Hindawi
Security and Communication Networks
Volume 2022, Article ID 5428539, 13 pages
https://doi.org/10.1155/2022/5428539

Research Article

CL-BC: A Secure Data Storage Model for Social Networks

Dawei Xu,[1,2] **Weiqi Wang**,[2] **Liehuang Zhu**,[1] **Jian Zhao**,[2] **Fudong Wu**,[2] **and Jiaqi Gao**[2]

[1]*School of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing 100081, China*
[2]*College of Cybersecurity, Changchun University, Changchun 130022, China*

Correspondence should be addressed to Liehuang Zhu; liehuangz@bit.edu.cn

Received 19 October 2021; Accepted 20 January 2022; Published 27 February 2022

Academic Editor: Zhe-Li Liu

While bringing convenience to people, social networks cause various security problems, such as information leakage, which belongs to the security category of data storage. The related research focuses more on the confidentiality design and lacks the discussion of data integrity protection technology. Thus, the electronic data editability invites the risk of data tampering and destruction, making other operations meaningless once the data are inaccurate. A social network-oriented data security storage model, CL-BC, included conceptual design, preliminary framework design, and detailed flow design, to improve the security threats such as overauthorization and illegal execution in data storage. The blockchain technology is combined with the improved Clark–Wilson model to achieve data integrity access control protection through execution rules and authentication rules. The system uses the subalgorithm of DR-BFT for reference when deploying the algorithm program and adopts the intelligent contract to complete the design rules. The experimental analysis shows the CL-BC model adapts to the actual security environment, ensures the data storage integrity in the social network stored process, and has application value.

1. Introduction

With the development of mobile Internet and social networks, the network has become an information transmission carrier in people's normal life, and the transmitted content is stored in virtual space constantly. A safe storage technology [1], electronic data storage technology, is introduced to improve the easy preservation of data storage management. There are usually two forms: (i) manually input the original paper version into electronic files that can be archived by scanning and other means; and (ii) generate files directly from the computer. Electronic technology is controlled by the centralized network, and the increase of operations, such as copying and moving [2], virtualization [3], and disorderly execution [4], makes it easy for data to be read and written illegally by the same network users.

Access control technology [5], as one of the important technologies to protect the integrity, evaluates whether the trader has the right to execute the current transaction according to the trader identity, to regulate the access behaviour to the network resource storage. In practice, mandatory access control is usually adopted. Such strategies need to consider both subjects, objects, and their relationships, and models are more reliable than free access control. Common mandatory access control models include Bell–Lapadula [6], Biba [7], Clark–Wilson [8], and Chinese Wall [9]. The Bell–Lapadula model is the first strategy to provide multiple levels of data confidentiality protection, divided by data sensitivity, commonly used in the military field. The Biba model, also oriented to the military field, protects the integrity of subject and object by assigning levels, and its rules of read and write attributes are opposite to the Bell–Lapadula model. Clark–Wilson is a business-oriented integrity security model, whose implementation is based on a comprehensive transaction processing mechanism. Chinese Wall is a security model applied to multiple organizations, places a "Wall" on the information transmission path with competing subjects, and only allows accessible information flow to pass through. However, most secure access control models operate in a centralized environment and rely on third parties to assign permissions, prone to single point failure

or the several dishonest participants collusion to undermine data integrity.

Blockchain technology effectively solves such problems. A distributed structure [10] determines multiple paths between nodes that send and receive information, and information transactions choose the shortest path for transmission. Even if a path fails, information transmission will not be affected, providing a better operating environment for the access control model. In addition, its imtamability, timing, and traceability of blockchain technology guarantee the transaction information integrity stored on the chain. Therefore, blockchain technology is often used in the field of data storage.

This article proposes a new integrity data storage model CL-BC, which includes conceptual design, preliminary frame design, and detailed flow design, to build the access control mechanism of distributed integrity data storage model and improve the security threats such as overauthorization and illegal execution in data storage. The Clark–Wilson model is the new design starting point, detailed rules are formulated, and the coherent process of rules is realized. The PBFT algorithm is combined with parts of DR-BFP to construct smart contracts for trading consensus, and finally, the model performance is evaluated.

The rest of the article is structured as follows: Section 2 introduces the basic knowledge of related work. Sections 3 and 4 describe the infrastructure and deployment process of the CL-BC storage model, respectively. Section 5 presents the security analysis and partial results of the experiment evaluation. Section 6 concludes.

2. Background and Related Work

2.1. Clark–Wilson Model. The Clark–Wilson model was put forward by Clark et al. [11] for the commercial environment. It proposed for the first time that different from the confidentiality importance in the military field, the prevention of fraud and error is the primary goal in such an environment, and these goals need to be achieved through integrity rather than privacy. The model proposes two core mechanisms for maintaining integrity: well-structured transaction and responsibilities separation. The former means users only perform transactions in a restrictive way. The latter allocates different stages of the same transaction to different users to avoid collusion and better maintain the data validity.

Some scholars do research on this basis. For example, Haraty et al. [12] implemented the protection policy using a combination of the Clark–Wilson model and role-based access control model and tested the model validity with Alloy language and profiler. Ramkumar [13] proposed a BSCI integrity framework for large-scale information systems, which transformed the problem of how to eliminate faults in complex software and protect the complex platforms integrity into verifying the FSM model correctness and ensuring simple process integrity in the system. Tsegaye et al. [14] proposed a model combining RBAC, ABAC, and Clark–Wilson to improve the information security in medical care. The three models complement each other to better maintain the security of electronic health records. Avorgbedor et al. [15] designed the I4 model based on the Clark–Wilson model to improve the implementation plan from the aspects of integrity, audit, authorization, and access control, and customize specific rules for identifying, verifying, and evaluating data integrity threats in the privacy environment to improve Facebook's frequent privacy exposure.

In short, the Clark–Wilson model, as an integrity access control model, has been introduced into various research fields. In terms of completeness, the model implements integrity access control policies through authentication rules and execution rules. Authentication rules introduce application integrity definition, and execution rules are independent of application security functions. Based on this, the resources integrity protection can be better realized, so we choose the Clark–Wilson model as the basic framework. However, the existing model is centralized and has the risk of collusion. Therefore, blockchain technology is selected to remedy the inherent shortcomings of the Clark–Wilson model.

2.2. Blockchain Technology. Blockchain [16] is an account, ledger, and independent database that participants on the chain jointly read, write, and store. It uses a chain structure to link effective data blocks, cryptography to ensure the data security in decentralized sharing, and peer-to-peer network and consensus mechanism to generate and update data, and uses smart contract to operate data to maintain the resources integrity.

The blockchain significant advantages, such as distributed storage [17], time-series data and tamper-resistant unforgeable, decentralization, and smart contracts automatic execution, and relying on distributed consensus agreement [18], satisfy trading integrity verification and improve the traditional data storage mode's low electronic degree, easy data loss, easy tampering, and inefficient legal service process. Blockchain technology is commonly used in different storage scenarios, such as in the field of cryptocurrency [19], medical care [20], agricultural product traceability [21], digital copyright [22], and legal [23].

The scholars have consistently put the confidentiality, integrity, and availability of data security in the blockchain system [24] at the top of the list. However, most scholars focus on the study of data security confidentiality [25, 26], ignoring that data integrity is more important in the business field.

Zhu et al. [27] use blockchain to protect data integrity, focusing on assessing security from the perspective of controllability, privacy protection, and unforgeability. Khan et al. [28] use blockchain ledgers to record metadata that can be used to distinguish between real and fake videos in surveillance recordings. However, the device can only send certain frames for authentication. Mercan et al. [29] proposed a cloud Internet of Things data integrity check scheme based on blockchain technology and homomorphic hash, which utilizes executer signature and hash computing to

maintain data integrity. However, it is only applicable to video and has limitations.

Relevant studies are still lacking, especially the research on the combination of blockchain technology and Clark–Wilson model. The integrity protection model proposed in this article improves the problems of fraud and error, uses blockchain technology combined with mandatory access control mechanism to prevent malicious modification by unauthorized users and improper use by authorized users, and ensures the resources execution by authorized users, to realize the data storage system integrity protection.

3. CL-BC Storage Model Architecture

3.1. Overview of the CL-BC Model. CL-BC is a security model for ensuring data integrity based on a distributed structure. This model was first proposed by our team, a security policy based on the characteristics of blockchain with modified attributes based on the Clark–Wilson model. Its architecture consists of three parts: data layer, integrity control layer, and transaction transmission layer from bottom to top, respectively, as shown in Figure 1. The data layer covers the raw data formed in the computer, and all resources generated during the transaction, the target of security model protection. We can think of this layer as a huge database, but it contains a wider range of categories, such as logs, images, and compressed packages. In this model, data of various categories are usually simply divided into CDIs and UDIs (see 3.2.1 for a relevant introduction). In the transaction transport layer, a transaction agreed upon by each node is packaged and broadcast in the blockchain network in a specified order to update the state of the world.

In CL-BC, combined with the actual evidence storage situation, usually multiple users operate a platform for different types of data at the same time. Therefore, an integrity control layer can contain many integrity control domains to improve the actual operation efficiency. Different integrity control domains are isolated from each other so that operations of each do not affect others, which controls the length of the chain and enables the isolation of different businesses in the operating environment, reducing the data leakage risk. With smart contracts, operations are performed strictly according to predefined integrity rules.

3.2. CL-BC Integrity Policy. Facing the data storage, it is essential to (i) prevent information from being maliciously tampered with by unauthorized users, (ii) prevent misoperation by the authorized users, and (iii) prevent information on the system platform from being inconsistent with real-world information. To that end, it is easier to enforce integrity than privacy. The strategy of the CL-BC model is introduced next from the following three aspects: basic elements, integrity core mechanism, and specific rules.

3.2.1. Basic Elements. First, define the elements of the model:

(1) *Subject and Object*. A basic concept in epistemology. The active and creative is called Subject S, and the objective is called Object O. The same thing can be divided into different categories under different limiting conditions. For example, a man can be both S and O. In this section, S is defined as the performer who changes the state of a transaction, that is, a user or a process; O is defined as the resource in the operation being performed, that is, a data item or a dataset.

(2) *Data Items*. The smallest unit used to record data. In this article, data items are divided into two categories: I) restricted data items (CDIs) c^k, to provide integrity protection and control; (ii) unrestricted data items (UDIs) u^k, containing all unrestricted content except CDIs. Different permissions are assigned to object data items according to their categories.

(3) *Dataset*. Multiple O's form a data element, and then, multiple data elements form a dataset G. (i) A restricted dataset by $C = \{c^1, \ldots, c^k\}$ can contain truthful name information, certificate number, etc.; (ii) an unrestricted dataset is determined by $U = \{u^1, \ldots, u^k\}$ and can contain contents such as users' virtual nicknames. In the same system, the attribute of G satisfies

$$\begin{aligned} \mathrm{C} \cup \mathrm{U} &= \left\{C^1, \ldots C^k, \right\} + \left\{U^1, \ldots, U^k\right\} \\ &= \left\{\sum_1^k c^k\right\} + \left\{\sum_1^k u^k\right\} \\ &= \mathrm{G}, \mathrm{C} \cup \mathrm{U} = \varnothing. \end{aligned} \tag{1}$$

(4) *State Model*. State model $\mathrm{A_T}$ contained in a system can be regarded as a two-dimensional function, and the variables are time and data items. For example, at t a G (g can be u^k or c^k) has a state of α_t^k in the information system. It can be expressed specifically as

$$\begin{aligned} \mathrm{A_T} &= \left\{\sum_{t=1}^{T}\sum_1^k {\alpha_t}^k\right\} \\ &= \left\{\sum_{t=1}^{T} {\alpha_t}^k + \cdots + {\alpha_t}^k\right\}, \end{aligned} \tag{2}$$

where T represents the standard time of the National Time Center, $t \leq T$.

(5) *Transaction Set*. The protected set of transactions in the system can be represented by the ternary relation B (S, O, β), any element of B is b, and $b = (\mathrm{s_i}, \mathrm{o_j}, \beta)$ represents a subject to perform β trading operation on an object, satisfying the attribute β (α_t^k)-> α_{t+1}^k. β contains IVPs and TPs, which are described next.

(6) *Integrity Verification Procedures (IVPs).* IVPs$\subset\beta$, (i) α_t^k of the g satisfying each moment t is verified always valid. (ii) Also it is used to verify the validity of g forming a new data item g' in dynamic β, that is, to verify whether the transaction process of all data α_{t-}^k-> α_{t+}^k is valid.

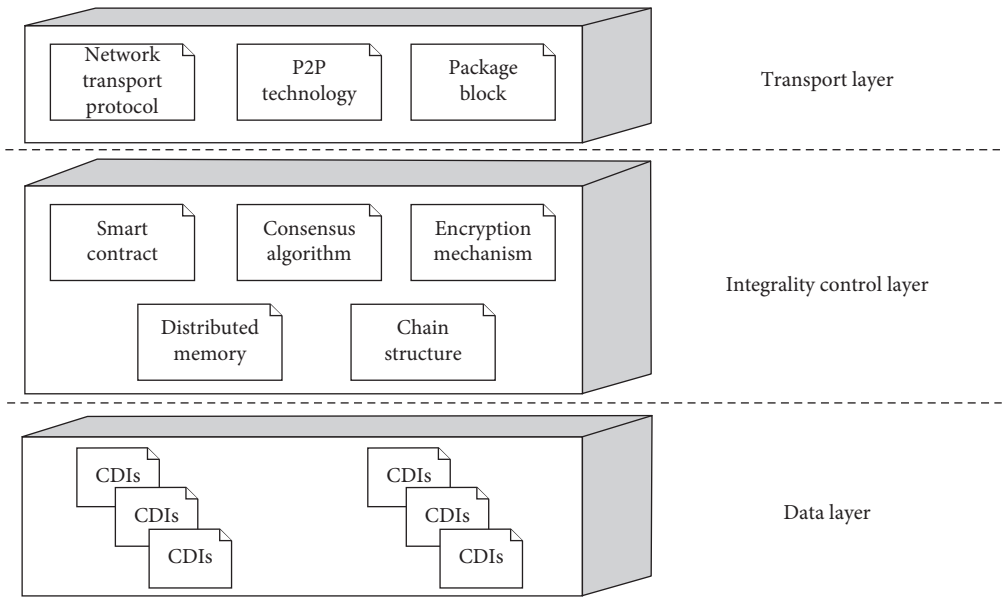

Figure 1: CL-BC architecture.

(7) *Translation Program (TPs)*. TPs $\subset\beta$, (i) the relation of ($\beta(k)$, O) in CR_2 is observed, and the $\beta(k)$ that satisfies the g is unique at the same time; (ii) and the states before α_{t-}^{k} and after α_{t+}^{k} of g are valid.

(8) *Security Model* P_T. P is a set of security functions to maintain the security of A_T model, expressed as P (A_T, IVPs, TPs, g, t). Ensure its security according to authentication rules and execution rules (see Section 3.2.3).

3.2.2. Security Attribute. The security attribute of this model is realized according to the security program guarantee of the Clark–Wilson model, expanded according to the actual application, thus forming the CL-BC model. The core model is to reach agreement both internally and externally to maintain the system state A_T, around the following attributes:

Attr.1. Integrity

Ensure the α_t^k of static g item g at each t is always valid and ensure that g forms a new valid g' in a restricted way in the dynamic β, with properties guaranteed by CR_1, CR_2, CR_5, ER_1, ER_4.

Attr.2. Access control

The control ability ling S to resource O is provided by CR_3, ER_2, ER_3, and ER_4.

Attr.3. Audit

Ensure the g in the execution of any β change process, and the validity of the system α_t^k, that is, maintain the validity of β (α_t^k)-> α_{t+}^k, guaranteed by CR_1, CR_4.

Attr.4. Accountability

Ensure the unique mapping of the S with corresponding β; that is, the number of β times performed by the g at t can only be 0 or 1. When β is traced, S can be precisely located. It is guaranteed by rules ER_2 and ER_3.

3.2.3. Transformation Process Rules

CR_1 (authentication): c^k must be valid at any time. The system has IVPs and integrates various risk indicators to ensure the c^k is in a valid state when verifying α_t^k. c^k authenticated is static data at different times and belongs to a noun. For example, it is effective to query the current real-name information of a subject.

CR_2 (consensus): it is necessary to ensure the data transaction process α_{t-}^k-> α_{t+}^k is valid. During the whole process of the TPs, the system ensures the transition of valid data c^k from the state α_{t-}^k of time t-to the α_{t+}^k of t+ is agreed upon. At this point, reach a consensus that the dynamic data of c^k in the α_{t-}^k-> α_{t+}^k state belong to a verb and exist a specific relationship between transaction β^k and object O: (β^k, O), where O $C=\{c^1,\ldots, c^k\}$. For example, when β is stored account information, O is the account data saved on the network after consensus reached.

CR_3 (authorization): ensures only the TPs can change the α_t^k of c^k. Meanwhile, the P_T needs to comply with the least privilege principle [30] and the responsibilities separation principle. The responsibilities separation is to satisfy the (S, (β^k, O)) rule described in ER_2.

CR_4 (backup): agreed transaction α_{t-}^k-> α_{t+}^k ensured to be traceable. State change α_{t-}^k-> α_{t+}^k backed up and recorded in a separate C in time, and the c^k at each moment used to restore the scene when the transaction or state is queried (use the timestamp to record the time node, and the Merkel tree to verify the restoration correctness).

CR_5 (UDI rule): guarantees the status of u^k is correct. The α_t^k of u^k is authenticated and, when reaches the restricted threshold, converted to c^k in time through the limited TPs, that is, TPs (u^k)-> c^k. Otherwise rejected.

ER_1 (well-structured transactions): ensures specific integrity attributes are executed within a specific t to maintain internal consistency of the system. The relation of (β^k, O) in CR_2 is observed to ensure the β^k of g executed at the same time is unique and the g verified to be valid α_t^k can be operated on the platform.

ER_2 (responsibilities separation): implements an access control policy for S. Associating β^k with S follows the (S, (β^k, O)) relation, clear user S performs an operation β^k on data O and still complies with ER_1, meaning g is guaranteed to perform unique β^k. In addition, TP and IVP operations of the g are performed using different S to maintain the system external consistency.

ER_3 (identity authentication): before performing each TPs, the real identity of S should be clear. Verify the identity of S that performs β to give S permission to the next step, thereby controlling α_t^k changes to the g. For instance, the entity that controls the stored content cannot participate in the authentication process.

ER 4 (initial authorization): grants S the permission to perform β. Initialize the authorization association before the system runs, and S cannot perform any β operation without authorization. The (S, (β^k, O)) relation in ER_2 can be defined or modified if and only if S has the verification authority and does not have the ability to perform the operation; that is, S can prove the correctness of β but cannot perform the TPs. The (S, (β^k, O)) relation in ER_2 can be defined or modified, only when S meets the authentication permission without the operation function, and in other words, only S proves the correctness of β but cannot perform TPs.

3.2.4. Encryption Protocol. In cryptography, the commonly used hash function H provides integrity guarantee by mapping any binary input into a fixed length, compressed version of the code, and the output is called hash value (see the specific implementation principles and common standards in [31]). This class of functions has [32] the following:

(i) *Collision Resistance*. Suppose there are two different trades β_1 and β_2, and the trade is Info_β and almost impossible to find two different variable values that satisfy condition H (Info_1) = H (Info_2). Even if the input variable changes slightly, the output H (Info_β) will be different.

(ii) *Irreversible Primordial Image*. Function H is unidirectional, i.e.,

$$\text{Info}_\beta = > H\left(\text{info}_\beta\right) \cap H\left(\text{info}_\beta\right) \neq > \text{info}_\beta, \tag{3}$$

(iii) *Problem-Friendly*. Introducing a prefix random variable v, make the function H (v|| Info_β), to a harder to guess, which can reflect the function of the corresponding value and is difficult to break out of the news Info_β. The hash function H(v|| Info_β), $v = T$.

Embedding the hash function value into a Merkle tree is a common way for blockchains to protect data integrity. As shown in Figure 2, a Merkle tree has I levels and j columns and at most 2^i leaves $L_0, \ldots, L_{N-1}$, at most $\sum_0^i 2^i = 2N - 1$. Layer I has 2i nodes, which are $(j, 0) \ldots, (j, 2^i - 1)$. The hash value of each leaf node is

$$\begin{aligned} y_{(i,j)} &= h\left(L_N\right) \\ &= h\left(T \| \text{info}_\beta\right). \end{aligned} \tag{4}$$

The path hash value is

$$y*_{(i,j)} = (y(i+1, 2j, y)(i+1, 2j+1)). \tag{5}$$

Most nodes store part of the tree structure, not the entire content, meet the condition of verifying the integrity state by the value of the root hash $y*_{(0,0)}$, and greatly save the storage space. If an S only knows the value of N (3,6) filled with yellow dotted line box, it only needs to request the complementary node N (3,7) value from other nodes, calculate by using hash function, combine with the complementary node of the grey solid wire box on the upper layer, finally calculate the root node value, and compare it with the real value, to independently verify whether the other node is honest.

3.3. CL-BC Components. Due to the access control mode of nodes, blockchain is divided into unlicensed blockchain and licensed blockchain. Nodes can join and exit unrestricted and trade through mining, and free nodes mine to generate transactions, which can be recorded on the blockchain when more than half of the nodes validate the transaction. The latter includes the consortium chain and private chain, and adds an authorization mechanism. Only authenticated nodes can join the network to participate in subsequent transactions, which is more suitable for application scenarios with high requirements on security and integrity. This article uses the licensed blockchain as the infrastructure and then describes the blockchain part of the CL-BC model, including transactions and smart contracts.

3.3.1. Transaction Structure. To maintain the complete and continuous execution of rule 3.2.3, that is, to ensure each transaction is recorded in the system in a complete and correct order, this article implements the integrity process from transaction β to packaged block with the transaction pool. Transaction pool is used in a nonpermissive chain, equivalent to a temporary cache to store transactions to be executed. A permissive chain, like a read-write set generated by an endorsement node to execute a proposal simulation, contains transactions not formally recorded on the blockchain. Once the consensus is reached and recorded on the blockchain, the β is removed from the trading pool. This article takes the license chain as an example, and its structure is shown in Figure 3.

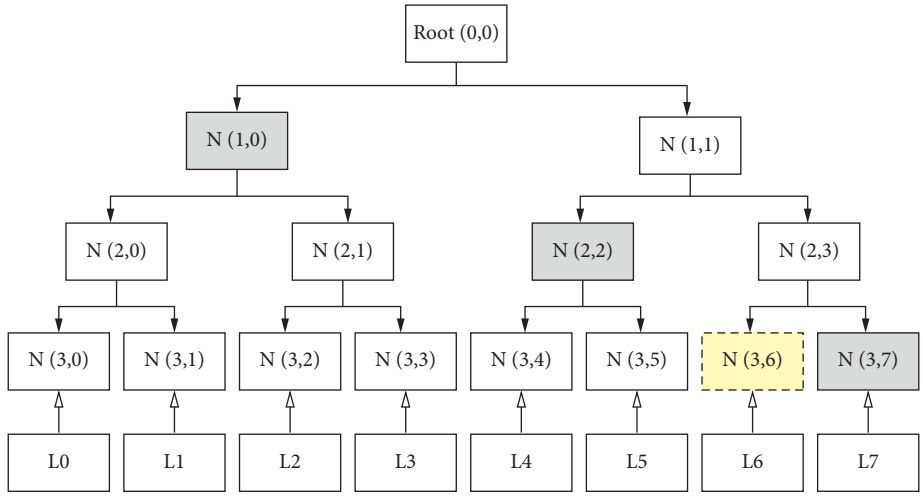

Figure 2: Merkle tree structure.

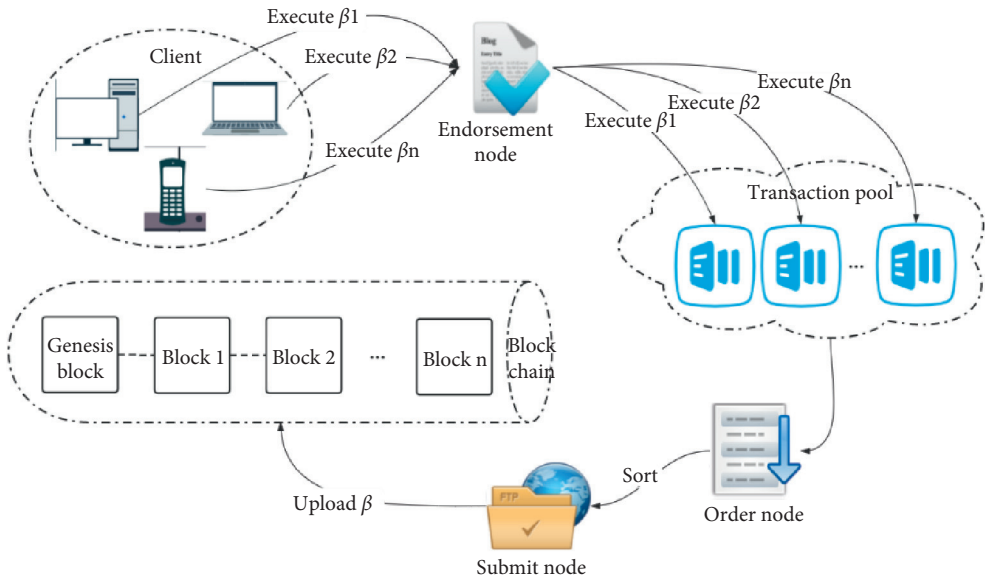

Figure 3: Trading pool structure.

3.3.2. Transaction Process. The rules described in Section 3.2.3 are assembled to form a simple CL-BC structure and integrity protection implementation process, and the specific process is shown in Figure 4. Each step can be instantiated in the model as follows:

(i) Steps 1–3 to verify the current state α_t^k.

(ii) When the conditions to continue trading β are met, a fork option corresponds to two types of data items: G$\in c^k$, the data audit operation and resource query operation in the corresponding instance. However, g$\in u^k$ follows the CR_5 rule, tps (u^k)-> c^k or rejected, corresponding to the data store operation.

(iii) After knowing the following operations, first assign identity permission to S applying for the transaction. ER_3 verifies the identity authenticity of S, ER_4 resets the permission of the operation to be performed, ER_2 controls the specific permission provisions and associates the specific transaction β^k with the executor S, and follows (S, β^k, O); it is clear that S performs β^k on one O. Even on the same occasion, different stages of program execution correspond to different transactions β and S permissions. A practical case, as shown in Table 1, explains that the reason permission needs to be constantly reassigned and also better realizes the function of responsibilities separation; that is, the S of storage and audit cannot be the same one. The row represents principal S, columns express transaction β, and Y/N indicates whether it has permission to perform the current operation.

(iv) Start execution β, ensuring a S executes a particular program O.

(v) Revalidate to ensure internal consistency.

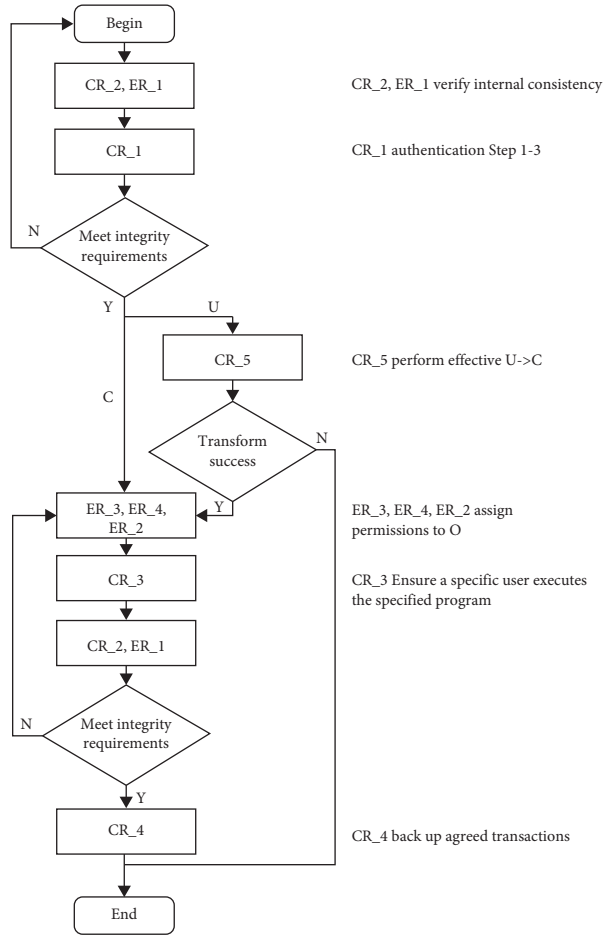

Figure 4: Flowchart.

(vi) CR_4 records the β process and stores in C to ensure the agreed transaction α_{t-}^{k}-> α_{t+}^{k} can be traced.

3.3.3. Smart Contract Format. Smart contract was first put forward by Nick Szabo [33]. Many kinds of contract terms (such as liens, bonds, and defining property rights) are embedded in the hardware and software that we deal with at first, gradually forming an automated "transaction agreement that executes contract terms by computer." Its design conditions should meet four basic principles: (i) observability, Ability to supervise, the contract itself and its implementation can be observed by the relevant organization; (ii) objective verifiability, Actions that have been performed or breached can be examined in subsequent investigation records; (iii) relativity, The execution of contract content has relative scope, and the limitation is different in different situations; Even in the same transaction process, different process stages of the user with different permissions to carry out the execution principal qualification; and (iv) enforceability, Minimize the number of steps required during execution.

Smart contract technology can reduce administrative costs and human risk, and accelerate business progress. In practical applications, smart contract technology is realized based on blockchain technology (see [34] for specific methods, latest development, and related platforms).

In this article, smart contract is the carrier of integrity control, and CL-BC logic rules are written in it by computer language. Contracts are defined, deployed, and invoked for different execution operations. The contract is mainly defined as follows:

Initiate (): used to initialize data g and configure execution permissions.

StoEvidence (): the β_{Store} executed by the filter S is stored in the blockchain, and relevant records in the transaction pool are deleted when the transaction records meeting the conditions are in the blockchain.

Auth (): the validation management phase of the β_{Store} returns the RESP value to StoEvidence () after successfully passing the validation phase.

4. The Deployment Process

This section describes the core transaction β of the CL-BC model in the data storage application. The key process in the deployment process is as follows:

Table 1: Responsibilities separation instances.

S	Data storage	Consensus review	Data query
A	Y	N	Y
B	N	Y	Y

(i) *Authorization Initialization Phase*. Smart contract and initial access domain of g are deployed, and S can store g in the access domain according to its own needs.

(ii) *Data Management Stage*. According to the g at different stages of β, the access permission management is updated in real time for S. S is not allowed to participate in two stages; that is, authentication S and storage S are not allowed to be the same one.

(iii) *Validation Phase*. Verification integrity stage; timely verify whether A_T is effective.

(iv) *Resource Query Management*. Query the information on the chain.

4.1. Authorization Initialization. In the initialization phase, the default access region Acc_x is first generated, and then, the system manager $\mathrm{m} \in \mathrm{S}$, as a trusted subject, adds the g to the region and deploys smart contract to initialize the Acc_x authorization. Perform the following steps:

(i) m selects the access domain identifier ID and chooses the $g \in G$ to join the region.

(ii) Grant corresponding permissions to S, namely, associated g and S.

(iii) Generate a public key pair in each access control domain for signature and verification.

(iv) m places the compiled smart contract in Acc_x and configures a fixed Add for the process to call.

(v) As described in Algorithm 1, an authorized Genesis Block is generated through the initialization program, responsible for the behaviour control of S, where m_{sig} is derived from the private key generated by m:

$$\mathrm{sig}\left\{\mathrm{Acc}_x, \mathrm{ID}_{\mathrm{Acc}}, \mathrm{m}, \mathrm{y}*_{(0,0)}, \mathrm{ACL}, \mathrm{Add}, \mathrm{Stamp}\right\}_{\mathrm{PK_m}}, \quad (6)$$

where the access control list (ACL) is similar to the second-order array in Table 1, and limited S can execute O through the β. Stamp proves that the transaction existed before the signature at time t.

In Algorithm 1, the content classification of G is defined as U and C, and then the (S, (β^k, O) rules followed in CR_2 and ER_3 are written to the ACL to complete the authorization proposal InitAuth () of the current transaction (lines 1–4). Next, through the consensus protocol implementation, internal consensus decision results are obtained (5–10), following the principle of responding nodes exceeding $(f+1)/3f+1$ in PBFT (f refers to the maximum of faulty nodes) [35]. If the agreement passes, the Genesis Block is generated and the world status is updated (11–15); otherwise, the transaction fails.

4.2. Storage Management. After the system is initialized, start β. The process is as follows:

(i) S downloads Acc_x private key and also generates a signature private key with identity representative.

(ii) S sends a transaction request ASK to Acc_x, and private key $\mathrm{PK_S}$ signature is attached to the request to facilitate the verification of real identity:

$$\mathrm{ASK}\left\{\mathrm{Info_S}, \mathrm{Add}, \beta_{\mathrm{Store}}, \mathrm{h}\left(t|\mathrm{Info}_\beta\right), \mathrm{T}\right\}_{\mathrm{PK_S}}. \quad (7)$$

The information $\mathrm{Info_S}$ of S and the storage deal β_{Store} coordinate with the system to judge whether S has the qualification to execute the current transaction.

(iii) Verify identity information to determine whether the β_{Store} can be successfully executed, as shown in Algorithm 2. First, verify the identity of S. If it is true, β_{Store} will be temporarily placed in the trading pool (1–3). Then, Algorithm 3 is called to obtain the authentication response Resp of the transaction request ASK. Only the result is correct, the world state is updated, and the verified β_{Store} in the transaction pool is deleted (5–9). Otherwise, close the deal (10–12).

4.3. Verification Management. This section screens transactions from $\{\beta\}$ in the trading pool can be successfully linked, drawing on the consensus idea of Fan et al. [36]:

(i) Package and encapsulate the disorderly trading β_i within the time zone t_1 to t_2 in the trading pool into blocks.

(ii) Verify the sorted transaction $\{\beta_{\mathrm{Sorted}}\}$ instead of the content Info_β to get the response at lines 6–10 in Algorithm 2. The verification process is shown in Algorithm 3. First, verification node calculates the hash value of the root node (1–3). Noting that when facing the same O, the verification node and the commit storage node cannot overlap. When the number of failed nodes is within the allowed range and the count of obtaining the same value y(str) from other nodes reaches n-f, it indicates the β is valid, and $\{\beta_{\mathrm{Sorted}}\}$ is valid (6–7). Otherwise, the integrity verification results are judged by verifying the root hash of the Merkle tree (8–14). When the response node reaches $f+1$, download and decrypt the private key for β_{Store}; the hash value of $\mathrm{H}(t \,||\, \beta_{\mathrm{Store}})$ will be obtained and the root value of $y^*_{(0,0)}$ will be calculated.

4.4. Resource Query Management. S queries the contents stored on the chain and submits query transaction β_{query} to the system through identity authentication [37]. When it is

```
Input: Acc_x, ID, {c^1,..., c^k}∈C, {u^1,..., u^k}∈U, ACL
Output: Genesis Block
(1) def Acc_x ← {u^1,..., u^k},{c^1,..., c^k};
(2) def ID_Acc ← ID;
(3) def ACL ← (S, (β^k, O));
(4) InitAuth ← Package(ID_Acc ||ACL|| m_sig);
(5) for u in range(Acc_x ⟶ ∑_0^i 2^i) then
(6)    Response ← N_(i,j).GetValid(InitAuth);
(7)      if Response = = True then
(8)      Valid_n ← Valid_n + 1;
(9)    end if;
(10) end for;
(11) if Valid_n < = (Acc_x ⟶ ∑_0^i 2^i) * 2/3 + 1/3 then
(12)    return False;
(13)    else then
(14)    Generate GenesisBlock;
(15)    Broadcast event UpdateWorldState(GenesisBlock);
(16) end if;
```

Algorithm 1:Initiate ().

```
Input: ASK
Output: DecisionResult
(1) def β_Store ← ASK;
(2) if PK_S ← True;
(3) Stored TradPool(β_Store);
(4) if Add ∈ ACL then
(5)    Resp ← Auth();
(6)    if Resp = = "YES" then
(7)    return string("Store successful!");
(8)    Broadcast event UpdateWorldState(β_Store );
(9)    {β} ← {β} − β_Store;
(10)    else if Resp = = "NO" then
(11)    return string("The request failed.");
(12)    else return string("No response!");
(13)    end if;
(14)   else then
(15)    return string("ERROR");
(16) end if;
```

Algorithm 2:StoEvidence().

```
Input: β_Store
Output: Resp
(1) def {β_Sorted} ← Sort(β_(t1,t2));
(2) def y^t_(i,j) ← h(t | β^t_i);
(3) def y*_(0,0) ← derived from (5);
(4) Broadcast event Verify (β_Store);
(5) for Ans_i in range (1, f+1)
(6) if y (str)^n > = n-f then
(7) Resp = = "YES";
(8) else if y (str)^n > = f+1 then
(9) Verify node download SK_S;
(10) Get(H(t|| β_Store)) ← Decrypt(SK_S);
(11) if H(t || β_Store)) = = y*_(0,0);
(12) Broadcast event y() ← y(y*_(0,0));
(13) Resp = = "YES";
(14) else Resp = = "NO";
(15) end if;
(16) end for;
```

Algorithm 3:Auth().

confirmed that S has the query conditions, the system finds the contents according to the index address Add.

5. Experiment and Analysis

5.1. Experimental Result. The simulation was carried out on a laptop equipped with 2.30 GHz Inter Core i7 CPU and 8.00 GB memory, and ubuntu18.04 was configured on Windows WSL2. Virtual nodes are deployed through a Docker container [38], which are mainly divided into Peer and Orderer. Open a main console as a client for β_{Store} and uploading the actual contents Info_β. Peer node verifies transaction validity; Orderer as a sort node and a transaction node, sorts and blocks β, verifies sorted $\{\beta_{\text{Sorted}}\}$, and broadcasts $\{\beta_{\text{Sorted}}\}$ determined to be valid.

Since the time to execute a trade is related to system performance, we only consider the time comparison between different operations in the same environment. Take 50 operations and calculate the average value. The system automatically responds every 3s. Figure 5 shows the system response time when there is no operation under this configuration, with an average value of 68.45 μs and times as the reference time.

Then, we simulated four different types of operations in groups, recorded the time required to determine the type of operation each time, and took the mean value to draw the result as shown in Figure 6. The abscissa represents different operation types, and the ordinate represents time cost.

In this model, maintaining data integrity is mainly reflected in (i) preventing unauthorized users from accessing data items, (ii) preventing unauthorized users from improperly modifying data items, and (iii) maintaining the feasibility of normal operation by authorized users, where (i) corresponds to "unreadable" in Figure 6, (ii) corresponds to "nonwritable," and (iii) corresponds to "readable" and "writable."

In Figure 6, the three bars on the left are close in height, whereas the bar on the right is much higher than the former. This is because of write operations on the blockchain, more operations described in 3.3.1 need to be performed, resulting in a longer response time. The response time of the system to reject malicious operation is even shorter than part of the normal operation time, so the integrity protection efficiency is high.

The effectiveness of the proposed CL-BC model is proved by the application of blockchain. The feasibility and controllability of the model are verified by setting different operation parameters.

5.2. Security Analysis. The data storage system in a secure access control environment to prevent β from being maliciously or accidently destroyed. Considered evaluation metrics include confidentiality, integrity, and availability. Confidentiality guarantee only opens permissions for authorized users, and integrity guarantee does not have permission to change data, and availability ensures that the needs of authorized users are met.

(i) In this model, we have the following:

(1) The irreversible preimage of the hash function guarantees the unidirectional data encryption of the system, namely, $\text{Info}_\beta \Rightarrow H(\text{Info}_\beta) \cap H(\text{Info}_\beta) \not\Rightarrow \text{Info}_\beta$.
Proof. It is difficult to reverse the original text in limited time. Take AntminerS17+, the Bitmain mining machine with the highest computation power of 73TH/s; for example, it performs $73 * 10^{12}$ hashes per second, and the annual hash operation is $73 * 10^{12} * 60 * 60 * 24 * 365 \approx 2 * 10^2$ times. Select the output 128-bit MD5 function for cracking, $2^{128} \div 2 * 10^{21} \approx 10^{18} >> 2.5\text{GY}$, far more than the maximum time unit cosmic year GY.

(2) The prefix of $H(T \| \text{Info}_\beta)$, the problem of friendliness, ensures S is difficult to map the function without knowing the original text.
Proof. Info_β cannot predict the hash value, although the pre-encryption information is known. Measured in femtosecond "ps," the smallest time unit in Verilog. One-minute time difference produces $6 * 1016$ prefixes and makes guessing hashes more difficult.

(ii) Smart contract technology and cryptography technology jointly maintain the system integrity:

(1) The anticollision property of hash function makes it easy to verify the β integrity in this system.
Proof. The output $\text{H}(\text{Info}_\beta)$ is different even if the input variable changes slightly, so when $\beta_1 \neq \beta_2$, it is almost impossible to find two different variable values that satisfy $H(\beta_1) = H(\beta_2)$. The output value is usually a fixed value, the range of β is large, and $H(\beta)$ is small in a limited range. At this time, MD5 is still selected to output 128 bits, and a round of hashing output needs 2^{128} times. The mining machine with 73TH/s computing power needs $2^{128} \div 2 * 10^{21} \approx 10^{18}$ years and has not calculated the prefix time, so it is almost impossible to forge a transaction with the same hash value in the limited time. Easy to tell if the data are complete by looking at the output hash value.

(2) Merkel root judges the integrity of transactions in the system.
Proof. Described in 3.2.4 and in line 8–14 of Algorithm 3 in 4.3, only the root hash value $y^*_{(0,0)}$ or the value of a node and its complementary node can be calculated, and then, the output value of the transaction can be compared with it.

(3) The improved PBFT algorithm to verify the integrity of β more efficiently.
Proof. In the process of verifying β, the conventional PBFT algorithm meets the $(f+1)/3f+1$ principle followed by the number of failed nodes; that is, all the responses are compared for n times to reach a conclusion. When the nodes

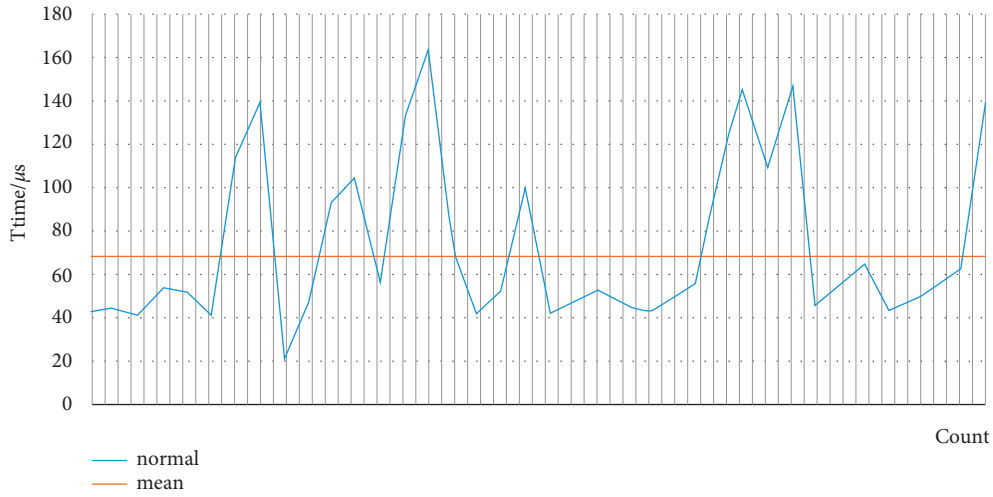

Figure 5: Response time and mean value without operation.

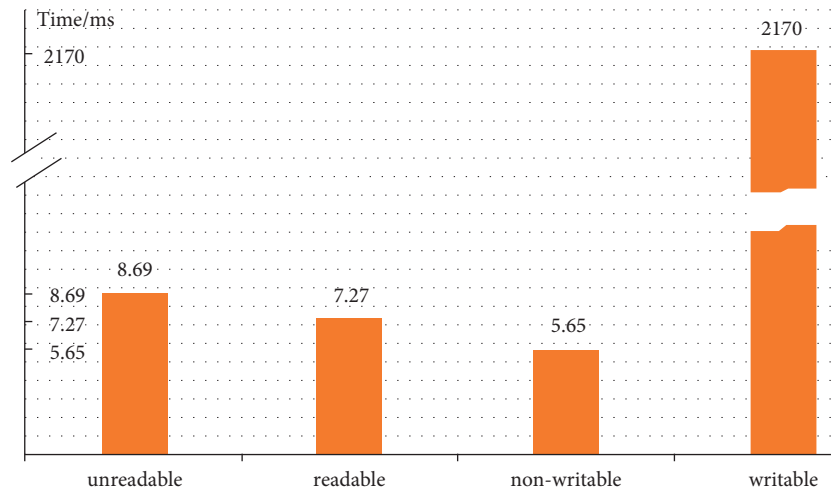

Figure 6: Response time for different operating states.

Table 2: Scheme comparison.

Performance	Ordinary storage	Clark–Wilson storage	CL-BC data storage
Distributed	N	N	Y
Identity authentication	Y	Y	Y
Auditability	N	Y	Y
Expandability	N	N	Y
Resist collusive attacks	N	N	Y
Data tampering	Easy	Hard	Hard

accept the same function value for many times, n-f times should be compared to judge the transaction integrity. When multiple nodes receive the same string, compare $f+1$ times to reach a consensus.

(iii) To improve the data storage system model, various CL-BC integrity protection rules are formulated in 3.2.3, and a coherent execution process for implementing these rules is introduced in 3.3.2. The complete transaction structure is described in 3.3.1, and the smart contract deployed according to the rules and transaction structure (see Section 4) is mandatory to automate the operation, effectively prevent all kinds of hardware and software failures and human failures, and better maintain the availability of the system.

5.3. Comparison with Existing Schemes. The storage model is extremely concerned with the private information control; however, a more important goal in this domain is to maintain data integrity to ensure transactions are authentic and valid. The Cl-BC model is an integrity-based access control security model and has the advantages of dispersibility, auditability, and immutability. By comparing with existing studies, the results are shown in Table 2. Compared with other schemes, the biggest advantage of CL-BC is that it has a distributed structure and can resist collusion attacks [39], to avoid the deception occurrence driven by private interests of traders, crucial to the maintenance of data integrity.

6. Conclusion

In this article, we propose a security data storage model CL-BC for social network to protect data integrity. We carry out conceptual design from the basic elements, security attributes, relevant rules, and encryption parts of the model, and plan the optimal data storage architecture and design detailed processes at each stage. Part of the DR-BFP algorithm is used to design and deploy blockchain smart contracts to improve data integrity. The experimental results and security analysis show that the model is better in integrity protection and has better verification function.

Data Availability

This research and proposed methodology were simulated, and all data are included already in the paper. So, there is no need for extra data.

Conflicts of Interest

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

Acknowledgments

This research was supported by the Scientific Research Project of the Education Department of Jilin Province (Grant no. JJKH20220602KJ).

References

[1] J. Chirillo and S. Blaul, *Storage Security Technology*, pp. 3–280, Publishing House of Electronics Industry, Beijing, China, 2004, in Chinese.

[2] S. Teerakanok and T. Uehara, "Copy-move forgery detection: a state-of-the-art technical review and analysis," *IEEE Access*, vol. 7, 2019.

[3] S. Mostafavi, V. Hakami, and M. Sanaei, "Quality of service provisioning in network function virtualization: a survey," *Computing*, vol. 103, pp. 917–991, 2021.

[4] M. Govindarajeswaran and A. J. A. Jeyarani, "Method and system for efficient execution of ordered and unordered tasks in multi-threaded and networked computing," pp. 1–14, 2017.

[5] A. Tellabi, J. Sassmanhausen, E. Bajramovic, and K. C Ruland, "Overview of authentication and access controls for I&C systems," in *Proceedings of the 2018 IEEE 16th International Conference on Industrial Informatics (INDIN)*, pp. 882–889, Porto, Portugal, July 2018.

[6] X. Yu, Z. X. Shu, Q. Li, and J Huang, "BC-BLPM: a multi-level security access control model based on blockchain technology," *NETWORKS & SECURITY*, vol. 18, no. 2, pp. 110–135, 2021.

[7] G. Liu, J. Zhang, J. H. Liu, and Y Zhang, "Improved Biba model based on trusted computing," *Security and Communication Networks*, vol. 8, no. 16, pp. 2793–2797, 2015.

[8] R. Mahalingam, "A blockchain based framework for information system integrity," *Blockchain Technologies And Applications*, vol. 16, no. 6, pp. 9–25, 2019.

[9] S. Fehis, O. Nouali, and M. T. Kechadi, "A new distributed Chinese Wall security policy model," *Journal of Digital Forensics, Security and Law*, vol. 11, no. 4, pp. 149–168, 2016.

[10] Y. Chen, H. Xie, K. Lv, S Wei, and C Hu, "DEPLEST: a blockchain-based privacy-preserving distributed database toward user behaviors in social networks," *Information Sciences*, vol. 501, pp. 100–117, 2019.

[11] D. D. Clark and D. R. Wilson, "A comparison of commercial and military computer security policies//1987 IEEE symposium on security and privacy," *IEEE*, vol. 1, pp. 27–29, 2014.

[12] R. A. Haraty, M. F. Naous, and A. Mourad, "Assuring consistency in mixed models," *Journal of Computational Science*, vol. 5, no. 4, pp. 653–663, 2014.

[13] M. Ramkumar, "A blockchain based framework for information system integrity," *Communications, China*, no. 6, pp. 1–17, 201916.

[14] T. Tsegaye and S. Flowerday, "A Clark-Wilson and ANSI role-based access control model," *Information and Computer Security*, vol. 28, no. 3, pp. 373–395, 2020.

[15] F. Avorgbedor and J. Liu, "Enhancing User Privacy Protection by Enforcing Clark-Wilson Security Model on Facebook," in *Proceedings of the 2020 IEEE International Conference on Electro Information Technology (EIT)*, pp. 155–161, IEEE, Chicago, IL, USA, August 2020.

[16] X. L. Cao, J. H. Zhang, and B. Liu, "A review of blockchain security, privacy, and performance issues," *Computer Integrated Manufacturing Systems*, vol. 27, no. 7, pp. 1–26, 2021, in Chinese.

[17] Q. Xia, J. Lv, X. Meng, and S. Ma, "Distributed user trace collection and storage system," *Journal of Beijing University of Aeronautics and Astronautics*, vol. 46, no. 3, pp. 548–562, 2020, in Chinese.

[18] Y. Xiao, N. Zhang, W. Lou, and Y. T Hou, "A survey of distributed consensus protocols for blockchain networks,"

IEEE Communications Surveys & Tutorials, vol. 22, no. 2, pp. 1432–1465, 2020.

[19] F. Rezaeibagha and Y. Mu, "Efficient micropayment of cryptocurrency from blockchains," *The Computer Journal*, vol. 62, no. 4, pp. 507–517, 2018.

[20] A. Musamih, K. Salah, R. Jayaraman et al., "A blockchain-based approach for drug traceability in healthcare supply chain," *IEEE Access*, no. 9, pp. 9728–9743, 2021.

[21] Q. Zhang, Y. Han, and Z. Su, "A storage architecture for high-throughput crop breeding data based on improved blockchain technology," *Computers and Electronics in Agriculture*, vol. 173, no. 6, pp. 1–14, 2020.

[22] M. A. Hasan, M. Tariq, Z. Chen et al., "A digital rights management system based on a scalable blockchain," *Peer-to-Peer Networking and Applications*, vol. 14, pp. 2665–2680, 2021.

[23] L. Wang, "Research and Design of Will Security Deposit System Based on Blockchain Technology," *Hefei: University of Science and Technology of China*, pp. 1–60, 2019, in Chinese.

[24] X. Han, Y. Yuan, and F. Y. Wang, "Security problems on blockchain: the state of the art and future trends," *Acta Automatica Sinica*, vol. 45, no. 01, pp. 206–225, 2019, in Chinese.

[25] L. J. Helsloot, G. Tillem, and Z. Erkin, "BAdASS: preserving privacy in behavioural advertising with applied secret sharing [J]," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications(JoWUA)*, vol. 10, no. 1, pp. 23–41, 2019.

[26] Y. Lee, B. Son, S. Park, J. Lee, and H. Jang, "A survey on security and privacy in blockchain-based central bank digital currencies," *Journal of Internet Services and Information Security (JISIS)*, vol. 11, no. 3, pp. 16–29, 2021.

[27] L. Zhu, Y. Wu, K. Gai, and R. C Kim-Kwang, "Controllable and trustworthy blockchain-based cloud data management," *Future Generation Computer Systems*, vol. 91, pp. 527–535, 2018.

[28] P. W. Khan, Y. C. Byun, and N. Park, "A data verification system for CCTV surveillance cameras using blockchain technology in smart cities," *Electronics*, vol. 9, no. 3, pp. 484–505, 2020.

[29] S. Mercan, M. Cebe, R. S. Aygun, and K Akkaya, "Blockchain-based video forensics and integrity verification framework for wireless Internet-of-Things devices," *Security and Privacy*, pp. 1–17, 2020.

[30] H. Wu, Z. Yu, D. Huang, H Zhang, and W Han, "Automated enforcement of the principle of least privilege over data source access," in *Proceedings of the 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications*, pp. 510–517, TrustCom), Guangzhou, China, January 2020.

[31] A. Zeyad, M. Ali, A. Abbas, A Abbas, and S. U Khan, "Secure hash algorithms and the corresponding FPGA optimization techniques," *ACM Computing Surveys*, vol. 53, no. 5, pp. 1–36, 2020.

[32] J. Zou, H. Zhang, Y. Tang, and L. Li, *Blockchain Technology Guide*, pp. 274–318, China Mechanical Engineering Press, Beijing, China, 2016.

[33] N. Szabo, "Smart contracts: building blocks for digital markets. Extropy: the journal of trahumanist thought," 1996, http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart_contracts_2.html [2020-08-20].

[34] Z. Zheng, S. Xie, H. N. Dai et al., "An overview on smart contracts: challenges, advances and platforms," *Future Generation Computer Systems*, vol. 105, pp. 1–16, 2019.

[35] W. Li, C. Feng, L. Zhang, H Xu, and B Cao, "A scalable multi-layer PBFT consensus for blockchain," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 5, pp. 1146–1160, 2020.

[36] Y. Q. Fan, H. Y. Wu, and H. Y. Paik, "Dr-Bft: A consensus algorithm for blockchain-based multi-layer data integrity framework in dynamic edge computing system," *Future Generation Computer Systems*, vol. 124, pp. 33–48, 2021.

[37] X. Dong, M. Li, Y. Du, and Q. Wu, "A biometric verification based authentication scheme using Chebyshev chaotic mapping," *Journal of Beijing University of Aeronautics and A*, vol. 45, no. 5, pp. 1052–1058, 2019, in Chinese.

[38] J. Long, H. Y. Tai, S. T. Hsieh, and J. M Yuan, "A lightweight design for high-performance serverless computing," *IEEE Software*, pp. 99–103, 2020.

[39] H. Li, L. Shen, Y. Wang, J Feng, H Tan, and Z Li, "Risk measurement method of collusion privilege escalation attacks for android apps based on feature weight and behavior determination," *Security and Communication Networks*, vol. 2021, no. 11, pp. 1–18, 2021.