voting protocol," in *Proceedings of the International Conference on E-Voting and Identity*, pp. 1–18, Springer, Tallinn, Estonia, September 2011.

[10] S. Wolchok, E. Wustrow, D. Isabel, and J. A. Halderman, "Attacking the Washington, D.C. Internet voting system," in *Proceedings of the International Conference on Financial Cryptography and Data Security*, pp. 114–128, Springer, Berlin, Germany, February 2012.

[11] J. Gerlach and U. Gasser, *Three Case Studies from Switzerland: E-Voting*Berkman Center Research Publication, Cambridge, MA, USA, 2009, https://cyber.harvard.edu/sites/cyber.law.harvard.edu/files/Gerlach-Gasser_SwissCases_Evoting.pdf.

[12] R. M. Alvarez, T. E. Hall, and A. H. Trechsel, "Internet voting in comparative perspective: the case of Estonia," *PS: Political Science & Politics*, vol. 42, no. 03, pp. 497–505, 2009.

[13] D. Springall, T. Finkenauer, Z. Durumeric et al., "Security analysis of the Estonian Internet voting system," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pp. 703–715, Scottsdale, AZ, USA, November 2014.

[14] L. Li, "An electronic voting scheme based on ElGamal homomorphic encryption for privacy protection," *Journal of Physics: Conference Series*, vol. 1544, 2020.

[15] M. A. Will, B. Nicholson, M. Tiehuis, and R. K. L. Ko, "Secure voting in the cloud using homomorphic encryption and mobile agents," in *Proceedings of the 2015 International Conference on Cloud Computing Research and Innovation (ICCCRI)*, pp. 173–184, Singapore, October 2015.

[16] Q. K. Nguyen, "Blockchain - a financial technology for future sustainable development," in *Proceedings of the 3rd International Conference on Green Technology and Sustainable Development (GTSD)*, pp. 51–54, Kaohsiung, Taiwan, November 2016.

[17] K. Fanning and D. P. Centers, "Blockchain and its coming impact on financial services," *Journal of Corporate Accounting & Finance*, vol. 27, no. 5, pp. 53–57, 2016.

[18] H. Lycklama à Nijeholt, J. Oudejans, and Z. Erkin, "DecReg," in *Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts BCC'17*, pp. 29–34, Abu Dhabi, UAE, April 2017.

[19] Deloitte, "Blockchain applications in insurance," 2016, https://www2.deloitte.com/content/dam/Deloitte/ch/Documents/innovation/ch-en-innovation-deloitte-blockchain-app-in-insurance.pdf.

[20] V. Paliwal, S. Chandra, and S. Sharma, "Blockchain technology for sustainable supply chain management: a systematic literature review and a classification framework," *Sustainability*, vol. 12, no. 18, 2020.

[21] J. Kishigami, S. Fujimura, H. Watanabe, A. Nakadaira, and A. Akutsu, "The blockchain-based digital content distribution system," in *Proceedings of the 2015 IEEE Fifth International Conference on Big Data and Cloud Computing*, pp. 187–190, Dalian, China, August 2015.

[22] S. Fujimura, H. Watanabe, A. Nakadaira, T. Yamada, A. Akutsu, and J. J. Kishigami, "BRIGHT: a concept for a decentralized rights management system based on blockchain," in *Proceedings of the IEEE 5th International Conference on Consumer Electronics-Berlin (ICCE-Berlin)*, pp. 345-346, Berlin, Germany, September 2015.

[23] J. Benet, "IPFS—content addressed, versioned, p2p file system," 2014, https://arxiv.org/abs/1407.3561.

[24] European Parliament, "Council, Regulations Council (E. U) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)," *Official Journal of the European Union*, vol. 59, 2016, http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=OJ:L:2016:119:TOC.

[25] R. Zhang, R. Xue, and L. Liu, "Security and privacy on blockchain," *ACM Computing Surveys*, vol. 52, no. 3, 2019.

[26] J. Bonneau, A. Narayanan, A. Miller, J. Clark, J. A. Kroll, and E. W. Felten, "Mixcoin: anonymity for bitcoin with accountable mixes," *Financial Cryptography and Data Security*, vol. 8437, pp. 486–504, 2014.

[27] G. Maxwell, "CoinJoin: Bitcoin privacy for the real world," 2013, https://bitcointalk.org/?topic=279249.

[28] O. Kurbatov, P. Kravchenko, O. Shapoval et al., "Anonymous decentralized e-voting system," in *Proceedings of the International Workshop on Conflict Management in Global Information Networks (CMiGIN 2019)*, Lviv, Ukraine, 2019.

[29] D. L. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms," *Communications of the ACM*, vol. 24, no. 2, pp. 84–90, 1981.

[30] R. Parizi, S. Homayoun, A. Yazdinejad, A. Dehghantanha, and K. K. R. Choo, "Integrating privacy enhancing techniques into blockchains using sidechains," in *Proceedings of the IEEE Canadian Conference of Electrical and Computer Engineering (CCECE)*, pp. 1–4, Edmonton, Canada, May 2019.

[31] H. Buch, "Improving performance and scalability of blockchain networks," 2019, https://www.wipro.com/blogs/hitarshi-buch/improving-performance-and-scalability-of-blockchain-networks/.

[32] Organization for the Advancement of Structured Information Standards, MQTT, "The standard for IoT messaging," 2020, https://mqtt.org/.

[33] OASIS, "Advanced message queuing protocol (AMQP) overview," 2012, http://docs.oasis-open.org/amqp/core/v1.0/os/amqp-core-overview-v1.0-os.html.

[34] iMatix Corporation & Contributors, "ZeroMQ message transport protocol," 2021, Accessed on: Jan. 8, 2021. [Online]. Available: https://rfc.zeromq.org/spec/23/.

[35] Z. Y. Lu and Z. B. Guo, "A method of data synchronization based on message oriented middleware and xml in distributed heterogeneous environments," in *Proceedings of the International Conference on Artificial Intelligence and Industrial Engineering*, pp. 210–212, Phuket, Thailand, July 2015.

[36] A. Cansever, U. Özel, O. Akin et al., "A distributed message queuing mechanism for a mailing system with high performance and high availability," in *Proceedings of the 2018 6th International Conference on Control Engineering & Information Technology (CEIT)*, pp. 1–5, Istanbul, Turkey, October 2018.

[37] Apache Software Foundation, "Apache Kafka," 2017, https://kafka.apache.org/documentation.

[38] Y. Kim and J. Park, "Hybrid decentralized PBFT blockchain framework for OpenStack message queue," *Human-centric Computing and Information Sciences*, vol. 10, no. 1, 2020.

[39] C. Esposito, F. Palmieri, and K.-K. R. Choo, "Cloud message queueing and notification: challenges and opportunities," *IEEE Cloud Computing*, vol. 5, no. 2, pp. 11–16, 2018.

[40] Kauri Team, "Eventeum," 2021, https://github.com/eventeum/eventeum.

[41] G. Wood, "ETH EREUM: A secure decentralised generalised transaction ledger," 2014, https://gavwood.com/paper.pdf.

[42] O. Corporation, "Oracle cloud infrastructure streaming service," 2021, https://docs.oracle.com/en-us/iaas/Content/Streaming/Concepts/streamingoverview.html.

[43] Amazon Web Services, "Inc, Amazon Simple queue service," 2021, https://aws.amazon.com/sqs/.

[44] E. Baizel, "Building an event-based application with amazon managed blockchain," 2020, https://aws.amazon.com/blogs/database/building-an-event-based-application-with-amazon-managed-blockchain/.

[45] D. A. Gritzalis, "Principles and requirements for a secure E-voting system," *Computers & Security*, vol. 21, no. 6, pp. 539–556, 2002.

[46] M. Christian, "Design of distributed voting systems," 2017, https://arxiv.org/abs/1702.02566.

[47] iMatix Corporation & Contributors, "Discussion on broker vs brokerless messaging models," 2017, http://wiki.zeromq.org/whitepapers:brokerless.

[48] AuthO Inc, "Introduction to JSON web tokens," 2021, https://jwt.io/introduction.

[49] D. J. Bernstein, "Curvecp: usable security for the internet," 2010, http://curvecp.org.
[50] E. Androulaki, A. Barger, V. Bortnikov et al., "Hyperledger fabric: a distributed operating system for permissioned blockchains," in *Proceedings of 13th EuroSys Conference*, Porto, Portugal, April 2018, https://arxiv.org/pdf/1801.10228.
[51] "ZeroMQ - an open-source universal messaging library," 2021, https://zeromq.org/.
[52] M. Dirk, "Docker: lightweight linux containers for consistent development and deployment," *Linux Journal*, vol. 2014, no. 239, 2014.
[53] The Linux Foundation, "Kubernetes," 2021, https://kubernetes.io/.
[54] iMatix Corporation & Contributors, "ZeroMQ Authentication Protocol (ZAP)," 2013.
[55] iMatix Corporation & Contributors, "CurveZMQ," 2013, http://curvezmq.org.
[56] The Linux Foundation, "Hyperledger explorer," 2021, https://www.hyperledger.org/use/explorer.
[57] The Linux Foundation, "Hyperledger caliper," 2020, https://www.hyperledger.org/use/caliper.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 3091104, 11 pages
https://doi.org/10.1155/2021/3091104

Research Article

A Blockchain-Based IoT Cross-Domain Delegation Access Control Method

Chao Li,[1] **Fan Li**,[1,2] **Lihua Yin**,[1] **Tianjie Luo**,[1,2] **and Bin Wang**[3]

[1]*Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou 510700, China*
[2]*Guangxi Key Laboratory of Cryptography and Information Security, Nanning 541004, China*
[3]*College of Electrical Engineering, Zhejiang University, Hangzhou 310058, China*

Correspondence should be addressed to Lihua Yin; yinlh@gzhu.edu.cn and Bin Wang; bin_wang@zju.edu.cn

Received 18 June 2021; Accepted 23 August 2021; Published 11 September 2021

Academic Editor: Yinghui Zhang

The collaborative demand in the Internet of Things (IoT) is becoming stronger. One of the collaborative challenges is the security of interoperability between different management domains. Although cross-domain access control mechanisms exist in IoT, the majority of them are based on a trusted third party. In addition, the heterogeneity of multidomain policies makes it difficult for authority delegation to satisfy the principle of least authority. In this paper, we propose a blockchain-based IoT cross-domain delegation access control method (CDDAC). The delegation-trajectory-on-blockchain strategy proposed enhances the scalability of the cross-domain delegation system. The presented multidomain delegation trajectory aggregation scheme supports the forensic analysis of the cross-domain delegation system. The performance of CDDAC is evaluated in the Ropsten, which is the Ethereum's official public blockchain test network. The experimental results show that CDDAC has faster delegation verification speed and higher decision-making efficiency than existing work, demonstrating the lightweight and scalability of the method.

1. Introduction

Internet of Things (IoT) has been widely used in many fields, such as smart healthcare [1], smart transport [2], and smart homes [3]. Among these fields, some scenarios have begun to trend towards requiring IoT devices from different domains to share data or collaborate, which makes a significant difference from traditional single-domain applications. In a traditional single-domain application, IoT devices belong to the same domain, in which the domain administrator could manage the devices security policies overall. For example, we assume that a hospital has only one domain, in which many IoT devices such as smart connected-beds, wearable ECG monitors, etc., are deployed to collect patient-related data. The domain administrator could define security policies to manage all the devices in the hospital, to specify which devices can be accessed by whom and under what circumstances; for example, a patient's ECG monitor can be accessed by his family and nurses. In contrast, in a cross-domain application, the users, devices, and data belong to different domains. Many functions require devices and data shared in multiple domains to be achieved. Assume an application requirement that "*If there is a traffic jam, the ambulances nearby then get the alarms and some new recommended navigation routes*," it requires the traffic domain and the hospital domain to collaborate. Traditional single-domain access control mechanisms are difficult to meet this requirement, since each domain administrator cannot manage the other domain devices.

To fulfil the IoT cross-domain access control requirement, Payne et al. [4] connect IoT domains according to certain agreements to form a virtual alliance. They propose the National Health Information Network (NHIN) uniting the IoT domains of multiple hospitals to form a virtual alliance of medical systems. The alliance facilitates a smoother information flow between doctors and patients and initially solves the problem of cross-domain access control. However, this kind of method faces challenges in cross-domain delegation [5]. Access right delegation is one of the ways to realize IoT cross-domain connection [6]. Due

to the heterogeneity of security constraints in IoT domains, universal authorization protocols are difficult to obtain. Authorization capabilities are commonly granted to users in the IoT domain by special carriers (e.g., the OAuth token, the secret URL of IFTTT), allowing independent decisions on whether to transfer access rights to users in another domains for reasons of convenience or emergency response, what is called cross-domain delegation. For example, a patient delegates access right of his wearable ECG monitor to his families and caregivers. In the case of a fire, the homeowner needs to delegate the capability to open smart locks to firefighters, which involves an ad hoc access right delegation. While decreasing the decision-making pressure of the trusted central server, delegation mechanism satisfies the decentralized and dynamic characteristics of IoT, which is considered an indispensable feature in large-scale IoT scenarios.

Previous researches have focused on cross-domain delegation and access control by trusted third parties [7,8], while trusted third parties are at risk of being attacked [9]. Furthermore, cross-domain key distribution is a challenging problem [10], and user privacy is more likely to be exposed. Blockchain, as a decentralized mechanism that does not require a trusted third-party potential, is seen as an opportunity to solve the problem of cross-domain delegation. Existing blockchain-based methods rely on two kind strategies of policy-on-blockchain [11–13] and right-on-blockchain [14,15], both of which can cause difficulties in policy changing and are constrained by blockchain performance. To address the shortcomings of existing work, a blockchain-based cross-domain delegation access control method, CDDAC, is proposed in this paper.

CDDAC leverages the decentralized characteristics of the blockchain, to ensure the feasibility and security of cross-domain delegation, and provides evidence for forensic analysis after malicious events have occurred. Our contributions are as follows:

We propose CDDAC, a cross-domain delegation access control scheme based on the blockchain. The delegation trajectory-on-blockchain strategy makes CDDAC more flexible and usable than other methods.

Based on CDDAC, we propose a multidomain delegation trajectory aggregation method with goal-directed logging. It supports forensic analysis of intra/cross-domain delegation and contributes to suspect validation and accountability after malicious behavior occurs.

We conduct simulation experiments of CDDAC on IoT devices. Experiment results show that CDDAC has a faster token verification speed compared with CapBAC and BlendCAC. In the meantime, CDDAC can maintain a high consensus efficiency in the blockchain system.

The remainder of this paper is organized as follows. In Section 2, we describe the work related to the use of blockchain for cross-domain access control. Section 3 mainly describes the system design of CDDAC. Section 4 describes the multidomain delegation trajectory aggregation scheme and the basic design of forensic analysis in CDDAC. Section 5 performs simulation experiments on CDDAC and analyzes the corresponding experimental results. Section 6 summarizes this paper.

2. Related Work

The existing blockchain-based cross-domain access control strategy can be divided into policy-on-blockchain and right-on-blockchain.

2.1. Policy-on-Blockchain. Writing access control policies into smart contracts is a common method in blockchain-based solutions. Novo et al. [10] propose a blockchain-based IoT access control framework, using a management hub to manage IoT devices and making decisions in accordance with the access control policies on the blockchain. But the access control policies need to be managed in a consistent way, and the policy definition or changing is complicated. As a method supporting flexible customization of access control policies, Ouaddah et al. [11] propose FairAccess, a blockchain-based access control framework. FairAccess allows resource owners to define access control or delegation policies and renew them to the blockchain. When there are more participants, the cost of policy synchronization will become unbearable. As a customized solution for cross-domain access control, IoT Passport proposed by Tang et al. [12] is a cross-platform blockchain framework, which uses blockchain authentication, authorization, and trust as the cornerstone to achieve cross-platform collaboration. However, the cross-domain policies of IoT Passport are recorded on the blockchain, which makes policy changing more expensive. Similarly, Gauhar et al. [16] propose a decentralized blockchain-based IoT access control framework, xDBAuth, used for single-domain or cross-domain access control and implemented a platform authentication mechanism in blockchain. The delegation policies of xDBAuth are saved on blockchain, facing difficulties in policy changing. In general, for the policy-on-blockchain strategy, the design of uploading policies to blockchain not only brings a huge synchronization burden, but also is not convenient for policy changing, which happens so frequently in access control systems.

2.2. Right-on-Blockchain. It is a conventional design to substantively pass access rights, which is used to reduce the complexity of delegation. Recently, this design has also been used in cross-domain access control works. Yuan et al. [17] attribute the challenge of cross-domain access control to security and consistent delegation policies and non-bypassable and transitive delegation control. Maesa et al. [18] propose a blockchain-based access management framework, which describes the operation of permission exchange. But the access control policies and rights delegation process are visible publicly on blockchain, which leaks the privacy of users. A similar study is BlendCAC proposed by Xu et al. [14]. BlendCAC uses capability tokens

managed by smart contracts to delete or revoke permissions. Although the overhead in authentication and token verification has been proven acceptable, the delegation process is much more complex. Nakamura et al. [15] propose a decentralized and trusted capability-based access control method. The issue is raised about the limitations of BlendCAC's entrusted records, and smart contracts are used to save and manage tokens. But it only designs a new token and does not solve the problem of poor scalability of such solutions. In short, for the right-on-blockchain strategy, the additional restrictions imposed will affect the delegation freedom, which goes against the original intention of free delegation.

In conclusion, the strategies of policy-on-blockchain and right-on-blockchain are difficult to achieve sufficiently flexible and scalable access control. We improve the above defects with the help of capability tokens and the strategy of delegation trajectory-on-blockchain. The proposed cross-domain delegation trajectory aggregation scheme provides support for forensic analysis. We only store the hash of the delegated trajectory on blockchain. The decision-making and analysis process are implemented without the blockchain. Our method does not involve the update of access control policies on blockchain, which brings a stronger scalability.

3. System Design

We propose a blockchain-based IoT cross-domain delegation access control method, which is called CDDAC. CDDAC uses the blockchain to ensure the reliability of the capability delegation trajectory in each alliance. To ensure the flexibility of delegation and reduce the complexity of smart contracts, we delegate the right of access decision-making to users and domain managers. The goal-directed logging and forensic analysis of the delegation trajectory provide more security while implementing access control policies.

CDDAC's architecture is shown in Figure 1. Similar to CapBAC [19], we use capability tokens to represent the access right to be delegated. Intra/cross-domain access requests from token owner will be centralized to the domain manager. After the legality verification and the access control policies decision, the delegation topology is generated and aggregated with the trajectory in the Delegation Trajectory Database (DTDB). Cross-domain access requests will be submitted to the managers of other domains; then, the procedure of cross-domain access will be completed. The policy changing is simplified to DTDB updating. The DTDB hash will be packaged and broadcast to all domain managers. After the transaction procedure, the smart contract with the hash will be redeployed to ensure the reliability of trajectory data in DTDB.

In this section, we will introduce the system design of CDDAC, including the capability token structure, domain manager, and delegation process.

3.1. Capability Token Structure. We design a capability token structure for CDDAC. The classic CapBAC uses nesting tokens to trace back the root token and verify its legitimacy to confirm whether the capability is valid. This design makes the size of the capability token increase with the depth of delegation, which is not scalable for IoT. Due to the size limit of smart contracts, nested tokens are not suitable for being used in the blockchain. We find that the token structure is essentially meant to save the delegation trajectory and confirm its legitimacy. However, retaining the delegator's token in each token is a waste of resources. When the delegation trajectory is extracted and its reliability is guaranteed, a nonnested capability token can be obtained. It will greatly reduce the size of tokens and the token processing overhead and make it possible to integrate with the blockchain. We propose the structure of CDDAC's capability token based on the above motivation.

The capability token is defined as

$$\begin{aligned} \mathrm{Cap}_{\mathrm{cross}} &= \{ID_A, ID_B, Cap_{\mathrm{root}}, \mathrm{Trace}, ET, C, \mathrm{Signature}_C\}, \\ \mathrm{Signature}_D &= f\left(ID_A, ID_B, \mathrm{Cap}_{\mathrm{root}}, ET, \mathrm{Trace}, C\right). \end{aligned} \tag{1}$$

(i) ID_A: *User A*'s identity document

(ii) ID_B: *User B*'s identity document

(iii) Cap_{root}: the token of the root owner with the capability

(iv) *Trace*: the delegation trajectory after *User A* adds *Node B*

(v) *ET*: the validity period of the capability

(vi) *C*: the blank bits containing context-related information, or whether the capability can be delegated

(vii) $Signature_C$: signature of capability token by domain manager

(viii) f: one-way hash function

In the capability token used by CDDAC, ID is used to identify the virtual/real identity of users and is used by the delegator to verify the token validity. Cap_{root} is the original capability token. The signature of the root user can be used to verify authenticity. *Trace* is the capability trajectory that the delegator knows, which is constructed by the delegator. The domain manager confirms the authenticity of *Trace* based on the records saved in the DTDB. It is worth noting that the *Trace* in a single capability token is incomplete. The complete delegation trajectory is aggregated by the domain manager based on the *Traces* in all capability tokens. *ET* records the validity period of the capability, which is used to judge the legality of the capability. *C* is used to save the context information, or to mark whether the capability can be delegated. $Signature_C$ is the digital signature of the domain manager, which means that the token has been received and the legality has been initially verified.

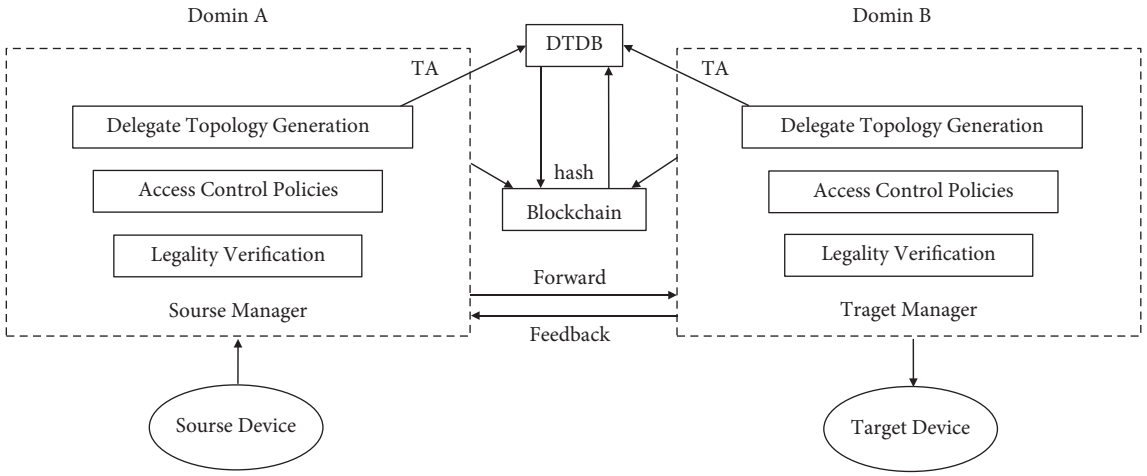


Figure 1: The basic architecture of CDDAC.

3.2. Domain Manager. The domain manager (DM) is responsible for collecting delegated trajectories, updating the DTDB, and updating the smart contract. All cross-domain access will be recorded and uploaded to DTDB by the domain manager, to guarantee the reliability of access with the help of blockchain.

3.3. Delegation Process. The delegate process of CDDAC is shown in Figure 2. First, the delegator sends a delegation request to the domain manager and presents the capability token constructed by itself. This step is to ensure that the domain manager can get a complete delegation trajectory. After the trajectory and root token are certified to be legal, it is deemed that the delegation meets the basic security requirements. The capability token will be digitally signed and transmitted to the delegator. Then, the domain manager will update the information in DTDB according to the delegation trajectory of the effective capability token. The delegated trajectory aggregation will be completed by the domain manager and uploaded to the DTDB. The DTDB returns the new *Hash* and redeploy the smart contract. Then, the access right delegation is completed.

3.4. Intra/Cross-Domain System Implementation

3.4.1. Intradomain System. The blockchain is not required for the intradomain access control system, but the delegation trajectory stored in DTDB is necessary. Figure 3 shows the delegated access control method in a single-domain system. The root token is issued by the cloud platform. To improve the scalability of tokens, we use a single-layer structure of capability tokens and adopt a weakly coupled central structure, which means that the domain manager only participates in the issuance procedure of the token, not in the actual use process of the capability. This method guarantees the freedom of delegation.

The delegation procedure is the same as described in Section 3.3. During the delegation process, all delegation trajectories will be aggregated and stored in DTDB. DM and DTDB record the delegation trajectory and do not evaluate the delegation accuracy. When the resource server (RS) receives an access request with a capability token, it needs to verify whether the token has been authenticated by the DM. If the user accesses RS for the first time, RS can request the DM to verify the security according to the current situation. For example, a new user shows a capability token and asks to open the classroom door at 0 : 00. RS judges that the behavior is abnormal according to the basic access control policies. Then, it queries the delegation trajectory, sends a warning to DM, and requests security analysis. RS can receive DM's instructions to deal with delegation changes, revocations, or other events. Context from IoT sensors can also be used for decision-making.

3.4.2. Cross-Domain System. Figure 4 shows a cross-domain access control system. We divide the cross-domain system into four layers according to their functions. The Delegation Layer is composed of users in multiple domains, including the subject and object of access control. The Access Control Layer consists of multiple-domain managers. In Figure 4, the domain managers of *Domain A* and *Domain B* are two domain servers (DS). This layer is mainly responsible for the collection and cross-domain access control of the multi-domain system. The Trajectory Storage Layer is the DTDB, which is responsible for the storage of delegate trajectories and the generation of new hash. The Blockchain Layer is the blockchain, which provides reliability guarantee for the delegation trajectory [20].

To make a final judgment on the legality, the root token must exist in each issued capability token. Therefore, cross-domain access requires root tokens in other domains, as the seed of the delegation chain. As shown in Figure 4, if a user in *Domain B* wants to access resources in *Domain A*, the resources in *Domain A* must delegate capability to *Domain B*. For example, if *Alice* issues the *Root Token* to *Bob*, *Bob* can access *Alice*'s resources in *Domain A* and continue to delegate the *Token αi*, so that other nodes in *Domain B* have the capability to access the resources. All delegation behavior described above should be recorded in DTDB.

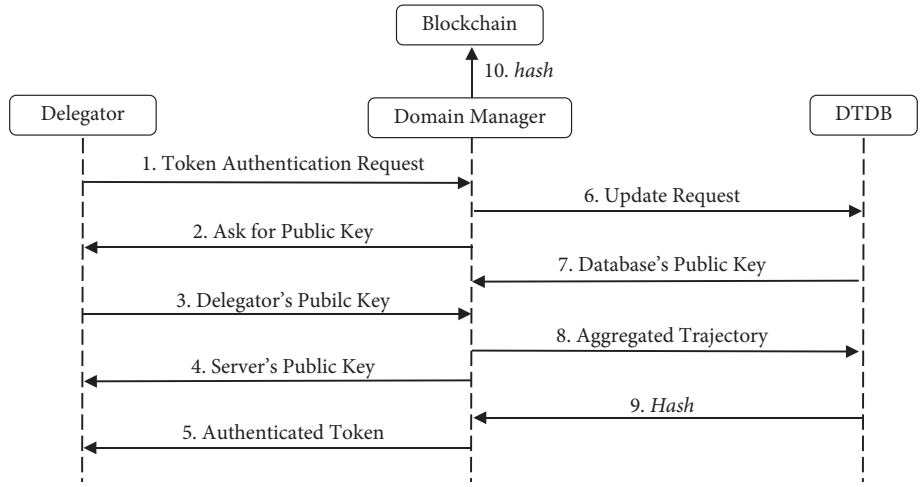


Figure 2: Delegation process of CDDAC.

The basic data structure in smart contract is a six-tuple$\langle \mathrm{ID}, Statue, URI, Hash, C, Signature_D \rangle$, where the following hold:

(i) ID: the identity of the domain manager

(ii) Statue: the statue of change, such as access recording or access, has been successful

(iii) *URI*: the storage location in DTDB

(iv) Hash: the hash value of DTDB

(v) *C*: the blank bits containing context-related information

(vi) $Signature_D : f \longrightarrow \mathrm{ID} \times \mathrm{Statue} \times URI \times \mathrm{Hash} \times C$ is a one-way hash function used as the signature of the domain manager

The trajectory modification behavior corresponds to the domain manager identity by *ID*. The $Signature_D$ also has the same effect. *Statue* is the statue of change, which is part of the goal-directed logging, to facilitate forensic analysis. *URI* indicates the data storage location in the DTDB. *Hash* is to confirm the integrity of DTDB, and it is also the main content of the smart contract. *C* can be used to record the context information such as time information.

4. Trajectory Aggregation and Forensic Analysis

In this section, we will introduce the multidomain trajectory aggregation method and the forensic analysis method in CDDAC.

4.1. Trajectory Aggregation. The delegation trajectory is written into the token by the delegator. Different branch nodes on the delegation chain do not have a comprehensive understanding of the delegation trajectory; therefore, the domain manager needs to extract and aggregate the delegation trajectory. We will introduce the workflow of delegation trajectory aggregation and goal-directed logging.

4.1.1. Delegation Trajectory Aggregation. As shown in Figure 5, *Users A–D* are delegated capability in turn. We find that the delegate trajectories from *Users A*, *B*, *C*, and *D* are not the same. They do not entirely contain each other. When *User B* delegates the capability to *User D*, he does not know that *User A* has delegated the capability to *User C*. The inherent information gap in the delegation process leads to the incompleteness of the delegation trajectory in a single capability token. Therefore, the domain manager needs to collect all the capability tokens, extract the topology of delegation trajectories, and aggregate them to obtain the complete delegation trajectory.

The delegation trajectory may cross multiple domains, and the domain managers may not know the identity of each user. Cross-domain access control also faces the challenge of devices shared by multiple domains, or the free devices. These make it difficult to obtain a complete cross-domain delegation trajectory. Therefore, the trajectories submitted by different domain managers should be aggregated twice for cross-domain delegation.

4.1.2. Goal-Directed Logging. Goal-directed logging mainly provides the ability to quickly respond to security incidents. When a security incident occurs, people usually hope to find vulnerabilities from the messy resource server logs. Additional log analysis [21] will extend the existence time of access control vulnerabilities, causing more serious damage to the system. In the meantime, relying on the access log that records the occurrence of access, the system has almost no dangerous warning functions. The dangerous behavior has occurred at least once when it is recorded. The importance of the goal-directed logging is thus reflected.

We propose a goal-directed logging for CDDAC. The delegation trajectory of the cross-domain delegation is saved in the DTDB. Figure 5 shows the workflow of delegation trajectory aggregation. The node structure of delegation trajectory in DTDB is [*Cap*, *User*, *EffectTime*, *AccessTime*]. Among them, [*Cap*] records the description of the capability. [*User*] records the user ID who has used this capability.

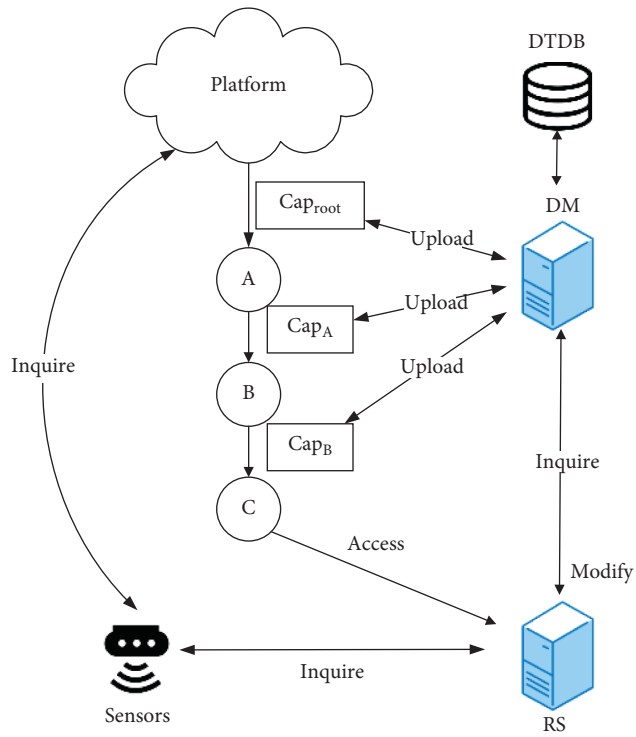


Figure 3: Delegated access control method in a single-domain system.

[*EffectTime*] records the validity period of the capability. [*AccessTime*] records the time when the user uses the capability. As each cross-domain access will be captured by the domain manager, all information needed can be obtained from the data packet. We use this method to achieve goal-directed logging of CDDAC.

4.2. Forensic Analysis. Due to the uncertainty of the capability flow, the cross-domain delegation systems are more likely to suffer security incidents caused by misallocation of capabilities. Since the capability delegation is determined by users, we cannot determine whether the delegation conforms to the principle of least privilege. It is also difficult to determine whether the capability delegation is under attack. The resource server can only detect the process legitimacy and passively provide capabilities recorded in the token. Therefore, the occurrence of security incidents is difficult to predict. Forensics analysis is particularly important to deal with such problems. CDDAC provides forensic analysis by establishing DTDB. In this section, we present the definition and examples of forensic analysis in CDDAC, including change warning and record analysis.

4.2.1. Definition. A stable access control scheme can be defined as a pair$\langle \Gamma, \Psi \rangle$, where $\Gamma$is a set of states and Ψ is the state-changing policies. Then, we can describe a forensic instance as follows:

$$F = < \gamma, \psi, p, q, \pi, L >, \quad (2)$$

where $\langle \gamma, \psi \rangle$ is a specific access control system, $\gamma \in \Gamma$ is the system state and $\psi \in \Psi$ is the rule caused the state change; p is the known past state of the system; q is a query by which we can get the information of the past state; π is the result of a forensic analysis; and L is the system logs.

In fact, we can get the past system state from p, $p \longrightarrow_{\psi}^{*} \gamma$. Then, we can establish a state-changing sequence $\gamma_1 \longrightarrow_{\psi} \gamma_2 \longrightarrow \ldots \longrightarrow_{\psi}^{*} \gamma$ as the evidence chain or the backtracking of errors. We use DTDB to catch the delegation behaviors as p and save the access logs as L in CDDAC. The DTDB supports multiple query methods q, constructs the capability trajectory, and finally gives a forensic analysis result π to users.

4.2.2. Change Warning. In a stable access control system, there will be fewer transfers, revocations, and modifications to sensitive capabilities. In other words, any changes to sensitive capabilities should be considered as threats, and administrator should be warned. The delegation system transfers the right of delegation capabilities to users, and the occasional misjudgment of the node brings a number of risks to the system. We believe that we should pay more attention to the change of sensitive capabilities, while ensuring the scalability of the system. Therefore, we reserve the user's delegation right for all capabilities, and the verification right of sensitive capabilities is given to the domain administrator. The specific approach is as follows. First, we divide the set of sensitive capabilities. For the sensitive delegation trajectory that we want to closely monitor, when the number of nodes increases (only the increase of nodes will bring a threat to a stable system), an early warning will be issued to

administrator, and the legitimacy of the completed delegation trajectory will be checked. If it does not meet the access control policies, we can revoke the capability at the first time. We use this method to achieve change warning in CDDAC.

4.2.3. Suspect Analysis. The suspicion analysis is mainly for the policy allocation errors. Thanks to the change warning mechanism, we have eliminated the misallocation of capabilities due to user's decision-making errors. However, the following situation is still possible: a "security" capability actually has the danger to cause a security incident, which is called the capability configuration error. Capability configuration errors often occur in the actual operation of system [22,23]. We cannot completely avoid the occurrence of security incidents. What is important is how to accurately and quickly conduct suspicion analysis.

In CDDAC, we use the *AccessTime* stored in the delegation trajectory for suspicion exclusion/conviction. *AccessTime* plays a role in goal-directed logging when security incidents occur. Goal-directed logging only records things that may be useful for forensic analysis. It has been expected to greatly reduce the size of log records that require forensic query [24]. In DTDB, the *AccessTime* stored by each node records the operation type and access time, which are the core information required for forensic analysis. *AccessTime* is automatically generated in delegated trajectory aggregation, which avoids additional data reduction work. We can easily record the location and time of a security incident and directly query the *Cap* and *AccessTime* stored in the DTDB. Then, we can evaluate whether the node may have the capability, whether it does possess the capability, and whether the capability is used. We can exclude users from suspicion, or convict users, in need of different situations.

5. Evaluation

We conduct experiments on the performance of CDDAC and compare it with existing works.

5.1. Implementation. Domain managers are two laptops with the following configurations: the CPU is 1.6 GHz Intel Core i5 (4 cores), the RAM is 8 GB, and the operating system is Ubuntu 16.04. Each laptop manages 2 to 8 Raspberry PI 4 Model B as IoT devices. Redis 5.0.8 is used to manage the delegation trajectory data. Ropsten is used as the blockchain on the public network, which is the Ethereum's official public test network [25]. We use it to evaluate the performance of our approach.

5.2. Performance

5.2.1. Token Processing Overhead. To evaluate the token efficiency of CDDAC, we randomly generate capability tokens with different delegation depths, to evaluate the token processing overhead of existing works. We define the token processing overhead as the time cost of obtaining the required information from the capability token. Because of the different token structures and processing flows of each scheme, the composition of the token processing overhead is different. In CapBAC, the token processing overhead is the time, which can be expressed as $\sum_{n=1}^{N}(T_n^{\text{Decryption}} + T_{\text{n}}^{\text{Verification}})$, where n is the number of token layers. In BlendCAC, the token processing overhead mainly includes the time for querying capability data from the smart contract and the time for parsing JSON data from the request. In CDDAC, the total processing time is composed of the decryption time, the verification time, the smart contract running time, and DTDB querying time. Token efficiency, which is the token processing overhead, represents the minimum resource consumption when the capability is used. High-efficiency tokens are accompanied by lower resource overhead, which is more suitable for lightweight IoT devices. It can be known from the calculation method that the token processing overhead is associated with the token layers and the number of nodes. Although we randomly generated the delegation trajectories, there is only one node at each layer of the capability token. We use this method to control irrelevant variables. RSA1024 is used for encryption and signature of all the three schemes. The experimental result is shown in Figure 6.

We compare the token processing overhead of CDDAC with CapBAC [19] and BlendCAC [14]. CDDAC uses the proposed single-layer capability tokens, CapBAC uses classic nested capability tokens, and BlendCAC uses capability tokens based on smart contract. The size of token increases with the number of layers in the three approaches, but the token processing procedure is different, which makes the difference of token processing overhead. Due to its layer-by-layer decryption design, the processing time of CapBAC increases greatly with the delegation depth, which makes it difficult to be used in large-scale systems. BlendCAC adopts the right-on-blockchain strategy. The key value of the token is extracted by the smart contract, so the token processing speed has little to do with the depth of delegation, which is about 200 ms. CDDAC uses the single-layer capability token, which prevents the excessive token complexity. The single verification greatly reduces the token processing overhead, so the processing time is always kept to a minimum. It shows that CDDAC's token design is more suitable for lightweight IoT devices.

5.2.2. Smart Contract Decision-Making Overhead. The decision-making overhead of a strategy is defined as the smart contract running time, and its changing trend is the manifestation of the scheme scalability. The decision-making cost of a scalable scheme should not increase drastically as the number of access control policies increases. We correspond policy entry to delegated behavior and evaluate the smart contract running time of the policy-on-blockchain, the right-on-blockchain, and the trajectory-on-blockchain strategies (which is used by CDDAC). The experimental result is shown in Figure 7.

It can be seen that the decision-making overhead of existing blockchain-based approaches increases as the smart contract complexity increases. Since the policy-on-blockchain

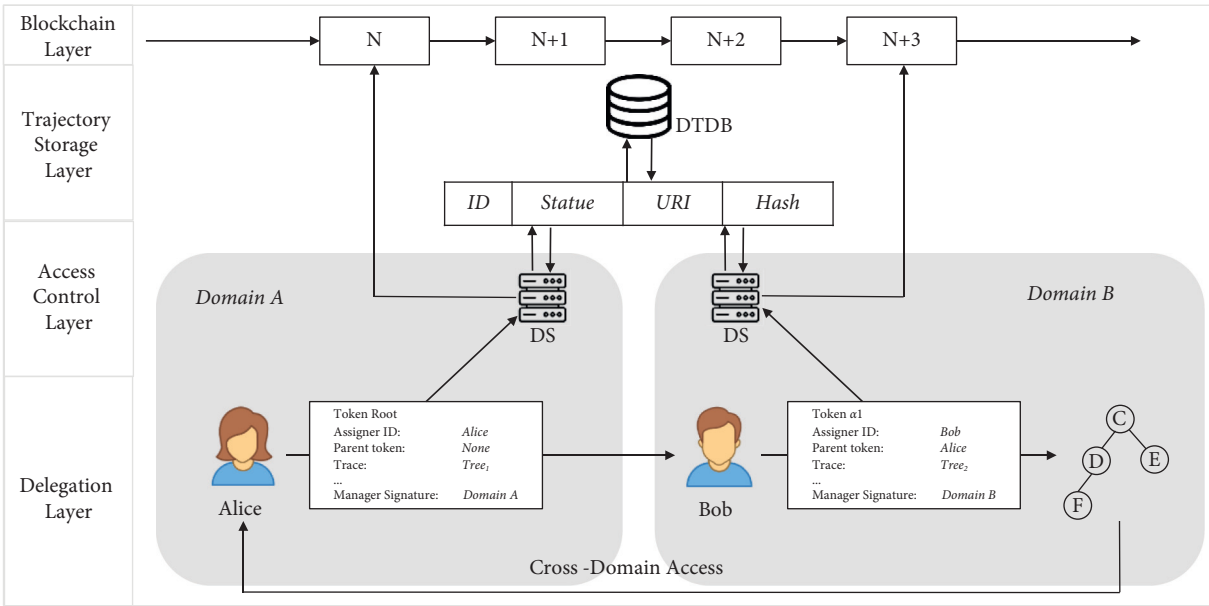


Figure 4: Delegated access control method in cross-domain system.

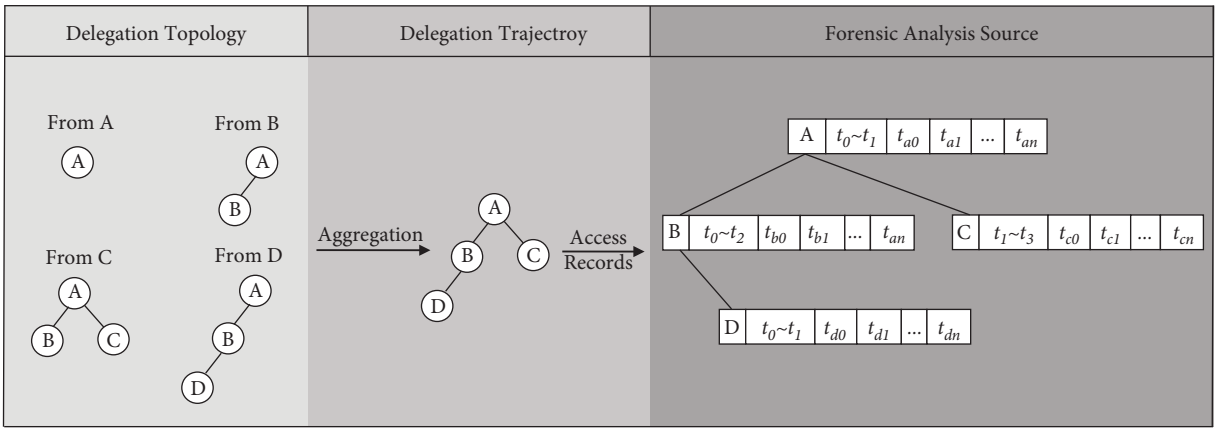


Figure 5: Workflow of delegation trajectory aggregation.

strategy writes access control policies by the form of smart contract, the complexity of smart contract will increase with the scale of policies, and its decision-making time will increase linearly with the number of policies, with the largest slope. The right-on-blockchain strategy extracts token attributes and checks whether the attributes exist in the smart contract, to make an access control decision. The decision-making time also expands with the expansion of the token scale, but the slope is smaller than the strategic plan. In trajectory-on-blockchain strategy, after obtaining the token, an inquiry will be initiated to DTDB, then decision will be made after verifying the authenticity. This procedure has little relevance to the number of policies, which brings an excellent scalability to CDDAC.

5.2.3. Policy Changing Overhead. Policy changing is an essential feature in an IoT access control system, which is always completed as a transaction in blockchain-based schemes. It is defined as the transaction time. We evaluate the policy changing overhead of three strategies. The result is shown in Figure 8.

The policy changing of the three strategies all involve the changing of smart contract. It will cost a long time to redeploy the smart contract, which brings the large cost of policy changing. The average cost is about 12 seconds. In policy-on-blockchain strategy, smart contracts need to be rewritten and deployed once a policy is changed, which enhance the policy changing complexity. The right-on-blockchain strategy modifies the token parameter set in the smart contract, to change access control policies. There is no need to rewrite the smart contract logic, so the policy is easier to change. The trajectory-on-blockchain strategy changes the content of DTDB, generates a new hash, and publishes it to a new smart contract. It has the lowest policy changing complexity, but it is still limited by the redeployment overhead of smart contract. When the

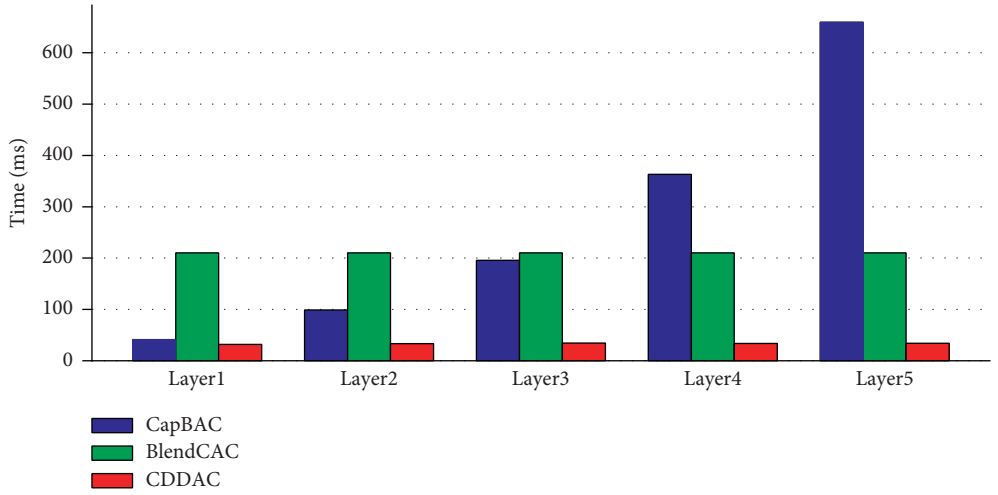


Figure 6: The token processing overhead of the three approaches under different capability token layers.

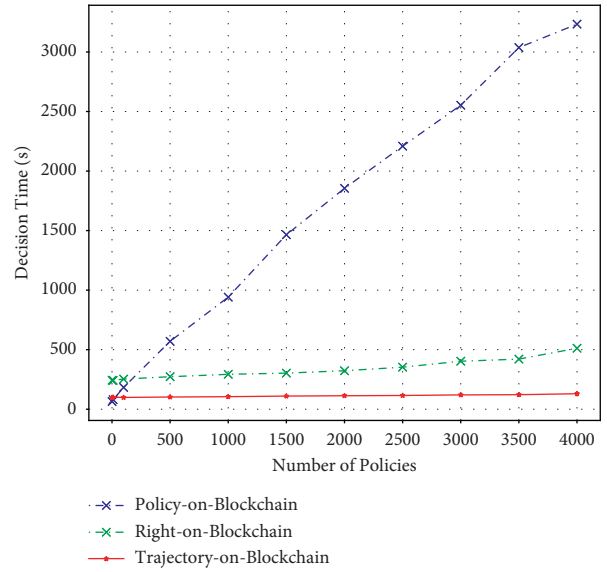


Figure 7: The smart contract decision-making overhead of the three strategies under different number of access control policies.

smart contract redeployment overhead is shortened, the policy changing overhead of CDDAC will gain a clear advantage.

5.3. Discussion. Benefiting from the design of extracting delegation trajectory, the complexity of CDDAC's capability token has been simplified, which greatly reduces the token processing overhead. The design of policy-on-blockchain and right-on-blockchain has been cancelled, which reduces the complexity of smart contracts, reduces the cost of decision-making, and ensures the high scalability of CDDAC. In the meantime, the policy changing process of CDDAC has been simplified. Although the test on the public blockchain network does not show obvious advantages, when the redeployment time of smart contracts is shortened, its efficiency will be reflected. The test of CDDAC has proved its significant progress in token processing overhead and smart contract decision-making overhead. CDDAC can maintain a token processing speed about 30 ms and a decision-making speed of about 110 ms. In short, as a delegation-oriented IoT cross-domain access control system, the lightweight and scalability of CDDAC show obvious advantages compared with existing works.

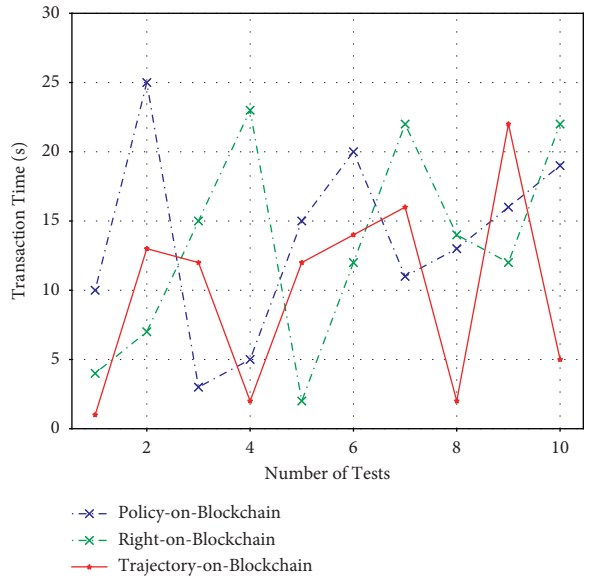


Figure 8: The policy changing overhead of the three strategies.

6. Conclusion

In this paper, we introduce CDDAC, a blockchain-based IoT cross-domain delegation access control method. The capability token structure of CDDAC is more suitable for lightweight devices. The adopted trajectory-on-blockchain strategy greatly enhances the scalability of the system and has a simpler policy changing process. We propose a multidomain delegation trajectory aggregation mechanism to support forensic analysis of intra/cross-domain delegation, which is beneficial to the confirmation and accountability of suspicion after malicious behavior occurs. We evaluate the performance of CDDAC in the Ropsten. The results show that CDDAC has the advantages of lightweight and scalability compared with similar research. In the future, we will design a privacy protection mechanism in the process of cross-domain delegation based on the idea of CDDAC and combine CDDAC with the alliance chain to achieve a more complete cross-domain delegation access control system.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported in part by the National Key R&D Program of China (no. 2018YFB2100400), National Science Foundation of China (no. 61872100), Industrial Internet Innovation and Development Project of China (2019), State Grid Corporation of China Co., Ltd., Technology Project (no. 5700–202019187A-0-0-00), and Guangxi Key Laboratory of Cryptography and Information Security (no. GXIS202119).

References

[1] S. B. Baker, W. Xiang, and I. Atkinson, "Internet of things for smart healthcare: technologies, challenges, and opportunities," *IEEE Access*, vol. 5, pp. 26521–26544, 2017.

[2] W. Tärneberg, V. Chandrasekaran, and M. Humphrey, "Experiences creating a framework for smart traffic control using aws iot," in *Proceedings of the 2016 IEEE/ACM 9th International Conference on Utility and Cloud Computing (UCC)*, pp. 63–69, Shanghai, China, December 2019.

[3] S. Zheng, N. Apthorpe, M. Chetty, and N. Feamster, "User perceptions of smart home IoT privacy," *Proceedings of the ACM on Human-Computer Interaction*, vol. 2, pp. 1–20, 2018.

[4] T. H. Payne, D. E. Detmer, J. C. Wyatt, and I. E. Buchan, "National-scale clinical information exchange in the United Kingdom: lessons for the United States," *Journal of the American Medical Informatics Association*, vol. 18, no. 1, pp. 91–98, 2011.

[5] J. Benaloh, M. Chase, E. Horvitz, and K. Lauter, "Patient controlled encryption: ensuring privacy of electronic medical records," in *Proceedings of the 2009 ACM Workshop on Cloud Computing Security*, pp. 103–114, Redmond, WA, USA, November 2009.

[6] Q. Alam, M. Alani, G. Ali, and F. Azim, "Towards a formal framework for cross domain access control," *International Information Institute (Tokyo). Information*, vol. 15, no. 10, p. 4303, 2012.

[7] J. Sun J and Y. Fang, "Cross-domain data sharing in distributed electronic health record systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 6, pp. 754–764, 2009.

[8] M. Alam, X. Zhang, K. Khan, and G. Ali, "XDAuth: a scalable and lightweight framework for cross domain access control and delegation," in *Proceedings of the 16th ACM symposium*

on Access control models and technologies, pp. 31–40, Innsbruck, Austria, June 2011.

[9] S. Sheikh and A. K. Chaturvedi, "Analysis of sensitive data security on trusted third party in cloud computing," *management*, vol. 17, p. 18, 2014.

[10] Q. Alam, S. Tabbasum, A. Malik, and M. Alam, "Formal verification of the xDAuth protocol," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 9, pp. 1956–1969, 2016.

[11] O. Novo, "Blockchain meets IoT: an architecture for scalable access management in IoT," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 1184–1195, 2018.

[12] A. Ouaddah, A. Abou Elkalam, and A. Ait Ouahman, "FairAccess: a new blockchain-based access control framework for the internet of things," *Security and Communication Networks*, vol. 9, no. 18, pp. 5943–5964, 2016.

[13] B. Tang, H. Kang, J. Fan, Q. Li, and R. Sandhu, "Iot passport: a blockchain-based trust framework for collaborative internet-of-things," in *Proceedings of the 24th ACM symposium on access control models and technologies*, pp. 83–92, Toronto, Canada, May 2019.

[14] R. Xu, Y. Chen, E. Blasch, and G. Chen, "Blendcac: a blockchain-enabled decentralized capability-based access control for iots," in *Proceedings of the 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pp. 1027–1034, Halifax, Canada, August 2018.

[15] Y. Nakamura, Y. Zhang, M. Sasabe, and S. Kasahara, "Exploiting smart contracts for capability-based access control in the Internet of Things," *Sensors*, vol. 20, no. 6, p. 1793, 2020.

[16] A. Gauhar, N. Ahmad, Y. Cao et al., "xDBAuth: blockchain based cross domain authentication and authorization framework for internet of things," *IEEE Access*, vol. 8, pp. 58800–58816, 2020.

[17] B. Yuan, Y. Jia, L. Xing, D. Zhao, X. Wang, and Y. Zhang, "Shattered chain of trust: understanding security risks in cross-cloud iot access delegation," in *Proceedings of the 29th ${$USENIX$}$ Security Symposium (${$USENIX$}$ Security 20)*, pp. 1183–1200, Boston, MA, USA, August 2020.

[18] D. D. F. Maesa, P. Mori, and L. Ricci, "Blockchain based access control," in *Proceedings of the IFIP international conference on distributed applications and interoperable systems*, pp. 206–220, Neuchâtel, Switzerland, June 2017.

[19] S. Gusmeroli, S. Piccione, and D. Rotondi, "A capability-based security approach to manage access control in the internet of things," *Mathematical and Computer Modelling*, vol. 58, no. 5–6, pp. 1189–1205, 2013.

[20] J. Chen, W. Gan, M. Hu et al.,et al. "On the construction of a post-quantum blockchain for smart city," *Journal of Information Security and Applications*, vol. 58, Article ID 102780, 2021.

[21] B. J. Jansen, ""Search log analysis: what it is, what's been done, how to do it," *Library & information science research*, vol. 28, no. 3, pp. 407–432, 2006.

[22] T. Das, R. Bhagwan, and P. Naldurg, "Baaz: a system for detecting access control misconfigurations," *USENIX Security Symposium*, vol. 17, pp. 161–176, 2010.

[23] T. Xu, H. M. Naing, L. Lu, and Y. Zhou, "How do system administrators resolve access-denied issues in the real world?" in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pp. 348–361, Denver Colorado USA, May 2017.

[24] N. Juma, X. Huang, and M. Tripunitara, "Forensic analysis in access control: foundations and a case-study from practice," ", in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1533–1550, Virtual Event USA, November 2020.

[25] Ethereum/ropsten. ethereum, 2021, https://github.com/ethereum/ropsten.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 7249470, 11 pages
https://doi.org/10.1155/2021/7249470

Research Article

A Collusion-Resistant Blockchain-Enabled Data Sharing Scheme with Decryption Outsourcing under Time Restriction

Xieyang Shen,[1] **Chuanhe Huang**,[1] **Xiajiong Shen**,[2] **Jiaoli Shi**,[3] **and Danxin Wang**[1]

[1]*School of Computer Science, Wuhan University, Wuhan, China*
[2]*Henan Key Laboratory of Big Data Analysis and Processing, Henan University, Kaifeng, China*
[3]*School of Information Science and Technology, Jiujiang University, Jiujiang, China*

Correspondence should be addressed to Chuanhe Huang; huangch@whu.edu.cn

Received 14 June 2021; Accepted 26 July 2021; Published 27 August 2021

Academic Editor: Yinghui Zhang

With the ever-increasing demands on decentralization and transparency of cloud storage, CP-ABE (Ciphertext Policy-Attribute-Based Encryption) has become a promising technology for blockchain-enabled data sharing methods due to its flexibility. However, real-world blockchain applications usually have some special requirements like time restrictions or power limitations. Thus, decryption outsourcing is widely used in data sharing scenarios and also causes concerns about data security. In this paper, we proposed a secure access control scheme based on CP-ABE, which could share contents during a particular time slot in blockchain-enabled data sharing systems. Specifically, we bind the time period with both ciphertexts and the keys to archive the goal of only users who have the required attributes in a particular time slot can decrypt the content. Besides, we use time slots as a token to protect the data and access control scheme when users want to outsource the decryption phase. The security analysis shows that our scheme can provide collusion resistance ability under a time restriction, and performance evaluations indicate that our scheme uses less time in decryption compared to other schemes while ensuring security.

1. Introduction

Traditional blockchain-enabled data sharing schemes usually assume that CSP (cloud service provider) can be trusted to keep data confidential. However, this assumption causes more concerns about the security and integrity of data since more and more end users tend to outsource the decryption phase to CSP due to their resource-constrained devices, for example, more and more smart devices with the duty of data storage and computation collecting private information under smart city scenarios [1]. To mitigate users' concerns about their data privacy and security, an access control scheme that can either prevent curious CSP from scanning data stored on the cloud or disclose nothing during the outsourcing decryption must be proposed [2].

Attribute-based encryption is considered by scholars as a novel solution for solving the problems stated above. ABE was first proposed by Sahai and Waters [3] and further developed two categories: Ciphertext-Policy ABE (CP-ABE) and Key-Policy ABE (KP-ABE) [4], depending on whether the access policies are embedded with the ciphertext or the user's private key. ABE can prevent both unauthorized users and curious servers from accessing the data and support data owners to encrypt their data before sending them to cloud servers. In CP-ABE, the access policy is binding with the ciphertext so that data owners do not need to update the ciphertext when attributes are changed. Thus, CP-ABE is more suitable for cloud access control environments and can be deployed in many scenarios.

Besides, time restriction is more and more common nowadays due to the sensitivity of the data in blockchain-enabled data sharing systems, such as video content [5] and personal health record [6]. The fine-grained access control has been paid much more attention in attribute-based encryption schemes, but it is still not easy to get the goal of adding time restrictions in these schemes. Furthermore, the semitrusted cloud server providers make these issues more serious as the providers themselves are curious about the

content stored in the cloud. From the time restriction aspect, for example, a malicious cloud service provider can easily predict the policy update or attribute revoke operation of a company (as a data owner), then the provider can delay the updating operation for a few hours or even a few minutes to let the revoked users get the data illegally and shirk its responsibility to the high latency of the network. Compared with recent ABE schemes [7], our scheme set time as a part of the key to examine both the cloud servers and the users. In this case, the time restriction can be seen as an attribute of the user and an examined standard on a cloud server.

In this paper, we focus on designing a collusion resistance access control scheme based on ABE and ensuring the safety of data after decryption outsourcing. We propose a secure access control scheme based on CP-ABE under time restrictions. For the above goals, we bind the time slot with both ciphertexts and secret keys in a blockchain-enabled data sharing scheme so that only the legal user (which means satisfying the access policy and the time restriction at the same time) can decrypt the data. Besides, we use a time slot as a token to make sure that the outsourcing party cannot get any information from the calculation phase.

The main contributions are summarized as follows:

(1) We propose a blockchain-enabled data sharing scheme based on CP-ABE by binding the time slot with both ciphertexts and secret keys. In this way, users must meet any request between attributes and time slot to decrypt the data.

(2) We propose a method in the multiservers scenario to prevent a new kind of collusion that a malicious cloud server provider does not execute the owner's update/revoked order in time to gain some time for revoked users to get the data illegally.

(3) We propose a method to change the time slot in outsourcing into a token to guarantee that the calculation phase in the outsourcing party will not leak any information about the data and the access policy.

The rest of our paper is organized as follows.

Related work is introduced in Section 2. In Section 3, we first list some preliminaries and then proposed our system architecture. A detailed scheme is presented in Section 4. We also propose our collusion resistance updating method in this section. Security analysis and performance evaluations are conducted in Section 5. Conclusion and further discussion are in Section 6.

2. Related Work

Outsourcing is a common solution for power limited devices to complete the task they could not afford in blockchain-enabled data sharing schemes [8]. Despite the consideration of computation and storage, outsourcing services are also applied to many scenarios such as big data analysis [9], attack detections [10], machine learning [11]. CP-ABE [12] is regarded as one of the most practical models for access control schemes in blockchain-enabled data sharing, for it not only allows the data owner to define the access policy from several attribute authorities [13] but also does not need a trustworthy third party to realize decentralization and transparency requirements for blockchain [14]. DAC-MACS (Data Access Control for Multi-Authority Cloud Storage) designed by Yang et al. [5], is one of the multiauthority schemes which propose effective and secure data access control schemes for video content sharing. However, users are required to transfer their private keys to the cloud for generating a decryption token for efficiency. A series of constructions exist to realize fine-grained access control for data sharing with CP-ABE in different ways. Yang et al. focused on efficient revocation [15] and multiauthority [12], respectively. Shi et al. designed a version key mechanism for direct revocation [16]. Unfortunately, most of the above schemes did not take time into consideration.

Time is a quite unique factor in some scenarios like video content sharing [5], online storage service [17], and weather reporting [18]. It has become an important prominent factor, especially in blockchain-enabled data sharing that can even decide the worth of the data. However, it also raises concerns about data security and end devices affordability. With the proposal of sharing time-sensitive data in a particular time period, several ABE schemes have taken time into consideration. Liu et al. [19] proposed a time-based proxy reencryption scheme, so that in a particular time slot, the access policy can control the access for users. Conversely, with the change of time period, data owners need to reencrypt the ciphertext, which is not suitable for blockchain-enabled data sharing systems. Yang et al. [5] proposed a time domain multiauthority ABE method that binds time with ciphertext and secret key, but computation cost on both data owner and user increases linearly. Hong et al. [20] designed an access control scheme based on both time and attributes, where cloud servers play an important role, including generating a token and updating ciphertext online at each of the time periods. However, the time period of this scheme is defined at the beginning of the system initialization. Thus, the time period cannot fit most of the situations in the real world. As in our scheme, the time slot information is considered an essential problem to achieve the goal. Each data owner can define the time slot on their own demand. Furthermore, we take the blockchain-enabled data sharing environment into consideration and further combine the collusion resistance ability with our scheme.

On the other hand, disclosure of private keys increases the security risks of data such as PHRs (Personal Health Record) or even the information of the COVID-19 pandemic [21]. Liu et al.[6] established a patient-centric framework in the multiauthority model and used signcryption to guarantee data security. Besides, Li et al. concentrate on scalability in access control schemes. In their scheme, users in the PHR system were divided into various security domains and different policies would be published to different domains according to the definition of PHR owners. ABE as cryptographic primitives were applied and the rules of encryption and key-distributed were also based on those primitives. What's more, they use a hash chain to ensure forward security. However, the work in [6] just

applied to the PHR environment and may lose their varieties of other scenarios.

Revoking is also an important part of attribute-based encryption in blockchain-enabled data sharing systems, as authorities must keep the data consistency of each user. The first Hybrid Revocable ABE scheme is proposed by Attrapadung and Imai [22], which allowed data owners to choose how to revoke an attribute online: direct revocation or an indirect one. Thus, the scheme can take both advantages of direct and indirect revocation and avoid the disadvantages. Other schemes like [23], proposed by Sahai et al., solved the problem of attribute dynamic updating by proposing an attribute delegation method. Furthermore, they use a segmented secret key to ensure attributes are granted or revoke that even under a more restrictive access policy. But the backward security cannot be assured because the scheme needs to reencrypt the ciphertext so that when a new user comes to join the system, with the later time slot, he or she can still decrypt the data. Yang and Jia [13] try to solve the key escrow problem by putting forward a novel CP-ABE scheme in which a two-party computation protocol was executed between Key Generation Center (KGC) and Data Storage Center (DSC). In the above schemes, we could see that attribute revocation requests were mostly demanded by attribute authorities rather than users, or to say revoked users might not want to request for revocation for many reasons.

3. System Architecture

In this section, we first introduce the related preliminary knowledge, then present the system model of our scheme, and introduce the proposed access control scheme. At last, we give the security model. For convenience, some notations are summarized in Table 1.

3.1. Preliminaries

3.1.1. Bilinear Maps. There exist two multiplicative cyclic groups G and G_T with prime order p and generator g; e: $G \times G \longrightarrow G_T$ is a bilinear map if and only if the following three properties are satisfied:

(1) Bilinearity: if $\forall u, v \in G$ and $x, y \in Z_p$, then we have $e(u^x, v^y) = e(u, v)^{xy}$

(2) Nondegeneracy:$e(g, g) \neq 1$

(3) Computability: $\forall u, v \in G$, $e(u, v)$ is an admissible algorithm

3.1.2. Collusion Resistance. A collusion attack [24] in ABE means two or more entities (users, cloud servers, or even authorities) can successfully decrypt the data they can not decrypt individually after some operations like exchange their secret key or share information with each other. However, collusions using time restriction have not been mentioned before. For example, Alice has just been revoked by the data owner within the current time slot t_1 with the attribute which satisfied the access policy, while Bob is a curious cloud server provider that stores the ciphertext. It is obvious that both of them cannot decrypt the ciphertext individually. However, if they are working together with each other, Alice can send the data access request at the end of t_1 to Bob, while Bob needs to update the ciphertext at the beginning of the next time slot t_2. However, Bob can respond to the request of Alice using the ciphertext in t_1 while shifting the blame to network latency. As a result, Alice can get the decrypt the ciphertext in the time slot she should not have access right in. This kind of collusion can be easily implemented as it is easy to predict the attribute revoke time for Alice. Our scheme must have the ability to resist this kind of collusion attack in any circumstances.

Table 1: List of notations.

$\mathcal{T}$	Time slot
D	Attribute index
U_d	Attribute set authorized by AA_d
U	Universal attribute set, s
CT	Ciphertext
gid	Global identity
λ	Global parameters
S_{gid}	Set of user's attributes s
$S_{gid,t}$	Attribute set of global id gid at time t
$SK_{gid,x}$	Secret key of attribute x for user with gid
ω	A set of random number $\in Z_p$
DEK	Decryption key

3.1.3. Time Slot. A time slot is a particular time period defined by the data owner. The length of a time slot can be a day, an hour, or even a minute. However, it is not achievable for a cloud server to update every ciphertext with the newest time slot because the overhead of updates increases exponentially. So, it is a good choice for the cloud to update the ciphertext when they get the access request. On the other side, the data owner can define the attribute of a user across different time slots. For example, the time slot defined by the owner is an hour while the attribute of Alice is granted and revoked at 9 : 30 and 11 : 30, respectively. For a clear explanation, we call a time slot a decryptable time slot if and only if a user has the validity completely covered and the time slot can decrypt the ciphertext.

3.2. System Structure. We build our blockchain-enabled data sharing system with time restrictions as follows. As we cannot predict the changing trend of user's attributes, we divide time into time slots to separate the operation of attributes. We define time slot as $\mathcal{T} = \{n \mid n \in 1, 2, \ldots, N\}$. As shown in Figure 1, the system model is constituted by four types of entities: cloud server providers (servers in the cloud), attribute authorities (AAs), data owners (owners), and data users (users).

Cloud servers play the role of data storing and executing computation steps in policy updating, users/attributes revoking part. Normally, we consider that cloud servers are curious but honest, which means that cloud server providers will give their best to get the data stored in the cloud as a prerequisite of doing what data owners want correctly. As in

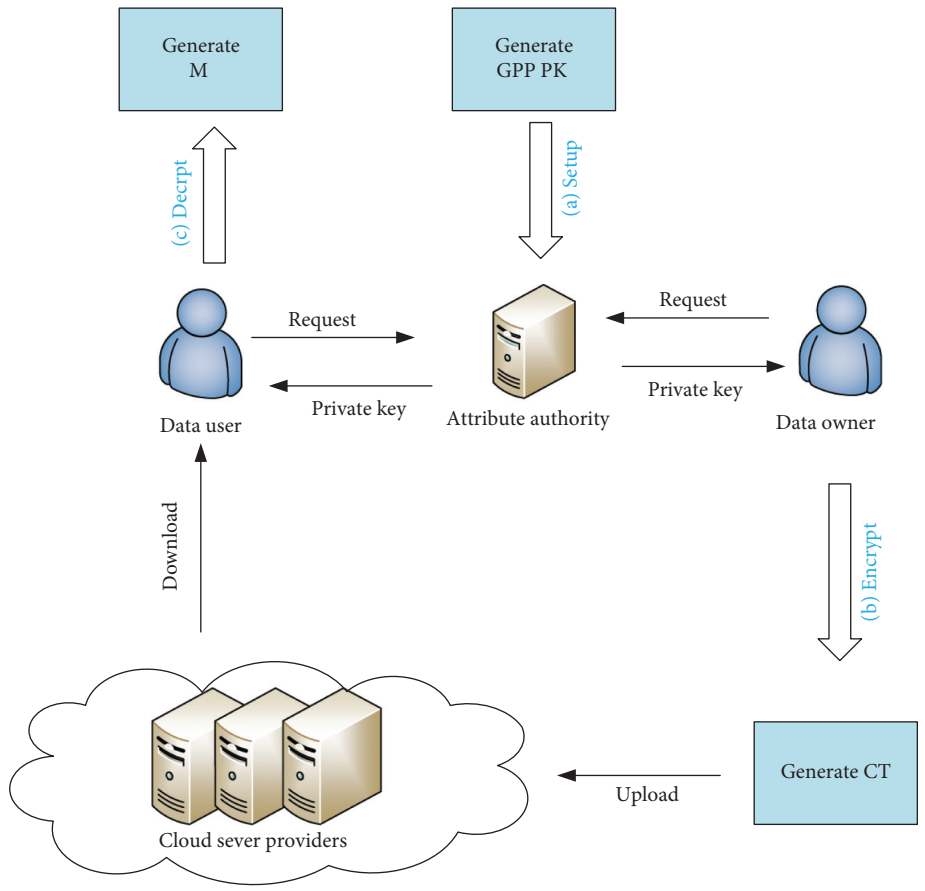


Figure 1: System model.

our model, we consider the situation that cloud servers may delay the update/revoke computation part for a short period of time to provide convenience for revoked users to gain data illegally. We derived this kind of situation into collusion between the cloud server and revoked users.

For attribute authority (AA for short), each of them is independent and responsible for granting or revoking attributes of users according to their roles or identities in their own domains. In our model, we consider that each attribute is only associated with a single AA (which may suit most of the situations in reality). However, each AA is in charge of a different number of attributes. That is to say, an attribute can only be authorized by one authority. We will identify the attributes by the index of the authorities below. For each authority, we use $\phi: U \longrightarrow D$ to map all attributes which belong to the AA to an identifier of the authority. AA can control the attributes or the structure in its domain. This kind of authorizing is required as the attributes changed periodically. As for time slots, it is not necessary to keep the length of every time slot the same for the reason that there may be different demands on time restrictions. In practice, this kind of requirement may largely reduce the computation cost on both authorities and data owners.

The data owner makes the definition of access policy before the data encryption on his side. Besides, he also makes a time span to set a time slot first. In the updating phase, the owner can update the slot or a new tree of time slot changing (usually not necessary). The encryption part on owners can be fast and light-weighted as owners only need to encrypt the data with the access policy designed by themselves. We define the ciphertext as CT_{A,t_e}. Besides, only those users who have the attributes satisfying the access control policy in the time slot t_e ($e \in \mathcal{T}$) can decrypt the data in CT_{A,t_e}.

In our system, when a user with gid gains a new attribute x, a new secret key $\mathrm{SK}_{\mathrm{gid},x}$ will be granted at the same time by corresponding AA. If the user wants to decrypt the data, he has to obtain the update keys first at this time slot (i.e., $\mathrm{UK}_{x,t}$) from the authority who can publish the attribute. After that, the user can compute decryption keys for a time slot t based on his secret keys and further uses them to decrypt the ciphertext. In this case, we can guarantee that users can only get the data in a single time slot because those users who do not update their attributes with the time slot in the past cannot satisfy any access control policies in our system. Considering the time consistency, we use the time slots we mentioned above to ensure that all entities in our model can check their current time with time slot at any second they want.

3.3. Security Model. In our scheme, we take these points into consideration: (1) The cloud server is curious about the ciphertext stored in the cloud, and they will try their best to decrypt them. (2) Cloud servers may send the data (in the

form of ciphertext) to unauthorized users. (3) Users and cloud servers may collude with each other. The security model is run between a challenger and an adversary $\mathcal{A}$, which is defined by the following game with two phases.

Setup. (1) The challenger first runs *GlobalSetup* algorithm and opens the access of GPP to the adversary. (2) $\mathcal{A}$ randomly select several AA*s* to play the role of corrupted AA and ask them to send their public key PK_d to $\mathcal{A}$.

Phase 1. $\mathcal{A}$ can request secret keys and update keys only by repeating the following steps:

(1) SK Query (gid, x): $\mathcal{A}$ sends a secret key request to those uncorrupted authorities by submitting a tuple (gid, x) where gid is the unique global identifier of a user and x is an attribute which is authorized by one of the uncorrupted authority. After receiving the queries, the challenger runs **SKeyGen** algorithm to return the corresponding secret keys $\mathrm{SK}_{\mathrm{gid},x}$ to $\mathcal{A}$.

(2) UKQuery (t, x): at the beginning of a time slot, $\mathcal{A}$ can ask those uncorrupted authorities to update their attributes or time slot update (if needed) and submit the pair of (t, x) to be updated by authorities. The challenger returns an update key $\mathrm{UK}_{x,t}$ to $\mathcal{A}$.

Challenge Phase. $\mathcal{A}$ submits two equal length messages M_0 and M_1, an access policy A^* (all attributes in A^* belongs to U), a time slot $t^* \in \mathcal{T}$ to the challenger. After this, the adversary should give the public key PK_d of all corrupted authorities whose attributes appear to the challenger. Then, the challenger flips a coin $\beta \in \{0, 1\}$ and sends to $\mathcal{A}$ the encrypted M_β using (A^*, t^*).

Phase 2. $\mathcal{A}$ can make as many queries as he wants according to Phase 1.

Guess. $\mathcal{A}$ submits a guess β' for β. The adversary $\mathcal{A}$ will win the game if $\beta\prime = \beta$ and satisfies the following demand.

(1) UKQ $(t, *)$ can only be queried on time slot after the time of all requests above, which means that the past time slot period of time in the system cannot be traced. Also, for any pair (t, x), UKQ phase can be executed only once because the corresponding authority will not publish the update key after the beginning of the time slot, which means the initialize phase in each slot is run by AA executes only once.

(2) For any queried gid, $S_{\mathrm{gid},t}$ ($S_{\mathrm{gid},t}$ stands for the set of gid and t) does not satisfy A^*. The advantage of $\mathcal{A}$ is defined as $|\Pr[\beta = \beta'] - 1/2|$.

4. Time Slot Access Control Scheme

In this section, we will explain our scheme step by step and list some algorithms if needed. Based on the algorithms defined in Section 3. Our scheme contains the four main phases: System Initialization which runs at the beginning of the whole system, Key Generation ran by each AA, Data Encryption phase for data owners to encrypt data with defined access policy, and Data Decryption ran by Users and computation outsourcing party. The workflow of our scheme is listed in Figure 2.

4.1. System Overview. Phase 1: system initialization: The system initialization phase is run at the beginning of the system and has two steps: Global Setup and Authority Setup.

(1) Global Setup:

$$\mathrm{GlobalSetup}(\lambda) \longrightarrow \mathrm{GPP}$$

The input of the global setup phase is the security parameters and the output is the Global public parameters GPP which will be used in other phases later. Set G and $G_{\mathcal{T}}$ as a bilinear group of prime order p with the bilinear group G which has the generator g. The global public parameter GPP used for key generation is published as $\mathrm{GPP} = (e, H, g, p)$; here e is a bilinear paring, and H is a hash function that maps every gid to elements of the group G.

(2) Authority Setup:

$$\mathrm{AuthoritySetup}(\mathrm{GPP}, U_d) \longrightarrow (\mathrm{PK}_d, \mathrm{MSK}_d)$$

Each AA must run a setup algorithm before publishing authorities. It takes the inputs as the global public parameters GPP , which outputs in *GlobalSetup* phase, the attribute domain U_d of the authority itself. The output of this phase is the master secret key MSK_d which is used for the authority itself and the public key PK_d , which sends to users. For any attribute x belongs to the attribute universe, the algorithm chooses the random exponents $\alpha_x, \beta_x, \gamma_x \in Z_p$. Besides, the algorithm also chooses a random element for the pseudorandom function F as the seed to generate the function. For each attribute that can be published by authorities, it chooses a random number $R = F(\tau_x, a) \in G$ (a_x denotes the attributes set of the authorities) and uses it to generate a secret key for the user. Here, we denote $U_d \times \mathcal{T} \longrightarrow G$ to be a hash function that maps both the attributes of the authorities and time slots in $U_d \times \mathcal{T}$ to elements of G. Then, the public key can be generated as $\mathrm{PK}_d = (\{e(g, g)^{\alpha_x}, g^{\beta_x}, g^{(1/\gamma_x)}\}_{x \in U_d}, H_d)$, where $\alpha_x, \beta_x, \gamma_x$ are combined to build the master secret key which is only kept by the authorities themselves.

So after the initialization phase, we get the global parameters GPP, public keys PK_d , and secret keys MSK_d generated by every authority. Each authority will further use these keys to generate a secret key for those users in their attribute domain.

Phase 2: key generation by AA.

$$\mathrm{SKeyGen}(\mathrm{gid}, x, \mathrm{GPP}, \mathrm{MSK}_{\phi(x)}) \longrightarrow (\mathrm{SK}_{\mathrm{gid},x})$$

Every AA runs the key generation algorithm for users in its domain. Each AA takes the global public parameters GPP, the master secret key $\mathrm{MSK}_{\phi(x)}$ generated on the last phase along with user's global identity gid as input and outputs the secret key for corresponding user as $\mathrm{SK}_{\mathrm{gid},x}$. The key generation algorithm is run by AA, and outputs the secret key $\mathrm{SK}_{\mathrm{gid},x}$, which associates with users' global identity and the corresponding attribute. The algorithm has two steps:

(1) Sets $u_{x,\mathrm{gid}} = 2^{h_x} + \mathrm{count}_x$ and adds the pair $(\mathrm{gid}, u_{x,\mathrm{gid}})$ to a List_x (a list of attribute trees for the

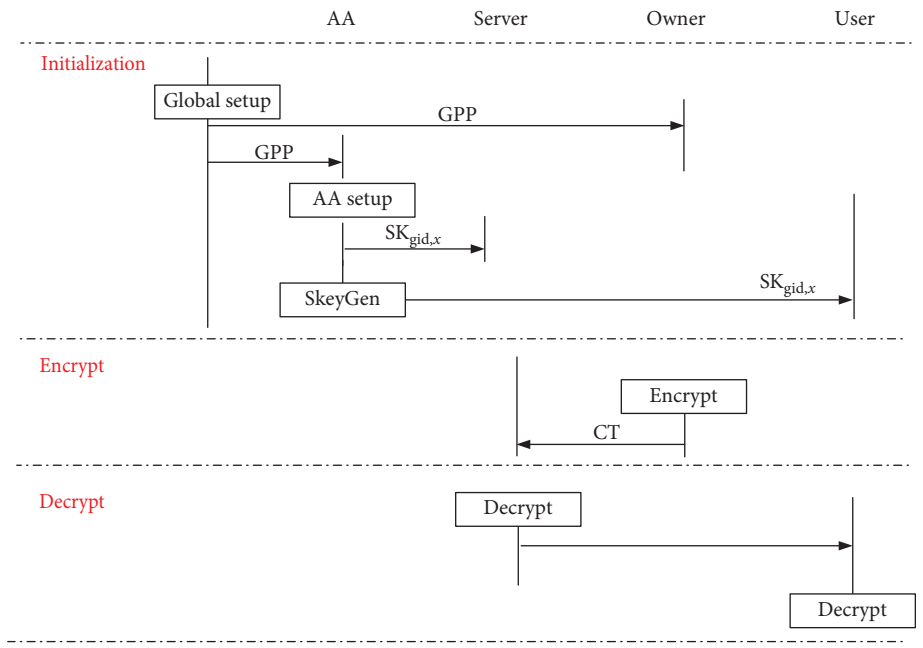


Figure 2: Workflow.

user x with the time restriction) and sets $\text{count}_x = \text{count}_x + 1$.

(2) The authority $\text{AA}_{\phi(x)}$ sends the secret key $\text{SK}_{\text{gid},x}$ to the user. We have to emphasize that every secret key for the attribute is only generated after the attribute is established for the sake of privacy preserving for user information.

Phase 3: data encryption:

$\text{Encrypt}(M, t_e, A, \text{GPP}, \{\text{PK}_d\}) \longrightarrow \text{CT}$

The encryption phase includes the input part of the Message M, access control policy A which is based on attributes may from several authorities, the time slot t_e, the global public parameters GPP, and the public keys $\{\text{PK}_d\}$. The output of this phase is the ciphertext CT which contains the access policy A.

Considering the difference in file type and scale, owners first encrypt their files by using a symmetric encryption algorithm and use our data encryption method to encrypt the secret key of the symmetric encryption. In this way, we could shorten the encrypt message length to reduce the computation cost on the owner side. In this Phase, M consists of two parts: one is the symmetric encryption key K and the other is the ciphertext encrypted by K. A is the access control policy and $\{\text{PK}_d\}$ are the set of public keys generated by the authorities; these public keys are related to the access policy. The data owner chooses a random number $r \in Z_p$ as the secret and keeps it without anyone else knowing it. After this, the owner chooses another two random vectors $\vec{v}$, $\vec{u}$ $\in\in Z_p$, and for each index $i \in \{1, 2, \ldots, m\}$, it selects another random oracle $r' \in Z_p$. So, the ciphertext is computed as follows:

$$\begin{aligned}\text{CT} = \Big\{\Big\{t_e, (A,\rho), C = M \cdot e(g,g)^e, C_{i,1} = e(g,g)^{\lambda_i} e(g,g)^{\alpha_{\rho(i)} r_i},\\ C_{i,2} = g^{\mu_i} g^{\beta_{\rho(i)} r_i}, C_{i,3} = g^{r_i}, C'_{i,3} = \left(g^{(1/r_x)}\right)^{r_i}, C_{i,4} = H_{\phi(\rho(i))}(\rho(i), t_e)^{r_i}\Big\}_{i=1}^{m}\Big\}.\end{aligned} \tag{1}$$

Then, the owner stores the ciphertext in cloud servers for later data sharing.

Phase 4: data decryption by users:

$\text{Decrypt}(\text{CT}, \text{GPP}, \{\text{PK}_d\}, \{\text{DK}_{\text{gid},x,t}\}_{x \in S \in \text{gid},t}) \longrightarrow (M) \text{or} \perp$

The decryption algorithm can run by users or outsourcing party which depends on user's choices. When users want to decrypt by themselves, the input includes the ciphertext CT along with the access policy A, the global public parameters GPP, the public keys PK_D and decryption keys as $\{\text{DK}_{\text{gid},x,t}\}_{x \in S \in \text{gid},t}$ which is related to the user's gid and current time slot. The algorithm outputs the plaintext M when the decryption succeeds or a token $\perp$ implying decryption fails for some reason. All users with proper attributes can download the ciphertext in which their attributes meet the demand of the access policy at the specific time slot t_e. The decryption phase must compute the decryption key first and then decrypt the ciphertext.

4.1.1. Decryption Key Computation. At the beginning of the decryption, the user with gid first needs to compute the decryption key for the current time slot as follows: $\mathrm{DecryptKey}(\mathrm{SK}_{\mathrm{gid},x}, t_e) \longrightarrow (\mathrm{DK}_{\mathrm{gid},x,t})$ or $\perp$. In this phase, the user first checks the attribute set and whether the corresponding attribute x as $x \in S_{\mathrm{gid},t}$ exists. If so, the decryption key can be computed as follows:

$$\mathrm{DK}_{\mathrm{gid},x,t} = \left(D_{\mathrm{gid},x,t} = K_{\mathrm{gid},x,v_x}, D'_{\mathrm{gid},x,t} = \left(E_{v_x}\right)^{K'_{\mathrm{gid},x,v_x}} \cdot K''_{\mathrm{gid},x,v_x}, D''_{\mathrm{gid},x,t} = \left(E'_{v_x}\right)^{K'_{\mathrm{gid},x,v_x}} \right). \tag{2}$$

It is worth noting that if a user has the right attributes with another time slot t' later than t_e they cannot compute the decryption key either. This feature has great resistance to the collusion attacks we mentioned before.

4.1.2. Ciphertext Decryption. After generating the decryption key with adequatea attributes and time slot, the decryption algorithm goes as follows: $\mathrm{Decrypt}(\mathrm{CT}, \mathrm{GPP}, \{\mathrm{PK}_d\}, \{\mathrm{DK}_{\mathrm{gid},x,t}\}_{x \in S_{\mathrm{gid},t}}) \longrightarrow (M)$ or $\perp$. For any $t \neq t_e$ or $S_{\mathrm{gid},t}$ that cannot satisfy (A, ρ), the algorithm outputs $\perp$. For those users who can be satisfied with conditions above, the decrypt algorithm goes in two steps. The first step is to find a $I = \{i \mid \rho(i) \in S_{\mathrm{gid},t_e}\}$ and compute the corresponding constants $\{\omega_i \mid i \in I\}$. For I, compute the following:

$$\begin{aligned}
\overline{C}_i &= \frac{C_{i,1} \cdot e\left(H(\mathrm{gid}), C_{i,2}\right)}{e\left(D_{\mathrm{gid},\rho(i),t_e}, C_{i,3}\right)} \cdot \frac{e\left(D'_{\mathrm{gid},\rho(i),t_e}, C'_{i,3}\right)}{e\left(D''_{\mathrm{gid},\rho(i),t_e}, C'_{i,4}\right)} \\
&= \frac{e(g,g)^{\lambda_i} e(g,g)^{\alpha_{\rho(i)} r_i} \cdot e\left(H(\mathrm{gid}), g^{\mu_i} g^{\beta_{\rho(i)} r_i}\right)}{e\left(g^{\alpha_{\rho(i)}} H(\mathrm{gid})^{\beta_{\rho(i)}}, g^{r_i}\right)} \cdot \frac{e\left(\left(R_{v_{\rho(i)}}\right)^{\alpha_{\rho(i)} \gamma_{\rho(i)} r_{\mathrm{gid},\rho(t),v_{\rho(i)}}}, g^{\left(r_i/\gamma_{\rho(i)}\right)}\right)}{e\left(\left(R_{v_{\rho(i)}}\right)^{\alpha_{\rho(i)} \gamma_{\rho(i)} r_{\mathrm{gid},\rho(t),v_{\rho(i)}}}, g^{r_i}\right)} \cdot \frac{e\left(H_{\phi(\rho(i))}(\rho(i), t_e)^{\varsigma_{v_{\rho(i)}, t} r_{\mathrm{gid},\rho(i),v_{\rho(i)}}}, g^{\left(r_i/\gamma_{\rho(i)}\right)}\right)}{e\left(g^{\left(\varsigma_{v_{\rho(i)}, t} r_{\mathrm{gid},\rho(i),v_{\rho(i)}}/\gamma_{\rho(i)}\right)}, H_{\phi(\rho(i))}(\rho(i), t_e)^{r_i}\right)} \\
&\quad \cdot \frac{e\left(\left(R_{v_{\rho(i)}}\right)^{\gamma_{\rho(i)} r_{\mathrm{gid}\, t, \rho(i), v_{\rho(i)}}}, g^{\left(r_i/\gamma_{\rho(i)}\right)}\right)}{e\left(\left(R_{v_{\rho(i)}}\right)^{r_{\mathrm{gid}\, t, \rho(i), v_{\rho(i)}}}, g^{r_i}\right)} \\
&= e(g,g)^{\lambda_i} \cdot e(H(\mathrm{gid}), g)^{\mu_i}.
\end{aligned} \tag{3}$$

And, the last step computes the following:

$$\begin{aligned}
\prod_{i \in I} \overline{C}_i^{\omega_i} &= e(g,g)^{\sum_{i \in I} \omega_{i\lambda_i}} \cdot e(H(\mathrm{gid}), g)^{\sum_{i \in I} \omega_i \mu_i} \\
&= e(g,g)^r.
\end{aligned} \tag{4}$$

Then, users can recover the symmetric encryption secret key by $K = C / e(g,g)^r$. After that, users can get the message by doing a symmetric decryption in which the computation cost is negligible.

If the user wants to outsource the decryption part to other computation devices, the first thing to do is to generate outsourced key and recover the key for decryption. The user generates the outsourced key as follows:

$$\mathrm{OK}_{\mathrm{gid},x,t} = \left(O_{\mathrm{gid},x,t} = K_{\mathrm{gid},x,v_x}, O'_{\mathrm{gid},x,t} = \left(E_{v_x}\right)^{K'^{(1/\sigma)}_{\mathrm{gid},x,v_x}} \cdot K''_{\mathrm{gid},x,v_x}, O''_{\mathrm{gid},x,t} = \left(E'_{v_x}\right)^{K'^{(1/\sigma)}_{\mathrm{gid},x,v_x}} \right). \tag{5}$$

Here, $\sigma \in Z^*$ is the recover key RK selected by the data user.

The second step is run by CSP or computing devices (they might be a node on the blockchain). After receiving the

OK from the user along with the ciphertext, they will partially decrypt the ciphertext as follows:

$$\overline{C}_i' = \frac{C_{i,1} \cdot e\left(H(\text{gid}), C_{i,2}\right)}{e\left(O_{\text{gid},\rho(i),t_e}, C_{i,3}\right)} \cdot \frac{e\left(O'_{\text{gid},\rho(i),t_e}, C'_{i,3}\right)}{e\left(O''_{\text{gid},\rho(i),t_e}, C_{i,4}\right)} = e(g,g)^{(\lambda_i/\sigma)} \cdot e(H(\text{gid}), g)^{(\mu_i/\sigma)}. \tag{6}$$

After that, they calculate CT′.

$$\text{CT}' = \prod_{i \in I} \overline{C}_i'^{\omega_i} = e(g,g)^{\sum_{i \in I}(\omega_i \lambda_i/\sigma)} \cdot e(H(\text{gid}), g)^{\sum_{i \in I}(\omega_i \mu_i/\sigma)} = e(g,g)^{(r/\sigma)}. \tag{7}$$

The last phase is executed on the user side. After the user gets CT′ from outsourcing entities, the user can use RK to retrieve the plaintext:

$$\begin{aligned} \text{De} &= \text{CT}\prime e(g,g)^{-(1/\sigma)} \\ &= \left(e(g,g)^{-(1/\sigma)}\right) e(g,g)^{-(1/\sigma)} \\ &= e(g,g)^{r}. \end{aligned} \tag{8}$$

4.2. Updating With Time Restriction. As we mentioned before, when a user tries to revoke an attribute or applies for a new attribute from AA, the secret keys associated with his attributes should be updated either. While in our scheme, this problem can be replaced by the changing of time slot. $\text{AA}_{\phi(x)}$ needs to run the update key algorithm when receiving the update request from data owners. The algorithm takes the input as the current time slot t, updates attribute $x \in U_{\phi(x)}$, and outputs the update key $\text{UK}_{x,t}$.

For example, when a data owner U_d tries to revoke an attribute x_i , he must send an updates request to the $\text{AA}_{\phi(x)}$ which is in charge of x_i. Then, $\text{AA}_{\phi(x)}$ selects a random set $\varpi\{\varpi_1, \varpi_2, \ldots, \varpi_k\} \in Z_p$ $(\sum_{n=1}^{k} \varpi_k = 0)$ where the numbers in the random set are equal to the numbers of the attribute authorities he wants to send to. After this phase, $\text{AA}_{\phi(x)}$ updates the attribute key component $\text{ASK}_{K,n}$ for the user U_d with the attribute x_i as $\text{ASK}'_{K,n} = D_i = g^{\gamma_n} H(i)^{\gamma_i}$, $D'_i = (g^{\gamma_i})^{\varpi_k}$.

By using updating algorithm, cloud servers can update the ciphertext without getting any sensitive information from the data owner. Meanwhile, only the cloud server knows about the s'. In this case, the new kind of collusion attacks between cloud and revoked users can be easily traced.

5. Security Analysis and Performance Evaluation

5.1. Security Analysis. As mentioned in the previous sections, in our scheme, the main difference between our scheme and Lewko and Waters scheme [25] is we embedded time slots into both ciphertexts and keys. So, the security of our scheme lies in the below attacks:

(1) Outsourcing entities try to infer information about the ciphertext and may compare the ciphertext to find differences in diverse data owners.

As the outsourcing entities undertake massive predecryption tasks from different users, they may collect different ciphertext. However, they cannot infer any information only from those ciphertexts as each CT has different t_e, access policy and bilinear paring.

(2) Users try to predict the data info from the ciphertext, which they cannot decrypt.

This attack assumption does not hold either because first users cannot get any reason about their failure on decryption but a symbol $\perp$.

Despite these attacks, our scheme is similar to the scheme in [25]. Thus, our scheme is secure in the bilinear group model, which is the same as the proof used in [25]. In the generic bilinear group model and random oracle model, there is no adversary that can break our scheme in polynomial time with a nonnegligible advantage in the security game we mentioned before. Moreover, we will make an analysis of our scheme from the other three parts: collusion resistance, data confidentially, and attributes revoke.

5.1.1. Collusion Resistance. There may exist several kinds of collude operations in our model and we will analyze them one by one. The first one is collusion between users. For those users whose attributes cannot meet the demand of access control policy, it is a common way to combine their secret keys with each other to get a new key that can decrypt the ciphertext. However, since the prime order of their secret key is randomly chosen by authorities, respectively, no matter what kind of attributes set they ever had, they cannot get a proper key in any case.

Another case is the collusion between an unauthorized user with a revoked user with gid_1. Users with gid_1 may want the secret key gid_2 once had to get a combination of attributes with their own gid_1. As mentioned before, this kind of collusion can not exist either because of the random oracle.

The last situation is as we listed in the introduction. For the traditional method, cloud server providers can forge the execution time of updating algorithm or even simply pull back the updating time and pass the buck to serious network delay. Users can select plural servers with their data stored and any one of the servers delay his updating does not make sense to those malicious revoked users. Because that the data owner can select the servers as his wish, it is nearly impossible for a revoked user to make a deal with all servers in the domain. From this point, we could say that this kind of collusion can barely exist.

5.1.2. Data Confidentiality. As we mentioned before, not all the channels are secure in our model. But for the data transmission part, all of the data are transmitted in ciphertext, so we only consider the situation below: the cloud

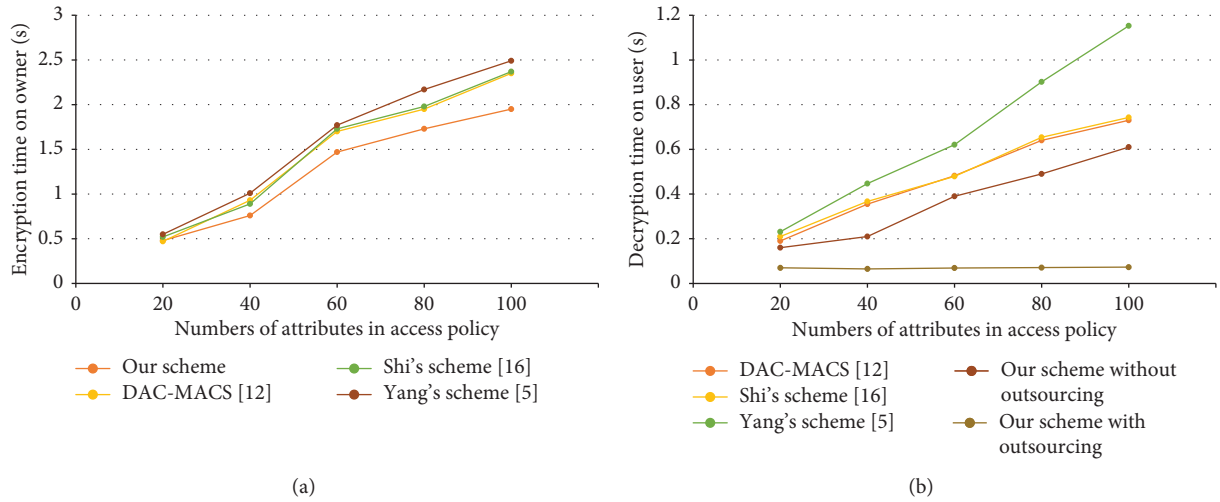


FIGURE 3: Encryption and decryption time cost with a different number of attributes. (a) Encryption time on the owner side. (b) Decryption time on the user side

server or any user who cannot access the data may get the tuple (Cipher, CT) at any time slot (here Cipher stands for the text transformed by plaintext). As the owner encrypts the plaintext such as symmetric encryption first, the Cipher does not leak any information about the data. On the other hand, CT can be decrypted only by those users with appropriate attributes, so attackers can not find any relations between Cipher and CT. Furthermore, any user who wants to recover the time slot or attributes in CT is not possible for the same reason.

5.1.3. Revocation. This part is slightly similar to the collusion resistance part as we mentioned before, and here we only consider the forward security and backward security of our scheme. Due to the fact that the time slot is running continuously, when a new user joins the system and is granted attributes from the authorities, the time slot match with his keys cannot be the period before his join. Thus, forward security can be ensured in this way. Similar to the forward security, when AA wants to revoke attributes of a user, the ciphertext cannot be decrypted by this user as the ciphertext is updated with a new time slot and the user cannot get a new key both with the revoked attributes and the new time slot.

5.2. Performance Evaluation

5.2.1. Experimental Setup. The simulation platform for our scheme is Ubuntu 14.04 with an Intel Core (TM) i5-5600U at 2.6 GHz and 4 GB RAM. The simulation environment is JPBC (Java Pairing-Based Cryptography library ver2.0.0) with 160-bit group order, 512-bit field size.

5.2.2. Algorithm Analysis. In our experiments, we did 10000 trials to limit the error range. We first compare the computation cost on owners. As we can see in Figure 3, the computation time takes about 10% less than Hur's scheme and the DAC-MACS. While in the decryption part, we can see that the computation time cost is much less. In real-world scenarios, it is obvious that the operation of downloading is much more than updating because data users in the cloud environment are huge. Meanwhile, as in a cloud-based system, it is more important to save the computation cost on the user side as mobile devices are widely used nowadays.

5.2.3. Attribute Impact. In this section, we try to simulate the impact on attribute numbers on both the encryption side and the decryption side. We set the 10 AA with different attributes from 1 to 20. Specifically, the cost on the decryption side (with both outsourcing and self-decryption) and the computation on encryption are evaluated in Figure 3 to show the impact on the attribute universe more precisely. The experiment shows that both encryption and decryption costs grow steadily with the rise of attributes number. Outsourced decryption time on the user side almost stays the same as no matter how many attributes are in the policy, the user only needs to compute the paring algorithm for once. Moreover, we also take the key generation time into consideration. As we can see from Figure 4, our attribute authority private key generation time is still between the DAC-MACS case and Shi's scheme, and the attribute authority private key generation time is proportional to the number of attributes.

5.2.4. Attribute Update Evaluation. As lots of the scheme only takes policy updates into consideration, we only compared our attribute updating algorithm with DAC-MACS, which have a similar part to ours. With the increase of the attributes, owners need more time to generate the update key while the time spent on servers almost stays the same because the major part of the computation task has been done on the cloud side. For most situations in reality, the owner can afford the computation cost.

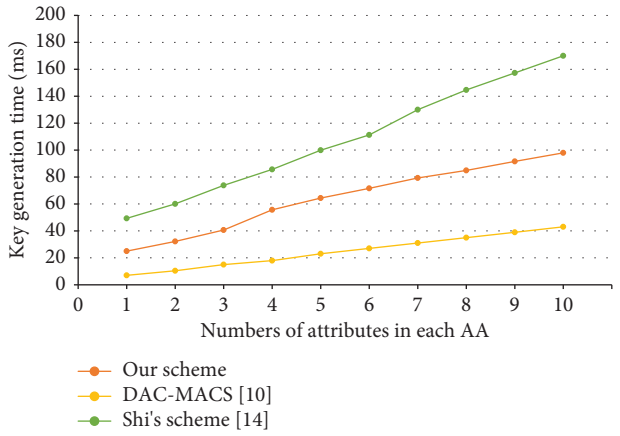


Figure 4: Key generation cost with different numbers of attributes in AA.

From the above aspects, both theoretical analysis and experiments results show that our scheme can provide time slot access control and attributes update at the cost of a slight increase of key generation phase. However, as the key is generated on AA, it barely has any effect since AA normally has enough computation power. Moreover, comparing with the existing schemes, our blockchain-enabled data sharing scheme has a high level of security of collusion resistance.

6. Discussion and Conclusion

In this paper, we have proposed a collusion-resistant CP-ABE blockchain-enabled data sharing scheme to achieve access control under a time restriction. Specifically, we have proposed this scheme using time slots as a token to bind with ciphertexts and keys to make sure that only the user with demanded attributes and in the particular time slot can decrypt the data. Besides, we considered a new kind of collusion, which might be common in our daily life that curious cloud servers may delay the policy update/attribute revoke algorithm for a short time to let the revoked user get data illegally. This kind of collusion is hard to trace because cloud servers can easily shift their responsibility to other reasons like network latency and so on. Furthermore, we used time slots to ensure data security while the decryption phase is outsourced. Further discussion on our schemes is about security issues on revocation and collusion resistance. In the future, we will keep on implementing our scheme and exploring the access control structures with a time restriction and taking time into consideration in the case of collusion between the revoked user and one of the authorities.

Data Availability

The data used to support the findings of this study are available from the authors upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was supported by the National Science Foundation of China (nos. 61772385 and 61572370.)

References

[1] Z. Tian, C. Luo, H. Lu, S. Su, Y. Sun, and M. Zhang, "User and entity behavior analysis under urban big data," *ACM/IMS Transactions on Data Science*, vol. 1, no. 3, 2020.

[2] Z. Tian, C. Luo, J. Qiu, X. Du, and M. Guizani, "A distributed deep learning system for web attack detection on edge devices," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 1963–1971, 2020.

[3] A. Sahai and B. Waters, "Fuzzy identity-based encryption," *Lecture Notes in Computer Science*, vol. 3494, pp. 457–473, 2005.

[4] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM conference on Computer and communications security*, pp. 89–98, Alexandria, VA, USA, October 2006.

[5] K. Yang, Z. Liu, X. Jia, and X. S. Shen, "Time-domain attribute-based access control for cloud-based video content sharing: a cryptographic approach," *IEEE Transactions on Multimedia*, vol. 18, no. 5, pp. 940–950, 2016.

[6] J. Liu, X. Huang, and J. K. Liu, "Secure sharing of personal health records in cloud computing: ciphertext-policy attribute-based signcryption," *Future Generation Computer Systems*, vol. 52, pp. 67–76, 2015.

[7] L. I. Youhuizhi, Z. Dong, K. Sha, C. F. Jiang, J. Wan, and Y. Wang, "TMO: time domain outsourcing attribute-based encryption scheme for data acquisition in edge computing," *IEEE Access*, vol. 7, Article ID 40240, 2019.

[8] B. Q. Baodong, R. H. Deng, S. L. Shengli, and S. M. Siqi, "Attribute-based encryption with efficient verifiable outsourced decryption," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 7, pp. 1384–1393, 2015.

[9] L. Zhang, C. Xu, Y. Gao, Y. Han, X. Du, and Z. Tian, "Improved Dota2 lineup recommendation model based on a bidirectional LSTM," *Tsinghua Science and Technology*, vol. 25, no. 6, pp. 712–720, 2020.

[10] Z. Gu, W. Hu, C. Zhang, H. Lu, L. Yin, and L. Wang, "Gradient shielding: towards understanding vulnerability of deep neural networks," *IEEE transactions on network science and engineering*, vol. 8, no. 2, 2020.

[11] Z. A. Lei, B. St, and C. Ll, "An finite iterative algorithm for sloving periodic Sylvester bimatrix equations," *Journal of the Franklin Institute*, vol. 357, no. 15, Article ID 10757, 2020.

[12] K. Yang, X. Jia, K. Ren, B. Zhang, and R. Xie, "DAC-MACS: effective data access control for m cloud storage systems," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 11, pp. 1790–1801, 2013.

[13] K. Yang and X. Jia, "Expressive, efficient, and revocable data access control for multiauthority cloud storage," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 7, pp. 1735–1744, 2014.

[14] D. Wang, L. Zhang, C. Huang, and X. Shen, "A privacy-preserving trust management system based on blockchain for vehicular networks," in *Proceedings of the 2021 IEEE Wireless Communications and Networking Conference (WCNC)*, Nanjing, China, March 2021.

[15] K. Yang, X. Jia, and K. Ren, "Attribute-based fine-grained access control with efficient revocation in cloud storage

systems," in *Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security*, pp. 523–528, Hangzhou, China, May 2013.

[16] J. Shi, C. Huang, J. Wang, K. He, and J. Wang, "An access control scheme with direct cloud-aided attribute revocation using version key," in *Proceedings of the International Conference on Algorithms and Architectures for Parallel Processing*, pp. 429–442, Dalian, China, August 2014.

[17] T. Kitagawa, H. Kojima, N. Attrapadung, and H. Imai, "Efficient and fully secure forward secure ciphertext-policy attribute-based encryption," *Lecture Notes in Computer Science*, vol. 7807, pp. 87–99, 2015.

[18] L. Zhang, Z. Huang, W. Liu, Z. Guo, and Z. Zhang, "Weather radar echo prediction method based on convolution neural network and Long Short-Term memory networks for sustainable e-agriculture," *Journal of Cleaner Production*, vol. 298, Article ID 126776, 2021.

[19] Q. Liu, G. Wang, and J. Wu, "Time-based proxy re-encryption scheme for secure data sharing in a cloud environment," *Information Sciences*, vol. 258, pp. 355–370, 2014.

[20] J. Hong, K. Xue, Y. Xue et al., "TAFC: time and attribute factors combined access control for time-sensitive data in public cloud," *IEEE Transactions on Services Computing*, vol. 13, no. 1, 2017.

[21] Z. Gu, L. Wang, and X. Chen, "Epidemic risk assessment by A novel communication station based method," *IEEE Transactions On Network Science And Engineering*, 2021.

[22] N. Attrapadung and H. Imai, "Attribute-based encryption supporting direct/indirect revocation modes," in *Proceedings of the IMA International Conference on Cryptography and Coding*, pp. 278–300, Cirencester, UK, December 2009.

[23] A. Sahai, H. Seyalioglu, and B. Waters, "Dynamic cciphertext delegation for attribute-based encryption," *Lecture Notes in Computer Science*, vol. 7417, pp. 199–217, 2012.

[24] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," in *Proceedings of a Workshop on the Theory and Application of Cryptographic Techniques*, pp. 10–18, Linz, Austria, April 1985.

[25] A. Lewko and B. Waters, "Decentralizing attribute-based encryption," in *Proceedings of the Advances in Cryptology - EUROCRYPT 2011*, pp. 568–588, Tallinn, Estonia, May 2011.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 9962575, 14 pages
https://doi.org/10.1155/2021/9962575

Research Article

Blockchain-Based Secure Outsourcing of Polynomial Multiplication and Its Application in Fully Homomorphic Encryption

Mingyang Song, Yingpeng Sang, Yuying Zeng, and Shunchao Luo

School of Computer Science and Engineering, Sun Yat-Sen University, Guangzhou, China

Correspondence should be addressed to Yingpeng Sang; sangyp@mail.sysu.edu.cn

Received 15 March 2021; Revised 26 May 2021; Accepted 7 June 2021; Published 25 June 2021

Academic Editor: Yinghui Zhang

The efficiency of fully homomorphic encryption has always affected its practicality. With the dawn of Internet of things, the demand for computation and encryption on resource-constrained devices is increasing. Complex cryptographic computing is a major burden for those devices, while outsourcing can provide great convenience for them. In this paper, we firstly propose a generic blockchain-based framework for secure computation outsourcing and then propose an algorithm for secure outsourcing of polynomial multiplication into the blockchain. Our algorithm for polynomial multiplication can reduce the local computation cost to $O(n)$. Previous work based on Fast Fourier Transform can only achieve $O(n\log(n))$ for the local cost. Finally, we integrate the two secure outsourcing schemes for polynomial multiplication and modular exponentiation into the fully homomorphic encryption using hidden ideal lattice and get an outsourcing scheme of fully homomorphic encryption. Through security analysis, our schemes achieve the goals of privacy protection against passive attackers and cheating detection against active attackers. Experiments also demonstrate our schemes are more efficient in comparisons with the corresponding nonoutsourcing schemes.

1. Introduction

As the development of the big data era, there is an increasing demand for large-scale time-consuming computations. Fortunately, with the emergence of cloud computing, computation outsourcing brings convenience to resource-constrained users. They can outsource complex computing tasks into the cloud by paying a fee and avoiding buying expensive high-performance hardware. It not only improves the resource utilization in cloud but also brings economic benefits to resource-constrained users. Nevertheless, the attractive computing scheme also causes security issues. A passive attacker in the cloud may be only curious about the privacy contained in the user's outsourced data, while an active attacker may make malicious damage or forge the results to sabotage the computation. Even if there is no attacker, computing errors caused by cloud hardware failure and software errors, etc. should also be considered. Furthermore, it should not be a great burden for the user to check the correctness of the returned results from the cloud; otherwise, the efficiency benefit of outsourcing will be nullified. Therefore, the secure and efficient outsourcing of computations that can not only protect the privacy of users but also ensure the correct results has become a hot research topic.

Gentry proposed a homomorphic encryption algorithm based on ideal lattice [1] for the first time, providing us with a direction to solve the privacy issues in computation outsourcing. The direction is a secure computation outsourcing mode: encryption-outsourcing-decryption (EOD). Even if we use the common EOD model, the device should also undertake the computations of secret key generation, encryption, verification, decryption, and so on, locally. These computations are also great burden for the resource-constrained devices (such as mobile phones and IoT nodes).

Blockchain has attractive features such as transparency, traceability, decentralization, and immutability, which make it an optimal approach for applications intrinsically with untrusted natures, such as computation outsourcing. A central trusted entity is not required for the computation outsourcing based on blockchain. Information about the whole data exchange process, computations, users, and computational nodes is recorded and is traceable in blockchain. Besides, smart contract can be utilized to digitally facilitate the implementation of whole transaction, which greatly improves the speed of building applications on blockchain. However, privacy is still an issue in the computation outsourcing based on blockchain.

Owing to the low efficiency of fully homomorphic encryption algorithms, the general computation outsourcing mode based on EOD is impractical on the resource-constrained devices. In this paper, we will outsource some complex computations in the fully homomorphic encryption using hidden ideal lattice (FHEHIL) [2] into a blockchain framework. The contributions of this paper can be summarized as follows:

(1) We propose a framework of blockchain-based computation outsourcing, in which we can implement secure outsourcing for FHEHIL. The framework has a credit-based task allocation strategy, which will significantly reduce the probability of malicious nodes participating in computing.

(2) We propose a secure outsourcing algorithm for polynomial multiplication, which reduces the local computation cost (including the cost on result verification) to $O(n)$. Previous work based on the Fast Fourier Transform (FFT) can only achieve $O(n\log(n))$ for the local cost. Besides, the algorithm can not only detect cheating but also identify cheating nodes combining with blockchain. The result verification in the outsourcing algorithm does not cause extra burden.

(3) We also extend the secure outsourcing algorithm of modular exponentiation, in [3], in our blockchain-based framework. The two algorithms for polynomial multiplication and modular exponentiation are employed in FHEHIL as basic operations, and the FHEHIL implementation on the blockchain-based framework can have higher efficiency compared with previous work.

2. Related Work

At present, research studies on secure outsourcing can be roughly divided into two directions. In one direction, a general outsourcing mechanism is studied. In this mechanism, a fully homomorphic encryption algorithm is designed and the EOD model is used to outsource any computations. After the work of Gennaro et al. [1], great progress has been made in the field of fully homomorphic encryption [4–6]. However, the fully homomorphic encryption algorithms have high computational complexity. Recently, there are many researches to reduce the computation cost of homomorphic encryption algorithm. For example, Su et al. accelerated the leveled Ring-LWE fully homomorphic encryption [7]. These research studies mainly focus on the efficiency of hardware. However, secure outsourcing complex computations of fully homomorphic encryption is a better way to improve efficiency for resource-constrained devices.

In the other direction, specific outsourcing algorithms are designed for various scientific computations, e g., modular exponentiation, solution of large-scale linear equations [8], bilinear pairings, and extend Euclidean. The Wei pairing and Tate pairing in algebraic curves are commonly used in key establishment and signature schemes in the field of cryptography. However, the computation of bilinear pairings is time-consuming in resource-constrained devices. Thus, many outsourcing schemes have been proposed [9–11]. To our knowledge, the scheme in [9] is the most efficient and secure till now. Because of the wide application in cryptography, the study on modular exponentiation is also a hot topic of research. Hohenberger and Lysyanskaya [12] proposed a modular exponentiation secure outsourcing scheme. Chen et al. [13] further improved its efficiency and verifiability. Ren et al. [14] proposed a scheme that only protects the privacy of exponent. Recently, Fu et al. [3] proposed a secure outsourcing scheme of modular exponentiation with hidden exponent and base. It has a stronger checkability. In cryptography, extended Euclidean algorithm is usually used to calculate modular inverse, which is widely used in RSA encryption algorithm. Similarly, Euclidean algorithm can be used to find the greatest common factor of two polynomials, which is commonly used in encryption algorithm based on Lattice. Zhou et al. [15] proposed the secure outsourcing algorithm of extended Euclidean algorithm.

Polynomial multiplication is likewise a commonly used operation in cryptography schemes, error correcting codes, and computer algebra. The complexity of polynomial multiplication is still a major open problem. Using the FFT, the local computation of polynomial multiplication can achieve the complexity of $O(n\log(n))$. Recently, some efficient polynomial multiplication methods based on the FFT are proposed. Harvey et al. proposed a faster method over finite fields Z_p when the degree of polynomials is less than p [16]. The efficiency has been further improved in [17]. For the hardware utilization, Liu et al. designed a high hardware efficiency polynomial multiplication on field-programmable gate array (FPGA) platform [18]. Hsu and Shieh proposed a method with less addition and multiplication [19] in 2020. Besides, there are also some research studies on reducing the space complexity of polynomial multiplication [20, 21]. However, the complexity of some research studies based on the acceleration of FFT remains at $O(n\log(n))$. Other methods using distributed computing to improve hardware utilization are not applied to the resource-constrained device. Till now, there are few research studies on the secure outsourcing of polynomial multiplication.

Due to the characteristics of the blockchain and Bitcoin [22], there are lots of research studies and applications on blockchain in recent years including the secure outsourcing. Lin et al. studied the secure outsourcing for bilinear pairings

based on blockchain [23]. Zheng et al. [24] proposed a secure outsourcing scheme for attribute-based encryption on blockchain. There are also some schemes [25, 26] of outsourced data integrity verification. The fairness problem in blockchain-based secure multiparty computation was also solved by multiple efforts. For example, Gao et al. [27] proposed a scheme which realized fairness by maintaining an open reputation system. This type of general scheme for secure multiparty computation can be cumbersome for the problems in secure outsourcing computation.

Andrychowicz et al. [28] utilized only scripts in Bitcoin currency to construct a fair protocol for secure multiparty lotteries, without relying on a trusted third party. Zhang et al. proposed BCPay in [29] and BPay in [30] to achieve fair payment for blockchain-based outsourcing services, which are compatible with the Bitcoin and Ethereum platforms. However, these frameworks can only provide fairness between the client and a single server. They are applicable to the outsourcing scenarios where the task is outsourced to a single server from the client. In the problem of our work, the computation task needs to be outsourced to multiple computational nodes simultaneously. Therefore, we propose a new one, considering the penalty on the cheating nodes, compensation on the honest nodes, and application of a credit-based scheme.

3. Notations and Background

3.1. Notations. We use upper case bold letters for matrices and $\det(M)$ for the determinant of matrix M. Lower case bold letters represent vectors (eg., $v = [v_1, \ldots, v_n]$), where v_i is the i^{th} element in v. We denote polynomial by lower case italics (eg., $f(x)$). For a rational number r, $\text{round}(r)$ represents the nearest integer to r. The rational vector v can also be rounded to $\text{round}(v) = [\text{round}(v_1), \ldots, \text{round}(v_n)]$. We use $v(x)$ for the polynomial form of the vector v. We use $v_1 \times v_2$ for polynomial multiplication ($v_1 \times v_2 = (v_1(x) \times v_2(x)) \bmod f(x)$) on the ring. We use $|v|$ for the norm of v and $|S|$ for the base of set S. We use $\mathrm{v} \circ \mathrm{w}$ for correlation ($v \circ w = [v_1 * w_1, \ldots, w_n]$). We use $R(v, f)$ for the rotation matrix of v whose i^{th} row is the coefficients of $v(x) \times x^{(i-1)} \bmod f(x)$. We use $x\gcd(a(x), b(x))$ for the extended Euclidean algorithm on $a(x)$ and $b(x)$. $\deg(f(x))$ represents the degree of $f(x)$. We use $\mathbf{F}_v$ to denote the coefficient set of Discrete Fourier transform (DFT) of v and $\mathbf{F}_v^-$ for the coefficient set of inverse DFT of $\mathbf{v}$.

3.2. Fully Homomorphic Encryption Using Hidden Ideal Lattice. The FHEHIL scheme [2] is described in Algorithm 1, including the components of key generation, encryption, and decryption. The related parameters are shown in Table 1.

As shown in Algorithm 1, it is obvious that polynomial multiplication is the primary computation in encryption and decryption. Therefore, our proposed algorithm for the secure outsourcing of polynomial multiplication can be directly applied. As for the key generation, the computing burden is from Step 7, 9, and 11. The main computational cost of Step 7 is on computing the determinant of matrix V. The most time-consuming operation in Step 9 is polynomial multiplication. Step 11 computes the inverse of polynomial. Below, we will analyze the detailed computations in Step 7 and 11 and demonstrate that polynomial multiplication and modular exponentiation are the main types of computations, which are the two improvement aims of this paper.

3.2.1. The Method of Computing d in Algorithm 1. Because of the characteristic of V, computing the determinant of matrix V needs only $\log(n)$ times of polynomial multiplication using the method in [31]. d is the free item of $g(z) = \prod_{i-0}^{n-1}(v(p_i) - z)$. Thus, $d = \prod_{i=0}^{n-1} v(p_i)$, where p_i is the root of $f(x) = 0$ in the complex domain and they satisfy equation (2). Due to equation (2), we have $d = \prod_{i=0}^{n-1} v(p_i) = \prod_{i=0}^{n/2-1} v(p_i)v(-p_i) = \prod_{i=0}^{n/2-1} a(p_i)$. Thus, the computation of d mainly involves polynomial multiplications and modular exponentiations.

3.2.2. The Method of Computing w in Algorithm 1 When d Is a Prime. In the algorithm of FHEHIL, if d is a prime, we can obtain w by performing $x\gcd(v, f)$ once. The extended Euclidean algorithm computes $x\gcd(a(x), b(x))$ to get $u(x)$, $v(x)$, and $d(x)$, satisfying $u(x) \times a(x) + v(x) \times b(x) = d(x)$. The specific procedures of secure outsourcing for extended Euclidean algorithm [15] are summarized in Algorithm 2. It can be seen that the local computations of this algorithm consist of mostly modular exponentiation and polynomial multiplications. Detailed analysis of Algorithm 2 can be found in [15].

3.2.3. The Method of Computing w in Algorithm 1 When d Is Not a Prime. When d is not a prime, the fastest method to calculate polynomial inverse at present is Gentry's method in [31]. The method is based on fast Fourier transform and halves the number of terms in each step to offset the doubling of the bit length of the coefficients. This method relays on $f(x) = x^{n+1}$, where n is a power of 2. The method is analyzed as follows.

Firstly, the second coefficient (g_1) of polynomial $g(z) = \prod_{i=0}^{n-1}(v(p_i) - z)$ can be computed, where p_i is the ith root of $f(x) = 0$ in the complex domain. We can get $w_0 = (g_1/n)$. Secondly, $v'(x) = x \times v(x) \bmod f(x)$ can be constructed. Then, the second coefficient g_1') of polynomial $g'(z) = \prod_{i=0}^{n-1}(v'(p_i) - z)$ can be computed. We can get $w_1 = (g_1'/n)$. Finally, other coefficients of w can be computed by

$$\frac{w_1}{w_0} = \frac{w_2}{w_1} = \cdots = \frac{w_{n-1}}{w_{n-2}}, \tag{1}$$

$$\left(p_i + \frac{n_j}{2}\right)^{2^i} = -\left(p_i^{2^i}\right)\left(n_j = \frac{n}{2^j}\right). \tag{2}$$

Since n is a power of 2 in $f(x)$, the roots satisfy equation (2). In the process of computing coefficients g_1 and g_1', the major computations are also polynomial multiplications and modular exponentiations.

Input: $\zeta, \gamma, \eta, t, n$
Output: SK = $\{d, w\}$, PK = $[p_1, \ldots, p_t]$
(1) **function** KEY GENERARTION $(\zeta, \gamma, \eta, t, n)$
(2) generate a random vector v satisfying $\left\{v \in Z^n, 2^{\gamma-1} < |v| < 2^{\gamma}, \sum_{i=0}^{n-1} v_i \bmod 2 = 1\right\}$
(3) generate $t-1$ random vectors $[g_1, \ldots g_{t-1}]$ satisfying $\{g_i \in Z^n, 2^{n-1} < [g_i] < 2^n\}$
(4) generate a random vector $\mathbf{g_t}$ satisfying $\left\{g_t \in Z^n, [g_i] < 2^n, \sum_{i-0}^{n-1} g_t|i| \bmod 2 = 1\right\}$
(5) generate $t-1$ random vectors $[r_1, \ldots, r_{t-1}]$ satisfying $\{r_i \in \{0, 1, -1\}^n, |r_i| < \zeta\}$
(6) generate a random vector $\mathbf{r_t}$ satisfying $\left\{r_t \in \{0, 1, -1\}^n, |r_t| < \zeta, \sum_{i=0}^{n-1} r_t|i| \bmod 2 = 1\right\}$
(7) $f(x) \longleftarrow x^n + 1$; $V \longleftarrow R(v, f)$; $d \longleftarrow |\det(V)|$
(8) **for** $i = 1 \longrightarrow t$ do
(9) $p_i \longleftarrow g_i \times v + r_i$
(10) **end for**
(11) Get **w** satisfying $w \times v = d \bmod f$
(12) **return** SK = $\{d, w\}$, PK = $[p_1, \ldots, p_t]$
(13) **end return**
(14) **function** ENCRYPTION (PK = $[p_1, \ldots, p_t]$, plaintext)
(15) generate $t-1$ random integer vectors $[s_1, \ldots, s_{t-1}]$ satisfying $\sum_{j=0}^{n-1} s_i[j] \bmod 2 = 0$
(16) generate a random vector s_t satisfying $\sum_{i=0}^{n-1} s_t[j] \bmod 2 =$ plaintext
(17) generate a random vector s_{t+1} satisfying $\sum_{j=0}^{n-1} s_{t+1}[j] \bmod 2 = 0$
(18) $\psi \longleftarrow \sum_{i=1}^{t} s_i \times p_i + s_{t+1}$
(19) **return** ψ
(20) **end function**
(21) **function** DECRYPTION (SK = $(d, w), \psi$)
(22) $\psi' \longleftarrow \text{round}(\psi \times w/d)$
(23) plaintext $\longleftarrow \psi'(1) \bmod 2$
(24) **return plaintext**
(25) **end function**

Algorithm 1: Fully homomorphic encryption using hidden ideal lattice.

Table 1: Parameters in the FHEHIL algorithm.

Parameters	Implication
ζ	The norm of random noise vector
γ	The bit length of norm of generating polynomial
η	The bit length of norm of the random multiplier vector
t	The number of vectors contained in the public key
n	The dimension of the hidden lattice (power of 2)

4. The Framework of Blockchain-Based Computation Outsourcing

This section introduces a blockchain-based computation outsourcing framework. The overall system model is illustrated in Figure 1, and the related symbols are described in Table 2. In Figure 1, we assume that, at least, one trusted third party is available to implement the smart contract. This is a generic model on which a variety of computational tasks can be implemented, including the tasks of secure outsourcing of polynomial multiplication and modular exponentiation. The two specific tasks will be given in details in the rest of this paper.

4.1. Registration. Users and computational nodes need to register before joining the network. They need to pay deposits in advance, the amount of which needs to be greater than a specified threshold or they will be rejected. The smart contract initializes the same credit score to all new nodes and users. After registration, information of users and computational nodes is written to the smart contract. The specific function is shown as Algorithm 3. In this algorithm, addr[p] is the deposit account of node p in the smart contract. Node p's asset privacy is protected because it is unnecessary for him to expose his total asset to the smart contract.

4.2. Computational Service. User p posts computing tasks, data, and rewards of each task to the smart contract. If the account balance of p is insufficient for the computations, smart contract refuses this service and reduces the credit score of p, to prevent malicious users from attacking the smart contract by constantly sending tasks that they cannot afford to. After the smart contract accepts the tasks of p, the user's computing tasks are stored in the task queue and wait for the selection of computational nodes.

To achieve a high benefit, the computational nodes will be active to undertake computing tasks. If multiple computational nodes select the same task at the same time, the node with the highest credit score will win the task. Nodes that undertake the computing task should submit the results after completing. If any computational node cannot finish on time, it will be added to the dishonest set. If all computational nodes can submit the results on time, the smart contract sends the results to user and initiates the period of dispute resolving. During this period, the user needs to verify the results locally and notify the smart contract whether the

Input: $a(x)$, $b(x)$
Output: $v(x)$, $v(x)$, $d(x)$
(1) generate $r(x)$ and $a \in Z$ randomly
(2) $f(x) \longleftarrow a(\alpha x), g(\alpha x) \longleftarrow b(x)$
(3) $a'(x) \longleftarrow r(x) \times f(x), b'(x) \longleftarrow r(x) \times g(x)$
(4) $a''(x) \longleftarrow u_{11}(x) \times a'(x) + u_{12}(x) \times b'(x), b''(x) \longleftarrow u_{21}(x) \times a'(x) + u_{22}(x) b'(x)$
(5) send $a''(x)$ and $b''(x)$ to computational node
(6) get $u''(x), v''(x)$ and $d''(x)$ from computational mode
(7) verify $a''(x) \times u''(x) + b''(x) \times v''(x) = d''(x)$; $\deg(u''(x)) < \deg(b''(x))/d''(x)$; $\deg(v''(x)) < \deg(a''(x))/d''(x)$;
(8) $u(x) \longleftarrow \alpha^{\deg(d''(x)) - \deg(r(x))}(u_{11}(\alpha^{-1}x) \times u''(\alpha^{-1}x) + u_{21}(\alpha^{-1}x) \times v''(\alpha^{-1}x))$
(9) $v(x) \longleftarrow \alpha^{\deg(d''(x)) - \deg(r(x))}(u_{12}(\alpha^{-1}x) \times u''(\alpha^{-1}x) + u_{22}(\alpha^{-1}x) \times v''(\alpha^{-1}x))$
(10) $d(x) \longleftarrow \alpha^{\deg(d''(x)) - \deg(r(x))} d''(\alpha^{-1}x)/r(\alpha^{-1}x)$

Algorithm 2: Securely outsource the extended Euclidean algorithm.

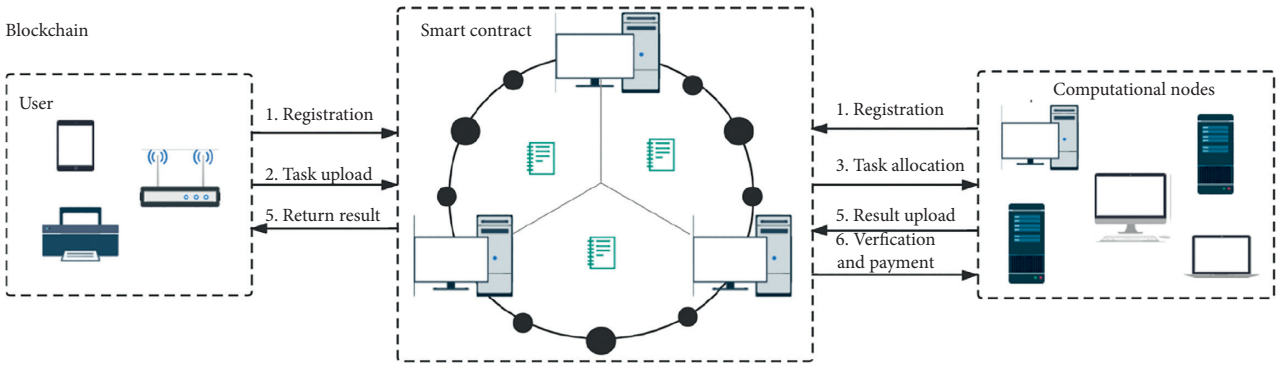


Figure 1: The framework of blockchain-based computation outsourcing.

Table 2: Symbols in the framework of blockchain-based computation outsourcing.

Symbol	Implication
addr[p]	The account address of p
User/node	The set of users/computational nodes
Rep[p]	The reputation of p
Task	The queue of published tasks
Allocated	The queue of allocated tasks
Data	The queue of published data
Time	The set of the latest result return time
Dishonest/honest	The set of dishonest/honest nodes
Result	The set of results
Δb	The amount of economic punishment
Δc	The amount of credit punishment/reward
Δt	The longest computing time consumption

results are accepted or not. If the period of dispute ends and there is no feedback received from the user, the smart contract assumes that the computations succeed and performs the reward and charge operations. If the user does not accept the results in the feedback, the smart contract will verify the results by itself. The specific function is shown as Algorithm 4.

4.3. Verification and Payment. If the user does not accept the computing results, the smart contract will perform verification operations to find dishonest nodes or users. In the verification, he will simulate all computations required by the user on the encrypted data uploaded by the latter. This means he will repeat exactly every step the computational nodes have carried out, so as to find which step is not correct and who is cheating. Decryptions are not required in the simulation, and thus, the data privacy of the user is protected.

To complete the verification, the smart contract should be equipped with the same function modules as those of the computational nodes. For example, in the secure outsourcing of polynomial multiplication, a function should be

```
(1) function Registration(p, role)
(2)   if the balance of addr[p] ≥ the threshold then
(3)     if role = user then
(4)       put addr[p] into User
(5)     else
(6)       put addr[p] into Node
(7)     end if
(8)       Rep[p] ⟵ Default credit value; state ⟵ true
(9)     else
(10)      state ⟵ false
(11)    end if
(12)    return false
(13) end function
```

Algorithm 3: Registration.

```
(1) function TASK UPLOAD (user, tasks, data, price)
(2)   if the balance of addr[user] ≥ |tasks| * price then
(3)     put tasks into Task; put data into Data
(4)   else
(5)     Rep[user]⟵Rep[user] − Δc
(6)   end if
(7) end function
(8) function TASK ALLOCATION (nodes, task)
(9)     if task ≠ ALLOCATED then
(10)      sort the nodes with their credit descending; send task and Data[task] to nodes [0]
(11)      put task into ALLOCATED; Time[task]⟵current time + Δt
(12)    end if
(13) end function
(14) function RESULT UPLOAD (node, task, result)
(15)    if current time < Time[task] then
(16)      put node into Honest; Put result into result
(17)    else
(18)      put node into Dinhonest
(19)    end if
(20) end function
(21) function RESULT UPLOAD(node, task, result)
(22)    if Dishonest = 0 and |Result| = |Task| then
(23)      send Result to user; Wait for the feedback from user
(24)    end if
(25)    call verification and payment
(26) end function
```

Algorithm 4: Computational service.

added into the smart contract to simulate the FFT/IFFT operations on the data uploaded by the user in case of disputes.

The dishonest nodes and users will be put into the dishonest set. If no one is put into the dishonest set, the user will pay the reward to all participating nodes. Otherwise, cheating nodes will be penalized and the honest nodes will be compensated. The credit score of the participating nodes that have correctly completed their tasks will increase, while the credit score of the malicious nodes will decrease. When the account balance of a node is lower than the threshold value or its credit score is reduced to zero, the system will remove it. The specific function is shown as Algorithm 5.

A malicious user with enough balance may constantly initiate transactions, aiming to increase the burden of the smart contract. However, he cannot refuse to pay because his deposit account is managed by the smart contract. By setting up a suitable threshold in the registration, sooner or later his balance will be used up by his attack.

```
(1) function VERIFICATION AND PAYMENT (feedback, Result, Honest, user, price)
(2)   if feedback ≥ user or (feedback = NULL and |Result| = |Task|) then
(3)     addr[user] ⟵ addr[user]-[Honest] ∗ price
(4)     for i = 1⟶|Honest| do
(5)       Rep[node_i] ⟵ Rep[node_i] + Δc; addr[node_i]⟵addr[node_i] + price
(6)     end for
(7)   else
(8)     stimulate all computations and put dishonest user or nodes into Dishonest
(9)     t ⟵ Δb ∗ [IDishonest]
(10)    for i = 1⟶|Dishonest| do
(11)      Rep[node_i]⟵Rep[node_i] + Δc; addr[node_i] ⟵ addr[node_i]−Δb
(12)    end for
(13)    for i = 1⟶|Honest|do
(14)      Rep[node_i] ⟵ Rep[node_i] + Δc; addr[node_i] ⟵ addr[node_i] + t/|Honest|
(15)    end for
(16)  end if
(17) end function
```

Algorithm 5: Verification and payment.

4.4. Security Analysis. Both the malicious user nodes and computational nodes can launch attacks to the framework, but their strategies are different. The malicious user nodes could launch a DDoS attack. A malicious computational node could destroy the computation by returning forged results or not returning any result.

The user nodes could employ two ways to launch a DDoS attack. One is to continuously publish the tasks that the user actually cannot afford to; the other is to maliciously inform the smart contract that the results are not accepted during the dispute resolving period. For the first attack, the smart contract will refuse to add the computing tasks and data to the queue and reduce the credit score of the user. Moreover, when the node's credit score drops below 0, the node will be removed. We can increase Δc in the function of TASK UPLOAD to remove malicious users as soon as possible. For the second attack, the smart contract has to simulate the computations of all nodes participating in the outsourcing according to the data and records. This attack has a greater impact on the smart contract, but it brings more loss to the attackers (including the financial punishment). Similarly, we can increase Δb in Algorithm 5 to mitigate the impact on smart contracts.

The computational nodes also have two ways to deploy attack: returning forged results or not returning any result. The cost of both attacks is the same (in financial and credit punishment). Since forged results render the smart contract to simulate the computations of all nodes, rational computational nodes prefer to attack by returning forged results, rather than return nothing. Similarly, we can increase Δb and Δc in Algorithm 5 to mitigate the impact on smart contracts. The proposed framework adopts a task allocation strategy based on credit scores. When malicious nodes are found, their credit score is reduced, and their probability of obtaining computing tasks in the next time is also reduced. We assume there are enough computational nodes which are willing to return correct results to fulfil the requirement of outsourcing, under the incentives of achieving financial and credit reward.

4.5. Compatibility Analysis. We know that the Bitcoin script is not Turing-complete, and Ethereum has a complete programming language on the blockchain to execute more complex smart contracts. It is easy to see that our framework is compatible with opcodes allowed by the Ethereum blockchain. Since the function of Verification and Payment (Algorithm 5) involves loops, which are not allowed by the Bitcoin script, our framework is not compatible with opcodes of the Bitcoin blockchain.

5. Polynomial Multiplication and Modular Exponentiation Secure Outsourcing Algorithm

5.1. Polynomial Multiplication Secure Outsourcing Algorithm. The computational complexity of traditional polynomial multiplication is $O(n^2)$, which is reduced to $O(n\log(n))$ by the FFT. In this section, we employ secure outsourcing to further reduce the local computational complexity to $O(n)$. The outsourcing is implemented in our proposed framework of blockchain-based secure computation outsourcing. The main idea of our algorithm is as follows. Firstly, the Fourier transform of the polynomial coefficients are securely outsourced. Secondly, correlation operation on the results of the Fourier transform is locally performed. Finally, the inverse Fourier transform on result of the correlation operation are securely outsourced. The specific process of our algorithm is shown as Algorithm 6.

5.1.1. Description. In Algorithm 6, the input polynomials are $f(x) = a_0 + a_1x + \cdots + a_{n-1}x^{n-1}$ and $g(x) = b_0 + b_1x + \cdots + b_{n-1}x^{n-1}$. The output is $t(x) = f(x) \times g(x) = c_0 + c_1x + \cdots + c_{2n-1}x^{2n-1}$. For convenience, polynomials are replaced with vectors of polynomial coefficients ($\mathbf{a} = [a_0, a_1, \ldots, a_{n-1}]$, $\mathbf{b} = [b_0, b_1, \ldots, b_{n-1}]$, and $\mathbf{c} = [c_0, c_1, \ldots, c_{2n-2}]$). Besides, $W_n^m = e^{-2\pi jm/n}$, in this section. We use $6p$ computational nodes in this algorithm, where p is a parameter associated with the

Input: a,b
Output: $\mathbf{c} = \mathbf{a} \times \mathbf{b}$
(1) **function** DISCRETE FOURIER TRANSFORM FOR RESERVED VECTOR (DFTRV) ($\mathbf{r}$, i)
(2) $n \longleftarrow$ the length of $\mathbf{r}$
(3) base $\longleftarrow W_n^i$
(4) $F_r[0] \longleftarrow r_i$
(5) **for** $j = 1 \longrightarrow n-1$ **do**
(6) $F_r[j] \longleftarrow F_r[j-1] *$ base
(7) **end for**
(8) **return** $\mathbf{F_r}$
(9) **end function**
(10) **function** INVERSE DISCRETE FOURIER TRANSFORM FOR RESERVED VECTOR(IDFTRV) ($\mathbf{r}$, i)
(11) $n \longleftarrow$ the length or $\mathbf{r}$
(12) base $\longleftarrow W_n^{-i}$
(13) $F_r^-[0] \longleftarrow r_i$
(14) **for** $j = 1 \longrightarrow n-1$ **do**
(15) $F_r^-[j] \longleftarrow F_r^-[j-1] *$ base
(16) **end for**
(17) **return** $\mathbf{F_r^-}$
(18) **end function**
(19) The user picks random parameters $i, j, \beta, k_1, k_2, k_3$, and generates
(20) $\mathbf{r_1} \longleftarrow L(i, k_1, n), \mathbf{r_2} \longleftarrow L(j, k_2, n), \mathbf{r_3} \longleftarrow L(\beta, k_3, 2n)$,
(21) $\mathbf{V} \longleftarrow T(\mathbf{a}, \mathbf{r_1}), \mathbf{U} \longleftarrow (\mathbf{a}, \mathbf{r_1}), \mathbf{Z} \longleftarrow T(\mathbf{b}, \mathbf{r_2}), \mathbf{S} \longleftarrow T(\mathbf{b}, \mathbf{r_2})$;
(22) The user calls TASK UPLOAD locally and uploads $\mathbf{U}, \mathbf{V}, \mathbf{Z}, \mathbf{S}$to the smart contract;
(23) The smart contract calls TASK ALLOCATION and sends vectors in $\mathbf{U}, \mathbf{V}, \mathbf{Z}, \mathbf{S}$to $4p$ nodes;
(24) The computational nodes compute $[\mathbf{F_{v_1}}, \ldots, \mathbf{F_{v_p}}]$, $[\mathbf{F_{u_1}}, \ldots, \mathbf{F_{u_p}}]$, $[\mathbf{F_{z_1}}, \ldots, \mathbf{F_{z_p}}]$, $[\mathbf{F_{s_1}}, \ldots, \mathbf{F_{s_p}}]$, and call RESULT UPLOAD;
(25) The smart contract calls RETURN RESULT;
(26) The user computes $\mathbf{F_{r_1}} \longleftarrow$ DFTRV$(\mathbf{r_1}, i), \mathbf{F_{r_2}} \longleftarrow$ DFTRV$(\mathbf{r_2}, j), \mathbf{F_{r_3}^-} \longleftarrow$ IDFTRV$(\mathbf{r_3}, \beta)$, and verifies equations (3) and (4) locally;
(27) The user computes $\mathbf{F_a} \longleftarrow \sum_{i=1}^{p} \mathbf{F_{v_i}} + \mathbf{F_{r_1}}, \mathbf{F_b} \longleftarrow \sum_{i=1}^{p} \mathbf{F_{z_i}} + \mathbf{F_{r_2}}, \mathbf{F_c} \longleftarrow \mathbf{F_a} \circ \mathbf{F_b}$;
(28) The user generates $\mathbf{D} \longleftarrow T(\mathbf{F_c}, \mathbf{r_3}), \mathbf{E} \longleftarrow T(\mathbf{F_c}, \mathbf{r_3})$, calls TASK UPLOAD locally and uploads $\mathbf{D}, \mathbf{E}$ to the smart contract;
(29) The smart contract calls TASK ALLOCATION and sends vectors in $\mathbf{D}, \mathbf{E}$ to $2p$nodes;
(30) The computational nodes compute $[\mathbf{F_{d_1}^-}, \ldots, \mathbf{F_{d_p}^-}]$, $[\mathbf{F_{e_1}^-}, \ldots, \mathbf{F_{e_p}^-}]$, and call RESULT UPLOAD;
(31) The smart contract calls RETURN RESULT;
(32) The user computes $\mathbf{c} \longleftarrow \sum_{i=1}^{p} \mathbf{F_{d_i}^-} + \mathbf{F_{r_3}^-}$, and verifies equations (5)–(7) locally;
(33) The user sends feedback to the smart contract;
(34) The smart contract calls VERIFICATION AND PAYMENT.

Algorithm 6: Secure outsourcing of polynomial multiplication.

number of computational nodes. There are six steps in the algorithm:

(1) Six parameters are picked randomly, three of which are $i, j, \beta, s.t. 0 \le n-1, 0 \le j \le n-1$ and $0 \le \beta \le 2n-2$. The other three are $k_1, k_2, k_3 \in_R Z$. We define that $L(i, k, n)$: $Z^3 \longrightarrow Z^n$ can generate one n-dimensional vector in which the i^{th} element is k, and all the other elements are zeros. We define that $T(\mathbf{v}, \mathbf{r})$: $z^{(n*2)} \longrightarrow Z^{(n*p)}$ can generate a random matrix $\mathbf{W} = [\mathbf{w_1}, \mathbf{w_2}, \ldots, \mathbf{w_p}]$ satisfying $\mathbf{v} = \sum_{i=1}^{p} \mathbf{w_i} + \mathbf{r}$. Then, the user generates $\mathbf{r_1} = L(i, k_1, n)$, $\mathbf{r_2} = L(j, k_2, n)$, $\mathbf{r_3} = L(\beta, k_3, 2n)$, $\mathbf{V} = T(\mathbf{a}, \mathbf{r_1})$, $\mathbf{U} = T(\mathbf{a}, \mathbf{r_1})$, $\mathbf{Z} = T(\mathbf{b}, \mathbf{r_2})$, and $\mathbf{S} = T(\mathbf{b}, \mathbf{r_2})$. In this way, one must know $\mathbf{r_1}$ to recover a from $\mathbf{V}$ or $\mathbf{U}$ and know $\mathbf{r_2}$ to recover $\mathbf{b}$ from $\mathbf{Z}$ or $\mathbf{S}$.

(2) The user uploads the data ($\mathbf{V}$, $\mathbf{U}$, $\mathbf{Z}$, and $\mathbf{S}$) and task requirement (i.e., FFT) to smart contract, by the function TASK UPLOAD. The smart contract calls the function TASK ALLOCATION and distributes the vectors in $\mathbf{V}$, $\mathbf{U}$, $\mathbf{Z}$, and $\mathbf{S}$ to $\mathbf{Z}$ and 4p computational nodes. After receiving the vectors, the computational nodes perform the Fourier transform on the received vectors and call the function RESULT UPLOAD. The smart contract collects the results and returns $[\mathbf{F_{v1}}, \ldots, \mathbf{F_{vp}}]$, $[\mathbf{F_{u1}}, \ldots, \mathbf{F_{up}}]$, $[\mathbf{F_{z1}}, \ldots, \mathbf{F_{zp}}]$, and $[\mathbf{F_{s1}}, \ldots, \mathbf{F_{sp}}]$ to the user by calling the function RETURN RESULT. Meanwhile, the user locally computes $\mathbf{F_{r1}}$ $\mathbf{F_{r2}}$ and $\mathbf{F_{r1}^-}$ which are Fourier transform of $\mathbf{r_1}$ and $\mathbf{r_2}$ and inverse Fourier transform of $\mathbf{r_3}$, respectively. Because there is only one nonzero coefficient in $\mathbf{r_1}$, as well as $\mathbf{r_2}$and $\mathbf{r_3}$, we simplify the DFT/IDFT, as shown in the DFTRV/IDFTRV functions in Algorithm 6.

(3) The user verifies equations (3) and (4). If they are valid, the user computes $\mathbf{F_a} = \sum_{i-1}^{p} \mathbf{F_a} + \mathbf{F_{r1}} + \mathbf{F_b}$, $\mathbf{F_b} = \sum_{i-1}^{p} \mathbf{F_{zi}} + \mathbf{F_{r2}}$, and $\mathbf{F_c} = \mathbf{F_a} \circ \mathbf{F_b}$:

$$\sum_{i-1}^{p} \mathbf{F}_{\mathbf{v_i}} \stackrel{?}{=} \sum_{i-1}^{p} \mathbf{F}_{\mathbf{u_i}}, \tag{3}$$

$$\sum_{i-1}^{p} \mathbf{F}_{\mathbf{z_i}} \stackrel{?}{=} \sum_{i-1}^{p} \mathbf{F}_{\mathbf{s_i}}. \tag{4}$$

(4) The user generates $\mathbf{D} = T(\mathbf{F}_c, \mathbf{r}_3)$ and $\mathbf{E} = T(\mathbf{F_c}, \mathbf{r}_3)$ and uploads them with the task requirement (i.e., IFFT) to smart contract, by the function TASK UPLOAD. Calling the function TASK ALLOCATION, the smart contract distributes the vectors in $\mathbf{D}$ and $\mathbf{E}$ to $2p$ computational nodes.

(5) After receiving the vectors, the computational nodes perform the inverse Fourier transform on the received vectors and return $[\mathbf{F^-_{d1}}, \ldots, \mathbf{F^-_{dp}}]$ and $[\mathbf{F^-_{e1}}, \ldots, \mathbf{F^-_{ep}}]$ to smart contract by the function TASK UPLOAD. Smart contract returns them to the user by calling the function RETURN RESULT.

(6) The user computes$\mathbf{c} = \mathbf{F^-_{rs}} + \sum_{i=1}^{p} \mathbf{F^-_{d_i}}$, selects two integers $m, l \in_R \{0, \ldots, 2n-2\}$ randomly, and verifies equations (5)–(7). If they are valid, the computing succeeds; otherwise, the computing fails. The user sends a message to the smart contract. Then, the latter calls the function VERIFICATION AND PAYMENT:

$$\sum_{i=1}^{p} \mathbf{F^-_{d_i}} \stackrel{?}{=} \sum_{i=1}^{p} \mathbf{F^-_{e_i}}, \tag{5}$$

$$\sum_{i=0}^{p} a_i b_{l-i} \stackrel{?}{=} \sum_{i=0}^{2n-2} W^{-li}_{2n-1} \mathbf{F_c}[i], \tag{6}$$

$$\sum_{i=0}^{2n-2} W^{mi}_{2n-1} c_i \stackrel{?}{=} \mathbf{F_c}[m]. \tag{7}$$

In Algorithm 6, if any verification fails, the user will report a cheating and the algorithm will come to an end.

Figure 2 demonstrates the procedures and data communications in the six steps of Algorithm 6.

5.1.2. Correctness and Complexity. Because $\sum_{i=0}^{p} \mathbf{v_i} + \mathbf{r}_1 = \mathbf{a}$ and $\sum_{i=0}^{p} \mathbf{z_i} + \mathbf{r}_2 = \mathbf{b}$ and the Fourier transform is a linear transform, we can have $\sum_{i=0}^{p} \mathbf{F}_{\mathbf{v}_1} + \mathbf{F}_{\mathbf{r}_1} = \mathbf{F_a}$ and $\sum_{i=1}^{p} \mathbf{F}_{\mathbf{z}_1} + \mathbf{F}_{\mathbf{r}_2} = \mathbf{F_b}$. We get $\mathbf{F_c} = \mathbf{F_a} \circ \mathbf{F_b}$. Because of the convolution theorem, $\mathbf{F^-_{F_c}} = \mathbf{F^-_{F_a \circ F_b}} = \mathbf{a} \times \mathbf{b} = \mathbf{c}$.

Using the DFTRV/IDFTRV in Algorithm 6, computing Fourier transform of r_1, r_2 and inverse Fourier transform of r_s needs 4_n multiplications. Because of the characteristics of $\mathbf{r}_1$,$\mathbf{r}_2$, and $\mathbf{r}_3$, only one multiplication is needed to compute each term in $\mathbf{F}_{\mathbf{r}_1}$, $\mathbf{F}_{\mathbf{r}_2}$, and $\mathbf{F^-_{rs}}$. Computing $\mathrm{F_a}$ and $\mathrm{F_b}$ needs $2pn$ additions. Computing $\mathbf{F_c}$ needs $2n$ multiplications. The verification of equations (3)–(5) takes $6(p-1)n$ additions. $6(p-1)n$ additions are needed to compute c. The final verification (equations (6) and (7)) needs $l + 2n$ multiplications. To sum up, we need $l + 8n$ multiplications and $10pn - 6n$ additions. The local complexity of multiplication in this algorithm is $O(n)$, and the local complexity of addition is $O(n)$. Therefore, the local complexity of this algorithm is $O(n)$.

5.1.3. Security against Passive Attackers. A participant may be a passive attacker. Passive attackers will follow the scripts of the algorithm while exploiting the intermediate information to breach the privacy of polynomials. In the following, we analyze the security of our algorithm against passive attackers.

The algorithm should protect the privacy of $f(x)$, $g(x)$, $\mathbf{c}$, and $\mathbf{F_c}$. As it is known to all, when all nodes collude, the passive attackers can get the most information, and the security of privacy is the lowest.

Since the operations on $f(x)$ and $g(x)$ are consistent, the risks of privacy leakage of them are the same. We analyze the security of $f(x)$ in the worst case, i e., collusion of all nodes. When all the computational nodes collude, they can guess a set of values $[a_0', a_1', \ldots, a_{n-1}']$, in which $n-1$ values are consistent with the true coefficients of $f(x)$, while one value is not. Because of $\mathbf{r}_1$, they even do not know the position of the false value. They still have to make a brute-force guessing. If $a_i \in D$, where the base of D is m, the attackers should traverse all possibilities by taking m different values for each coefficient. In this case, the attackers have to make m^n attempts to get $f(x)$. However, $a_i \in Z$ in FHEHIL. Then, $m \longrightarrow \infty$, and the attackers cannot get $f(x)$.

$\mathbf{F_c}$ and $\mathbf{c}$ are also privacy-protected. For the security of $\mathbf{F_c}$, when all the computational nodes collude, the passive attackers can guess a set of values $F_c^1[0]$, $F_c^1[1]$, $\ldots, F_c^1[2n-2]$, in which $2n-2$ values are consistent with the true coefficients of $\mathbf{F_c}$, while one value is not. However, the existence of $\mathbf{r}_3$ shows that passive attackers do not know the position of the false value. The only way to attack is by making a brute-force guessing. Same as $\mathbf{a}$ and $\mathbf{b}$, the domain of the coefficients of $\mathbf{F_c}$ is infinite. Therefore, the attackers cannot get $\mathbf{F_c}$. For the security of $\mathbf{c}$, on the one hand, the attackers cannot compute $\mathbf{c}$ using inverse Fourier transform without knowing $\mathbf{F_c}$. On the other hand, it is easy to see that when lacking $\mathbf{F^-_{ra}}$, attackers cannot get $\mathbf{c}$ which is equal to $\sum_{i=1}^{p} \mathbf{F^-_{di}} + \mathbf{F^-_{rs}}$.

5.1.4. Security against Active Attackers. A participant may also be an active attacker. Active attackers will inject false computations into the algorithm to tamper with the whole process. In the following, we analyze the security of our algorithm against active attackers.

Active attackers may return forged values to damage computing. To damage computing without being detected, attackers prefer to make minimal changes on results. In Algorithm 6, it is easy to see that the lowest risk way for computational nodes to cheat is to tamper with only one item of the results returned to the user, while the other items are correct.

There is one way to cheat in the process of securely outsourcing Fourier transform. For example, the nodes of

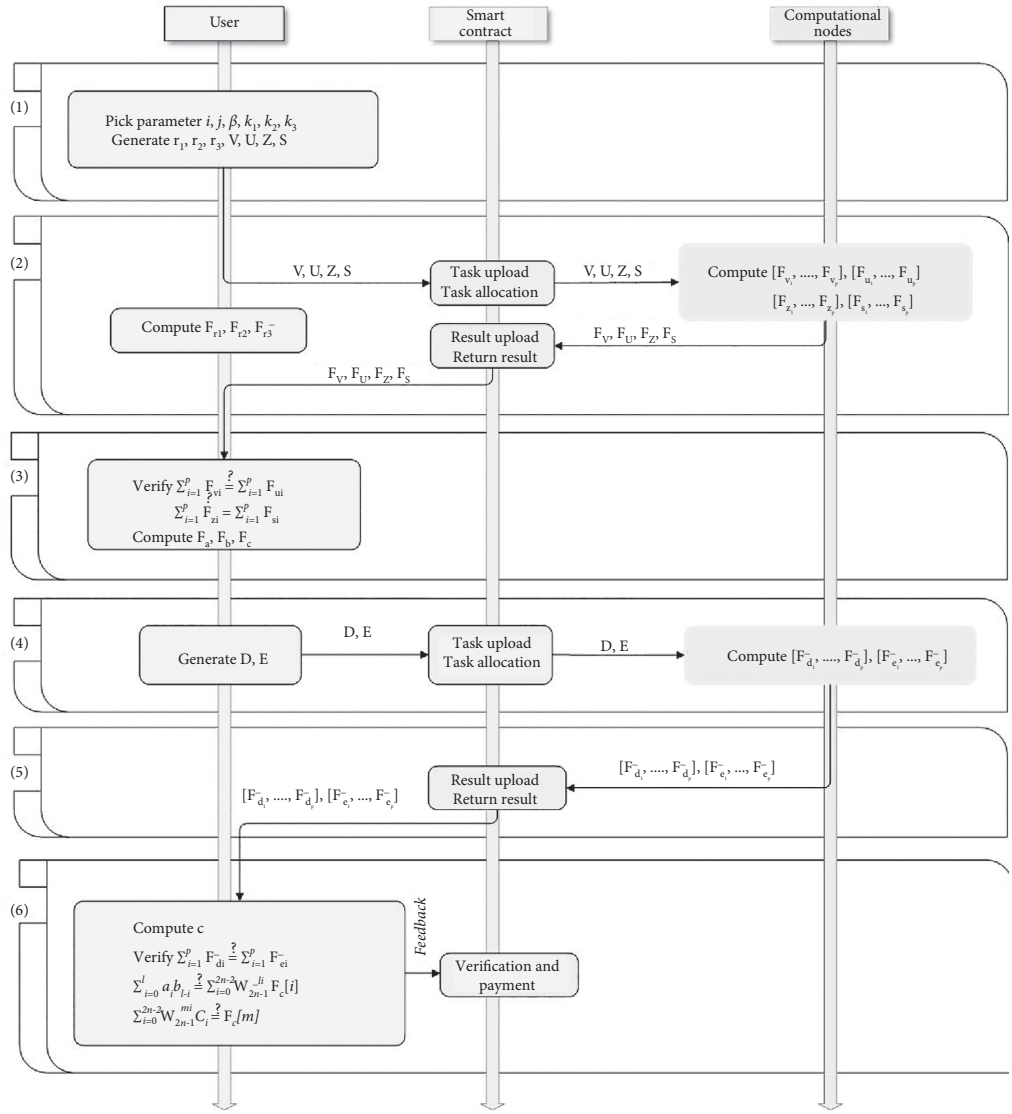


Figure 2: Blockchain-based secure outsourcing of polynomial multiplication.

computing the DFT of $f(x)$ perform honestly, while the nodes of computing the DFT of $g(x)$ do not. One node n_j changed the i^{th} term in $\mathbf{F}_{\mathbf{z}_j}$ and another node n_k' also changed the i^{th} term in $\mathbf{F}_{\mathbf{s}_k}$. This way of cheating can nullify the verification at equations (3) and (4) and damage the result of $\mathbf{F_b}$, whereas the error in the i^{th} term of $\mathbf{F_b}$ will be propagated to $F_c[i]$ and every term of c. Equation (6) computes the correct $c_1 = \sum_{i=0}^{l} a_i b_{l-i}$ and the false $c_l' = \sum_{i=0}^{2n-2} W_{2n-1}^{-li} \mathbf{F_c}[i]$. Since $\mathbf{F_c}[i]$ is false, c_l' is not equal to c_l for any random l. The verification at equation (6) can certainly detect this cheating.

There is the other way to cheat in the process of securely outsourcing the inverse DFT of $\mathbf{F_c}$. One node n_j changed the i^{th} term in $\mathbf{F}^-_{\mathbf{d}_j}$ and another node n_k' also changed the i^{th} term in $\mathbf{F}^-_{\mathbf{e}_k}$. In this way, we can nullify the verification at equation (9) and return a false item c_i. The false item c_i causes that $F_c'[m] = \sum_{i=0}^{2n-2} W_{2n-1}^{mi} c_i$ is also a false value in equation (7). $\mathbf{F_c'}[\mathbf{m}]$ will not be equal to $\mathbf{F_c}[m]$, for all $0 \le m \le 2n-2$. The verification at equation (7) can certainly detect this way of cheating.

5.2. Secure Outsourcing of Modular Exponentiation. To the secure outsourcing of modular exponentiation, we extend the algorithm in [3] and apply it to the blockchain. In our extension, six modular exponentiation pairs are outsourced to six computational nodes, instead of a single cloud, aiming to protect against possible attacks on small discrete logarithms. The process is shown as Algorithm 7. The input is two integers. The output is u^d, where u is the base and d is the exponent. In a similar way to Figure 2, Algorithm 7 can be implemented in the framework of Blockchain-based computation outsourcing, but we omit it due to limitation of space.

5.2.1. Correctness and Complexity. It is easy to prove that the algorithm is correct from equations (8) and (9). In the process of parameter generation, there are two exponentiations, two divisions, and two multiplications. Two exponentiations and six multiplications are involved during the verification. Compared with the exponentiation, the

Input: u, d
Output: u^d
(1) The user generates random parameters $g_1, g_2, e, k_1, k_2 \in Z$,
(2) and computes $v_1 \longleftarrow g_i^e, v_2 \longleftarrow g_2^e, w_1 \longleftarrow (u/g_1), w_2 \longleftarrow (u/g_2)$,
(3) $t_1 \longleftarrow d - k_1 e, l_1 \longleftarrow d - k_2 t_1$;
(4) The user uploads $(k_1, v_1), (k_1, v_2), (l_1, w_1), (k_2, w_1), (l_1, w_2), (k_2, w_2)$ to the smart contract;
(5) The smart contract distributes $(k_1, v_1), (k_1, v_2), (l_1, w_1), (k_2, w_1), (l_1, w_2), (k_2, w_2)$ to 6 computational nodes;
(6) The computational nodes compute b^a after receiving (a, b) and return results to the smart contract;
(7) The user gets $v_1^{k_1}, v_2^{k_1}, w_1^{l_1}, w_1^{k_2}, w_2^{l_1}, w_2^{k_2}$ from the smart contract;
(8) The user verifies $v_1^{k_1} w_1^{l_1} (g_1 w_1^{k_2})^{t_1} = v_2^{k_1} w_2^{l_1} (g_2 w_2^{k_2})^{t_1}$.
(9) If the verification is valid, $u^d \longleftarrow v_1^{k_1} w_1^{l_1} (g_1 w_1^{k_2})^{t_1}$

Algorithm 7: Secure outsourcing of modular exponentiation.

complexity of multiplication and division can be ignored. However, in the algorithm, the exponents are e and t_1 which are much smaller than the original exponent d through the transformation of $t_1 = d - k_1 e$. Therefore, local complexity will be greatly reduced:

$$v_1^{k_1} w_1^{l_1} \left(g_1 w_1^{k_2}\right)^{t_1} = v_1^{k_1} g_1^{t_1} w_1^{l_1} \left(w_1^{k_2}\right)^{t_1} = v_1^{k_2} g_1^{t_2} w_1^{l_1 + k_2 t_1} = g_1^d w_1^d = v^d, \tag{8}$$

$$v_2^{k_1} w_2^{l_1} \left(g_2 w_2^{k_2}\right)^{t_1} = v_2^{k_1} g_2^{t_1} w_2^{l_1} \left(w_2^{k_2}\right)^{t_1} = v_2^{k_2} g_2^{t_2} w_2^{l_1 + k_2 t_1} = g_2^d w_2^d = v^d. \tag{9}$$

5.2.2. Security. The only way to pass the verification is that the six computational nodes perform correctly. The forged results of active attackers cannot pass the verification in Step 8. The user only needs to know whether the results are correct or not, and the smart contract can detect the cheating nodes according to the records.

We analyze the security against passive attackers in the worst case, i e., the conspiring of six computational nodes. The exponents k_1, l_1, and k_2 are visible for attackers, while the other exponents t_1, d, and e are not. The bases v_1, v_2, w_1, and w_2 are visible for attackers, while g_1, g_2, and e are not. We discover that the privacy of u may leak in [3], which sends six pairs to a single node in the cloud. In [3], the base and exponent are about 1000 bit, while the parameters including g_1, g_2, e, k_1, and k_2 are only 64-bit long, to reduce the overhead of local computation. The shorter bit length of parameters may promote an easier attack on the small discrete logarithms. In this kind of attack, an attacker in the cloud can exhaust x so that $w_1^* * v_1 = w_2^* * v_2$. Then, e is breached. The attacker then exhausts g, satisfying $g^e = v_1$. Finally, the cloud can obtain u by $w_1 * g$.

We solve this attack by distributing six modular exponentiation pairs to six computational nodes, which increases the difficulty of the above attack.

6. Results and Discussion

In this section, we conduct three types of experiments. Firstly, we evaluate the efficiency of the secure outsourcing of polynomial multiplication in various numbers of polynomial multiplications and compare it with the traditional nonoutsourcing method using FFT. Secondly, we evaluate the efficiency of the secure outsourcing of polynomial multiplication by varying the numbers of items and bit length of coefficients and also compare it with the nonoutsourcing method. Finally, we complete the secure outsourcing for FHEHIL in blockchain, analyze the time consumption of each step, and compare it with the nonoutsourcing method. The experiments are simulated on two machines with Intel Core i7 processor running at 2.90 GHz and 16G memory as a cloud server and Intel Core i5 processor running at 1.80 GHz and 8G memory as a local user. The communication bandwidth is 20 Mbps.

6.1. The Evaluation of Secure Outsourcing of Polynomial Multiplication. We make experiments to evaluate the efficiency of the secure outsourcing algorithm for polynomial multiplication. We implement this experiment using Python3 language.

We compare the secure outsourcing of polynomial multiplication with the nonsourcing algorithm on time consumption with different numbers of polynomial multiplication, in which $n = 1024$, $p = 3$, and all the coefficients are 512-bit long. As demonstrated in Figure 3, it is easy to see that when the number of polynomial multiplications is less than 60, the efficiency of the outsourcing scheme is lower than the nonoutsourcing scheme due to the communication time consumption. However, when the number of polynomial multiplications increases above 60, the efficiency of the outsourcing scheme becomes higher than the nonoutsourcing scheme. When the number of polynomial multiplications is less than 300, the bottleneck of the outsourcing scheme is the time consumption on nodes' computations and interactions. When the number of polynomial

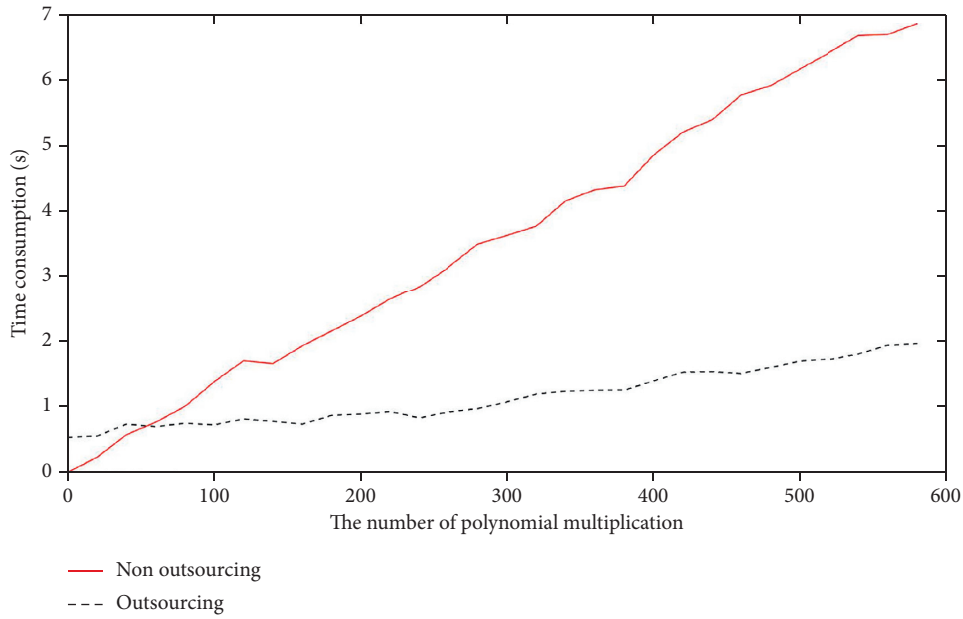


Figure 3: Comparison of time consumption on the outsourcing and nonoutsourcing scheme of polynomial multiplications.

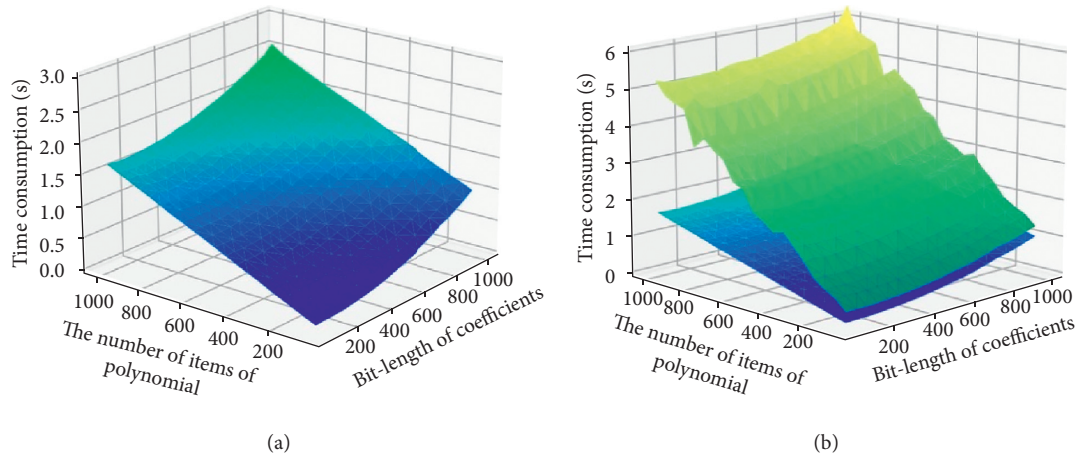


Figure 4: Comparison of time consumption on the outsourcing and nonoutsourcing scheme of polynomial multiplications.

multiplications becomes larger, the bottleneck is the time consumption on local computation.

We make another type of experiments to analyze the influence of number of terms and bit length of coefficients on the efficiency of secure outsourcing of polynomial multiplication. We count the time consumption of 400 random polynomial multiplications, with the number of polynomial items varying from 50 to 1000, and the item bit length varying from 50 to 1000. Figure 4(a) demonstrates that the time consumption increases with the increase of items of polynomials and bit length of coefficients. Moreover, the number of terms has a more obvious effect on time consumption. Besides, compared with the nonoutsourcing polynomial multiplication, our method always has a higher efficiency under all scales of data, as shown in Figure 4(b).

6.2. The Evaluation of Blockchain-Based Secure Outsourcing Scheme of Fully Homomorphic Encryption Using Hidden Ideal Lattice. We employ the relevant security parameters recommended in [2], i.e., $n = 1024$, $t = 310$, and $p = 3$. Our outsourcing scheme consists of the local user's program and the computational nodes' program. Our outsourcing scheme is compared with the nonoutsourcing scheme. The programs are written in Python3, and the smart contract based on the Ethereum platform is written in Solidity. The smart contract interacts with computational nodes' program and local program by the interface provided by Web3.

Figure 5 demonstrates the running time at all stages of the two schemes. This figure does not display the time consumption on generating parameters in the FHEHIL because that is not what we are improving. In the process of

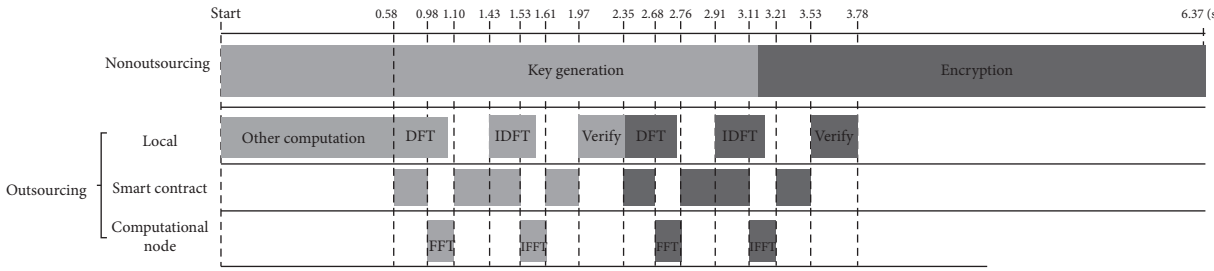


Figure 5: Time consumption on stages of the outsourcing and nonoutsourcing schemes.

Table 3: Details of time consumption in key generation.

	Verification (s)	DFTRV/IDFTRV (s)	Communication (s)	Others (s)	FFT/IFFT (s)
User	0.37	0.89	0.46	0.58	0
Smart contract	0	0	1.01	0	0
Computational nodes	0	0	0.52	0	0.19

Table 4: Details of time consumption in encryption.

	Verification (s)	DFTRV/IDFTRV (s)	Communication (s)	Others (s)	FFT/IFFT (s)
User	0.24	0.93	0.43	0.11	0
Smart contract	0	0	0.99	0	0
Computational nodes	0	0	0.37	0	0.20

computing w efficiency is slightly improved. Compared with the non-sourcing scheme, our scheme saves about 2.6 s. The overall time consumption is improved by about 40.7% (the unmarked areas in Figure 5 are the communication time consumption for interacting with the blockchain). Table 3 shows the detailed time consumption of different entities (user, smart contract, and computational nodes) in different stages (verification, communication, DFTRV/IDFTRV, FFT/IFFT, and other computations) for Key Generation. Table 4 shows the detailed time consumption of different entities in different stages for encryption.

The time consumption of decryption is not shown in Figure 5. Since there is only one polynomial multiplication, the time consumption of communication is dominant in the process of decryption, as illustrated in Figure 3. Therefore, the time consumption of outsourcing decryption (0.379 s) is larger than the nonoutsourcing decryption (0.103 s).

7. Conclusions

In this paper, we propose a secure outsourcing algorithm for polynomial multiplication that reduces the local complexity to $O(n)$. According to security analysis, our algorithm is secure against passive and active attackers. We also propose a framework for blockchain-based computation outsourcing. It has a credit-based task allocation strategy, which significantly reduces the probability of failed computations. Using this framework, we implement the secure outsourcing of FHEHIL, in which the basic computations including polynomial multiplication and modular exponentiation can be securely outsourced by our proposed algorithms. The security analysis and experimental results show that our proposed outsourcing schemes are secure and efficient. In the future, we will apply the secure outsourcing of FHEHIL into some practical secure computation problems, such as the millionaire problem, and set operation problems.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Disclosure

The conference version of this paper has been published in the 21st International Conference on Parallel and Distributed Computing, Applications, and Technologies (PDCAT 2020).

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the Key-Area Research and Development Program of Guangdong Province (No. 2020B010164003), the Science and Technology Program of Guangzhou, China (No. 201904010209), and the Science and Technology Program of Guangdong Province, China (No. 2017A010101039).

References

[1] R. Gennaro, C. Gentry, and B. Parno, "Non-interactive verifiable computing: outsourcing computation to untrusted workers," in *Proceedings of the Advances in Cryptology-CRYPTO 2010*, pp. 465–482, Santa Barbara, CA, USA, August 2010.

[2] T. Plantard, W. Susilo, and Z. Zhang, "Fully homomorphic encryption using hidden ideal lattice," *IEEE transactions on information forensics and security*, vol. 8, no. 12, pp. 2127–2137, 2013.

[3] A. Fu, S. Li, S. Yu, Y. Zhang, and Y. Sun, "Privacy-preserving composite modular exponentiation outsourcing with optimal checkability in single untrusted cloud server," *Journal of Network and Computer Applications*, vol. 118, pp. 102–112, 2018.

[4] Z. Brakerski, "Fully homomorphic encryption without modulus switching from classical gapsvp," in *Proceedings of the Annual Cryptology Conference*, pp. 868–886, Santa Barbara, CA, USA, August 2012.

[5] Z. Brakerski and V. Vaikuntanathan, "Efficient fully homomorphic encryption from (standard) LWE," *SIAM Journal on Computing*, vol. 43, no. 2, pp. 831–871, 2014.

[6] G. Craig, S. Amit, and B. Waters, "Homomorphic encryption from learning with errors: conceptually-simpler, asymptotically-faster, attribute-based," in *Proceedings of the Annual Cryptology Conference*, pp. 75–92, Santa Barbara, CA, USA, August 2013.

[7] Y. Su, B. Yang, C. Yang, and L. Tian, "Fpga-based hardware accelerator for leveled ring-lwe fully homomorphic encryption," *IEEE Access*, vol. 8, pp. 168008–168025, 2020.

[8] F. Chen, T. Xiang, and Y. Yang, "Privacy-preserving and verifiable protocols for scientific computation outsourcing to the cloud," *Journal of Parallel and Distributed Computing*, vol. 74, no. 3, pp. 2141–2151, 2014.

[9] X. Chen, W. Susilo, D. S. Wong, J. Ma, S. Tang, and Q. Tang, "Efficient algorithms for secure outsourcing of bilinear pairings," *Theoretical Computer Science*, vol. 562, pp. 112–121, 2015.

[10] B. Kang, M. Lee, and J. Park, "Efficient delegation of pairing computation," *International Association of Cryptologic Research*, vol. 259, 2005.

[11] Y. Ren, N. Ding, T.-Y. Wang, H. Lu, and D. Gu, "New algorithms for verifiable outsourcing of bilinear pairings," *Science China Information Sciences*. vol. 59, no. 9, Article ID 99103, 2016.

[12] S. Hohenberger and A. Lysyanskaya, "How to securely outsource cryptographic computations," in *Proceedings of the Theory of Cryptography Conference*, pp. 264–282, Cambridge, MA, USA, February 2005.

[13] X. Chen, L. Jin, J. Ma, Q. Tang, and W. Lou, "New algorithms for secure outsourcing of modular exponentiations,," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 9, pp. 2386–2396, 2013.

[14] Y. Ren, N. Ding, X. Zhang, H. Lu, and D. Gu, "Verifiable outsourcing algorithms for modular exponentiations with-improved checkability," in *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*, pp. 293–303, NewYork, NY, USA, May 2016.

[15] Q. Zhou, C. Tian, H. Zhang, Y. Jia, and F. Li, "How to securely outsource the extended euclidean algorithm for large-scale polynomials over finite fields," *Information Sciences*, vol. 512, pp. 641–660, 2020.

[16] D. Harvey, J. Van Der Hoeven, and G. Lecerf, "Faster polynomial multiplication over finite fields," *Journal of the Association for Computing Machinery*, vol. 63, no. 6, p. 52, 2016.

[17] D. Harvey and J. van der Hoeven, "Faster polynomial multiplication over finite fields using cyclotomic coefficient rings," *Journal of Complexity*, vol. 54, Article ID 101404, 2019.

[18] W. Liu, S. Fan, A. Khalid, C. Rafferty, and M. O'Neill, "Optimized schoolbook polynomial multiplication for compact lattice-based cryptography on fpga," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 27, no. 10, pp. 2459–2463, 2019.

[19] H. J. Hsu and M. D. Shieh, "Vlsi architecture of polynomial multiplication for bgv fully homomorphic encryption," in *Proceedings of the 2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–4, Monterey, CL, USA, October 2020.

[20] P. Giorgi, B. Grenet, and D. S. Roche, "Generic reductions for in-place polynomial multiplication," 2019.

[21] V. Nakos, "Nearly optimal sparse polynomial multiplication," 2019, https://arxiv.org/abs/1901.09355.

[22] S. Nakamoto, "Bitcoin: a peer-to-peer electronic cash system," Technical report, Springer, Berlin, Germany, 2019.

[23] C. Lin, D. He, X. Huang, X. Xie, and K. -Kwang Raymond Choo, "Blockchain-based system for secure outsourcing of bilinear pairings," *Information Sciences*, vol. 527, pp. 590–601, 2020.

[24] H. Zheng, J. Shao, and G. Wei, "Attribute-based encryption with outsourced decryption in blockchain," *Peer-to-Peer Networking and Applications*, vol. 13, no. 5, pp. 1643–1655, 2020.

[25] H. Kun, J. Xin, Z. Wang, and G. Wang, "Outsourced data integrity verification based on blockchain in untrusted environment," *World Wide Web*, vol. 43, pp. 1–24, 2020.

[26] H. Wang, X. A. Wang, W. Wang, and S. Xiao, "A basic framework of blockchain-based decentralized verifiable outsourcing," in *Proceedings of the International Conference on Intelligent Networking and Collaborative Systems*, pp. 415–421, Oita, Japan, September 2019.

[27] H. Gao, Z. Ma, S. Luo, and Z. Wang, "BFR-MPC: a blockchain-based fair and robust multi-party computation scheme," *IEEE Access*, vol. 7, pp. 110439–110450, 2019.

[28] M. Andrychowicz, S. Dziembowski, D. Malinowski, and K. Mazurek, "Secure multiparty computations on Bitcoin," *Communications of the ACM*, vol. 59, no. 4, pp. 76–84, 2016.

[29] Y. Zhang, R. H. Deng, X. Liu, and Z. Dong, "Blockchain based efficient and robust fair payment for outsourcing services in cloud computing," *Information Sciences*, vol. 462, pp. 262–277, 2018.

[30] Y. Zhang, R. H. Deng, X. Liu, and Z. Dong, "Outsourcing service fair payment based on blockchain and its applications in cloud computing," *IEEE Transactions on Services Computing*, vol. 73, p. 1, 2018.

[31] G. Craig and S. Halevi, "Implementing gentry's fully-homomorphic encryption scheme," in *Proeedings of the Annual international conference on the theory and applications of cryptographic techniques*, pp. 129–148, Tallinn, Estonia, May 2011.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 9993855, 11 pages
https://doi.org/10.1155/2021/9993855

Research Article

PMAB: A Public Mutual Audit Blockchain for Outsourced Data in Cloud Storage

Hanzhe Yang[1], **Ruidan Su**[1], **Pei Huang**[1], **Yuhan Bai**[1], **Kai Fan**[1], **Kan Yang**[2], **Hui Li**[1] and **Yintang Yang**[3]

[1]*State Key Laboratory of Integrated Service Networks, Xidian University, Xi'an 710071, China*
[2]*Department of Computer Science, University of Memphis, Memphis 38152, TN, USA*
[3]*Key Laboratory of the Ministry of Education for Wide BandGap Semiconductor Materials and Devices, Xidian University, Xi'an 710071, China*

Correspondence should be addressed to Ruidan Su; rdsu@xidian.edu.cn

Received 4 March 2021; Accepted 19 May 2021; Published 2 June 2021

Academic Editor: Qi Li

With the rapid growth of data, limited by the storage capacity, more and more IoT applications choose to outsource data to Cloud Service Providers (CSPs). But, in such scenarios, outsourced data in cloud storage can be easily corrupted and difficult to be found in time, which brings about potential security issues. Thus, Provable Data Possession (PDP) protocol has been extensively researched due to its capability of supporting efficient audit for outsourced data in cloud. However, most PDP schemes require the Third-Party Auditor (TPA) to audit data for Data Owners (DOs), which requires the TPA to be trustworthy and fair. To eliminate the TPA, we present a Public Mutual Audit Blockchain (PMAB) for outsourced data in cloud storage. We first propose an audit chain architecture based on Ouroboros and an incentive mechanism based on credit to allow CSPs to audit each other mutually with anticollusion (any CSP is not willing to help other CSPs conceal data problems). Then, we design an audit protocol to achieve public audit efficiently with low cost of audit verification. Rigorous analysis explains the security of PMAB using game theory, and performance analysis shows the efficiency of PMAB using the real-world dataset.

1. Introduction

With the rapid technological advancements in Internet of Things (IoT), more terminals and better transmission efficiency also mean that mass data is generated while providing more convenience [1]. Massive terminal data and limited storage capacity make these IoT applications have to turn to Cloud Service Providers (CSPs) to obtain professional data storage support as Data Owners (DOs). In other words, technological advancements promote the integration of cloud services and IoT. In particular, cloud services are located in the data layer of IoT and interact with application servers to provide data services [2].

However, cloud services not only provide convenience for IoT but also challenge the privacy and security of data generated by terminals [3, 4]. As the data is stored in the cloud, the Data Owner will lose the strong control over the data. CSPs may be damaged by external threats, such as hacking or natural disasters, and even they may tamper with data for their own benefit. These external and internal attacks can damage the integrity of remote data [5]. If the integrity of data cannot be audited in time, with the damaged data being used for key calculation or operation, incalculable disaster will be triggered. The remote outsourcing data audit technology can assure the data integrity with only a small amount of interaction, which can just solve the above-mentioned security problems.

In 2008, Ateniese et al. [6] first proposed a partially dynamic Provable Data Possession (PDP) protocol. As a classic remote outsourcing data audit technology, PDP later developed the characteristics of dynamic audit, batch verification, and public audit [7–11]. The traditional public audit involves the interaction between multiple parties, which leads to the trust problem. For example, centralized storage makes audit results easy to be tampered with, TPA may help CSPs conceal data problems for profit, and so on.

The problem of multiparty trust in traditional data integrity audit makes it an inevitable trend to integrate blockchain technology into data integrity audit [12]. Yue et al. [12] and Liu et al. [13], respectively, proposed the prototype of data integrity audit framework combining IoT and P2P cloud storage environment with blockchain, but its application scenarios are relatively limited. Yu et al. [14] used blockchain for audit proof storage, and the Data Owner completed the audit of data integrity by verifying the audit proof stored on blockchain. Xu et al. [15] used blockchain to arbitrate disputed audit results. Huang et al. [16] completed verification of audit tasks and record of dynamic operations through representative nodes of the consortium chain built by PBFT consensus. Lu et al. [17] used Fabric (Consortium Blockchain) to store audit records and proposed a reputation system for TPA. TPA is an entity that makes profit through audit. The remuneration paid by DOs to TPA must be less than the actual value of the audited data; otherwise, the audit will be meaningless. Therefore, TPA is easy to be bribed by the benefits (more than audit remuneration but less than data value) paid by malicious CSP. In this case, collusion attacks are difficult to avoid.

Fan et al. [18] proposed an automated audit architecture based on Ethereum (Common Blockchain), which uses smart contracts to perform audit tasks and pay related compensation. Although Common Blockchain can effectively avoid collusion attacks because of its large scale of consensus nodes and effective incentive mechanism, it is difficult to reach an acceptable execution efficiency under larger-scale audit verification. Despite the fact that Consortium Blockchain is more efficient, there still exists the nothing-at-the-attack [19]. Without an effective incentive mechanism, collusion attacks will not be well resisted. PMAB is based on Consortium Blockchain and ensures mutual supervision through effective credit-based incentive mechanism, which strengthens the supervision of CSPs while auditing data. In [18], Verifiable Delay Function (VDF) is used to realize automatic audit; that is, the system automatically generates secure random source to generate audit challenge without DOs' participation, which further reduces the cost of DOs. However, the security of the random source comes from the continuous computing power consumption, which is not efficient enough. Therefore, due to the lack of customized blockchain design for audit protocol, the existing schemes still suffer from excessive overheads and collusion attacks.

To tackle the above challenges, we propose a Public Mutual Auditing Blockchain (PMAB) for outsourced data in cloud storage to solve collusion attacks in the public audit scheme, greatly reduce the audit cost, and improve the audit efficiency. The contributions of this paper can be summarized as follows:

(i) We present a customized blockchain architecture PMAB for public audit, which enables all CSPs to automatically audit each other through audit contract and releases DOs from data audit cost

(ii) We propose a credit-based incentive mechanism to resist collusion attacks while quantifying behaviors of entities

(iii) We put forward a consensus for PMAB that combines an efficient public audit protocol. After rigorous security and performance analysis, our scheme can achieve expected security goal and audit efficiency significantly ahead of existing schemes

The outline of this paper is as follows: we first introduce the background knowledge, the system model, threat model, and design goals. In the latter, we describe the concrete constructions of PMAB and audit protocol. After that, security and performance analyses are detailed. Finally, the summary and future work of this paper are presented.

2. Preliminaries

2.1. Ouroboros. Ouroboros is a kind of blockchain consensus based on Proof of Stake (PoS), which was proposed by Kiayias et al. [20] and proved secure. It uses Publicly Verifiable Secret Sharing Scheme (PVSS) [21] to generate antibiased random numbers as random source of the representative election algorithm Follow the Satoshi (FTS), so that the candidate can be elected as the representative node with a certain probability, which is equal to the proportion of the candidate's stake to the overall stake of all candidates.

3. Problem Statement

3.1. System Model. PMAB considers a public data audit scenario for outsourced data in cloud storage, which is mainly composed of Data Owner (DO), Cloud Service Providers (CSPs), and Regulator (R) as shown in Figure 1. Audit chain and credit chain are two distributed ledgers maintained by CSPs and *R*, which, respectively, record audit information and credit of each entity. After outsourcing data to CSPs, DO (e.g., an IoT application collects data via their terminals) generates the audit contract with CSPs and *R* (Steps 1 and 2). In public audit, the audited CSP provides proof to the audit chain according to the challenge (Steps 3 and 4); then some CSPs complete audit verification and credit settlement under the supervision of *R* (Step 5). Finally, DO can obtain audit and credit settlement results through these two distributed ledgers (Step 6). The specific roles of all entities in PMAB are described as follows:

(i) **DO** has limited communication, computation, and storage resources. It outsources data to CSPs and achieves public audit with PMAB

(ii) **CSP** provides DOs with significant storage space and computation capability. It is also responsible for maintaining two distributed ledgers, while responding proof to challenges and completing public audit

(iii) **R** is also responsible for maintaining two distributed ledgers while supervising public audit process and administrating PMAB

3.2. Threat Model. PMAB considers that some corrupted CSPs will try to bribe other CSPs to conceal their data problem in audit verification. DO is honest but curious; it

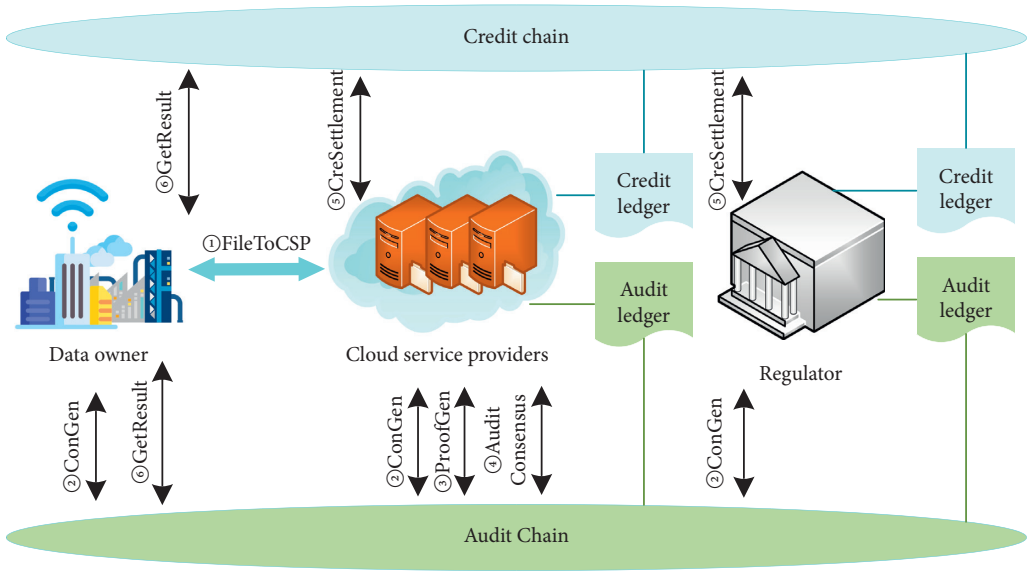


Figure 1: System model of PMAB.

will try to obtain the identity and audited outsourcing data of other DOs based on the audit information from audit chain. R is assumed to be a trustworthy regulatory agency that supervises cloud storage services.

3.3. Design Goals. To achieve secure and efficient automated data audit under the above threat model, PMAB should achieve the following goals about anticollusion, privacy preserving, efficiency, automated audit, and dynamic audit:

Anticollusion. PMAB should prevent corrupted CSPs from passing audit verification through collusion attacks

Privacy Preserving. Except for R, CSP, and DO participating in the audit contract, all other entities cannot obtain the specific identity and outsourced data information of the DO

Antiforgery. The audit proof forged by malicious CSP cannot pass the audit verification

Antireplace. For malicious CSP, when generating audit proof, it cannot use the combination of intact data block related information to get the proof of damaged data block

Efficiency. The average cost of batch audit in the audit protocol of PMAB should be limited to a very low and constant level, and the overall verification and the consensus time of PMAB should be controlled within a limited time

Automated Audit. PMAB should achieve automatic audit periodically based on audit contracts

Dynamic Audit. The remote data that is modified dynamically can be audited timely and effectively

4. Public Mutual Audit Blockchain

4.1. Design Overview of PMAB. As the analysis above, all public audit schemes based on blockchain cannot audit efficiently and resist collusion attacks at the same time.

In PMAB, we innovatively use the mutual audit between CSPs instead of TPA's audit. According to the game theory, we design an incentive mechanism based on credit, so that no CSP is willing to help other CSPs conceal data problems. Furthermore, based on Ouroboros [20], we design an audit protocol that combines with the blockchain consensus to efficiently and automatically complete public audits.

Therefore, the description of PMAB is mainly divided into two parts: basic blockchain structure and audit protocol.

4.2. Basic Blockchain Structure of PMAB. Blockchain architecture is the basic design of PMAB, which is mainly composed of two parts, namely, credit chain and audit chain. In this part, we will introduce the core of credit chain (i.e., the incentive mechanism) and the basic data structure in audit chain.

4.2.1. Incentive Mechanism. Incentive mechanism is the power source and security cornerstone of blockchain system. The credit value credit is the core of PMAB incentive mechanism, which mainly comes from CSPs' initial Credit, deposit, and audit remuneration reward paid by DOs. The candidate node with higher credit is more likely to be elected as a representative node. Moreover, the credit lost by collusion will outweigh the creditgained, and rational CSPs will conduct honest audits to maximize benefits. Some key concepts related to creditare described below:

(i) *initial Credit*. When each CSP joins PMAB, it needs to pay some deposits in exchange for *initial Credit*, which will be confiscated when the malicious behavior of this CSP is found. Only when initial Credit reaches the threshold can it become a candidate node.

(ii) *deposit*. When the audit contract is constructed, the CSP needs to mortgage *deposit*, of which *dataValue* and *penalty* are equal in half.

(iii) *dataValue*. As the compensation that CSP pays to DO when audit fails, it represents the value of data.

(iv) *penalty*. As a fine for the malicious behavior of CSP when being audited.

(v) *bonusPool*. All forfeited *initialCredit* and penalty will be put into *bonusPool*, and the honest CSPs participating in the audit will divide up *bonusPool*.

4.2.2. Block and AuditContract. *Block* and *AuditContract* are basic data structure in audit chain. *Block* stores the contents and results of each audit. *AuditContract* keeps the specific information of each audit task. The whole contract is stored in *R*, the associated DO, and CSP, all CSPs just keep *conHeader*, which determines the audit task information of each consensus.

The structures of *Block* and *AuditContract* are shown in Figures 2(a) and 2(b). Descriptions of key fields are as follows:

(i) *nonce*. A random value obtained by PVSS [21] in Ouroboros [20] is used as random source for this audit

(ii) *proof*. All audit proofs collected by the representative node during the audit

(iii) *auditCons*. The collection of audit tasks covered in this audit

(iv) *verResult*. The results of this audit

(v) *ConPk*. The public key to be used in the audit verification

(vi) *auditRate*. The proportion of audited data to outsourced data

(vii) *Rproof*. A proof of the overall stored data provided by *R*

4.3. High Description of Audit Protocol. This part focuses on the audit protocol of PMAB, which is divided into Setup phase and Audit phase. There are system parameters initialization and audit preparation in Setup phase. Audit phase includes the generation and verification of audit proof, as well as credit settlement. In addition, in order to verify the remote data of dynamic operation in time, PMAB supports dynamic audit.

4.3.1. Setup Phase. In this phase, the system parameters are first initialized in *KeyGen*. Then CSPs join PMAB in *SystemIni*. After DO preprocesses files which will be outsourced and uploads them to CSP in FileToCSP, the *AuditContract* is constructed by DO, CSP, and *R* in *AuditConGen*.

KeyGen. With a security parameter λ, two elliptic curve groups G_1 and G_2 and a multiplicative group G_T of the large prime order p, a bilinear pairing e: $G_1 \times G_2 \longrightarrow G_T$, a field Z_p of residue classes modulo p, two random generators $g_1 \in G_1$ and $g_2 \in G_2$, a pseudorandom permutation (PRP) $\pi(\cdot)$, and a pseudorandom function (PRF) $f(\cdot)$ are picked.

$$S\mathbb{P} = \{G_1, G_2, G_T, g_1, g_2, e, \pi, f\}. \quad (1)$$

Furthermore, for the convenience of expression, $\sigma_\epsilon(\cdot)$ is used to represent a signature signed by entity ϵ and ID_ϵ is used to represent the unique identicator of entity ϵ.

SystemIni. After the new CSP exchanges *initialCredit* (Icr) from *R*, *R* broadcasts new node access notification NAN = $(t, ID_{CSP}, CSP_{\text{address}}, Icr, \sigma_R(\text{NAN}))$ to all CSP nodes, where t is the timestamp and CSP_{address} is the network address of new CSP. After receiving the NAN, other CSP nodes establish the connection with new node.

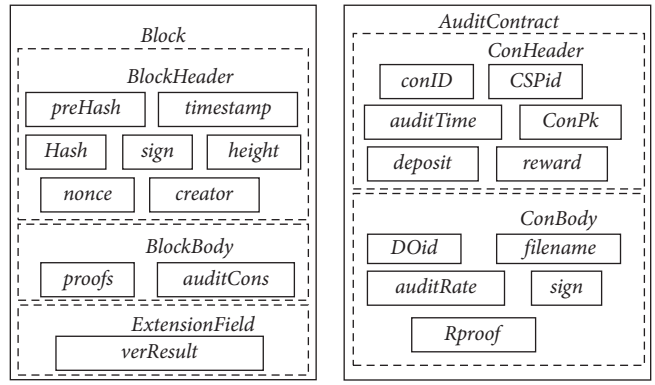
FileToCSP. Assuming that the file that DO needs to store is*F*, DO divides *F*into following data blocks:$F = \{m_1, m_2, \ldots, m_i, \ldots, m_n\}, i \in [1, n], m_i \in Z_p$. DO generates a random parameter $\omega_F \in Z_p^*$ for *F*, thereby obtaining the verification random number set $R_F = \{r_i\}, i \in [1, n]$, where $r_i = f_{\omega_F}(i)$. Then $\text{sk} = \alpha \in Z_p^*$ is randomly selected as audit private key; thereby BLS homomorphic verification tags $\sigma_i = (g_1^{(m_i + r_i)})^\alpha$ for each data block m_i are generated, thereby obtaining a tag set $\sigma = \{\sigma_i\}, i \in [1, n]$. Finally, DO sends a tag collection message $\text{TC} = \{t, F, \sigma, \sigma_{DO}(\text{TC})\}$ to the CSP that stores outsourced data.

AuditConGen. Assuming that DO and CSP have negotiated *deposit*, *auditRate*, *auditTime*, and other information for *AuditContract*, DO generates audit public key $ConPk = g_2^\alpha$ and sends R_F to R. After *R* generates$Rproof = g_1^{(r_1 + r_2 + \ldots + r_n)}$, CSP fills all information in the *AuditContract*, especially$sign = \sigma_{CSP|DO|R}(AuditContract)$. Then DO can delete *F* and σ locally. After receiving $con = (t, ID_R, ConHeader, \sigma_R(con))$, each CSP stores *ConHeader* in local contract collection Con.

4.3.2. Audit Phase. This part mainly focuses on the detailed process of data audit. Before each round of audit, the CSP whose *initialCredit* reaches the threshold will participate in representative election as a candidate. After PVSS and FTS [17, 18], we get random source *Random* and a representative *Rep*, and other candidates become endorsers Endo. After the audited CSP obtains the audit proof *P* through *ProofGen* based on *Random*, PMAB completes the audit consensus and appends *Block* to audit chain in *AuditConsensus*. After *CreSettlement*, audit result *verResult* is added to audit chain and *credit* is updated to credit chain. Finally, DO can obtain audit results related to itself from audit chain. For ease of understanding, the following description is based on a scenario, where a CSP is audited by multiple DOs.

ProofGen. After obtaining the random Source *Random*, each CSP checks whether it needs to be audited this *round based on local contract collection Con*. If it does, the corresponding challenge set *chal* will be calculated according to *Random*, and the audit proof set *P* will be generated.

CSP first generates two keys $k_1 = f_{\text{Random}}(\text{height})$ and$k_2 = f_{\text{Random}}(\text{height} + 1)$. The audit contract set



(a) (b)

Figure 2: The structure of *Block* and *AuditContract*. (a) The structure of *Block*. (b) The structure of *AuditContract*.

$ConCache = \{Contract_j\}_{j\in[1,K]}$ that needs to be executed by the CSP in this round is generated, where K represents the number of audit contracts in *ConCache*. According to $auditRate_j$ of $Contract_j$ in *ConCache* and the actual size n_j of the audited file, the number of challenged blocks for this audit contract z_j is computed as $z_j = \lceil auditRate_j \times n_j \rceil$. Furthermore, the challenge set of current round $Chal = \{chal_j\}_{j\in[1,K]}$ of the CSP is obtained, where $chal_j = \{i_l, v_l\}, i_l = \pi_{k_1}(l), v_l = f_{k_2}(l), l \in [1, z_j]$. CSP then calculates the tag proof $TP_j = \prod_{chal_j} \sigma_{i_l}^{v_l}$ and the data block proof $DP_j = g_1^{\sum_{chal_j} m_{i_l}\cdot v_l}$ corresponding to each $chal_j$, thereby forming the tag proof set $\Phi = \{\prod_{\text{Chal}} \text{TP}_j\}$ and the data block proof set $\mu = \{DP_j\}_{j\in[1,K]}$. Finally, the proof set $P = \{\Phi, \mu\}$ of the CSP is obtained.

AuditConsensus. As shown in Figure 3 an audit consensus is conducted after *ProofGen*. First, the audited CSP sends its own proof message $proof = \{t, P, \sigma_{CSP}(proof)\}$ to *Rep* and Endo nodes. *Rep* packs received *proof* into a message $proofs = \{t, \{P\}, \sigma_{\text{Re}p}(proofs)\}$ and broadcasts it to all *Endo* nodes, where N represents the number of nodes that need to execute audit contracts. After receiving*proofs*, Endo nodes compare it to *proof* they received; if *proof* is included in *proofs*, they will pack the message $response = \{t, proofs, \sigma_{En\ do}(response)\}$ and send it to *Rep*. After collecting allresponse, *Rep* packs the message $Rproof\ \text{Re}quest = \{t, \{response_s\}_{s\in[1,N]}, \sigma_{CSP_{\text{Re}p}}(Rproof\ \text{Re}quest)\}$ and sends it to *R*. To verify $Rproof Request$, *R* compares *proofs* of all response in $Rproof Request$. If they are the same, *R* will calculate the random number proof $proof_R = \{\xi_s\}_{s\in[1,N]}$ required for this round of proof consensus, where $\xi_s = \left\{RP_j = g_1^{\sum_{chal_j} r_{i_l}\cdot v_l}\right\}_{j\in[1,K]}$, and then package message $Rproof Response = \{t, proof_R, \sigma_R(Rproof Response)\}$ and send it to *Rep*.

Then, *Rep* fills $Rproof Request|Rproof Response$ in *proofs*, *ConCache* in *auditCons*, *Random* in nonce, and so on while generating a new *Block*.

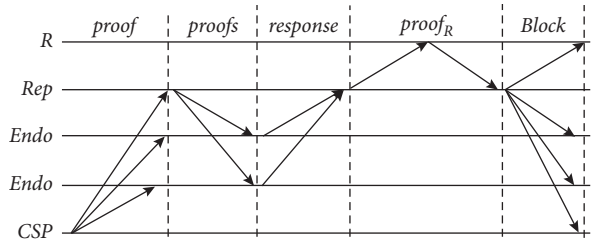


Figure 3: The process for generating new *Block*.

Finally, *Block* is broadcast to all CSPs and *R*.

CreSettlement. After *AuditConsensus*, all *Endo* and *Rep* nodes verify the proof P to audit the outsourced data. The verification operation is to verify whether the following equation holds for audited CSP.

$$e(\Phi, g_2) = \prod_{ConCache} e\left(DP_j \cdot RP_j, ConPk_j\right). \quad (2)$$

If it holds, $Ver = true$; otherwise, $Ver = false$. Then message $verify = \{t, Ver, \sigma_{Rep/En\ do}(verify)\}$ is sent to *R*.

After receiving all $verify$, *R* verifies whether there are different verification results. If all $verify$ are the same, verification result set *RVer* and the malicious nodes set *Mal* will be empty (i.e, all *Endo* and *Rep* are honest); otherwise, *R* will use equation (2) to further verify proofs for the dispute and then get $RVer = \{verify_s\}_{s\in[1,A]}$ and $Mal = \{ID_{Endo_s/Rep_s}\}_{s\in[1,M]}$, where A represents the number of disputed proofs and Mrepresents the number of malicious nodes. After receiving $ack = \{\{Ver_s\}_{s\in[1,N]}, RVer, Mal, t, \sigma_R(ack)\}$ from *R*, all nodes put it into *verResult* of the corresponding *Block* in audit chain. Then all CSPs and *R* will conduct credit settlement based onack. First, the total reward *totalReward* of all executed audit contracts in *ConCache* is calculated and put in *bonusPool*, and the DO in the audit contract is compensated for data corruption. If *dataValue*, *Mal* is not empty, all *initialCredits* of the CSP and *penalty* in the corresponding audit contract involved are put into *bonusPool*. Finally, *credit* in *bonusPool* will be

obtained by virtuous *Endo* and *Rep* (*Rep* can get an extra part).

To prove the correctness of audit process, equation (2) can be derived as follows:

$$\begin{aligned} e(\Phi, g_2) &= e\left(\prod_{Chal}\prod_{Chal_J}\left(\left(g_1^{\left(m_{i_l}+r_{i_l}\right)}\right)^{\alpha_j}\right)^{v_l}, g_2\right) \\ &= \prod_{ConCache} e\left(\prod_{chal_j}\left(g_1^{\left(m_{i_l}+r_{i_l}\right)}\right)^{v_l}, g_2^{\alpha_j}\right) \\ &= \prod_{ConCache} e\left(g^{\sum_{chal_j} m_{i_l}\cdot v_l + \sum_{chal_j} r_{i_l}\cdot v_l}, ConPk_j\right) \\ &= \prod_{ConCache} e\left(DP_j \cdot RP_j, ConPk_j\right). \end{aligned} \tag{3}$$

4.3.3. Dynamic Audit. EMAB supports dynamic auditing; that is, it supports DO in auditing data after dynamical operations, which mainly consist of insertion, modification, and deletion. The details are described below.

(i) *Insertion*. Suppose that DO wants to insert a data block m_j in file F, j is the index position to be inserted, and $1 \le j \le n+1$, where $n$represents the number of data blocks of origin F. DO updates the local FIT data (the auxiliary data linked list corresponding to file F) and inserts the new node $(B_j = n+1, r_j = f_{\omega_F}(n+1))$ into the $j-$th position of FIT. DO calculates the BLS-HVT $\sigma_j = (g_1^{(m_j+r_j)})^{\alpha}$ of m_j and sends the message $insert = \{t, j, m_j, \sigma_j, \sigma_{DO}(insert)\}$ to the CSP to help update m_j and σ_j. The message $up\ da\ teRproof = \{t, j, r_j, \sigma_{DO}(up\ da\ te\ Rproof)\}$ is sent to R to help update R_F.

(ii) *Modification*. Suppose that DO wants to update the data block m_j in file F. The message $up\ da\ te = \{t, j, m_j, \sigma_j, \sigma_{DO}(update)\}$ is sent to the CSP to help it update the data block m_j and BLS-HVT σ_j, where $\sigma_j = (g_1^{m_j+r_j})^{\alpha}$.

(iii) *Deletion*. Suppose that DO wants to delete data block m_j in file F. DO moves the $j-$th node in F's FIT to the end of the chain and sets its B_j to -1. The message $de\ lete = \{t, j, \sigma_{DO}(delete)\}$ is sent to CSP to help it delete the corresponding data block m_j and BLS-HVT σ_j. The message $de\ leteRproof = \{t, j, \sigma_{DO}(deleteRproof)\}$ is sent to R to help it delete the corresponding random number element r_j.

All dynamic operation records are stored in the dynamic operation domain of corresponding audit contract after all participants sign. In the following audit consensus, all new data will be applied.

5. Security Analysis

In this part, we mainly analyze anticollusion, privacy preserving, antiforgery, and antireplace described in Section 3.3.

5.1. Anticollusion. PMAB can resist collusion attacks from consensus nodes (*Rep* and *Endo*). Colluding nodes cannot deceive R by sending wrong verify to bypass corrupted data blocks. They definitely betray each other because honest behavior is more profitable than collusion.

Because consensus nodes will send $verify$ to R at the same time in each round of audit consensus, this process can be regarded as a static and complete information game. For simplicity, we take two consensus nodes, namely, $player_1$ and $player_2$, for example. Suppose that v_1 and v_2 are $dataValue$ of $player_1$ and $player_2$, respectively, p_1 and p_2 are *initialCredit* (and $penalty$) of $player_1$ and $player_2$ $(v_1 < p_1, v_1 < p_2, v_2 < p_1, v_2 < p_1)$, m, $(0 < m < v_2)$ is the cost of bribery, and u is the reward of audit.

The game elements are as follows:

(i) $players\{player_1, player_2\}$

(ii) $strategy\{honest, malicious\}$

(iii) $utility$ Profit matrix when both $players$ have data problems and only $player_2$ has data problems as Tables 1 and 2 show, respectively

When both players have data problems, for $player_2$, it is easy to know $u - v_2 > u - p_2 - v_2$ and $p_1 + u - v_2 > u$, so honest must be the dominant strategy of $player_2$. Similarly, it is easy to know that for, $player_1$, *honest* is also the dominant strategy. So, the Nash equilibrium point in this case falls in case (*honest*, *honest*). Therefore, in this case, no collusion problem occurs. When only $player_2$ has data problems, for $player_2$, it is easy to know $u - v_2 > u - p_2 - v_2$ and $p_1 + u - v_2 > u - m$, so *honest* must be the dominant strategy of $player_2$. For $player_1$, it is easy to know $u > u - p_1$ and $P_2 + u > u + m$, so *honest* must be the dominant strategy of $player_1$. So the Nash equilibrium point in this case falls in case (*honest*, *honest*). Therefore, in this case, no collusion problem occurs.

The situation of more players is similar to the situation of two players. In summary, PMAB can avoid collusion problems.

5.2. Privacy Preserving. Apart from R and audited CSP, all other CSPs cannot obtain the relationship between audit tasks and DOs from *Con*, and the specific data block information from P, that is, PMAB, can protect DO's identity privacy and data privacy.

Identity Privacy Protection. All *ConHeader* are stored in CSPs' local contract collection *Con*. Only the audit public key *ConPk* in *ConHeader* is associated with DO, but *ConPk* of each *ConHeader* of DO can be different. If there is no duplicate *ConPk*, it is impossible to get the association between *ConPk* and DO. So the privacy of the DO's identity is protected.

Table 1: Profit matrix when both *players* have data problems.

		$player_2$	
		honest	*malicious*
$player_1$	*honest*	$u - v_1, u - v_2$	$p_2 + u - v_1, u - p_2 - v_2$
	malicious	$u - p_1 - v_1, p_1 + u - v_2$	u, u

Table 2: Profit matrix when only $player_2$ has data problems.

		$player_2$	
		honest	*malicious*
$player_1$	*honest*	$u, u - v_2,$	$p_2 + u, u - p_2 - v_2$
	malicious	$u - p_1, p_1 + u - v_2$	$u + m, u - m$

Data Privacy Protection. In the audit consensus, it is difficult for the consensus nodes to obtain $\sum_{chal_j} m_{i_l} \cdot v_l$ from $DP_j = g_1^{\sum_{chal_j} m_{i_l} \cdot v_l}$ because of DLP. Moreover, even if $\sum_{chal_j} m_{i_l} \cdot v_l$ is given, the specific information of m_{i_l} cannot be solved out without knowing the number of $m_{i_l} \cdot v_l$. Therefore, PMAB can ensure that consensus nodes cannot obtain the data information of the audit data during the verification process, which protects data privacy.

5.3. Antiforgery. If the data block m_i within *chal* has been modified to $m_i + off_i$ by the CSP, where off_i denotes the modification part, to adapt the new DP^*, a new TP^* should be computed as follows:

$$\begin{aligned} TP^* &= \prod_{chal} \left(\left(g_1^{m_{i_l} + r_{i_l} + off_{i_l}} \right)^{sk} \right)^{v_l} \\ &= \prod_{chal} \left(\left(g_1^{m_{i_l} + r_{i_l}} \right)^{sk} \cdot \left(g_1^{off_{i_l}} \right)^{sk} \right)^{v_l} \\ &= \prod_{chal} \sigma_{i_l}^{v_l} \cdot g_1^{sk \cdot \sum_{chal} off_{i_l} \cdot v_l} \\ &= TP \cdot g_1^{sk \cdot \sum_{chal} off_{i_l} \cdot v_l}. \end{aligned} \tag{4}$$

Because this CSP only owns TP, it needs to know sk for obtaining TP^*. However, *sk* is a private key of the DO. In our assumption, *sk* cannot be obtained by others. Hence, the audit proof cannot be forged by a CSP, and PMAB can resist forgery attack.

5.4. Antireplace. Suppose that a corrupted data block m_j has been checked, and two data blocks m_{j_1} and m_{j_2} are intact. To obtain the HVT of m_j, the correct combination of σ_{j_1} and σ_{j_2} should be found. Since $\sigma_{j_1} = (g_1^{(m_{j_1} + r_{j_1})a})^{sk}$, $\sigma_{j_2} = (g_1^{(m_{j_2} + r_{j_2})})^{sk}$, a CSP sets that

$$\begin{aligned} \sigma_j^* &= \sigma_{j_1}^{\alpha_{j_1}} \cdot \sigma_{j_2}^{\alpha_{j_2}} \\ &= \left(\left(g_1^{\left(m_{j_1} + r_{j_1} \right)} \right)^{sk} \right)^{\alpha_{j_1}} \cdot \left(\left(g_1^{\left(m_{j_2} + r_{j_2} \right)} \right)^{sk} \right)^{\alpha_{j_2}} \\ &= \left(g_1^{\alpha_{j_1} \cdot \left(m_{j_1} + r_{j_1} \right) + \alpha_{j_2} \cdot \left(m_{j_2} + r_{j_2} \right)} \right)^{sk}, \end{aligned} \tag{5}$$

where $\alpha_{j_1}, \alpha_{j_2} \in Z_p$. If σ_j^* is to be equal to σ_j, $\alpha_{j_1} \cdot (m_{j_1} + r_{j_1}) + \alpha_{j_2} \cdot (m_{j_2} + r_{j_2}) = m_j + r_j$ must be satisfied. In order to meet this requirement, r_{j_1}, r_{j_2}, and r_j must be known. But CSP cannot get them based on the information it already has. For example, if g_1, m_j, and $(g_1^{(m_j + r_j)})^{sk}$ are known, it is required to solve r_j. r_j is unknown, so $r_j + m_j$ is also unknown. If g_1 is given, solving $r_j + m_j$ from $(g_1^{(m_j + r_j)})^{sk}$ is a DLP problem. So in polynomial time the probability of solving r_j is negligible.

Similarly, solving r_{j_1} and r_{j_2} is the same as solving r_j. So replace attack from CSP can be resisted in EMAB.

6. Performance Analysis

In this part, we focus on the theoretical and experimental analyses of PMAB's performance through comparing them with similar schemes: Dredas [18], Fabric [17], and CAB [16]. The notations used in the performance analysis are shown in Table 3.

6.1. Theoretical Analysis. The comparison of entities' computation cost with Dredas [18], Fabric [17], and CAB [16], also supporting public audit by blockchain, is shown in Table 4.

Computation overheads are mainly distributed in *FileToCSP*, *ProofGen*, and *AuditConsensus* in the comparison. In order to provide the reference for comparison, we test 1000 times and then obtain the average cost of each operation; that is, $H = 18.98\,ms, E = 9.64\,ms, P = 4.63\,ms$, and $M = 2.51\,ms$. From Table 4, we can see that the DO and consensus node in the PMAB cost much less. Firstly, in *FileToCSP*, DO generates tags for all data blocks to be uploaded. In this part, compared with all other schemes, $nH + nM$ operations are avoided in PMAB. Then, in *ProofGen*, audited CSP computes challenges and corresponding proofs. In this part, $zM - E$ operations are avoided in PMAB, compared with other fastest schemes. Finally, in *AuditConsensus*, the smart contract in Dredas [18], the TPA in Fabric [17], or each consensus node in CAB [16] and PMAB verifies the correctness of proofs. Proof verification is the core part of public audit, and it is also the efficiency bottleneck of the whole public audit. In this part, the verification cost of Fabric and CAB increases linearly, while the verification cost of Dredas [18] and PMAB remains at a

Table 3: Notation definitions of performance analysis.

Notation	Description
H	Hash function mapping a string to a point on G_1 and G_2.
E	Modular exponentiation on G_1 and G_2.
P	Bilinear pairing operation of e.
M	Point multiplication on G_1 and G_2.
n	The total number of data blocks outsourced.
z	The number of challenged data blocks.

Table 4: Comparison of computation cost.

Schemes	DO	CSP	TPA	Consensus node
Dredas	$nH + 2nE + nM$	$zH + (2z + 1)E + 2(z - 1)M$	—	$2E + 2P + 2M$
Fabric	$nH + 2nE + nM$	$zE + (2z - 1)M$	$2P + (z + 1)E + zM$	—
CAB	$nH + 2nE + nM$	$zE + (2z - 1)M$	—	$zH + (z + 1)E + 3P + zM$
PMAB	$2nE$	$(z + 1)E + (z - 1)M$	—	$2P + M$

lower and constant level, and furthermore PMAB avoids $2E + M$ operations compared with Dredas [18].

6.2. Experimental Analysis. We evaluate performance of PMAB by conducting several experiments using JDK 1.8 on Ubuntu 16.04 system equipped with Intel Core i5-8400 CPU at 2.3 GHz and 4 GB RAM. We also use Docker to virtualize different nodes. WebSocket and Netty are used for TCP and HTTP communication, respectively. All pairing operations and related calculations on an elliptic curve are implemented with JPBC library v2.0.0 and type A pairing parameters, in which the group order is set to 160 bits and the base field order is 512 bits. The signature algorithm is implemented by the identity-based signature in [22] with JPBC library. The hash algorithm implemented is SHA-512 in BouncyCastle library. The encryption and decryption algorithm uses RSA-1024 in the security library JCE (Java Cryptography Extension) of Java. The test datasets stem from China-Brazil Earth Resources Satellite (CBERS) on Amazon Web Service (AWS). The image files in CBERS are converted to Cloud Optimized GeoTIFF format in order to optimize its use for cloud-based applications. Each test file is divided into 10,000 4 KB data blocks. According to LFT (Loss Function Theory) presented in [12], the optimal balance between the high detection probability and the low verification cost can be achieved by challenging a limited number of data blocks. Therefore, the sample size of data blocks in our experiments is changed from 50 to 500.

FileToCSPTime. Figure 4 shows the computation cost of DO in *FileToCSP*. With the sample size increasing, it is obvious that the growth rate of *FileToCSP*'s computation cost in PMAB (0.91 s~8.85 s) is less than half as much as other schemes (1.85 s~18.21 s).

ProofGenTime. Figure 5 shows audited CSP's computation cost during generating audit proof. Dredas [18] takes significantly more time (1.81 s~18.04 s) because there are many heavy operations such as H and E on G in the proof

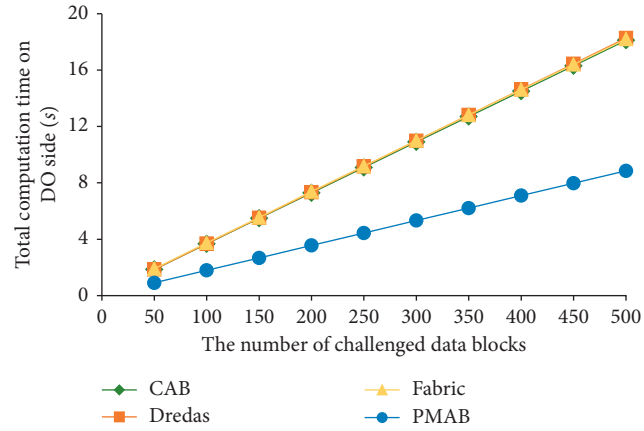


Figure 4: The computation cost comparison in *FileToCSP*.

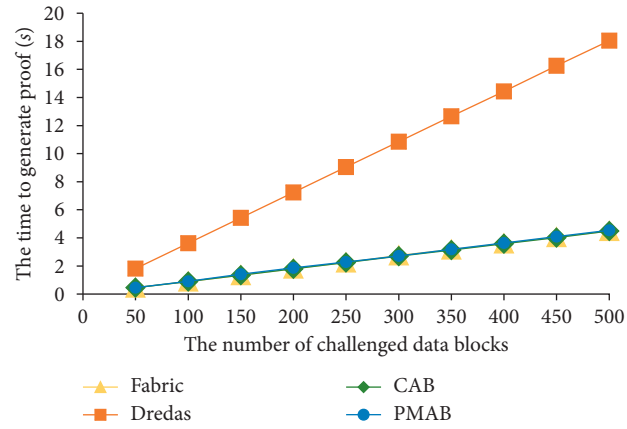


Figure 5: The computation cost comparison in *ProofGenTime*.

generation, while the proof generation times of PMAB and the remaining schemes are almost the same (0.45 s~4.54 s).

AuditVerifyTime. Figure 6 shows the verification time of the TPA in Fabric [17] and the consensus node in Dredas [18], CAB [16], and PMAB spend, respectively, where the

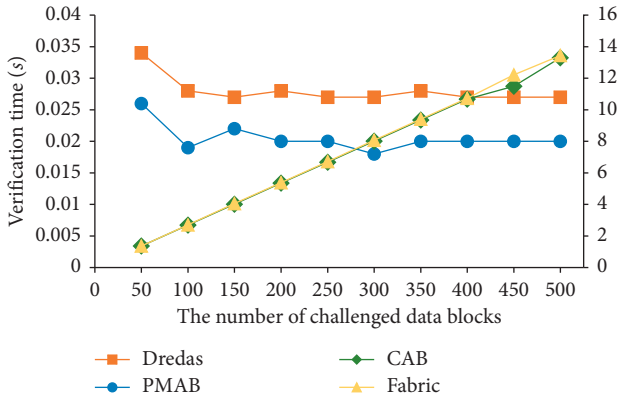


Figure 6: The verification cost comparison in *AuditVerifyTime*.

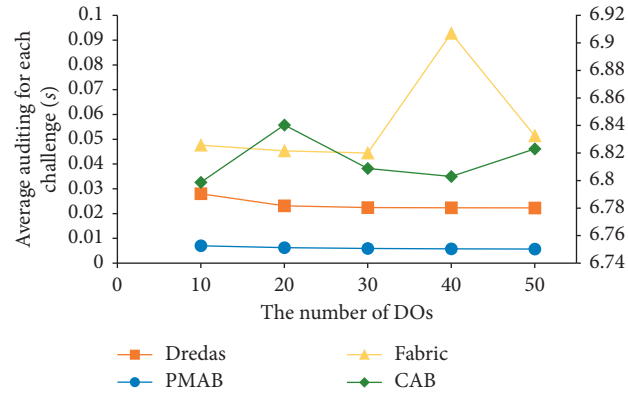


Figure 7: The computation cost comparison in *BatchAuditTime*.

verification time of Dredas [18] and PMAB ranges from 0.019 s to 0.034 s and the verification time of Fabric [17] and CAB [16] ranges from 1.36 s to 13.44 s. It is obvious that there is a linear relationship between the number of challenged blocks and the verification time of Fabric [17] and CAB [16], while the verification times of Dredas [18] and PMAB tend to be 27 ms and 20 ms, respectively. Thanks to less expensive operations such as H and E, our verification cost is more acceptable to each involved verifier compared with other schemes.

BatchAuditTime. In Figure 7, we compare the batch auditing with Dredas [18], Fabric [17], and CAB [16] under the condition that each DO challenges the same CSP with 250 data blocks in a challenge set. The average audit time of Dredas [18] and PMAB ranges from 0.007 s to 0.028 s, and the average audit time of Fabric [17] and CAB [16] ranges from 6.679 s to 6.83 s. It is obvious that, with the increase in the number of aggregated audit tasks, the average audit time cost of Fabric [17] and CAB [16] fluctuates between 6.8 s and 6.9 s, while Dredas [18] and PMAB tend to 0.02 s and 0.005 s, respectively. PMAB is four times faster than the fastest of other schemes in average audit time of the batch audit. Considering that the single validation time of PMAB in Figure 6 is only one-third faster than that of Dredas [18], our batch audit efficiency is higher.

RandomGenTime. In each consensus of the blockchain, a random source will be obtained to complete the current round of audit tasks, that is, automatic audit. The random sources in Dredas [18] are generated by a verifiable random function (VRF). In order to ensure the freshness and security of the generated random source, CSP needs to execute VRF by taking the nonce of the new Ethereum block as a seed, until the block is fully confirmed by the Ethereum network; that is, the block cannot be tampered with afterwards. Then the VRF is terminated, and the corresponding random source is obtained. The smart contract verifies the validity of the random source before using it. However, the validation process can be divided into K parallel tasks by using K process states provided by CSP, so the validation time is $(1/K)$ of the generation time. According to [23], the average time to generate a new block in Ethereum is 14 s. Generally, eight blocks are generated before the block is confirmed by the network, so it takes at least 112 s to get the random source. Assuming $K = 100$, that is, there are 100 parallel verification processes, the verification takes nearly 2 s. Therefore, the random source generation time of Dredas [18] is constant at 114 s. In the random source generation process of our scheme, each consensus node has to send and process $2(n-1)$ messages (n is the number of consensus nodes) and do n hash operations and $n-1$ encryption operations. If a node does not send open, each node should decrypt $E_i(open_i)$ it receives to solve this case. Figure 8 shows the comparison between PMAB and Dredas [18] in terms of the random source generation time, where the time cost of PMAB ranges from 0.25 s to 3 s and the time cost of Dredas [18] is always 114 s. The time cost of our random source generation method (PVSS [21]) increases linearly and slowly with increase in the number of consensus nodes. We roughly estimate that it will take more than 3000 consensus nodes to spend as much random source generation time as Dredas [18]. However, PMAB uses the *threshol d* of *initialCre di t* to limit the number of consensus nodes. Only a few consensus nodes are required to complete PVSS [21] and audit consensus. Therefore, PMAB's random source generation is more efficient.

ParallGenProof. In the face of large-scale audit case, according to the formation processes of Φ and μ, our protocol in PMAB actually supports parallel generation and aggregation of audit proofs employing MapReduce principle [24]. CSP will divide the whole task of proof generation into small tasks of the same scale for parallel execution and then aggregate the results of single tasks. We set 10 audit tasks as a group and make CSP process the audit proof generation process in parallel, testing the change of proof generation time when the number of audit tasks grows from 100 to 1000, in which 250 data blocks were questioned for each audit task. As shown in Figure 9, it is clear that when CSP is faced with a large number of requests, it proves that the generation time is nearly constant (22.68 s) and does not affect the consensus overhead of PMAB.

ConsensusTime. As shown in Figure 10, we test the time variation of PMAB's *AuditConsensus* per round as the

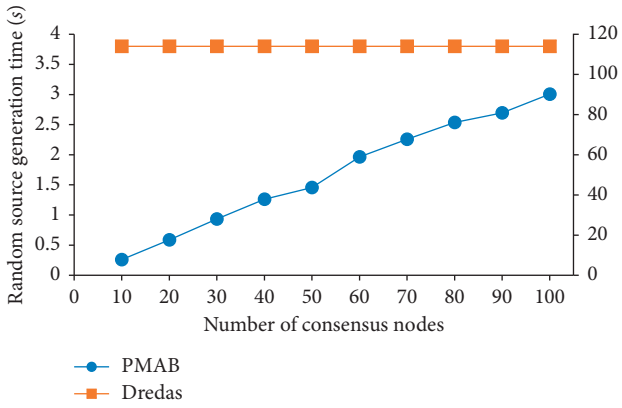


Figure 8: The random source generation time comparison in *RandomGenTime*.

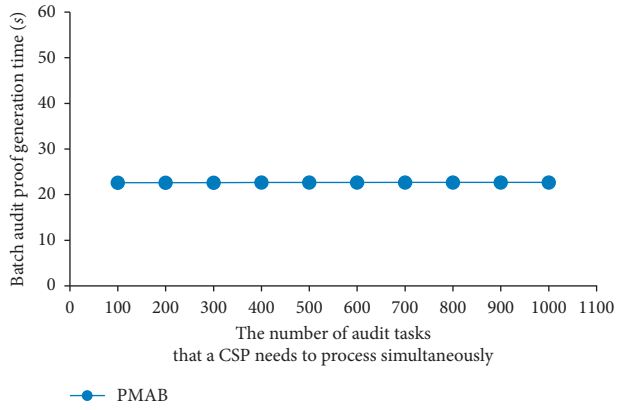


Figure 9: The computation cost in *ParallGenProof*.

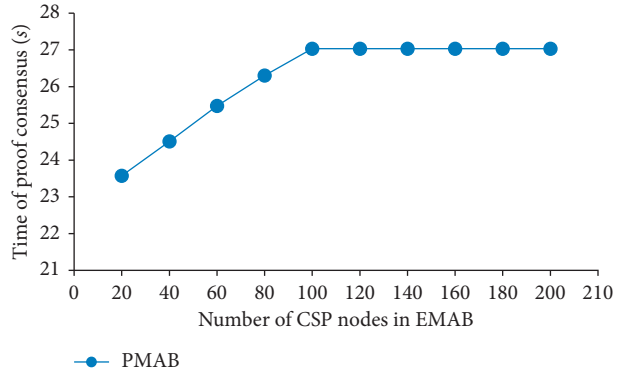


Figure 10: The communication cost in *ConsensusTime*.

number of CSPs increased (from 20 to 200). Each CSP node has 1000 audit tasks and each audit task challenged 250 data blocks. It is obvious that the time of PMAB's audit consensus tends to be constant (27.03 s) with the increase of the number of CSPs, since only a limited number of consensus nodes are needed to complete the audit consensus (the number of consensus nodes in this experiment is limited to less than 100). Therefore, PMAB has strong scalability and stability.

7. Conclusions

In this paper, PMAB for outsourced data in cloud storage was proposed. In PMAB, in order to achieve the goal of automatic audit safely, we constructed an audit chain architecture based on Ouroboros [20] and an incentive mechanism based on credit to allow CSPs to audit each other mutually with anticollusion. In addition, an audit protocol was designed to achieve public audit efficiently with low cost of audit verification. Security and performance analyses showed that PMAB achieves great audit efficiency and security goals. In future work, we will aim to research more specific incentive mechanism quantitative design and efficient problem positioning in batch audit.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported in part by the National Key R&D Program of China (no. 2018YFB0803900), the National Natural Science Foundation of China (nos. 92067103 and 61772403), the Key Research and Development Program of Shaanxi (no. 2021ZDLGY06-02), the Key Scientific Research Program of Education Department of Shaanxi (no. 20JY015), the Fundamental Research Funds for the Central Universities (no. JBF211502), the Natural Science Foundation of Shaanxi Province (no. 2019ZDLGY12-02), the Natural Science Basic Research Plan in Shaanxi Province of China (no. 2020JM-184), the Shaanxi Innovation Team Project (no. 2018TD-007), the Xi'an Science and Technology Innovation Plan (no. 201809168CX9JC10), and National 111 Program of China B16037.

References

[1] L. Liu, O. De Vel, Q.-L. Han, J. Zhang, and Y. Xiang, "Detecting and preventing cyber insider threats: a survey," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 2, pp. 1397–1417, 2018.

[2] X. Jia, D. He, Q. Liu, and K.-K. R. Choo, "An efficient provably-secure certificateless signature scheme for internet-of-things deployment," *Ad Hoc Networks*, vol. 71, pp. 78–87, 2018.

[3] G. Lin, S. Wen, Q.-L. Han, J. Zhang, and Y. Xiang, "Software vulnerability detection using deep neural networks: a survey," *Proceedings of the IEEE*, vol. 108, no. 10, pp. 1825–1848, 2020.

[4] M. Wang, T. Zhu, T. Zhang, J. Zhang, S. Yu, and W. Zhou, "Security and privacy in 6G networks: new areas and new challenges," *Digital Communications and Networks*, vol. 6, no. 3, pp. 281–291, 2020.

[5] O. Gireesha, N. Somu, and K. Krithivasan, "IIVIFS-WASPAS: an integrated Multi-Criteria Decision-Making perspective for

cloud service provider selection," *Future Generation Computer Systems*, vol. 103, pp. 91–110, 2020.

[6] A. G. Ateniese, R. Di Pietro, L. V. Mancini, and G. Tsudik, "Scalable and efficient provable data possession," in *Proceedings of the 4th international Conference on Security and Privacy in Communication Netowrks*, pp. 1–10, ACM, Istanbul, Turkey, September 2008.

[7] C. C. Erway, A. Kupcu, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," *ACM Transaction on Information and System Security*, vol. 17, no. 4, pp. 15.1–15.29, 2015.

[8] Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, "Enabling public auditability and data dynamics for storage security in cloud computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 5, pp. 847–859, 2010.

[9] Y. Zhu, G. J. Ahn, H. Hu, and S. S. Yau, "Dynamic audit services for outsourced storages in clouds," *IEEE Transactions on Services Computing*, vol. 6, no. 2, pp. 227–238, 2013.

[10] H. Tian, Y. Chen, C.-C. Chang et al., "Dynamic-hash-table based public auditing for secure cloud storage," *IEEE Transactions on Services Computing*, vol. 10, no. 5, pp. 701–714, 2015.

[11] J. Shen, J. Shen, X. Chen, X. Huang, and W. Susilo, "An efficient public auditing protocol with novel dynamic structure for cloud data," *IEEE Transactions on Information Forensics & Security*, vol. 12, no. 99, pp. 2402–2415, 2017.

[12] D. Yue, R. Li, Y. Zhang, W. Tian, and C. Peng, "Blockchain based data integrity verification in p2p cloud storage," in *Proceedings of the 2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS)*, pp. 561–568, IEEE, Singapore, Singapore, December 2018.

[13] B. Liu, X. L. Yu, S. Chen, X. Xu, and L. Zhu, "Blockchain based data integrity service framework for iot data," in *Proceedings of the 2017 IEEE International Conference on Web Services (ICWS)*, pp. 468–475, IEEE, Honolulu, HI, USA, June 2017.

[14] H. Yu, Z. Yang, and R. O. Sinnott, "Decentralized big data auditing for smart city environments leveraging blockchain technology," *IEEE Access*, vol. 7, pp. 6288–6296, 2018.

[15] Y. Xu, J. Ren, Y. Zhang, C. Zhang, B. Shen, and Y. Zhang, "Blockchain empowered arbitrable data auditing scheme for network storage as a service," *IEEE Transactions on Services Computing*, vol. 13, no. 2, pp. 289–300, 2020.

[16] P. Huang, K. Fan, H. Yang, K. Zhang, and Y. Yang, "A collaborative auditing blockchain for trustworthy data integrity in cloud storage system," *IEEE Access*, vol. 99, p. 1, 2020.

[17] N. Lu, Y. Zhang, W. Shi, S. Kumari, and K. K. R. Choo, "A secure and scalable data integrity auditing scheme based on hyperledger fabric," *Computers & Security*, vol. 92, Article ID 101741, 2020.

[18] K. Fan, Z. Bao, M. Liu, A. V. Vasilakos, and W. Shi, "Dredas: decentralized, reliable and efficient remote outsourced data auditing scheme with blockchain smart contract for industrial Iot," *Future Generation Computer Systems*, vol. 110, pp. 665–674, 2019.

[19] J. Brown-Cohen, A. Narayanan, A. Psomas, and S. M. Weinberg, "Formal barriers to longest-chain proof-of-stake protocols," in *Proceedings of the 2019 ACM Conference on Economics and Computation*, pp. 459–473, Phoenix, AZ, USA, June 2019.

[20] A. Kiayias, A. Russell, B. David, and R. Oliynykov, "Ouroboros: A provably secure proof-of-stake blockchain protocol," in *Annual International Cryptology Conference*, pp. 357-358, Springer, Berlin, Germany, 2017.

[21] M. Stadler, "Publicly verifiable secret sharing," *Advances in Cryptology—Eurocrypt'96*, pp. 190–199, Berlin, Germany, 1996.

[22] J. C. N. S. Kenneth and G. Paterson, "Efficient identity-based signatures secure in the standard model," in *Information Security and Privacy*, pp. 207–222, Springer, Berlin, Germany, 2006.

[23] H. Chen, M. Pendleton, L. Njilla, and S. Xu, "A survey on Ethereum systems security," *ACM Computing Surveys*, vol. 53, no. 3, pp. 1–43, 2020.

[24] J. Dean and S. Ghemawat, "MapReduce," *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, 2008.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 3680359, 20 pages
https://doi.org/10.1155/2021/3680359

Research Article

Fine-Grained and Controllably Redactable Blockchain with Harmful Data Forced Removal

Huiying Hou,[1] **Shidi Hao**,[1] **Jiaming Yuan**,[2] **Shengmin Xu**,[3] **and Yunlei Zhao**[1]

[1]*College of Computer Science and Technology, Fudan University, Shanghai 200433, China*
[2]*College of Computer and Information Science, University of Oregon, Eugene, OR, USA*
[3]*School of Information Systems, Singapore Management University, Singapore*

Correspondence should be addressed to Yunlei Zhao; ylzhao@fudan.edu.cn

Received 13 April 2021; Revised 26 April 2021; Accepted 11 May 2021; Published 29 May 2021

Academic Editor: Yinghui Zhang

Notoriously, immutability is one of the most striking properties of blockchains. As the data contained in blockchains may be compelled to redact for personal and legal reasons, immutability needs to be skillfully broken. In most existing redactable blockchains, fine-grained redaction and effective deletion of harmful data are mutually exclusive. To close the gap, we propose a fine-grained and controllably redactable blockchain with harmful data forced removal. In the scheme, the originator of the transaction has fine-grained control over who can perform the redaction and which portions of the transaction can be redacted. The redaction transaction is performed after collecting enough votes from miners. All users can provide the index of the block containing the harmful data to receive rewards, which are borne by the malicious user who initially posted the data. Miners can forcibly remove the harmful data based on the index. The malicious user will be blacklisted if the reward is not paid within a period of time, and any transaction about such user will not be performed later. In addition, the scheme supports the redaction of additional data and unexpended transaction output (UTXO) simultaneously. We demonstrate that the scheme is secure and feasible via formal security analysis and proof-of-concept implementation.

1. Introduction

The first application of blockchains is Bitcoin [1, 2], which has revolutionized the financial industry. Ever since, hundreds of such cryptocurrencies rise which do not rely on a central trusted authority. The applications of blockchains go far beyond their use in cryptocurrencies [3–6]. Recently, blockchains have entered numerous domains of applications, such as supply chains, digital twins, insurance, healthcare, or energy. In brief, a blockchain is a decentralized, distributed, potentially public, and immutable log of objects.

Blockchains can be of different types. They can be public as Bitcoin or Ethereum, where the consensus protocol is executed between many pseudonymous participants. Here, the blockchain can be read and written by everyone. Such public blockchains can also be viewed as permissionless because everyone can join the system, participate in the consensus protocol, and establish smart contracts. Blockchains, however, can also be private (also called enterprise or permissioned blockchains) such as Hyperledger, Ethereum Enterprise, Ripple, or Quorum. Here, all the participants and their (digital) identities are known to one or more trusted organizations. Actors have write and read permissions. Such private blockchains can thus be viewed as permissioned because they restrict the actors who can contribute to the consensus on the system state to validate the block transactions. Once an object (such as a block or a transaction) is included in the blockchain (be it private or public), it is persisted and cannot be altered ever again. While immutability is a crucial property of the blockchain, it is often desirable to allow breaking the immutability for personal and legal reasons.

The debate about the immutability of the blockchain becomes more acute due to the adoption of the General Data Protection Regulation (GDPR) by the European Union

(EU). Several provisions of the GDPR are essentially incompatible with the immutable blockchains. In particular, the GDPR imposes that the data have the right to be forgotten, while blockchains such as Bitcoin and Ethereum do not allow to remove any data [7]. In addition, by using the immutability of a blockchain, malicious users can broadcast illegal or harmful data, such as (child) pornography and violence information around the world by spending a small fee. The data will be permanently stored and cannot be modified after they are stable on the chain. It is an enormous challenge for law enforcement agencies such as Interpol [8, 9]. One idea is to "filter" all incoming data to check for malicious content before inserting the data into the chain. However, the recent work of Matzutt et al. [10] showed that the above idea is not feasible. Hence, how to skillfully break the immutability of blockchains is an important and urgent problem to be solved.

To solve the above problem, Ateniese et al. [11] first introduced the concept of redactable blockchain and proposed an elegant solution based on chameleon hash functions [12]. The solution addresses the redaction problem of blockchains at the block level, which is coarse grained.

The redactable blockchain should meet the following two properties: (1) the originator of a transaction can specify a fine-grained access control policy about who can modify the transaction and which portions of the transaction can be redacted; (2) the harmful information contained in the previous block can be removed. Unfortunately, there is no redactable blockchain that meets both requirements.

In this paper, we explored how to effectively realize the fine-grained redactable blockchain. Our thought for realizing fine-grained redaction and effective deletion of harmful data simultaneously is shown in Figure 1. In order to support fine-grained access control, a promising way is to adopt the policy-based chameleon hash function (PCH) [13], which allows the originator of a transaction to specify a fine-grained access control policy about who can modify the transaction. However, it may incur the following issue by adopting the PCH. The malicious originator of the transaction may design an access policy that only allows him/her to modify the transaction to store undeletable harmful information in a blockchain. This does not satisfy the second property. To solve the above problem, we try to combine the technology proposed in [14]. The technology allows all users to create removal transactions by spending some transaction fees. Miners then vote on the transaction, and the harmful information is removed if enough votes are collected within a period of time. Obviously, this does not motivate users to actively remove harmful information from the chain because the user is not only rewarded for doing so but also needs to spend transaction fees. In order to motivate users, in this paper, the users create removal transactions without spending transaction fees. If the transaction passes the verification, the originator will obtain the reward paid by the malicious user who posted the harmful information. In addition, this technique only supports the deletion of additional information in the block and needs to store some "old state," that is, the hash value of the original transaction.

In practice, the redactable blockchains should meet the following three properties: (1) the originator of a transaction can specify a fine-grained access control policy about who can modify the transaction and which portions of the transaction can be redacted; (2) the harmful information contained in the previous block can be removed; (3) the data type that can be redacted is various. In order to support the redaction of various data types, we adopt the idea of the scheme in [15]. In this paper, the blockchain protocol not only supports removing additional information of the block but also redacting UTXO in the transaction. In order to reduce the storage space, we try to adopt a policy-based sanitizable signature [16]. However, in this way, the number of blocks of the signed data cannot be changed, and the set of inadmissible blocks needs to be stored. To solve this problem, we propose an improved policy-based sanitizable signature which allows that the number of blocks of the message m can be changed.

1.1. Contributions. In this paper, we first explore how to effectively realize the fine-grained redaction of blockchains while removing the harmful data. We then propose a fine-grained and controllably redactable blockchain protocol with harmful data forced removal. In a nutshell, the contribution of this paper can be summarized as follows:

(i) We propose a fine-grained and controllably redactable blockchain protocol with harmful data forced removal. Our scheme not only supports the usual redaction of transactions but also the forced removal of harmful information in the blockchain. The originator of the transaction can specify a fine-grained access control structure about who can redact the transaction and which portions of the transaction can be redacted. Authorized users may spend transaction fees to initiate a redaction transaction to redact the above transaction. Any user can initiate a transaction that contains the index of the block included harmful information without spending transaction fees. If the block does contain the harmful information, the miner who creates the new block can forcibly delete the harmful information. Thus, the harmful data can be removed; even the malicious users specify an access control that only they can modify the data. The user who provided the index of the block can receive the reward which is borne by the malicious user who initially posted the data. The malicious user will be blacklisted if the rewards are not paid within a period of time, and any transaction about the user will not be performed later. Furthermore, the scheme supports not only the redaction of additional data but also UTXO, i.e., unspent transaction outputs.

(ii) We present an improved policy-based sanitizable signature scheme, which is based on the scheme in [16]. In our scheme, the number of blocks of the signed data can be changed, and the set of

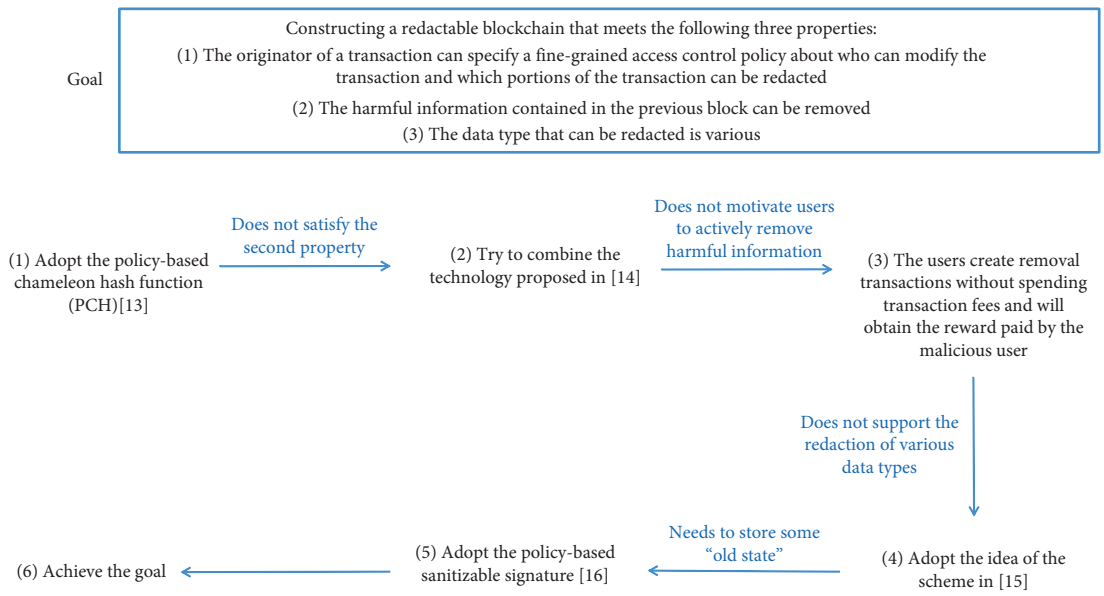


Figure 1: The flowchart of the idea.

inadmissible blocks does not need to be stored. Users who satisfy the access control policy can modify the portions of the signed data that are allowed to be modified. The authorized users can generate the valid signatures for the modified data without interacting with the original signer. The data owner does not need to collect the identities of the candidate authorized users in advance as the proxy signature schemes would require.

(iii) We demonstrate that the proposed scheme is secure and feasible via formal security analysis and proof-of-concept implementation. Specifically, we implement a full-fledged blockchain system, which achieves all the basic functionalities of Ethereum Enterprise. Separately, the blockchain system, including a subset of Ethereum Enterprise's script language, allows the authorized user to redact the transaction and the miner to delete the harmful data. We evaluate the performance of the blockchain system for chain validation in different scenarios. The results show that the redactable blockchain protocol produces only an insignificant (no more than 3.8%) overhead compared to the immutable blockchain.

1.2. Related Work. The concept of sanitizable signature was introduced by Ateniese et al. [17]. A sanitizable signature scheme allows a sanitizer to update the signed data without interacting with the original signer. In order to ensure the security of the scheme, two necessary security requirements are defined in their scheme: (1) unforgeability, that is, only authorized sanitizers can generate the new valid signatures for the updated data; (2) transparency, that is, the updated data and their signatures are indistinguishable from the original information and corresponding signatures. Unfortunately, they did not give a complete definition of the sanitizable signature nor did they provide the formal security analysis. Brzuska et al. [18, 19] provided the formal definition of sanitizable signatures and gave the formalized definition of the basic security requirements. They introduced five formal security requirements, unforgeability, immutability, privacy, transparency, and accountability, and analyzed the relationships between these security requirements. Canard et al. [20] proposed a generic construction of the trapdoor sanitizable signature. In this scheme, the sanitizer can generate the valid signature for the updated data after receiving the trapdoor key from the original signer. Using an accountable chameleon hash, Lai et al. [21] proposed an accountable trapdoor sanitizable signature. However, neither of the above two schemes gives the concrete construction of the sanitizable signature. After that, many concrete sanitizable signature schemes were proposed [22–24]. All of the above sanitizable schemes are not suitable for blockchain rewriting since none of the aforementioned schemes support fine-grained control over candidate sanitizers.

Attribute-based encryption schemes can provide fine-grained access control [25–27]. In order to provide fine-grained access control, some attribute-based sanitizable signature schemes are proposed [16, 28–30]. The scheme in [28] did not give the specific construct of the attribute-based sanitizable signature. The scheme in [29] did not support the expressive access structure. The scheme in [30] only provided an all-or-nothing solution for data modification. The number of blocks of the signed data cannot be changed, and the set of inadmissible blocks needs to be stored in [16]. In a real environment of blockchain rewriting, the number of blocks of the transaction may be changed, and the set of inadmissible blocks does not need to be contained in its signature. Therefore, in this paper, we improve the policy-based sanitizable signature scheme [16] and propose an

improved policy-based sanitizable signature. In this paper, the number of blocks of the signed data can be changed, and the set of inadmissible blocks does not need to be stored. Furthermore, we present a fine-grained and controllably redactable blockchain protocol with harmful data forced removal based on the improved policy-based sanitizable signature scheme.

1.3. Organization. The rest of this paper is organized as follows. In Section 2, we briefly review the preliminaries required in this paper. The system model and design goals are given in Section 3. In Section 4, we introduce the proposed improved policy-based sanitizable signature scheme. We describe the proposed blockchain protocol in Section 5. In Section 6, we introduce the security analysis of the proposed protocol. We evaluate the performance of the proposed protocol in Section 7. Finally, we come to the conclusion in Section 8.

2. Preliminaries

2.1. Notions. We list the notations used in our scheme in Table 1.

2.2. Access Structure. A collection $\mathbb{A} \in 2^{\mathbb{U}} \backslash \{\varnothing\}$ is an access structure on $\mathbb{U}$, where $\mathbb{U}$ denotes attributes' universe. If a set is contained in $\mathbb{A}$, it is the authorized set. Otherwise, it is an unauthorized set. A collection $\mathbb{A}$ is monotone if $C \in \mathbb{A}$ for $\forall B, C \in \mathbb{A}$ and $B \subseteq C$.

2.3. Public Key Encryption. A public key encryption scheme Π consists of the following five algorithms:

(i) $\mathrm{PPGen}_\Pi(1^\kappa)$: this algorithm takes the security parameter κ as the input and outputs the public parameters PP_Π.

(ii) $\mathrm{KGen}_\Pi(\mathrm{PP}_\Pi)$: this algorithm takes the public parameters PP_Π as the input and outputs the public and private key $(\mathrm{pk}_\Pi, \mathrm{sk}_\Pi)$.

(iii) $\mathrm{Enc}_\Pi(\mathrm{pk}_\Pi, m)$: this algorithm takes the public key pk_Π and the message m as the input and outputs a ciphertext c.

(iv) $\mathrm{Dec}_\Pi(\mathrm{sk}_\Pi, c)$: this algorithm takes the private key sk_Π and the ciphertext c as the input and outputs the message m.

(v) $\mathrm{KVrf}_\Pi(\mathrm{sk}_\Pi, \mathrm{pk}_\Pi)$: this algorithm takes the public and private key $(\mathrm{pk}_\Pi, \mathrm{sk}_\Pi)$ as the input and outputs 1 if sk_Π belongs to pk_Π. Otherwise, it outputs 0.

The detailed definition of correctness and security of the public key encryption (PKE) is given in [16]. In this paper, we require correctness and IND-CCA2 security for PKE.

Definition 1. (Π IND-CCA2 security). A public encryption scheme Π is IND-CCA2 secure [16] if for any probabilistic polynomial-time (PPT) adversary $\mathscr{A}$, there exists a negligible function ν such that

Table 1: Notations.

Notation	Meaning
$\mathbb{A}$	A monotone collection
$\mathbb{U}$	The attributes' universe
Π	A public key encryption scheme
k	The security parameter
PP_Π	The public parameters of Π
$(\mathrm{pk}_\Pi, \mathrm{sk}_\Pi)$	The public and private key of Π
m	The message
c	The ciphertext
Σ	A digital signature scheme
PP_Σ	The public parameters of Σ
$(\mathrm{pk}_\Sigma, \mathrm{sk}_\Sigma)$	The signer's public and private key in Σ
σ	The signature in Σ
L	A NP-language
Ω	A noninteractive proof system for L
crs_Ω	A common reference string
x	The statement
ω	The corresponding witness
π	The proof
$\mathrm{PP}_{\mathrm{PCH}}$	The public parameters of PCH
$(\mathrm{sk}_{\mathrm{PCH}}, \mathrm{pk}_{\mathrm{PCH}})$	The master key pair of PCH
$\mathbb{S}$	The set of attributes
$\mathrm{sk}_{\mathbb{S}}$	The user's secret key in PCH
h	The hash value
r	The randomness
m'	The modified message
r'	The new randomness
$\mathrm{PP}_{\mathrm{P3S}}$	The public parameters of P3S
$(\mathrm{sk}_{\mathrm{P3S}}, \mathrm{pk}_{\mathrm{P3S}})$	The master key pair of P3S
$(\mathrm{sk}^{\mathrm{sig}}_{\mathrm{P3S}}, \mathrm{pk}^{\mathrm{sig}}_{\mathrm{P3S}})$	The signer's key pair in P3S
$(\mathrm{sk}^{\mathrm{san}}_{\mathrm{P3S}}, \mathrm{pk}^{\mathrm{san}}_{\mathrm{P3S}})$	The sanitizer's key pair in P3S
M	The description of modification

$$\left| \Pr\left[\mathrm{Exp}^{\mathrm{IND-CCA2}}_{\mathscr{A},\Pi}(k) = 1\right] - \frac{1}{2} \right| \leq \nu(k). \tag{1}$$

The corresponding experiment is depicted in Figure 2.

2.4. Digital Signature. A digital signature scheme Σ consists of the following four algorithms:

(i) $\mathrm{PPGen}_\Sigma(1^\kappa)$: this algorithm takes the security parameter κ as the input and outputs the public parameters PP_Σ.

(ii) $\mathrm{KGen}_\Sigma(\mathrm{PP}_\Sigma)$: this algorithm takes the public parameters PP_Σ as the input and outputs signer's public and private key $(\mathrm{pk}_\Sigma, \mathrm{sk}_\Sigma)$.

(iii) $\mathrm{Sign}_\Sigma(\mathrm{sk}_\Sigma, m)$: this algorithm takes the private key sk_Σ and the message m as the input and outputs the signature σ.

(iv) $\mathrm{Verf}_\Sigma(\mathrm{pk}_\Sigma, m, \sigma)$: this algorithm takes the public key pk_Σ, the message m, and the signature σ as the input and outputs 1 if σ is valid. Otherwise, it outputs 0.

The formal security definition of the digital signature is given in [16]. In this paper, we require correctness and existential unforgeability (eUNF-CMA) for the digital signature.

$\mathrm{Exp}^{IND-CCA2}_{\mathcal{A},\Pi}(\mathrm{k})$
$\mathrm{PP}_\Pi \leftarrow_r PPGen_\Pi(1^k)$
$(sk_\Pi, pk_\Pi) \leftarrow_r KGen_\Pi(\mathrm{PP}_\Pi)$
$b \leftarrow_r \{0,1\}$
$((m_0^*, m_1^*), \mathrm{state}_\mathcal{A}) \leftarrow_r \mathcal{A}^{Dec_\Pi(sk_\Pi,\cdot)}(pk_\Pi)$
$If |m_0^*| \neq |m_1^*| \vee m_0^* \notin \mathcal{M} \vee m_1^* \notin \mathcal{M}$:
$c^* \leftarrow \perp$
$Else$:
$c^* \leftarrow_r Enc_\Pi(pk_\Pi, m_b^*)$
$b^* \leftarrow_r \mathcal{A}^{Dec'_\Pi(sk_\Pi,\cdot)}(state_\mathcal{A}, c^*)$
$where\ Dec'_\Pi$ on input sk_Π and c :
$return \perp$ if $c = c^*$
$return\ Dec_\Pi(sk_\Pi, c)$
$return\ 1$ if $b^* = b$
$return\ 0$

Figure 2: Π IND-CCA2 security.

Definition 2. (Σ unforgeability). A digital signature scheme Σ is unforgeable [16] if for any PPT adversary $\mathcal{A}$, there exists a negligible function ν such that

$$\left|\Pr\left[\mathrm{Exp}^{\mathrm{eUNF-CMA}}_{\mathcal{A},\Sigma}(k) = 1\right]\right| \leq \nu(k). \tag{2}$$

The corresponding experiment is depicted in Figure 3.

2.5. Noninteractive Zero-Knowledge Proof (NIZK). Let $L = \{x | \exists \omega: R(x, \omega) = 1\}$, where L is a NP-language with associated witness relation R. A noninteractive proof system Ω for the language L consists of the following three algorithms:

(i) $\mathrm{PPGen}_\Omega(1^\kappa)$: this algorithm takes the security parameter κ as the input and outputs the common reference string (CRS) crs_Ω.

(ii) $\mathrm{Prove}_\Omega(\mathrm{crs}_\Omega, x, \omega)$: this algorithm takes CRS crs_Ω, the statement x, and the corresponding witness ω as the input and outputs the proof π.

(iii) $\mathrm{Verify}_\Omega(\mathrm{crs}_\Omega, x, \pi)$: this algorithm takes CRS crs_Ω, the statement x, and the proof π as the input and outputs 1 if π is valid. Otherwise, it outputs 0.

The security of the noninteractive zero-knowledge proof (NIZK) is given in [16]. In this paper, we require completeness for NIZK. In addition to completeness, we require two standard security notions for zero-knowledge proofs of knowledge: zero knowledge and simulation-sound extractability. We define them analogous to the definitions given in [16]. Informally speaking, zero knowledge says that the receiver of the proof π does not learn anything except the validity of the statement.

Definition 3. (completeness). A noninteractive proof system is called complete if for all $k \in N$, $\mathrm{crs}_\Omega \leftarrow_r \mathrm{PPGen}(1^k)$, $x \in L$, ω such that $R(x, \omega) = 1$, $\pi \leftarrow_r \mathrm{Prove}_\Omega(\mathrm{crs}_\Omega, x, \omega)$, it holds that $\mathrm{Verify}_\Omega(\mathrm{crs}_\Omega, x, \pi)$.

2.6. Policy-Based Chameleon Hashes. A policy-based chameleon hash (PCH) allows the user, who owns attributes' set that satisfied the access structure, to compute a hash collision [13]. Specifically, a PCH contains the following six PPT algorithms:

$\mathrm{Exp}^{eUNF-CMA}_{\mathcal{A},\Sigma}(\mathrm{k})$
$\mathrm{PP}_\Sigma \leftarrow_r PPGen_\Sigma(1^k)$
$(sk_\Sigma, pk_\Sigma) \leftarrow_r KGen_\Sigma(\mathrm{PP}_\Sigma)$
$Q \leftarrow \emptyset$
$(m^*, \sigma^*) \leftarrow_r A^{Sign'_\Sigma(sk_\Sigma,\cdot)}(pk_\Sigma)$
where $Sign'_\Sigma$ on input sk_Σ and m:
$\sigma \leftarrow_r Sign_\Sigma(sk_\Sigma, m)$
Set $Q \leftarrow Q \cup \{m\}$
return σ
return 1 if $Verf_\Sigma(pk_\Sigma, m^*, \sigma^*) = 1 \wedge m^* \notin Q$
return 0

Figure 3: Σ unforgeability.

(i) $\mathrm{PPGen}_{\mathrm{PCH}}(1^\kappa)$: this is the public parameters' generation algorithm. It takes the security parameter κ as the input and outputs the public parameters $\mathrm{PP}_{\mathrm{PCH}}$.

(ii) $\mathrm{MKeyGen}_{\mathrm{PCH}}(\mathrm{PP}_{\mathrm{PCH}})$: this is the master key generation algorithm. It takes the public parameter $\mathrm{PP}_{\mathrm{PCH}}$ as the input and outputs the master key pair $(\mathrm{sk}_{\mathrm{PCH}}, \mathrm{pk}_{\mathrm{PCH}})$.

(iii) $\mathrm{KGen}_{\mathrm{PCH}}(\mathrm{sk}_{\mathrm{PCH}}, \mathbb{S})$: this is the user's secret key generation algorithm. It takes the master secret key $\mathrm{sk}_{\mathrm{PCH}}$ and the set of attributes $\mathbb{S} \subseteq \mathbb{U}$ as the input and outputs the user's secret key $\mathrm{sk}_\mathbb{S}$.

(iv) $\mathrm{Hash}_{\mathrm{PCH}}(\mathrm{pk}_{\mathrm{PCH}}, \mathbb{A}, m)$: this is the hash algorithm. It takes the master public key $\mathrm{pk}_{\mathrm{PCH}}$, the access structure $\mathbb{A} \in 2^\mathbb{U} \backslash \{\varnothing\}$, and the message m as the input and outputs the hash value h and the randomness r.

(v) $\mathrm{Verify}_{\mathrm{PCH}}(\mathrm{pk}_{\mathrm{PCH}}, m, h, r)$: this is the verification algorithm. It takes the master public key $\mathrm{pk}_{\mathrm{PCH}}$, the message m, the hash value h, and the randomness r as the input and outputs a bit $b = 1$ if h and r are valid. Otherwise, $b = 0$.

(vi) $\mathrm{Adapt}_{\mathrm{PCH}}(\mathrm{pk}_{\mathrm{PCH}}, \mathrm{sk}_\mathbb{S}, m, m', h, r)$: this is the adaption algorithm. It takes the public key $\mathrm{pk}_{\mathrm{PCH}}$, the user's secret key $\mathrm{sk}_\mathbb{S}$, the message m, the modified message m', the hash value h, and some randomness r as the input and outputs a new randomness r'.

The detailed definition of correctness and security of the policy-based chameleon hash is given in [13].

2.7. Policy-Based Sanitizable Signature. A policy-based sanitizable signature (P3S) allows the user, who owns attributes' set that satisfied the access structure, to modify the data and generate the valid signatures for the modified data [16]. Specifically, a P3S contains the following ten PPT algorithms:

(i) $\mathrm{ParGen}_{\mathrm{P3S}}(1^\lambda)$: this is the public parameters' generation algorithm. It takes the security parameter λ as the input and outputs the public parameters $\mathrm{PP}_{\mathrm{P3S}}$.

(ii) $\mathrm{Setup}_{\mathrm{P3S}}(\mathrm{PP}_{\mathrm{P3S}})$: this is the master key generation algorithm. It takes the public parameters $\mathrm{PP}_{\mathrm{P3S}}$ as the input and outputs the master key pair $(\mathrm{sk}_{\mathrm{P3S}}, \mathrm{pk}_{\mathrm{P3S}})$.

(iii) $\mathrm{KGenSig}_{\mathrm{P3S}}(\mathrm{PP}_{\mathrm{P3S}})$: this is the signer's key pair generation algorithm. It takes the public parameters $\mathrm{PP}_{\mathrm{P3S}}$ as the input and outputs the signer's key pair $(\mathrm{sk}_{\mathrm{P3S}}^{\mathrm{sig}}, \mathrm{pk}_{\mathrm{P3S}}^{\mathrm{sig}})$.

(iv) $\mathrm{KGenSan}_{\mathrm{P3S}}(\mathrm{PP}_{\mathrm{P3S}})$: this is the sanitizer's key pair generation algorithm. It takes the public parameters $\mathrm{PP}_{\mathrm{P3S}}$ as the input and outputs the sanitizer's key pair $(\mathrm{sk}_{\mathrm{P3S}}^{\mathrm{san}}, \mathrm{pk}_{\mathrm{P3S}}^{\mathrm{san}})$.

(v) $\mathrm{Sign}_{\mathrm{P3S}}(\mathrm{PP}_{\mathrm{P3S}}, \mathrm{sk}_{\mathrm{P3S}}^{\mathrm{sig}}, m, A, \mathbb{A})$: this is the signing algorithm. It takes the public parameters $\mathrm{PP}_{\mathrm{P3S}}$, the signer's secret key $\mathrm{sk}_{\mathrm{P3S}}^{\mathrm{sig}}$, the message m, the description of admission A, and the access structure $\mathbb{A}$ as the input and outputs a signature σ.

(vi) $\mathrm{AddSan}_{\mathrm{P3S}}(\mathrm{sk}_{P3S}, \mathrm{pk}_{\mathrm{P3S}}^{\mathrm{san}}, \mathbb{S})$: this is the secret sanitizing key generation algorithm. It takes the master secret key $\mathrm{sk}_{\mathrm{P3S}}$, the sanitizer's public key $\mathrm{pk}_{\mathrm{P3S}}^{\mathrm{san}}$, and the set of attributes $\mathbb{S}$ as the input and outputs the secret sanitizing key $\mathrm{sk}_{\mathbb{S}}$ for the sanitizer.

(vii) $\mathrm{Verify}_{\mathrm{P3S}}(\mathrm{pk}_{\mathrm{P3S}}, \mathrm{pk}_{\mathrm{P3S}}^{\mathrm{sig}}, \sigma, m)$: this is the verification algorithm. It takes the master public key $\mathrm{pk}_{\mathrm{P3S}}$, the signer's public key $\mathrm{pk}_{\mathrm{P3S}}^{\mathrm{sig}}$, the signature σ, and the corresponding message m as the input and outputs a bit $b = 1$ if the signature σ is valid. Otherwise, $b = 0$.

(viii) $\mathrm{Sanitize}_{\mathrm{P3S}}(\mathrm{pk}_{\mathrm{P3S}}, \mathrm{pk}_{\mathrm{P3S}}^{\mathrm{sig}}, \mathrm{sk}_{\mathrm{P3S}}^{\mathrm{san}}, \mathrm{sk}_{\mathbb{S}}, \sigma, m, M)$: this is the new signature generation algorithm. It takes the master public key $\mathrm{pk}_{\mathrm{P3S}}$, the signer's public key $\mathrm{pk}_{\mathrm{P3S}}^{\mathrm{sig}}$, the sanitizer's secret key $\mathrm{sk}_{\mathrm{P3S}}^{\mathrm{san}}$, the secret sanitizing key $\mathrm{sk}_{\mathbb{S}}$, the signature σ, the corresponding message m, and the description of modification M as the input and outputs the new signature σ' for the modified message m'.

(ix) $\mathrm{Proof}_{\mathrm{P3S}}(\mathrm{pk}_{\mathrm{P3S}}, \mathrm{sk}_{\mathrm{P3S}}^{\mathrm{sig}}, \sigma, m)$: this is the proof generation algorithm. It takes the master public key $\mathrm{pk}_{\mathrm{P3S}}$, the signer's secret key $\mathrm{sk}_{\mathrm{P3S}}^{\mathrm{sig}}$, the signature σ, and the corresponding message m as the input and outputs the proof π_{P3S}.

(x) $\mathrm{Judge}_{\mathrm{P3S}}(\mathrm{PP}_{\mathrm{P3S}}, \mathrm{pk}_{\mathrm{P3S}}, \mathrm{pk}_{\mathrm{P3S}}^{\mathrm{sig}}, \sigma, m, \pi_{\mathrm{P3S}})$: this is the proof verification algorithm. It takes the public parameter $\mathrm{PP}_{\mathrm{P3S}}$, the master public key $\mathrm{pk}_{\mathrm{P3S}}$, the signer's public key $\mathrm{pk}_{\mathrm{P3S}}^{\mathrm{sig}}$, the signature σ, the corresponding message m, and the proof π_{P3S} as the input and outputs a bit $b = 1$ if the proof π_{P3S} is valid. Otherwise, $b = 0$.

The detailed definition of correctness and security of the policy-based sanitizable signature is given in [16].

2.8. Blockchain Protocol. Let Γ denote an immutable blockchain protocol such as Ethereum Enterprise. The nodes in the blockchain protocol obtain their local chain C based on a common genesis block. The nodes in the blockchain protocol collect transactions in the whole blockchain ecosystem and then package these transactions into a new block. The chain becomes longer as nodes agree on a new block. Nodes can access the blockchain protocol through the following interfaces.

(i) $\{C', \perp\} \leftarrow \Gamma \cdot \mathrm{updateChain}$: returns the chain C' if it is the longer and the valid chain in the blockchain ecosystem. Otherwise, it returns $\perp$.

(ii) $\{0, 1\} \leftarrow \Gamma \cdot \mathrm{validateChain}(C)$: takes the chain C as the input and outputs 1 iff the chain is valid according to the public set of rules.

(iii) $\{0, 1\} \leftarrow \Gamma \cdot \mathrm{validateBlock}(B)$: takes the block B as the input and outputs 1 iff the block is valid according to the public set of rules.

(iv) $\Gamma \cdot \mathrm{broadcast}(x)$: takes the transaction x as the input and broadcasts it to all nodes in the blockchain ecosystem.

3. Problem Formulation

3.1. System Model. As shown in Figure 4, the system model of the proposed redactable blockchain protocol consists of four entities: the trusted authority (TA), the miners, the users, and the authorized users. Note that the model in this paper is similar to the model in [13]. It is more applicable to permissioned blockchains, such as Hyperledger, Ethereum Enterprise, Ripple, and Quorum.

(i) Trusted authority (TA): trusted authority (TA) is fully honest and responsible for generating the signing private key for users who posted the redactable transaction, issuing the attributes and attributes' key for authorized users, and sending keys to miners.

(ii) Miners: miners are fully honest and have powerful computing resources. They are responsible for packaging transactions in the network to generate the new block and removing harmful information from the previous blocks.

(iii) Users: users may be malicious. They can post the usual transaction or the transaction containing the index of the block which includes harmful information to the network. Users get fine-grained control over which users can redact their usual transaction and which portions of the transaction can be redacted. The malicious users may specify an access structure that only allows themselves and their conspirators to redact the transaction.

(iv) Authorized users: the authorized users are semi-honest in the sense that they can modify the portions of the transaction that are allowed to be modified and generate the new valid signatures for the updated data that are indistinguishable from the signatures that the originator generated for the original transaction.

3.2. Design Goals. In order to realize a "healthy" blockchain protocol, the proposed fine-grained and controllably redactable blockchain with harmful data forced removal should satisfy the following properties:

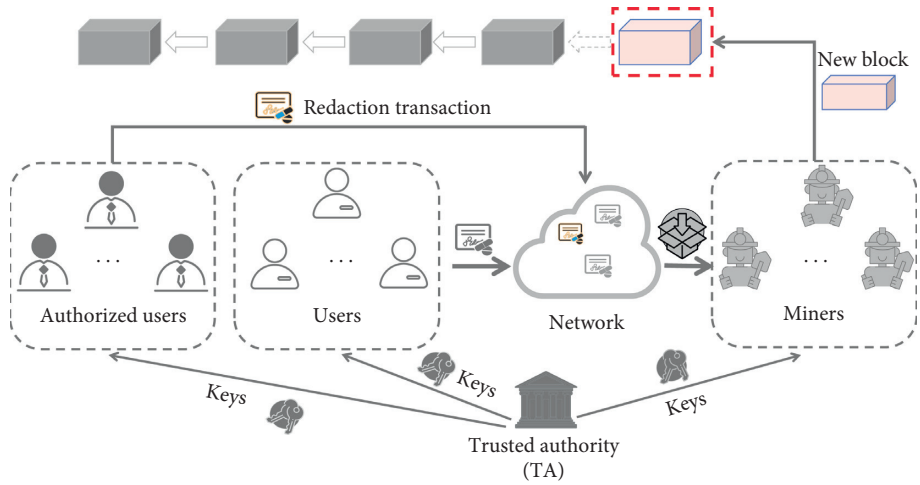


Figure 4: The system model.

(i) Controlled redaction: only authorized users can redact the portions of the transaction that are allowed to be redacted.

(ii) Accountability: the authorized user who redacts the transaction can be tracked.

(iii) Correctness: correctness ensures that the redacted blockchain is "healthy." Specifically, a "healthy" blockchain should meet the following characteristics:

(a) Chain growth: let C_1 and C_2 denote two chains possessed by two honest users at rounds r_1 and r_2, respectively. Then, $\text{len}(C_2) - \text{len}(C_1) \geq \tau \cdot (r_2 - r_1)$, where τ is the speed coefficient and $r_2 > r_1$.

(b) Chain quality: generally speaking, the chain quality says that the ratio of adversarial blocks in any segment of a chain held by an honest party is no more than a fraction $0 < \mu \leq 1$, where μ is the fraction of resources controlled by the adversary.

(c) Editable common prefix: the usual common prefix says that if C_1 and C_2 are two chains possessed by two honest users at rounds r_1 and r_2, for $r_2 > r_1$, C_1 is a prefix of C_2. It can be formally denoted as $C_1^k \leq C_2$, where C_1^k is the chain obtained by removing the last k blocks from C_1, $k \in \mathbb{N}$ is the common prefix parameter. Note that the proposed editable blockchain inherently does not satisfy the common prefix. Suppose the voting phase for the redaction transaction T_i^* is still on at round r_1. At round r_2, the voting phase is complete, and T_i^* replaces T_i, i.e., the redacted block B_i^* replaces B_i. In C_1^k, the i-th block is B_i instead of B_i^* as in C_2. Thus, C_1^k is not the common prefix of C_1 and C_2. We extend this definition. The chains C_1 and C_2 satisfy one of the following:

(1) $C_1^k \leq C_2$

(2) The voting phase is complete, and B_i^* replaces B_i if $B_i^* \in C_2^{(r_2 - r_1)+k}$, $B_i^* \notin C_1^k$

3.3. Threat Model

Definition 4. (controlled redaction). Controlled redaction ensures that only authorized users can redact the portions of the transaction that are allowed to be redacted. In order to formally describe the controlled redaction, we introduce a game between the challenger $\mathscr{C}$ and the adversary $\mathscr{A} = (\mathscr{A}_1, \mathscr{A}_2)$. Here, we consider two adversaries. One of the adversaries is the adversary $\mathscr{A}_1$, who does not possess the attributes' set which satisfies the access control policy. Another is the adversary $\mathscr{A}_2$, who tries to redact the inadmissible portions of the transaction. In order to show how $\mathscr{A}_1$ and $\mathscr{A}_2$ attack the redactable blockchain protocol, we introduce the game between the challenger $\mathscr{C}$ and adversaries $\mathscr{A}_1$ and $\mathscr{A}_2$, respectively.

Firstly, we describe the game between the challenger $\mathscr{C}$ and the adversary $\mathscr{A}_1$. Trusted authority (group manager) is viewed as a challenger $\mathscr{C}$, and the unauthorized user is viewed as an adversary $\mathscr{A}_1$. This game includes the following phases:

(i) Setup phase: the challenger $\mathscr{C}$ runs the $\text{ParGen}_{\text{P3S}}$ and $\text{Setup}_{\text{P3S}}$ algorithm to generate the public parameters PP_{P3S} and the master private/public key pair $(\text{sk}_{\text{P3S}}, \text{pk}_{\text{P3S}})$. Then, $\mathscr{C}$ holds the master private key sk_{P3S} locally. Finally, $\mathscr{C}$ sends the master public key pk_{P3S} and the public parameters PP_{P3S} to the adversary $\mathscr{A}_1$.

(ii) Query phase:

(a) $\text{KGenSan}_{\text{P3S}}$ queries: the adversary $\mathscr{A}_1$ queries sanitizer's private/public key pair for the public parameters PP_{P3S}. $\mathscr{C}$ runs $\text{KGenSan}_{\text{P3S}}$ algorithm and returns the private/public key pair (x_2, y_2) to $\mathscr{A}_1$.

(b) Sign queries: the adversary $\mathscr{A}_1$ queries the signature for the master public key pk_{P3S}, the signature for the transaction m, the set of admissible blocks A, and the access structure $\mathbb{A}$. $\mathscr{C}$ runs $\text{KGenSig}_{\text{P3S}}$ to generate the signing key and then runs Sign algorithm to produce the

signature σ. Finally, $\mathscr{C}$ returns the signature σ to $\mathscr{A}_1$.

(c) $\text{AddSan}_{\text{P3S}}$ queries: the adversary $\mathscr{A}_1$ queries the sanitizer's attribute key for sk_{P3S}, $\text{pk}_{\text{P3S}}^{\text{San}}$, and the attributes' set $\mathbb{S}$ such that $\mathbb{A}(\mathbb{S}) = 0$. $\mathscr{C}$ runs $\text{AddSan}_{\text{P3S}}$ algorithm and returns the sanitizer's attribute key $\text{sk}_{\mathbb{S}} \leftarrow (\sigma_{\text{sk}_{\mathbb{S}}}, \text{sk}_{\mathbb{S}}')$ to $\mathscr{A}_1$.

(d) $\text{Verify}_{\text{P3S}}$ queries: the adversary $\mathscr{A}_1$ queries the verification result for pk_{P3S}, $\text{pk}_{\text{P3S}}^{\text{Sig}}$, σ, and m. $\mathscr{C}$ runs $\text{Verify}_{\text{P3S}}$ algorithm and returns the result to $\mathscr{A}_1$.

(e) $\text{Sanitize}_{\text{P3S}}$ queries: the adversary $\mathscr{A}_1$ queries the sanitizable signature for pk_{P3S}, pk_{P3S}^{Sig}, sk_{P3S}^{San}, $sk_{\mathbb{S}}$, m, σ, and m'. $\mathscr{C}$ runs $\text{Sanitize}_{\text{P3S}}$ algorithm and returns the new signature σ' to $\mathscr{A}_1$.

(f) $\text{Proof}_{\text{P3S}}$ queries: the adversary $\mathscr{A}_1$ queries $(\pi_{\text{P3S}}, \text{pk})$ for pk_{P3S}, $\text{sk}_{\text{P3S}}^{\text{Sig}}$, σ, and m. $\mathscr{C}$ runs $\text{Proof}_{\text{P3S}}$ algorithm and returns $(\pi_{\text{P3S}}, \text{pk})$ to $\mathscr{A}_1$.

(g) $\text{Judge}_{\text{P3S}}$ queries: the adversary $\mathscr{A}_1$ queries the judge result for pk_{P3S}, $\text{sk}_{\text{P3S}}^{\text{Sig}}$, σ, and m. $\mathscr{C}$ runs $\text{Judge}_{\text{P3S}}$ algorithm and returns the result to $\mathscr{A}_1$.

(iii) Challenge phase: the adversary $\mathscr{A}_1$ adaptively chooses the authorized user's attributes' set $\mathbb{S}$ $(\mathbb{A}(\mathbb{S}) = 0)$. Then, $\mathscr{A}_1$ runs $\text{Sanitize}_{\text{P3S}}$ algorithm to generate the challenged signature σ^* for the challenged transaction m^*. Finally, the adversary $\mathscr{A}_1$ sends $(\mathbb{S}, m^*, \sigma^*)$ to $\mathscr{C}$.

(iv) Verify phase: the adversary $\mathscr{A}_1$ performs polynomial queries as in the query phase. Consider the adversary $\mathscr{A}_1$ has made L queries, and let $Q = \{\text{sk}_{\mathbb{S}}, \mathbb{S}, m_i, A_i, \mathbb{A}_i, \sigma_i\}_{i=1}^{[|Q|]}$ denote the set of information obtained through these queries. $\mathscr{C}$ runs $\text{Verify}_{\text{P3S}}(\text{pk}_{\text{P3S}}, \text{pk}_{\text{P3S}}^{\text{Sig}}, A, \mathbb{A}, m^*, \sigma^*)$ algorithm and outputs a bit b_0. If $b_0 = 1$, $\mathscr{C}$ checks whether there exists an $i \in [|Q|], \sigma^*)$ such that $\mathbb{A}(\mathbb{S}) = 0$. If there is such an i, the challenger $\mathscr{C}$ outputs $b_1 = 1$. Otherwise, $\mathscr{C}$ outputs $b_1 = 0$.

We say that the adversary $\mathscr{A}_1$ wins if $b_1 = 1$. In the above game, we want to show that the adversary $\mathscr{A}_1$, who does not possess the attributes' set $\mathbb{S}$ such that $\mathbb{A}(\mathbb{S}) = 0$, should not generate the new valid witness for the transaction. The adversary's goal is to correctly generate the valid signature σ' for the transaction m^*. We set the advantage of a polynomial-time adversary $\mathscr{A}_1$ in this game to be $\Pr[b_1 = 1]$. We say the proposed scheme satisfies the unforgeability of the signature if for any polynomial-time adversary $\mathscr{A}_1$, $\Pr[b_1 = 1] < (1/\text{poly}(n))$ for sufficiently large n, where poly stands for a polynomial function.

Then, we describe the game between the challenger $\mathscr{C}$ and the adversary $\mathscr{A}_2$. Trusted authority (group manager) is viewed as a challenger $\mathscr{C}$, and the authorized user is viewed as an adversary $\mathscr{A}_2$. This game includes the following phases:

(i) Setup phase: the challenger $\mathscr{C}$ runs the $\text{ParGen}_{\text{P3S}}$ and $\text{Setup}_{\text{P3S}}$ algorithm to generate the public parameters PP_{P3S} and the master private/public key pair $(\text{sk}_{\text{P3S}}, \text{pk}_{\text{P3S}})$. Then, $\mathscr{C}$ holds the master private key sk_{P3S} locally. Finally, $\mathscr{C}$ sends the master public key pk_{P3S} and the public parameters PP_{P3S} to the adversary $\mathscr{A}_2$.

(ii) Query phase:

(a) $\text{KGenSan}_{\text{P3S}}$ queries: the adversary $\mathscr{A}_2$ queries sanitizer's private/public key pair for the public parameters PP_{P3S}. $\mathscr{C}$ runs $\text{KGenSan}_{\text{P3S}}$ algorithm and returns the private/public key pair (x_2, y_2) to $\mathscr{A}_2$.

(b) Sign queries: the adversary $\mathscr{A}_2$ queries the signature for the master public key pk_{P3S}, the signature for the message m, the set of admissible blocks F, and the access structure $\mathbb{A}$. $\mathscr{C}$ runs $\text{KGenSig}_{\text{P3S}}$ to generate the signing key and then runs Sign algorithm to produce the signature σ. Finally, $\mathscr{C}$ returns the signature σ to $\mathscr{A}_2$.

(c) $\text{AddSan}_{\text{P3S}}$ queries: the adversary $\mathscr{A}_2$ queries the sanitizer's attribute key for sk_{P3S}, $\text{pk}_{\text{P3S}}^{\text{San}}$, and the attributes' set $\mathbb{S}$ such that $\mathbb{A}(\mathbb{S}) = 1$. $\mathscr{C}$ runs $\text{AddSan}_{\text{P3S}}$ algorithm and returns the sanitizer's attribute key $\text{sk}_{\mathbb{S}} \leftarrow (\sigma_{\text{sk}_{\mathbb{S}}}, \text{sk}_{\mathbb{S}}')$ to $\mathscr{A}_2$.

(d) $\text{Verify}_{\text{P3S}}$ queries: the adversary $\mathscr{A}_2$ queries the verification result for pk_{P3S}, $\text{pk}_{\text{P3S}}^{\text{Sig}}$, σ, and m. $\mathscr{C}$ runs $\text{Verify}_{\text{P3S}}$ algorithm and returns the result to $\mathscr{A}_2$.

(e) $\text{Sanitize}_{\text{P3S}}$ queries: the adversary $\mathscr{A}_2$ queries the sanitizable signature for pk_{P3S}, $\text{pk}_{\text{P3S}}^{\text{Sig}}$, $\text{sk}_{\text{P3S}}^{\text{San}}$, $\text{sk}_{\mathbb{S}}$, m, σ, and m'. $\mathscr{C}$ runs $\text{Sanitize}_{\text{P3S}}$ algorithm and returns the new signature σ' to $\mathscr{A}_2$.

(f) $\text{Proof}_{\text{P3S}}$ queries: the adversary $\mathscr{A}_2$ queries $(\pi_{\text{P3S}}, \text{pk})$ for pk_{P3S}, $\text{sk}_{\text{P3S}}^{\text{Sig}}$, σ, and m. $\mathscr{C}$ runs $\text{Proof}_{\text{P3S}}$ algorithm and returns $(\pi_{\text{P3S}}, \text{pk})$ to $\mathscr{A}_2$.

(g) $\text{Judge}_{\text{P3S}}$ queries: the adversary $\mathscr{A}_2$ queries the judge result for pk_{P3S}, $\text{sk}_{\text{P3S}}^{\text{Sig}}$, σ, and m. $\mathscr{C}$ runs $\text{Judge}_{\text{P3S}}$ algorithm and returns the result to $\mathscr{A}_2$.

(iii) Challenge phase: the adversary $\mathscr{A}_2$ adaptively chooses the authorized user's attributes' set $\mathbb{S}$ $(\mathbb{A}(\mathbb{S}) = 1)$. Then, $\mathscr{A}_2$ runs $\text{Sanitize}_{\text{P3S}}$ algorithm to generate the challenged signature σ^* for the challenged message m^* which does not contain all inadmissible blocks. Finally, the adversary $\mathscr{A}_2$ sends $(\mathbb{S}, m^*, \sigma^*)$ to $\mathscr{C}$.

(iv) Verify phase: the adversary $\mathscr{A}_2$ performs polynomial queries as in the query phase. Consider the adversary $\mathscr{A}_2$ has made L queries, and let $Q = \{\text{sk}_{\mathbb{S}}, \mathbb{S}, m_i, A_i, \mathbb{A}_i, \sigma_i\}_{i=1}^{[|Q|]}$ denote the set of information obtained through these queries. $\mathscr{C}$ runs $\text{Verify}_{\text{P3S}}(\text{pk}_{\text{P3S}}, \text{pk}_{\text{P3S}}^{\text{Sig}}, A, \mathbb{A}, m^*, \sigma^*)$ algorithm and outputs a bit b_0. If $b_0 = 1$, $\mathscr{C}$ checks whether there exists an $i \in [|Q|], m^*$ which does not contain all inadmissible blocks. If there is such an i, the challenger $\mathscr{C}$ outputs $b_1 = 1$. Otherwise, $\mathscr{C}$ outputs $b_1 = 0$.

We say that the adversary $\mathscr{A}_2$ wins if $b_1 = 1$. In the above game, we want to show that the adversary $\mathscr{A}_2$, who redacts

the inadmissible blocks, should not generate the new valid signature. The adversary's goal is to correctly generate the valid signature σ' for the message m^*. We set the advantage of a polynomial-time adversary $\mathscr{A}_2$ in this game to be $\Pr[b_1 = 1]$. We say the proposed scheme satisfies controlled redaction if for any polynomial-time adversary $\mathscr{A}_2$, $\Pr[b_1 = 1] < (1/\text{poly}(n))$ for sufficiently large n, where poly stands for a polynomial function.

Definition 5. (accountability). We say that the proposed fine-grained and controllably redactable blockchain with harmful data forced removal satisfies accountability if TA can extract signer's identity from any valid transaction's signature with nonnegligible probability.

4. The Improved Policy-Based Sanitizable Signature

4.1. Algorithm Definition. Let PCH denote a policy-based chameleon hash, Ω label a simulation-sound extractable noninteractive zero-knowledge proof (NIZK) system, f be a one-way function, Π denote an IND-CCA2-secure public key encryption scheme, and Σ be an eUNF-CMA-secure signature scheme. Specifically, the improved policy-based sanitizable signature is described as follows:

(i) $\text{ParGen}_{\text{P3S}}(1^\kappa)$: it takes a security parameter κ as the input and outputs $\text{PP}_{\text{P3S}} = (\text{crs}_\Omega, \text{PP}_\Pi, \text{PP}_\Sigma, \text{PP}_{\text{PCH}}, f, h)$, where $\text{PP}_\Pi \leftarrow \text{PPGen}_\Pi(1^\kappa)$, $\text{crs}_\Omega \leftarrow \text{PPGen}_\Omega(1^\kappa)$, $\text{PP}_\Sigma \leftarrow \text{PPGen}_\Sigma(1^\kappa)$, $\text{PP}_{\text{PCH}} \leftarrow \text{PPGen}_{\text{PCH}}(1^\kappa)$, $f: D_f \longrightarrow R_f$ is a one-way function, and H is a cryptographic hash function.

(ii) $\text{Setup}_{\text{P3S}}(\text{PP}_{\text{P3S}})$: it takes PP_{P3S} as the input and outputs $(\text{sk}_{\text{P3S}}, \text{pk}_{\text{P3S}}) \leftarrow (\text{sk}_{\text{PCH}}, \text{sk}_\Sigma), (\text{pk}_{\text{PCH}}, \text{pk}_\Sigma)$, where $(\text{sk}_{\text{PCH}}, \text{pk}_{\text{PCH}}) \leftarrow \text{MKeyGen}_{\text{PCH}}(\text{PP}_{\text{PCH}})$ and $(\text{sk}_\Sigma, \text{pk}_\Sigma) \leftarrow \text{KGen}_\Sigma(\text{PP}_\Sigma)$.

(iii) $\text{KGenSig}_{\text{P3S}}(\text{PP}_{\text{P3S}})$: it takes PP_{P3S} as the input and outputs $(\text{sk}^{\text{Sig}}_{\text{P3S}}, \text{pk}^{\text{Sig}}_{\text{P3S}}) \leftarrow ((x_1, \text{sk}'_\Sigma, \text{sk}_\Pi), (y_1, \text{pk}'_\Sigma, \text{pk}_\Pi))$, where $x_1 \leftarrow D_f$, $y_1 \leftarrow f(x_1)$, $(\text{sk}_\Pi, \text{pk}_\Pi) \leftarrow \text{KGen}_\Pi(\text{PP}_\Pi)$, and $(\text{sk}'_\Sigma, \text{pk}'_\Sigma) \leftarrow \text{KGen}_\Sigma(\text{PP}_\Sigma)$.

(iv) $\text{KGenSan}_{\text{P3S}}(\text{PP}_{\text{P3S}})$: it takes PP_{P3S} as the input and outputs (x_2, y_2), where $x_2 \leftarrow D_f$ and $y_2 \leftarrow f(x_2)$.

(v) $\text{Sign}_{\text{P3S}}(\text{pk}_{\text{P3S}}, \text{sk}^{\text{Sig}}_{\text{P3S}}, m, A, \mathbb{A})$: it takes pk_{P3S}, $\text{sk}^{\text{Sig}}_{\text{P3S}}$, the message m, the set of admissible blocks A, and the access structure $\mathbb{A}$ as the input and outputs $\perp$ if $\mathbb{A} = \varnothing$. Otherwise, it outputs $\sigma \leftarrow (h, r, A, \sigma_m, \mathbb{A}, \pi, c)$, where $(h, r) \leftarrow \text{Hash}_{\text{PCH}}(\text{pk}_{\text{PCH}}, m, \mathbb{A})$, $\sigma_m \leftarrow \text{Sign}_\Sigma(\text{sk}'_\Sigma, (\text{pk}_{\text{P3S}}, \text{pk}^{\text{Sig}}_{\text{P3S}}, A, H(i \| m_{!A}), h, \mathbb{A}))$, $c \leftarrow \text{Enc}_\Pi(\text{pk}_\Pi, y_1)$, and $\pi \leftarrow \text{Prove}_\Omega\{(x_1, x_2, \text{sk}_\Pi, \sigma_{\text{sk}_\mathbb{S}})$: $(y_1 = f(x_1) \wedge c = \text{Enc}_\Pi(\text{pk}_\Pi, y_1) \wedge \text{KVrf}_\Pi(\text{sk}_\Pi, \text{pk}_\Pi) = 1) \vee (y_2 = f(x_2) \wedge c = \text{Enc}_\Pi(\text{pk}_\Pi, y_2) \wedge \text{Verf}_\Sigma(\text{pk}_\Sigma, (y_2, \text{pk}_{\text{P3S}}), \sigma_{\text{sk}_\mathbb{S}}) = 1)\}(l)$. Note that $l = (\text{PP}_{\text{P3S}}, \text{pk}_{\text{P3S}}, \text{pk}^{\text{Sig}}_{\text{P3S}}, h, r, m, A, \mathbb{A}, H(i \| m_{!A}), \sigma_m, c)$.

(vi) $\text{AddSan}_{\text{P3S}}(\text{sk}_{\text{P3S}}, \text{pk}^{\text{San}}_{\text{P3S}}, \mathbb{S})$: it takes sk_{P3S}, $\text{pk}^{\text{San}}_{\text{P3S}}$, and the attributes' set $\mathbb{S}$ as the input and outputs the sanitizing key $\text{sk}_\mathbb{S} \leftarrow (\sigma_{\text{sk}_\mathbb{S}}, \text{sk}'_\mathbb{S})$, where $\sigma_{\text{sk}_\mathbb{S}} \leftarrow \text{Sign}_\Sigma(\text{sk}_\Sigma, (\text{pk}^{\text{San}}_{\text{P3S}}, \text{pk}_{\text{P3S}}))$ and $\text{sk}'_\mathbb{S} \leftarrow \text{KGen}_{\text{PCH}}(\text{sk}_{\text{PCH}}, \mathbb{S})$.

(vii) $\text{Verify}_{\text{P3S}}(\text{pk}_{\text{P3S}}, \text{pk}^{\text{Sig}}_{\text{P3S}}, \sigma, m)$: it takes pk_{P3S}, $\text{pk}^{\text{Sig}}_{\text{P3S}}$, σ, and m as the input and outputs 1 if π and σ_m are valid, $\text{Verify}_{\text{PCH}}(\text{pk}_{\text{PCH}}, m, r, h) = 1$, and $H(i \| m_{!A})$ can be computed from the message m. Otherwise, it outputs $\perp$.

(viii) $\text{Sanitize}_{\text{P3S}}(\text{pk}_{\text{P3S}}, \text{pk}^{\text{Sig}}_{\text{P3S}}, \text{sk}^{\text{San}}_{\text{P3S}}, \text{sk}_\mathbb{S}, m, \sigma, m')$: it takes pk_{P3S}, $\text{pk}^{\text{Sig}}_{\text{P3S}}$, $\text{sk}^{\text{San}}_{\text{P3S}}$, $\text{sk}_\mathbb{S}$, m, σ, and m' as the input. If $\sigma_{\text{sk}_\mathbb{S}}$ or σ is not valid, it outputs $\perp$. Otherwise, the sanitizer computes $r' \leftarrow \text{Adapt}_{\text{PCH}}(\text{pk}_{\text{PCH}}, \text{sk}_\mathbb{S}, m, m', h, r)$, $c' \leftarrow \text{Enc}_\Pi(\text{pk}_\Pi, y_2)$, and $\pi' \leftarrow \text{Prove}_\Omega\{(x_1, x_2, \text{sk}_\Pi, \sigma_{\text{sk}_\mathbb{S}})$: $(y_1 = f(x_1) \wedge c' = \text{Enc}_\Pi(\text{pk}_\Pi, y_1) \wedge \text{KVrf}_\Pi(\text{sk}_\Pi, \text{pk}_\Pi) = 1) \vee (y_2 = f(x_2) \wedge c' = \text{Enc}_\Pi(\text{pk}_\Pi, y_2) \wedge \text{Verf}_\Sigma(\text{pk}_\Sigma, (y_2, \text{pk}_{\text{P3S}}), \sigma_{\text{sk}_\mathbb{S}}) = 1)\}(l)$. Note that $l = (\text{PP}_{\text{P3S}}, \text{pk}_{\text{P3S}}, \text{pk}^{\text{Sig}}_{\text{P3S}}, h, r', m', A, \mathbb{A}, H(i \| m_{!A}), \sigma_m, c')$. Then, the sanitizer sets $(\sigma', m') \leftarrow ((h, r', A, \sigma_m, \mathbb{A}, \pi, c'), m')$. If (σ', m') is not valid, this algorithm outputs $\perp$. Otherwise, it outputs (σ', m').

(ix) $\text{Proof}_{\text{P3S}}(\text{pk}_{\text{P3S}}, \text{sk}^{\text{Sig}}_{\text{P3S}}, \sigma, m)$: it takes pk_{P3S}, $\text{sk}^{\text{Sig}}_{\text{P3S}}$, σ, and m as the input and outputs (π_{P3S}, pk), where $\text{pk} \leftarrow \text{Dec}_\Pi(\text{sk}_\Pi, c)$, $\pi_{\text{P3S}} \leftarrow \text{Prove}_\Omega\{(\text{sk}_\Pi, x_1)$: $\text{pk} = \text{Dec}_\Pi(\text{sk}_\Pi, c) \wedge \text{KVrf}_\Pi(\text{sk}_\Pi, \text{pk}_\Pi) = 1 \wedge y_1 = f(x_1)\}(l)$, and $l = (\text{PP}_{\text{P3S}}, \text{pk}_{\text{P3S}}, \text{pk}^{\text{ig}}_{\text{P3S}}, \sigma, \text{pk}, m)$.

(x) $\text{Judge}_{\text{P3S}}(\text{pk}_{\text{P3S}}, \text{pk}^{\text{Sig}}_{\text{P3S}}, \text{pk}, \pi_{\text{P3S}}, \sigma, m)$: it takes pk_{P3S}, $\text{pk}^{\text{Sig}}_{\text{P3S}}$, pk, π_{P3S}, σ, and m as the input. If σ and π_{P3S} are valid, it outputs 1. Otherwise, it outputs 0.

The improved policy-based sanitizable signature replaces the inadmissible block set $m_{!A}$ in [16] with $H(i \| m_{!A})$ to allow that the number of blocks of the message m can be changed, and the set of inadmissible blocks does not need to be stored. Here, $m_{!A}$ denotes the set of blocks that are not allowed to be modified. The security definition and analysis are given in Appendixes A and B, respectively.

5. The Proposed Protocol

5.1. An Overview. The workflow of the proposed blockchain protocol can be described as follows. Firstly, users can generate a local chain C based on the common genesis block genesis and initialize the redaction transaction list R, the removal transaction list D, the penalty payment transaction list P, and the blacklist L to be empty. After that, users run $\Gamma \cdot \text{updateChain}$ to obtain the longest chain in the blockchain network. When the user wants to redact the previous transaction, he/she first broadcasts a redaction transaction by spending some transaction fees. The transaction will be added to the list R if it is valid. The miners vote on the transaction. The transaction can be executed if enough votes are collected within a period of time as shown in Figure 5. When a user finds harmful information contained in a block, he/she creates a removal transaction containing the index of the block without spending transaction fees. Miners create

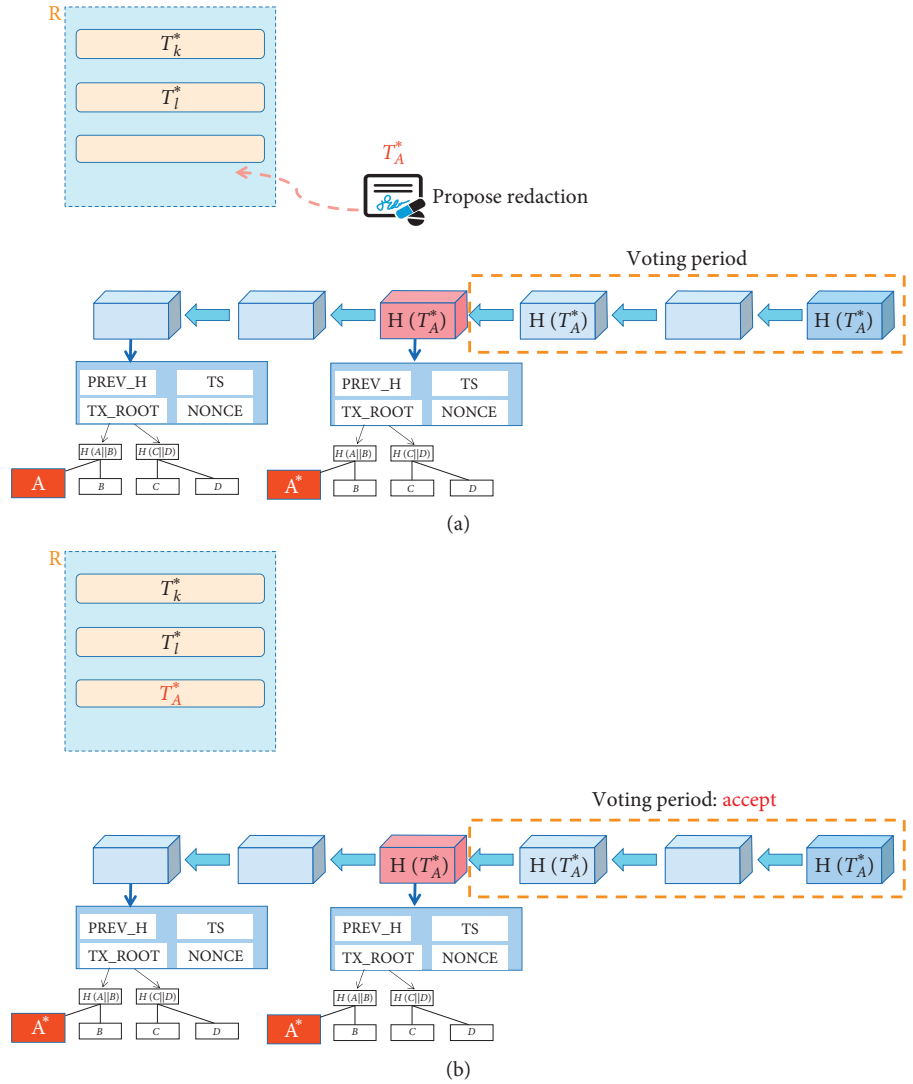


Figure 5: The redaction of the transaction. (a) Proposing a redaction A^* for a transaction A. (b) After a successful voting phase, A^* replaces A in the chain.

new blocks that contain at least one transaction in D and one in P if they are not empty. The miner removes the harmful information from the block according to the provided index. Meanwhile, the miner generates a penalty payment transaction added to P as shown in Figure 6. The transaction will be removed from list P after the penalty is paid by the malicious user. If the malicious user fails to pay the penalty within a period of time, he/she will be added to the blacklist L. After that, all transactions relating to the malicious user will never be performed.

5.2. Description of the Proposed Protocol. The proposed blockchain protocol runs in a sequence of rounds r and consists of the following six algorithms (Figures 7–10):

(i) Initialization: get the local chain $C \leftarrow$genesis, where genesis denotes a common genesis block. Set round $r \leftarrow 1$, and initialize empty lists R, D, P, and L.

(ii) Chain update: at the beginning of each round r, users run $\{C', \perp\} \leftarrow \Gamma \cdot \text{updateChain}$ to get the longest chain C' in the blockchain network.

(iii) Propose a redaction: the user proposes a redaction of the transaction T_A by spending some transaction fees.

  (a) Firstly, the user creates a redaction transaction redact T using the new transaction T_A^* as shown in Figure 7. In this process, the improved policy sanitizable signature is used to generate the witness for the transaction. We can see from Figure 7 that the hash values h for T_A and (T_A^*) are the same. Therefore, the hash value of this block will not be changed after redacting the transaction.

  (b) Then, he/she runs $\Gamma \cdot \text{broadcast}(T_A^*)$ to broadcast the redacted transaction to the blockchain network.

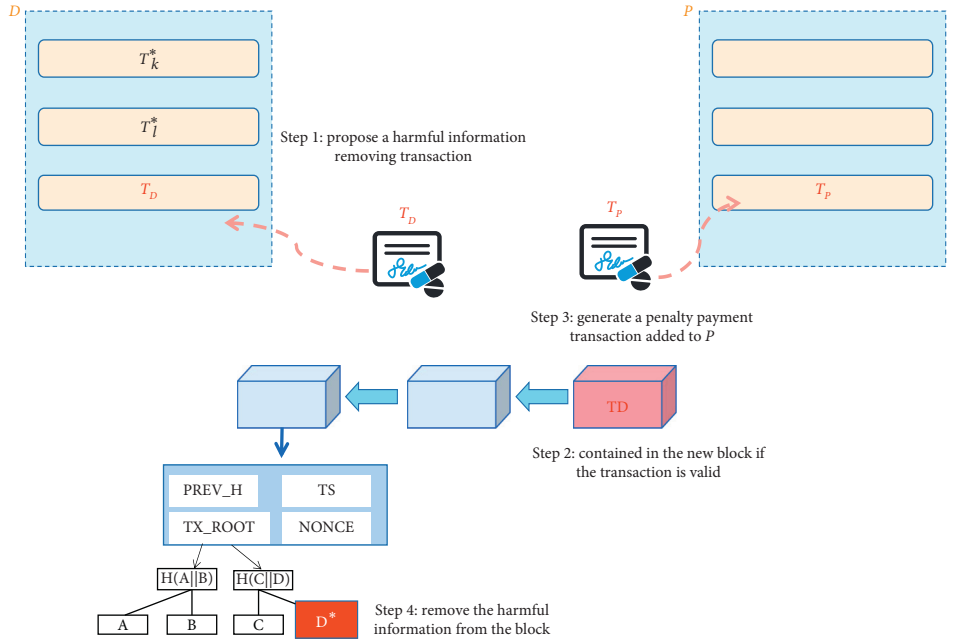


Figure 6: The removal of harmful information.

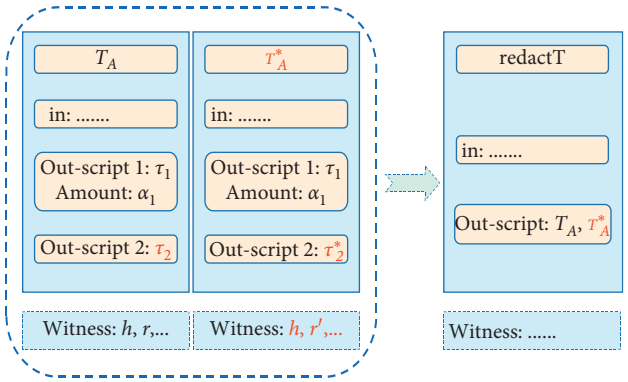


Figure 7: The transaction redact T.

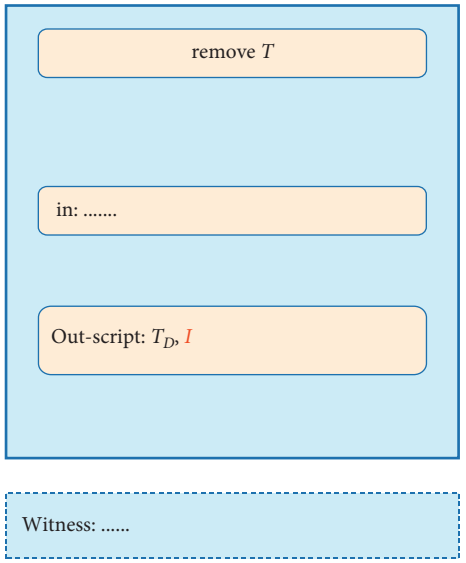


Figure 8: The transaction remove T.

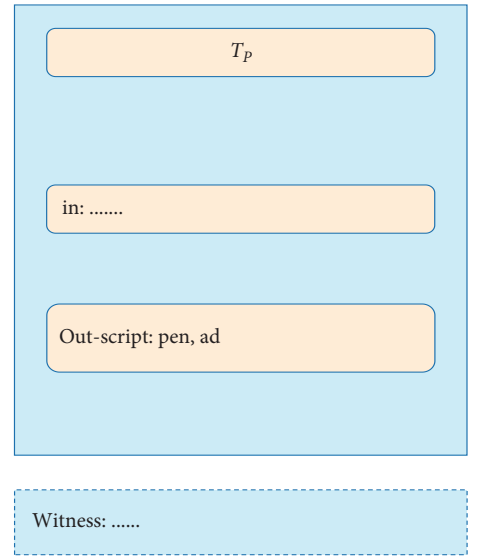


Figure 9: The transaction penatly T.

(c) Finally, miners add the transaction rdact T to the list R if the data τ_2 are UTXO. Otherwise, the transaction is discarded.

(iv) Propose a removal of harmful information when the user finds that the transaction T_D, contained in the block with the index I, has the harmful information.

(a) Firstly, as shown in Figure 8, the user creates a removal transaction remove T, which does not cost transaction fee and contains the block's index I and the transaction T_D.

(b) Then, he/she broadcasts the transaction remove T.

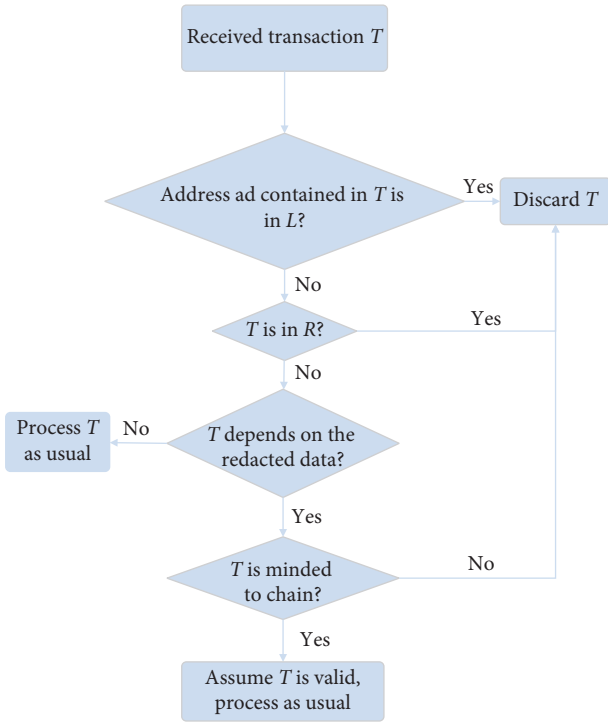


Figure 10: The verification of the received transaction T.

(c) The transaction remove T will be added to the list D if the block I does contain the harmful information. Meanwhile, the penalty payment transaction T_P will be created and added to the list P. As shown in Figure 9, the transaction T_P contains the amount of the penalty pen and the address ad of the malicious user who posts the harmful information. The transaction T_P will be removed from the list P after the malicious user pays the penalty.

(v) Redacting the chain:

(a) For the candidate transaction T_A in the list R, the miner substitutes it with the new transaction T_A^* if the voting process on it has been completed and enough votes $v \geq \rho$ have been collected within a period of time t. The transaction T_A is discarded if the votes $v < \rho$ within a period of time t. If the voting on T_A is still in progress, nothing will be done. Here, ρ denotes the threshold of votes and can be specified by consensus among all users in the blockchain network.

(b) For the candidate transaction T_D in the list D, the miner removes the harmful information from T_D which is contained in the block with the index I.

(c) For the candidate transaction T_P in the list P, the miner first verifies whether the malicious user pays the penalty within a period of time t_1. If the malicious user pays the penalty, the transaction is removed from P. If the malicious user does not pay the penalty within a specified period of time, the user is added to the blacklist L, and the transaction T_P is removed from the list P.

(vi) Creating a new block: the miner collects all transactions from the network for the $r-$th round and builds a new block B which meets the following conditions:

(a) It contains at least one transaction in D and one in P if they are not empty.

(b) It contains a vote $H(T_A)$ on the candidate transaction in the list R if the voting on T_A is still in process and the miner is willing to endorse.

(c) All transactions contained in it comply with the usual transaction rules in the Ethereum Enterprise blockchain, and the validation process is shown in Figure 10.

Finally, if all blocks contained in the local chain C satisfy $\Gamma \cdot \text{validateBlock}(B) = 1$ and $\Gamma \cdot \text{validateChain}(C) = 1$, the miner extends the local chain $C \leftarrow C \| B$ and broadcasts the extended chain to the blockchain network.

6. Security Analysis

In this section, we analyze the security of the fine-grained and controllably redactable blockchain protocol with harmful data forced removal in terms of correctness, controlled redaction, and accountability.

Theorem 1 (correctness). The correctness of a blockchain consists of the following three aspects:

(1) Chain growth: if the based immutable blockchain protocol Γ satisfies chain growth, the extended editable blockchain protocol Γ' also satisfies chain growth.

(2) Chain quality: if the based immutable blockchain protocol Γ satisfies chain quality, the extended editable blockchain protocol Γ' also satisfies chain growth for any $\rho > \mu$. Here, ρ denotes the ratio of blocks containing the votes of the redacted transaction within a period of time.

(3) Common prefix: if the based immutable blockchain protocol Γ satisfies the common prefix, the extended editable blockchain protocol Γ' also satisfies the common prefix.

Proof.

(1) Chain growth: we note that the redaction in Γ' cannot reduce the length of the chain C by removing a block from the chain. Thus, the redact operations have no effect on the length of the chain. In conclusion, Γ' satisfies chain growth if Γ satisfies chain growth.

(2) Chain quality: suppose the adversary $\mathcal{A}$ posts a malicious redaction transaction T_i^* for the previous

transaction T_i. $\mathscr{A}$ mines at most μ ratio of blocks in the voting phase because the adversary only has μ computational power. Thus, T_i^* cannot be performed due to $\rho > \mu$. In conclusion, only the honest redaction transaction T_i^* can be performed and added to the chain.

(3) Common prefix: if the chain C_2 is not redacted, Γ' runs as the immutable blockchain Γ. Thus, Γ' satisfies the common prefix. If the chain C_2 is redacted and the redacted block B_i^* replaces B_i in C_2, the voting phase for the block B_i^* is completed, and enough votes are received. In conclusion, the extended editable blockchain protocol Γ' also satisfies the common prefix. □

Theorem 2 (controlled redaction). In the proposed scheme, for each PPT adversary $\mathscr{A}$, it is computationally infeasible to generate a valid signature for the redacted transaction.

Proof. To prove this theorem, we consider two types of adversaries. One of the adversaries is the adversary $\mathscr{A}_1$ who does not possess the attributes' set which satisfies the access control policy. Another is the adversary $\mathscr{A}_2$, who tries to redact the inadmissible portions of the transaction. In order to show how $\mathscr{A}_1$ and $\mathscr{A}_2$ attack the redactable blockchain protocol, we introduce the two games between the challenger $\mathscr{C}$ and adversaries $\mathscr{A}_1$ and $\mathscr{A}_2$, respectively. Firstly, we define a game between the challenger $\mathscr{C}$ and the adversary $\mathscr{A}_1$.

Game 1: in Game 1, both the challenger $\mathscr{C}$ and the adversary $\mathscr{A}_1$ perform as defined in the security definition.

(i) Setup phase: the adversary $\mathscr{A}_1$ does as in the "Threat Model."

(ii) Query phase: the adversary $\mathscr{A}_1$ does as in the "Threat Model."

(iii) Challenge phase: the adversary $\mathscr{A}_1$ adaptively chooses the authorized user's attributes' set $\mathbb{S}$ ($\mathbb{A}(\mathbb{S}) = 0$). Then, $\mathscr{A}_1$ runs $\text{Sanitize}_{\text{P3S}}$ algorithm to generate the challenged signature σ^* for the challenged transaction m^*. Finally, the adversary $\mathscr{A}_1$ sends $(\mathbb{S}, m^*, \sigma^*)$ to $\mathscr{C}$.

(iv) Verify phase: the adversary $\mathscr{A}_1$ performs polynomial queries as in the query phase. Consider the adversary $\mathscr{A}_1$ has made L queries, and let $Q = \{\text{sk}_{\mathbb{S}}, \mathbb{S}, m_i, A_i, \mathbb{A}_i, \sigma_i\}_{i=1}^{[|Q|]}$ denote the set of information obtained through these queries. $\mathscr{C}$ runs $\text{Verify}_{\text{P3S}}(\text{pk}_{\text{P3S}}, \text{pk}_{\text{P3S}}^{\text{Sig}}, A, \mathbb{A}, m^*, \sigma^*)$ algorithm and outputs a bit b_0. If $b_0 = 1$, $\mathscr{C}$ checks whether there exists an $i \in [|Q|], \sigma^*)$ such that $\mathbb{A}(\mathbb{S}) = 0$. If there is such an i, the challenger $\mathscr{C}$ outputs $b_1 = 1$. Otherwise, $\mathscr{C}$ outputs $b_1 = 0$.

Suppose $b_1 = 1$, that is, the adversary $\mathscr{A}_1$ wins, we can say that the adversary $\mathscr{A}_1$ breaks the security of the policy-based sanitizable signature because the adversary's goal is to correctly generate the valid signature σ' for the transaction m^*. According to the security of the policy-based sanitizable signature (unforgeability), the probability of each adversary, who does not possess the attributes' set $\mathbb{S}$ such that $\mathbb{A}(\mathbb{S}) = 0$, is negligible.

Then, we define a game between the challenger $\mathscr{C}$ and the adversary $\mathscr{A}_2$.

Game 2: in Game 2, both the challenger $\mathscr{C}$ and the adversary $\mathscr{A}_2$ perform as defined in the security definition.

(i) Setup phase: the adversary $\mathscr{A}_2$ does as in the "Threat Model."

(ii) Query phase: the adversary $\mathscr{A}_2$ does as the adversary $\mathscr{A}_2$ in the query phase.

(iii) Challenge phase: the adversary $\mathscr{A}_2$ adaptively chooses the authorized user's attributes' set $\mathbb{S}$ ($\mathbb{A}(\mathbb{S}) = 1$). Then, $\mathscr{A}_2$ runs $\text{Sanitize}_{\text{P3S}}$ algorithm to generate the challenged signature σ^* for the challenged message m^* which does not contain all inadmissible blocks. Finally, the adversary $\mathscr{A}_2$ sends $(\mathbb{S}, m^*, \sigma^*)$ to $\mathscr{C}$.

(iv) Verify phase: the adversary $\mathscr{A}_2$ performs polynomial queries as in the query phase. Consider the adversary $\mathscr{A}_2$ has made L queries, and let $Q = \{\text{sk}_{\mathbb{S}}, \mathbb{S}, m_i, A_i, \mathbb{A}_i, \sigma_i\}_{i=1}^{[|Q|]}$ denote the set of information obtained through these queries. $\mathscr{C}$ runs $\text{Verify}_{\text{P3S}}(\text{pk}_{\text{P3S}}, \text{pk}_{\text{P3S}}^{\text{Sig}}, A, \mathbb{A}, m^*, \sigma^*)$ algorithm and outputs a bit b_0. If $b_0 = 1$, $\mathscr{C}$ checks whether there exists an $i \in [|Q|], m^*$ which does not contain all inadmissible blocks. If there is such an i, the challenger $\mathscr{C}$ outputs $b_1 = 1$. Otherwise, $\mathscr{C}$ outputs $b_1 = 0$.

Suppose $b_1 = 1$, that is, the adversary $\mathscr{A}_2$ wins, we can say that the adversary $\mathscr{A}_2$ breaks the security of the policy-based sanitizable signature because the adversary's goal is to correctly generate the valid signature σ' for the transaction m^*. According to the security of the policy-based sanitizable signature (immutability), the probability of each adversary, who redacts the inadmissible blocks, is negligible.

In conclusion, the proposed blockchain protocol achieves controlled redaction. In other words, only authorized users can redact the admissible portions of the transaction T_i. □

Theorem 3 (accountability). In the proposed blockchain protocol, trusted authority (group manager) can extract the identity of the originator of the transaction or the authorized user from any valid witness with nonnegligible probability.

Proof. We prove accountability by a sequence of games.

(i) Game 0: as Game 0 in [16].

(ii) Game 1: as Game 0, but we replace crs_Ω with the one generated by $(\text{crs}_\Omega, \tau) \leftarrow \text{SIM}_1(1^\kappa)$, i.e., the simulator SIM_1 takes the security parameter 1^κ as the input and then outputs $(\text{crs}_\Omega, \tau)$. Finally, the challenger $\mathscr{C}$ keeps the trapdoor τ and starts simulating all proofs.

(iii) Assume towards contradiction that the adversary behaves differently. We can then build an adversary $\mathscr{B}$ which breaks the zero-knowledge property of the

underlying proof system. The reduction works as follows. Our adversary $\mathscr{B}$ receives crs_Ω from its own challenger and embeds it into $\mathrm{PP}_{\mathrm{P3S}}$ and generates all other values honestly. All proofs are then generated using the oracle provided and embedded honestly. Then, whatever $\mathscr{A}$ outputs is also output by $\mathscr{B}$. $|\Pr[S_0] - \Pr[S_1]|$ is negligible, where $\Pr[S_X]$ denotes the advantage of the adversary in Game X. Note that this also means that all proofs are now simulated, even though they still prove valid statements.

(iv) Game 2: as Game 1, but we replace crs_Ω with the one generated by $(\mathrm{crs}_\Omega, \tau, \xi) \leftarrow \xi_1(1^\kappa)$, i.e., the simulator ξ_1 takes the security parameter 1^κ as the input and then outputs $(\mathrm{crs}_\Omega, \tau, \xi)$. Finally, the challenger $\mathscr{C}$ keeps the trapdoors τ and ξ. Let E_2 be the event that $\mathscr{A}$ can distinguish this replacement with non-negligible probability. Moreover, note that, by definition, crs_Ω is exactly distributed as in the prior hop.

(v) As we only keep one additional value, i.e., ξ, this is only an internal change. $|\Pr[S_1] - \Pr[S_2]|$ is negligible.

(vi) Game 3: as Game 2, but we abort if the adversary outputs valid $(\mathrm{pk}^*, m^*, \sigma^*)$ for which we cannot (as the holder of $\mathrm{sk}_{\mathrm{P3S}}^{\mathrm{Sig}}$) calculate pk which makes $\mathrm{Judge}_{\mathrm{P3S}}(\mathrm{pk}^*, \mathrm{pk}_{\mathrm{P3S}}^{\mathrm{Sig}}, \mathrm{pk}, \pi_{\mathrm{P3S}}, \sigma^*, m^*)$ output 0. Let this event be E_3.

If E_3 occurs, we have a bogus proof π contained in σ^* as it proves a false statement. Thus, $\mathscr{B}$ proceeds as in the prior game (doing everything honestly, but using simulated proofs and simulated crs_Ω) and can simply return the statement claimed to be proven by π and π itself. $|\Pr[S_2] - \Pr[S_3]|$ is negligible.

In conclusion, the proposed blockchain protocol achieves accountability. □

7. Performance

In this section, we first give functionality comparison among our redactable blockchain protocol and several related redactable blockchain protocols [11, 13–15]. Then, we analyze the computational burden of our redactable blockchain protocol through several experiments.

7.1. Functionality Comparison. We give functionality comparison among our scheme and the related schemes [11, 13–15]. As shown in Table 2, our scheme is the only one that satisfies all of the following properties: fine-grained access control, controllable edit, accountability, and supporting the redaction of both additional information and UTXO. The schemes in [11, 14] cannot support fine-grained access control. The scheme in [13] cannot effectively support harmful data deletion. All of these related redactable blockchain protocols cannot support controllable edit, accountability, and the editing of both additional information and UTXO.

7.2. Proof-of-Concept Implementation. To evaluate the practicality of the proposed blockchain protocol, we implement a full-fledged blockchain system in Python 3.5.3, which is carried out on a desktop with an Intel Core (TM) i5-4300 CPU @ 2.13 GHz and 8.0 GB RAM.

The blockchain system can achieve all the basic functionalities of Ethereum Enterprise. Separately, the blockchain system, including a subset of Ethereum Enterprise's script language, allows the authorized user to redact the transaction and the miner to delete the harmful data. We rely on the PoW consensus mechanism as Ethereum Enterprise does.

We evaluate the performance of the blockchain system for chain validation in different scenarios. In order to measure the cost time of chain validation, we validate chains containing different number of blocks and redaction transactions. A new chain is created and validated 50 times in each experiment, and the cost time of chain validation is the arithmetic mean of the run time of all runs. Each chain consists of up to 50,000 blocks, which approximate a one-year snapshot of the Ethereum Enterprise. Each block includes 1000 transactions (Figures 11–14).

(i) *Overhead Compared to the Immutable Blockchain.* In order to evaluate the overhead of the redactable blockchain protocol with no redactions performed compared to the immutable blockchain, in the series of experiments, the length of chains ranges from 10,000 to 50,000 blocks. As shown in Figure 11, the redactable blockchain protocol has only a more tiny overhead than the immutable blockchain. With the increase of the length of the chain, the overhead is smaller. The reason is that the only extra step of the redactable blockchain is to check if any votes are contained in the new block. The run time of this step is negligible compared to the time of chain validating when the length of the chain is larger enough.

(ii) *Overhead by the Number of Redactions.* In order to evaluate the overhead of the redactable blockchain protocol with the increasing number of redactions compared to the redactable blockchain with no redaction, in the series of experiments, the number of redactions ranges from 1000 to 5000. As shown in Figure 12, the overhead is linear in the number of redactions because we need to collect the votes for the redaction in the voting phase.

(iii) *Overhead by the Number of Removals.* In order to evaluate the overhead of the redactable blockchain protocol with the increasing number of removals compared to the redactable blockchain with no removal, in the series of experiments, the number of removals ranges from 1000 to 5000. As shown in Figure 13, the overhead is linear in the number of removals because the miner generating the new block needs to remove the harmful information from the previous block.

Table 2: Comparison of functionality among our redactable blockchain protocol and related protocols.

Protocols	Fine-grained access control	Controllable edit	Harmful data deletion	Accountability	Data type
[11]	×	×	√	×	Additional information
[13]	√	×	×	×	Additional information
[14]	×	×	√	×	Additional information
Ours	√	√	√	√	Additional information and UTXO

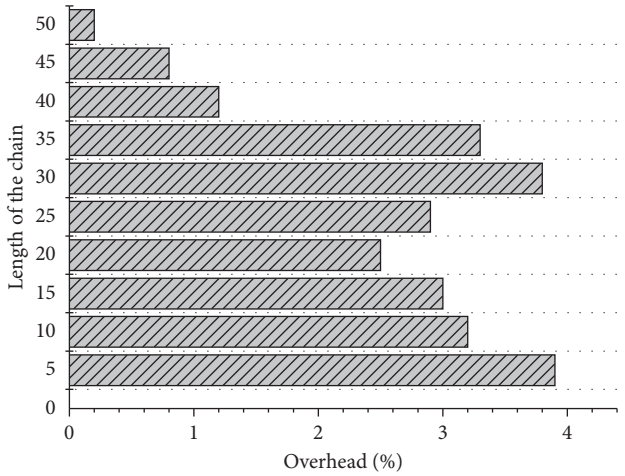


Figure 11: The overhead of the redactable blockchain without performing redaction compared to the immutable chain.

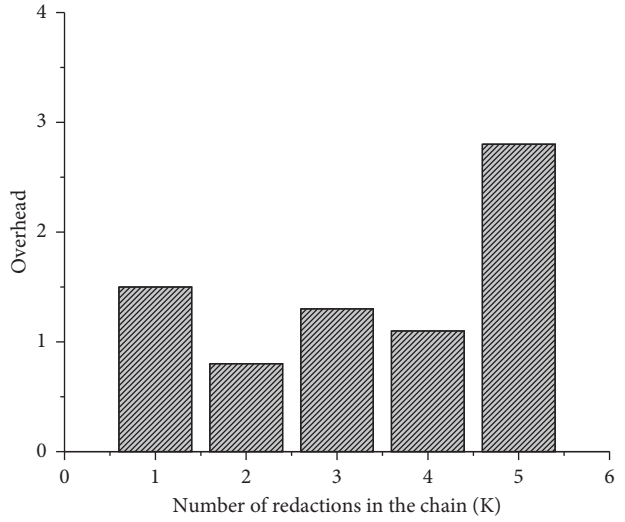


Figure 12: The overhead of the redactable blockchain for an increasing number of redactions compared to the redactable chain with no redaction.

(iv) *Overhead by the Voting Parameter ρ.* In order to evaluate the overhead of the redactable blockchain protocol with different voting periods, in the series of experiments, we set that the number of redactions is 1000, and the threshold ratio of the votes is $\rho \geq (1/2)$. As shown in Figure 14, the overhead is small and is most linear in the voting period.

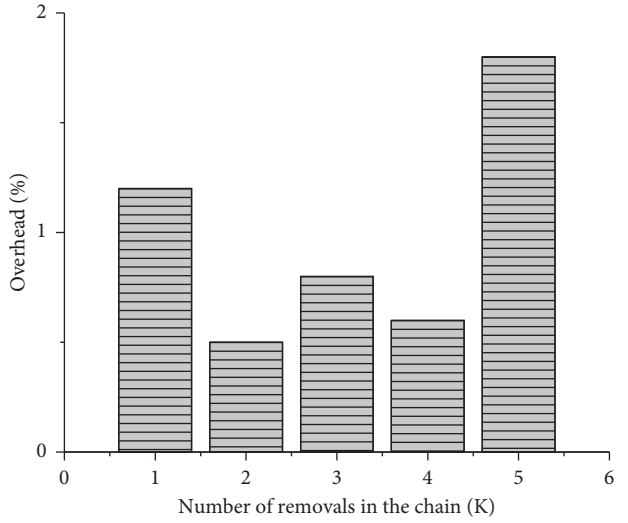


Figure 13: The overhead of the redactable blockchain for an increasing number of removals compared to the redactable chain with no removal.

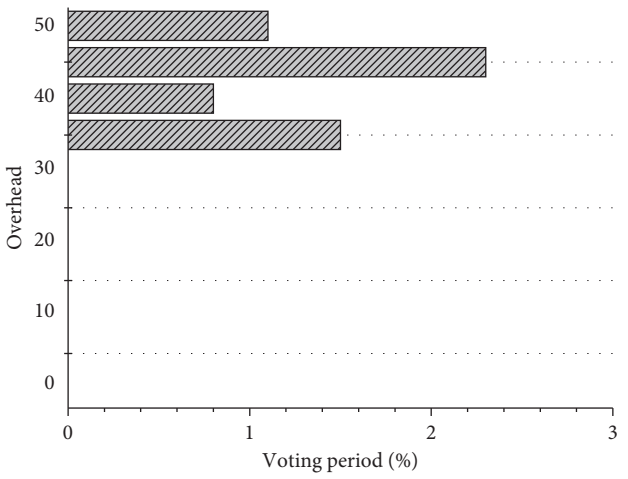


Figure 14: The overhead of the redactable blockchain for increasing voting periods compared to the redactable chain on a fixed voting period.

8. Conclusions

In this paper, we proposed a fine-grained and controllably redactable blockchain with harmful data forced removal. Our scheme not only supports the usual redaction of transactions but also the forced removal of harmful information in the blockchain. The originator of the transaction

could specify a fine-grained access control structure about who could redact the transaction and which portions of the transaction could be redacted. Any user could initiate a transaction that contains the index of the block which included harmful information without spending transaction fees. If the harmful information is contained in a block, it was forced to be deleted by the miner who created the new block. The user who provided the index of the block could receive the reward which was borne by the malicious user. The malicious user would be blacklisted if rewards were not paid within a period of time, and any transaction about the user would not be performed later. Furthermore, the scheme supported not only the redaction of additional data but also UTXO. Finally, we demonstrated that the scheme was secure and feasible via formal security analysis and proof-of-concept implementation.

Note that the proposed fine-grained and controllably redactable blockchain protocol with harmful data forced removal is suitable for permissioned blockchains, such as Hyperledger, Ethereum Enterprise, Ripple, and Quorum. There is another type of blockchain called permissionless blockchain, such as Bitcoin and Ethereum. Constructing the redactable permissionless blockchain protocol is a challenge and an interesting open problem. In our future work, we will also focus on designing more sophisticated solutions to the redactable permissionless blockchain protocol.

Appendix

A. Security Definition of the Improved Policy-Based Sanitizable Signature

In the following, we give the security definition of the improved policy-based sanitizable signature. Due to the limited space, we select several security aspects to highlight, and the rest of the security aspects can be seen in [16].

Definition 6. (unforgeability). In order to formally describe the unforgeability of the signature, we introduce a game between the challenger $\mathscr{C}$ and the adversary $\mathscr{A}$ to show how the adversary $\mathscr{A}$ is against the unforgeability of the signature. Trusted authority (group manager) is viewed as a challenger $\mathscr{C}$, and the unauthorized user is viewed as an adversary $\mathscr{A}$ in our security definition. This game includes the following phases:

(i) Setup phase: firstly, the challenger $\mathscr{C}$ runs the $\text{ParGen}_{\text{P3S}}$ and $\text{Setup}_{\text{P3S}}$ algorithm to generate the public parameters PP_{P3S} and the master private/public key pair $(\text{sk}_{\text{P3S}}, \text{pk}_{\text{P3S}})$. Then, $\mathscr{C}$ holds the master private key sk_{P3S} locally. Finally, $\mathscr{C}$ sends the master public key pk_{P3S} and the public parameters PP_{P3S} to the adversary $\mathscr{A}$.

(ii) Query phase:

(a) $\text{KGenSan}_{\text{P3S}}$ queries: the adversary $\mathscr{A}$ queries sanitizer's private/public key pair for the public parameters PP_{P3S}. $\mathscr{C}$ runs $\text{KGenSan}_{\text{P3S}}$ algorithm and returns the private/public key pair (x_2, y_2) to $\mathscr{A}$.

(b) Sign queries: the adversary $\mathscr{A}$ queries the signature for the master public key pk_{P3S}, the signature for the message m, the set of admissible blocks A, and the access structure $\mathbb{A}$. $\mathscr{C}$ runs $\text{KGenSig}_{\text{P3S}}$ to generate the signing key and then runs Sign algorithm to produce the signature σ. Finally, $\mathscr{C}$ returns the signature σ to $\mathscr{A}$.

(c) $\text{AddSan}_{\text{P3S}}$ queries: the adversary $\mathscr{A}$ queries the sanitizer's attribute key for sk_{P3S}, $\text{pk}_{\text{P3S}}^{\text{San}}$, and the attributes' set $\mathbb{S}$ such that $\mathbb{A}(\mathbb{S}) = 0$. $\mathscr{C}$ runs $\text{AddSan}_{\text{P3S}}$ algorithm and returns the sanitizer's attribute key $\text{sk}_{\mathbb{S}} \leftarrow (\sigma_{\text{sk}_{\mathbb{S}}}, \text{sk}'_{\mathbb{S}})$ to $\mathscr{A}$.

(d) $\text{Verify}_{\text{P3S}}$ queries: the adversary $\mathscr{A}$ queries the verification result for pk_{P3S}, $\text{pk}_{\text{P3S}}^{\text{Sig}}$, σ, and m. $\mathscr{C}$ runs $\text{Verify}_{\text{P3S}}$ algorithm and returns the result to $\mathscr{A}$.

(e) $\text{Sanitize}_{\text{P3S}}$ queries: the adversary $\mathscr{A}$ queries the sanitizable signature for pk_{P3S}, $\text{pk}_{\text{P3S}}^{\text{Sig}}$, $\text{sk}_{\text{P3S}}^{\text{San}}$, $\text{sk}_{\mathbb{S}}$, m, σ, and m'. $\mathscr{C}$ runs $\text{Sanitize}_{\text{P3S}}$ algorithm and returns the new signature σ' to $\mathscr{A}$.

(f) $\text{Proof}_{\text{P3S}}$ queries: the adversary $\mathscr{A}$ queries $(\pi_{\text{P3S}}, \text{pk})$ for pk_{P3S}, $\text{sk}_{\text{P3S}}^{\text{Sig}}$, σ, and m. $\mathscr{C}$ runs $\text{Proof}_{\text{P3S}}$ algorithm and returns $(\pi_{\text{P3S}}, \text{pk})$ to $\mathscr{A}$.

(g) $\text{Judge}_{\text{P3S}}$ queries: the adversary $\mathscr{A}$ queries the judge result for pk_{P3S}, $\text{sk}_{\text{P3S}}^{\text{Sig}}$, σ, and m. $\mathscr{C}$ runs $\text{Judge}_{\text{P3S}}$ algorithm and returns the result to $\mathscr{A}$.

(iii) Challenge phase: the adversary $\mathscr{A}$ adaptively chooses the authorized user's attributes' set $\mathbb{S}$ $(\mathbb{A}(\mathbb{S}) = 0)$. Then, $\mathscr{A}$ runs $\text{Sanitize}_{\text{P3S}}$ algorithm to generate the challenged signature σ^* for the challenged message m^*. Finally, the adversary $\mathscr{A}$ sends $(\mathbb{S}, m^*, \sigma^*)$ to $\mathscr{C}$.

(iv) Verify phase: the adversary $\mathscr{A}$ performs polynomial queries as in the query phase. Consider the adversary $\mathscr{A}$ has made L queries, and let $Q = \{\text{sk}_{\mathbb{S}}, \mathbb{S}, m_i, A_i, \mathbb{A}_i, \sigma_i\}_{i=1}^{[|Q|]}$ denote the set of information obtained through these queries. $\mathscr{C}$ runs $\text{Verify}_{\text{P3S}}(\text{pk}_{\text{P3S}}, \text{pk}_{\text{P3S}}^{\text{Sig}}, A, \mathbb{A}, m^*, \sigma^*)$ algorithm and outputs a bit b_0. If $b_0 = 1$, $\mathscr{C}$ checks whether there exists an $i \in [|Q|], \sigma^*$ such that $\mathbb{A}(\mathbb{S}) = 0$. If there is such an i, the challenger $\mathscr{C}$ outputs $b_1 = 1$. Otherwise, $\mathscr{C}$ outputs $b_1 = 0$.

We say that the adversary $\mathscr{A}$ wins if $b_1 = 1$. In the above game, we want to show that the adversary $\mathscr{A}$, who does not possess the attributes' set $\mathbb{S}$ such that $\mathbb{A}(\mathbb{S}) = 0$, should not generate the new valid signature. The adversary's goal is to correctly generate the valid signature σ' for the message m^*. We set the advantage of a polynomial-time adversary $\mathscr{A}$ in this game to be $\Pr[b_1 = 1]$. We say the proposed scheme satisfies the unforgeability of the signature if for any polynomial-time adversary $\mathscr{A}$, $\Pr[b_1 = 1] < (1/\text{poly}(n))$ for sufficiently large n, where poly stands for a polynomial function.

Definition 7. (immutability). In order to formally describe the immutability of the signed data, we introduce a game between the challenger $\mathscr{C}$ and the adversary $\mathscr{F}$ to show how the adversary $\mathscr{F}$ is against the immutability of the signed

data. Trusted authority (group manager) is viewed as a challenger $\mathscr{C}$, and the authorized sanitizer is viewed as an adversary $\mathscr{F}$ in our security definition. This game includes the following phases:

(i) Setup phase: firstly, the challenger $\mathscr{C}$ runs the $\text{ParGen}_{\text{P3S}}$ and $\text{Setup}_{\text{P3S}}$ algorithm to generate the public parameters PP_{P3S} and the master private/public key pair $(\text{sk}_{\text{P3S}}, \text{pk}_{\text{P3S}})$. Then, $\mathscr{C}$ holds the master private key sk_{P3S} locally. Finally, $\mathscr{C}$ sends the master public key pk_{P3S} and the public parameters PP_{P3S} to the adversary $\mathscr{F}$.

(ii) Query phase:

(a) $\text{KGenSan}_{\text{P3S}}$ queries: the adversary $\mathscr{F}$ queries sanitizer's private/public key pair for the public parameters PP_{P3S}. $\mathscr{C}$ runs $\text{KGenSan}_{\text{P3S}}$ algorithm and returns the private/public key pair (x_2, y_2) to $\mathscr{F}$.

(b) Sign queries: the adversary $\mathscr{F}$ queries the signature for the master public key pk_{P3S}, the signature for the message m, the set of admissible blocks F, and the access structure $\mathbb{A}$. $\mathscr{C}$ runs $\text{KGenSig}_{\text{P3S}}$ to generate the signing key and then runs Sign algorithm to produce the signature σ. Finally, $\mathscr{C}$ returns the signature σ to $\mathscr{F}$.

(c) $\text{AddSan}_{\text{P3S}}$ queries: the adversary $\mathscr{F}$ queries the sanitizer's attribute key for sk_{P3S}, $\text{pk}_{\text{P3S}}^{\text{San}}$, and the attributes' set $\mathbb{S}$ such that $\mathbb{A}(\mathbb{S}) = 1$. $\mathscr{C}$ runs $\text{AddSan}_{\text{P3S}}$ algorithm and returns the sanitizer's attribute key $\text{sk}_{\mathbb{S}} \leftarrow (\sigma_{\text{sk}_{\mathbb{S}}}, \text{sk}_{\mathbb{S}}')$ to $\mathscr{F}$.

(d) $\text{Verify}_{\text{P3S}}$ queries: the adversary $\mathscr{F}$ queries the verification result for pk_{P3S}, $\text{pk}_{\text{P3S}}^{\text{Sig}}$, σ, and m. $\mathscr{C}$ runs $\text{Verify}_{\text{P3S}}$ algorithm and returns the result to $\mathscr{F}$.

(e) $\text{Sanitize}_{\text{P3S}}$ queries: the adversary $\mathscr{F}$ queries the sanitizable signature for pk_{P3S}, $\text{pk}_{\text{P3S}}^{\text{Sig}}$, $\text{sk}_{\text{P3S}}^{\text{San}}$, $\text{sk}_{\mathbb{S}}$, m, σ, and m'. $\mathscr{C}$ runs $\text{Sanitize}_{\text{P3S}}$ algorithm and returns the new signature σ' to $\mathscr{F}$.

(f) $\text{Proof}_{\text{P3S}}$ queries: the adversary $\mathscr{F}$ queries $(\pi_{\text{P3S}}, \text{pk})$ for pk_{P3S}, $\text{sk}_{\text{P3S}}^{\text{Sig}}$, σ, and m. $\mathscr{C}$ runs $\text{Proof}_{\text{P3S}}$ algorithm and returns $(\pi_{\text{P3S}}, \text{pk})$ to $\mathscr{F}$.

(g) $\text{Judge}_{\text{P3S}}$ queries: the adversary $\mathscr{F}$ queries the judge result for pk_{P3S}, $\text{sk}_{\text{P3S}}^{\text{Sig}}$, σ, and m. $\mathscr{C}$ runs $\text{Judge}_{\text{P3S}}$ algorithm and returns the result to $\mathscr{F}$.

(iii) Challenge phase: the adversary $\mathscr{F}$ adaptively chooses the authorized user's attributes' set $\mathbb{S}$ $(\mathbb{A}(\mathbb{S}) = 1)$. Then, $\mathscr{F}$ runs $\text{Sanitize}_{\text{P3S}}$ algorithm to generate the challenged signature σ^* for the challenged message m^* which does not contain all inadmissible blocks. Finally, the adversary $\mathscr{F}$ sends $(\mathbb{S}, m^*, \sigma^*)$ to $\mathscr{C}$.

(iv) Verify phase: the adversary $\mathscr{F}$ performs polynomial queries as in the query phase. Consider the adversary $\mathscr{F}$ has made L queries, and let $Q = \{\text{sk}_{\mathbb{S}}, \mathbb{S}, m_i, A_i, \mathbb{A}_i, \sigma_i\}_{i=1}^{[|Q|]}$ denote the set of information obtained through these queries. $\mathscr{C}$ runs $\text{Verify}_{\text{P3S}}(\text{pk}_{\text{P3S}}, \text{pk}_{\text{P3S}}^{\text{Sig}}, A, \mathbb{A}, m^*, \sigma^*)$ algorithm and outputs a bit b_0. If $b_0 = 1$, $\mathscr{C}$ checks whether there exists an $i \in [|Q|], m^*$ which does not contain all inadmissible blocks. If there is such an i, the challenger $\mathscr{C}$ outputs $b_1 = 1$. Otherwise, $\mathscr{C}$ outputs $b_1 = 0$.

We say that the adversary $\mathscr{F}$ wins if $b_1 = 1$. In the above game, we want to show that the adversary $\mathscr{F}$, who redacts the inadmissible blocks, should not generate the new valid signature. The adversary's goal is to correctly generate the valid signature σ' for the message m^*. We set the advantage of a polynomial-time adversary $\mathscr{F}$ in this game to be $\Pr[b_1 = 1]$. We say the proposed scheme satisfies the unforgeability of the signature if for any polynomial-time adversary $\mathscr{F}$, $\Pr[b_1 = 1] < (1/\text{poly}(n))$ for sufficiently large n, where poly stands for a polynomial function.

Definition 8. (traceability). We say an improved policy-based sanitizable signature supports traceability if the trusted authority (group manager) can extract signer's identity from any valid signature with nonnegligible probability.

B. Security Analysis of the Improved Policy-Based Sanitizable Signature

In this section, we analyze the security of the improved policy-based sanitizable signature in terms of unforgeability, immutability, and traceability.

Theorem 4 (unforgeability). Any PPT adversaries can forge a policy-based sanitizable signature for some message with negligible probability.

Proof. To prove unforgeability, we use a sequence of games:

(i) Game 0: as Game 0 in [16].

(ii) Game 1: as Game 0, but we replace crs_{Ω} with the one generated by $(\text{crs}_{\Omega}, \tau) \leftarrow \text{SIM}_1(1^{\kappa})$, i.e., the simulator SIM_1 takes the security parameter 1^{κ} as the input and then outputs $(\text{crs}_{\Omega}, \tau)$. Finally, the challenger $\mathscr{C}$ keeps the trapdoor τ and starts simulating all proofs.

Assume towards contradiction that the adversary behaves differently. We can then build an adversary $\mathscr{B}$ which breaks the zero-knowledge property of the underlying proof system. The reduction works as follows. Our adversary $\mathscr{B}$ receives crs_{Ω} from its own challenger and embeds it into PP_{P3S} and generates all other values honestly. All proofs are then generated using the oracle provided and embedded honestly. Then, whatever $\mathscr{A}$ outputs is also output by $\mathscr{B}$. $|\Pr[S_0] - \Pr[S_1]|$ is negligible. Note that this also means that all proofs are now simulated, even though they still prove valid statements.

(iii) Game 2: as Game 1, but we replace crs_{Ω} with the one generated by $(\text{crs}_{\Omega}, \tau, \xi) \leftarrow \xi_1(1^{\kappa})$, i.e., the simulator ξ_1 takes the security parameter 1^{κ} as the input and then outputs $(\text{crs}_{\Omega}, \tau, \xi)$. Finally, the challenger $\mathscr{C}$

keeps the trapdoors τ and ξ. Let E_2 be the event that $\mathcal{A}$ can distinguish this replacement with non-negligible probability. Moreover, note that, by definition, crs_Ω is exactly distributed as in the prior hop.

As we only keep one additional value, i.e., ξ, this is only an internal change. $|\Pr[S_1] - \Pr[S_2]|$ is negligible.

(iv) Game 3: as Game 2, but we abort if the adversary was able to generate a signature σ_m^* on a string never generated by the signing oracle. Let this event be E_3.

Assume, towards contradiction, that event E_3 occurs. We can then construct an adversary $\mathcal{B}$ which breaks the unforgeability of the underlying signature scheme, namely, $\mathcal{B}$ receives pk of the signature scheme. This is embedded in pk'_Σ, while all other values are generated as in Game 2. All oracles are simulated honestly, but $\mathrm{Sign}'_{\mathrm{P3S}}$. The only change is, however, that the generation of each σ_m is outsourced to the signature generation oracle. Then, whenever E_3 occurs, $\mathcal{B}$ can return $((\mathrm{pk}_{\mathrm{P3S}}, \mathrm{pk}^{\mathrm{Sig}}_{\mathrm{P3S}}, A, H(i\|m_{!A}), h, \mathbb{A}), \sigma_m^*)$. These values can easily be compiled using $\mathcal{A}$'s output, i.e., (m^*, σ^*). Note that this already includes that the adversary cannot temper with A. $|\Pr[S_2] - \Pr[S_3]|$ is negligible.

(v) Game 4: as Game 3, but we abort if the adversary was able to generate (m^*, σ^*) for which m^* should not have been derivable. Let this event be E_4.

Assume, towards contradiction, that event E_4 occurs. We can then construct an adversary $\mathcal{B}$ which breaks the strong insider collision resistance of the used PCH, namely, $\mathcal{B}$ receives $\mathrm{pk}_{\mathrm{PCH}}$ of the PCH. This is embedded in $\mathrm{pk}_{\mathrm{P3S}}$, while all other values are generated as in Game 3. The GetSan oracle is simulated honestly. Calls to the $\mathrm{Sign}'_{\mathrm{P3S}}$ oracle are done honestly, but the hash is generated using the $\mathrm{Hash}'_{\mathrm{PCH}}$ oracle. Calls to the $\mathrm{AddSan}'_{\mathrm{P3S}}$ oracle are simulated as follows. If a key for a simulated sanitizer (obtained by a call to the GetSan oracle) is to be generated, it is rerouted to $\mathrm{KGen}''_{\mathrm{PCH}}$. If the adversary wants to get a key for itself, it is rerouted to the $\mathrm{KGen}'_{\mathrm{PCH}}$ oracle, and the answer is embedded honestly in the response. Sanitization requests are performed honestly (but simulated proofs), with the exception that adaptions for simulated sanitizers are done using the $\mathrm{Adapt}'_{\mathrm{pch}}$ oracle. So far, the distributions are equal. Then, whenever the adversary outputs (m^*, σ^*) such that the winning conditions are fulfilled, our reduction $\mathcal{B}$ can return $(m^*, r^*, m'^*, r'^*, h^*)$. The values can be compiled from (m^*, σ^*) and the transcript from the signing oracle (note that we already excluded that the adversary can temper with the hash h). $|\Pr[S_3] - \Pr[S_4]|$ is negligible.

(vi) Game 5: as Game 4, but we abort if the adversary was able to generate (m^*, σ^*) but has never made a call $\mathrm{AddSan}'_{\mathrm{P3S}}$. Let this event be E_5.

Assume, towards contradiction, that event E_5 occurs. We can then construct an adversary $\mathcal{B}$ which breaks the unforgeability of used Σ or the one-wayness of the used one-way function f, namely, $\mathcal{B}$ receives pk_Σ of Σ and f, and $f(x) = y$ from its own challenger. This is embedded in $\mathrm{pk}_{\mathrm{P3S}}$ (and, of course, the public parameters), while all other values are generated as in Game 4. y is embedded in $\mathrm{pk}^{\mathrm{Sig}}_{\mathrm{P3S}}$. For signing, the proofs are already simulated, and thus, x is not required to be known. For each call to $\mathrm{AddSan}_{\mathrm{P3S}}$ for keys for which the adversary knows the corresponding secret keys, $\mathcal{B}$ calls its signature oracle to obtain such a key. For simulated sanitizers, those signatures do not need to be obtained as the proofs are already simulated. Then, whenever the adversary outputs (m^*, σ^*), $\mathcal{B}$ extracts values $(x_1, x_2, sk_\Pi, \sigma')$. If $f(x_1) = y$, $\mathcal{B}$ can return x_1 to break the one-wayness of f. In the other case, $\mathcal{B}$ can return $((f(x_2), \mathrm{pk}_{\mathrm{P3S}}), \sigma')$ as its own forgery attempt for Σ. If extraction fails or a wrong statement was proven, SSE does not hold. A reduction is straightforward. $|\Pr[S_4] - \Pr[S_5]|$ is negligible. Now, the adversary can no longer win the unforgeability game; this game is computationally indistinguishable from the original game, which concludes the proof. □

Theorem 5 (immutability). For each PPT adversary, the advantage of generating valid signatures for altered immutable parts is negligible.

Proof. To prove immutability, we use a sequence of games:

(i) Game 0: as Game 0 in [16].

(ii) Game 1: as Game 0, and we abort if the adversary outputs $(\mathrm{pk}^*, m^*, \sigma^*)$ such that the winning conditions are met. Let this event be E_1.

Assume, towards contradiction, that event E_1 occurs. We can then build an adversary $\mathcal{B}$ which breaks the unforgeability of the used signature scheme, namely, we know that A (which also contains the length of the message and all nonmodifiable blocks along with their location), along with $\mathrm{pk}_{\mathrm{PCH}}$, is signed. As, however, by definition, the message m^* must be different from any derivable message, A w.r.t. $\mathrm{pk}_{\mathrm{PCH}}$ was never signed in this regard. Thus, $(\mathrm{pk}^*, \mathrm{pk}^{\mathrm{Sig}}_{\mathrm{P3S}}, A^*, H^*(i\|m_{!A}), h^*, \mathbb{A}^*)$ was never signed by the signer.

Constructing a reduction $\mathcal{B}$ is now straightforward. Our reduction $\mathcal{B}$ receives the public key pk'_Σ (along with the public parameters) from its own challenger. This public key is embedded as pk'_Σ. All other values are generated honestly. If a signature σ_m is to be generated, $\mathcal{B}$ asks its own oracle to generate that signature, embedding it into the response $\mathcal{A}$ receives. At some point, $\mathcal{A}$ returns $(\mathrm{pk}^*, m^*, \sigma^*)$. The forgery can be extracted as described above. $|\Pr[S_0] - \Pr[S_1]|$ is negligible. We stress that, by construction, a sanitizer always exists. Now, the adversary can no longer win the immutability game; this game is computationally indistinguishable from the original game, which concludes the proof. □

Theorem 6 (traceability). Trusted authority (group manager) can extract the identity of the originator of the

transaction or the authorized user from any valid witness with nonnegligible probability.

Proof. We prove traceability by a sequence of games:

(i) Game 0: as Game 0 in [16].

(ii) Game 1: as Game 0, but we replace crs_Ω with the one generated by $(\mathrm{crs}_\Omega, \tau) \leftarrow \mathrm{SIM}_1(1^\kappa)$, i.e., the simulator SIM_1 takes the security parameter 1^κ as the input and then outputs $(\mathrm{crs}_\Omega, \tau)$. Finally, the challenger $\mathscr{C}$ keeps the trapdoor τ and starts simulating all proofs.

Assume towards contradiction that the adversary behaves differently. We can then build an adversary $\mathscr{B}$ which breaks the zero-knowledge property of the underlying proof system. The reduction works as follows. Our adversary $\mathscr{B}$ receives crs_Ω from its own challenger and embeds it into $\mathrm{PP}_{\mathrm{P3S}}$ and generates all other values honestly. All proofs are then generated using the oracle provided and embedded honestly. Then, whatever $\mathscr{A}$ outputs is also output by $\mathscr{B}$. $|\Pr[S_0] - \Pr[S_1]|$ is negligible. Note that this also means that all proofs are now simulated, even though they still prove valid statements.

(iii) Game 2: as Game 1, but we replace crs_Ω with the one generated by $(\mathrm{crs}_\Omega, \tau, \xi) \leftarrow \xi_1(1^\kappa)$, i.e., the simulator ξ_1 takes the security parameter 1^κ as the input and then outputs $(\mathrm{crs}_\Omega, \tau, \xi)$. Finally, the challenger $\mathscr{C}$ keeps the trapdoors τ and ξ. Let E_2 be the event that $\mathscr{A}$ can distinguish this replacement with nonnegligible probability. Moreover, note that, by definition, crs_Ω is exactly distributed as in the prior hop.

As we only keep one additional value, i.e., ξ, this is only an internal change. $|\Pr[S_1] - \Pr[S_2]|$ is negligible.

(iv) Game 3: as Game 2, but we abort if the adversary outputs valid $(\mathrm{pk}^*, m^*, \sigma^*)$ for which we cannot (as the holder of $\mathrm{sk}_{\mathrm{P3S}}^{\mathrm{Sig}}$) calculate pk which makes $\mathrm{Judge}_{\mathrm{P3S}}(\mathrm{pk}^*, \mathrm{pk}_{\mathrm{P3S}}^{\mathrm{Sig}}, \mathrm{pk}, \pi_{P3S}, \sigma^*, m^*)$ output 0. Let this event be E_3.

If E_3 occurs, we have a bogus proof π contained in σ^* as it proves a false statement. Thus, $\mathscr{B}$ proceeds as in the prior game (doing everything honestly, but using simulated proofs and simulated crs_Ω) and can simply return the statement claimed to be proven by π and π itself. $|\Pr[S_2] - \Pr[S_3]|$ is negligible. □

Data Availability

We thank the authors of [14] for providing their implementation to us. We emailed Dominic Deuber and Bernardo Magri and obtained the source code for their scheme named "Redactable Blockchain in the Permissionless Setting." [14] We then extended and improved the source code to implement our scheme. We cannot expose the source code of the scheme in [14] without the permission of its authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

We thank the authors of [14] for providing their implementation to us. This work was supported by the National Natural Science Foundation of China (Grant nos. U1536205, 61472084, 61972094, and 62032005), National Key Research and Development Program of China (Grant no. 2017YFB0802000), Shanghai Innovation Action Project (Grant no. 16DZ1100200), Shanghai Science and Technology Development Funds (Grant no. 16JC1400801), Shandong Provincial Key Research and Development Program of China (Grant nos. 2017CXGC0701 and 2018CXGC0701), and the Young Talent Promotion Project of Fujian Science and Technology Association.

References

[1] S. Nakamoto, "Bitcoin: a peer-to-peer electronic cash system," 2008, https://nakamotoinstitute.org/bitcoin/.

[2] F. Tschorsch and B. Scheuermann, "Bitcoin and beyond: a technical survey on decentralized digital currencies," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 2084–2123, 2016.

[3] C. Badertscher, P. Gaži, A. Kiayias, A. Russell, and V. Zikas, "Ouroboros genesis: composable proof-of-stake blockchains with dynamic availability," in *Proceedings of the ACM SIGSAC on Computer and Communications Security*, pp. 913–930, Toronto Canada, October 2018.

[4] L. Breidenbach, I. Cornell Tech, P. Daian, F. Tramer, and A. Juels, "Enter the hydra: towards principled bug bounties and exploit-resistant smart contracts," in *Proceedings of the 27th USENIX Security*, Baltimore, MD, USA, August 2018.

[5] J. A. Garay, A. Kiayias, and N. Leonardos, "The bitcoin backbone protocol: analysis and applications," in *Proceedings of the EUROCRYPT 2015*, pp. 281–310, Sofia, Bulgaria, April 2015.

[6] A. Kiayias, A. Russell, B. David, and R. Oliynykov, "Ouroboros: a provably secure proof-of-stake blockchain protocol," in *Proceedings of the Annual International Cryptology Conference*, pp. 357–388, Santa Barbara, CA, USA, August 2017.

[7] L. D. Ibanez, K. O'Hara, and E. Simperl, "On blockchains and the general data protection regulation," *European Parliament Think Tank*, 2018, https://www.europarl.europa.eu/thinktank/en/document.html?reference=EPRS_STU%282019%29634445.

[8] Interpol, "Interpol cyber research identifies malware threat to virtual currencies," 2015, https://www.interpol.int/News-and-Events/News/2015/INTERPOL-cyber-research-identifies-malware-threat-to-virtual-currencies.

[9] G. Tziakouris, "Cryptocurrenciesła forensic challenge or opportunity for law enforcement? an interpol perspective," in *Proceedings of the IEEE Security & Privacy*, vol. 16, no. 4, pp. 92–94, San Francisco, CA, USA, May 2018.

[10] R. Matzutt, J. Hiller, M. Henze et al., "A quantitative analysis of the impact of arbitrary blockchain content on bitcoin," in *Proceedings of the 22nd FC*, Nieuwpoort, Curaçao, February 2018.

[11] G. Ateniese, B. Magri, D. Venturi, and E. Andrade, "Redactable blockchainCorCrewriting history in bitcoin and

friends," in *Proceedings of the Euro S & P 2017*, pp. 111–126, Paris, France, April 2017.

[12] J. Camenisch, D. Derler, S. Krenn, and H. C. P, hls, K. Samelin, and D. Slamanig, "Chameleon-hashes with ephemeral trapdoors," in *Proceedings of the IACR PKC*, Amsterdam, The Netherlands, March 2017.

[13] D. Derler, K. Samelin, D. Slamanig, and C. Striecks, "Fine-grained and controlled rewriting in blockchains: chameleon-hashing gone attribute-based," in *Proceedings of NDSS*, San Diego, CA, USA, February 2019.

[14] D. Deuber, B. Magri, and S. A. K. Thyagarajan, "Redactable blockchain in the permissionless setting," in *Proceedings of the SP2019*, pp. 19–23, San Francisco, CA, USA, May 2019.

[15] M. Florian, S. Henningsen, S. Beaucamp, and B. Scheuermann, "Erasing data from blockchain nodes," in *Proceedings of the EuroS&P*, pp. 367–376, Stockholm, Sweden, June 2019.

[16] K. Samelin and D. Slamanig, "Policy-based sanitizable signatures," in *Proceedings of the CT-RSA 2020*, pp. 538–563, San Francisco, CA, USA, February 2020.

[17] G. Ateniese, D. H. Chou, B. de Medeiros, and G. Tsudik, "Sanitizable signatures," in *Proceedings of the ESORICS 2005*, vol. 3679, pp. 159–177, Milan, Italy, September 2005.

[18] C. Brzuska, M. Fischlin, T. Freudenreich et al., "Security of sanitizable signatures revisited," in *Proceedings of the PKC 2009*, pp. 317–336, Irvine, CA, USA, March 2009.

[19] C. Brzuska, M. Fischlin, A. Lehmann, and Schr, D. der, "Unlinkability of sanitizable signatures," in *Proceedings of the PKC 2010*, pp. 444–461, Paris, France, May 2010.

[20] S. Canard, F. Laguillaumie, and M. Milhau, "Trapdoor sanitizable signatures and their application to content protection," in *Proceedings of the ACNS 2008*, pp. 258–276, New York, NY, USA, June 2008.

[21] J. Lai, X. Ding, and Y. Wu, "Accountable trapdoor sanitizable signatures," in *Proceedings of the ISPEC 2013*, pp. 117–131, Lanzhou, China, May 2013.

[22] K. Miyazaki, G. Hanaoka, and H. Imai, "Digitally signed document sanitizing scheme based on bilinear maps," in *Proceedings of the 2006 ACM Conference on Computer and Communications Security*, pp. 343–354, Alexandria, VA, USA, October 2006.

[23] T. H. Yuen, W. Susilo, J. K. Liu, and Y. Mu, "Sanitizable signatures revisited," in *Proceedings of the CANS 2008*, pp. 80–97, Hong-Kong, China, December 2008.

[24] S. Agrawal, S. Kumar, A. Shareef, and C. P. Rangan, "Sanitizable signatures with strong transparency in the standard model," in *Proceedings of the Inscrypt 2009*, pp. 93–107, Shanghai, China, October 2010.

[25] J. Ning, X. Huang, W. Susilo, K. Liang, X. Liu, and Y. Zhang, "Dual access control for cloud-based data storage and sharing," *IEEE Transactions on Dependable and Secure Computing*, vol. 99, p. 1, 2020.

[26] J. Ning, Z. Cao, X. Dong, K. Liang, L. Wei, and K. R. Choo, "CryptCloud+: secure and expressive data access control for cloud storage," *IEEE Transactions on Service Computing*, vol. 99, p. 1, 2018.

[27] J. Ning, Z. Cao, X. Dong, H. Ma, L. Wei, and K. Liang, "Auditable σ-times outsourced attribute-based encryption for access control in cloud computing," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 1, pp. 94–105, 2018.

[28] X. Liu, J. Ma, J. Xiong, J. Ma, and Q. Li, "Attribute based sanitizable signature scheme," *Journal of Communications*, vol. 34, pp. 148–155, 2013.

[29] L. Xu, X. Zhang, X. Wu, and W. Shi, "ABSS: an attribute-based sanitizable signature for integrity of outsourced database with public cloud," in *Proceedings of the 5th ACM Conference on Data and Application Security and Privacy*, pp. 167–169, San Antonio, TX, USA, March 2015.

[30] R. Mo, J. Ma, X. Liu, and Q. Li, "FABSS: attribute-based sanitizable signature for flexible access structure," in *Proceedings of the ICICS 2017*, Beijing, China, December 2017.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 6695964, 11 pages
https://doi.org/10.1155/2021/6695964

Research Article

CLE against SOA with Better Data Security Storage to Cloud 5G

Huige Wang,[1] **Xing Chang**,[2] **and Kefei Chen**[3,4]

[1]*Network and Security Department, Anhui Science and Technology University, Bengbu 233030, China*
[2]*Microsoft Asia-Pacific Technology Co., Ltd., Shanghai 200240, China*
[3]*Department of Mathematics, Hangzhou Normal University, Hangzhou 311121, China*
[4]*Westone Cryptologic Research Center, Beijing 100070, China*

Correspondence should be addressed to Huige Wang; whgexf@163.com

Received 25 December 2020; Revised 4 March 2021; Accepted 9 April 2021; Published 28 May 2021

Academic Editor: Yinghui Zhang

Cloud 5G and Cloud 6G technologies are strong backbone infrastructures to provide high data rate and data storage with low latency for preserving QoS (Quality of Service) and QoE (Quality of Experience) in applications such as driverless vehicles, drone-based deliveries, smart cities and factories, remote medical diagnosis and surgery, and artificial-intelligence-based personalized assistants. There are many techniques to support the aforementioned applications, but for privacy preservation of Cloud 5G, the existing methods are still not sufficient. Public key encryption (PKE) scheme is an important means to protect user data privacy in Cloud 5G. Currently, the most common PKE used in Cloud 5G is CCA or CPA secure ones. However, its security level maybe not enough. SOA security is a stronger security standard than CPA and CCA. Roughly speaking, PKE with SOA security means that the adversary is allowed to open a subset of challenger ciphertexts and obtains the corresponding encrypted messages and randomness, but the unopended messages and randomness remain secure in the rest of the challenger ciphertexts. Security against SOA in PKEs has been a research hotspot, especially with the wide discussion in Cloud 5G. We revisited the SOA-CLE and proposed a new security proof, which is more concise and user friendly to understand privacy preservation in Cloud 5G applications.

1. Introduction

Cloud 5G achieves high data transmission speed, large data storage, and low latency mobile communication. According to the inherent property of electromagnetic waves: the higher the frequency, the shorter the wavelength, so it tends to propagate like a straight line. From the last few years, we have witnessed a paradigm shift with a major focus on mission critical applications and ultra-reliable low latency applications (URLCC) such as AR/VR, autonomous vehicles, e-healthcare, smart education, and so on, the aim of which is to provide QoS (Quality of Service) and QoE (Quality of Experience) to the end users with high data storage and low latency. Starting from driverless vehicles and drone-based deliveries, smart cities and factories, remote medical diagnosis and surgery, and artificial-intelligence-based personalized assistants, there is enormous number of applications around us which require strong network backbone infrastructure for QoS and QoE preservation. Based on the above applications and the advantages in Cloud 5G, in the years to come, Cloud 5G and Cloud 6G technologies are expected to provide high data rate with low latency and large data storage for preserving QoS and QoE. Although there are many techniques in the literature which can resolve these issues, the existing methods are still not sufficient to privacy preservation in the application in Cloud 5G. Hence, secure protocols and encryption schemes are required to resolve the aforementioned issues. Public key encryption (PKE) scheme is an important means to protect user data privacy in Cloud 5G. Currently, the most commonly used means to protect user data privacy is CPA (chosen-plaintext attacks) or CCA (chosen-ciphertext attacks) secure PKEs where the latter provides the decryption queries and thus is stronger than the former. However, SOA is a stronger security standard than CCA because the SOA security allows additional opening partial ciphertexts. Specially, in particular, due to the inherent advantages of certificateless public key (CLE), it solves the certificate

management problem in the traditional public key cryptography and the key-escrow problem [8] in IBE schemes. Security against SOA in CLEs has been a research hotspot, especially with the wide discussion in Cloud 5G [11, 12]. In this paper, we focus on the research on the SOA secure CLE.

The definition of SOA was first proposed by Dwork et al. at FOCS99 [4], which is an important target to measure the security of PKE. SOA security mainly applies to multiple-user settings where a subset of the challenge ciphertexts is allowed to open for the adversary. From the opened ciphertexts, the adversary can get not only the message but also the randomness. The question that we want to solve is how to make the remaining unopened ciphertexts secure? Following Dwork's work, SOA secure IBE and public key encryption (PKE) with SOA security have been widely developed [2, 5, 7]. CLE is another form of public key encryption system. Compared with IBE and PKE, CLE has the advantages of removing the certificate management in PKI-based PKE and key escrow in IBE. However, the study on CLE with SOA security is still rare.

1.1. Motivation. In the CLE system, a user's private key is jointly generated by the KGC and the user. The user's public key is generated by using the secret value generated by itself instead of the identity information. Obviously, compared with PKI-based PKE (hereafter, we abbreviated "PKI-based PKE" as "PKE") and IBE, CLE removes the disadvantages that exist in both schemes, namely, the certificate transaction in PKE and key escrow in IBE. Due to the merits of this notion, many CLEs with various security models (e.g., IND-CPA [9] and IND-CCA [1, 13]) were presented. As in PKE and IBE settings, implementing SOA security in CLE is also important. However, the particular security model makes constructing CLEs with SOA security more intractable. With more and more applications for CLE (such as cloud computing), implementing SOA security in CLE becomes more and more critical. In 2016, Wang et al. proposed an SOA secure CLE [14] under the standard DDH assumptions where the scheme is user friendly in construction and more efficient in practical applications. Recently, the relative discussions about Cloud 5G have become a new research focus, especially its data security and privacy protection. Due to the notable efficiency and security level, SOA secure CLE has been regarded as one of the most practical candidate encryption algorithms for Cloud 5G. However, we find that there are still some disadvantages needed to avoid such as complex security proof and obscure proof process. Based on this, we revisited the scheme in [14] and improved the security proof to make it more concise and easier to understand.

1.2. Reviewing the Contribution in [14]. In the scheme of [3], the authors proposed a one-sided publicly opening identity-based encryption scheme (1SPO-IBE) and, based on which, constructed an IBE scheme with SOA security. Adopting the similar method, the authors in [14] resolved the SOA security in CLE. More concretely, they first proposed a one-sided publicly opening certificateless encryption scheme (1SPO-CLE). Then, based on the proposed 1SPO-CLE, they presented a CLE scheme that is SOA secure in the case of two-type adversary model (i.e., CLE security model where an adversary refers to a user who is granted the ability to change the public key but does not know the master key; another one means the malicious KGC, who is not granted the ability to change the public key but knows the master secret key). The core idea is that we first combined one-bit CLE and 1SPO to generate a 1SPO-CLE with IND-CPA security in the CLE settings and then showed that a multi-bit CLE scheme with SOA security can be constructed from the 1SPO-CLE scheme under the one-time signature and CDH assumptions.

1.3. Revisiting the Reduction from SOA to CPA in [14]. In [14], the authors constructed an IND-CPA secure 1SPO-CLE scheme by combining the 1SPO and one-bit CLE scheme. A CLE scheme that encrypts 1 bit messages is called 1SPO if it is possible, given the public parameter par, public key PK_{id}, and the ciphertext c that encrypts message 0 with the randomness r to efficiently open the ciphertext c into another randomness used to encrypt message 1. In particular, the opening process is required to be done without any secret information. Furthermore, they proved that if the 1 bit 1SPO-CLE is IND-CPA secure, then the multi-bit CLE from it is SOA secure. Specifically, the encryption process is performed as follows. If the message is 1, then the encryption process follows specific rules and the correctness of the resulted ciphertext can be checked with some secret information; otherwise, the generated ciphertext is sampled randomly and uniformly from the ciphertext space. As stated in [3], the domain used as the ciphertext space is also required to have the property of sampleability and invertible sampleability in order to guarantee that the resulted scheme has the property of 1SPO.

1.4. Revisiting 1SPO-CLE Construction in [14]. In [14], the authors gave a concrete construction based on one-time signature and CDH assumptions. Specifically, the 1SPO-CLE is designed as follows. Assume G_1 and G_2 are both sampleable and invertibly sampleable domains as in [3]. If the encrypted message is 1, then the encryption of 1 is processed as $c = (c_1, c_2, \sigma, \mathrm{svk}) \longleftarrow \mathrm{Encrypt}_{ex}(\mathrm{par}, \mathrm{PK}_{id}, 1)$, where par is the public parameter and PK_{id} is the public key, and the first two values c_1, c_2 have certain structure and the value σ is a signature for certain medians generated in the encryption, while the last value svk is the signature verification key. If the encrypted message is 0, then the first three elements of its encryption are all random. In particular, if c is an encryption of 1, then the medians u, K, and r can be always correctly recovered from c with the private key SK_{id}. Then, using these medians and the output of the equations $\mathrm{sign} \cdot \mathrm{Ver}(svk, \sigma, u, id) = 1$ and $u^{x_{id}} = \mathrm{PK}_{id}^{H_2(K \| \mathrm{PK}_{id} \| id)}$, the decryption algorithm $\mathrm{Decryp}_{ex}(\mathrm{par}, c, \mathrm{SK}_{id})$ decides whether the ciphertext c encrypts 0 or 1, where x_{id} is the secret value.

1.5. Revisiting the Security Proof of IND-CPA [14]. In this paper, we revisited the IND-CPA security proof of the 1 bit 1SPO-CLE scheme. Since the security proof in [14] is long and unintelligible, we do not intend to describe the difference between their scheme and ours. Below, we will directly describe our proof ideas and proof process. IND-CPA security means that given a ciphertext, no PPT adversary could distinguish which bit has been encrypted even if the adversary has the ability to replace public key or knows the master key (i.e., type 1 adversary and type 2 adversary) in the SOA security game. We present the proof of IND-CPA security for our concrete construction (for 1SPO-CLE scheme) under the two types of attacks defined in CLE. Briefly, under type 1 attack (where the adversary is granted the ability to change the public key but does not know the master key), we reduce the IND-CPA security to the assumption of one-time signature, where the reduction (the adversary that breaks one-time signature) performs the simulation itself except that the signature part is constructed by querying its signing oracle. However, unfortunately, under type 2 attack (where the adversary knows the master key but cannot change the public key), when we try to complete the reduction from the IND-CPA security to the CDH assumption, some obstacles arise. Namely, in the construction of challenge ciphertext, since the value r, as the exponent part of the challenge g^r, is unknown to the CDH adversary, it results in that the $c_1 = rQ_{id^*}P + rP_{\text{pub}}$ part cannot be computed. Luckily, we find a way to solve this problem. Specifically, we do this by allowing the reduction algorithm (the CDH adversary) to query its CDH challenger to obtain c_1. Of course, to do this, we assume that computing r from rP is not easier than computing r from g^r. In fact, this can be done over the elliptic curve groups.

1.6. Other Related Work. We note that in the past few years, there emerged many remarkable SOA secure systems in PKE setting such as the schemes proposed by Bellare et al. [2], Fehr et al. [5], and Huang et al. [6]. Recently, SOA secure IBE also made rapid progress. In 2011, Bellare et al. [3] proposed two SO-CPA secure IBEs. In 2014, Lai et al. [10] proposed SO-CCA secure IBE using cross authentication codes. In 2016, Wang et al. proposed an SO-CPA secure CLE scheme [14] which avoids the problem of certificate management in PKE settings and key escrow in IBE settings. However, the security proof in [14] is complex and ambiguous.

1.7. Our Contribution. Our SO-CPA secure certificateless encryption scheme (CLE) is constructed based on the technique of one-sided public openability (1SPO) and one-bit CPA secure CLE. Specifically, by combining the techniques of 1SPO and one-bit CLE, we construct an IND-CPA secure 1SPO-CLE scheme. 1SPO means that given a system parameter par, public key PK_{id}, and a ciphertext c encrypting message 0 under randomness r, it enables to open the ciphertext c to another message and randomness pair $(1, r')$. This method is very challenging since the opening process does not need any secret key to participate in. Interestingly, by revisiting, we found that this method can provide us concise security proof in order to obtain the desired security. In particular, this design implies that 1 bit 1SPO-CLE with IND-CPA security implies multi-bit CLE with the same security. In more detail, the scheme is outlined as follows. If the encrypted message is 1, then its ciphertext preserves a certain structure and can be detected with some secret information. On the contrary, if the encrypted message is 0, its ciphertext takes on a random status and thus is not checkable due to its unstructured property. These properties described above are just what we need for revisiting the CLE with SO-CPA security in [14].

2. Preliminary

In the following, we give several assumptions used in this paper.

(i) $\text{sign.Skg}(1^\lambda)$: taking a security parameter 1^λ as input, this algorithm outputs a signature/verification key pair (ssk, svk).

$\text{sign.Sig}(\text{ssk}, m)$: on input signature key ssk and a message $m \in \mathcal{M}$, this algorithm outputs a signature σ.

$\text{sign.Ver}(\text{svk}, (m, \sigma))$: on input a verification key svk, a signature σ and a message m, this algorithm outputs 1, if σ is valid, and 0 otherwise.

Definition 1 (discrete logarithm assumption (DL)). Assume that G is a multiplicative group with prime order q and $g \in G$ is a generator. Given g, $y = g^a$, computing is difficult, where $a \longleftarrow_{\$} \{0, \ldots, q-1\}$. Formally, for all probabilistic polynomial time (short for PPT) adversary $\mathcal{A}$, there exists a negligible function negl such that $\text{Adv}^{\text{DL}}_{G,A}(\lambda) = \Pr[A(g, y) \longrightarrow a | g \in G, y = g^a, a \longleftarrow_{\$} \{0, \ldots, q-1\}] \leq \text{negl}(\lambda)$, where negl (λ) is a negligible function in the security parameter λ.

Definition 2 (computational Diffie–Hellman assumption). Assume that G is a cyclic group with prime order q and $g \in G$ is a generator. Given g, g^a, g^b, *computing* g^{ab} *is difficult, where* $a, b \longleftarrow_{\$} \{0, \ldots, q-1\}$. Formally, for all PPT adversary $\mathcal{A}$, there exists a negligible function negl such that: $\text{Adv}^{\text{CDH}}_{G,A}(\lambda) = \Pr[A(g, g^a, g^b) \longrightarrow g^{ab} | a, \quad b \longleftarrow_{\$} \{0, \ldots, q-1\}] \leq \text{negl}(\lambda)$.

Definition 3 (one-time signature). Let $\mathcal{M}$ be message space, $\mathcal{R}$ be randomness space, and $\mathcal{S}$ be the signature space. A signature scheme sign = (sign.Skg, sign.Sig, sign.Ver) consists of three (probabilistic) polynomial time algorithms:

We say that a message/signature pair (m, σ) is valid if for all λ, all $(\text{ssk}, \text{svk}) \longleftarrow \text{sign.Skg}(1^\lambda)$, all $m \in \mathcal{M}$, and all $\sigma \longleftarrow \text{sign.Sig}(\text{ssk}, m)$, the equation $\text{sign.Ver}(\text{svk}, (m, \sigma)) = 1$ holds.

We say that a signature scheme sign = (sign.Skg, sign.Sig, \\sign.Ver) is one-time unforgeable under chosen-message attack if for any PPT adversary $\mathcal{A}$,

the success probability of $\mathcal{A}$ in the following experiment (see Figure 1) is negligible.

2.1. Detailed Legend for Figure 1. This figure describes one-time unforgeability experiment for one-time signature denoted in Section 2, where an adversary and a challenger participate in the experiment and interact with each other. Specifically, in this experiment, the challenger first invokes the algorithm $(\text{ssk}, \text{svk}) \longleftarrow \text{sign.Skg}(1^\lambda)$ to generate a pair of signature key and verification key (ssk, svk). The signature key ssk is used to sign a message and the verification key svk is used to verify whether a given signature is valid. Given a verification key, the adversary outputs a message/signature forge pair (m^*, σ^*) with multiple times of signature queries to oracle $\mathcal{O}_{\text{ssk}}^{\text{sign.Sig}}(\cdot)$. When the message/signature forge does not belong to the queried items to oracle $\mathcal{O}_{\text{ssk}}^{\text{sign.Sig}}(\cdot)$ and the forge can verify, the experiment outputs 1 which denotes that the adversary wins the experiment. Particularly, the oracle $\mathcal{O}_{\text{ssk}}^{\text{sign.Sig}}(\cdot)$ means that when an adversary delivers a message m, the oracle returns a signature σ.

In the above experiment, we allow the adversary to query $\mathcal{O}_{\text{ssk}}^{\text{sign.Sig}}(\cdot)$ oracle only one time. Assume that the adversary output a message/signature pair satisfying $(m^*, \sigma^*) \neq (m, \sigma)$ and $\text{sign.Ver}(\text{svk}, (m^*, \sigma^*)) = 1$. Then, we say that the adversary gives a successful forge. Formally, the scheme sign is unforgeable, if there exists a negligible function negl such that

$$\text{Adv}_{\text{sign},\mathcal{A}}^{\text{otUF}}(\lambda) = \Pr\left[\text{Exp}_{\text{sign},\mathcal{A}}^{\text{otUF}}(\lambda) = 1\right]. \quad (1)$$

Definition 4 (efficiently sampleable and invertible domain 3). Here, we define two PPT randomized algorithms that are sampleable and invertible, respectively:

(i) (efficient sampling) We say that a domain D is efficiently sampleable if there exists a PPT algorithm Sample s.t. $x \longleftarrow \text{Sample}(D; R)$ is uniformly distributed over D for randomness $R \longleftarrow R_{\text{Sample}}$, where R_{Sample} is randomness space.

(ii) (efficient invertible sampling) We say that a domain D is efficiently invertible sampleable, if there exists a PPT invertible algorithm Sample^{-1} s.t. $\text{Sample}^{-1}(D, x)$ outputs R uniformly distributed over R_{Sample} for $\text{Sample}(D; R) = x$ and any $x \in D$.

Note that the Sample algorithm has sampling failure probability ζ if the sampling algorithm Sample outputs $\perp$ with probability at most ζ and invertible sampling failure probability θ if the invertible algorithm Sample^{-1} outputs $\perp$ with probability at most θ.

Definition 5 (one-sided public openability (1SPO)). A scheme has the 1SPO property if for a ciphertext $C = (c_0, c_1)$ which is the encryption result of 0 under identity ID and public key PK, where c_0 and c_1 are randomly distributed over an efficiently sampleable and invertible domain G w.r.t. algorithms Sample and Sample^{-1}, there exists an algorithm P Open To Zero $(\text{PK}, \text{ID}, C = (c_0, c_1))$ that can use the algorithm Sample^{-1} to open (c_0, c_1). Namely, $(R_0, R_1) \longleftarrow P\,\text{Open}(\text{PK}, \text{ID}, (c_0, c_1))$ with $R_0 \longleftarrow \text{Sample}^{-1}(G, c_0)$ and $R_1 \longleftarrow \text{Sample}^{-1}(G, c_1)$.

$\text{Exp}_{\text{sign},\mathcal{A}}^{\text{otUF}}(\lambda)$:
1. $(ssk, svk) \leftarrow \text{sign.Skg}(1^\lambda)$
2. $(m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathcal{O}_{ssk}^{\text{sign.Sig}}(\cdot)}(svk)$
3. when $(m^*, \sigma^*) \neq (m, \sigma)$ and $\text{sign.Ver}(svk, (m^*, \sigma^*)) = 1$, output 1.

$\mathcal{O}_{ssk}^{\text{sign.Sig}}(m)$:
Return $(m, \sigma) \leftarrow \text{sign.Sig}(ssk, m)$

Figure 1: OT-signature.

3. Extractable 1SPO-CLE

3.1. Extractable 1SPO-CLE. An extractable certificateless encryption consists of the following algorithms:

(i) *Setup*: the algorithm $\text{Setup}_{ex}(1^\lambda)$ takes a security parameter λ as input and outputs a master key msk and a public parameter par, where par defines an identity space ν and ciphertext space Space_c.

(ii) *Partial private key generation*: the algorithm $\text{ParPrivKeyGen}_{ex}(\text{par}, \text{id}, \text{msk})$ takes a public parameter par, an identity $\text{id} \in \text{Space}_{\text{id}}$, and a master key msk as input and outputs the partial private key d_{id}.

(iii) *Secret key generation*: the algorithm $\text{SecValGen}_{ex}(\text{par}, \text{id})$ takes an identity id and the public parameters par as input and outputs the secret value x_{id}.

(iv) *Private key generation*: the algorithm $\text{PrivKeyGen}_{ex}(\text{par}, d_{\text{id}}, x_{\text{id}})$ takes the public parameter par, a user's partial private key d_{id}, and secret value x_{id} as input and outputs the private key $\text{SK}_{\text{id}} = (d_{\text{id}}, x_{\text{id}})$.

(v) *Public key generation*: the algorithm $\text{PubKeyGen}_{ex}(\text{par}, x_{\text{id}})$ takes a public parameter par and a user's secret value x_{id} as input and outputs the user's public key PK_{id}.

(vi) *Encryption*: the algorithm $\text{Encrypt}_{ex}(\text{par}, m, \text{PK}_{\text{id}})$ takes a public parameter par, a message $m \in \{0, 1\}$, and a user's public key PK_{id} and returns the ciphertext c by using the defined algorithm if $m = 1$; otherwise, it returns c by sampling randomly from the ciphertext space.

(vii) *Decryption*: the algorithm $\text{Decrypt}_{ex}(\text{par}, c, \text{SK}_{\text{id}})$ takes a public parameter par, a ciphertext c, and a private key SK_{id} as input and outputs $m \in \{0, 1\}$.

(viii) *Correctness*: the correctness follows that in [14]; here we omitted it in order to save space.

Definition 6 (see [5] (1SPO-CLE)). An extractable 1SPO-CLE is a scheme with the property of one-sided public openability in the CLE setting and is associated with a PPT

public algorithm P Open To Zero, so that for all $(\text{par}, \text{msk}) \longleftarrow \text{Setup}_{ex}(1^k)$, $c \longleftarrow \text{Encrypt}_{ex}(\text{par}, 0, \text{PK}_{\text{id}})$, $\text{PK}_{\text{id}} \longleftarrow \text{PubKeyGen}_{ex}(\text{par}, x_{\text{id}})$, $x_{\text{id}} \longleftarrow \text{SecValGen}_{ex}(\text{par}, \text{id})$ and $\text{id} \in \text{Space}_{\text{id}}$, P Open To Zero $(\text{par}, \text{PK}_{\text{id}}, c)$ distributes uniformly at random over $\text{Coins}(\text{par}, \text{PK}_{\text{id}}, c, 0)$. Here, $\text{Coins}(\text{par}, \text{PK}_{\text{id}}, c, 0)$ represent the set of random coins $\{R | c = \text{Encrypt}_{ex}(\text{par}, 0, \text{PK}_{\text{id}}; R)\}$.

As described in [14], the multi-bit 1SPO-CLE can be constructed from 1 bit 1SPO-CLE. Since the concrete construction and security overlap with that in [14], here we do not dwell on it, but, for completeness, we describe it in Appendices A and B.

4. Proposed Extractable 1SPO-CLE

4.1. Construction. In this section, we describe the 1SPO-CPA secure 1-bit CLE scheme. We mainly focus on the following algorithms:

Setup. The algorithm $\text{Setup}_{ex}(1^\lambda)$ first takes a security parameter λ as input and then runs a group generator $\text{GR}(\lambda)$ to get a group description (G_1, G_2, e, q). Here, G_1 and G_2 are both groups of prime order q, G_1 is an additive group, and G_2 is a multiplicative group. We also notice that both G_1 and G_2 are efficiently sampleable and invertible domain associated with algorithms Sample and Sample^{-1} shown in [3]. e: $G_1 \times G_1 \longrightarrow G_2$ is a non-degenerate bilinear map, and P is a non-zero generator of G_1. Let H_1: $\{0,1\}^l \longrightarrow Z_q^*$, H_2: $G_1 \times G_2 \times \{0,1\}^l \longrightarrow Z_q^*$, H_3: $G_2^2 \longrightarrow G_1$ be three hash functions. Pick $s \longleftarrow_{\$} Z_q^*$, set master key msk: $= s$, and compute $P_{\text{pub}} = sP$ and $g = e(P, P) \in G_2$. Let sign = (sign.Skg, sign.Sig, sign.Ver) be one-time signature scheme with signature space G_2. Finally, the public parameter is set as $par := \{(G_1, G_2, e, q, P), P_{\text{pub}}, g\}$.

Partial Private Key Generation. The algorithm $\text{ParPrivKeyGen}_{ex}(\text{par}, \text{id}, \text{msk})$ first takes the public parameter par, an identity $\text{id} \in \{0,1\}^l$, and the master secret key msk as input and proceeds as follows. It computes the partial private key $d_{\text{id}} = (1/(s + H_1(\text{id})))P \in G_1$. This can be done since if q is large enough, the probability that the unlikely event $s + H_1(\text{id}) = 0 (\text{mod} q)$ happens is negligible.

Secret Key Value Generation. The algorithm $\text{SecValGen}_{ex}(\text{par}, \text{id})$ first takes the public parameters par and an identity id as input and then randomly selects a value $x_{\text{id}} \longleftarrow Z_q^*$ as the secret value.

Private Key Generation. The algorithm $\text{PrivKeyGen}_{ex}(\text{par}, d_{\text{id}}, x_{\text{id}})$ first takes the public parameter par, the partial private key d_{id}, and the secret value x_{id} as input and then returns $\text{SK}_{\text{id}} = (d_{\text{id}}, x_{\text{id}})$ as the private key.

Public Key Generation. The algorithm $\text{PubKeyGen}_{ex}(\text{par}, x_{\text{id}})$ first takes the public parameter par and the secret value x_{id} as input and then computes the public key $\text{PK}_{\text{id}} = g^{x_{\text{id}}}$.

Encryption. The algorithm $\text{Encrypt}_{ex}(\text{par}, m \in \{0,1\}, \text{PK}_{\text{id}})$ first takes the public parameter par, a message $m \in \{0,1\}$, and the public key PK_{id} as input. It then encrypts m as follows:

First, check whether $(\text{PK}_{\text{id}})^q \doteq 1_{G_2}$. If not, abort; otherwise, compute $(\text{ssk}, \text{svk}) \longleftarrow \text{sign.Skg}(1^\lambda)$ and proceed as follows.
If $m = 1$, pick $K \longleftarrow_{\$} G_1$, compute $r = H_2(K, \text{PK}_{\text{id}}, \text{id})$, $c_1 = rH_1(id)P + rP_{\text{pub}}$, $\sigma = \text{sign.Sig}(\text{ssk}, g^r, \text{id}) \in G_2$, and $c_2 = K + H_3(g^r, \text{PK}_{\text{id}}^r)$.
If $m = 0$, pick $c_1 \longleftarrow_{\$} \text{Sample}_{G_1}$, $c_2 \longleftarrow_{\$} \text{Sample}_{G_1}$, and $\sigma \longleftarrow_{\$} \text{Sample}_{G_2}$.
Finally, the ciphertext is set as $c = (c_1, c_2, \sigma, \text{svk})$.

Decryption. The algorithm $\text{Decrypt}_{ex}(\text{par}, c, \text{SK}_{\text{id}})$ takes the public parameter par, a ciphertext c, and a private key SK_{id} as input. To decrypt a ciphertext $c = (c_1, c_2, \sigma, \text{svk})$, firstly compute $u = e(c_1, d_{\text{id}}) = g^r$ and $K = c_2 - H_3(u, u^{x_{\text{id}}})$ and verify whether $\text{sign.Ver}(\text{svk}, \sigma, u, \text{id}) = 1$ holds; if not, outputs $\perp$; otherwise, verify whether $u^{x_{\text{id}}} = \text{PK}_{\text{id}}^{H_2(K, \text{PK}_{\text{id}}, \text{id})}$ holds; if so, set $m = 1$; otherwise, $m = 0$.

Correctness. If $c = (c_1, c_2, \sigma, \text{svk})$ is the encryption of 1, then the equations $u = e(c_1, d_{\text{id}}) = e(rH_1(\text{id})P + rP_{\text{pub}}, (1/(s + H_1(\text{id})))P) = g^r$, $K = c_2 - H_3(u, u^{x_{\text{id}}})$, $\text{sign.Ver}(\text{svk}, \sigma, u, \text{id}) = 1$, and $u^{x_{\text{id}}} = \text{PK}_{\text{id}}^{H_2(K, \text{PK}_{\text{id}}, \text{id})}$ hold, so the decryption always recovers 1. If $c = (c_1, c_2, \sigma, \text{svk})$ is the encryption of 0, since $c_1 \longleftarrow_{\$} \text{Sample}_{G_1}$, $c_2 \longleftarrow_{\$} \text{Sample}_{G_1}$ and $\sigma \longleftarrow_{\$} \text{Sample}_{G_2}$ are sampled uniformly and randomly. So, $\Pr[e(c_1, d_{\text{id}})^{x_{\text{id}}} = \text{PK}_{\text{id}}^r = \text{PK}_{\text{id}}^{H_2(K, \text{PK}_{\text{id}}, \text{id})}] \leq (1/q(\lambda))$ (we assume that $q(\lambda)$ is large enough which, in turn, results in a negligible quantity for $(1/q(\lambda))$).

4.2. Security

Theorem 1. *Assume the hash functions H_1, H_2, and H_3 are random oracles, and the scheme sign = (sign.Skg, sign.Sig, sign.Ver) is one-time signature scheme. Let Π' be extractable 1SPO-CLE scheme proposed in Section 4.1 and G_1 and G_2 be PR-sampleable (pseudorandom-sampleable) with negligible sampling failure probability. Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be any IND-CPA type 1 and type 2 adversaries against scheme Π', respectively, and are allowed to make polynomial times of queries to H_2 and H_3; then, the scheme Π' is IND-CPA secure under both type 1 adversary and type 2 adversary.*

Proof. We first prove that, for type 1 adversary, the security can be reduced to the security of one-time signature scheme *sign* and then prove that, for type 2 adversary, the security can be reduced to the computational Diffie–Hellman assumption (short for CDH). In the following, we describe the reduction between the adversary $\mathcal{A}_{sig}$ (which tries to break the one-time signature scheme) and the type 1 adversary $\mathcal{A}_1$ and the reduction between the adversary $\mathcal{A}_{cdh}$ (which tries to break the CDH assumption) and the type 2 adversary $\mathcal{A}_2$, respectively. □

4.2.1. Type 1 Adversary

Setup: the adversary $\mathcal{A}_{\text{sig}}$ (which has the signing verification key svk) first generates the public parameter $par := \{(G_1, G_2, e, q, P), P_{\text{pub}}, g\}$ and the master key s, where $P_{\text{pub}} = sP$ and $g = e(P, P) \in G_2$, and then sends the public parameter par to the adversary $\mathcal{A}_1$.

Partial private key query: on receiving the identity id, if id $\notin$ ChID, where ChID is the challenge identity set, the adversary $\mathcal{A}_{\text{sig}}$ invokes the partial private key generation algorithm to obtain the partial private key d_{id} and sends it to the adversary $\mathcal{A}_1$; otherwise it aborts. Concretely, the adversary $\mathcal{A}_{\text{sig}}$ first queries the random oracle H_1 to get Q_{id} and then computes $d_{\text{id}} = (1/(s + Q_{\text{id}}))P \in G_1$. Note that the oracle H_1 here is stateful and assume that all oracles in the following are stateful.

Pr*ivate key query*: on receiving the identity id, if id $\notin$ ChID, where ChID is the challenge identity set, $\mathcal{A}_{\text{sig}}$ first invokes the secret value generation algorithm and the partial private key generation algorithm to get the secret value x_{id} and the partial private key d_{id} and then sets the private key as $\text{SK}_{\text{id}} = (d_{\text{id}}, x_{\text{id}})$, i.e., $\text{SK}_{\text{id}} = (1/(s + H_1(\text{id}))P, x_{\text{id}})$; otherwise it aborts.

Public key query: on receiving the identity id, $\mathcal{A}_{\text{sig}}$ first invokes the secret value generation algorithm to get x_{id} and then computes the public key as $\text{PK}_{\text{id}} = g^{x_{\text{id}}}$.

Re*place public key query*: on receiving the identity id, $\mathcal{A}_{\text{sig}}$ replaces the original public key $\text{PK}_{\text{id}} = g^{x_{\text{id}}}$ with the new public key $\text{PK}'_{\text{id}} = g^{x_{\text{id}}\prime}$.

Challenge: on receiving the challenge identity id^* and the challenge message $m_0 = 0$, $m_1 = 1$ and the public key PK_{id^*}, the adversary $\mathcal{A}_{\text{sig}}$ computes challenge ciphertext as follows.

First flip a coin $b \longleftarrow_{\$} \{0, 1\}$ and then check whether $(\text{PK}_{\text{id}^*})^q \doteq 1_{G_2}$; if not, abort; otherwise, proceed as follows.

If $m_b = 1$, do the following steps.

(1) First, pick $K \longleftarrow_{\$} G_1$, and then for tuple $(K, \text{PK}_{\text{id}^*}, \text{id}^*)$, query oracle H_2 to get r.
(2) For id^*, query oracle H_1 to get Q_{id^*}.
(3) Compute g^r and $\text{PK}_{\text{id}^*}^r$, and then for $(g^r, \text{PK}_{\text{id}^*}^r)$, query oracle H_3 to get h.
(4) Compute $c_1 = rQ_{\text{id}^*}P + rP_{\text{pub}}$ and $c_2 = K + h$.
(5) For (g^r, id^*), query signature oracle to get σ.

If $m_b = 0$, pick $c_1 \longleftarrow_{\$} \text{Sample}_{G_1}$, $c_2 \longleftarrow_{\$} \text{Sample}_{G_1}$ and $\sigma \longleftarrow_{\$} \text{Sample}_{G_2}$ at random. Then, the final challenge ciphertext is set as $c = (c_1, c_2, \sigma, \text{svk})$.

From above, we can see that the adversary $\mathcal{A}_{\text{sig}}$ provides perfect simulation for $\mathcal{A}_1$. Now we do the following analysis.

Analysis: let the challenge ciphertext $c = (c_1, c_2, \sigma, \text{svk})$. In the experiment, since $\mathcal{A}_1$ does not know d_{id^*}, it cannot compute the value $u = g^r$. Assume $\mathcal{A}_1$ guess $u' = g^{r'}$ randomly. Then, by the one-time signature scheme sign, the verification equation $\text{sign.Ver}(\text{svk}, \sigma, u', \text{id}^*) = 1$ does not hold with overwhelming probability.

4.2.2. Type 2 Adversary

Setup: the adversary $\mathcal{A}_{\text{cdh}}$ (which has the challenge $(\text{PK}_{\text{id}^*}, u = g^r)$) first generates the public parameter $par := \{(G_1, G_2, e, q, P), P_{\text{pub}}, g\}$ and the master key s, where $P_{\text{pub}} = sP$ and $g = e(P, P) \in G_2$, and then sends the public parameter par to the adversary $\mathcal{A}_2$.

Pr*ivate key query*: in this phase, If id $\notin$ ChID, where ChID is the challenge identity set, the adversary $\mathcal{A}_{\text{cdh}}$ first invokes the secret value generation algorithm to get secret value x_{id} and computes partial private key d_{id}, and then sets the private key as $\text{SK}_{\text{id}} = (d_{\text{id}}, x_{\text{id}})$, i.e., $\text{SK}_{\text{id}} = (1/(s + H_1(\text{id}))P, x_{\text{id}})$.

Public key query: in this phase, if id $\notin$ ChID, the adversary $\mathcal{A}_{\text{cdh}}$ first invokes the secret value generation algorithm to get secret value x_{id} and then computes the public key as $\text{PK}_{\text{id}} = g^{x_{\text{id}}}$; otherwise it aborts.

Challenge: on receiving the challenge identity id^* and the challenge message $m_0 = 0$, $m_1 = 1$ and the public key PK_{id^*}, the adversary $\mathcal{A}_{\text{cdh}}$ computes challenge ciphertext as follows. First sample a random $b \longleftarrow_{\$} \{0, 1\}$, then check whether $(\text{PK}_{\text{id}^*})^q \doteq 1_{G_2}$; if not, abort; otherwise, compute $(\text{ssk}, \text{svk}) \longleftarrow \text{sign.Skg}(1^\lambda)$ and proceed as follows.

If $m_b = 1$, do the following steps.

(1) Pick $c_2 \longleftarrow_{\$} G_1$.
(2) For id^*, query oracle H_1 to get Q_{id^*}.
(3) Query the CDH challenger to get c_1, where c_1 is computed as $c_1 = rQ_{\text{id}^*}P + rP_{\text{pub}}$.
(4) For $(u = g^r, \text{id}^*)$, compute signature σ.
(5) Set the challenge ciphertext as $c = (c_1, c_2, \sigma, \text{svk})$.

From above, it is easy to see that we implicitly set $r = H_2(K, \text{PK}_{\text{id}^*}, \text{id}^*)$ for $K = c_2 - h$ and $h = H_3(g^r, \text{PK}_{\text{id}^*}^r)$. In addition, we require here that computing r from rP is not easier than computing r from g^r.

If $m_b = 0$, pick $c_1 \longleftarrow_{\$} \text{Sample}_{G_1}$, $c_2 \longleftarrow_{\$} \text{Sample}_{G_1}$, and $\sigma \longleftarrow_{\$} \text{Sample}_{G_2}$ at random.

Then, set the challenge ciphertext as $c = (c_1, c_2, \sigma, \text{svk})$.

From above, we can see that the adversary $\mathcal{A}_{\text{cdh}}$ provides perfect simulations for the adversary $\mathcal{A}_2$. Now we do the following analysis.

Analysis: let $c = (c_1, c_2, \sigma, \text{svk})$ be the challenge ciphertext. In the experiment, $\mathcal{A}_2$ knows $u = g^r$ and PK_{id^*}; by the CDH assumption, it is still difficult to compute $u^{x_{id^*}}$ and K to make the verification equation $u^{x_{\text{id}^*}} = \text{PK}_{\text{id}^*}^{H_2(K, \text{PK}_{\text{id}^*}, \text{id}^*)}$ hold.

This completes the proof of Theorem 1.

Table 1: Comparison in exponent, pairing, and security model.

	Exponent	Pairing	Security model	Need key escrow?	Simplified proof?
Scheme LoR [3]	14	5	SM	Yes	—
Scheme BBoR [3]	15	1	SM	Yes	—
Scheme [14]	6	2	ROM	No	No
Our scheme	6	2	ROM	No	Yes

5. Comparisons and Discussion

The authors in [14] first proposed an SOA secure certificateless encryption scheme. In this paper, we improved it to make the security proof more concise and user friendly. Although in [14], they gave an efficiency analysis, here, to make it easier to understand, we give a more detailed comparison with the existing similar schemes, especially with that in [3, 14]. The detailed comparison results are shown in Table 1. Similarly, in terms of complexity, we also just make comparisons among them on the cost of the additive and multiplicative operations, especially on the exponent and the pairing operations. In addition, we also compare them in "security model," "whether key escrow is needed," and "whether a simplified proof is provided." From this table, we can see that in [3], the first scheme requires 14 exponents and 5 pairings and the second scheme requires 15 exponents and 1 pairing, while in [14], the scheme only needs 6 exponents and 2 pairings. By comparison, we can see that our scheme not only realizes a simplified security proof but also obtains the same efficiency and security level as that of [14].

6. Result

As shown in Table 1, compared with the schemes in [3], our scheme is practical in real applications which is mainly reflected in the following 4 aspects: (1) our scheme can be instantiated from very standard assumption such as computational Diffie–Hellman; (2) the used one-time signature can be constructed from standard assumption such as one-way function; (3) the hash functions such as random oracles in our scheme are very easily run on a low-configured device; (4) our scheme has better efficiency as analyzed in Section 5. Specifically, our scheme has 8 exponents and 3 pairings less than that of the first scheme in [3] and has 1 pairing more than that of the second scheme in [3], respectively. In addition, compared with [14], our scheme has more concise and user-friendly security proof.

7. Conclusions

This paper proposed a certificateless public key encryption against selective opening attacks (SOA), which is suitable for the data storage in Cloud 5G environment. This scheme is proved secure in the ROM under the assumptions of CDH and security of one-time signature. The advantage of the scheme is that it eliminates both certificate management and key management in PKI-based PKE and IBE settings and is practical in Cloud 5G settings. Compared with [14], our scheme not only has more concise and user-friendly security proof but also achieves the same level of security, which strengthens the data security storage in Cloud 5G applications.

Appendix

A. How to Construct l-Bit 1SPO-CLE from 1-Bit 1SPO-CLE

Let $II = (\text{Setup}_{ex}, \text{ParPrivKeyGen}_{ex}, \text{SecValGen}_{ex}, \text{PrivKeyGen}_{ex}, \text{PubKeyGen}_{ex}, \text{Encrypt}_{ex}, \text{Decrypt}_{ex})$ be a 1 bit 1SPO-CLE scheme. An l-bit CLE scheme $II^l = (\text{Setup}^l_{ex}, \text{ParPrivKeyGen}^l_{ex}, \text{SecValGen}^l_{ex}, \text{PrivKeyGen}^l_{ex}, \text{PubKeyGen}^l_{ex}, \text{Encrypt}^l_{ex}, \text{Decrypt}^l_{ex})$ with message space $\{0,1\}^l$ is constructed as follows:

$$\begin{aligned} \text{Setup}^l_{ex} &= \text{Setup}_{ex}, \\ \text{ParPrivKeyGen}^l_{ex} &= \text{ParPrivKeyGen}_{ex}, \\ \text{SecValGen}^l_{ex} &= \text{SecValGen}_{ex}, \\ \text{PrivKeyGen}^l_{ex} &= \text{PrivKeyGen}_{ex}, \\ \text{PubKeyGen}^l_{ex} &= \text{PubKeyGen}_{ex}, \end{aligned} \tag{A.1}$$

where $c = c[1] \| \cdots \| c[l] \longleftarrow \text{Encrypt}^l_{ex}(\text{par}, \text{PK}_{\text{id}}, M \in \{0,1\}^l)$ such that $c[i] \longleftarrow \text{Encrypt}_{ex}(\text{par}, M[i], \text{PK}_{\text{id}})$ and $M[i]$ is the i-th bit of M.

$\text{Decrypt}^l_{ex}(c)$: decrypt component $c[i]$ for each $i \in [l]$ and every message bit $M[i]$, then return $M = M[1] \cdot M[l]$.

The security is shown in Appendix B.

B. Security

In the security definition, there are two types of adversaries: type 1 adversary $\mathcal{A}_1$ and type 2 adversary $\mathcal{A}_2$. Type 1 adversary is a malicious user, who can replace the user's public key but cannot know the master key. Type 2 adversary is a malicious KGC, who can know the master key but cannot replace the user's public key.

In Figure 2 (resp. Figure 3), IND-CPA1 game is for Type 1 adversary $\mathcal{A}_1$ in CLE (resp. IND-CPA2 is for Type 2 adversary $\mathcal{A}_2$). We have $\text{Adv}^{\text{IND-CPA-1}}_{II}(\mathcal{A}_1) = 2 \cdot \Pr[\text{INDCPA } 1^{\mathcal{A}_1}_{II} \Rightarrow \text{true}] - 1$ (resp. $\text{Adv}^{\text{IND-CPA-2}}_{II}(\mathcal{A}_2) = 2 \cdot \Pr[\text{INDCPA } 2^{\mathcal{A}_2}_{II} \Rightarrow \text{true}] - 1$). We say that II is IND-CPA-1

(resp. IND-CPA-2) secure if $\mathrm{Adv}_{\Pi}^{\mathrm{IND-CPA-1}}(\mathcal{A}_1)$ (resp. $\mathrm{Adv}_{\Pi}^{\mathrm{IND-CPA-2}}(\mathcal{A}_2)$) is negligible for all PPT $\mathcal{A}_1$ (resp. $\mathcal{A}_2$).

B.1. Detailed Legend for Figure 2. This figure describes indistinguishable chosen-message attack1 experiment for certificateless encryption scheme, where an adversary and a challenger participate in the experiment and interact with each other. Specifically, in this experiment, the challenger first invokes the algorithm $(\mathrm{par}, \mathrm{msk}) \longleftarrow \mathrm{Setup}_{ex}(1^k)$ to generate (par, msk), where par is taken as the common input and msk is used to generate private key and partial private key. The partial private key oracle proc.ParPrivKeyGen(id) invokes the partial key generation algorithm $d_{\mathrm{id}} \longleftarrow \mathrm{ParPrivKeyGen}_{ex}(\mathrm{par}, \mathrm{id}, \mathrm{msk})$ to return a partial private key d_{id}. The secret value oracle proc.SecValGen(id) invokes the secret value generation algorithm $x_{\mathrm{id}} \longleftarrow \mathrm{SecValGen}_{ex}(\mathrm{par}, \mathrm{id})$ to return a secret value x_{id}. The private key oracle proc.PrivKeyGen(id) invokes the private key generation algorithm $\mathrm{PrivKeyGen}_{ex}(\mathrm{par}, d_{\mathrm{id}}, x_{\mathrm{id}})$ to return a private key. The oracle proc.PubKeyGen(id) invokes the public key generation algorithm $\mathrm{PubKeyGen}_{ex}(\mathrm{par}, x_{\mathrm{id}})$ which takes as input a public parameter and a secret value and returns a public key $\mathrm{PK}_{\mathrm{id}}$ for user id. The replace public key oracle proc.RePubKey $(\mathrm{PK}_{\mathrm{id}}', \mathrm{PK}_{\mathrm{id}}, \mathrm{id})$ takes as input a fresh public key $\mathrm{PK}_{\mathrm{id}}'$, the original public key $\mathrm{PK}_{\mathrm{id}}$, and an identity is and finally returns the replace public key $\mathrm{PK}_{\mathrm{id}}'$. The challenge oracle proc.LR$(m_0, m_1, \mathrm{PK}_{\mathrm{id}}, \mathrm{id})$ takes as input two messages (m_0, m_1), $\mathrm{PK}_{\mathrm{id}}$, and id and returns a challenge ciphertext c which encrypts challenge message m_0 or m_1 randomly. Finally, the experiment gives an output $b = b'$, which denotes whether the adversary wins or not in the experiment.

B.2. Detailed Legend for Figure 3. This figure describes indistinguishable chosen-message attack2 experiment for certificateless encryption scheme, where an adversary and a challenger participate in the experiment and interact with each other. Specifically, in this experiment, the challenger first invokes the algorithm $(\mathrm{par}, \mathrm{msk}) \longleftarrow \mathrm{Setup}_{ex}(1^k)$ to generate (par, msk), where the public parameter par is taken as a common input in all the other algorithms and msk is used to generate private key and partial private key. The oracle proc.SecValGen(id) invokes to return a secret value x_{id}. The private key oracle proc.PrivKeyGen(id) invokes the private key generation algorithm $\mathrm{PrivKeyGen}_{ex}(\mathrm{par}, d_{\mathrm{id}}, x_{\mathrm{id}})$ to return a private key. The public key oracle proc.PubKeyGen(id) invokes the public key generation algorithm $\mathrm{PubKeyGen}_{ex}(\mathrm{par}, x_{\mathrm{id}})$ to return a public key $\mathrm{PK}_{\mathrm{id}}$. The challenge oracle proc.LR$(m_0, m_1, \mathrm{PK}_{\mathrm{id}}, \mathrm{id})$ takes as input two messages (m_0, m_1) chosen by the adversary, $\mathrm{PK}_{\mathrm{id}}$, and id and returns a challenge c which encrypts challenge message m_0 or m_1 randomly. Finally, the experiment outputs $b = b'$, which denotes whether the adversary wins or not in the experiment.

Figures 4–6 are presented for the SO-CPA security for the scheme Π^l where we define two types of adversaries. Both $\mathcal{M}$ $(\alpha \in \{0,1\}^*)$ and $\mathcal{R}$ denote a randomized algorithm. $\mathcal{A}_1$ and $\mathcal{A}_2$ denote type 1 and type 2 SOA adversaries, respectively. In particular, both of the two type of adversaries are only allowed to make one time of query to NewMg before making the Corrupt query. The simulator $\mathcal{S}$ in Figure 6 is an SOA-simulator and is only required to make one time of query to the oracles NewMg and Corrupt.

We say that a CLE scheme Π^l is SIM-SO-CPA secure if for every PPT $\mathcal{M}$, $\mathcal{R}$, $\mathcal{A}_1$, and adversary $\mathcal{A}_2$, there exists a PPT simulator $\mathcal{S}$ such that $\mathrm{Adv}_{\Pi^l,n,\mathcal{S},\mathcal{M},\mathcal{R}}^{\mathrm{SO-CPA-1}}(\mathcal{A}_1) = \Pr[\mathrm{Game}_{\Pi^l,n,\mathcal{A}_1,\mathcal{M},\mathcal{R}}^{\mathrm{SO-CPA-REAL\,1}} \Rightarrow 1] - \Pr[\mathrm{Game}_{\Pi^l,n,\mathcal{S},\mathcal{M},\mathcal{R}}^{\mathrm{SO-CPA-IDEAL}} \Rightarrow 1] \leq \mathrm{negl}(\lambda)$. $\mathrm{Adv}_{\Pi^l,n,\mathcal{S},\mathcal{M},\mathcal{R}}^{\mathrm{SO-CPA-2}}(\mathcal{A}_2) = \Pr[\mathrm{Game}_{\Pi^l,n,\mathcal{A}_2,\mathcal{M},\mathcal{R}}^{\mathrm{SO-CPA-REAL\,2}} \Rightarrow 1] - \Pr[\mathrm{Game}_{\Pi^l,n,\mathcal{S},\mathcal{M},\mathcal{R}}^{\mathrm{SO-CPA-IDEAL}} \Rightarrow 1] \leq \mathrm{negl}(\lambda)$.

B.3. Detailed Legend for Figure 4. This figure describes selective opening chosen-message attack1 real experiment for certificateless encryption scheme, where an adversary and a challenger participate in the experiment and interact with each other. Specifically, in this experiment, the challenger first invokes $(\mathrm{par}, \mathrm{msk}) \longleftarrow \mathrm{Setup}_{ex}(1^k)$ to generate (par, msk), where the value par is taken as a common input and the value msk is used to generate private key and partial private key. The partial private key oracle proc.ParPrivKeyGen(id) invokes the algorithm $d_{\mathrm{id}} \longleftarrow \mathrm{ParPrivKeyGen}_{ex}(\mathrm{par}, \mathrm{id}, \mathrm{msk})$ to produce a partial private key d_{id}. The secret value oracle proc.SecValGen(id) invokes the secret value generation algorithm $x_{\mathrm{id}} \longleftarrow \mathrm{SecValGen}_{ex}(\mathrm{par}, \mathrm{id})$ to generate a secret value x_{id} associated with id. The oracle proc.PrivKeyGen(id) invokes the private key generation algorithm $\mathrm{PrivKeyGen}_{ex} \backslash\backslash (\mathrm{par}, d_{\mathrm{id}}, x_{\mathrm{id}})$ to generate a private key. The public key oracle proc.PubKeyGen(id) invokes the public key generation algorithm $\mathrm{PubKeyGen}_{ex}(\mathrm{par}, x_{\mathrm{id}})$ to generate a public key PK_{id}. The replace oracle proc.RePubKey $(\mathrm{PK}_{\mathrm{id}}', \mathrm{PK}_{\mathrm{id}}, \mathrm{id})$ replaces an old public key with a freshly replaced $\mathrm{PK}_{\mathrm{id}}'$. The challenge oracle proc.NewMg$(\mathbf{id}, \mathbf{PK}, \alpha)$ first takes as input $\mathbf{id}$, PK, and α, and then checks whether $\mathbf{id}$ has been queried to the private key oracle or the replace public key oracle; if not, the challenger samples a message m according to distribution $\mathcal{M}$ determined by α. Then, it samples a randomness $r[i]$ and computes a challenge ciphertext c for message m. The corrupt oracle proc.Corrupt(I) on input a corrupt set 13 I chosen by the adversary and returns the opening $\mathbf{m}[I], \mathbf{r}[I]$. Finally, the experiment outputs $b = b'$, which denotes whether the adversary wins or not in the experiment.

B.4. Detailed Legend for Figure 5. This figure describes selective opening chosen-message attack1 real experiment for certificateless encryption scheme, where an adversary and a challenger participate in the experiment and interact with each other. Specifically, in this experiment, the challenger first invokes the algorithm $(\mathrm{par}, \mathrm{msk}) \longleftarrow \mathrm{Setup}_{ex}(1^k)$ to sample (par, msk). The oracle proc.SecValGen(id) invokes the algorithm $x_{\mathrm{id}} \longleftarrow \mathrm{SecValGen}_{ex}(\mathrm{par}, \mathrm{id})$ to return a secret value x_{id} for user id. The private key oracle proc.PrivKeyGen(id) invokes the private key generation

proc.Initialization(k)
$(par, msk) \leftarrow \text{Setup}_{ex}(1^k)$;
$b \leftarrow_{\$} \{0,1\}$;
return par
proc.ParPrivKeyGen(id)
If $id \in ChID$ then return $\perp$;
$PpID \leftarrow PpID \cup \{id\}$;
return $\text{ParPrivKeyGen}_{ex}(par, id, msk)$
proc.SecValGen(id)
return $\text{SecValGen}_{ex}(par, id)$
proc.ParPrivKeyGen(id)
If $id \in ChID$ then return $\perp\!\!\perp$;
$SkID \leftarrow SkID \cup \{id\}$;
$d_{id} \leftarrow \text{ParPrivKeyGen}_{ex}(par, id, msk)$;
$x_{id} \leftarrow \text{SecValGen}_{ex}(par, id)$;
return $\text{PrivKeyGen}_{ex}(par, d_{id}, x_{id})$
proc.PubKeyGen(id)
$x_{id} \leftarrow \text{SecValGen}_{ex}(par, id)$;
return $\text{PubKeyGen}_{ex}(par, x_{id})$
proc.RePubKey(PK'_{id}, PK_{id}, id)
$PK_{id} \leftarrow PK'_{id}$;
$RpID \leftarrow RpID \cup \{id\}$;
return PK_{id}
proc.LR(m_0, m_1, PK_{id}, id)
If $id \in SkID$
then return $\perp$;
$ChID \leftarrow ChID \cup \{id\}$;
c= $\text{Encrypt}_{ex}(par, m_b, PK_{id})$;
return c
proc.Finalize(b')
return $(b = b')$

Figure 2: Game INDCPA 1.

proc.Initialization(λ)
$(par, msk) \leftarrow \text{Setup}_{ex}(1^\lambda)$;
$b \leftarrow_{\$} \{0,1\}$;
return (par, msk)
proc.SecValGen(id)
If $id \in ChID$ then return $\perp$;
$SeID \leftarrow SeID \cup id$
return $\text{SecValGen}_{ex}(par, id)$
proc.ParvKeyGen(id)
If $id \in ChID$ then return $\perp$;
$SkID \leftarrow SkID \cup id$;
$d_{id} \leftarrow \text{ParPrivKeyGen}_{ex}(par, id, msk)$;
$x_{id} \leftarrow \text{SecValGen}_{ex}(par, id)$;
return $\text{PrivKeyGen}_{ex}(par, d_{id}, x_{id})$
proc.PubKeyGen(id)
$x_{id} \leftarrow \text{SecValGen}_{ex}(par, id)$;
return $\text{PubKeyGen}_{ex}(par, x_{id})$
proc.LR(m_0, m_1, PK_{id}, id)
If $id \in SeID \quad id \in SkID$
then return $\perp$;
$ChID \leftarrow ChID \cup id$;
$c \leftarrow \text{Encrypt}_{ex}(par, m_b, PK_{id})$;
return c
proc.Finalize(b')
return $(b = b')$

Figure 3: Game INDCPA 2.

proc.Initialization(λ)
$(par, msk) \leftarrow \text{Setup}_{ex}(1^\lambda)$;
return par
proc.ParPrivKeyGen(id)
If $id \in ChID$ then return $\perp$;
$PpID \leftarrow PpID \cup \{id\}$
return $\text{ParPrivKeyGen}_{ex}(par, id, msk)$
proc.SecValGen(id)
return $\text{SecValGen}_{ex}(par, id)$
proc.PrivKeyGen(id)
If $id \in ChID$ then return $\perp$;
$SkID \leftarrow SkID \cup \{id\}$;
$d_{id} \leftarrow \text{ParPrivKeyGen}_{ex}(par, id, msk)$;
$x_{id} \leftarrow \text{SecValGen}_{ex}(par, id)$;
return $\text{PrivKeyGen}_{ex}(par, d_{id}, x_{id})$
proc.PubKeyGen(id)
$x_{id} \leftarrow \text{SecValGen}_{ex}(par, id)$;
return $\text{PubKeyGen}_{ex}(par, x_{id})$
proc.RePubKey(PK'_{id}, PK_{id})
$PK_{id} \leftarrow PK'_{id}$;
$RpID \leftarrow RpID \cup \{id\}$;
return PK_{id}
proc.NewMg($\mathbf{id}, \mathbf{PK}, \alpha$)
If $\mathbf{id} \cap SkID \neq \phi$ or $\mathbf{id} \cap RpID = \phi$
then return $\perp$;
$ChID \leftarrow ChID \cup \mathbf{id}$;
$\mathbf{m} \leftarrow_{\$} \mathcal{M}(\alpha)$;
For i in 1 to n
$\mathbf{r}[i] \leftarrow_{\$} Coins(par, \mathbf{m}[i])$
$\mathbf{c}[i] \leftarrow \text{Encrypt}_{ex}^{I}(par, \mathbf{m}[i], \mathbf{PK}_{\mathbf{id}[i]}; \mathbf{r}[i])$;
return $\mathbf{c}$
proc.Corrupt(I)
return $\mathbf{m}[I], \mathbf{r}[I]$
proc.Finalize(out)
return $\mathcal{R}(\mathbf{m}, ChID, I, out)$

Figure 4: $\text{Game}^{\text{SO-CPA-REAL 1}}_{\Pi^I, n, \mathcal{M}, \mathcal{R}}$.

algorithm $\text{PrivKeyGen}_{ex}\backslash\backslash(\text{par}, d_{\text{id}}, x_{\text{id}})$ to generate a private key. The oracle proc.PubKeyGen (id) invokes the public key generation algorithm PubKeyGen_{ex} (par, x_{id}) to return a public key PK_{id}. The challenge oracle proc.NewMg (**i d**, **PK**, α) first checks whether the identity **i d** is legal; if not, the challenger samples a message m according to

proc. Intialization (λ)
$(par, msk) \leftarrow \text{Setup}_{ex}(1)$;
return (par, msk)
proc.SecValGen (id)
If $ID \cap ChID \neq \phi$ then return $\perp$;
$SeID \leftarrow SeID \cup id$
return $\text{SecValGen}_{ex}(par, id)$
proc.PrivKeyGen (id)
If $id \cap ChID \neq \phi$ then return $\perp$;
$SkID \leftarrow SkID \cup id$;
$d_{id} \leftarrow \text{ParPrivKeyGen}_{ex}(par, id, msk)$;
$x_{id} \leftarrow \text{SecValGen}_{ex}(par, id)$;
return $\text{PrivKeyGen}_{ex}(par, d_{id}, x_{id})$
proc.PubKeyGenn(par, id)
$x_{id} \leftarrow \text{SecValGen}_{ex}(par, id)$;
return $\text{PubKeyGen}_{ex}(par, x_{id})$
proc.NewMg($\mathbf{id}, \mathbf{PK}, \alpha$)
If $\mathbf{id} \cap SeID \neq \phi$ $\mathbf{id} \cap SkID \neq \phi$
then return $\perp$;
$ChID \leftarrow ChID \cup \mathbf{id}$;
$\mathbf{m} \leftarrow_{\$} \mathcal{M}(\alpha)$;
For i in 1 to n
$\mathbf{r}[i] \leftarrow_{\$} Coins(par, \mathbf{m}[i])$;
$\mathbf{c}[i] \leftarrow \text{Encrypt}_{ex}^{l}(par, \mathbf{m}[i], \mathbf{PK}_{\mathbf{id}[i]}; \mathbf{r}[i])$;
return $\mathbf{c}$
proc.Corrupt(I)
return $\mathbf{m}[I], \mathbf{r}[I]$
proc.Finalize (out)
return $\mathcal{R}(\mathbf{m}, ChID, I, out)$

Figure 5: $\text{Game}_{II^l,n,\mathcal{M},\mathcal{R}}^{\text{SO-CPA-REAL 2}}$.

proc.Initialization
return $\perp$
proc.NewMg(id,**PK**,α)
ChID $\leftarrow$ ChID $\cup$ id;
$\mathbf{m} \leftarrow_{\$} \mathcal{M}(\alpha)$;
return $\perp$
proc.Corrupt(I)
return $\mathbf{m}[I]$
proc.Finalize(out)
return $\mathcal{R}(\mathbf{m}, ChID, I, out)$

Figure 6: $\text{Game}_{II^l,n,\mathcal{M},\mathcal{R}}^{\text{SO-CPA-IDEAL}}$.

distribution $\mathcal{M}$ determined by α. Then, it samples a randomness $r[i]$ and invokes encryption algorithm to generate a challenge ciphertext c for message m. The corrupt oracle proc.Corrupt(I) takes as input a corrupt set I (which is chosen by the adversary), and returns the opening messages $\mathbf{m}[I]$ and randomnesses $\mathbf{r}[I]$. Finally, the experiment outputs $b = b'$, which denotes whether the adversary wins or not in the experiment.

B.5. Detailed Legend for Figure 6. This figure describes selective opening chosen-message attack ideal experiment for certificateless encryption scheme, where an adversary and a simulator participate in the experiment and interact with each other. Specifically, in this experiment, during the initialization phase, the challenger returns nothing for an adversary, while the challenge oracle proc.NewMg $(\mathbf{id}, \mathbf{PK}, \alpha)$ only samples messages m according to distribution $\mathcal{M}$ determined by α but returns nothing to the adversary. In the corruption phase, the challenger opens the partial messages $\mathbf{m}[I]$ according to the set I chosen by the adversary. Finally, the experiment returns an output of a relation with respect to an input tuple $(\mathbf{m}, \text{ChID}, I, \text{out})$.

C. Conversion from 1SPO to SIM-SO-CPA

Here, we use a theorem (i.e., Theorem 2) to demonstrate how to reduce the SIM-SO-CPA security to 1SPO security.

Theorem 2 (see [14]). *Let II be a 1-bit 1SPO-CLE scheme with a δ one-sided opener P Open To Zero algorithm [3] and II^l the l-bit 1SPO-CLE scheme from II. $\mathcal{A}_1$ and $\mathcal{A}_2$ are type 1 adversary and type 2 adversary against SO-CPA security of II^l, respectively. Let $\mathcal{R}$ be a PPT relation and $\mathcal{M}$ be a PPT message sampler. Then, there exist $\mathcal{S}$ and two $\mathcal{B}_1$ and $\mathcal{B}_2$ such that*

$$\text{Adv}_{II^l,n,\mathcal{M},\mathcal{R},\mathcal{S}}^{\text{SO-CPA-1}}(\mathcal{A}_1) \leq \text{nl} \cdot \text{Adv}_{II}^{\text{IND-CPA-1}}(\mathcal{B}_1) + \text{nl}\,\delta,$$
$$\text{Adv}_{II^l,n,\mathcal{M},\mathcal{R},\mathcal{S}}^{\text{SO-CPA-2}}(\mathcal{A}_2) \leq \text{nl} \cdot \text{Adv}_{II}^{\text{IND-CPA-2}}(\mathcal{B}_2) + \text{nl}\,\delta. \tag{C.1}$$

Proof. This proof process is exactly the same as that of Theorem 1 in [14], so we will not repeat it here in order to save space. □

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The first author was supported by the National Key Research and Development Program of China (grant no. 2017 YFB0802000), the National Natural Science Foundation of China (grant no. NSFC61702007), and other foundations (grant nos. 2019M661360 (KLH2301024), gxbjZD27, KJ2018A0533, XWWD201801, and ahnis20178002). The third author was supported by the National Key Research and Development Program of China (grant no.

2017YFB0802000) and the National Natural Science Foundation of China (grant no. U1705264).

References

[1] J. Baek, R. Safavi-Naini, and W. Susilo, "Certificateless public key encryption without pairing," in *Lecture Notes in Computer Science*, pp. 134–148, Springer, Berlin, Germany, 2005.

[2] M. Bellare, D. Hofheinz, and S. Yilek, "Possibility and impossibility results for encryption and commitment secure under selective opening," in *Advances in Cryptology-EUROCRYPT 2009*, pp. 1–35, Springer, Berlin, Germany, 2009.

[3] M. Bellare, B. Waters, and S. Yilek, "Identity-based encryption secure against selective opening attack," in *Theory of Cryptography*, pp. 235–252, Springer, Berlin, Germany, 2011.

[4] C. Dwork, M. Naor, O. Reingold, and L. Stockmeyer, "Magic functions," in *Foundations of Computer Science (FOCS 1999)*, pp. 523–534, Springer, Berlin, Germany, 1999.

[5] S. Fehr, D. Hofheinz, E. Kiltz, and H. Wee, "Encryption schemes secure against chosen-ciphertext selective opening attacks," in *Advances in Cryptology-EUROCRYPT 2010*, pp. 381–402, Springer, Berlin, Germany, 2010.

[6] Z. Huang, S. Liu, and B. Qin, "Sender-equivocable encryption schemes secure against chosen-ciphertext attacks revisited," in *Public-Key Cryptography-PKC 2013*, pp. 369–385, Springer, Berlin, Germany, 2013.

[7] D. Jia, Y. Liu, and B. Li, "IBE with tight security against selective opening and chosen-ciphertext attacks," *Designs, Codes and Cryptography*, vol. 88, no. 7, pp. 1371–1400, 2020.

[8] J. T. Ning and G. S. Poh, "Update recovery attacks on encrypted database within two updates using range queries leakage," *IEEE Transactions on Dependable and Secure Computing*, 2020.

[9] J. Lai, R. H. Deng, S. Liu, and W. Kou, "Rsa-based certificateless public key encryption," in *Information Security Practice and Experience*, pp. 24–34, Springer, Berlin, Germany, 2009.

[10] J. Z. Lai, D. Robert, S. Liu, J. Weng, and Y. Zhao, "Identity-based encryption secure against selective opening chosen-ciphertext attack," in *EUROCRYPT*, pp. 11–15, Springer, Berlin, Germany, 2014.

[11] J. Ning, Z. Cao, X. Dong, K. Liang, L. Wei, and K. K. R. Choo, "Cryptcloud+: secure and expressive data access control for cloud storage," *IEEE Transactions on Services Computing*, vol. 14, 2018.

[12] J. Ning, X. Huang, W. Susilo, K. Liang, X. Liu, and Y. Zhang, "Dual access control for cloud-based data storage and sharing," *IEEE Transactions on Dependable and Secure Computing*, 2020.

[13] C. Sur, C. D. Jung, Y. Park, and K. H. Rhee, "Chosen-ciphertext secure certificateless proxy re-encryption," in *Communications and Multimedia Security*, pp. 214–232, Springer, Berlin, Germany, 2010.

[14] H. Wang, K. Chen, B. Qin, and L. Wang, "Certificateless encryption secure against selective opening attack," *Security and Communication Networks*, vol. 9, no. 18, pp. 5600–5614, 2016.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 6632518, 12 pages
https://doi.org/10.1155/2021/6632518

Research Article

Publicly Verifiable Outsourcing Computation for QR Decomposition Based on Blockchain

Huimin Wang,[1] **Dong Zheng**,[1,2] **and Qinglan Zhao**[1]

[1]*National Engineering Laboratory for Wireless Security, Xi'an University of Posts and Telecommunications, Xi'an 710121, China*
[2]*Westone Cryptologic Research Center, Beijing 100070, China*

Correspondence should be addressed to Qinglan Zhao; zhaoqinglan@foxmail.com

Received 9 December 2020; Revised 7 February 2021; Accepted 8 March 2021; Published 24 March 2021

Academic Editor: Jianting Ning

In the Big Data Era, outsourcing computation has become increasingly significant as it supplies computation resources for clients with limited resources. However, there are still many security challenges such as payment fairness, privacy protection, and verification. In this paper, we propose a secure publicly verifiable outsourcing computation scheme for the large-scale matrix QR decomposition. In the proposed scheme, client can pay for outsourcing services through blockchain-based payment system which achieves the payment fairness. Moreover, to protect privacy, both permutation matrix and block diagonal matrix are applied in encryption process. Meanwhile, to achieve the public verification, the computational complexity is reduced by using the matrix digest technology. It is worth mentioning that our scheme is provable and secure under the co-CDH assumption.

1. Introduction

Cloud computing, a new computing technology and service concept, has appeared in the public's vision and serves customers in a pay-per-use manner [1–3]. It has promoted the development of the emerging fields such as smart medical systems in recent years.

Outsourcing computation, as one of the basic applications of cloud computing, can reduce significantly the clients' computational burden [4]. There are two parts, including payment and computation, in outsourcing scheme. For payment part, it often requires online payment and relies on trusted third party such as bank. To realize secure and fair payment of outsourcing services without relying on any third party, fair payment framework based on blockchain has been used for outsourcing services in cloud computing [5]. For computation part, service requester submits the data-to-service provider, which might get service requester's privacy from the data. Therefore, there exist many security challenges during the outsourcing process.

About the protection of client privacy, computing tasks authorized to cloud server involve some important sensitive information frequently, such as core technology of a company and patient health records. So, it is important for users to conceal their data information before uploaded to the cloud server. The previous works have attempted to protect the confidentiality of the data. For example, full homomorphic encryption [6], a cryptographic technique, can allow service provider to perform valid and meaningful operations on ciphertext. However, the existing schemes based on FHE suffer from high computation complexity.

Moreover, the result verification is vital as well. Since the process of cloud computing is not transparent to the public who will upload their data, the public should detect in time whether there are any errors in outsourcing result. There may be many reasons to produce an invalid and wrong result, such as hardware malfunction, software bugs, or malicious hackers. Furthermore, a semihonest cloud server [7] might work dishonestly or even cut down calculation steps due to huge benefits.

Considering financial expenses, an outsourcing computating scheme should be highly efficient. That is, the user's computation in the outsourced process is far less than the computation of their task directly. Otherwise, the outsourcing will get meaningless for the client.

Matrix computation has many applications in scientific and engineering fields. The outsourcing matrix computation also involves the above security challenges. We research on secure outsourcing matrix QR decomposition computation and propose a publicly verifiable scheme based on blockchain. The system consists of two parts: blockchain-based payment and publicly verifiable computation. In this paper, we focus on designing publicly verifiable computation scheme and we refer the reader to reference [5] to know more about blockchain-based payment.

1.1. Contributions. The contribution of this paper can be described from the following three points:

(i) We multiply a sparse block diagonal matrix with the original matrix to protect the client's privacy. The computational complexity is $O(n^2)$ in the encryption process.

(ii) The scheme provides public verification. To reduce the workload of the verifier, we use matrix digest technique which transforms any matrix into a specific vector with chosen parameter in the verification of QR decomposition.

(iii) We show the soundness of the scheme through detailed theoretical analysis, including correctness, security, and efficiency. It is proved that the scheme achieves secure under *co*-CDH difficulty assumption.

1.2. Related Work. Looking back on the development of outsourcing computation in the past decades, many schemes have been designed for different scientific computations. Atallah et al. [8] proposed the concept of the scientific computing outsourcing firstly. To protect privacy of clients, researchers have devoted to design the secure outsourcing schemes [9–14]. Salinas et al. [9] mentioned a privacy-preserving transformation method by adding random matrix to original matrix. For the verifiability of the outsourcing results, Golle and Mironov [15] firstly realized this goal in their scheme. Then, a verifiable scheme about any random function was designed by Gentry [6] which provided the formal concept of verifiable computing. Banabbas et al. [16] put forward to a verifiable scheme for high-degree polynomial functions.

Nevertheless, in many applications, verification needs to be public. In other words, any customer can verify it. Recently, some experts turned their attention to public verifiable computation. Fioreand Gennaro [17] allowed service requester to verify the result with a noninteractive evidence. Meanwhile, Parno et al. [18] gave the concept of the correctness and security which had established a connection between public verification computation and attribute-based encryption (ABE). In addition, Fiore and Gennaro [17] also designed the matrix multiplication outsourcing scheme according to Yao's Garbled Circuits [19]. Different from traditional scheme [11], the scheme [20] has achieved that all clients can share a common matrix M to perform matrix multiplication, which did not protect the security of the original matrix.

Jia et al. [12] took the privacy protection into account, where the matrix can be arbitrary. Li et al. [21] improved its efficiency compared to previous work and achieved the public verification. Zhang et al. [13] reduced the computing overhead in the verification process hugely. The scheme in [22] not only achieved the public verification but also protected privacy information of original data, where matrix digest was utilized to reduce the overhead of key generation and cloud computing.

The above results [12, 17, 20–22] are about publicly verifiable solutions for matrix multiplication. However, there is no publicly verifiable solution about matrix decomposition. Matrix decomposition, as one of the basic matrix operations, has many application scenarios [23–26]. Luo et al. [27] had designed a secure algorithm for QR decompositions without the public verification achieved. We propose a scheme which achieves the promising public verification under the amortized model for QR decomposition of large-scale matrices. To protect privacy, sparse matrices which cut down the computational complexity from $O(n^3)$ to $O(n^2)$ are applied during encryption.

1.3. Organization. In Section 2, it introduces related definitions of verifiable computing and significant mathematical knowledge. Section 3 details the proposed scheme for the publicly verifiable computation of the QR decomposition. The correctness, security, and efficiency analysis are shown in Section 4. At last, it ends up with our conclusion in Section 5.

2. Preliminaries

In this part, we introduce some related definitions, mathematical knowledge, and important techniques.

2.1. Publicly Verifiable Computation. As mentioned by Gennaro [28], a public verifiable computation scheme $\mathscr{VC}$ not only allows a client to outsource his computating task but also states that the outsourcing result is correct and verifiable. The formal definitions of these properties for public verifiable computation are presented in [22, 28]. For the sake of integrity, we give some related definitions before introducing our scheme.

Definition 1. A outsourcing scheme $\mathscr{VC}$ consists of the following five subalgorithms:

(i) $\mathbf{KeyGen}(1^\lambda, \mathscr{F}) \longrightarrow$ (SK, PK):

Given the random selected parameter λ, a public key PK is produced to protect the function $\mathscr{F}$. Simultaneously, a private key SK saved by the service requester secretly is generated by this algorithm.

(ii) $\mathbf{ProbGen}_{\mathrm{SK}}(x) \longrightarrow (\tau_x, \sigma_x)$:

The client performs the encryption together with the SK and gets a decoding value τ_x stored

confidentially, where the input x of function is encoded into a encrypted result σ_x.

(iii) $\mathbf{Compute}_{\mathrm{PK}}(\sigma_x) \longrightarrow (\sigma_y)$:
According to the PK and the encrypted σ_x, the outsourcing server provider produces a blinded output σ_y.

(iv) $\mathbf{Verify}_{\mathrm{SK}}(\tau_x, \sigma_y) \longrightarrow (y \cup \perp)$:
Based on σ_y and τ_x, if σ_y of function $\mathcal{F}$ is correct, it outputs y. Otherwise, it outputs the symbol $\perp$.

(v) $\mathbf{Solve}_{\mathrm{SK}}(\tau_x, \sigma_y) \longrightarrow (y)$:
The algorithm decodes σ_y to generate the final result $y = \mathcal{F}(x)$ with SK and τ_x.

Next, we focus on these properties in publicly verifiable computation scheme $\mathcal{VC}$, including correctness, security, privacy, and efficiency.

Definition 2 (correctness). For a function $\mathcal{F}$, we say the verifiable outsourcing scheme $\mathcal{VC}$ is correct if the key generation algorithm generates keys $(\mathrm{PK}, \mathrm{SK}) \leftarrow \mathbf{KeyGen}(\mathcal{F}, 1^{\lambda})$ and satisfies the following condition: $\forall x \in \mathrm{Domain}(\mathcal{F})$, $y = \mathcal{F}(x) \leftarrow \mathbf{Verify}_{\mathrm{SK}}(\tau_x, \sigma_y)$ if $(\sigma_x, \tau_x) \leftarrow \mathbf{ProbGen}_{\mathrm{SK}}(x)$ and $\sigma_y \leftarrow \mathbf{Compute}_{PK}$.

The formalized definition of security of a verifiable computation outsourcing scheme $\mathcal{VC}$ is introduced, where a malicious server cannot persuade the verifier to output a invalid result $\hat{y}$ according to the function $\mathcal{F}$ and input x, e.g., $\mathcal{F}(x) \neq \hat{y}$.

Now, we abstract this objective fact with an experiment which is expressed as below.

Experience $\mathbf{Exp}_{\mathcal{A}}^{\mathrm{Verify}}[\mathcal{VC}, \mathcal{F}, \lambda]$;
For $i = 1, \ldots, l = \mathrm{poly}(\lambda)$;
$(\mathrm{PK}, \mathrm{SK})^{R} \leftarrow \mathbf{KeyGen}(\mathcal{F}, \lambda)$;
$x_i \leftarrow \mathcal{A}(\mathrm{PK}, x_1, \delta_1, \ldots, x_{(i-1)}, \delta_{(i-1)})$;
$(\delta_i, \tau_i) \leftarrow \mathbf{ProbGen}_{\mathrm{SK}}(x_i)$
$(i, \hat{\delta}_y) \leftarrow \mathcal{A}(\mathrm{PK}, x_1, \delta_1, \ldots, x_l, \delta_l)$
$\hat{y} \leftarrow \mathbf{Verify}_{\mathrm{SK}}(\tau_i, \hat{\delta}_y)$;
If $\hat{y} = \perp$ and $\hat{y} \neq \mathcal{F}(x_i)$, output 1, else 0.

Here, $\mathrm{poly}(\cdot)$ is defined as a polynomial.

Given an oracle access, the adversary can produce the encryption of multiple problems. Considering a known input, the adversary can persuade the verifier to work smoothly, where any error is unable to be detected in the output.

Definition 3 (security). For a verifiable computation outsourcing scheme $\mathcal{VC}$, the capability of an adversary $\mathcal{A}$ in the above experiment is defined as follows:

$$\mathrm{Adv}_{\mathcal{A}}^{\mathrm{Verify}}(\mathcal{VC}, \mathcal{F}, \lambda) = \Pr\left[\mathrm{Exp}_{\mathcal{A}}^{\mathrm{Verify}}(\mathcal{VC}, \mathcal{F}, \lambda) = 1\right]. \tag{1}$$

For a function $\mathcal{F}$, the scheme $\mathcal{VC}$ is secure if, for any adversary $\mathcal{A}$ running in probabilistic polynomial time (PPT),

$$\mathrm{Adv}_{\mathcal{A}}^{\mathrm{Verify}}(\mathcal{VC}, \mathcal{F}, \lambda) \leq \mathrm{negli}(\lambda), \tag{2}$$

where $\mathrm{negli}(\cdot)$ is a negligible function of its input.

If the outputs of the ProbGen algorithm over two different inputs are indistinguishable, we think a $\mathcal{VC}$ scheme is private. To define privacy of a verifiable computation scheme, we need an experiment. Given the public key PK for the scheme, the adversary $\mathcal{A}$ treats x_0 and x_1 as two inputs randomly. Then, he is given the encoded version of one of x_0 and x_1 and must guess which one was encoded. In the process, the oracle $\mathrm{PubProbGen}_{\mathrm{SK}}(x)$ calls $\mathrm{ProbGen}_{\mathrm{SK}}(x)$ to obtain (δ_x, τ_x) and returns only the public part δ_x. Now, the experiment is described below.

Experience $\mathbf{Exp}_{\mathcal{A}}^{\mathrm{Priv}}[\mathcal{VC}, \mathcal{F}, \lambda]$;
$(\mathrm{PK}, \mathrm{SK}) \overset{R}{\leftarrow} \mathbf{KeyGen}(\mathcal{F}, \lambda)$;
$(x_0, x_1) \leftarrow \mathcal{A}^{\mathrm{PubProbGen}_{\mathrm{SK}}(\times)}(\mathrm{PK})$
$(\delta_0, \tau_0) \leftarrow \mathbf{ProbGen}_{\mathrm{SK}}(x_0)$
$(\delta_1, \tau_1) \leftarrow \mathbf{ProbGen}_{\mathrm{SK}}(x_1)$
$y \overset{R}{\leftarrow} \{0, 1\}$;
$\hat{y} \leftarrow \mathcal{A}^{\mathrm{PubProbGen}_{\mathrm{SK}}(\times)}(\mathrm{PK}, x_0, x_1, \delta_y)$
If $y = \hat{y}$, output 1, else 0.

Definition 4 (privacy). According to the above experiment, the ability of an adversary $\mathcal{A}$ is defined as

$$\mathrm{Adv}_{\mathcal{A}}^{\mathrm{Priv}}(\mathcal{VC}, \mathcal{F}, \lambda) = \left|\Pr\left[\mathrm{Exp}_{\mathcal{A}}^{\mathrm{Priv}}(\mathcal{VC}, \mathcal{F}, \lambda) = 1\right] - \frac{1}{2}\right|. \tag{3}$$

A verifiable computation scheme is private if, for a function $\mathcal{F}$ any adversary $\mathcal{A}$ running in PPT,

$$\mathrm{Adv}_{\mathcal{A}}^{\mathrm{Priv}}(\mathcal{VC}, \mathcal{F}, \lambda) \leq \mathrm{negli}(\lambda), \tag{4}$$

where $\mathrm{negli}(\cdot)$ is a negligible function of its input.

Definition 5. (efficiency). A verifiable scheme $\mathcal{VC}$ must be highly efficiency for the client. That is, the time for encryption and verification in the scheme should be shorter than the time to accomplish the computing task directly by itself.

2.2. Bilinear Pairings. The knowledge about bilinear pairings, in a verifiable computating scheme $\mathcal{VC}$, will be introduced as follows.

Let G_1, G_2, and G_T be three multiplicative cyclic groups with the same large prime order p and g_1 and g_2 be generators of G_1 and G_2, respectively. A bilinear pairing is a map $e: G_1 \times G_2 \longrightarrow G_T$, which has the following three characteristics:

(i) Bilinearity: for any $\alpha, \beta \in Z_p$ and $g^{\alpha} \in G_1, h^{\beta} \in G_2$, the equation $e(g^{\alpha}, h^{\beta}) = e(g, h)^{\alpha\beta}$ holds

(ii) Computability: there exist a valid algorithm for solving $e(g,h)$ for any $(g,h) \in G_1 \times G_2$

(iii) Nondegeneracy: for any $g \in G_1$, if, for all $h \in G_2$, equation $e(g,h) = 1$ is true, $g = 1$, where g and h can be interchanged

According to the above, the related definitions about computational assumptions can be described as follows.

Definition 6 (co-CDH problem). Given $g, g^{\alpha} \in G_1, h, h^{\beta} \in G_2$, and $\alpha, \beta \in_R F_p^*$, compute $g^{\alpha\beta}$.

Definition 7 (co-CDH assumption). Given $g, g^{\alpha} \in G_1$ and $h, h^{\beta} \in G_2$, for randomly selecting $\alpha, \beta \in_R F_p^*$, if the probability to compute $g^{\alpha\beta}$ is negligible in any PPT, the co-computational Diffie–Hellman assumption holds in G_1.

2.3. Matrix Digest Technique. As an one-way irreversible mapping process, matrix digest [22, 29] refers to transform an any matrix into a specific vector with a chosen parameter, which makes computational complexity reduce from $O(n^3)$ to $O(n^2)$. In fact, a matrix consists of some column vectors, in which a vector is also a special matrix. For example, a square matrix $\widehat{A}$ can be denoted as $\widehat{A} = (\overrightarrow{a}_1, \ldots, \overrightarrow{a}_i, \ldots, \overrightarrow{a}_n)$, where $\overrightarrow{a}_i \in Z^{n\times 1}$ is a column vector. By this novel technique, we can transform the matrix $\widehat{A}$ into the vector $\overrightarrow{b}$ by a row vector $\overrightarrow{s} \in Z^*$, where the vector $\overrightarrow{b} = \overrightarrow{s}\widehat{A}$ is a matrix digest of $\widehat{A}$.

There are three properties of matrix digest:

(i) Deterministic: the matrix digest of a matrix will be determined uniquely by the known parameter vector, *i.*e., if $\overrightarrow{s}$ is chosen as the parameter, $\overrightarrow{b}$ must be unique for $\widehat{A}$.

(ii) Computable: the result of matrix digest essentially is a vector and retains the computing ability of the initial matrix.

(iii) Irreversible: given a matrix digest, it is difficult for anyone to detect initial matrix and selected parameter. Furthermore, if the matrix digest and the parameter are known at the same time, the initial matrix cannot be obtained as well.

3. The Proposed Scheme

3.1. Treat Model. The semihonest model introduced in [7] is an honest but curious one with an untrustworthy cloud server as the main adversary. It was also mentioned in [30], where participants in the outsourcing are required to honestly execute the designed scheme. With returning a correct result, semihonest cloud will try to recover sensitive information of the data. Our scheme is based on a semihonest cloud server and introduces an independent data center which is trustworthy.

3.2. System Model. Considering the public verification, we give a system model about outsourcing computation with the following five entities introduced, as shown in Figure 1.

(i) Data center ($\mathscr{DC}$): some keys are produced by $\mathscr{DC}$. After initializing parameters, it generates the private keys and some public keys. Next, it takes advantage of the private key to generate the evaluation key for $\mathscr{CS}$. Finally, it sends the private key to $\mathscr{C}$ and $\mathscr{V}$ over the secure channel.

(ii) Client ($\mathscr{C}$): first of all, $\mathscr{C}$ should deposit enough money into $\mathscr{B}_{\mathscr{P}}$ for the cost of outsourcing services. Meanwhile, a request is sent to $\mathscr{CS}$ about solving a QR decomposition of the large-scale matrix. To protect privacy, $\mathscr{C}$ needs to encode the original matrix before the private matrix is uploaded to $\mathscr{CS}$. Then, a verification key should be generated for $\mathscr{V}$.

(iii) Cloud server ($\mathscr{CS}$): like $\mathscr{C}$, $\mathscr{CS}$ also needs to provide deposits to $\mathscr{B}_{\mathscr{P}}$. As service provider, $\mathscr{CS}$ needs to perform QR decomposition of the encryption matrix and earn fees from $\mathscr{C}$. Moreover, a proof sent to $\mathscr{V}$ together with computating results is generated by using the evaluation key. Finally, the result matrices will be transmitted to $\mathscr{C}$. If $\mathscr{C}$ has no objection to the outsourcing result within a specified time, $\mathscr{CS}$ will provide a proof $\mathrm{OS}_{\mathrm{end}}$ to $\mathscr{B}_{\mathscr{P}}$ and get the corresponding fees. Otherwise, $\mathscr{CS}$ provides compensation to $\mathscr{C}$.

(iv) Verifier ($\mathscr{V}$): any verifier can be regarded as $\mathscr{V}$. Utilizing the verification key and the proof, $\mathscr{V}$ will examine the correctness of the outsourcing results.

(v) Blockchain payment ($\mathscr{B}_{\mathscr{P}}$): we take advantage of the payment system based on blockchain $\mathscr{B}_{\mathscr{P}}$. After receiving deposit from $\mathscr{C}$ and $\mathscr{CS}$, $\mathscr{B}_{\mathscr{P}}$ provides a proof OS_{start} for $\mathscr{CS}$ to confirm to start the outsourcing service.

3.3. Process Description. The system model consists of two parts: blockchain-based payment system and publicly verifiable outsourcing computing system. In blockchain-based payment system, $\mathscr{C}$ needs to provide the corresponding deposits in $\mathscr{B}_{\mathscr{P}}$ as the cost of service before requesting $\mathscr{CS}$ to perform QR decomposition of large-scale matrix $\widehat{A}$. Meanwhile, $\mathscr{CS}$ also deposits the compensation in $\mathscr{B}_{\mathscr{P}}$ as a guarantee for honest computing. If outsourcing result is correct, $\mathscr{CS}$ can obtain the corresponding service fees from $\mathscr{B}_{\mathscr{P}}$. Otherwise, $\mathscr{C}$ informs $\mathscr{B}_{\mathscr{P}}$ to terminate the payment process, and $\mathscr{CS}$ will accept punishment and provide compensation to $\mathscr{C}$. In this paper, we focus on designing publicly verifiable outsourcing computation for QR decomposition scheme called $\mathscr{PVCMD}$-$\mathscr{QR}$.

$\mathscr{PVCMD}$-$\mathscr{QR}$ can be divided into five phases including initialization phase, encryption phase, computation phase, verification phase, and decryption phase. To better understand this process, a flowchart is shown in Figure 2. Now, the specific process of the scheme is described below.

In the initialization phase, $\mathscr{DC}$ runs the KeyGen algorithm. Here, it randomly generates three n-dimensional key vectors $\overrightarrow{s}$, $\overrightarrow{l}$, and $\overrightarrow{k}$ as the private key SK to produce the public key $\mathrm{PK} = (\mathrm{PK}_1, \mathrm{PK}_2, \mathrm{PK}_3)$ and the evaluation key EK. $\overrightarrow{s}$ and $\overrightarrow{l}$ are delivered to $\mathscr{C}$ and $\mathscr{V}$ through the secure

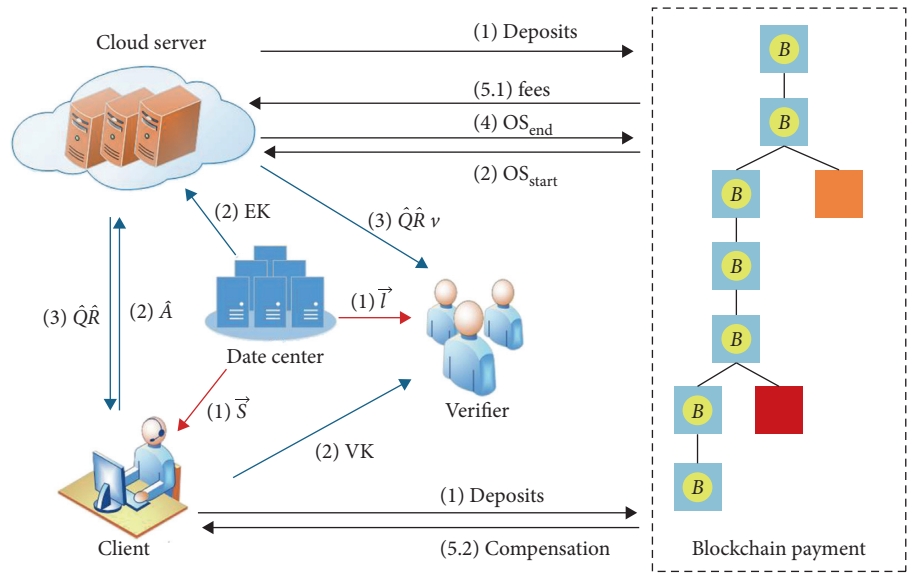


Figure 1: System model.

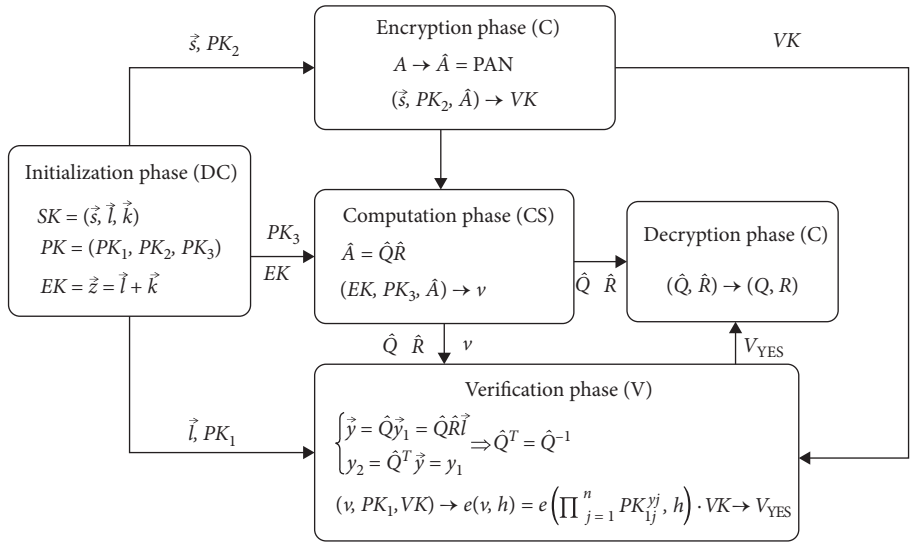


Figure 2: A plan flowchart of the proposed scheme.

channel, respectively. In the encryption phase, ProbGen algorithm is executed. $\mathscr{C}$ encrypts a full rank privacy matrix A into $\hat{A} = \text{PAN}$, $A \in Z_p^{n\times n}$, where block-diagonal upper triangular matrix N is constructed by $\mathscr{C}$. Meanwhile, the verification key VK is generated by PK_2, $\vec{s}$, and $\hat{A}$. Here, we make full use of the technique called matrix digest during the process of generating the VK in our scheme. By multiplying $\vec{s}$ and the encryption matrix $\hat{A}$ to obtain the vector $\vec{b}$, $\mathscr{V}$ uses PK_2 and $\vec{b}$ to create VK. Then, $\hat{A}$ is uploaded to $\mathscr{CS}$, and VK is provided to any $\mathscr{V}$ simultaneously. Then, the compute algorithm is implemented in the computation phase. $\mathscr{CS}$ receives $\hat{A}$ to perform QR decomposition. Using EK from $\mathscr{DC}$, it generates a value v for $\mathscr{V}$. After getting the orthogonal matrix $\hat{Q}$ and the upper triangular matrix $\hat{R}$, $\mathscr{V}$ begins to execute the verify algorithm by using both the key VK and the proof v in the verification phase. If the verification is true, V_{YES} is sent to $\mathscr{C}$ at once and $\mathscr{V}$ informs $\mathscr{C}$ to accept the decomposition results. By the matrix digest, the verifier uses the vector $\vec{l}$ and the decomposition results to produce the vector $\vec{y} = (y_1, \ldots, y_j, \ldots, y_n)$. It is this technology that prevents $\mathscr{V}$ from having to traverse each element of the result matrices. Eventually, in the decryption phase, utilizing the unit permutation matrix P^T and the inverse matrix N^{-1} of the matrix N, $\mathscr{C}$ runs the solve algorithm and decrypts the result matrices to get the orthogonal matrix Q and the upper triangular matrix R of A.

3.4. Specific Algorithms of $\mathscr{PVCMD}$-$\mathscr{QR}$. The $\mathscr{PVCMD}$-$\mathscr{QR}$ scheme consists of five subalgorithms, including KeyGen, ProbGen, Compute, Verify, and Solve.

Algorithm 1 (KeyGen) is executed by $\mathscr{DC}$.

Input:
$\mathcal{F}$, λ;
Output:
para and SK, PK, EK;
(1) Step 1: initialization
(2) compute para = $(p, G_1, G_2, G_T, g, h, \tilde{g}, \tilde{h})$.
(3) Step 2: generating keys
(4) generate SK and EK.
(5) compute PK
(6) Step 3: return (para, SK, PK, EK)

Algorithm 1: KeyGen algorithm.

There exist two cyclic groups G_1 and G_2 with the order p, where g is generator of G_1 and h is generator of G_2. So, a bilinear pairings can be described as $G_1 \times G_2 \longrightarrow G_T$. For any $\delta \in F_p^*$, it calculates $\tilde{h} = h^\delta$ and $\tilde{g} = g^\delta$. Afterwards, it publishes the parameter para = $(p, G_1, G_2, G_T, g, h, \tilde{g}, \tilde{h})$.

(a) It generates three vectors $\overrightarrow{s}$, $\overrightarrow{l}$, and $\overrightarrow{k}$ randomly, $s_i, l_i, k_i \in F_p^* (1 \le i \le n)$. Then, we regard these three vectors as the secret key SK. The algorithm uses the vectors $\overrightarrow{l}$ and $\overrightarrow{k}$ to determine the evaluation key EK = $\overrightarrow{z} = \overrightarrow{l} + \overrightarrow{k}$.

(b) By using SK = $(\overrightarrow{s}, \overrightarrow{l}, \overrightarrow{k})$, it produces the public key PK = $(\mathrm{PK}_1, \mathrm{PK}_2, \mathrm{PK}_3)$, which is defined as follows:

$$\begin{cases} \mathrm{PK}_1 = (\mathrm{PK}_{11}, \mathrm{PK}_{12}, \ldots, \mathrm{PK}_{1n}), \\ \mathrm{PK}_2 = (\mathrm{PK}_{21}, \mathrm{PK}_{22}, \ldots, \mathrm{PK}_{2n}), \\ \mathrm{PK}_3 = (\mathrm{PK}_{31}, \mathrm{PK}_{32}, \ldots, \mathrm{PK}_{3n}), \end{cases} \tag{5}$$

where $\mathrm{PK}_{1i} = g^{s_i}$, $\mathrm{PK}_{2i} = e(g^{k_i}, \tilde{h})$, and $\mathrm{PK}_{3i} = \tilde{g}^{s_i}$ for $i = 1$ to n.

Algorithm 2 (ProbGen)is expressed and is executed by $\mathcal{C}$ to encrypt a privacy matrix A.

(a) $\mathcal{C}$ needs to perform $\widehat{A} = \mathrm{PAN}$, where $N \in Z_p^{n \times n}$. In particular, the matrix P is an n-order unit permutation matrix, and the matrix N is a sparse block diagonal square matrix, whose main diagonal is composed of several matrices N_i and the remaining positions are all 0 elements:

$$N = \begin{pmatrix} N_1 & 0 & \cdots & 0 \\ 0 & N_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & N_t \end{pmatrix}_{n \times n}, \tag{6}$$

where the submatrix N_i $(i = 1, 2, \ldots, t)$ is lower-order upper triangular invertible matrix, where $N_i \in Z_p^*$ is saved secretly.

(b) It uses the encryption matrix $\widehat{A}$ to obtain the verification key VK as below:

(i) The client uses $\overrightarrow{s}$ to produce the auxiliary vector $\overrightarrow{b}$, where $\overrightarrow{b} = \overrightarrow{s} \cdot \widehat{A}$.
(ii) PK_2 and $\overrightarrow{b}$ are used to generate VK, namely,

Input:
$\overrightarrow{s}$, PK_2, and A;
Output:
$\widehat{A}$ and VK
(1) Step 1: transforming matrix A
(2) Produce the matrices P and N.
(3) Encode A into $\widehat{A}$.
(4) Step 2: obtaining VK
(5) Compute $\overrightarrow{b}$.
(6) Obtain VK by equation (7).
(7) Step 3: return ($\widehat{A}$, VK)

Algorithm 2: ProbGen algorithm.

$$\mathrm{VK} = \prod_{i=1}^{n} \mathrm{PK}_{2i}^{b_i}. \tag{7}$$

In Algorithm 3 (Compute), $\mathcal{CS}$ conducts QR decomposition of $\widehat{A}$. The process of algorithm is shown in Algorithm 3.

(a) Breaking $\widehat{A}$ into $\widehat{A} = \widehat{Q}\widehat{R}$, $\mathcal{CS}$ performs operation of QR decomposition.

(b) It should calculate a proof v to prove the correctness of the decomposition results.

(i) An n-dimensional auxiliary vector $\overrightarrow{m}$ is generated, where $m_i = \prod_{j=1}^{n} \mathrm{PK}_{3j}^{\widehat{A}_{ji}}$ $(1 \le i \le n)$.
(ii) It uses the vector $\overrightarrow{m}$ and EK to generate the value $v = \prod_{i=1}^{n} m_i^{z_i}$.

Algorithm 4 (Verify) is conducted after $\mathcal{V}$ receives the information from other participants. The specific process is explained in Algorithm 4.

(a) $\mathcal{V}$ should inspect the orthogonality of $\widehat{Q}$.

(i) $\mathcal{V}$ uses $\overrightarrow{l}$ and $\widehat{R}$ to generate a intermediate vector $\overrightarrow{y}_1$. By multiplying $\widehat{Q}$ by the column vector $\overrightarrow{y}_1$, it can generate a result vector $\overrightarrow{y}$, which is carried out in field of real number, namely,

$$\begin{aligned} \overrightarrow{y}_1 &= \widehat{R}\,\overrightarrow{l}, \\ \overrightarrow{y} &= \widehat{Q}\,\overrightarrow{y}_1. \end{aligned} \tag{8}$$

Input:
$\widehat{A}$, PK_3 and EK;
Output:
$\widehat{Q}$, $\widehat{R}$ and v
(1) Step 1: QR decomposition of $\widehat{A}$
(2) Decompose $\widehat{A}$ into $\widehat{Q}$ and $\widehat{R}$.
(3) Step 2: obtaining v
(4) Generate $\overrightarrow{m}$.
(5) Compute v with (EK, $\overrightarrow{m}$).
(6) Step 3: return $\widehat{Q}$, $\widehat{R}$, and v

Algorithm 3: Compute algorithm.

Input:
$(\widehat{Q}, \widehat{R})$, v, $\overrightarrow{l}$, PK_1 and VK;
Output:
V_{YES} or V_{NO}
(1) Step 1: checking the orthogonality of $\widehat{Q}$
(2) Get $\overrightarrow{y}_1$ with $(\widehat{R}, \overrightarrow{l})$.
(3) Produce $\overrightarrow{y}$ with $(\widehat{Q}, \overrightarrow{y}_1)$.
(4) Compute $\overrightarrow{y}_2$ with $(\widehat{Q}^T, \overrightarrow{y})$.
(5) $\overrightarrow{y}_2 \stackrel{?}{-} \overrightarrow{y}_1 = 0$.
(6) Step 2: checking the correctness of $\widehat{Q}$ and $\widehat{R}$
(7) Compute $e(v, h)$.
(8) Compute $e(\prod_{j=1}^{n} PK_{1j}^{y_j}, \widetilde{h}) \cdot VK$.
(9) $e(v, h) = e(\prod_{j=1}^{n} PK_{1j}^{y_j}, \widetilde{h}) \cdot VK \longrightarrow V_{YES}$
(10) $e(v, h) \neq e(\prod_{j=1}^{n} PK_{1j}^{y_j}, \widetilde{h}) \cdot VK \longrightarrow V_{NO}$
(11) Step 3: return V_{YES} or V_{NO}

Algorithm 4: Verify algorithm.

(ii) It multiplies the vector $\overrightarrow{y}$ by $\widehat{Q}^T$ to obtain a new vector denoted by $\overrightarrow{y}_2$, where $\widehat{Q}^T$ is transpose of matrix $\widehat{Q}$. Due to property of orthogonal matrix, if $\overrightarrow{y}_2 = \widehat{Q}^T\overrightarrow{y} = \widehat{Q}^T\widehat{Q}\overrightarrow{y}_1 = \overrightarrow{y}_1$ is true, next step (b) is executed.

(b) If the following equation holds in the finite field, $\mathscr{V}$ outputs V_{YES}. Otherwise, it outputs V_{NO}:

$$e(v, h) = e\left(\prod_{j=1}^{n} PK_{1j}^{y_j}, \widetilde{h}\right) \cdot VK. \tag{9}$$

Suppose that the nonsingular matrix A can be decomposed into $A = QR$. Algorithm 5 (Solve) is executed by $\mathscr{C}$ to obtain both Q and R.

(a) $\mathscr{C}$ gets the transposed matrix P^T and the inverse matrix N^{-1} of N which needs to solve the inverse of the upper triangular submatrix N_i for $i = 1$ to t.

(b) Multiplying P^T by the left of $\widehat{Q}$ and N^{-1} by the right of $\widehat{R}$, the QR decomposition of matrix A can be obtained:

$$\begin{cases} Q = P^T \cdot \widehat{Q}, \\ R = \widehat{R} \cdot N^{-1}. \end{cases} \tag{10}$$

Input:
$\widehat{Q}$ and $\widehat{R}$;
Output:
Q and R
(1) Solve P^T and N^{-1}.
(2) Obtain Q and R by equation (10).
(3) Return Q and R.

Algorithm 5: Solve algorithm.

4. Protocol Analysis

In this section, $\mathscr{PVCMD}$-$\mathscr{QR}$ is analyzed from the perspectives of correctness, security, and efficiency.

4.1. Correctness Analysis

4.1.1. ProbGen Algorithm. Since the result of QR decomposition is unique when the main diagonal elements are positive in the upper triangular matrix, not all matrices can be decomposed and the square matrix to be decomposed must be invertible and nonsingular. Therefore, conditions of decomposition of the input matrix $\widehat{A}$ should satisfy $|\widehat{A}| \neq 0$.

In fact, after the matrix A is encrypted, this condition is still satisfied. From $\widehat{A} = PAN$, we get the equation $|\widehat{A}| = |P| \cdot |A| \cdot |N|$. Specifically, since both P and N are invertible matrices, $|P| \neq 0$ and $|N| \neq 0$. In addition, the privacy matrix A is a full rank matrix, $|A| \neq 0$.

4.1.2. Verify Algorithm. Considering the property of the orthogonal matrix, there is $Q^TQ = I$, where I represents the identity matrix. If the vectors $\overrightarrow{y}_2$ and $\overrightarrow{y}_1$ are identical, $\widehat{Q}$ must be an orthogonal matrix. However, the matrix $\widehat{R}$ can be observed directly. The results are correct if the parties involved in the scheme execute the agreement honestly.

Before verification, it is necessary for $\mathscr{V}$ to compute the result vector $\overrightarrow{y}$, which can be obtained by $\overrightarrow{y} = \widehat{Q}(\widehat{R}\overrightarrow{l})$.

Because of equations (11) and (12), equation (13) holds

$$\overrightarrow{b} \cdot \overrightarrow{l} = (\overrightarrow{s}\widehat{A})\overrightarrow{l} = \overrightarrow{s}(\widehat{Q}\widehat{R}\overrightarrow{l}) = \overrightarrow{s} \cdot \overrightarrow{y}, \tag{11}$$

$$b_i = \sum_{j=1}^{n} s_j \widehat{A}_{ji}, \quad 1 \leq i \leq n, \tag{12}$$

$$\begin{aligned} \sum_{i=1}^{n} b_i l_i &= \sum_{i=1}^{n} \left(\sum_{i=1}^{n} s_j \widehat{A}_{ji} \right) l_i \\ &= \sum_{i=1}^{n} s_j \left(\sum_{i=1}^{n} \widehat{Q}_{ji} \sum_{i=1}^{n} \widehat{R}_{ik} l_k \right) = \sum_{i=1}^{n} s_j y_j. \end{aligned} \tag{13}$$

According to equation (13), we have

$$
\begin{aligned}
e(\mathbf{v}, h) &= e\left(\prod_{i=1}^{n} m_i^{z_i}, h\right) = e\left(\prod_{i=1}^{n}\left(\prod_{j=1}^{n} \mathrm{PK}_{3j}^{\widehat{A}_{ji}}\right)^{z_i}, h\right) = e\left(\prod_{i=1}^{n}\left(\prod_{j=1}^{n} \widetilde{g}^{s_j \widehat{A}_{ji}}\right)^{z_i}, h\right) \\
&= e\left(\prod_{i=1}^{n} g^{\delta\left(\sum_{j=1}^{n} s_j \widehat{A}_{ji}\right) z_i}, h\right) = e\left(\prod_{i=1}^{n} g^{\delta b_i z_i}, h\right) = e\left(\prod_{i=1}^{n} g^{b_i l_i}, \widetilde{h}\right) e\left(\prod_{i=1}^{n} g^{b_i k_i}, \widetilde{h}\right) \\
&= e\left(g^{\sum_{j=1}^{n} s_j y_j}, \widetilde{h}\right) \prod_{i=1}^{n} e\left(g^{k_i}, \widetilde{h}\right)^{b_i} = e\left(\prod_{j=1}^{n} \mathrm{PK}_{1j}^{y_j}, \widetilde{h}\right) \cdot \mathrm{VK}.
\end{aligned} \tag{14}
$$

In short, equation (9) is established and the verification is successful and sound.

4.2. Security Analysis

Theorem 1. *The publicly verifiable computation scheme $\mathcal{PVCMD}$-$\mathcal{QR}$ is secure under the co-CDH in group G_1.*

Proof. We follow Definition 3 to illustrate the theoretical analysis for security of our proposed scheme.

To prove this theorem, there are two adversaries $\mathcal{A}$ and $\mathcal{B}$. Suppose that the adversary $\mathcal{A}$ has a very strong ability to destroy the soundness of $\mathcal{PVCMD}$-$\mathcal{QR}$ with a probability ε so that it can obtain important information of the scheme. However, the challenger $\mathcal{B}$ with these information from adversary $\mathcal{A}$ tries the best to address the co-CDH problem with a nonnegligible probability ε', and $\varepsilon \approx \varepsilon'$.

To break this assumption, challenger $\mathcal{B}$ accesses the random oracle $O_{\text{co-CDH}}$ which generates $g, g' = g^{\alpha} \in G_1$ and $h, h' = h^{\beta} \in G_2$ as the result of output in return and selects $\alpha, \beta \in F_P^*$.

Then, the challenger $\mathcal{B}$ simulates adversary $\mathcal{A}$ to carry out this soundness experiment:

Adversary $\mathcal{B}$ denotes $\widetilde{g}' = g'^{\delta}$ and $\widetilde{h}' = h'^{\delta}$ by selecting $\delta \in F_p^*$ randomly as well as generates the parameter para$'$, namely, para$' = (p, G_1, G_2, G_T, e, g, \widetilde{g}', h, \widetilde{h}')$. It uses parameters para$'$ to generate PK_1' and PK_3' respectively, which are shown as follows:

$$
\mathrm{PK}_1' = \left(\mathrm{PK}_{11}', \mathrm{PK}_{12}', \ldots, \mathrm{PK}_{1n}'\right), \tag{15}
$$

where $\mathrm{PK}_{1i}' = g'^{s_i}$, $(1 \le i \le n)$.

$$
\mathrm{PK}_3' = \left(\mathrm{PK}_{31}', \mathrm{PK}_{32}', \ldots, \mathrm{PK}_{3n}'\right), \tag{16}
$$

where $\mathrm{PK}_{3i}' = g'^{s_i}$, $(1 \le i \le n)$.

Then, it generates an auxiliary vector $m' \in G_1^{n \times 1}$, where $m_i' = \prod_{j=1}^{n} \mathrm{PK}_{3j}^{'\widehat{A}_{ji}}$ for i to n.

According to m' and PK_1', it computes the public key PK_2' which is a vector. In other words, the expression of PK_2' is determined directly:

$$
\mathrm{PK}_2' = \left(\mathrm{PK}_{21}', \mathrm{PK}_{22}', \ldots, \mathrm{PK}_{2n}'\right). \tag{17}
$$

We take each element of PK_2' as the following:

$$
\mathrm{PK}_{2i}' = \left(\frac{e\left(m_i^{'z_i}, h\right)}{e\left(\mathrm{PK}_{1i}^{'y_i}, \widetilde{h}'\right)}\right)^{(1/b_i)} = \left(\frac{e\left(\left[\prod_{j=1}^{n} \mathrm{PK}_{3j}^{'\widehat{A}_{ji}}\right]^{z_i}, h\right)}{e\left(\mathrm{PK}_{1i}^{'y_i}, \widetilde{h}'\right)}\right)^{(1/b_i)}. \tag{18}
$$

There is an important condition for the above expression to be correct, namely, $b_i \neq 0$. Since the matrix $\widehat{A}$ is full rank and $\overrightarrow{s} \in F_p^*$, the vector $\overrightarrow{b} = \overrightarrow{s} \cdot \widehat{A}$ is a nonzero vector. In addition, it defines evaluation key $\mathrm{EK}' = (z_1, \ldots, z_i, \ldots, z_n)$.

Therefore, the challenger $\mathcal{B}$ can obtain some corresponding information eventually to complete this experiment such as PK_2', para$'$, and EK'.

Secondly, different from the real output of the KeyGen algorithm, the distribution of the output of the random oracle O_{keyGen} is independent and indistinguishable. So, we have reason to believe these two facts:

(a) According to the vectors $\overrightarrow{b}$ and $\overrightarrow{y} = \widehat{Q}(\widehat{R}\,\overrightarrow{l})$, the following formula must be correct:

$$
e\left(\prod_{i=1}^{n} m_i^{'z_i}, h\right) = e\left(\prod_{j=1}^{n} \mathrm{PK}_{1j}^{'y_j}, \widetilde{h}'\right) \cdot \prod_{i=1}^{n} \mathrm{PK}_{2i}^{'b_i}. \tag{19}
$$

(b) The vector $\overrightarrow{m}$ and the key $\mathrm{PK} = (\mathrm{PK}_1, \mathrm{PK}_2, \mathrm{PK}_3)$ are identical to the statistically distribution of $\overrightarrow{m'}$ and $\mathrm{PK}' = (\mathrm{PK}_1', \mathrm{PK}_2', \mathrm{PK}_3')$ severally.

Then, adversary $\mathcal{A}$ takes advantage of $\widehat{A}$ and PK_2' to access the random oracle O_{ProbGen}. The attacker $\mathcal{B}$ imitates O_{ProbGen} and takes the matrix $\widehat{A}$ and VK' as output, where we denote $\mathrm{VK}' = \prod_{i=1}^{n} \mathrm{PK}_{2i}^{'b_i}$.

Finally, adversary $\mathcal{A}$ will return a value v and fake results $(\widehat{Q}^*, \widehat{R}^*)$ with $\widehat{Q}^* \widehat{R}^* \neq \widehat{A}$. Then, challenger $\mathcal{B}$ will calculate the result vector $\overrightarrow{y}^* = \widehat{Q}^*(\widehat{R}^* \overrightarrow{l})$ and verify whether $\overrightarrow{y}^* = \overrightarrow{y}$ is true. If the invalid result $\overrightarrow{y}^*$ passes this verification, it means that challenger $\mathcal{B}$ has failed and does not break the assumption. Otherwise, it returns the following expression

of $g^{\alpha\beta}$ and declares that the difficult assumption co-CDH is broken, while it is impractical:

$$g^{\alpha\beta} = \left(\frac{v}{\prod_{i=1}^{n} m_i^{'z_i}}\right)^{\left[\delta\sum_{j=1}^{n} s_j\left(y_j^* - y_j\right)\right]^{-1}} = \left(\frac{v}{\prod_{i=1}^{n}\left[\prod_{j=1}^{n} \mathrm{PK}_{3j}^{'\widehat{A}_{ji}}\right]^{z_i}}\right)^{\left[\delta\sum_{j=1}^{n} s_j\left(y_j^* - y_j\right)\right]^{-1}}. \tag{20}$$

Now, we are going to provide the specific process of the above solving $g^{\alpha\beta}$.

If the wrong vector $\overrightarrow{y}^*$ passes the verification, the following equation (21) must be true:

$$e(v,h) = e\left(\prod_{j=1}^{n} \mathrm{PK}_{1j}^{'y_j^*}, h'\right)\cdot \mathrm{VK}'. \tag{21}$$

Considering equation (18), it is achieved that

$$e\left(\prod_{i=1}^{n} m_i^{'z_i}, h\right) = e\left(\prod_{j=1}^{n} PK_{1j}^{'y_j}, \widetilde{h}'\right)\cdot \mathrm{VK}'. \tag{22}$$

To divide equations (21) with (22), we obtain

$$\begin{aligned} e\left(\frac{v}{\prod_{i=1}^{n} m_i^{'z_i}}, h\right) &= e\left(\prod_{j=1}^{n} \mathrm{PK}_{1j}^{'\left(y_j^* - y_j\right)}, \widetilde{h}'\right) \\ &= e\left(\prod_{j=1}^{n} g^{'s_j\left(y_j^* - y_j\right)}, \widetilde{h}'\right). \end{aligned} \tag{23}$$

As $g' = g^{\alpha}, \widetilde{h}' = h^{\delta\beta}$, and $m_i' = \prod_{j=1}^{n} \mathrm{PK}_{3j}^{'\widehat{A}_{ji}}$, it is concluded that

$$\begin{aligned} e\left(\frac{v}{\prod_{i=1}^{n} m_i^{'z_i}}, h\right) &= e\left(\prod_{j=1}^{n} g^{\alpha s_j\left(y_j^* - y_j\right)}, \widetilde{h}'\right) \\ &= e\left(g^{\alpha\beta\sum_{j=1}^{n}\delta s_j\left(y_j^* - y_j\right)}, h\right). \end{aligned} \tag{24}$$

Hence, if $(y_j^* - y_j) \neq 0$ and $\delta s_j \neq 0$, there is

$$g_{j=1}^{\alpha\beta\sum n\delta s_j\left(y_j^* - y_j\right)} = \frac{v}{\prod_{i=1}^{n} m_i^{'z_i}}, \tag{25}$$

namely,

$$g^{\alpha\beta} = \left(\frac{v}{\prod_{i=1}^{n}\left[\prod_{j=1}^{n} \mathrm{PK}_{3j}^{'\widehat{A}_{ji}}\right]^{z_i}}\right)^{\left[\delta\sum_{j=1}^{n} s_j\left(y_j^* - y_j\right)\right]^{-1}}. \tag{26}$$

Hence, if this scheme is destroyed by adversary $\mathcal{A}$ with a certain probability ε, challenger $\mathcal{B}$ is able to break the co-CDH with a nonnegligible advantage ε'. In summary, $\mathcal{PVCMD}$-$\mathcal{QR}$ is secure under the co-CDH in group G_1.

4.3. Efficiency Analysis. In this section, we intend to give a detailed analysis of the computational overhead of $\mathcal{PVCMD}$-$\mathcal{QR}$.

The matrix N, where the order of each submatrix N_i for $i = 1$ to t is chosen randomly from 2 to w ($w \ll n$), is produced by $\mathcal{C}$, so there will be many combinations in reality. However, since the matrix N is sparse with the computational complexity $O(n^2)$ for solving N^{-1}, the computational overhead is not taken into consideration about generation of N and N^{-1}.

To simplify the analysis, suppose that each submatrix N_i is a w-order upper triangular matrix in the main diagonal of N. However, the inverse matrix of the w-order upper triangular matrix is obtained easily, so it is convenient to obtain the inverse matrix N^{-1} of N, where the inverse N_i^{-1} of submatrix N_i ($1 \le i \le t$) is placed in the corresponding position, like this

$$N^{-1} = \begin{pmatrix} N_1^{-1} & 0 & \cdots & 0 \\ 0 & N_2^{-1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & N_t^{-1} \end{pmatrix}_{n\times n}. \tag{27}$$

Therefore, we suppose that the order n of the original matrix A should meet this condition, namely, $n = wt$.

In KeyGen algorithm, three vectors $\overrightarrow{s}$, $\overrightarrow{l}$, and $\overrightarrow{k}$ which are generated randomly require $3n$ random numbers in the group operation. Next, it needs to calculate PK_1, PK_2, PK_3, and EK separately. The public key PK_1 is an n-dimensional vector where there is $\mathrm{PK}_{1i} = g^{s_i}$, so $\mathcal{DC}$ will execute n exponential operations to obtain PK_1. Since the public key PK_2, an n-dimensional vector, is obtained by n exponential operations and n pairing operations similarly, where $\mathrm{PK}_{2i} = e(g^{k_i}, \widetilde{h})$, additionally, n exponential operations needs to be performed to get the public key PK_3. As the evaluation key EK is also an n-dimensional vector, $\mathcal{DC}$ should perform n additions.

To perform ProbGen algorithm where we have $\widehat{A} = \mathrm{PAN}$, $\mathcal{C}$ needs to use a sparse block diagonal upper triangular matrix N and a unit permutation matrix P. Therefore, there are $n^2 + (1/2)(w+1)n^2$ multiplications and $(1/2)(w-1)n^2$ additions in encryption operation. On the contrary, $\mathcal{C}$ also computes a verification key $\mathrm{VK} = \prod_{i=1}^{n} \mathrm{PK}_{2i}^{b_i}$ with the vector $\overrightarrow{b}$. To get the vector, it is going to perform n^2 multiplications. Therefore, both n exponentials and $n-1$ multiplications should be required in the process of generating VK.

$\mathcal{CS}$ executes the QR decomposition of the matrix $\widehat{A}$ according to Compute algorithm. It is necessary to produce the value v, where this process involves n exponentiation operations and $n-1$ multiplications. However, before generating the value v, it should utilize the public key PK_3 to get an n-dimensional auxiliary vector $\overrightarrow{m}$, which requires n^2 exponential and $n(n-1)$ multiplications operations.

Table 1: Computation cost of each phase in $\mathcal{PVCMD}$-$\mathcal{QR}$.

Algorithm	Computation cost
KeyGen	$3n\text{Ex} + n\text{Pa} + n\text{Ad} + 3n\text{Ge}$
ProbGen	$[(1/2)(w+5)n^2 + n - 1]\text{Mu} + n\text{Ex} + (1/2)(w-1)n^2\text{Ad}$
Compute	$\text{De} + (n^2-1)\text{Mu} + (n^2+n)\text{Ex}$
Verify	$(1/2)(5n^2+3n)\text{Mu} + n\text{Ex} + (3/2)(n^2-n)\text{Ad} + 2\text{Pa}$
Solve	$[(1/4)(w+5)n^2 + T_1 n]\text{Mu} + [(1/4)(w-1)n^2 + T_2 n]\text{Ad}$

Table 2: Computation cost of $\mathcal{PVCMD}$-$\mathcal{QR}$ scheme for different problem sizes.

Dimension	KeyGen (ms)	ProbGen (ms)	Compute (ms)	Verify (ms)	Solve (ms)
$n = 400$	33.910000	13512.718000	167927.372000	88.412800	11574.687000
$n = 500$	43.037000	28093.517000	264634.147000	139.749000	22764.303000
$n = 600$	52.619000	55298.470000	377979.489000	203.310000	39072.307000
$n = 700$	62.410000	97193.756000	513340.065000	278.004000	61935.196000
$n = 800$	70.836000	159190.287000	670820.937000	365.882000	92610.783000

In Verify algorithm, $\mathcal{V}$ must generate an n-dimensional result vector $\overrightarrow{y}$, firstly, where $(1/2)n(n+1) + n^2$ multiplications and $(1/2)(n-1)n + n(n-1)$ additions should be performed. When checking the orthogonal property of the matrix $\widehat{Q}$, $\mathcal{V}$ is asked to compute n^2 multiplications. Then, it takes advantage of v, $\overrightarrow{y}$, and VK to verify whether $e(v,h) = e(\prod_{j=1}^{n} \text{PK}_{1j}^{y_j}, \widetilde{h}) \cdot \text{VK}$ holds, which needs n exponentiation operations, $n-1+1$ multiplications, and two pairing operations in this phases.

We also should take the computation cost of *Solve* algorithm. $\mathcal{C}$ has to decrypt the matrices $\widehat{Q}$ and $\widehat{R}$ to obtain the result of QR decomposition of original matrix A. Therefore, n^2 multiplication operations are carried out for solving $Q = P^T\widehat{Q}$. In order to compute $R = \widehat{R}N^{-1}$, it will deal with $(1/4)(w+1)n^2 + ((1/4)w + (1/3) - (1/12)w^2)n$ multiplications and $(1/4)(w-1)n^2 + ((1/4)w + (1/3) - (1/12)w^2)n$ additions.

Here, we denote an exponentiation operation with Ex, a multiplication operation with Mu, an addition operation with Ad, a pairing operation with Pa, a matrix decomposition with De, and a random number generation operation with Ge. T_1 and T_2 are described as follows:

$$\begin{cases} T_1 = \dfrac{1}{4}w + \dfrac{1}{3} - \dfrac{1}{12}w^2, \\ \\ T_2 = \dfrac{1}{4}w - \dfrac{1}{6} - \dfrac{1}{12}w^2, \end{cases} \tag{28}$$

where $w \ll n$.

In summary, the computation cost of each algorithm is shown in Table 1.

According to the above analysis, the computational complexity of the client is $O(n^2)$ and is lower than to accomplish QR decomposition directly.

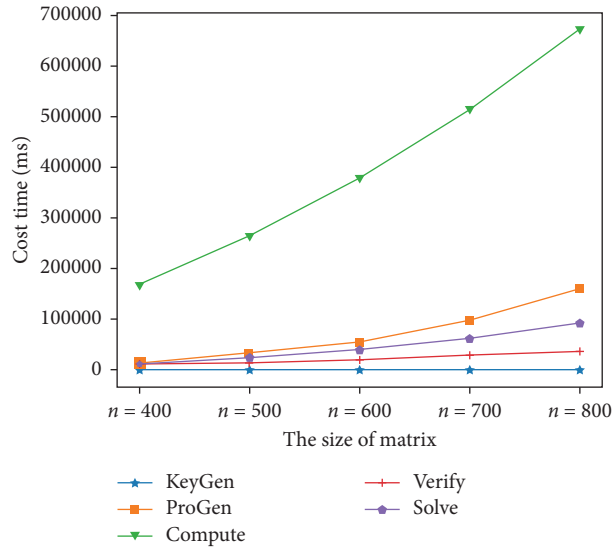


Figure 3: Computational time cost for each algorithm.

4.4. Experiment Analysis. Here, we evaluate the proposed scheme with experiments. Using C language, we emulate the data center DC, the client C, the cloud server CS, and the verifier V on a laptop with Intel Core(TM) i5-8265U CPU processor, 8 GB RAM memory.

To better describe the computational efficiency of the proposed $\mathcal{PVCMD}$-$\mathcal{QR}$ scheme, we simulate all these algorithms in our scheme (i.e., KeyGen, ProbGen, Compute, Verify, and Solve). First, we assume that the order of each submatrix of the block diagonal matrix N is identical, $w = 25$. The computation costs with different scales of the problem are listed in Table 2, and the specific trend is shown in Figure 3. The experiment shows that the overhead of the client side is smaller than the CS, as listed in Table 3.

Table 3: Comparison of computation cost between the client and cloud side.

Dimension	Client cost in $\mathcal{PVCMD}$-$\mathcal{QR}$ (ms)	Cloud server cost in $\mathcal{PVCMD}$-$\mathcal{QR}$ (ms)
$n = 400$	25175.817800	167927.372000
$n = 500$	50997.569000	264634.147000
$n = 600$	94574.087000	377979.489000
$n = 700$	159406.956000	513340.065000
$n = 800$	252166.952000	670820.937000

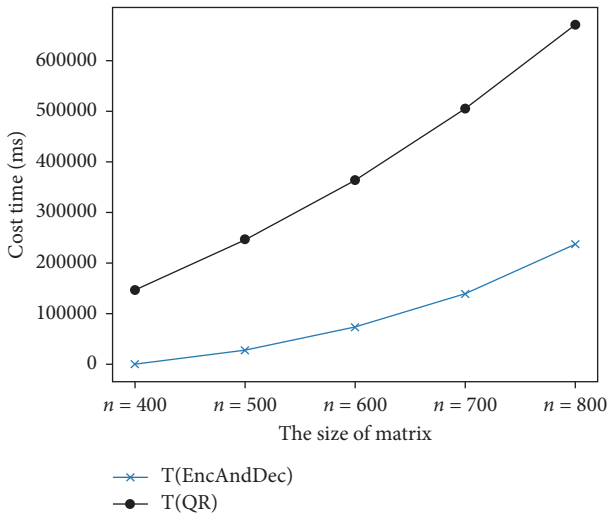


Figure 4: Efficiency comparison between T(EncAndDec) and T(QR).

Then, we illustrate the superiority of the outsourcing computation in Figure 4, in which we mainly consider the time cost of $\mathcal{C}$. In Figure 4, the symbol T(EncAndDec) represents the time cost of $\mathcal{C}$ in encryption and decryption phases of outsourcing process, and the symbol T(QR) means the time cost is required for the $\mathcal{C}$ to compute QR decomposition of matrix A directly. Compared to directly computing QR decomposition on the original matrix A, the $\mathcal{PVCMD}$-$\mathcal{QR}$ scheme is more efficient obviously as the dimension of matrix increases.

5. Conclusion

Aiming at the public verification outsourcing computation, this paper proposes a new publicly verifiable scheme with blockchain payment under the amortized model for QR decomposition of large-scale matrix. The sensitive data information is protected by using the sparse matrix. Therefore, client can upload his/her privacy matrix to the outsourcing service provider to perform QR decomposition. Simultaneously, the matrix digest technique is applied to the verification operation of outsourcing computation, which cuts down the workload of verifier dramatically. Afterwards, we also provide the specific theoretical proof of the correctness, safety, and efficiency of the $\mathcal{PVCMD}$-$\mathcal{QR}$ scheme, and the result proves that the scheme is secure under the co-CDH assumption.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by National Natural Science Foundation of China under Grant nos. 61902314, 62072371, and 61772418, Natural Science Basic Research Plan in Shaanxi Province of China under Grant no. 2018JZ6001, and Basic Research Program of Qinghai Province under Grants no. 2020-ZJ-701.

References

[1] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging it platforms: vision, hype, and reality for delivering computing as the 5th utility," *Future Generation Computer Systems*, vol. 25, no. 6, pp. 599–616, 2009.

[2] J. Ning, Z. Cao, X. Dong, K. Liang, L. Wei, and K.-K. R. Choo, "Cryptcloud+: secure and expressive data access control for cloud storage," *IEEE Transactions on Services Computing*, vol. 14, pp. 111–124, 2021.

[3] J. Ning, X. Huang, W. Susilo, K. Liang, X. Liu, and Y. Zhang, "Dual access control for cloud-based data storage and sharing," *IEEE Transactions on Dependable and Secure Computing*, no. 99, p. 1, 2020.

[4] R. Chow, P. Golle, M. Jakobsson et al., "Controlling data in the cloud: outsourcing computation without outsourcing control," in *Proceedings of the 2009 ACM Workshop on Cloud Computing Security*, pp. 85–90, ACM, Chicago, IL, USA, November 2009.

[5] Z. Yinghui, D. Robert, L. Ximeng, and Z. Dong, "Outsourcing service fair payment based on blockchain and its applications in cloud computing," *IEEE Transactions on Services Computing*, p. 1, 1939.

[6] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*, pp. 169–178, Bethesda, MD, USA, May 2009.

[7] X. Chen, "Introduction to secure outsourcing computation," *Synthesis Lectures on Information Security, Privacy, & Trust*, pp. 1–93, Morgan & Claypool, San Rafael, CA, USA, 2016.

[8] M. J. Atallah, K. N. Pantazopoulos, J. R. Rice, and E. E. Spafford, "Secure outsourcing of scientific computations," *Advances in Computers*, vol. 54, pp. 215–272, 2002.

[9] S. Salinas, C. Luo, X. Chen, W. Liao, and P. Li, "Efficient secure outsourcing of large-scale sparse linear systems of equations," *IEEE Transactions on Big Data*, vol. 4, no. 1, pp. 26–39, 2018.

[10] X. Chen, X. Huang, J. Li, J. Ma, W. Lou, and D. S. Wong, "New algorithms for secure outsourcing of large-scale systems of linear equations," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 1, pp. 69–78, 2015.

[11] X. Lei, X. Liao, T. Huang, and F. Heriniaina, "Achieving security, robust cheating resistance, and high-efficiency for outsourcing large matrix multiplication computation to a malicious cloud," *Information Sciences*, vol. 280, pp. 205–217, 2014.

[12] K. Jia, H. Li, D. Liu, and S. Yu, "Enabling efficient and secure outsourcing of large matrix multiplications," in *Proceedings of the 2015 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, IEEE, San Diego, CA, USA, December 2015.

[13] S. Zhang, H. Li, Y. Dai, J. Li, M. He, and R. Lu, "Verifiable outsourcing computation for matrix multiplication with improved efficiency and applicability," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 5076–5088, 2018.

[14] S. Zhang, C. Tian, H. Zhang, J. Yu, and F. Li, "Practical and secure outsourcing algorithms of matrix operations based on a novel matrix encryption method," *IEEE Access*, vol. 7, pp. 53823–853838, 2019.

[15] P. Golle and I. Mironov, "Uncheatable distributed computations," in *Cryptographers Track at the RSA Conference*, vol. 2020, pp. 425–440, Springer, Berlin, Germany, 2001.

[16] S. Benabbas, R. Gennaro, and Y. Vahlis, "Verifiable delegation of computation over large datasets," in *Annual Cryptology Conference*, pp. 111–131, Springer, Berlin, Germany, 2011.

[17] D. Fiore and R. Gennaro, "Publicly verifiable delegation of large polynomials and matrix computations, with applications," *IACR Cryptology ePrint Archive*, vol. 2012, p. 281, 2012.

[18] B. Parno, M. Raykova, and V. Vaikuntanathan, "How to delegate and verify in public: verifiable computation from attribute-based encryption," in *Proceedings of the Theory of Cryptography Conference*, pp. 422–439, Taormina, Italy, March 2012.

[19] A. C. Yao, "Protocols for secure computations," in *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science (SFCS 1982)*, pp. 160–164, IEEE, Washington, DC, USA, November 1982.

[20] K. Elkhiyaoui, M. Önen, M. Azraoui, and R. Molva, "Efficient techniques for publicly verifiable delegation of computation," in *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*, pp. 119–128, Xi'an, China, May 2016.

[21] H. Li, S. Zhang, T. H. Luan, H. Ren, Y. Dai, and L. Zhou, "Enabling efficient publicly verifiable outsourcing computation for matrix multiplication," in *Proceedings of the 2015 International Telecommunication Networks and Applications Conference (ITNAC)*, pp. 44–50, IEEE, Sydney, Australia, November 2015.

[22] X. Zhang, T. Jiang, K.-C. Li, A. Castiglione, and X. Chen, "New publicly verifiable computation for batch matrix multiplication," *Information Sciences*, vol. 479, pp. 664–678, 2019.

[23] O. E. Bronlund and T. L. Johnsen, "QR-factorization of partitioned matrices: solution of large systems of linear equations with non-definite coefficient matrices," *Computer Methods in Applied Mechanics and Engineering*, vol. 3, no. 2, pp. 153–172, 1974.

[24] S. Li, J. Wen, F. Luo, T. Cheng, and Q. Xiong, ""A location and reputation aware matrix factorization approach for personalized quality of service prediction," in *Proceedings of the 2017 IEEE International Conference on Web Services (ICWS)*, pp. 652–659, IEEE, Honolulu, HI, USA, June 2017.

[25] W. Lo, J. Yin, S. Deng, Y. Li, and Z. Wu, "An extended matrix factorization approach for QOS prediction in service selection," in *Proceedings of the 2012 IEEE Ninth International Conference on Services Computing*, pp. 162–169, IEEE, Honolulu, HI, USA, June 2012.

[26] J. Zhu, P. He, Z. Zheng, and M. R. Lyu, "Online QOS prediction for runtime service adaptation via adaptive matrix factorization," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 10, pp. 2911–2924, 2017.

[27] C. Luo, K. Zhang, S. Salinas, and P. Li, "SecFact: secure large-scale QR and LU factorizations," *IEEE Transactions on Big Data*, p. 1, 2017.

[28] R. Gennaro, C. Gentry, and B. Parno, "Non-interactive verifiable computing: outsourcing computation to untrusted workers," in *Annual Cryptology Conference*, pp. 465–482, Springer, Berlin, Germany, 2010.

[29] G. Sheng, C. Tang, W. Gao, and Y. Yin, "Md-$\mathscr{VC}_{\text{Matrix}}$: an efficient scheme for publicly verifiable computation of outsourced matrix multiplication," in *Proceedings of the International Conference on Network and System Security*, pp. 349–362, Springer, Taipei, Taiwan, September 2016.

[30] S. Micali, O. Goldreich, and A. Wigderson, "How to play any mental game," in *Proceedings of the Nineteenth ACM Symposium on Theory of Computing, STOC*, pp. 218–229, New York, NY, USA, January 1987.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 5523489, 10 pages
https://doi.org/10.1155/2021/5523489

Research Article

Controlled Sharing Mechanism of Data Based on the Consortium Blockchain

Jin Li,[1] Songqi Wu,[1] Yundan Yang,[1] Fenghui Duan,[1] Hui Lu,[2] and Yueming Lu[1]

[1]*School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing 100876, China*
[2]*Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou 510006, China*

Correspondence should be addressed to Hui Lu; luhui@gzhu.edu.cn

Received 19 January 2021; Revised 18 February 2021; Accepted 8 March 2021; Published 22 March 2021

Academic Editor: Qi Li

In the process of sharing data, the costless replication of electric energy data leads to the problem of uncontrolled data and the difficulty of third-party access verification. This paper proposes a controlled sharing mechanism of data based on the consortium blockchain. The data flow range is controlled by the data isolation mechanism between channels provided by the consortium blockchain by constructing a data storage consortium chain to achieve trusted data storage, combining attribute-based encryption to complete data access control and meet the demands for granular data accessibility control and secure sharing; the data flow transfer ledger is built to record the original data life cycle management and effectively record the data transfer process of each data controller. Taking the application scenario of electric energy data sharing as an example, the scheme is designed and simulated on the Linux system and Hyperledger Fabric. Experimental results have verified that the mechanism can effectively control the scope of access to electrical energy data and realize the control of the data by the data owner.

1. Introduction

Regarding the threat of data leakage, Verizon summarized the 2019 Data Breach Investigation Report (DBIR) to provide important points. In response to real data on 41,686 security incidents and 2013 data breaches from a total of 73 data sources from 86 countries, DBIR noted that the median direct loss to the threatened organization was $8,000 for a commercial e-mail threat and $25,000 for a computer data breach. Among them, there were 927 data leakage incidents in the financial and insurance industries. Network application attacks, abuse of privileges, and various errors accounted for 72%. It can be seen from this that data leakage losses from all walks of life are huge. In the process of storing and sharing data, there are mainly risks of data tampering and data leakage [1–4] due to single points of failure in centralised storage centres, malicious tampering, and inadequate access control mechanisms, so it is vital to find a way to achieve trusted storage and controlled flow of data.

Blockchain [5], as the core technology of recording the transaction book history of bitcoin system, has been widely concerned by all sectors of society since its inception. With the gradual development of Ethereum and Hyperledger Fabric, its features such as distributed storage and smart contract deployment and enforcement provide new ideas for solving data leakage problems in data sharing.

By taking advantage of the decentralized storage and data nontampering [6, 7] and data traceability features of blockchain, it is possible to achieve trusted storage of data and avoid the risks of centralised data storage such as single point of failure and data tampering.

The existing main methods of data protection using blockchain technology focus on the realization of secure data storage scheme [8]. However, after uploading the data to the blockchain, it is also crucial to ensure that the shared data can be trusted by controlling the boundaries of the data flow and achieving controlled data sharing. In traditional blockchain networks, all nodes are explicitly visible to the data on the chain, which does not apply to some sharing scenarios like electric energy data sharing. Therefore, blockchain data being visible to all user nodes can be a disadvantage in the realization. To address the

problem of data leakage due to explicit storage of data [9], consortium blockchain Hyperledger Fabric provides multichannel [10] data isolation protection method, so that the data is only visible to the joint maintenance account book of each organization and node in the channel, which enables effective control over the extent of data flows. However, there is still a risk of leakage after the nodes in the channel access the data on the chain during the sharing of data; at the same time, it does not meet the need for granular and complex access control of the data in the chain.

Therefore, in combination with attribute-based cryptography [11], it is possible to formulate data access policies for user-specific access and decryption. The data provider can formulate a data access policy based on the identities and attributes of the users to complete the granular access control of the data. Simultaneously, the data access process is recorded in the private account ledger that cannot be tampered by the data owner, so as to guarantee the traceability of data lifecycle processes. This paper proposes a mechanism that can realize trusted storage of data and granular access control and lifecycle management process for recording of data and take electric energy data sharing as an example to realize controlled sharing of electric energy data. The specific contributions of this paper are as follows:

(1) This paper presents a trusted storage scheme for electrical energy metadata by constructing electrical energy data storage consortium blockchain. Through the description of standardized metadata on the chain and the combination of distributed file system FastDFS, the chain aggregation storage of electric energy private data is realized, which solves the problems of high timing requirement and large amount of data in the storage of electric energy privacy data and provides the technical basis for controllable sharing and safe utilization of electric energy data.

(2) Using attribute-based encryption technology, based on the existing Fabric-CA in Hyperledger Fabric, implement user attributes key dynamic generation and safe distribution operations, which solve the problems of key abuse and privacy data leakage due to the ability of private key generator to decrypt all data in traditional attribute-based encryption technology. It realizes the data owner to formulate a data access policy based on the identities and attributes of the users to complete the granular access control of the data. It also effectively solves the key distribution challenges associated with traditional ABE encryption schemes.

(3) Using the privacy data mechanism proposed by Hyperledger Fabric [12], the data owner records the data access process to form a private ledger that can be seen only by the access data participants, which is used as the maintenance ledger of their own data to ensure the traceability of the controlled data flow process.

The related work and background are introduced in the second section. The third section shows the controlled sharing mechanism of electrical energy data based on consortium blockchain. The fourth section presents the experimental results and analysis. The fifth section provides a summary of the paper and puts forward the direction of future work.

2. Related Work and Background

For the study of trusted storage and access control of electrical energy data based on blockchain, a cloud blockchain fusion model (CBFM) is proposed in [13]. The power data is accurately identified through the image of parallel vision system in the cloud, and the power data storage scheme based on blockchain is implemented by using Hyperledger Fabric, which solves the problem of safe and accurate storage of electric energy data, but it does not consider the problems of data leakage in the process of storage and sharing of a large amount of electric energy data. A blockchain-based multiparty computing scheme is proposed in [14], and solutions are proposed for the fairness issues in MPC, as well as a solution for the secure sharing of data [15]. An SGX-based approach to blockchain for IoT applications is presented. Multiple Intel Software Guard Extensions (SGX) distributed Oracle servers are utilized to ensure data availability, combined with Intel SGX and TLS communication to ensure data integrity. In [16], a blockchain block authentication scheme based on group signatures is proposed. The solution is proposed to address the problem of limited computing resources of mobile blockchain devices. It also ensures the traceability of transaction data and distributed deployment of computational resources.

In [17], a trusted data acquisition model for power systems is proposed in conjunction with blockchain technology. The model realizes the authenticity of the underlying equipment state parameters of the power grid. In order to protect the privacy information in the power consumption data, a blockchain-based privacy data and identity protection scheme is proposed in [18]. The group membership data is recorded in a private blockchain, and, by using pseudonyms, the user's private identity within the group is hidden, and fast authentication of identity is achieved in combination with a Bloom filter. To address the issue of data privacy and leakage in IoT systems, a blockchain-based IoT architecture [19] has been proposed, which enables data access control, privacy, and confidentiality of data shared in a blockchain-based IoT ecosystem. It uses the attribute encryption (ABE) technique to ensure authenticity and ensure the privacy and confidentiality of shared data in the IOT [20, 21]ecosystem based on blockchain.

Reference [22] proposes a framework for storage sharing based on blockchain, IPFs, and ABE. Complete policy control of data access by the owner by distributing keys for blockchain transactions realizes data encryption sharing and granular access control in distributed storage Ethereum system.

It can be seen from the above research that, combined with the storage structure of distributed file system and attribute-based encryption algorithm, the trusted storage and controlled sharing mechanism of electric energy data can be realized by building the consortium blockchain, which can be used as a continuous framework for the interaction of electric energy data calculation and storage, so as to meet the application requirements of large-scale electric energy data trusted sharing in the future.

2.1. Blockchain and Hyperledger. Bitcoin, as the earliest technical application of blockchain technology, has attracted widespread attention because of its decentralized, unalterable, and traceable transaction characteristics. From a data perspective, blockchain technology is essentially a distributed database that collectively maintains and stores all historical transaction data in a decentralized and trustless way. The distributed ledger maintained by blockchain only supports query and addition but does not support modification and deletion. The use of hash linked list and Merkel tree structure ensures that no node can illegally tamper with the ledger.

Hyperledger Fabric [23, 24] is the representative of enterprise-level open-source blockchain. It has proposed many schemes in terms of permission control and privacy protection, in which version 1.2 has started to support the application of privacy Transaction (SideDB). The privacy transaction protection method caches the temporary database through the authorized endorser, synchronizes the transaction to other authorized endorsers and committers through the gossip protocol, and finally returns the hash value of the key-value pair of the private data to the client node to complete the endorsement. In the client phase, the client phase submits the hashes of the privacy data to the sorting service node for the normal winding-up process. After the block containing the transaction is synchronized to the whole network nodes, the authorized node checks and synchronizes the privacy data according to the authorization policy and then verifies the integrity of the privacy data according to the hash value of the public transaction. Finally, in the process of ledger submission, the authorized node updates from the temporary cache database to the private ledger to realize the recording and protection of privacy data.

Fabric CA is the digital certificate authentication center of Hyperledger, which mainly provides the functions of user identity registration, digital certificate issuance, and digital certificate extension and revocation. Before adding transaction information to Hyperledger Fabric, it is necessary to obtain legal identity authentication from authentication authorization node (CA peer) and then package the transaction information into blocks for broadcast throughout the network. All nodes in the network can verify the legitimacy and effectiveness of the transaction. Finally, the consensus mechanism is used to realize the consensus of all nodes in the network, and legal blocks are joined in the blockchain so that the information on transactions cannot be tampered with.

2.2. FastDFS Distributed File System. FastDFS [25] is an open-source lightweight distributed file system developed by Using C language, which can work well on UNIX-like systems and pursue high performance and high scalability. The overall design is based on the principle of simplicity and efficiency to solve the problem of large user access and large capacity file storage. FastDFS has good performance of redundant backup, load balancing, and online expansion, which is suitable for storing small-sized and medium-sized files, such as documents, pictures, and multimedia files.

FastDFS distributed file system is mainly composed of tracker, storage, and client [26]. Tracker is mainly responsible for the scheduling of storage, and multiple tracker clusters are formed in pairs to achieve load balancing. Storage is mainly responsible for file storage and redundant backup. FastDFS uses grouping mechanism to divide storage cluster into GROUPs and realizes load balancing, application isolation, and copy number customization independently among groups [27]. There can be multiple storage servers in the same group. The storage in the group is also peer-to-peer. The storages in the same group are connected with each other for file synchronization. The storage capacity of a group is subject to the storage with the smallest memory storage capacity of the group. When the system capacity is insufficient, the horizontal expansion can be realized by adding the group. When the storage access pressure in a group is too large, the vertical expansion can be realized by adding storage in the group. The client side of FastDFS is an application server using FastDFS access interface, which can be deployed on it by using its own development projects.

2.3. Attribute-Based Encryption Technology. Goyal et al. were the first to propose attributed-based encryption, which uses identity to define a series of attribute sets, and its definition is divided into key policy attribute-based encryption (KP-ABE) and ciphertext policy attribute-based encryption (CP-ABE) [28]. The CP-ABE is related to the secret message, the user's private key, and the set of attributes. The user can only decrypt the plaintext message for access control if the generated private key and the set of attributes embedded in the secret message match, and the access control policy matches exactly. Simultaneously, the granularity of the ciphertext accessibility control mechanism can be flexibly selected according to the strictness of the specified policy when the encryption or key is generated. In the application scenario of electric energy data sharing, the data owner determines the access user list of encrypted data, and the CP-ABE associated with decryption strategy and ciphertext can better meet the data demand of electric energy sharing and realize the access control of data on the chain.

3. System Model

The controlled sharing mechanism of data based on the consortium blockchain is mainly composed of the data storage consortium blockchain construction method, the distributed file system FastDFS application, and the distributed application DAPP (Decentralized Application) program development.

The construction of the data storage consortium blockchain ensures that the underlying data storage cannot be modified and uses attribute-based encryption to complete data access control to meet the needs of granular access control and secure sharing of data. Through the construction of a data flow transfer book, the original data life cycle management is recorded, and each data control is effectively recorded. Party's data transfer process. The distributed file system FastDFS solves the storage expansion problem of the data storage consortium blockchain and, based on its lightweight and developable nature, realizes the return of the source data ciphertext storage path and the source data file hash calculation operation, and the file hash is on the chain data storage data description providing a basis to authenticate the data integrity; the storage paths are used for recording in the current ledger records and the data providers can control the life cycle of data by changing the storage location. Distributed application (DAPP) is a decentralized operation application running on the blockchain network, which can better store user information and protect user privacy. In the controlled sharing mechanism of electric energy data based on the consortium blockchain, distribution through deployment of the distributed application DAPP realizes client operations such as data on-chain storage, controlled access, and data lifecycle management. This article takes the meter code table record storage and controlled access scenario in the electric energy metering system as an example to effectively solve the problem of safe storage and controlled sharing of electric energy data. The overall scheme is shown in Figure 1.

The electric energy data storage consortium chain utilizes the Hyperledger Fabric architecture at the bottom and uses the smart contract to realize the controlled access to the chain of electric energy data based on attribute-based encryption and the data life cycle management based on the fabric private data, so as to realize the safe storage, controlled access, and life cycle management of the electric energy data. On the distributed file system FastDFS, the return of the secret storage path of the source data of electric energy and the hash calculation of the source data file are realized, and the returned hash value is stored in the blockchain as the description of the data file, and, through the calculated hash value of the file, the electric energy data integrity verification function can be realized; the secret storage path of the source data of electric energy and the decryption key of the source data secret are used to access the transaction to form a private transaction. Recorded in the data owner's private ledger, the distributed application DAPP is used to realize client operations such as the storage of electric energy data on the chain and the completion of electric energy data access transactions. The specific construction process is as follows.

3.1. Construction of Electric Energy Data Storage Consortium Blockchain. The structure of the power energy data storage union chain is shown in Figure 2. The system is composed of a variety of intelligent terminal devices, each collection master system, blockchain system, FastDFS distributed file system, and distributed application (DAPP) integrated client. After the electric energy data is generated by intelligent terminal equipment, it is transmitted to the main station of acquisition system through wireless transmission network or optical fiber network. The main station of the system is composed of data center and control center.

The control center realizes client visualization by building DAPP. The data center realizes distributed storage by using FastDFS. The control center encrypts the source data to the data center through DAPP and transmits the file hash and data description and source data returned by FastDFS to the consortium blockchain network through Fabric-SDK-Go interface. The consortium blockchain calls smart contracts to aggregate and process the electric energy data to form metadata. After the data is standardized, the data is encrypted with attributes and is then uploaded. The nodes of the consortium blockchain run a consensus algorithm together and enter the data into the electrical energy data store through audit checks. The consortium blockchain forms a ledger structure to realize the decentralized safe and reliable storage and access control of electric energy data. Each data owner and access node initiate data access request and reply through smart contract and form access transaction records in the participant's private ledger to realize the data owner's life cycle management of data.

3.2. Electric Energy Data Access Control Policy. CP-ABE is used to implement an access control solution for electrical energy data storage consortium blockchain sharing. With the help of Fabric-CA module, CP-ABE initialization, key generation, and distribution operations are realized, and electric energy data encryption and chain operation are completed by using smart contract. The attribute definition of CP-ABE is realized by using channel ID, organization ID, and user ID in Hyperledger network as user attributes, and the access control policy is defined by the data provider to achieve access control of the data in the blockchain. The specific operation process is mainly divided into three stages: key generation and distribution, data encryption chain, and access control.

In the phase of key generation and distribution, the initialization and key generation and distribution are mainly completed by Fabric-CA and DAPP through Fabric-SDK-Go communication. By inputting the system security parameter λ, the main public key PK and key MK in CP-ABE scheme are generated:

$$\text{Setup}\left(1^{\lambda}\right) \longrightarrow (\text{MK}, \text{PK}). \tag{1}$$

The UCR is the certificate request submitted by the user, and Fabric-CA generates the user key SK related to the attribute set A for the user requestor using a randomization code based on the attribute set A in the user request and uses the user public key U_{PK} in UCR to encrypt the user key SK to form ciphertext CT_{usk} and attach the certificate U_{cert} issued by Fabric-CA for the user. Simultaneous interpreting cert. is sent to the user requester.

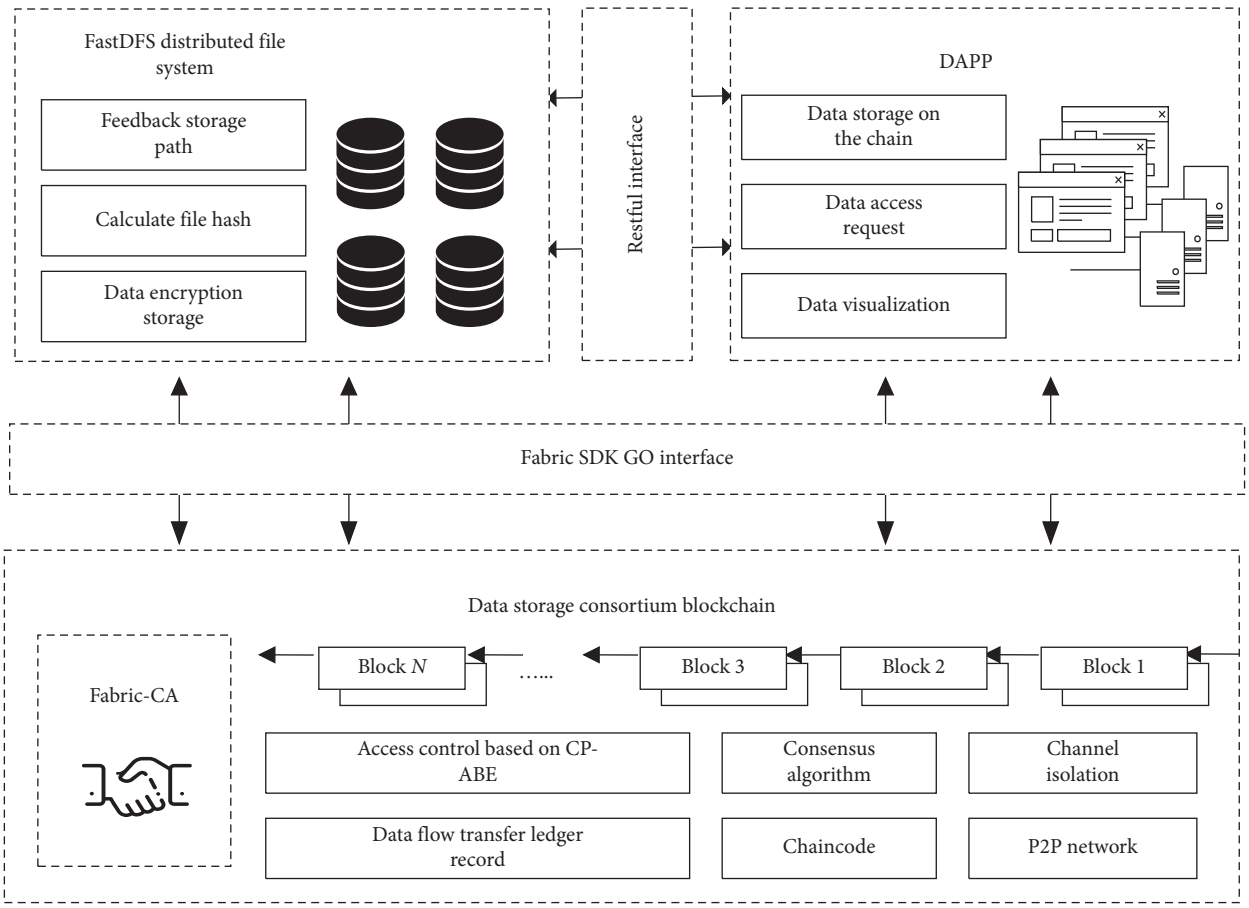


Figure 1: The overall scheme of controlled sharing mechanism of data based on the consortium blockchain.

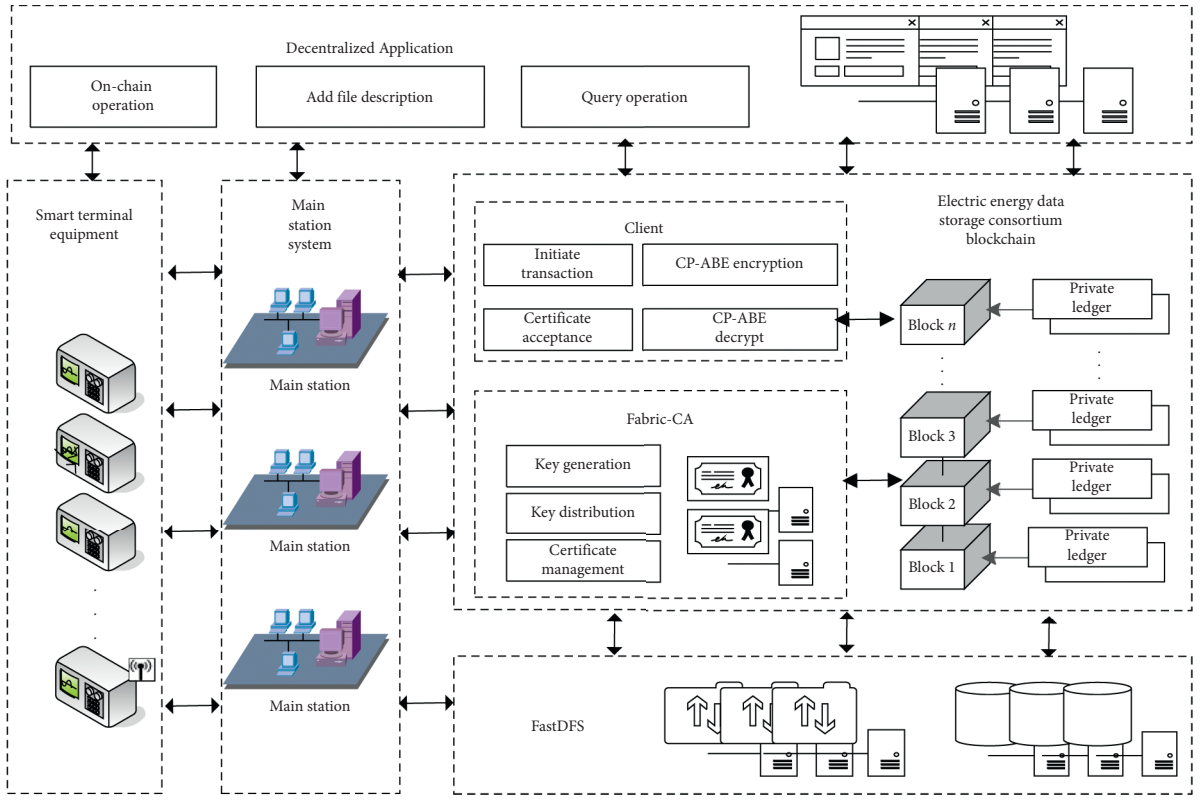


Figure 2: Electric energy data storage consortium blockchain structure.

$$\text{KeyGen}(\text{PK}, \text{MK}, A) \longrightarrow \text{USK}. \quad (2)$$

In the data encryption stage, before submitting the link-up request, the data owner uses the randomization algorithm to encrypt the submitted data in an attribute-based manner. The algorithm is input into the system public key PK and the data to be encrypted T_A and access control policy P_A generate ciphertext CT_A based on attribute encryption.

$$\text{Encrypt}(\text{PK}, T_A, P_A) \longrightarrow \text{CT}_A. \quad (3)$$

In the access control stage, after the data owner links the encrypted data to the chain, other users request the corresponding information of the transaction ciphertext in the blockchain network through the client to obtain the corresponding ciphertext CT_A. Decrypt the ciphertext by using the visitor attribute key USK. When the private key attribute meets the policy P_A in CT_A, the user can decrypt to get the plaintext M_A corresponding to the encrypted data, andimplement user level access control.

$$\text{Decrypt}(\text{CT}_A, \text{PK}, \text{USK}) \longrightarrow T_A. \quad (4)$$

3.3. Construction of Electric Energy Data Life Cycle Management Ledger. SideDB based on Hyperledger realizes the life cycle management of data for the data owners in the power energy data consortium blockchain. The transaction ledger is formed by recording the access process of the original data of the electric energy data, which is maintained in the private ledger of the data access participants.

The hash values of private transactions are also publicly recorded on the chain to enable verification of transactions. The data owner can complete the life cycle management and access control of the data by changing the source data storage path and data encryption key. The specific process of forming the ledger is shown in Figure 3.

Data visitors submit data access requests to data owners through DAPP. Data owners sign messages and verify their identities. For example, DAPP submits access transactions including data access party, data storage path, and data decryption key in FastDFS. Temporary database is cached by authorized endorser. Gossip protocol realizes message synchronous access transaction to other authorized endorsers and committers. Then, the hash value of the key-value pair of the access transaction data is returned to the data owner client node to complete the endorsement. The data providing client stage submits the hash value of the access transaction data to the ordering service node for normal uplink process. After the block containing the access transaction is synchronized to the whole network node, the transaction participant node checks and synchronizes the privacy data according to the authorization policy and then verifies the integrity of the privacy data according to the hash value of the public transaction. Finally, in the process of ledger submission, the transaction participant node updates from the temporary cache database to the private ledger to realize the access record of electric energy data and the control of the data owner on the original data.

Its smart contract design is shown in Algorithm 1. The user sends a transaction request to the accounting node and submits his own attribute set Role = $(r_1, r_2, \ldots, r_n)$, and the accounting node verifies according to the requested file KeyId and the search area and the blockchain ledger verifies whether the user complies with the access control policy of shared files. If the user matches, the accounting node will check whether it owns the metadata of the file and, if so, send the subkey to the requesting user. The user can decrypt the ciphertext to obtain the metadata set and download the file according to the metadata set. The data holder records the transaction behavior and records the user, file storage address, and attribute key in a personal privacy database.

4. Performance Analysis

A prototype experiment is designed to analyze the performance and feasibility of the solution. The experimental environment is configured with an Intel Core i5 processor, 16 GB of RAM, 460 GB of hard disk space, an Ubuntu 16.04LTS desktop, and programming languages Java and Go. The blockchain is deployed by Hyperledger Fabric. Three servers with official Fabric clients are deployed as blockchain nodes and smart contracts are deployed. According to the definition of access control policy for CP-ABE, three basic attributes are selected: channel ID, organization ID, and user ID. The three peers belong to the same channel CHANEL1 and two organizations Org1 and Org2, and the user IDs are CHANEL1. Org1. User1, CHANEL1. Org1. User2, and CHANEL1. Org2. User1. The access control policies are defined randomly.

The experimental data are recorded in the code table of China Southern Power Grid Co, Ltd. from 2014 to 2015, and the minimum data unit is about 120,000 15 MB data generated at the same time node. In order to verify the authenticity and effectiveness of the controlled sharing mechanism of electric energy data, three key links in the controlled sharing mechanism are selected for testing. The three key links are as follows: the system encrypts and stores the electric energy data to FastDFS, uploads the electric energy data description to the blockchain, and forms the account book of the electric energy data access transaction. We test the time consumption of each link.

4.1. Performance Test of Electric Energy Data Storage to FastDFS. In order to realize the reliable sharing of electric energy data, the power grid system is divided into different subregions in the production scenario, and the power consumption situation of the area is reported regularly. FastDFS is used to upload the hourly electric energy data and extract the file size, storage location, and other information as the description of electric energy data.

In the experiment, 15.0 MB files generated by 12,000 collection nodes were selected as the minimum granularity of upload data. The number of uploaded files was increased from 1 to 50, and the impact on the performance of data uploading to FastDFS distributed file system module was tested. The experimental results are shown in Figure 4, and

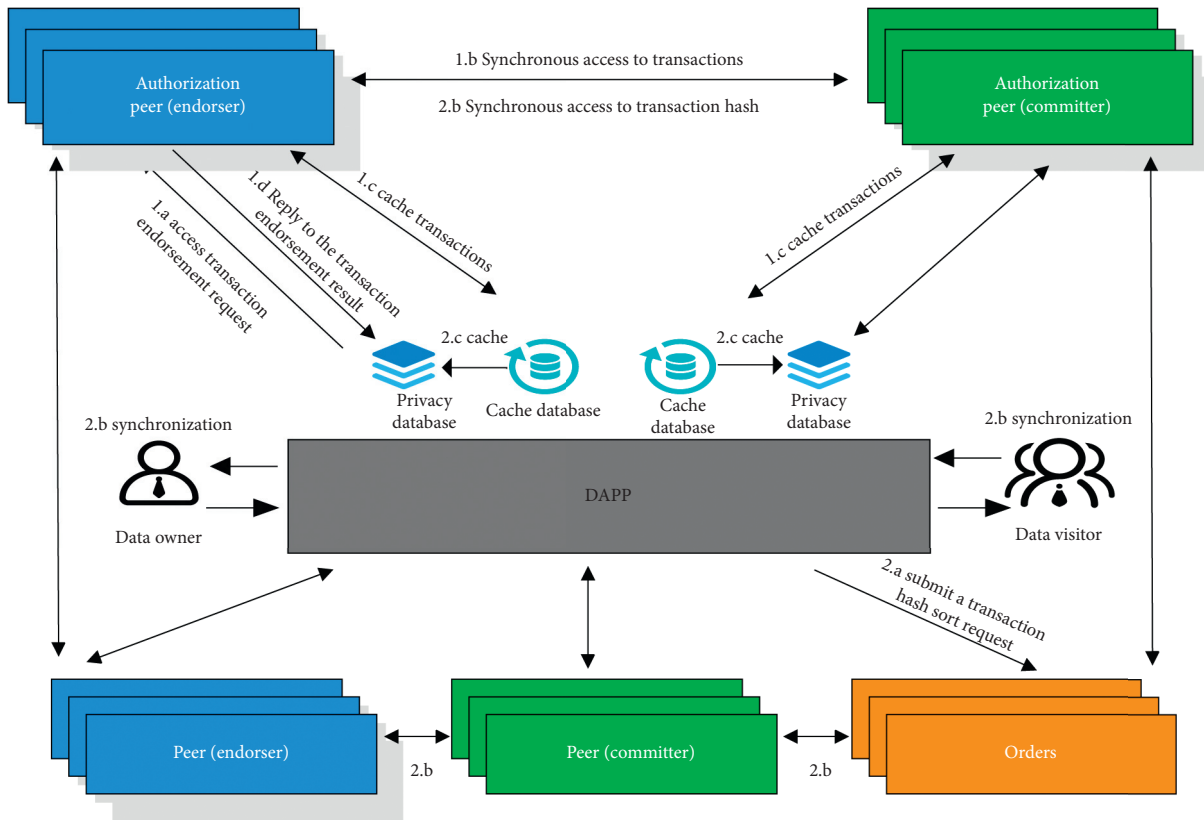


Figure 3: The process of forming the ledger.

Input: User, KeyID, node
Output: bool
(1) send Request To Node (KeyId, Role), ←User
(2) retrieve Ledger (KeyId)
(3) getAcp (KeyId)
(4) foreach $i \in$ Role
(5) **if** verifyRole (i) == ture **then**
(6) **break**
(7) **else**
(8) refuse
(9) flag = searchLocalDatabase (KeyId)←node
(10) **if** flag == ture **then**
(11) response (key_{share}) ⟶ User
(12) address = getPiter←User
(13) download (address)←User
(14) decrypt (key_{share})
(15) (User, uri, key_{share}) ⟶ SideDB
(16) **return** true;

Algorithm 1: Algorithm of data access transaction.

the upload time of electric energy data files to FastDFS increased from 213 ms to 12049 ms, the hash time increased from 42 ms to 3363 ms, and the total time increased from 255 ms to 15412 ms.

As the number of file uploads increases, the FastDFS upload storage and file hash calculation time increases linearly, and the time consumption of uploading data to FastDFS storage is relatively large.

4.2. Performance Test of Electric Energy Data Description Encryption Chain. The client node releases the electric energy data and uploads it to the blockchain request. The blockchain node requests to call the chain code and input the hash of the electric energy data file and other data descriptions as parameters, executes the chain code to realize CP-ABE encryption, and writes the execution result of the chain code into the blockchain ledger after

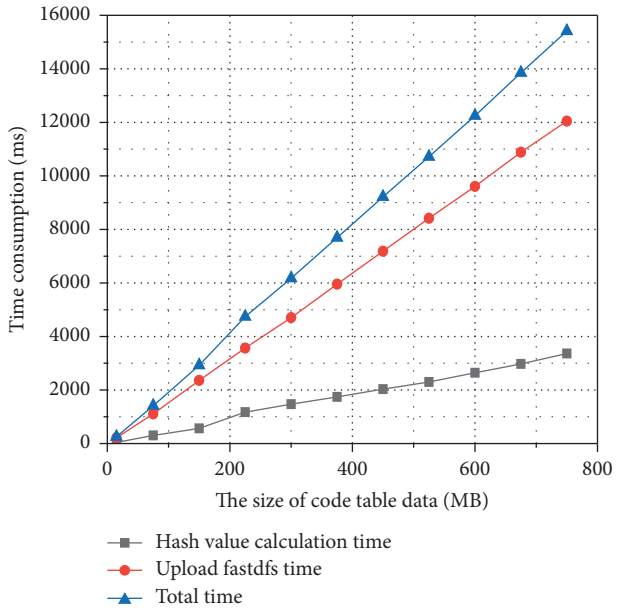


Figure 4: The experimental results of electric energy data storage to FastDFS.

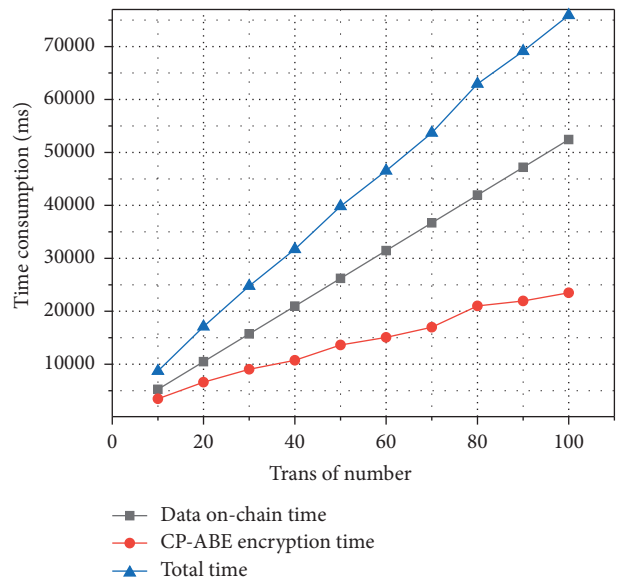


Figure 5: The experimental results of electric energy data description encryption chain.

reaching a consensus among the nodes. The experimental results are shown in Figure 5.

Assuming that 10~100 electric energy data records are added to the blockchain ore pool in the same period of time, the time for test encryption and chain connection to reach a consensus is about 8.72 s–75.916 s. The data encryption time is 3.47s~23.48 s, and the data link time is 5.25 s~52.43 ms. The reason is that DAPP is based on Fabric-SDK-Go platform. It needs docker-compose to generate fabric image, instantiates chain code to interact with fabric platform, and uses restful interface to call chain code to realize opening up, which is different from fabric throughput concept.

4.3. Performance Test of Electric Energy Data Access Transaction Ledger. The transaction ledger is formed by recording the access process of the original data of electric energy data, which is maintained in the privacy ledger of the data access participants.

The experiment measures the transaction delay and compares the access transaction query with the public transaction query. The results are shown in Figure 6. In a period of time, 10~100 access transactions are uploaded to form the access transaction account book, and the average time for reaching a consensus is initially 421 ms. As the network environment becomes stable, the average time consumption decreases to 361.61 ms. The average time consumption of public transaction query and access transaction query tends to be stable, with the average of 283.55 ms and 218.24 ms. It can be seen that the query efficiency of access transaction ledger stored in private ledger is lower than that of public transaction query, but it is within the acceptable range of users.

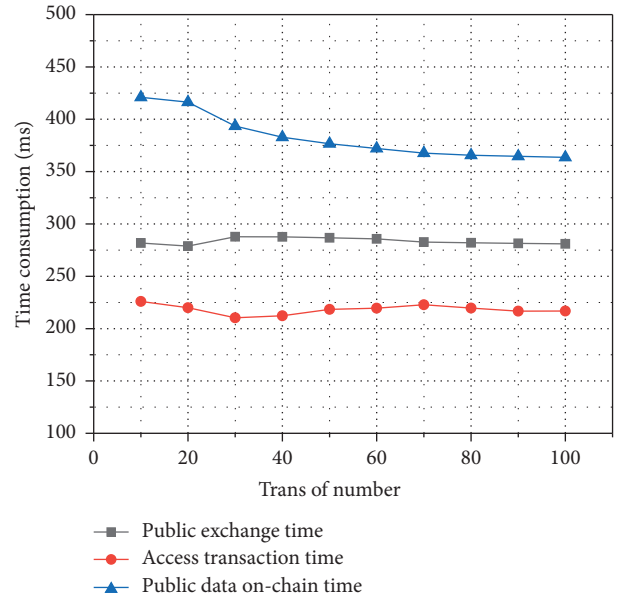


Figure 6: The experimental results of electric energy data access transaction ledger.

5. Conclusion and Future Work

This paper proposes a data-controlled sharing framework, which provides a new solution for data secure storage and controlled sharing. Realize the credible storage of data by building a data storage consortium blockchain, using ABE to complete data access control, meeting the need for granular access control and secure sharing of data, and controlling the scope of data flow; by building a data flow transfer book to record original data life cycle management, the data transfer process of each data controller is effectively recorded, so that the data owner can complete the life cycle management and

access control of the data by changing the source data storage path and data encryption key.

In the future, we will address the security issues facing the secure sharing of data and applications between blockchains. In this paper, although we propose a data-controlled sharing framework, it will be useful to maintain data sharing between multiple blockchains to meet data sharing scenarios.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Key R&D Program of China (Grant no. 2019YFB2102400).

References

[1] M. Li, Y. Sun, H. Lu, S. Maharjan, and Z. Tian, "Deep reinforcement learning for partially observable data poisoning attack in crowdsensing systems," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6266–6278, 2020.

[2] Z. Tian, C. Luo, J. Qiu, X. Du, and M. Guizani, "A distributed deep learning system for web attack detection on edge devices," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 1963–1971, 2020.

[3] Q. Tan, Y. Gao, J. Shi, X. Wang, B. Fang, and Z. Tian, "Toward a Comprehensive Insight Into the Eclipse Attacks of Tor Hidden Services," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1584–1593, 2019.

[4] H. Lu, C. Jin, X. Helu et al., "Research on intelligent detection of command level stack pollution for binary program analysis," *Mobile Network and Applications*, 2020.

[5] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the internet of things," *IEEE Access*, vol. 4, pp. 2292–2303, 2016.

[6] C. Xu, K. Wang, P. Li et al., "Making big data open in edges: a resource-efficient blockchain-based approach," *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 4, pp. 870–882, 2019.

[7] P. Danzi, A. E. Kalor, C. Stefanovic, and P. Popovski, "Analysis of the communication traffic for blockchain synchronization of IoT devices," in *Proceedings of the 2018 IEEE International Conference on Communications (ICC)*, pp. 1–7, Kansas City, MO, USA, May 2018.

[8] H. Yin, D. Guo, K. Wang, Z. Jiang, Y. Lyu, and J. Xing, "Hyperconnected network: a decentralized trusted computing and networking paradigm," *IEEE Network*, vol. 32, no. 1, pp. 112–117, 2018.

[9] H. Lu, C. Jin, X. Helu, C. Zhu, N. Guizani, and Z. Tian, "AutoD: intelligent blockchain application unpacking based on JNI layer deception call," *IEEE Network*, vol. 99, pp. 1–7, 2020.

[10] *A Blockchain Platform for the Enterprise—Hyperledger-Fabricdocs Master Documentation*, https://hyperledger-fabric.readthedocs.io/en/release-1.3/.

[11] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS 2006)*, pp. 89–98, Alexandria, VA, USA, October 2006.

[12] E. Androulaki, A. Barger, V. Bortnikov et al., "Hyperledger fabric: a distributed operating system for permissioned blockchains," in *Proceedings of the Thirteenth EuroSys Conference (EuroSys '18)*, pp. 1–15Article 30, New York, NY, USA, April 2018.

[13] S. Cao, J. Zou, X. Du, and X. Zhang, "A successive framework: enabling accurate identification and secure storage for data in smart grid," in *Proceedings of the ICC 2020—2020 IEEE International Conference on Communications (ICC) IEEE*, pp. 1–6, Dublin, Ireland, June 2020.

[14] H. Gao, Z. Ma, S. Luo, and Z. Wang, "BFR-MPC: a blockchain-based fair and robust multi-party computation scheme," *IEEE Access*, vol. 7, pp. 110439–110450, 2019.

[15] S. Woo, J. Song, and S. Park, "A distributed oracle using intel SGX for blockchain-based iot applications," *Sensors*, vol. 20, no. 9, p. 2725, 2020.

[16] S. Zhang and J. H. Lee, "A group signature and authentication scheme for blockchain-based mobile-edge computing," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4557–4565, 2019.

[17] T. Yang, F. Zhai, J. Liu, M. Wang, and H. Pen, "Self-organized cyber physical power system blockchain architecture and protocol," *International Journal of Distributed Sensor Networks*, vol. 14, no. 10, 2018.

[18] Z. Guan, G. Si, X. Zhang et al., "Privacy-preserving and efficient aggregation based on blockchain for power grid communications in smart communities," *IEEE Communications Magazine*, vol. 56, no. 7, pp. 82–88, 2018.

[19] Y. Rahulamathavan, C. W. Phan, S. Misra, and M. Rajarajan, "Privacy-preserving blockchain based IoT ecosystem using attribute-based encryption," in *Proceedings of the 2017 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, pp. 1–6, Bhubaneswar, India, December 2017.

[20] M. Shafiq, Z. Tian, A. K. Bashir, X. Du, and M. Guizani, "CorrAUC: a malicious Bot-IoT traffic detection method in IoT network using machine learning techniques," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3242–3254, 2020.

[21] J. Qiu, Z. Tian, C. Du, Q. Zuo, S. Su, and B. Fang, "A survey on access control in the age of Internet of Things," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 4682–4696, 2020.

[22] W. Shangping, Z. Yinglong, and Z. Yaling, "A blockchain-based framework for data sharing with fine-grained access control in decentralized storage systems," *IEEE Access*, vol. 6, pp. 38437–38450, 2018.

[23] S. Nathan, P. Thakkar, and B. Vishwanathan, "Performance benchmarking and optimizing hyperledger fabric blockchain platform," in *Proceedings of the 2018 IEEE 26th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, pp. 264–276, Milwaukee, WI, USA, September 2018.

[24] J. Sousa, A. Bessani, and M. Vukolic, "A byzantine fault-tolerant ordering service for the hyperledger fabric blockchain platform," in *Proceedings of the 2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pp. 51–58, Luxembourg City, Luxembourg, June 2018.

[25] H. Che and H. Zhang, "Exploiting FastDFS client-based small file merging," in *Proceedings of the International Conference on Artificial Intelligence & Engineering Applications*, pp. 242–246, Hong Kong, China, November 2016.

[26] X. Liu, Q. Yu, and J. Liao, "FastDFS: a high performance distributed file system," *ICIC Express Letters. Part B, Applications: An International Journal of Research and Surveys*, vol. 5, no. 6, pp. 1741–1746, 2014.

[27] M. R. Kaseb, M. H. Khafagy, I. A. Ali, and E. S. M. Saad, "Redundant independent files (RIF): a technique for reducing storage and resources in big data replication," in *Trends and Advances in Information Systems and Technologies*, pp. 182-183, Springer, Berlin, Germany, 2018.

[28] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proceedings of the IEEE Symposium on Security & Privacy*, pp. 321–334, Berkeley, CA, USA, May 2017.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 6623639, 10 pages
https://doi.org/10.1155/2021/6623639

Research Article

A Blockchain-Based Public Auditing Protocol with Self-Certified Public Keys for Cloud Data

Hongtao Li,[1] **Feng Guo**,[2] **Lili Wang**,[3] **Jie Wang**,[1] **Bo Wang**,[1] **and Chuankun Wu**[2]

[1]*College of Mathematics & Computer Science, Shanxi Normal University, Linfen 041000, China*
[2]*School of Information Science and Engineering, Linyi University, Linyi 276002, China*
[3]*College of Mathematics and Informatics, Fujian Normal University, Fuzhou 350117, China*

Correspondence should be addressed to Feng Guo; 25576152@qq.com

Received 11 December 2020; Revised 18 January 2021; Accepted 11 February 2021; Published 23 February 2021

Academic Editor: Yinghui Zhang

Cloud storage can provide a way to effectively store and manage big data. However, due to the separation of data ownership and management, it is difficult for users to check the integrity of data in a traditional way, which leads to the introduction of the auditing techniques. This paper proposes a public auditing protocol with a self-certified public key system using blockchain technology. The user's operational information and metadata information of the file are formed to a block after verified by the checked nodes and then to be put into the blockchain. The chain structure of the block ensures the security of auditing data source. The security analysis shows that attackers can neither derive user's secret key nor derive users' data from the collected auditing information in the presented scheme. Furthermore, it can effectively resist against not only the signature forging attacks but also the proof forging attacks. Compared with other public auditing schemes, our scheme based on the self-certified public key system has been improved in storage overhead, communication bandwidth, and verification efficiency.

1. Introduction

Cloud storage, which provides a way to effectively store and manage big data [1], is an important branch of cloud computing. Because cloud storage has superiorities of low cost, scalable, location-independent, and high performance [2–4], more and more individuals and businesses tend to outsource their data to the cloud. Although the advantages of cloud storage services are many and huge, it still faces a variety of security challenges [4–14].

For example, the security of data sharing and storage in the same group is an urgent issue to be solved in the cloud environment [6]. In other words, since the cloud users lose the management of data, a cloud service provider (CSP) must satisfy users' need for the security of stored data [7]. And users cannot verify the integrity of their data with traditional methods owing to the trust gap between users and CSP. In addition, cloud storage also faces many internal and external security threats [8–10]. Firstly, malicious attackers might do their best to retrieve users' outsourced data, even to destroy and delete the outsourced data. Then, the confidentiality, integrity, and availability of users' stored data are destroyed. Secondly, the user's outsourced data might also be illegally manipulated by CSP. For instance, CSP may selectively conceal certain errors in user's outsourced data due to Byzantine failures [11]. Furthermore, CSP might deliberately delete data that are rarely accessed by ordinary users in order to reduce storage space and save bandwidth [12, 13]. Finally, users may not be able to timely know the data changes and they may lack trust on CSP. Then, disputes arise, although those disputes may be caused by users' own improper operations [14]. Therefore, it is critical and significant to develop efficient data auditing techniques to check the confidentiality, integrity, and availability of stored data.

After data are outsourced to the cloud, users would delete local data and lose the management of outsourced data. Therefore, users can use audit technology to remotely verify whether the outsourced data are correct. The most core challenge of cloud data auditing is how to efficiently

check the cloud data integrity. To address this problem, a proof of retrievability (PoR) protocol [15] and a provable data possession (PDP) protocol [16] have been presented in 2007, respectively.

In typical PoR protocol, the user first encodes the data file with error-correcting code before outsourcing data to the CSP. Therefore, the user can reconstruct the entire file from the CSP's partial response. However, PoR protocol is applicable for static data. And it does not support third-party auditing and is a typical private auditing scheme. In private auditing, remote verification operation is performed directly between user and CSP. The user is the only source of verification results, while CSP and users do not trust each other and users cannot provide convincing auditing results for verification. Furthermore, user's burden is increased due to insufficient computing resources. Since one of the important motivations of outsourcing data is to reduce the user's burden of storage management, it is not recommended that users audit their data frequently.

To address this problem, a PDP protocol was first provided by Ateniese et al. [16]. In PDP, the RSA-based homomorphic authenticator is employed to check the data integrity and an independent authorized third-party auditor (TPA) was introduced. TPA can not only provide independent audit results but also bear the communication overhead and computation costs. Compared with PoR, PDP makes the process of verification more convenient and efficient and is more suitable for public auditing [5, 7, 10, 11, 16–25].

The public audit has advantages over private auditing, so it has attracted much attention of researchers. Since the idea of public auditing was raised in 2007 [16], a lot of auditing protocols have been designed in recent years [10, 12, 18–28].

In 2010, Wang et al. [22] also provided a similar architecture for public audit scheme with privacy-preserving property. To overcome the data leakage to the TPA, CSP integrates the aggregate value of the data blocks with random masking. However, the lack of strict performance analysis has greatly affected the practical application of the scheme. Furthermore, the length of data block must be equal to the size of cryptosystem. That means the storage space of tags generated for data blocks must be equal to the size of the original file [26]. This shows that the efficiency of the presented public auditing scheme is low. In order to improve the efficiency, Wang et al. [10] extended the above auditing protocol to multiuser settings. The extended protocol can support batch verification. However, the expected goal has not been achieved because the implementation of verification and updation brings higher computing and communication costs to TPA [27]. In 2011, Wang et al. [12] implemented complete data dynamics by using a Merkle hash tree (MHT), while the implementation of verification and updation also makes communication cost of protocol higher [21].

In 2013, Wang et al. [10] found that there was a risk of leakage of data information in the proposed scheme with public auditability [16]. Then, they designed a privacy-preserving scheme, which combined homomorphic linear authenticator (HLA) and random masking technique. Nevertheless, the designed scheme does not have the ability to protect the identity privacy of signers [28]. In order to reduce the computational cost and communication overhead, Zhu et al. [18] proposed a new public audit scheme based on index hash table (IHT), which is employed to organize the data properties for auditing. However, the index table is a sequence table. If you need to locate a certain element, it will take an average of half the total length of the table. This resulted in very efficient update operations, such as insertion and deletion [21]. In addition, these update operations would inevitably change the serial numbers of some blocks. Then, it is necessary to recalculate the tags of those blocks. In this way, CSP would require more extra computational costs and unnecessary communication overhead [19].

Then, Tian et al. [19] designed a public auditing protocol based on dynamic hash table (DHT) to support data dynamics, which claims to address the problem in Zhu's scheme [18]. The dynamic hash table is a single linked sequence table. Though the proposed public auditing protocol is efficient, there are still some drawbacks in this scheme. Firstly, because time stamps for verification are generated by the user and TPA only serves the user, CSP may suffer from the collusion attack launched by the user and TPA [21]. Secondly, there is no index switcher in the proposed scheme. Then, the relationship between the index number and the serial number of a certain data block cannot be clearly known. Finally, the proposed protocol still has relatively high computational costs.

In addition, Shen et al. [21] designed a novel public auditing protocol based on a new dynamic structure to overcome the drawbacks in [18]. The proposed dynamic structure consists of a doubly linked info table and a location array. Though the above protocols can effectively achieve public auditing, search operations in those schemes are relatively inefficient in the verification phase and the updating phase [27].

In 2018, Jin et al. [20] presented a scheme by employing an index switcher. Then, the relationship between the index number and the tag number of a certain data block can be clearly known. And there is no need to recalculate the tags caused by block update operations. Nevertheless, the index switcher needs to be periodically transformed among the systems, which will inevitably result in huge extra costs. Moreover, such an index switcher is not a complete structure. And how to switch between the two constituent tables is not explained in the proposed scheme [21].

In 2019, Ding et al. [29] proposed a public auditing protocol that is intrusion-resilient to mitigate the damage caused by key exposure problems. The protocol divides the lifetime of files stored in the cloud into several periods, each of which is further divided into several refreshing periods. The auditing key is updated every time period, and the secret value used to update the auditing key changes during each refreshing period. These two update operations are performed by the client and the third-party auditor (TPA).

In 2020, Garg et al. [30] proposed an efficient data integrity auditing method for cloud computing. The objective of this protocol is to minimize the computational complexity of the client during the system setup phase. Based on the

properties of bilinear pairings, the protocol is publicly verifiable and supports dynamic manipulation of data. The security of the protocol depends on the stability of the calculation of the Diffie–Hellman problem (CDHP) in the random oracle model (ROM).

The nature of blockchain is particularly suitable for data accounting and auditing. Because of its shared and fault-tolerant database, it has attracted the interest of the research community. Blockchain uses cryptography to build trust in peers to protect interactions of them. Meanwhile, it adopts consensus algorithm to ensure the block data are not changed, which is very suitable for data security in the cloud. In the past few years, some cloud security schemes based on blockchain have been proposed. Li et al. proposed a security framework for cloud data audit using blockchain technology, in which user's operational information on the file is formed to a block after validated by all checked nodes in the blockchain network and then put into the blockchain [31]. Linn et al. proposed a data auditing framework for health scenarios based on blockchain, in which blockchain was used as an access moderator to control the access to outsourced shared data [32]. Fu et al. introduced a privacy-aware blockchain-based auditing system for shared data in cloud applications [33]. Ghoshal et al. proposed an auditing mechanism based on blockchain structure, in which any user can perform the validation of selected files efficiently [34]. Fu proposed a blockchain-based secure data-sharing protocol under decentralized storage architecture [35]. Miao et al. proposed a decentralized and privacy-preserving public auditing scheme based on blockchain (DBPA), in which a blockchain is utilized as an unpredictable source for the generation of (random) challenge information, and the auditor is required to record the audit process onto the blockchain [36]. Li et al. proposed a public auditing scheme with the blockchain technology to resist the malicious auditors [37]. In addition, through the experimental analysis, we demonstrate that our scheme is feasible and efficient. Due to the limited capacity of blocks in the blockchain, only very important security information is considered to be stored in blocks; otherwise, the system performance will not be acceptable.

This paper proposes a public auditing protocol with a self-certified public key system using blockchain technology. The chain structure of the block ensures the security of auditing data source. Taking the security and efficiency into account, a novel public auditing scheme for cloud data is proposed in this paper based on a self-certified public key system. The contributions of this paper are as follows. Firstly, recent related public auditing protocol are introduced. Secondly, we propose a public auditing protocol with a self-certified public key system using blockchain technology, in which the security and efficiency are taken into account. Finally, we conduct detailed theory analysis of the security and efficiency of the new scheme.

The outline of the paper is as follows: the research background and necessary preliminaries for the new public auditing system are firstly introduced. In the latter, the corresponding algorithm of the proposed scheme is described. Then, the security and efficiency of the new scheme are comprehensively analyzed from four aspects. Finally, a few concluding remarks are given in the last section.

2. System Model and Desired Objectives

In general, our public auditing scheme includes the following four entities: *CSP*, *TPA*, *user*, and blockchain. The system model is shown in Figure 1.

CSP, who has large-scale computing and storage resources, provides users with on-demand data storage services. CSP is considered as an untrustworthy party. For their own self-interest or maintaining their reputations, CSP may choose to conceal the data errors from the users. To reduce the amount of storage space and save bandwidth, CSP may deliberately delete some data that users rarely access. Furthermore, the CSP may launch some attacks on TPA. For example, CSP may try to forge some legitimate data blocks and their corresponding tags in order to pass verification phase.

TPA, who undertakes audit tasks for users, provides fair and objective audit results. TPA is supposed to be credible but curious. More concretely, TPA can perform auditing credibly in the verification phase, but it may be curious about the privacy information of users' data and even may try to derive the users' data contents.

User, who has large amounts of data, outsources the data to the cloud. Then, he/she can enjoy the reliability of data storage and high-performance services. The maintenance overhead can also be reduced. However, due to the loss of the management of outsourced data, users will have a strong desire to periodically check the integrity and correctness of those data.

We use *blockchain* to store user's operations on the file and metadata information of the uploaded file. The system does not care where files are stored but only stores a file URL in metadata file. We take advantage of the blockchain's tamper-resistant nature to ensure the reliability of operation logs and file metadata. Metadata information is used to audit the integrity of the data, and the analysis operation log can be used for behavioral audit.

Based on the above description of public auditing scheme model, the desired objectives to be achieved must be given for designing a secure and efficient public auditing scheme.

Public Auditing. Any authorized TPA is allowed to verify the correctness and integrity of user's data stored in the cloud.

Blockless Verification. During the verification process, TPA does not need to audit cloud data by retrieving the data blocks.

Storage Correctness. CSP, who does not store the intact data as required, cannot pass the audit.

Privacy Preserving. TPA cannot derive users' data contents from the collected auditing information during the verification phase.

Batch Auditing. TPA can efficiently deal with multiple audit tasks from different users. It not only reduces the number of communications between TPA and CSP during the auditing phase but also enhances the verification efficiency [17, 23].

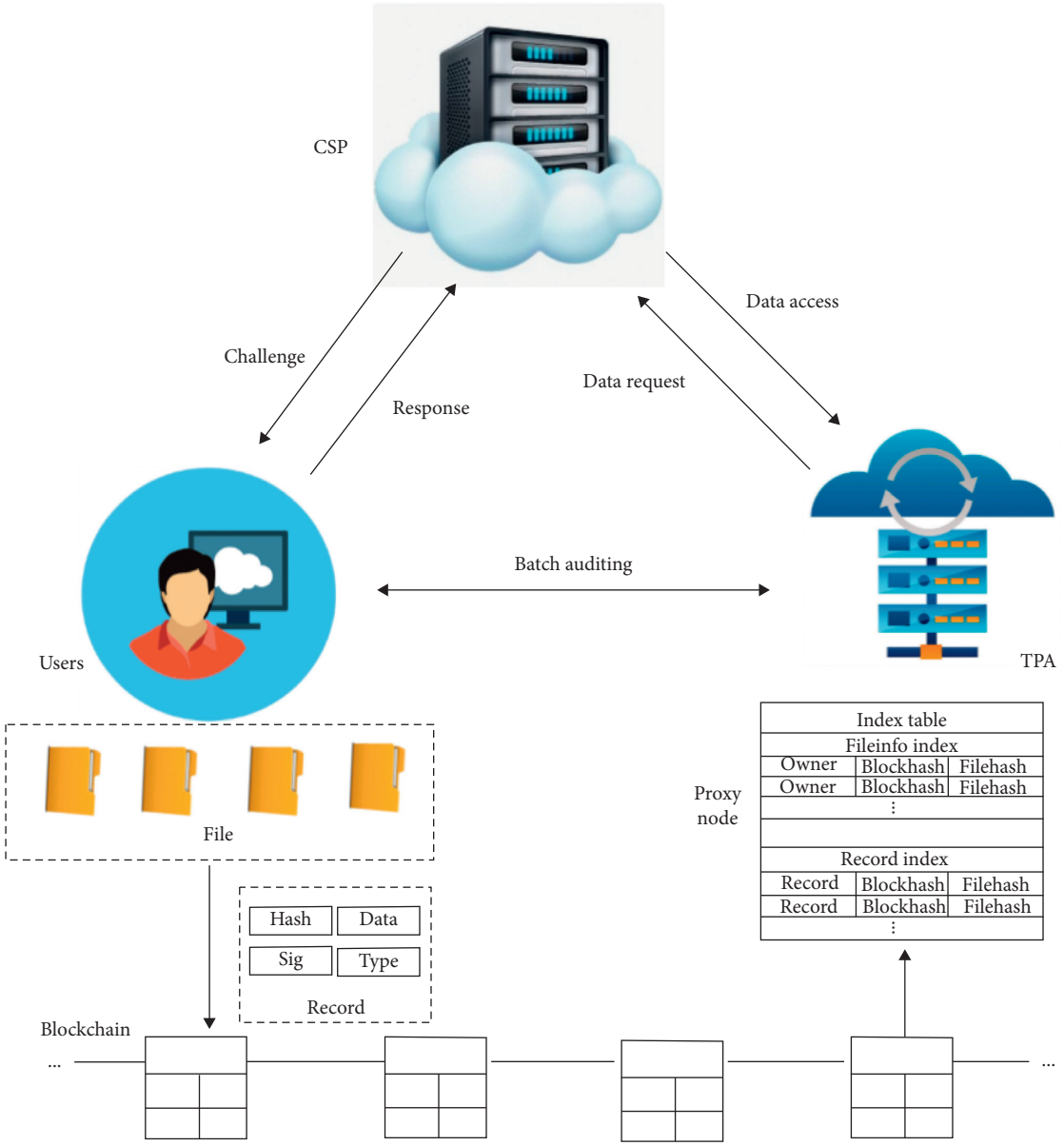


Figure 1: System model.

Lightweight. The public auditing scheme should have less communication overhead and lower computation cost.

3. Preliminaries

3.1. Self-Certified Public Key. The notion of self-certified public key (SCPK) was first introduced by Girault [38]. The user's public key is derived from the signature of the user's secret key with his/her identity in the SCPK system. The signature is signed by the system authority using the system's secret key. And the user's identity, public key, and secret key satisfy a computationally unforgettable mathematical relationship. While using the keys to perform encryption and decryption, signature verification, key agreement, or other cryptographic operations, the public key can be implicitly authenticated in the process of signature verification. In addition, each public key does not have a separate certificate and the verifier does not need to authenticate the certificate of the public key. Consequently, the SCPK system can reduce the storage space and computational overhead in public key schemes. Moreover, the user's private key is chosen by himself and the system authority who cannot get the private key from the transmitted data and cannot forge the signature as a user. Compared with ID-based public key system, the SCPK system has higher security and is more suitable for applications in open network environment.

3.2. Bilinear Map. Let G_1 and G_2 be two multiplicative cyclic groups of prime order p and g be a generator of G_1. A bilinear map is a map e: $G_1 \times G_1 \longrightarrow G_2$ with the following properties [39]:

Bilinearity: for all $u, v \in G_1$ and $a, b \in Z_p$, $e(u^a, v^b) = e(u, v)^{ab}$.

Computability: the map e is efficiently computable.

Nondegeneracy: $e(g, g) \neq 1$.

3.3. HVA. Homomorphic verifiable authenticator (HVA) is a basic component of public auditing [10, 19, 40]. Specifically, HVA can be generated based on digital signatures, such as RSA-based signature and BLS-based signature. Therefore, such HVAs can be considered as homomorphic verifiable signatures. Taking advantage of HVA, a public auditor can verify the integrity of outsourced data without downloading the original data. Generally speaking, HVA has the following properties [41, 42]:

Blockless Verifiability. Without knowing the actual data content, TPA can verify the integrity of the data blocks based on the proof constructed by HVAs.

Homomorphism. Let G_1 and G_2 be multiplicative groups, whose orders are a large prime p. Let "$\oplus$" and "$\otimes$" be operations in G_1 and G_2. If a map function $f: G_1 \longrightarrow G_2$ satisfies homomorphism, then $\forall g_1, g_2 \in G$, $f(g_1 \oplus g_2) = f(g_1) \otimes f(g_2)$.

Nonmalleability. Let σ_1 and σ_2 be signatures of block m_1 and block m_2, respectively. Given a certain block $m' = \alpha_1 m_1 + \alpha_2 m_2$, where α_1 and α_2 are two random numbers in $\mathbb{Z}_p^*$. For any user, if he/she does not know the private key, he/she cannot simply generate the legitimate signature σ' of block m' based on σ_1 and σ_2.

3.4. Merkle Tree. Merkle hash tree (MHT) is an authentication structure built based on hashes of data. The leaf node of Merkle tree stores the hashes of data elements (a file or a collection of files). The nonleaf node stores the hashes of its child nodes. MHT can identify whether the data were altered by comparing the calculated root hash with the value held by the validator. In blockchain network, MHT is used to store transaction's hash and check transaction's authenticity.

Figure 2 shows block structure in blockchain. Each block header saves the root hash of all transaction t_i in this block. The root hash participates in the hash operation of block header, and thus any modification to transaction data will lead to the change of the root hash, which will result in the hash change of the block header. In this paper, the user's operational information and metadata information of files are put into the blockchain. The chain structure of the block ensures the security of auditing data source.

3.5. Security Assumptions. The security of our new public auditing scheme will be based on the CDH assumption and DL assumption.

3.5.1. Computational Diffie–Hellman (CDH) Problem. Let G be a multiplicative cyclic group. The order of G is a large prime p. The generator of G is g. The CDH problem is described as follows: given two random numbers $a, b \in Z_p$ and $(g, g^a, g^b) \in G$, compute the value $g^{ab} \in G$.

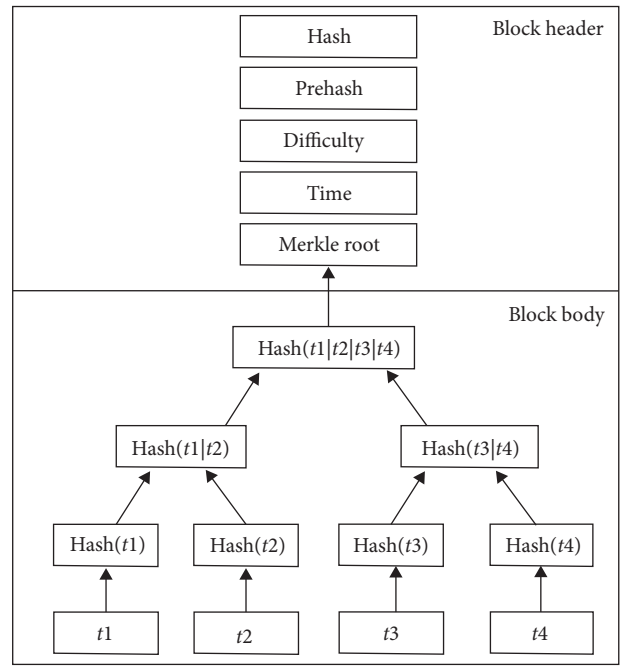


Figure 2: Block structure in blockchain.

Definition 1. **CDH assumption:** the probability that any probabilistic polynomial-time adversary $\mathcal{A}$ solves the CDH problem can be negligible, namely,

$$\Pr\left(A_{\mathrm{CDH}}\left(g, g^a, g^b \in G\right) \longrightarrow g^{ab} \in G: \forall a, b \underset{R}{\in} Z_p\right) \leq \varepsilon. \tag{1}$$

In other words, it is computationally feasible to solve the CDH problem or impossible to solve the CDH problem in a limited time.

3.5.2. Discrete Logarithm (DL) Problem. Let G be a multiplicative cyclic group. The order of G is a large prime p. The generator of G is g. The DL problem is described as follows: given $h \in G$, compute $a \in G$, such that $h = g^a$.

Definition 2. **DL assumption:** the probability that any probabilistic polynomial-time adversary $\mathcal{A}$ solves the DL problem can be negligible, namely,

$$\Pr\left(A_{\mathrm{DL}}(g, h \in G) \longrightarrow a \in Z_p, s.t.\ h = g^a\right) \leq \varepsilon. \tag{2}$$

In other words, it is computationally feasible to solve the DL problem or impossible to solve the DL problem in a limited time.

4. Public Auditing Scheme Based on SCPK

Then, we describe how to construct our public auditing scheme based on the SCPK system in more detail.

(1) *System Initialization Phase.* Let G_1 and G_2 be two groups of a large prime order p and g be a generator of

G_1. Let e be a bilinear map with e: $G_1 \times G_1 \longrightarrow G_2$. Let h be a hash function expressed as h: $\{0,1\}^* \longrightarrow G_1$. Suppose that the outsourced file F is divided into n data blocks, i.e., $F = \{m_1, m_2, \ldots, m_n\}$. Assume the identities of the user and file are ID_1 and ID_2, respectively.

(2) *Key Generation Phase*. In the system, TPA can be used as a trusted authority who is responsible for the user's registration and the generation of user's public key. TPA first publishes the modulus N and its public key pk. The private key of TPA is sk. The length of N is more than 1024 bits, and $\mathrm{pk} \times \mathrm{sk} = \varphi(N)$, where $\varphi(\cdot)$ is Euler's totient function. Then, the user selects a random number $a \in Z_p$ as his private key and calculates $v = g^a \mathrm{mod} N$. After that, the user sends the v and his identity ID_1 to TPA who will calculate user's public key $y = (v - \mathrm{ID}_1)^{h(\mathrm{ID}_1)^{-1}} \bmod N$ and send y to user. After receiving y, the user verifies the validity of equation $v = (y^{h(\mathrm{ID}_1)} + \mathrm{ID}_1) \mathrm{mod}\ N$. If the equation holds, then the running result of this stage is $\{\mathrm{SK}, \mathrm{PK}\} = \{(a), (,)\}$, where u is the random element of G_1.

(3) *Signature Generation Phase*. With the public parameter g and his private key a, the user generates a signature $\sigma_i = h(v_i t_i) \cdot (g^{h(m_i)})^a$ for each data block m_i. The mentioned v_i is m_i's version number, and t_i is m_i's time stamp. Then, let the signature set of all blocks be $\sigma = \{\sigma_i, i \in [1,n]\}$.

(4) *File Tag Generation Phase*. To ensure the integrity of the unique file identifier ID_2, the user computes the file tag $\vartheta = TQ\mathrm{ID}_1\mathrm{ID}_2\mathrm{SIG}\{\mathrm{sk}, TQ\mathrm{ID}_1\mathrm{ID}_2\}$ with his private key $\mathrm{sk} = a$. In the equation, $T = g^x \mathrm{mod}\ N$ and $Q = g^{x-a} \mathrm{mod}\ N$, where $x \in Z_p$ is a random number chosen by the user.

Finally, the user sends the data information $\mathrm{ID}_2, \{v_i, t_i, i \in [1,n]\}$ to the TPA for auditing and uploads $F, \sigma, \vartheta, \mathrm{ID}_1$ to the CSP for storage.

(5) *Block Tag Generation Phase*. After receiving F, σ, ϑ, CSP further generates a tag $\theta_i = e(\sigma_i, g)$ for each block m_i by using the bilinear map e. Then, CSP stores the verification metadata $\vartheta, \theta = \{\theta_i, i \in [1,n]\}$ along with the file $F = \{m_1, m_2, \ldots, m_n\}$.

(6) *File Identifier Check Phase*. The user delegates the verification task of a certain file to the TPA. Then, TPA requests the corresponding file tag ϑ from CSP and verifies the equation $Q(y^{h(\mathrm{ID}_1)} + \mathrm{ID}_1) \mathrm{mod}\ N = T$ with user's public key y. If the verification fails, TPA informs the user that the files have been corrupted; otherwise, verification continues.

(7) *Challenge Generation Phase*. TPA launches the verification challenge to the CSP in this stage. TPA first chooses a random number $k \in Z_p$ and calculates $K = g^k$, which is called random masking and is used to achieve privacy preserving [39]. Then, TPA sends the challenge information $\mathrm{chal} = \{\mathrm{idx}_i, r_i, K, i \in [1,c]\}$ to CSP, where idx_i is the index of the blocks to be checked, $r_i \in Z_P$ is the random number, and c is the selected number of the blocks to be checked [12].

(8) *Proof Generation Phase*. After receiving the challenge information, CSP would generate corresponding proofs of required blocks, which contain two parts: the tag proof and the data proof. More specifically, CSP generates the tag proof as follows:

$$T = \prod_{i \in [1,c]} \theta_i^{r_i}, \tag{3}$$

which can indicate the tags' correctness. And CSP generates the data proof as follows:

$$D = e(t, K)^M, \tag{4}$$

where $M = \sum_{i \in [1,c]} r_i \cdot h(m_i)$ and $t = y^{h(\mathrm{ID}_1)} + \mathrm{ID}_1$. The data proof can indicate the data's integrity. Then, CSP sends the proof $\{T, D\}$ to TPA.

(9) *Proof Verification Phase*. After receiving the proof, TPA would check whether the proof is valid. More concretely, TPA checks whether

$$D \cdot e\left(\prod_{i \in [1,c]} h(v_i | t_i)^{r_i}, K\right) \stackrel{?}{=} T^k, \tag{5}$$

holds. If the above verification equation holds, it shows that the outsourced data in the cloud are integral; otherwise, it shows that the data are incomplete.

The correctness of the above equation can be demonstrated as follows:

$$
\begin{aligned}
D \cdot e\left(\prod_{i\in[1,c]} h(v_i t_i)^{r_i}, K\right) &= e\left(t, g^k\right)^M \cdot e\left(\prod_{i\in[1,c]} h(v_i t_i)^{r_i}, K\right) \\
&= e\left(t^M, g^k\right) \cdot e\left(\prod_{i\in[1,c]} h(v_i t_i)^{r_i}, g^k\right) \\
&= e\left(g^{a\cdot M}, g^k\right) \cdot e\left(\prod_{i\in[1,c]} h(v_i t_i)^{r_i}, g^k\right) \\
&= e\left(g^{a\cdot \sum_{i\in[1,c]} r_i \cdot h(m_i)}, g^k\right) \cdot e\left(\prod_{i\in[1,c]} h(v_i|t_i)^{r_i}, g^k\right) \\
&= e\left(\prod_{i\in[1,c]} g^{h(m_i)\cdot a\cdot r_i}, g^k\right) \cdot e\left(\prod_{i\in[1,c]} h(v_i t_i)^{r_i}, g^k\right) \\
&= e\left(\prod_{i\in[1,c]} h(v_i t_i)^{r_i} \cdot g^{h(m_i)\cdot a\cdot r_i}, g^k\right) \\
&= \prod_{i\in[1,c]} e\left(h(v_i t_i) \cdot g^{h(m_i)\cdot a}, g\right)^{r_i k} \\
&= \prod_{i\in[1,c]} \theta_i^{r_i k} \\
&= T^k.
\end{aligned}
\tag{6}
$$

5. Security Proof and Performance Analysis

In the proposed public auditing scheme, CSP is assumed to be an untrustworthy party and TPA is considered credible but curious. CSP may conceal the data errors or deliberately delete some data. TPA may be curious about the privacy information of users' data and even may try to derive the users' data contents. Then, necessary security and performance analyses of the new scheme will be comprehensively demonstrated in this section.

First of all, let us analyze the security of the self-certified public key system.

If an attacker attempts to retrieve the user's secret key a from his/her public key y, he/she must calculate the secret key from the equation $g^a = (y^{h(\mathrm{ID}_1)} + \mathrm{ID}_1) \bmod N$. In this way, he/she will face the difficulty of computing discrete logarithm modulo N. In other words, the attacker's probability of success is to solve the discrete logarithm problem and factorization problem. Moreover, even TPA knows v and ID_1; the difficulty for him to retrieve the user's secret key a is also equivalent to the difficulty of computing discrete logarithm.

Another scenario is that an attacker tries to derive user's secret key a from the user's signature. For the file tag $\vartheta = TQ\mathrm{ID}_1\mathrm{ID}_2\mathrm{SIG}\{\mathrm{sk}, TQ\mathrm{ID}_1\mathrm{ID}_2\}$, the attacker should obtain x from $T = g^x \bmod N$ or $Q = g^{x-a} \bmod N$. However, the difficulty for him to achieve it is also equivalent to the difficulty of computing discrete logarithm problem. For the block signature $\sigma_i = h(v_i t_i) \cdot (g^{h(m_i)})^a$, the attacker should compute a from the equation. He also faces the difficulty of computing discrete logarithm problem.

The final scenario is that an attacker tries to impersonate the signer to forge a valid signature without knowing the signer's secret key a. For the file tag, the above analysis shows that the attacker cannot reveal the user's secret key. Then, he cannot forge a valid signature that can pass the verification. For the block signature, Definition 2 indicates that the probability that any probabilistic polynomial-time adversary $\mathcal{A}$ solves the DL problem can be negligible. Then, it is computationally infeasible for the attacker to forge a valid HVA in a limited time. The proof, which is demonstrated in the security analysis of [12], is omitted in this paper.

Secondly, we discuss the unforgeability of proofs.

In the presented public auditing scheme, CSP sends the proof $\{T, D\}$ to TPA after the proof generation phase. The above analysis shows that the tag proof T cannot be forged owing to the CDH assumption. Then, we only need to prove that the data proof cannot be forged. Suppose CSP sends a fake proof $\{T, D^*\}$ to TPA, where $D = e(t, K)^{M^*}$ and $M^* = \sum_{i\in[1,c]} r_i \cdot h(m_i^*)$. If CSP wants to pass the verification, the equation

$$e\left(g^{a\cdot\sum_{i\in[1,c]} r_i\cdot h(m_i)}\cdot\prod_{i\in[1,c]} h(v_i t_i)^{r_i}, g^k\right) = e\left(g^{a\cdot\sum_{i\in[1,c]} r_i\cdot h(m_i^*)}\cdot\prod_{i\in[1,c]} h(v_i t_i)^{r_i}, g^k\right), \tag{7}$$

must hold. Then, we can deduce that $\sum_{i\in[1,c]} r_i \cdot h(m_i^*) = \sum_{i\in[1,c]} r_i \cdot h(m_i)$ according to the properties of bilinear maps. However, this contradicts the above assumption. That is to say, the data proof is unforgeable. In summary, our presented scheme can effectively resist against the forging attacks launched by CSP.

Thirdly, we discuss the communication and computation overhead, which are reduced by introducing the batch auditing.

With the batch auditing, multiple verification tasks from different users can be handled concurrently. Suppose that TPA sends d challenges to CSP. Then, the tag proof T_j and the data proof D_j are calculated separately. And CSP figures out the aggregate proofs according to the following equation:

$$\begin{aligned} T_B &= \prod_{j=1}^{d} T_j, \\ D_B &= \prod_{j=1}^{d} D_j, \end{aligned} \tag{8}$$

where $T_j = \prod_{i\in[1,c]} \theta_{ji}^{r_{ji}}$, $D_j = e(t_j, K_j)^{M_j}$, $t_j = y^{h(\mathrm{ID}_j)} + \mathrm{ID}_j$, and $M_j = \sum_{i\in[1,c]} r_{ji} \cdot h(m_{ji})$. ID_j is the identity of the j-th user. Then, CSP sends the aggregate proofs $\{T_B, D_B\}$ to TPA. Once received, TPA checks whether the equation

$$D_B \cdot \prod_{j=1}^{d} e\left(\prod_{i\in[1,c]} h\left(v_{ji}||t_{ji}\right)^{r_{ji}}, K_j\right) \stackrel{?}{=} T_B^{k_j}, \tag{9}$$

holds. v_{ji} and t_{ji} are the version number and time stamp of block m_i for the j-th user. k_j is the random number chosen by TPA for the j-th user. $K_j = g^{k_j}$ is the random masking calculated by TPA for the j-th user. $r_{ji} \in Z_P$ is the random number chosen by TPA for the j-th user.

If the above verification equation holds, it shows that our scheme can realize the batch auditing. Then, its correctness can be demonstrated as follows:

$$\begin{aligned} D_B \cdot \prod_{j=1}^{d} e\left(\prod_{i\in[1,c]} h\left(v_{ji}t_{ji}\right)^{r_{ji}}, K_j\right) &= \prod_{j=1}^{d} e\left(t_j, K_j\right)^{M_j} \cdot \prod_{j=1}^{d} e\left(\prod_{i\in[1,c]} h\left(v_{ji}t_{ji}\right)^{r_{ji}}, K_j\right) \\ &= \prod_{j=1}^{d} e\left(t_j^{M_j}, g^{k_j}\right) \cdot \prod_{j=1}^{d} e\left(\prod_{i\in[1,c]} h\left(v_{ji}t_{ji}\right)^{r_{ji}}, g^{k_j}\right) \\ &= \prod_{j=1}^{d} e\left(g^{a_j\cdot M_j}, g^{k_j}\right) \cdot \prod_{j=1}^{d} e\left(\prod_{i\in[1,c]} h\left(v_{ji}t_{ji}\right)^{r_{ji}}, g^{k_j}\right) \\ &= \prod_{j=1}^{d} e\left(g^{a_j\cdot\sum_{i\in[1,c]} r_{ji}\cdot h(m_{ji})}, g^{k_j}\right) \cdot \prod_{j=1}^{d} e\left(\prod_{i\in[1,c]} h\left(v_{ji}t_{ji}\right)^{r_{ji}}, g^{k_j}\right) \\ &= \prod_{j=1}^{d} e\left(\prod_{i\in[1,c]} g^{h(m_{ji})\cdot a_j\cdot r_{ji}}, g^{k_j}\right) \cdot \prod_{j=1}^{d} e\left(\prod_{i\in[1,c]} h\left(v_{ji}t_{ji}\right)^{r_{ji}}, g^{k_j}\right) \\ &= \prod_{j=1}^{d} e\left(\prod_{i\in[1,c]} h\left(v_{ji}t_{ji}\right)^{r_{ji}} \cdot g^{h(m_{ji})\cdot a_j\cdot r_{ji}}, g^{k_j}\right) \\ &= \prod_{j=1}^{d} \prod_{i\in[1,c]} e\left(h\left(v_{ij}t_{ji}\right) \cdot g^{h(m_{ji})\cdot a_j}, g\right)^{r_{ji}k_j} \\ &= \prod_{j=1}^{d} \prod_{i\in[1,c]} \theta_{ji}^{r_{ji}k_j} \\ &= \prod_{j=1}^{d} T_j^{k_j} \\ &= T_B^{k_j}. \end{aligned} \tag{10}$$

Finally, our new scheme is based on the self-certified public key system. Compared with other public auditing schemes [10, 12, 18, 19, 21–28, 38, 39], there is no public key certificate included in the public authentication parameters. And there is no need to store and transmit the public key certificate before the interaction of auditing. Then, the validation and validity of public key certificate is omitted. The verification of public key is hidden in the process of the verification of signature. Consequently, the storage space and communication bandwidth are saved. The network load and transmission delay are reduced. The verification efficiency of public and the authentication efficiency of the scheme are improved.

6. Discussion and Conclusions

In this paper, we present a public auditing protocol with a self-certified public key system using blockchain technology, which differs from the state-of-the-art schemes. The user's operational information and metadata information of the file are formed to a block after verified by the checked nodes and then to be put into the blockchain. The chain structure of the block ensures the security of auditing data source. Comprehensive analyses show that attackers cannot derive user's secret key in the proposed scheme. TPA cannot derive users' data from the collected auditing information during the verification phase. Attackers cannot impersonate the signer to forge a valid signature without knowing the signer's secret key. The presented scheme can also effectively resist against the forging attacks launched by CSP. The realization of batch auditing and the efficiency of the scheme are also discussed in this paper. Compared with other public auditing schemes, the storage space and communication bandwidth are saved in our public auditing scheme. The network load is also reduced. In addition, the verification efficiency of public key and the authentication efficiency of the scheme are improved.

However, in the actual cloud storage environment, a lot of various data need to be updated dynamically motivated by various application requirements. For instance, users might try to perform insertion operation owing to the incomplete outsourced data or might try to delete some data that are no longer used. Our public auditing scheme does not specifically discuss dynamic data auditing, which can be referred to DHT [19] or put forward as a new structure in our future research. Furthermore, TPA may dishonestly perform public auditing protocols and may even collude with CS to deceive users. Some existing public audit schemes use blockchain to resist against malicious TPA. However, CS may guess the challenge messages, and there is a risk that user information may be disclosed to TPA during the audit process. The above questions will be the focus of our future research.

Data Availability

All data, models, and codes generated or used during the study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was supported by the National Natural Science Foundation of China under grant no. 61702316, the Natural Science Foundation of Shanxi Province under grant nos. 201801D221177 and 201901D111280, the Educational Research Projects of Young and Middle-Aged Teachers in Fujian Education Department under grant no. JAT170142, the Key Research and Development Project of Shandong Province under grant no. 2019JZZY010134, and the Graduate Education Reform Research Project of Shanxi Province under grant no. 2020YJJG145.

References

[1] H. Wang, Z. Zheng, L. Wu et al., "New directly revocable attribute-based encryption scheme and its application in cloud storage environment," *Cluster Computing*, vol. 20, no. 3, pp. 2385–2392, 2017.

[2] M. N. O. Sadiku, S. M. Musa, and O. D. Momoh, "Cloud computing: opportunities and challenges," *IEEE Potentials*, vol. 33, no. 1, pp. 34–36, 2014.

[3] K. Yang and X. Jia, "Data storage auditing service in cloud computing: challenges, methods and opportunities," *World Wide Web*, vol. 15, no. 4, pp. 409–428, 2012.

[4] Z. Xia, X. Wang, X. Sun et al., "A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 2, pp. 340–352, 2016.

[5] K. Ren, C. Wang, and Q. Wang, "Security challenges for the public cloud," *IEEE Internet Computing*, vol. 16, no. 1, pp. 69–73, 2012.

[6] J. Shen, T. Zhou, X. Chen et al., "Anonymous and traceable group data sharing in cloud computing," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 4, pp. 912–925, 2018.

[7] J. Ryoo, S. Rizvi, W. Aiken et al., "Cloud security auditing: challenges and emerging approaches," *IEEE Security & Privacy*, vol. 12, no. 6, pp. 68–74, 2014.

[8] Q. Jiang, J. Ma, and F. Wei, "On the security of a privacy-aware authentication scheme for distributed mobile cloud computing services," *IEEE Systems Journal*, vol. 12, no. 2, pp. 2039–2042, 2018.

[9] Z. Fu, X. Wu, C. Guan et al., "Toward efficient multi-keyword fuzzy search over encrypted outsourced data with accuracy improvement," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 12, pp. 2706–2716, 2016.

[10] C. Wang, S. S. M. Chow, Q. Wang et al., "Privacy-preserving public auditing for secure cloud storage," *IEEE Transactions on Computers*, vol. 62, no. 2, pp. 362–375, 2013.

[11] C. Wang, K. Ren, W. Lou et al., "Toward publicly auditable secure cloud data storage services," *IEEE Network*, vol. 24, no. 4, pp. 19–24, 2010.

[12] Q. Wang, C. Wang, K. Ren et al., "Enabling public auditability and data dynamics for storage security in cloud computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 5, pp. 847–859, 2011.

[13] J. Li, Y. K. Li, X. Chen et al., "A hybrid cloud approach for secure authorized deduplication," *IEEE Transactions on*

Parallel and Distributed Systems, vol. 26, no. 5, pp. 1206–1216, 2015.

[14] M. Mowbray, "The fog over the grimpen mire: cloud computing and the law," *Scripted*, vol. 6, p. 132, 2009.

[15] A. Juels and B. S. Kaliski Jr, "PORs: proofs of retrievability for large files," in *Proceedings of the 14th ACM conference on Computer and communications security*, pp. 584–597, Acm, New Delhi, India, March 2007.

[16] G. Ateniese, R. Burns, R. Curtmola et al., "Provable data possession at untrusted stores," in *Proceedings of the 14th ACM conference on Computer and communications security*, pp. 598–609, Acm, New Delhi, India, March 2007.

[17] K. Yang and X. Jia, "An efficient and secure dynamic auditing protocol for data storage in cloud computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 9, pp. 1717–1726, 2013.

[18] Y. Zhu, G. J. Ahn, H. Hu et al., "Dynamic audit services for outsourced storages in clouds," *IEEE Transactions on Services Computing*, vol. 6, no. 2, pp. 227–238, 2013.

[19] H. Tian, Y. Chen, C. C. Chang et al., "Dynamic-hash-table based public auditing for secure cloud storage," *IEEE Transactions on Services Computing*, vol. 10, no. 5, pp. 701–714, 2017.

[20] H. Jin, H. Jiang, and K. Zhou, "Dynamic and public auditing with fair arbitration for cloud data," *IEEE Transactions on Cloud Computing*, vol. 6, no. 3, pp. 680–693, 2018.

[21] J. Shen, J. Shen, X. Chen et al., "An efficient public auditing protocol with novel dynamic structure for cloud data," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 10, pp. 2402–2415, 2017.

[22] C. Wang, Q. Wang, K. Ren et al., "Privacy-preserving public auditing for data storage security in cloud computing," in *Proceeding of the Infocom, 2010 proceedings IEEE*, pp. 1–9, San Diego, CA, USA, March 2010.

[23] Y. Zhu, H. Hu, G. J. Ahn et al., "Cooperative provable data possession for integrity verification in multicloud storage," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 12, pp. 2231–2244, 2012.

[24] C. C. Erway, A. Küpçü, C. Papamanthou et al., "Dynamic provable data possession," *ACM Transactions on Information and System Security (TISSEC)*, vol. 17, no. 4, p. 15, 2015.

[25] F. Sebé, J. Domingo-Ferrer, A. Martinez-Balleste et al., "Efficient remote data possession checking in critical information infrastructures," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 8, pp. 1034–1038, 2008.

[26] H. Shacham and B. Waters, *Compact Proofs of retrievability*, pp. 90–107, Springer, Berlin, Germany, 2008.

[27] W. Chen, H. Tian, C. C. Chang et al., "Adjacency-hash-table based public auditing for data integrity in mobile cloud computing," *Wireless Communications and Mobile Computing*, vol. 2018, 2018.

[28] S. G. Worku, C. Xu, J. Zhao et al., "Secure and efficient privacy-preserving public auditing scheme for cloud storage," *Computers & Electrical Engineering*, vol. 40, no. 5, pp. 1703–1713, 2014.

[29] R. Ding, Y. Xu, J. Cui et al., "A public auditing protocol for cloud storage system with intrusion-resilience," *IEEE Systems Journal*, vol. 2019, no. 99, pp. 1–12, 2019.

[30] N. Garg, S. Bawa, and N. Kumar, "An efficient data integrity auditing protocol for cloud computing," *Future Generation Computer Systems*, vol. 109, pp. 306–316, 2020.

[31] C. Li, J. Hu, K. Zhou, Y. Wang, and H. Deng, "Using blockchain for data auditing in cloud storage," in *Cloud Computing and Security. ICCCS 2018. Lecture Notes in Computer Science*, X. Sun, Z. Pan, and E. Bertino, Eds., Springer, Berlin, Germany.

[32] L. Linn and M. Koo, "Blockchain for health data and its potential use in health it and health care related research," *Journal of Medical Internet Research*, vol. 2016, 2016.

[33] A. Fu, S. Yu, Y. Zhang, H. Wang, and C. Huang, "Npp: a new privacyaware public auditing scheme for cloud data sharing with group users," *IEEE Transactions on Big Data*, vol. 99, 2017.

[34] S. Ghoshal and G. Paul, "Exploiting block-chain data structure for auditorless auditing on cloud data," in *ICISS 2016*, I. Ray, M. S. Gaur, M. Conti, D. Sanghi, and V. Kamakoti, Eds., pp. 359–371, Springer, Berlin, Germany, 2016.

[35] Y. Fu, "Meta-key: a secure data-sharing protocol under blockchain-based decentralised storage architecture," 2017, http://arxiv.org/abs/1710.07898.

[36] Y. Miao, Q. Huang, M. Xiao et al., "Decentralized and privacy-preserving public auditing for cloud storage based on blockchain," *IEEE Access*, vol. 8, p. 1, 2020.

[37] S. Li, J. Liu, G. Yang et al., "A blockchain-based public auditing scheme for cloud storage environment without trusted auditors," *Wireless Communications and Mobile Computing*, vol. 2020, no. 9, pp. 1–13, 2020.

[38] M. Girault, *Self-certified Public keys*, pp. 490–497, Springer, Berlin, Germany, 1991.

[39] C. Liu, R. Ranjan, X. Zhang et al., "Public auditing for big data storage in cloud computing--A survey," in *Computational science and engineering (CSE), 2013 IEEE 16th international conference on. IEEE*, pp. 1128–1135, Sydney, Australia, December 2013.

[40] B. Wang, B. Li, H. Li et al., "Enabling public auditing for shared data in cloud storage supporting identity privacy and traceability," *Journal of Systems and Software*, vol. 113, pp. 130–139, 2016.

[41] Q. Lin, H. Yan, Z. Huang et al., "An ID-based linearly homomorphic signature scheme and its application in blockchain," *IEEE Access*, vol. 6, pp. 20632–20640, 2018.

[42] Panda, "Public auditing for shared data with efficient user revocation in the cloud," *IEEE Transactions on Services Computing*, vol. 8, no. 1, pp. 92–106, 2015.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 6619689, 11 pages
https://doi.org/10.1155/2021/6619689

Research Article

Blockchain-Enabled Public Key Encryption with Multi-Keyword Search in Cloud Computing

Zhenwei Chen,[1] **Axin Wu**,[2] **Yifei Li**,[1] **Qixuan Xing**,[1] **and Shengling Geng**[3,4]

[1]*School of Cyberspace Security, Xi'an University of Posts and Telecommunications, Xi'an 710121, China*
[2]*College of Cybersecurity, Jinan University, Guangzhou 510632, China*
[3]*School of Computer, Qinghai Normal University, Xining, Qinghai 810008, China*
[4]*Institute of Plateau Science and Sustainable Development, Xining, Qinghai, China*

Correspondence should be addressed to Shengling Geng; geng_sl@126.com

Received 3 November 2020; Revised 18 December 2020; Accepted 2 January 2021; Published 21 January 2021

Academic Editor: Qi Li

The emergence of the cloud storage has brought great convenience to people's life. Many individuals and enterprises have delivered a large amount of data to the third-party server for storage. Thus, the privacy protection of data retrieved by the user needs to be guaranteed. Searchable encryption technology for the cloud environment is adopted to ensure that the user information is secure with retrieving data. However, most schemes only support single-keyword search and do not support file updates, which limit the flexibility of the scheme. To eliminate these problems, we propose a blockchain-enabled public key encryption scheme with multi-keyword search (BPKEMS), and our scheme supports file updates. In addition, smart contract is used to ensure the fairness of transactions between data owner and user without introducing a third party. At the data storage stage, our scheme realizes the verifiability by numbering the files, which ensures that the ciphertext received by the user is complete. In terms of security and performance, our scheme is secure against inside keyword guessing attacks (KGAs) and has better computation overhead than other related schemes.

1. Introduction

Cloud storage is a removable storage method that brings great convenience to people. Therefore, the problem of data security is increasingly important. Generally speaking, cloud storage has three structures. First, public cloud storage service provides a wealth of resources, such as network services and storage, and users can access these resources through the Internet at low prices. Second, internal cloud storage is located inside the corporate firewall, and users have independent storage control rights. Third, hybrid cloud storage provides both public cloud services and internal cloud services. The core is to meet the visits required by customers. While eliminating the user's local storage hardware and management overhead, the data are out of the user's physical control, so data security is greatly threatened. When users upload data to cloud storage media, they need to solve the security problem of the data, and people often upload it after encryption. Secure search usually refers to the effective search of encrypted data; to solve the problem of how to use the server to complete the secure keyword search when the encrypted data are stored in the cloud under the premise of incomplete trust, scholars proposed the searchable encryption (SE) as the core technology of secure search.

SE is a new technology that supports users to search for keywords in ciphertexts. It mainly solves how to use untrusted servers to implement secure keyword search in a cloud storage environment so that users can securely search data in ciphertext state, specifically, search the keywords according to the keywords of interest. SE systems are divided into symmetric [1] and asymmetric [2–4] forms. Although the calculation amount of public key SE is greater than that of symmetric SE, data owners and users do not need to pass the key negotiation before searching, which is more secure and has greater practical value.

In terms of the usability of SE scheme, multi-keyword search [4, 5] is more in line with the user's search experience. Compared with single-keyword search, it can locate the search more accurately. In the actual scenario, the server may be honest but curious and will want to obtain some sensitive information. Therefore, it is very important to verify the correctness of the results [6]. However, this scheme is static and cannot operate data dynamically. Although some SE schemes [7, 8] support dynamic update of files and verifiability of ciphertext, they will bring a lot of computational overhead. Therefore, the practical SE scheme needs to be designed and proposed.

In this paper, we propose the BPKEMS scheme in the blockchain scenario; the main contributions are as follows.

(1) *Multi-Keyword Search*. The BPKEMS scheme has some good features, such as multi-keyword search and file updates. In addition, the data owner and data user can generate a shared key when encrypting files. By using the Diffie–Hellman (DH) key exchange protocol, they can get the shared key without any interaction.

(2) *Fairness*. In this scheme, the blockchain mechanism is used to ensure the fairness of the transaction between data owner and user without a third party.

3*Verifiability*. On the blockchain platform, we use smart contract to store index and trapdoor information and perform search services to ensure the accuracy of search results. In addition, we number the files, and the user can verify the ciphertext of the file after receiving the result, which can avoid some malicious behavior of the cloud server.

2. Related Work

In recent years, cloud computing technology has been rapidly developed, and a series of studies have been done on security issues. In order to enhance the security of data on the server, Dawn et al. [1] first proposed a symmetric SE scheme, but it was in one-to-one mode, which has triggered people's research on SE because the one-to-one mode cannot meet people's needs. For the many-to-one model, Boneh et al. [2] first proposed the public key SE scheme and gave the concept of SE security based on public key encryption in 2004. But in certain environments, the many-to-one mode is not practical. In 2011, Curtmola et al. [9] constructed a one-to-many SE model based on Naor broadcast encryption technology [10], but the user's key replacement in this model requires a great deal of overhead. In a large-scale network environment, data transmission is complicated. Wang et al. [11] constructed a many-to-many mode encryption scheme based on Shamir's secret sharing technology [12] and the identity-based encryption technology in [2] to realize the interaction retrieval of multiple users in the server. In order to effectively solve the problem of interactive retrieval when there are multiple recipients, Yuan et al. [13] proposed a one-to-many public key ciphertext time release searchable encryption cryptographic model. In the one-to-many model, only authenticated users can enjoy the search service, and the queried keywords are specified, and they can decrypt it when it knows that it will be released in the future. Zhong et al. [14] proposed a many-to-one homomorphic encryption scheme, which overcomes the limitations of traditional one-to-one mode.

In terms of the security of SE, about the scheme [2] proposed by Boneh, only the semantic security of index ciphertext can be achieved, but it cannot resist KGAs. In 2009, Tang and Chen [15] put forward a public key SE scheme. The keywords should be registered before using, which can resist KGAs, but the keywords must be registered in advance, which makes the performance of the scheme not high. In 2013, Fang et al. [16] presented the scheme belonging to public key cryptography, which can resist KGAs; the scheme defines a public key SE model and two important security concepts: one is for inside attacks and the other is for external attacks. However, a large number of bilinear pairing calculations result in a low efficiency of Fang's scheme [16].

In recent years, scholars have conducted a lot of research on inside attacks. In 2013, Xu et al. [17] proposed a scheme with two trapdoors (fuzzy trapdoor and precision trapdoor) and claimed that the scheme can resist inside KGAs. In this scheme, the adversary intelligently obtains the fuzzy trapdoor, but some keyword information about the trapdoor is not known, and it is restricted in terms of security and efficiency. In 2015, Chen et al. [18] introduced a new framework to prevent inside KGAs. They used two servers to realize the scheme, but the limitation is that the two servers cannot be associated. However, anyone can generate legal trapdoors for keywords, which will make data privacy issues easy to discover. Shao et al. proposed a method [19] that can resist KGAs. In the SE scheme of a designated tester, the security of the scheme is redefined as IND-KGA-SERVER. In the presence of a digital signature, it can resist the server's KGAs. In 2016, Chen et al. [20] proposed a scheme using two servers to resist inside KGAs, and the scheme has high efficiency. However, due to the two assumptions that two cloud servers cannot be connected, this is difficult to achieve in practice. In 2017, Huang and Li [21] proposed a public key authentication encryption scheme based on keyword search. The ciphertext generation process of this scheme requires the key of the data owner. Although the scheme can resist the inside KGAs, it cannot achieve the chosen keyword ciphertext indistinguishability. Kang and Liu [22] proposed a completely secure public key encryption scheme composed of bilinear pairing and TF/IDF algorithm. This scheme achieves security under static assumptions. By comparing with previous SE schemes, their scheme's performance is superior to other schemes. In terms of security, this scheme can avoid revealing privacy due to the curiosity of the adversary. In 2018, Wu et al. [23] proposed an efficient and secure public key SE scheme with privacy protection. This scheme uses a DH shared key and is proven to resist KGAs.

In the Internet of Things (IoT) environment, Wu et al. [24] proposed a certificateless searchable public key authentication encryption scheme, which can resist KGAs at the same time and also has a higher efficiency. Ma et al. [25]

designed a new multi-keyword certificateless public key encryption scheme for IoT deployment. Lu and Li [26] proposed a new PEKS scheme, which not only can resist the existing three types of KGAs but also improves the shortcomings of the designated server. With the development of blockchain [27, 28], the combination of searchable encryption technology and blockchain technology solves the problem of trusted third party in traditional schemes and greatly improves the practicability of searchable encryption. Li et al. [30] proposed a searchable encryption system model of blockchain and designed a practical scheme for the system model. In 2019, Li et al. put forward a scheme [31] based on [30], which also improved enablement. In order to be suitable for the electronic medical scene, Chen et al. [32] proposed a SE scheme suitable for this scene under the blockchain technology. This scheme also adopts symmetric encryption method and uses smart contract as the authoritative entity to ensure the credibility of the server in the scheme. Zheng et al. [6] proposed an SE scheme which can verify the correctness of the results, but it cannot support data update operation. The SE scheme proposed by Sun et al. [7] and Xia et al. [8] can not only support dynamic update but also verify the results, and it also has low computational efficiency. Therefore, we are committed to solving these problems.

3. Preliminaries

In this section, we review the relevant background materials required in understanding our scheme and introduce some notations in Table 1.

Table 1: Main symbols.

Symbol	Meaning
$\mathrm{pk}_o, \mathrm{sk}_o$	Public/secret key pair of data owner
$\mathrm{pk}_u, \mathrm{sk}_u$	Public/secret key pair of data user
$\mathrm{pk}_s, \mathrm{sk}_s$	Public/secret key pair of cloud server
K	Shared key for data owner and data user
N_i	Encrypted file index
Ns	Encrypted file index set
$F = \{f_i\}_{i\in[1,t]}$	File index set
F'	Returned file set
$C = \{C_i\}_{i\in[1,t]}$	Ciphertext set of F
$C' = \{C_i'\}_{i\in[1,t]}$	Packed ciphertext
$W = \{w_i\}_{i\in[1,m]}$	Keyword dictionary
$I = \{I_0, I_1, I_2, I_j\}_{j\in[1,m]}$	Index set for F
σ_2'	The intermediate value
σ_2	The final value
$W' = \{w_i'\}_{i\in[1,l]}$	Queried keyword set
$T_{W'} = \{T_{W,1}', T_{W,2}'\}$	Trapdoor for W'
$L = \{\rho_1(\tau)\}_{\tau\in[1,l]}$	Location set of W' in W

3.1. Bilinear Pairing. Let G_1, G_2 be two multiplicative cycle groups. A map e: $G_1 \times G_1 \longrightarrow G_2$ is called a symmetric bilinear pairing if it has the following properties:

(1) *Bilinear.* $e(u^a, v^b) = e(u, v)^{ab}$, $\forall u, v \in G_1$, and $\forall a, b \in Z_p$.

(2) *Nondegenerate.* $e(g, g) \neq 1$. Let $1 \in G_2$ be the identity element of G_2 group.

(3) *Computable.* $\forall u, v \in G_1$, $e(u, v)$; there is a polynomial time algorithm that can easily calculate e.

3.2. Decisional Diffie–Hellman (DDH) Problem. Given a generator g of G_1, then $\{g, g^a, g^b, g^c\} \in G_1$, where $a, b, c \in Z_p$. The DDH problem is to determine whether g^c is equal to $g\{ab\}$. Assuming that the DDH problem is difficult, it means that no adversary can solve the problem with a probability that cannot be ignored.

3.3. Blockchain. In this section, let us briefly describe the smart contract, gas system, system model, threat model, and security model.

3.3.1. Smart Contract. Blockchain is important and has a wide range of applications, such as the Internet of Things and edge computing, and blockchain can be used in 5G handover authentication [33]. The smart contract (SC) is considered as the core technology of the second-generation blockchain, which was proposed by Szabo [34]. The carrier of the SC is the blockchain, and its essence is an automatically executed computer code. The code describes the terms of the agreement between the buyer and the seller and is directly written into the code of the blockchain. Satisfying the predetermined terms is the trigger condition for the code to be executed. Since the execution of the code does not require human intervention, it is called automatic execution.

As a computer program, a SC is a part of application software and a digitally represented program [35]. Although it is a code representation of contract terms, it is not a contract in the legal sense. In addition, the construction of SC comes from the blockchain framework, which is a public billing system, which can carry out secure value transfer without a trusted third party, and the correctness of the contract code execution is guaranteed by the consensus mechanism. Therefore, SC can be understood as a computer protocol, which can be executed automatically without human intervention.

3.3.2. Gas System. In Ethereum, once the SC is set, it is forbidden to modify it. In order to prevent malicious users from setting up an infinite loop running contract, Ethereum requires users to pay for each step of the deployment contract. The basic unit of cost is gas. Gas is equivalent to the fuel needed to deploy and execute SC. Without fuel, SC cannot be used. This mechanism maintains the operation of the economic system of Ethereum.

In a gas system, there are some important parameters. Gas price means that users need to pay for each unit of gas. Each block has a gas limit, that is, the maximum amount of gas allowed in a single block, which can be used to determine how many transactions can be packaged in a single block. Both gas price and gas limit are set by the transaction sender itself. If the total amount of gas consumed by the operation

exceeds gas limit, the operation will be voided, the transaction is not packaged in the block and the transaction amount is refunded, and the gas fee that has been performed will still be charged [36]. Only if the user's current amount is greater than gas limit times gas price, the transaction will be executed successfully. For gas price, if the value is too high, the transaction may be executed first, and if it is too low, the transaction speed will be slow.

3.4. System Model. In this section, we introduce the system model of the scheme, as shown in Figure 1.

(1) *Data Owner* (*DO*). The main work of data owner is to calculate the keyword index and the file ciphertext and then send the file ciphertext to the cloud server and the keyword index to the smart contract.

(2) *Data User* (*DU*). The main work is to calculate the trapdoor and upload it to the smart contract. Then, data user gets the corresponding file ciphertext from cloud server and verifies it. Finally, data user decrypts the file ciphertext.

(3) *Cloud Server* (*CS*). The main work of the cloud server is to store the data uploaded by the data owner and receive the file index from smart contract. Next, the cloud server forwards the corresponding file ciphertext to the data user.

(4) *Smart Contract* (*SC*). Smart contract's main job is to receive indexes and trapdoors to match and then send the search result to the cloud server through a transaction.

(5) *Trusted Authority* (*TA*). The trusted authority is responsible for generating public/private key pairs for data owner, data user, and the cloud server.

3.5. Algorithms in System Model. Here, we introduce the six algorithms in our scheme: Setup, KeyGen, Enc, Trap, Search, and Verification and Decryption.

(1) *Setup* (1^{λ}). The algorithm inputs a public parameter λ and outputs a global public parameter SP.

(2) *KeyGen* (SP). This algorithm takes SP as inputs, and it outputs the DO's public key pk_o and private key sk_o. The public and private keys of DU and CS are generated in a manner similar to DO.

(3) *Enc* $(\mathrm{SP}, \mathrm{pk}_u, \mathrm{sk}_o, F)$. This algorithm inputs SP, pk_u, and sk_o. Then, it outputs the keyword indexes I, file ciphertext C, packed ciphertext C', and encrypted file index set Ns.

(4) *Trap* $(W', \mathrm{pk}_s, \mathrm{pk}_o, \mathrm{sk}_u)$. This algorithm takes queried keyword set W', CS's public key pk_s, DO's public key pk_o, and DU's private key sk_u as input and it outputs the corresponding trapdoor $T_{W'}$ and location information L.

(5) *Search* $(I, T_{W'}, L, \mathrm{sk}_s)$. This algorithm inputs $I, T_{W'}, L$, the CS's private key sk_s. Then, it outputs the encrypted file index set Ns. Note that the search process is run in the blockchain, using the privacy key of CS. Therefore, in the execution of smart contract, there will be an interaction with CS first.

(6) *Verification and Decryption* (SP, C, C', Ns). The algorithm takes SP, Ns, file ciphertext set C, and packed ciphertext C' as input and it outputs the verification results and file set F'.

3.6. Threat Model and Security Model. In this scheme, TA is completely trusted, the DU is malicious, and the CS is semitrusted. For example, the semitrusted CS may want to learn the original file information or return partial search results. DU may also maliciously accuse the CS not returning correct search results. In the payment phase, the CS may want to obtain the search fee from the DU without providing the search result. In addition, a malicious DU may want to get the correct search results from the CS without paying the search fee. Next, we introduce the security model of our scheme.

We define that our scheme needs to satisfy two security goals, one is trapdoor indistinguishability and the other is index indistinguishability. Two games are needed to prove them.

(1) In Game 1, we assume the adversary A is a semitrusted CS or a malicious DU. Therefore, A can get the private key of CS or DU, but he cannot perform trapdoor query on the selected challenge keywords w_0, w_1. The scheme does not get an effective trapdoor, which can ensure the indistinguishability of the index if there is no adversary to distinguish the index of the keyword w_0 or w_1.

(2) In Game 2, A may be a semitrusted CS, and A may get the private key of CS. The trapdoor of the scheme requires that A cannot distinguish $w_0 (W_0)$ or $w_1 (W_1)$.

Definition 1. In Game 1 and Game 2, the scheme can resist inside KGAs if there is no adversary to break the indistinguishability of indexes and trapdoors with a nonignorable advantage. The sequence of games is the interaction between challenger **C** and adversary **A**; pay attention to the semitrusted CS acting as **A**'s role.

4. Construction of the BPKEMS Scheme

4.1. Setup. Input a security parameter λ, and then TA runs the Setup algorithm to generate the system parameters $\mathrm{SP} = (g, h, H_1)$. We set g as a generator of G_1, and h and H_1 are two collision-resistant hash functions, where h: $\{0,1\}^* \longrightarrow Z_p$, H_1: $\{0,1\}^* \times \{0,1\}^* \longrightarrow \{0,1\}^*$. Then, TA publishes the public parameters SP.

4.2. Key Generation. The scheme runs the KeyGen algorithm to generate the public/private key pair for DO, DU, and CS. The detailed generation process is as follows:

(1) KeyGen_o: randomly choose an element $a \in Z_p$ as the private key sk_o and then compute the public key $\mathrm{pk}_o = g^a$.

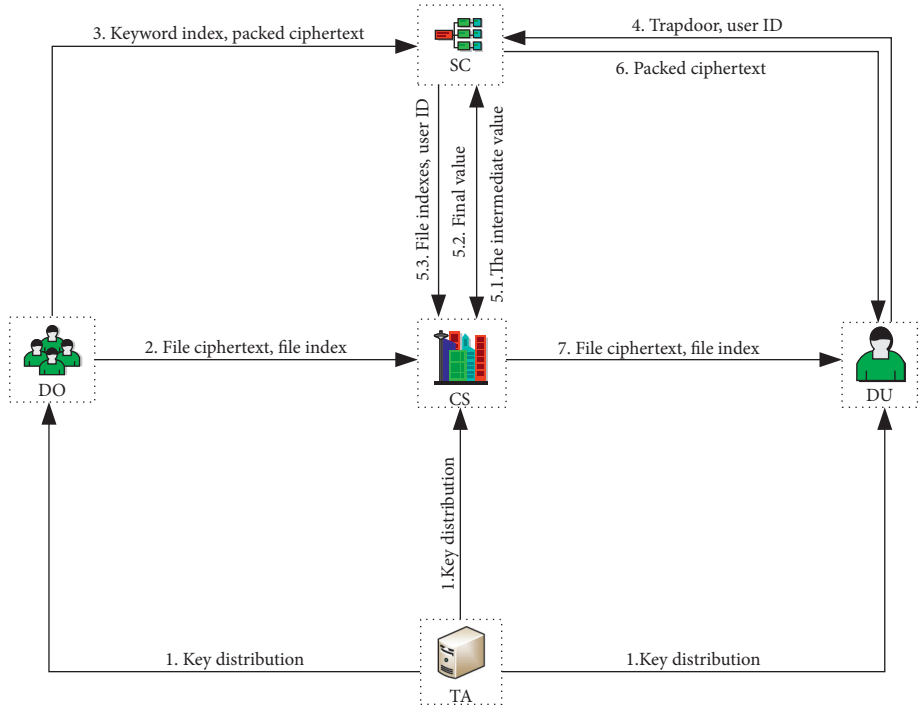


Figure 1: The system model.

(2) KeyGen$_u$: pick an element $b \in Z_p$ as the DU's private key sk_u and compute the public key $e(g,g)^{(1/b)}$, g^b. The DU's public key has two parts, which we define as $\mathrm{pk}_u = \{\mathrm{pk}u_1, \mathrm{pk}u_2\}$, where $\mathrm{pk}u_1 = e(g,g)^{(1/b)}$, $\mathrm{pk}u_2 = g^b$.

(3) KeyGen$_s$: choose an element $c \in Z_p$ as the private key sk_s and then compute the CS's public key $\mathrm{pk}_s = g^c$.

4.3. Ciphertext Generation. Before generating a keyword index, DO first defines the reward offer to be paid per search to himself and sends this setting to the SC. Upon receiving the file set $F = \{f_1, \ldots, f_t\}$, we define the keyword dictionary as $W = \{w_1, \ldots, w_n\}$. DO extracts the keywords in each file. The DO uses the Enc algorithm to output the indexes I, file ciphertexts C, and packed ciphertext C'.

(1) First, DO needs to generate keyword index $I_i = \{I_0, I_1, I_2, I_j\}$, where $i \in [1,t]$. DO randomly chooses an element $r \in Z_p$. Next, he calculates the $I_{i,0} = e(g,g)^r$, $I_{i,1} = (\mathrm{pk}u_2)^r = g^{br}$, $I_{i,2} = e(g,g)^{(ar/b)}$, $I_{i,j} = g^{-rh(w_j)}$, where $j \in [1,n]$.

(2) Second, DO encrypts each file $f_i \in F$. Here, we use a symmetric encryption algorithm when encrypting files. The difference is that we use the idea of DH key exchange to share the key K for DO and DU, and DO uses its own private key sk_o and DU's public key $\mathrm{pk}u_2$ to calculate it, where $K = \mathrm{pk}u_2^{\mathrm{sk}_o} = g^{ba}$. Then, for each file f_i, $C_i = \mathrm{Enc}_K(f_i)$.

(3) Third, DO numbers the file f_i, encrypts the file index i with the key K, obtains the encrypted file index $N_i = \mathrm{Enc}_K(i)$, stores the N_i and the ciphertext C_i together, and then performs a hash operation to obtain the result $M_i = H_1(N_i, C_i)$.

The file indexes N_i, M_i are packed as ciphertext C_i'. Next, upload the encrypted file index set Ns and ciphertext set C to the CS. Then, send the packed ciphertext set C' and index set I to the SC for querying operation.

4.4. Ciphertext Update. In this part, we describe how to update files, for example, modify, insert, and delete operations. For modification and insertion of files, blockchain and encryption protect the index and encrypted files from leaking sensitive information. The detailed file update operations are shown in Figure 2.

(1) *Modification.* Suppose a file m_k needs to be changed to m_k', and DO needs to recalculate its ciphertext, that is, $c_k' = \mathrm{Enc}_K(m_k')$.

(2) *Insertion.* When adding a new file at the k-th position, add the ciphertext at the corresponding position with c_k'.

(3) *Deletion.* When a file needs to be deleted, only the file and index value need to be deleted from the CS.

4.5. Trapdoor. In this section, the Trap algorithm was run by DU. When a DU wants to query keywords $W' = \{w_1', \ldots, w_l'\}$, he needs to generate trapdoor $T_{W'}$ for these keywords. The trapdoor consists of two parts, one is $T_{W',1}$ and the other is $T_{W',2}$.

(1) DU randomly selects an element $\varphi \in Z_p$, let $T_{W',1} = \varphi$.

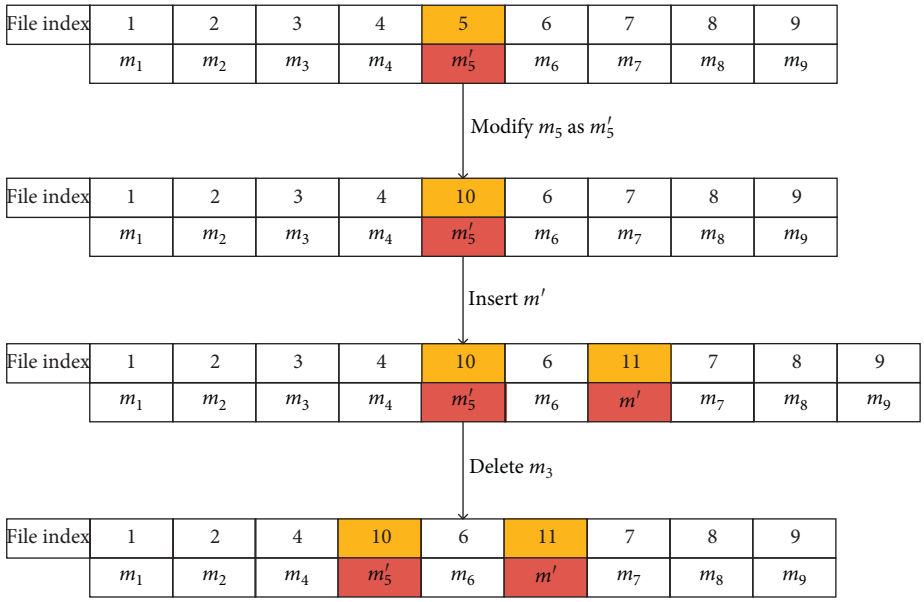


Figure 2: The update operations of files.

(2) DU computes $T_{W',2} = (\mathrm{pk}_s^{-\varphi}\mathrm{pk}_o^{(1/b)})^{(1/(b-\sum_{\pi=1}^{l} h(w'_\pi)))}$.

We need to record the keyword location L, which expresses the location from W' to W. We define a mapping function ρ: $w'_\pi = w_{\rho(\pi)}$. After the user generates the trapdoor $T_{W'}$, the user sets a time limit node T_1, uploads the trapdoor with the location $L = \{\rho(1), \ldots, \rho(l)\}$ to the SC, and performs the deposit operation from his account. Then, the user sends the trapdoor $T_{W'}$ and the time limit node T_1 to the SC. Next, he uploads his own identity ID to request the SC to perform the search service.

4.6. Search. The Search algorithm is run by SC. SC and blockchain are combined for search service. Here, we give the definition of some symbols. owner and user represent the respective accounts of DO and DU. userdeposit expresses the current deposit in blockchain. DU deposits his account balance into the blockchain system user. The price per unit of gasoline is denoted by gasprice. The total cost of each complete search operation is expressed as cost. Gaslsrch and Gassrch, respectively, express the gas limit and the cost of calling the search algorithm. After receiving the DU's ID and requesting the search service, perform the following algorithm.

(1) First, check whether the current time T_2 is less than T_1. If yes, perform the following steps. If no, process is stopped.

(2) Check whether userdeposit is greater than Gaslsrch × gasprice; if yes, the user's current deposit userdeposit can complete the next search service, and the SC starts to run. If no, stop it.

(3) The SC computes the intermediate value $\sigma'_2 = I_{i,0}^{T_{W',1}}$. Then, σ'_2 is sent to CS. CS calculates the final value $\sigma_2 = \sigma_2'^{c}$ and returns it to SC.

(4) Compute $\sigma_1 = e(I_{i,1} \cdot \Pi_{\tau=1}^{l} I_{i,\rho(\tau)}, T_{W',2})$ and $\sigma_3 = I_{i,2}$.

(5) Calculate whether equation $\sigma_1 \cdot \sigma_2 = \sigma_3$ is true. If so, output 1 to indicate that the search was successful. Then, the SC sends the search results to the CS. Otherwise, output 0, indicating failure, and the search service will be stopped. Finally, the SC will record the encrypted file index N_i and then start the next query until all files are retrieved. Finally, SC sends the file index set Ns to CS. We describe the transaction during search in Algorithm 1.

4.7. Verification and Decryption Phase. In this section, DU performs the verification and decryption algorithms. The SC sends file index set Ns and DU's ID that satisfies the search request to CS. Then, CS transmits the file ciphertext set C and encrypted file index set Ns to the DU according to N_i. In Algorithm 1, we describe the search process for each round.

During the interaction between SC and DU, the packed ciphertext C'_i is obtained by the DU after the SC is successfully retrieved. Then, the user verifies $N_{\mathrm{SC}} = N_{\mathrm{CS}}$, where N_{SC} represents the file index sent by the SC and N_{CS} represents the file index sent by CS. If above indexes are the same, it proves that the CS did not send wrong files, and then verify $M' = H_1(N_i, C_i)$, $M \stackrel{?}{=} M'$.

If the file index N_i and ciphertext C_i are hashed and the result is equal, it proves that the CS has not tampered with the ciphertext data. Finally, DU uses its own private key sk_u and DO's public key pk_o to generate the shared key $K = \mathrm{pk}_o^{\mathrm{sk}_u} = g^{ab}$ of the encrypted file and decrypts the file ciphertext C_i, where $f_i = \mathrm{Dec}_K(C_i)$. Finally, DU gets the decrypted file set F'.

4.8. Correctness. Formula (1) indicates that the index and trapdoor match successfully.

(1) if $T_2 < T_1$ and \$userdeposit > Gaslsrch × \$gasprice + \$offer then;
(2) Compute σ_1, σ_2, σ_3;
(3) if $\sigma_1 \cdot \sigma_2$ is the same as σ_3 then;
(4) Return the file indexes N_i to CS;
(5) else;
(6) Return 0;
(7) Set $cost=offer$+Gassrch × $gasprice$;
(8) Send $offer$ to $owner$. Then, send Gassrch × $gasprice$ to executor of a deal;
(9) Finally, set $userdeposit=userdeposit-cost$;
(10) else;
(11) Send $userdeposit$ to $user$;
(12) end;

Algorithm 1: Ciphertexts retrieval.

$$e\left(I_{i,1}\cdot\prod_{\tau=1}^{l}I_{i,\rho(\tau)},T_{W',2}\right)I_{i,0}^{cT_{W',1}}=e\left(g^{br}\cdot\prod_{\tau=1}^{l}g^{-vh\left(w_{\rho_1(\tau)}\right)},\left(g^{(a/b)}g^{c(-\varphi)}\right)^{1/\left(b-\sum_{\tau=1}^{l}h\left(w_\tau'\right)\right)}\right)e(g,g)^{rc\varphi}$$
$$=e\left(g^{r},g^{(a/b-c\varphi)}\right)e(g,g)^{rc\varphi}=e(g,g)^{(ar/b-rc\varphi)}e(g,g)^{rc\varphi}=e(g,g)^{(ar/b)}=I_{i,2}. \quad (1)$$

5. Security Analysis

In order to show that our scheme is practical in terms of security and performance, we introduced the security and performance analysis in detail.

5.1. Fairness. Because the blockchain interacts with each entity on a transaction basis and each transaction is transparent, it can be guaranteed that the results of each query are correct and there will be no malicious tampering. Fairness is achieved through the use of SC. In Ethereum, all operations or transactions are associated with gas, and each operation will consume some of the gas on SC, and the person who provides the data (such as DO) will be rewarded accordingly. At the same time, users also need to pay for the files they retrieve. Without the involvement of a third party, the blockchain can ensure that users get correct and complete search results, and malicious operations will be detected. In addition, the user has determined a limited time to ensure the fairness of the transaction because the transaction needs to be completed within the specified time node. If the time limit is exceeded, the user will stop the search service.

5.2. Credibility. The search results given by the blockchain must be honest and credible. The operations on SC are transparent and cannot be tampered, so we can be confident that the results returned by the SC are credible. At the same time, it effectively prevents malicious server from attacking this scheme. In addition, the transparency of the blockchain can ensure the correctness of the results, and the verification on the user side can also achieve the same effect. Nothing can be used as a malicious tamper with the search results. Entities connected to the blockchain can verify the actions of other entities at any time.

5.3. Confidentiality. This scheme can resist KGAs in theory. The security of this scheme should realize the indistinguishability of index and trapdoor. Note that in Game 1, adversary **A** can query both the private key and trapdoor. Importantly, trapdoor queries need to exclude previously defined challenge keywords. Corresponding to the definition of Game 2, we can get that **A** can query the index ciphertext and CS's private key, the limitation is that **A** cannot query the challenge keywords w_0 and w_1.

Theorem 1. *Through the proof analysis under the random oracle model, we can see that if the adversary solves the corresponding difficult problem with a negligible probability for both Game 1 and Game 2, then our scheme can resist KGAs.*

Proof. The proof of theorem is supported by the following two lemmas. As long as their security requirements can be satisfied, our scheme is secure in the description of theorem. The detailed process is as follows. In Game 1, if the DDH assumption holds, the scheme achieves index indistinguishability. In Game 2, the scheme can ensure that it can resist chosen keyword attacks under the random oracle model.

(1) In Game 1, we analyze the symmetric key used to encrypt files between DO and DU, which is generated through negotiation between the two entities. The CS must obtain the private key of one before it can generate a shared key or intercept it during the transmission of the public channel. However, our scheme does not require transmission. Therefore, the CS must obtain the private key of one of them to decrypt the ciphertext of the file f_i. Therefore, in our scheme, the shared key K is secure.

(2) The security of our scheme can be analyzed from two parts. The first is the generation of the index. Assuming that a DU wants to query a keyword set W', CS must generate a valid index $\widetilde{I} = \{\widetilde{I}_o, \widetilde{I}_1, \widetilde{I}_2, \{\widetilde{I}_j\}\}$. CS first needs to obtain the private key $\mathrm{sk}_o = a \in Z_p$ of DO. The private key of DO is kept secret; CS can only assume that it has obtained a private key $\widetilde{a}$ of the DU. But the size of Z_p is p, which is a large prime number. Therefore, the probability of selecting the right one is $(1/p)$, which is negligible. On the other hand, CS assumes that the keyword set $W' = \{w'_1, \ldots, w'_l\} = \widetilde{W'} = \{\widetilde{w}_1, \ldots, \widetilde{w}_l\}$ selected by CS is equal to the keyword set that DU wants to query, which is equivalent to randomly selecting l equal sets from n keywords, with a probability of $(1/C_n^l)$. Assuming that the range of the key set n is large enough, the above probability is also small enough. In summary, CS cannot perform inside KGAs.

(3) In Game 2, given a valid index $I = \{I_0, I_1, I_2, \{I_j\}\}$, CS cannot generate a valid trapdoor for matching. The generation of the trapdoor requires the use of the private key sk_u of the DU. We assume that the private key of the DU is $\mathrm{sk}_u = b$. CS randomly selects a element $\widetilde{b} \in Z_p$ as the private key of the DU. The equal probability is $(1/p)$, so the probability can be ignored. Through the above analysis, our scheme can resist inside KGAs.

Here, we introduce the location privacy of keywords. In the paper, we use the location mapping function $\rho(\cdot)$. The location privacy of queried keywords can be protected using random mask technology, for example, pseudorandom functions. The pseudorandom function confuses the position of the real keyword so as not to riot the position of the real keyword. Try not to let users know more information. For cloud server, the index location is exposed, but the keywords are encrypted, so the security of the scheme will not be affected.

6. Performance Analysis

In order to show that our scheme is effective, in this part, we compare three schemes in terms of functions. In addition, we discuss the computation overhead and communication overhead of our scheme with two other schemes: Yang's scheme [5] and Xu's scheme [37].

First, we compare the functions of the three schemes, as shown in Table 2. We can see that by comparing the functions of the four aspects, we can see the functional differences between those schemes. The check mark means that this condition is satisfied, and the wrong sign means that the condition is not satisfied. It is compared by whether it supports multi-keyword retrieval, whether it supports dynamic update of files, whether it supports blockchain, and whether it supports fair payment between users. We can see that our scheme supports the four functions, scheme [37] only supports multi-keyword search, and scheme [5] only does not support dynamic update of files. The dynamic update of files can ensure the flexibility of the scheme. By using the blockchain, you can take advantage of the transparency, immutability, and traceability. Especially the SC running on the blockchain can ensure fair payment between users.

Table 2: Functional comparison.

	Our scheme	Yang's scheme [5]	Xu's scheme [37]
Multi-keyword	✓	✓	✓
Update	✓	×	×
Blockchain	✓	✓	×
Fair payment	✓	✓	×

6.1. Theoretical Analysis. In Table 3, we compare the computation overhead of our scheme with the other two schemes [5, 37]. In terms of computation overhead, we mainly consider some time-consuming operations; T_M represents a multiplication operation, T_H represents a hash operation, T_E represents an exponential operation, and T_P represents a pair operation. In Table 4, we compare our scheme with other schemes [5, 37] in terms of communication overhead. We define the element length of G_1, G_2, Z_p as $|G_1|, |G_2|, |Z_p|$. In addition, we define m to represent the number of keywords contained in each file and l to represent the number of queried keywords.

Regarding the computation overhead, we compare the characteristics of each scheme in Table 3. In the key generation stage, we can see that our scheme is in the middle of the three at this stage, and the efficiency is higher than that of scheme [5] and lower than that of scheme [37]. In the keyword encryption and trapdoor generation phases, the calculation amount of the three schemes increases linearly with the number of encrypted keywords and queried keywords, but our scheme is the most efficient among the three, which are $(3+m)T_E + mT_H + T_P$ and $3T_E + lT_H + T_M$, respectively. In the search stage, we set the number of keywords to be queried to 1. It can be seen from the table that the calculation amount of the three schemes is constant, but our scheme has the highest efficiency. Therefore, based on the above theoretical analysis, our scheme has the highest efficiency.

Regarding communication overhead, we compare the public key size, encryption size, and trapdoor size with the other two schemes. We can see from Table 4 that the size of the public key generated by the three schemes remains unchanged. In the encryption phase, the size of the storage of our scheme is almost the same as scheme [5] but is smaller than the storage size of scheme [37]. In the trapdoor generation stage, in scheme [37] the size of trapdoor increases linearly with the number of queried keywords, and therefore, it will consume a lot of storage resources. Our scheme and

Table 3: Computation overhead.

	Our scheme	Yang's scheme [5]	Xu's scheme [37]
KeyGen	$4T_E + T_p$	$2T_H + 4T_E$	$3T_E$
Enc	$(3+m)T_E + mT_H + T_P$	$mT_M + mT_H + (2m+3)T_E$	$mT_M + 3mT_H + (2m+2)T_E$
Trapdoor	$3T_E + lT_H + T_M$	$(l+1)T_M + lT_H + (2l+3)T_E$	$3lT_H + (2l+1)T_E$
Search	$T_M + T_E + T_P$	$T_M + 3T_P$	$T_M + T_E + 3T_P$

Table 4: Communication overhead.

	Our scheme	Yang's scheme [5]	Xu's scheme [37]
pk size	$\|G_1\|$	$\|G_1\|$	$\|G_1\|$
Enc size	$(m+1)\|G_1\| + 2\|G_2\|$	$(m+3)\|G_1\|$	$(m+2)\|G_1\| + m\|Z_p\|$
Trapdoor size	$\|G_1\| + \|Z_p\|$	$3\|G_1\|$	$3\|G_1\| + l\|Z_p\|$

scheme [5] are constant and therefore have good storage characteristics.

6.2. Empirical Analysis. In this part, we emulate our scheme, Yang's scheme [5], and Xu's scheme [37]. We use the Java Pairing-Based Cryptography (JPBC) Library. The implementation equipment of the scheme is a HP desktop computer with a 3.00 GHz Intel Core i5-8500 processor and 8 GB memory. In the experiment, we used the Type A elliptic curve. We analyzed three schemes by comparing Enc, Trap, and Search algorithms. In the Enc algorithm, we set the number of keywords in steps of 10, increasing from 1 to 50 in turn. In Trap and Search, the number of keywords we set is also increasing from 10 to 50 in steps of 10. In each of the above experiments, after 50 cycles, the average value of the calculation cost is calculated to ensure that the results are relatively valid. It can be seen from Figures 3–5 that our scheme is the most effective. Below we briefly explain the content of the icon.

In Figure 3, we can see that our scheme has the smallest slope, which has great advantages compared with the other two schemes. Due to the frequent hashing operations and exponential operations, the coefficients of our scheme (m and $3+m$) are larger, so the structure of the scheme is simpler. With the increase of keywords, the advantages will become more and more obvious.

In Figure 4, we can find that the time consumed is constant with the number of keywords that users query. In the process of generating trapdoors of our scheme, exponential operations and multiplication operations are constants, and hash operations increase linearly with the increase of keywords. However, you can see that in the other two schemes, the slope of growth is much larger than that of our scheme, and it takes time to hash to Z_p which is much shorter than hashing to group G.

In Figure 5, the efficiency gap between our scheme and the other two schemes is not obvious. Because of the pair operation, the number of operations of exponential operation is almost constant. For the operation after hashing the keyword, whether it is the aggregation of addition or the aggregation of multiplication, the time consumed by a single operation is very small. Therefore, as the number of keywords increases, the trend of time changes is not obvious. But judging from the change trend in Figure 5, our scheme still has some advantages.

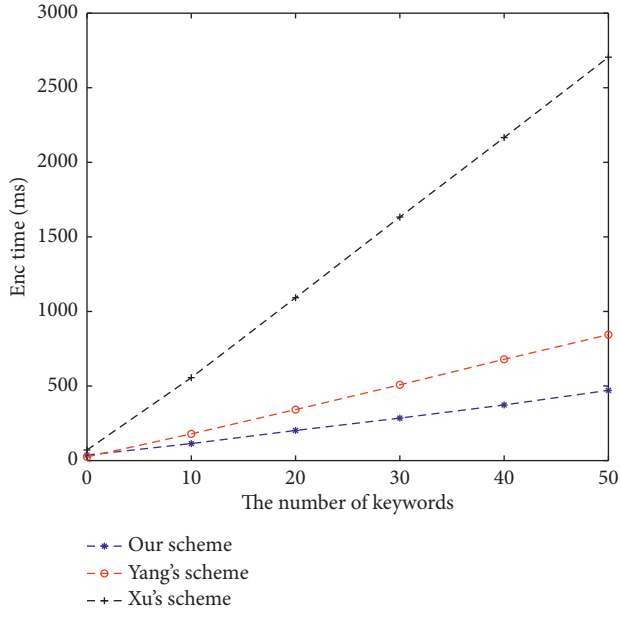


Figure 3: The computation overhead of Enc algorithm.

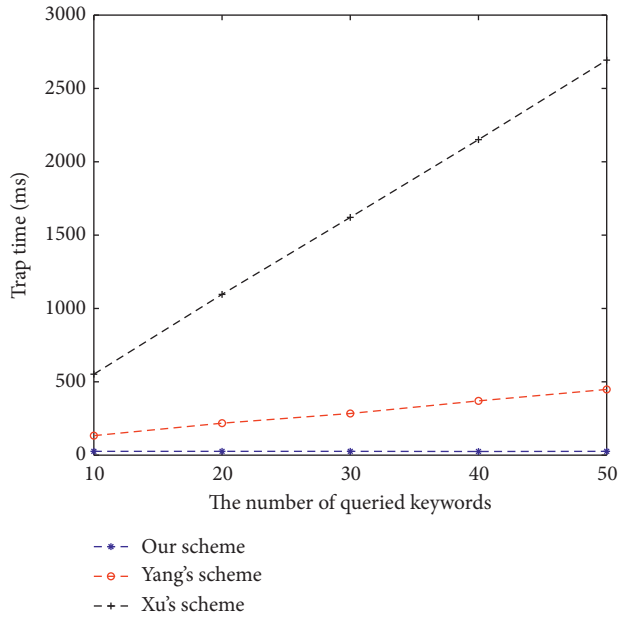


Figure 4: The computation overhead of Trap algorithm.

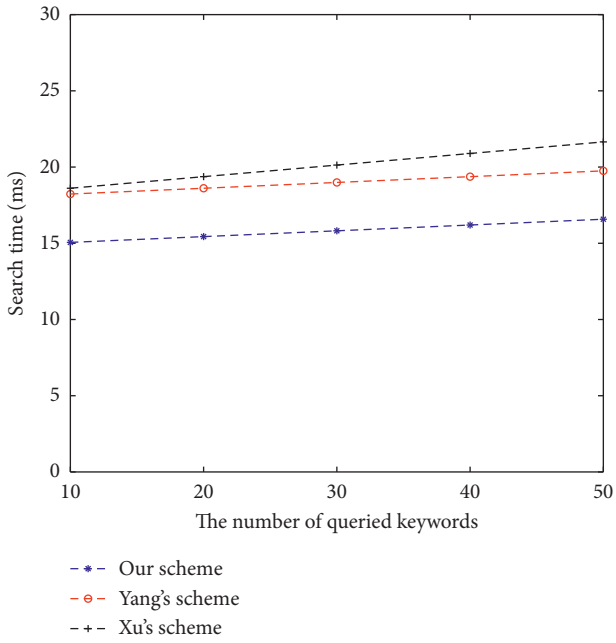


Figure 5: The computation overhead of Search algorithm.

7. Conclusion

With the development of cloud computing, a secure search cryptography scheme is becoming increasingly important. In this paper, we present a BPKEMS scheme in the blockchain scenario, which supports secure retrieval of conjunctive keywords, dynamic update of files, and verification of ciphertext. In addition, our scheme can resist KGAs. In terms of efficiency, we implemented this scheme through simulation and compared it with other schemes [5, 37], and it shows that our scheme is more practical.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was supported by the National Key R&D Program of China (grant no. 2017YFB0802000), the National Natural Science Foundation of China (grant nos. 61862055 and 62072369), and the Major Research and Development Project of Qinghai (grant no. 2020-SF-140).

References

[1] D. X. Song, D. W. Wagner, and A. Perrig, "Practical Techniques for Searches on Encrypted Data," in *Proceedings of the 2000 IEEE Symposium on Security and Privacy*, pp. 44–55, Berkeley, CA, USA, February 2000.

[2] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in *Advances in Cryptology-EUROCRYPT 2004*, C. Cachin and J. L. Camenisch, Eds., pp. 506–522, Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[3] T. Chi, B. Qin, and D. Zheng, "An efficient searchable public-key authenticated encryption for cloud-assisted medical internet of things," *Wireless Communications and Mobile Computing*, vol. 2020, pp. 1–11, Article ID 8816172, 2020.

[4] Y. Miao, X. Liu, K.-K. R. Choo, R. H. Deng, H. Wu, and H. Li, "Fair and dynamic data sharing framework in cloud-assisted internet of everything," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 7201–7212, 2019.

[5] X. Yang, G. Chen, M. Wang, T. Li, and C. Wang, "Multi-keyword certificateless searchable public key authenticated encryption scheme based on blockchain," *IEEE Access*, vol. 8, pp. 158765–158777, 2020.

[6] Q. Zheng, S. Xu, and G. Ateniese, "Vabks: verifiable attribute-based keyword search over outsourced encrypted data," in *Proceedings of the IEEE INFOCOM 2014-IEEE Conference on Computer Communications*, pp. 522–530, IEEE, Toronto, ON, Canada, May 2014.

[7] W. Sun, X. Liu, W. Lou, Y. T. Hou, and H. Li, "Catch you if you lie to me: efficient verifiable conjunctive keyword search over large dynamic encrypted cloud data," in *Proceedings of the 2015 IEEE Conference on Computer Communications (INFOCOM)*, pp. 2110–2118, IEEE, Kowloon, HK, USA, May 2015.

[8] Z. Xia, X. Wang, X. Sun, and Q. Wang, "A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 2, pp. 340–352, 2015.

[9] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: improved definitions and efficient constructions," *Journal of Computer Security*, vol. 19, 2011, https://www.microsoft.com/en-us/research/publication/searchable-symmetric-encryption-improved-definitions-and-efficient-constructions-2/.

[10] A. Fiat and M. Naor, "Broadcast encryption,," in *Annual International Cryptology Conference*, pp. 480–491, Springer, Berlin, Germany, 1993.

[11] P. Wang, H. Wang, and J. Pieprzyk, "Threshold privacy preserving keyword searches," in *Proceedings of the International Conference on Current Trends in Theory and Practice of Computer Science*, pp. 646–658, Springer, Nový Smokovec, SL, Europe, January 2008.

[12] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612-613, 1979.

[13] K. Yuan, Z. Liu, C. Jia, J. Yang, and S. Lv, "Public key timed-release searchable encryption in one-to-many scenarios," *Acta Electronica Sinica*, vol. 43, no. 4, pp. 760–768, 2015.

[14] H. Zhong, J. Cui, R. Shi, and C. Xia, "Many-to-one homomorphic encryption scheme," *Security and Communication Networks*, vol. 9, no. 10, pp. 1007–1015, 2016.

[15] Q. Tang and L. Chen, "Public-key encryption with registered keyword search," in *Proceedings of the European Public Key Infrastructure Workshop*, pp. 163–178, Springer, Pisa, Italy, September 2009.

[16] L. Fang, W. Susilo, C. Ge, and J. Wang, "Public key encryption with keyword search secure against keyword guessing attacks without random oracle," *Information Sciences*, vol. 238, pp. 221–241, 2013.

[17] P. Xu, H. Jin, Q. Wu, and W. Wang, "Public-key encryption with fuzzy keyword search: a provably secure scheme under

keyword guessing attack," *IEEE Transactions on Computers*, vol. 62, no. 11, pp. 2266–2277, 2012.

[18] R. Chen, Y. Mu, G. Yang, F. Guo, and X. Wang, "A new general framework for secure public key encryption with keyword search," in *Proceedings of the Australasian Conference on Information Security and Privacy*, pp. 59–76, Springer, Brisbane, QLD, Australia, July 2015.

[19] Z.-Y. Shao and B. Yang, "On security against the server in designated tester public key encryption with keyword search," *Information Processing Letters*, vol. 115, no. 12, pp. 957–961, 2015.

[20] R. Chen, Y. Mu, G. Yang, F. Guo, and X. Wang, "Dual-server public-key encryption with keyword search for secure cloud storage," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 4, pp. 789–798, 2015.

[21] Q. Huang and H. Li, "An efficient public-key searchable encryption scheme secure against inside keyword guessing attacks," *Information Sciences*, vol. 403, pp. 1–14, 2017.

[22] Y. Kang and Z. Liu, "A fully secure verifiable and outsourced decryption ranked searchable encryption scheme supporting synonym query,," in *Proceedings of the 2017 IEEE Second International Conference on Data Science in Cyberspace (DSC)*, pp. 223–231, IEEE, Shenzhen, China, June 2017.

[23] L. Wu, B. Chen, S. Zeadally, and D. He, "An efficient and secure searchable public key encryption scheme with privacy protection for cloud storage," *Soft Computing*, vol. 22, no. 23, pp. 7685–7696, 2018.

[24] L. Wu, Y. Zhang, M. Ma, N. Kumar, and D. He, "Certificateless searchable public key authenticated encryption with designated tester for cloud-assisted medical internet of things," *Annals of Telecommunications*, vol. 74, no. 7-8, pp. 423–434, 2019.

[25] M. Ma, D. He, N. Kumar, K.-K. R. Choo, and J. Chen, "Certificateless searchable public key encryption scheme for industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 2, pp. 759–767, 2017.

[26] Y. Lu and J. Li, "Efficient searchable public key encryption against keyword guessing attacks for cloud-based emr systems," *Cluster Computing*, vol. 22, no. 1, pp. 285–299, 2019.

[27] Y. Zhang, R. Deng, X. Liu, and D. Zheng, "Outsourcing service fair payment based on blockchain and its applications in cloud computing," *IEEE Transactions on Services Computing*, vol. 123, pp. 89–100, 2018.

[28] Y. Zhang, R. H. Deng, X. Liu, and D. Zheng, "Blockchain based efficient and robust fair payment for outsourcing services in cloud computing," *Information Sciences*, vol. 462, pp. 262–277, 2018.

[29] Y. Zhang, R. H. Deng, J. Shu, K. Yang, and D. Zheng, "Tkse: trustworthy keyword search over encrypted data with two-side verifiability via blockchain," *IEEE Access*, vol. 6, pp. 31077–31087, 2018.

[30] H. Li, F. Zhang, J. He, and H. Tian, "A searchable symmetric encryption scheme using blockchain," 2017, http://arxiv.org/abs/1711.01030.

[31] H. Li, H. Tian, F. Zhang, and J. He, "Blockchain-based searchable symmetric encryption scheme," *Computers & Electrical Engineering*, vol. 73, pp. 32–45, 2019.

[32] L. Chen, W.-K. Lee, C.-C. Chang, K.-K. R. Choo, and N. Zhang, "Blockchain based searchable encryption for electronic health record sharing," *Future Generation Computer Systems*, vol. 95, pp. 420–429, 2019.

[33] Y. Zhang, R. Deng, E. Bertino, and D. Zheng, "robust and universal seamless handover authentication in 5g Hetnets," *IEEE Transactions on Dependable and Secure Computing*, vol. 99, 2019.

[34] N. Szabo, "Formalizing and securing relationships on public networks," *First Monday*, vol. 2, no. 9, 1997.

[35] A. Miller, Z. Cai, and S. Jha, "Smart contracts and opportunities for formal methods," in *Proceedings of the International Symposium on Leveraging Applications of Formal Methods*, pp. 280–299, Springer, Limassol, Cyprus, November 2018.

[36] S. Hu, C. Cai, Q. Wang, C. Wang, X. Luo, and K. Ren, "Searching an encrypted cloud meets blockchain: a decentralized, reliable and fair realization," in *Proceedings of the IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, pp. 792–800, IEEE, Honolulu, Hawaii, USA, April 2018.

[37] L. Xu, J. Li, X. Chen, W. Li, S. Tang, and H.-T. Wu, "Tcpedcks: towards time controlled public key encryption with delegatable conjunctive keyword search for internet of things," *Journal of Network and Computer Applications*, vol. 128, pp. 11–20, 2019.