Hindawi
IET Information Security
Volume 2024, Article ID 6616095, 21 pages
https://doi.org/10.1049/2024/6616095

Research Article

A Novel Security Scheme Supported by Certificateless Digital Signature and Blockchain in Named Data Networking

Bing Li,[1] **Mingxuan Zheng**,[1] **and Maode Ma**[2]

[1] *Shenzhen University, Shenzhen, China*
[2] *College of Engineering at Qatar University, Doha, Qatar*

Correspondence should be addressed to Mingxuan Zheng; zhengmingxuanww@163.com

Received 4 August 2023; Revised 7 January 2024; Accepted 17 February 2024; Published 12 March 2024

Academic Editor: Youwen Zhu

The publication of this article was funded by Qatar National Library.

Named Data Networking (NDN) is a promising network architecture that differs from the traditional TCP/IP network, as it focuses on data rather than the host. A new secure model is required to provide the data-oriented trust instead of the host-oriented trust. This paper proposes a new secure solution in the NDNs named Secure Mechanism supported by Certificateless Digital Signature and Blockchain (CLDS-B). The CLDS-B scheme employs a certificateless digital signature to guarantee the authentication and integrity of data. On the one hand, the key escrow problem has been solved to eliminate the risks of compromised private key generators; on the other hand, the data name has been bound to the public key to prevent the false public key. Moreover, the blockchain is used to manage cryptographic information. Each domain designates an information service entity to join the blockchain so that the consumer could retrieve the cryptographic information public parameter in the local domain if necessary. Furthermore, due to the decentralization of the blockchain, the CLDS-B would be robust to resist the single-node failure. Simulation results show that the CLDS-B scheme outperforms a classic NDN scheme, although it shows slightly inferior to the other secure NDN scheme. The security verification and analysis show that the CLDS-B would resist the key escrow attack. The CLDS-B would be a competitive solution in scenarios with a high-security level.

1. Introduction

As a candidate for future Internet, Named Data Networking (NDN) [1] is based on a data-centric architecture. Different from the traditional IP networks, the data in the NDNs would be accessed by data names at the network layer rather than host addresses. Moreover, in NDNs, routers are equipped with large memories to cache the forwarding data. If the request for cached data arrives later, routers would reply to the request directly with the cached data instead of forwarding it to the data sources. As a result, the load of the data sources would be relieved and the delay of the reply would become smaller in NDNs. Moreover, the overhead of the network would be reduced [2].

The data-centric architecture implies that NDNs have new security requirements to protect data rather than the communication channels in the IP networks. The data-oriented security mainly includes data authentication, data integrity, data confidentiality, and data privacy [3, 4]. Data authentication and data integrity are our focuses. Data authentication requires that the received data must be produced by the authenticated data sources no matter which network entity data are replied by; and data integrity requires that the received data must not be tampered with by others after they are produced by the data sources. IP networks use the digital signature to provide security services of the authentication and integrity for the communication channels. Therefore, similarly, NDNs have incorporated digital signatures to ensure data authentication and integrity. A signature would be generated by the data source to attach to the data to be delivered in the NDNs. As a result, public keys of data sources would be required to be distributed to all the data requesters. It seems that the traditional public key infrastructure (PKI) [5] is a good candidate for public key distribution. In PKI, certificate authorities (CAs) would serve as the trust anchor. It would issue certificates and publish the valid certificates on the website. The certificates bind the identity information and the public key information. All the users who need the information on public keys could download the certificates and then retrieve the public key from the certificates. However, the certificate issue and management are costly and resource-intensive. To eliminate the cumbersome certificate management, the concept of an identity-based cryptosystem (IBC) has been proposed in [6]. In NDNs, IBC would use identity information as the public key directly. Since the identity information is usually assumed to be known by users, there are no additional demands for public key distribution so as to eliminate the need to request and exchange certificates. However, in the context of the IBC, a highly trusted third party is required to generate the private keys, which leads to the problem of key escrow to violate the nonrepudiation. Key escrow involves entrusting the generation and storage of private keys to a trusted third party. However, this method may compromise nonrepudiation, as the third party could potentially deny generating the private key, thus impacting the trustworthiness and traceability of transactions.

Therefore, the concept of certificateless public key cryptography (CL-PKC) has been proposed in [7], which avoids the key escrow problem in the IBC and reduces the complexity of traditional PKI-based cryptosystems. The CL-PKC is a convergence between the traditional PKI and the IBC. On the one hand, the Key Generation Center (KGC), as a trusted third party, would still be employed by certificateless public key systems to generate partial private keys instead of complete private keys. The complete private keys are generated using the partial private keys designated by KGC and the secret values chosen by the key owners. As a result, KGC would just know the partial private key rather than the complete private keys, and only the key owners could know their complete private keys. The key escrow problem is solved by the CL-PKC so that true nonrepudiation would be achieved. On the other hand, the user generates their own public key, and it does not require authentication from a CA. Consequently, the public key does not need a certificate to validate its authenticity. In this sense, CL-PKC should be a better solution to provide the public key for NDNs.

Moreover, trust models for authentication and key management are also important challenges for NDN security. At present, most trust models are based on centralized trust anchors. Traditionally, centralized trust authorities usually hold the authority and trustworthiness to reliably authenticate and verify the user's identity. Moreover, the structures of the centralized trust models are relatively simple, so that their management would be easy. However, the centralized trust models are vulnerable since they may suffer from the risk of single-node failure. The centralized trust authority would become the main subject of the distributed Denial of Service (DDoS) attacks. Once it is compromised, the whole cryptographic system would sustain information leakage and privacy violations. Therefore, decentralized models would be pursued by NDNs. Blockchain, as a decentralized distributed system, should be a solution that can effectively mitigate the abovementioned risks. On the one hand, blockchain can generate and manage cryptographic keys with its decentralized feature, which can solve the possible single-node failure problem and other security problems such as opacity and dependency. On the other hand, blockchain adopts a peer-to-peer architecture to transfer data directly between nodes, reducing the risk of information leakage and tampering. Although certificateless digital signature encryption is a digital signature and authentication technology without a trust center. It still requires some trusted authority to distribute the public keys and identity information. Blockchain technology ensures the security and reliability of public keys while enabling the management of them.

In this paper, we focus on the security mechanisms in NDNs to implement data-oriented authentication and to verify the integrity of Data packets. We propose a blockchain-based certificateless digital signature scheme (CLDS-B) for secure communication in NDNs. In the CLDS-B, the certificateless digital signature algorithm would be employed to sign and verify the data. Since no certificates are required, there is no heavy load of the certificate management. Moreover, the KGC does not know the complete private keys of the data sources, so that there is no risk of the key escrow. On the other hand, the blockchain is used to store the public keys of the data sources. Due to the decentralized structure of the blockchain, a single failure would not cause the trust collapse of the entire network. Meanwhile, this solution maintains a comparable network performance with inspired by the solution in [8], which. However, there are still some security services that need to be improved to fix the key escrow problem. Compared with the solution in [8], the CLDS-B scheme achieves a higher level of security with the ability to solve the problem of key escrow.

The major contributions in this paper could be summarized as follows:

- We propose the novel CLDS-B scheme to improve the security and nonrepudiation of signatures to mitigate some flaws in the AHISM-B scheme, avoiding a potential disaster of forged signatures from key escrow.
- We conduct simulation experiments to evaluate the performance of the CLDS-B and compare it with the classic NDN scheme in [9], the HISM-B scheme in [10], and the AHISM-B scheme in [8].

The rest of this paper is organized as follows: Section 2 describes related work on security mechanisms in NDNs. Section 3 introduces certificateless digital signature and the network model. We then propose the security-enhanced CLDS-B scheme in Section 4. Section 5 provides a security analysis of the CLDS-B scheme. Section 6 formally validates the CLDS-B scheme. Section 7 performs performance evaluation of the CLDS-B scheme and comparison with other schemes. Section 8 concludes the whole paper.

2. Related Work

2.1. Secure Model in NDN. NDNs bring new challenges to traditional security mechanisms. In recent years, several solutions to address these challenges have been proposed, which can be broadly classified into traditional PKI-based encryption schemes and identity-based encryption schemes.

In the literature for NDNs, the solution of PKI-based cryptography has focused on two areas, including public key management and namespaces. Public key management deserves attention due to the widespread use of public key signatures in NDNs. It is argued in [11] that an NDN requires public key certificates as a trusted assertion. Also, it uses well-defined certificate formats and various systems and protocols that support certificate distribution and revocation to authenticate and manage public keys. In addition, several methods for providing certificates and the design of certificate revocation have been discussed. The issuance and revocation of certificates play an important role in securing and trusting data in the NDNs. The certificate revocation problem in NDNs has been explored in [12] with a proposal of a certificate revocation framework named CertRevoke. The framework aims to address the security and efficiency issues of certificate revocation in the NDNs. Specifically, CertRevoke utilizes naming conventions and trust models to ensure the legitimacy of revocation records and to improve efficiency by caching these records in the networks. At the same time, a new idea and approach to certificate management in NDNs has been presented. In addition to public key management, namespace management is an important safeguard to ensure that NDNs can share and transmit data efficiently. The NDN Certificate Management Protocol has been proposed in [13] to manage cryptographic keys and certificates. It uses a namespace to generate certificates and certificates of subnamespaces. Meanwhile, a flexible mechanism to obtain trust between delegated certificates has been designed in [14], which explores two requirements needed to implement trust establishment in NDNs, which are namespace management and public key management. Through a survey of NDN applications, a framework has been derived to systematically evaluate and assess namespace and public key management systems and relate their functionality to Domain Name System Security Extensions and Web PKI. In addition, existing approaches have been compared with the two most prominent implementations currently available on the Internet. A valuable reference for the design and implementation of future NDN applications has been provided. However, although the PKI-based encryption algorithm is an important approach to secure data exchanges in NDNs, there are some disadvantages and limitations, such as the digital certificates need to be verified and distributed with a high management cost. At the same time, the certificate needs to be issued by a third party, and once it fails, the whole communication system is exposed to great risks.

Recently, in the field of authentication and encryption, the applicability of IBC in NDNs has been explored. The requirements of a naming system for NDNs have been defined in [15] to provide security services that bind naming and content. Then, PKI and hierarchical identity-based cryptography are combined to enhance the security of the NDNs. A distributed authentication and authorization scheme (DAAS) has been proposed in [16] that addresses the excessive traffic overhead caused by secure distributed data sharing. The attribute manifest distribution and automatic attribute updates proposed by the DAAS scheme can reduce the cost of retrieval, which is well suited for the NDNs that use data names for retrieval. A signature scheme has been proposed in [17] based on the concept of a hyperelliptic curve's identity that emphasizes the integrity and authenticity of the content. It cannot only protect NDNs from possible content poisoning attacks (CPA) but can also provide the same level of security as the Rivest–Shamir–Adleman (RSA), bilinear pairing, and the elliptic curve cryptosystem (ECC). The access control on the information service entity scheme (ACISE) has been proposed in [18], which is supported by an ISE for NDNs. The IBC is used to generate private keys and signatures for authorized consumers at the ISE. The ACISE scheme is not subject to cache contamination attacks and can maintain a small response latency under attacks. A solution, called AHISM in [8], enables secure communication and data sharing in NDNs, which works based on blockchain. On the one hand, hierarchical identity-based cryptography is used to bind data names to public keys in AHISM. Valid public parameters would be requested by the consumer using the Interest packet so that the consumer could compose the producer's public key to authenticate the producer and verify the integrity of the requested Data packets. On the other hand, a blockchain is used to manage the public parameters to avoid disasters due to the failure of a single node. However, these schemes based on IBC would face the problem of key escrow that violates the nonrepudiation. If the private key generated by a third party is attacked and someone impersonates a third party to forge a signature, consumers would be exposed to security threats.

In summary, the researches on secure NDN schemes have still been challenged by the certificate management and the key escrow. Inspired by literatures [8–10], we would explore the certificateless digital signatures in NDNs for data authentication and data integrity.

2.2. Certificateless Signature. As a suitable alternative to traditional PKI and identity-based digital signature schemes, certificateless signature schemes have gained wide attention from both academia and industry since they get rid of the complex certificate management and prevent the private key generator to know users' private keys. Yeh et al. [19] proposed a certificateless signature scheme specifically designed for Internet of Things (IoT) smart devices. In addition, other researchers [20–22] have also been attempting to provide more efficient certificateless signature methods for data authentication in industrial IoT infrastructure. The certificateless digital signatures, with their outstanding features, have been widely deployed in various practical applications, such as e-healthcare [23, 24] and vehicular ad hoc networks [25, 26].

In practical environments, to meet different business requirements, researchers have combined the certificateless cryptography scheme with other cryptographic primitives to design and propose many certificateless digital signature schemes with special properties. Among them, certificateless schemes based on proxy resignature can solve the problem of long signature chain conversion and uncertainty in trust relationships between communication parties [27, 28].

Certificateless signature schemes with batch verification could simultaneously verify the correctness of a large number of signatures from different signers for different messages, improving the efficiency of signature verification [29, 30]. The integrated digital signature schemes have also attracted much attention [31, 32] in order to support heterogeneous device environments where sensors in industrial IoT environments adopt ID-based cryptographic systems, and smart devices adopt CL-based cryptographic systems. Therefore, the certificate-less signature scheme has broad prospects for development and application scenarios and would become one of the important technological solutions for future digital signatures and identity verification.

2.3. Certificateless Signature in NDN. Recent research has made significant strides in NDN and certificateless signature schemes. In order to address the security challenges in the IoT environment, Huang et al. [33] proposed a certificateless group signature scheme based on mobile edge computing (MEC). It has offloaded signature pressure from the data source to the MEC server to satisfy the resource-constrained IoT devices. Hussain et al. [34] focused on defense against CPA in NDN-based IoT networks. Additionally, Ullah et al. [35] and Rao et al. [36] applied NDN to the healthcare sector. They proposed the NDN-based medical IoT framework that adopts lightweight certificateless signatures and utilizes the hyperelliptic curve cryptosystem to enhance security while reducing costs. These studies collectively advance the security and efficiency of NDN in IoT applications, offering new ideas and solutions for future network security and optimization.

In this paper, we focus on improving the efficiency of certificateless signature schemes applied to the NDNs.

3. System Model

Certificateless digital signature aims to address some of the limitations of PKI-based digital signature schemes and identity-based digital signature schemes. Unlike previous approaches, certificateless digital signature does not require the use of digital certificates issued by trusted third parties, nor does it face the key escrow problem associated with identity-based encryption algorithms. It combines private keys and keys generated by a KGC to solve the trust management issue of keys. Therefore, adopting a certificateless trust management approach has the potential to provide a solution for trust management in NDN.

3.1. Generic Certificateless Digital Signature. In a certificateless signature scheme, there are three legitimate participants: signer, verifier, and KGC. The scheme consists of the following algorithms:

System Initialization: The algorithm is performed by the KGC. Its input is a system parameter k, and its outputs are the system master key s and the system public parameters *params*.

Secret Value Generation: The algorithm is performed by KGC. Its inputs are system parameters *params*, system master key s, and user identity ID, and its output is the user's secret value DA.

Private Value Generation: The algorithm is performed by the signer. Its inputs are system parameters params, user identity ID, and a random value, and its output is the user's private value XA.

Partial Private Key Generation: The algorithm is performed by the signer. Its inputs are system parameters params, user's secret value DA, and private value XA, and its output is the user's partial private key SA.

Partial Public Key Generation: The algorithm is performed by the signer. Its inputs are system parameters *params*, user's private value XA, and its output is the user's partial public key PA.

Signing: The algorithm is performed by the signer. Its inputs are the message m to be signed, the user's identity ID, and the user's partial private key SA, and its output is a signature σ.

Verification: The algorithm is performed by the verifier. Its inputs are the message m, the signature σ, system public parameters *params*, the signer's partial public key PA, and the user's identity ID. If the signature is valid, its output is True; otherwise, its output is False.

3.2. Network Model. The topology of our network is shown in Figure 1, which is composed of the fundamental network devices in NDNs, including producers, consumers, routers, and special servers, named ISEs.

The producers, consumers, and routers in the network are all devices in a standard NDN. The producer would play the role of signer, and the consumer would play the role of verifier. An ISE would play the role of a KGC, which is used to generate the producer's partial private key and ISE's key pair.

The entire network consists of the blockchain network and the NDN. The blockchain network is a multidomain network whose participants are the ISEs designated by each domain. It takes charge of the management of the cryptographic information, including public parameters and partial public keys of producers. In NDNs, producers would generate the signature for every Data packet, and consumers would verify the signature encapsulated in the received Data packet to guarantee the data integrity and data authentication. The partial private key used to signature generation would fetch from ISE, and the public key used to signature verification may retrieve form the blockchain network.

3.2.1. Two Assumptions. In this work, there are two assumptions as follows: (1) There are secure channels between the producers and their ISEs. These secure channels are used to distribute the domain parameters, and producer's secret value to producers, and to register producer's public keys to the ISE. (2) The ISE has distributed the domain public key to all the consumers located at its domain.

3.2.2. Data Naming. The hierarchical structure would be used to name data in our scheme, which consists of three fields, starting with "/." The first field indicates the category to which the name belongs. The second field provides identifier information, and the third field contains data information. Each field is organized hierarchically into one or more

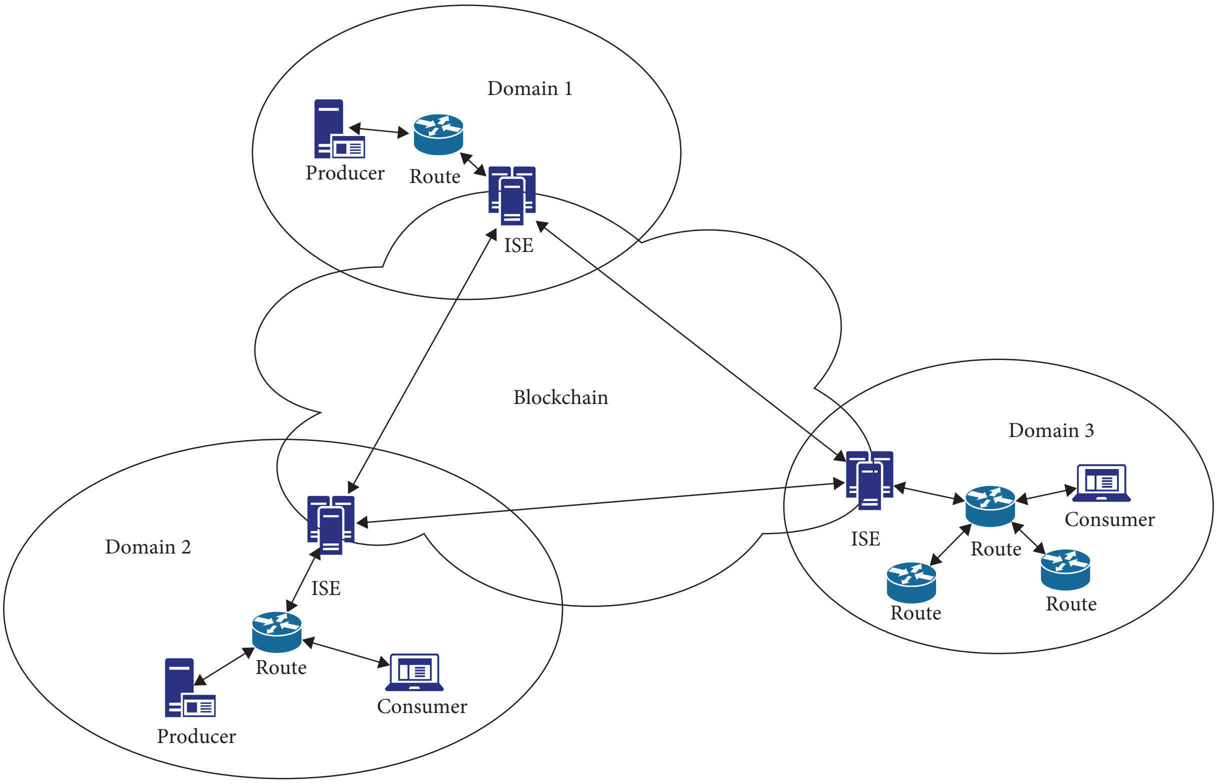


FIGURE 1: Network model.

components separated by a "/." There are two categories of data name: the names whose first fields are "/ndn/Data" and the names whose first fields is "/ndn/Crypto." The former one is used to name the data that are requested by the application at the consumers and are published by producers. Therefore, its second fields are the producers' routing identifiers, i.e., their identities IDs, and its third field usually includes the file name, the file version, and the segment number. The latter one is used to name the data that are managed by the ISEs using the blockchain. Therefore, its second fields are the ISEs' routing identifiers, and the third fields are usually the identities of the producers and the version. The name indicates the data are the cryptographic information of a producer, which is managed by the ISE.

3.3. Blockchain Smart Contracts. In the CLDS-B scheme, the design based on Hyperledger Fabric serves as the underlying infrastructure of the blockchain network. This scheme utilizes blockchain to manage cryptographic information, effectively reducing the computational costs and preventing attacks such as public key substitution and forged signatures.

In this scheme, one ISE would be authorized to join the blockchain network at the blockchain initialization. After that, other ISEs should request to join the blockchain network. The request would be reviewed by ISEs that have already joined the network. Once approved, the ISEs would synchronize block information and would install the chaincode to become a new member in the blockchain. All the ISEs that have joined the blockchain would take charge of announce the cryptographic information for all the producers that locate at the same domain to the ISEs. The cryptographic information includes the producer's identity ID, the producer's partial public key PA, the domain public parameters params, and their validity period. If this announcement is approved by the consensus algorithm, a new block encapsulating the cryptographic information, would be added to the blockchain. Since blocks in the blockchain would be synchronized among all participating ISEs, any ISE can obtain cryptographic information for all producers in any domain from its local blockchain.

This approach effectively utilizes blockchain technology to manage cryptographic information, enhancing system security and trustworthiness. Additionally, the integration with Hyperledger Fabric leverages its mature network infrastructure and smart contract functionality to provide robust support for the certificateless signature scheme (CLDS-B).

3.4. Threat Model. Our network model involves two types of separate adversaries as follows: Attacker $\mathscr{A}_{\mathrm{I}}$ is a malicious signer who cannot obtain part of the legitimate user's private key, and attacker $\mathscr{A}_{\mathrm{II}}$ is a malicious but passive KGC, who cannot obtain the user's secret value and replace the user's public key.

Game I. The game is completed by the interaction between the challenger $\mathscr{C}_{\mathrm{I}}$ and the attacker $\mathscr{A}_{\mathrm{I}}$.

Initialization Phase: input security parameter k, then run the Setup algorithm to generate params and s. $\mathscr{C}_{\mathrm{I}}$ only sends params to $\mathscr{A}_{\mathrm{I}}$.

Table 1: Notations and descriptions.

Notations	Descriptions
Interest packet	The packet that sent by a consumer to request data according to the data name in NDNs
Data packet	The packet that sent by a data provider (including a producer and an ISE) to rely on the Interest packet with the data content
Params	Public parameters used in CL-PKC
ID	Producer's identity
DA	Secret value
XA	Producer's privacy value
SA	Producer's partial private key
PA	Producer's partial public key
PK	Producer's public key
SK	Producer's private key
PK_I	Domain public key
SK_I	Domain private key
σ_p, σ_I	Signature
m	Message

Inquiry Phase:

Secret-Value-Query: $\mathscr{A}_{\mathrm{I}}$ selects an identity ID_i to submit to $\mathscr{C}_{\mathrm{I}}$. When a submitted partial private key query is received, $\mathscr{C}_{\mathrm{I}}$ runs the Secret-Value-Gen algorithm to generate secret value s_i and return s_i to $\mathscr{A}_{\mathrm{I}}$.

Private-Value-Query: $\mathscr{A}_{\mathrm{I}}$ selects an identity ID_i to submit to $\mathscr{C}_{\mathrm{I}}$. When a submitted private value query is received, $\mathscr{C}_{\mathrm{I}}$ runs the Partial-Value-Gen algorithm to generate and return a secret value x_i to $\mathscr{A}_{\mathrm{I}}$. If the partial public key of the signer ID_i is replaced, $\mathscr{C}_{\mathrm{I}}$ terminates this query.

Partial-Public-Key-Query: $\mathscr{A}_{\mathrm{I}}$ selects an identity ID_i and submits it to $\mathscr{C}_{\mathrm{I}}$. When receiving the public key interrogation submitted by $\mathscr{A}_{\mathrm{I}}$, $\mathscr{C}_{\mathrm{I}}$ runs the Partial-Public-Key-Gen algorithm to generate and return the partial public key PA_i to $\mathscr{A}_{\mathrm{I}}$.

Replacement-Partial-Public-Key-Query: $\mathscr{A}_{\mathrm{I}}$ selects an identity ID_i and a new public key to submit to $\mathscr{C}_{\mathrm{I}}$. When receiving the replacement partial public key query submitted by $\mathscr{A}_{\mathrm{I}}, \mathscr{C}_{\mathrm{I}}$ replaces the original partial public key PA_i with the new partial public key PA_i'.

Signature-Query: $\mathscr{A}_{\mathrm{I}}$ selects an identity ID_i and message m to submit to $\mathscr{C}_{\mathrm{I}}$. Upon receiving the signature query submitted by $\mathscr{A}_{\mathrm{I}}$, $\mathscr{C}_{\mathrm{I}}$ runs the Sign algorithm to generate the signature σ_i of message m and returns σ_i to $\mathscr{A}_{\mathrm{I}}$. Here, σ_i satisfies VALID $\leftarrow$ $Verify\,(Params, m, \sigma_i, PA_i', ID_i)$, where PA_i' is the current partial public key, and this partial public key can be the partial public key after $\mathscr{A}_{\mathrm{I}}$ replaces it.

Forgery Phase: $\mathscr{A}_{\mathrm{I}}$ outputs $(ID_i^*, m^*, \sigma_i^*)$, where ID_i^* is the target user that $\mathscr{A}_{\mathrm{I}}$ chooses to forge the signature, m^* is the forged message, σ_i^* is the forged signature about (ID_i^*, m^*). If the forged signature σ_i^* is a valid signature and $\mathscr{A}_{\mathrm{I}}$ could not submit $ID_i^* = ID_i$ to Secret-Value-Query and Signature-Query, then attacker $\mathscr{A}_{\mathrm{I}}$ wins the game.

Game II. The game is completed by the interaction between the challenger $\mathscr{C}_{\mathrm{II}}$ and the attacker $\mathscr{A}_{\mathrm{II}}$.

Initialization Phase: Input security parameter k, then run the Setup algorithm to generate params and s. $\mathscr{C}_{\mathrm{II}}$ sends params and s to $\mathscr{A}_{\mathrm{II}}$.

Inquiry Phase: The inquiry process is consistent with Game I.

Forgery Phase: $\mathscr{A}_{\mathrm{II}}$ outputs $(ID_i^*, m^*, \sigma_i^*)$, where ID_i^* is the target user that $\mathscr{A}_{\mathrm{II}}$ chooses to forge the signature, m^* is the forged message, σ_i^* is the forged signature about (ID_i^*, m^*). If the forged signature σ_i^* is a valid signature and $\mathscr{A}_{\mathrm{II}}$ could not submit $ID_i^* = ID_i$ to Private-Value-query and Signature-query, then attacker $\mathscr{A}_{\mathrm{I}}$ wins the game.

If attackers $\mathscr{A}_{\mathrm{I}}$ and $\mathscr{A}_{\mathrm{II}}$ can win Game I and Game II, respectively, the scheme is unforgeable under adaptation and identity attacks.

4. The Proposed CLDS-B Scheme

The CLDS-B scheme employs the certificateless digital signature and the blockchain to achieve data authentication and data integrity in NDNs. It consists of four phases: network initialization, producer registration, Data packet publication, and Data packet verification.

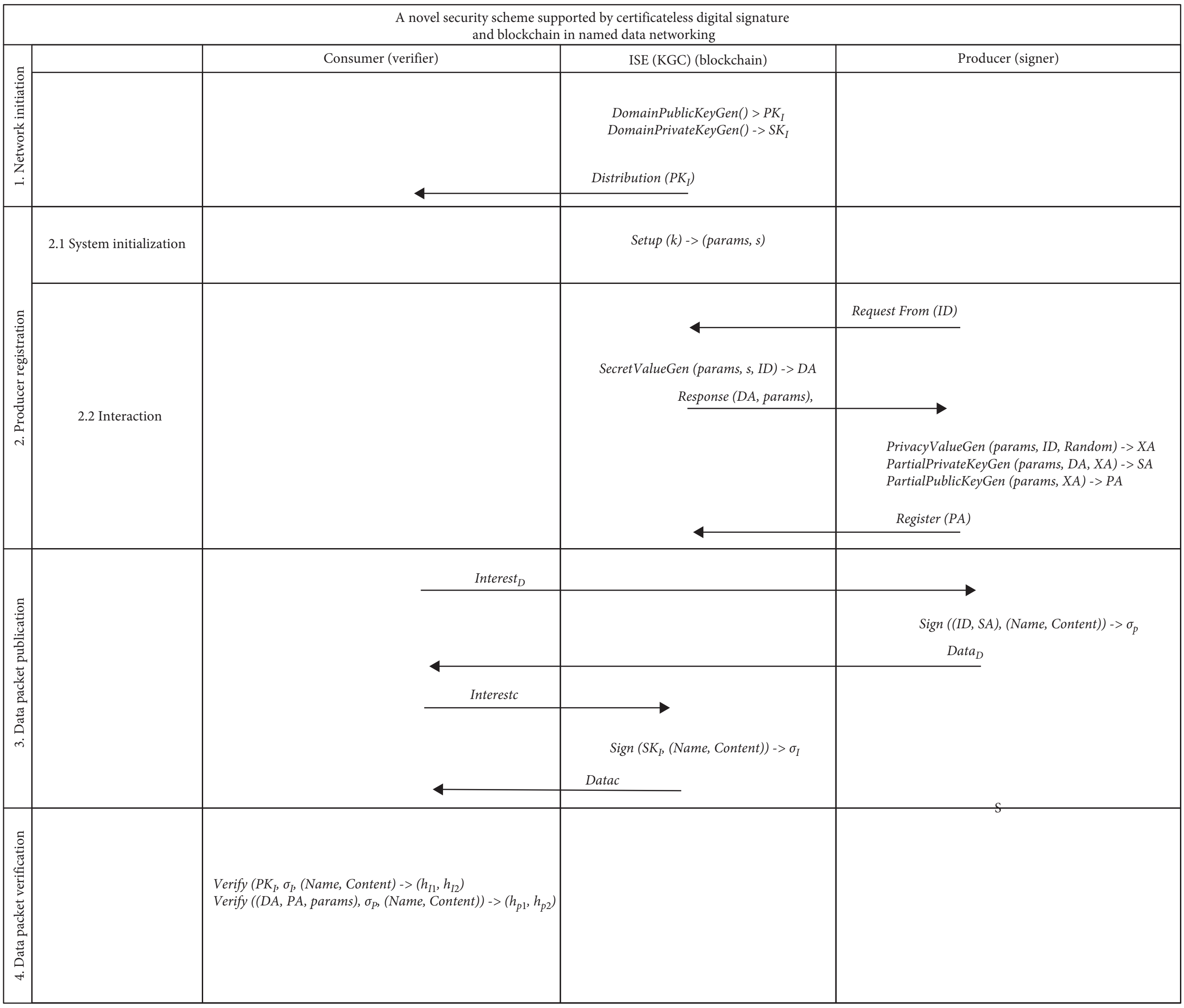
The notations used in this section and their descriptions are defined in Table 1. The complete process of our proposed solution is shown in Figure 2.

4.1. Network Initiation. An ISE is responsible for generating the domain key pair following Expressions (1) and (2), including the domain public key PK_I and the domain private key SK_I. It would keep the domain private key SK_I confidential and notify the domain public key PK_I to all the consumers in the same domain as Assumption (2).

$$DomainPublicKeyGen(\,) \rightarrow PK_I, \tag{1}$$

$$DomainPrivateKeyGen(\,) \rightarrow SK_I. \tag{2}$$

4.2. Producer Registration. In this phase, the producers would generate their key pairs with the assistance of their ISEs and would register their public keys at their ISEs. The key pair


A novel security scheme supported by certificateless digital signature
and blockchain in named data networking
Consumer (verifier)
ISE (KGC) (blockchain)
Producer (signer)
1. Network initiation
DomainPublicKeyGen() > PK$_I$
DomainPrivateKeyGen() -> SK$_I$
Distribution (PK$_I$)
2. Producer registration
2.1 System initialization
Setup (k) -> (params, s)
2.2 Interaction
Request From (ID)
SecretValueGen (params, s, ID) -> DA
Response (DA, params),
PrivacyValueGen (params, ID, Random) -> XA
PartialPrivateKeyGen (params, DA, XA) -> SA
PartialPublicKeyGen (params, XA) -> PA
Register (PA)
3. Data packet publication
Interest$_D$
Sign ((ID, SA), (Name, Content)) -> σ$_p$
Data$_D$
Interestc
Sign (SK$_I$, (Name, Content)) -> σ$_I$
Datac
S
4. Data packet verification
Verify (PK$_I$, σ$_I$, (Name, Content) -> (h$_{I1}$, h$_{I2}$)
Verify ((DA, PA, params), σ$_P$, (Name, Content)) -> (h$_{p1}$, h$_{p2}$)

Note:
Interest$_D$: request for data
Data$_D$: reply using data
Interestc: request for crptographic information
Datac: reply using crptographic information

Figure 2: Complete process of CLDS-B scheme.

generation would follow the general process of the certificateless algorithm that has been introduced in Section 3.1.

4.2.1. System Initialization. The ISE would play the role of the KGC, which uses the system parameters k as its input and generates the system public parameters *params* and the system master key s as output, as shown in Expression (3).

$$Setup\ (k) \rightarrow (Params, s). \tag{3}$$

After generating the *params* and s, KGC would specify their validity period t.

4.2.2. Interaction. The interaction between producers and their ISEs for registration would be divided into two stages, as shown in Figure 3. Based on Assumption (1), the information exchange in this phase would be protected within a secure channel.

(1) The producer initiates a request, which includes its identity information *ID*. The ISE would authenticate the producer. The authentication method could be online or offline, which has been beyond this paper. Upon successful authentication, the ISE would calculate a secret value *DA* using *params* and *ID*, as shown in Expression (4). Subsequently, ISE would respond to the producer with both *DA* and *params*.

$$SecretValueGen\ (Params, s, ID) \rightarrow DA. \tag{4}$$

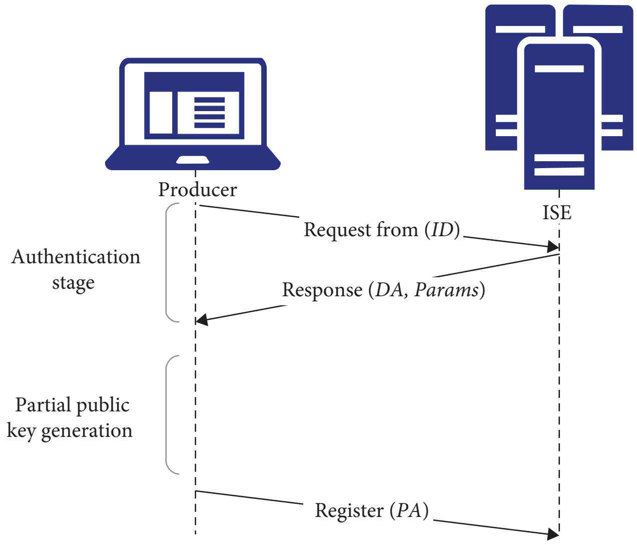


Figure 3: Information exchanges.

After receiving *DA* and *params*, the producer would calculate the privacy value *XA* itself. It chooses a random value, *random*, and calculates its privacy value using the public parameter *params*, its identity *ID*, and random, as shown in Expression (5).

$$\mathit{PrivacyValueGen}\,(\mathit{Params}, \mathit{ID}, \mathit{random}) \rightarrow \mathit{XA}. \quad (5)$$

After *XA* has been calculated, the producer uses the public parameter *params*, the secret value *DA*, and the *XA* to calculate the partial private key *SA*, as shown in Expression (6).

$$\mathit{PartialPrivateKeyGen}\,(\mathit{Params}, \mathit{DA}, \mathit{XA}) \rightarrow \mathit{SA}. \quad (6)$$

At the same time, the producer also calculates the partial public key *PA* from the public parameter *params* and the privacy value *XA*, as shown in Expression (7).

$$\mathit{PartialPublicKeyGen}\,(\mathit{Params}, \mathit{XA}) \rightarrow \mathit{PA}. \quad (7)$$

At the end of this stage, the producer has owned its key pair. The private key is composed of the *SA* and the *ID*, and the public key is composed of the *PA*, the *ID*, and the *params*. Since the *random* is chosen by the producer, the ISE cannot calculate the same *SA* as the producer, so that it cannot know the private key of the producer.

(2) After the *PA* has been calculated, the producer would send a registration request to ISE. In the registration request, the *PA* would be notified to the ISE. For an authenticated producer, ISE would store the producer's identity *ID*, public parameters *params*, and the received *PA* onto the blockchain. Consequently, the blockchain would record the *ID*s, the *params*, and the *PA*s for all the authentication producers.

4.3. Data Packet Publication. As shown in Figure 4, both producers and ISEs would publish packets in NDNs.

When an Interest packet is received with the name prefix "/ndn/Data," the producer would query its local memory for the requested data by the received Interest and then publish a Data packet. The Data packet would include the requested data name, *Name*, the request data content, *Content*, and the signature, σ_p, at least. The calculation process of signature is as follows: Using a hash function to calculate the digest for the *Name* and *Content*. Then, the digest would be signed using the producer's private key to generate a signature σ_p according to Equation (8). Here, the producer's private key is composed of the *ID* and the *SA*. The *ID* used in the public key must be the same as the second field of the *Name*.

$$\mathit{Sign}\,((\mathit{ID}, \mathit{SA}), (\mathit{Name}, \mathit{Content})) \rightarrow \sigma_p. \quad (8)$$

When an Interest packet is received with the name prefix "/ndn/Crypto," the ISE would query its blockchain the requested cryptographic data and then publish a Data packet. The Data packet would include the requested data name, *Name*, the requested cryptographic information, *Content*, and the signature, σ_I, at least. The cryptographic information, *Content*, includes the public parameters, *params*, their expiration date, and the producer's partial public key, *PA*. The calculation process of signature is as follows: using a hash function to calculate the digest for the *Name* and *Content*. Then, the digest would be signed using the domain private key SK_I to generate a signature σ_I according to Equation (9).

$$\mathit{Sign}\,(SK_I, (\mathit{Name}, \mathit{Content})) \rightarrow \sigma_I. \quad (9)$$

4.4. Data Packet Verification. As shown in Figure 5, when a Data packet arrives, the consumer would process the Data packet with two steps, including data name examination and Data packet verification. In the data name examination, the first field of the data name (*Name*) is examined. If the first field is neither "/ndn/Crypto" nor "/ndn/Data," the consumer would discard the Data packet directly due to the illegal data name. Otherwise, Data packet verification would be triggered. In the Data packet verification, the consumer would retrieve the corresponding public key to verify the signature as follows:

If the name prefix is "/ndn/Data," the Data packet would be considered to be published by the producer, so that the consumer must obtain the public key of the producer to verify the Data packet. The consumer would query the producer's cryptographic information in its local memory according to the producer's routing identifier that is contained in the second field of the *Name*. If the cryptographic information could be found in the local memory, the expiration date in the cryptographic information would be checked. If it indicates the cryptographic information does not expire, the cryptographic information would be used directly for the Data packet verification. Otherwise, no matter whether the cryptographic information expires or it cannot be found, the consumer would send a new Interest packet to request

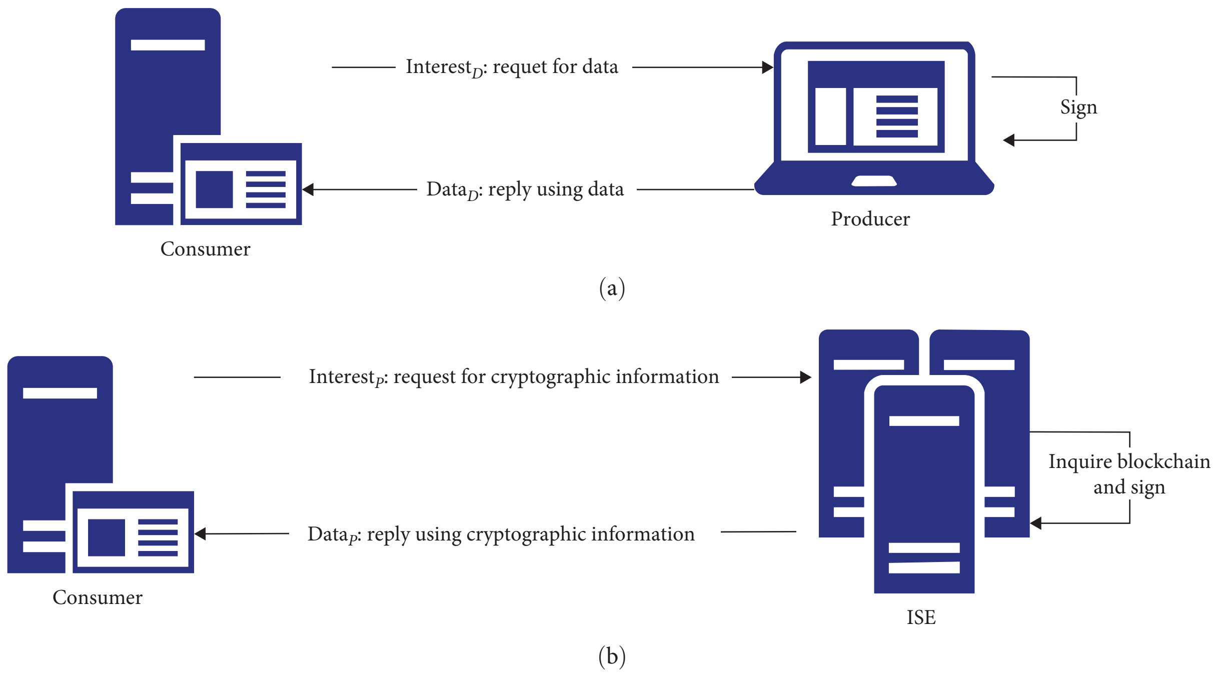

Interest$_D$: requet for data
Sign
Data$_D$: reply using data
Producer
Consumer
(a)
Interest$_P$: request for cryptographic information
Inquire blockchain and sign
Data$_P$: reply using cryptographic information
Consumer
ISE
(b)

Figure 4: Data packet publication: (a) request data; (b) request cryptographic information.

the cryptographic information of the producer. In detail, the first field of the data name in the new Interest packet would be "/ndn/Crypto," and the second field of the data name would be the routing identifier of the ISE that locates the same domain as the consumer. Moreover, the third field of the data name would be the routing identifier of the producer. Therefore, the data name would indicate that the new Interest would be routed to the ISE and the cryptographic information of the producer is requested. As a result, a new Data packet would be received as a response to the new Interest. Its verification detail would be described in the next paragraph. If the verification result is true, the cryptographic information contained in the *Content* would be used for the verification of the current Data packet. The cryptographic information fetched from the local memory or the Data packet whose name prefix is "/ndn/Crypto" would include the public parameters, *params*, their expiration date, and the producer's partial public key, *PA*. The public key of the producer would be composed of the *params*, the *PA*, and the *ID*. The *params* and the *PA* could be obtained from the fetched cryptographic information easily, and the ID would refer to the routing identifier of the producer contained in the second field of the data name. As a result, the signature could be verified according to Equation (10). In detail, the received signature σ_p would be decrypted by the producer's public key to get the digest h_{p1}, and the hash algorithm is used to calculate the other digest for the *Name* and *Content* as h_{p2}. If h_{p1} is equal to h_{p2}, the verification result is *True*. The Data packet would be determined as the one with data authentication and data integrity. Therefore, it would be forwarded to the NDN application at the consumer. Otherwise, the verification result is *False*. The Data packet would be determined as the one from the unauthenticated producer or the one that has been modified. Therefore, it would be discarded as follows:

$$Verify\left((params, PA, ID), \sigma_p, (Name, Content)\right) \rightarrow \left(h_{p1}, h_{p2}\right). \quad (10)$$

When the name prefix is "/ndn/Crypto," the Data packet would be considered to be published by the ISE so that the consumer would verify the Data packet using the domain public key, PK_I. The PK_I has been distributed from the ISE to all network entities in the same domain as Assumption (2) so that the consumer could fetch the PK_I from its local memory easily. The signature σ_I is verified according to Equation (11). In detail, the received signature σ_I would be decrypted by PK_I to get the digest h_{I1}, and the hash algorithm is used to calculate the other digest for the *Name* and *Content* as h_{I2}. If h_{p1} is equal to h_{I2}, the verification result is *True*. The Data packet would be determined as the one with data authentication and data integrity. Therefore, the cryptographic information in the *Content* field would be used to verify the Data packet with the name prefix "/ndn/Data" and would also be stored in the local memory of the consumer. Otherwise, the verification result is *False*. The Data packet would be determined as the one that is published by an unauthenticated producer or the one that has been modified. Therefore, it would be discarded as follows:

$$Verify\left(PK_I, \sigma_I, (Name, Content)\right) \rightarrow \left(h_{I1}, h_{I2}\right). \quad (11)$$

5. Security Analysis

This section would analyze the security of our CLDS-B. Both the formal analysis and the informal analysis would be presented, respectively.

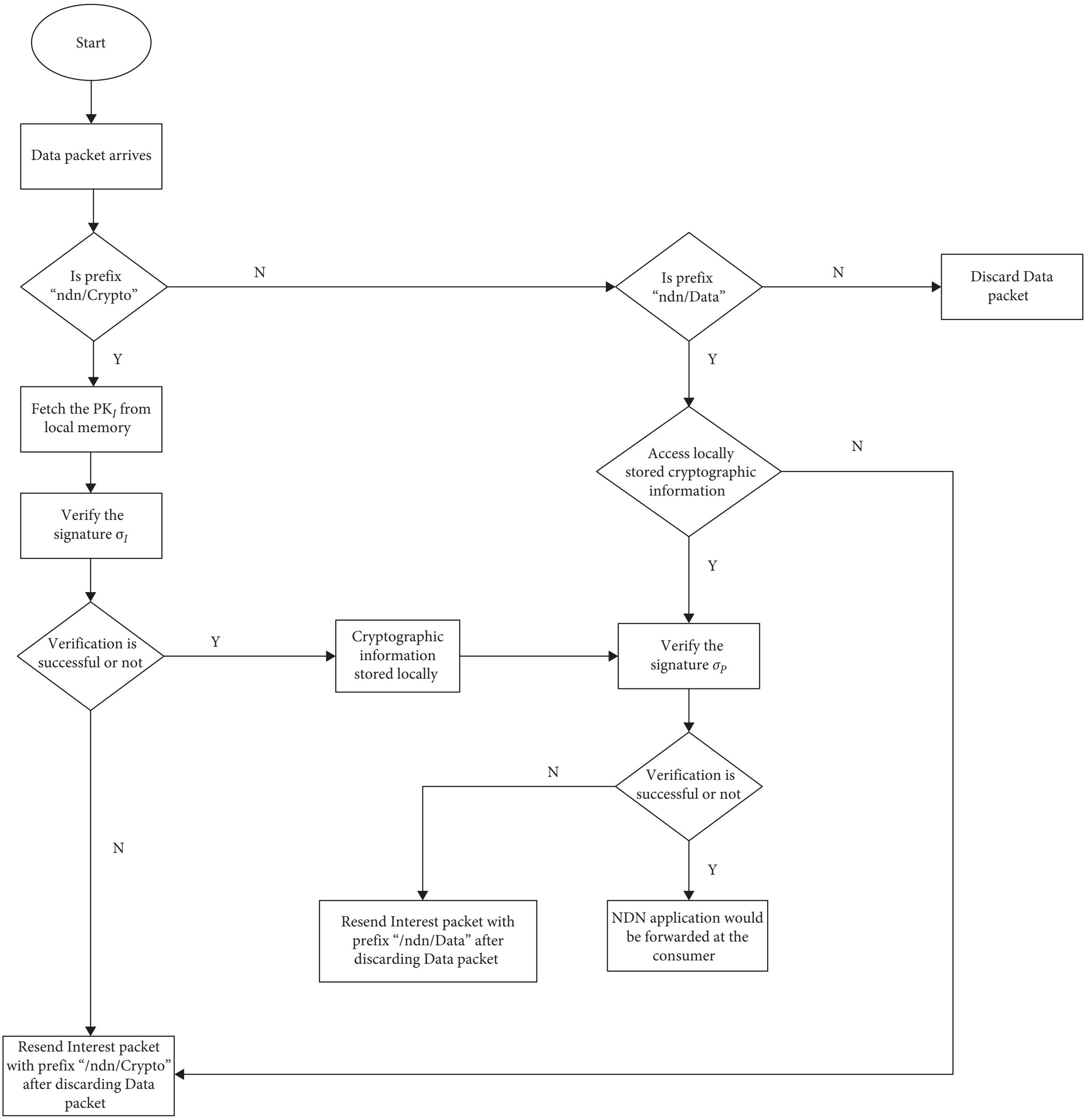

Start
Data packet arrives
Is prefix "ndn/Crypto"
N
Y
Fetch the PK_I from local memory
Verify the signature σ_I
Verification is successful or not
Y
N
Cryptographic information stored locally
Is prefix "ndn/Data"
N
Discard Data packet
Y
Access locally stored cryptographic information
N
Y
Verify the signature σ_P
Verification is successful or not
N
Y
Resend Interest packet with prefix "/ndn/Data" after discarding Data packet
NDN application would be forwarded at the consumer
Resend Interest packet with prefix "/ndn/Crypto" after discarding Data packet

Figure 5: Data packet verification.

5.1. *Formal Security Analysis*

5.1.1. Certificateless Signature Algorithm. Any Certificateless Signature Algorithm that can provide high-security strength could be a candidate for our CLDS-B scheme. Here, a certificateless digital signature based on elliptic curve cryptography is selected as our referenced algorithm [8]. There are three legitimate participants in the CLDS-B: the producer, the consumer, and the ISE. The producer and the consumer would play the role of the signer and the role of the verifier, respectively, and the ISE would play the role of a KGC. The scheme consists of the following algorithm:

Setup: It is executed by the ISE. A prime number q with a length of λ-bit is selected first. G is an elliptic additive group of order q over finite field F_p. $P \in G$ is a generator. Then, the ISE randomly selects the system master private key $s \in Z_q^*$ and calculates $P_{\text{pub}} = sP$. At the same time, it picks three hash functions as follows:

$$H_1 : \{0,1\}^* \times G \rightarrow Z_q^*, \tag{12}$$

$$H_2 : \{0,1\}^* \times G \times G \rightarrow Z_q^*, \tag{13}$$

$$H_3 : \{0,1\}^* \times Z_q^* \times G \times G \rightarrow Z_q^*. \tag{14}$$

At last, the ISE would combine the system parameters as follows:

$$Params : \left(G, P, P_{\text{pub}}, H_1, H_2, H_3\right). \tag{15}$$

In summary, the Setup algorithm inputs parameter λ and outputs s and system parameter *params* at an ISE.

Secret-Value-Generation: It is also executed by the ISE. After obtaining the producer's routing identity ID_i, the ISE randomly chooses $r_i \in Z_q^*$ and then computes $R_i = r_iP$, $h_i = H_1(ID_i, R_i)$, $s_i = (r_i + h_is)$ mod q. After that, it must verify an equation $s_iP = R_i + h_iP_{\text{pub}}$, since $s_iP = (r_i + h_is)P = R_i + h_iP_{\text{pub}}$. If the equation holds, the s_i would be referred to as the secret value DA_i and then the (s_i, R_i) would be delivered to the producer using a secure channel described in Assumption (1). Otherwise, the ISE must restart the Secret-Value-Generation.

Private-Value-Generation: It is executed by the producer. An number $x_i \in Z_q^*$ is randomly selected as the privacy value XA_i.

Partial-Private-Key-Generation: It is also executed by the producer. The partial private key, SA_i, would be composed of the secret value and the privacy value, i.e., $SA_i = (s_i, x_i)$, and the private key of the producer would be viewed as the combination of the SA_i and the ID_i.

Partial-Public-Key-Generation: It is also executed by the producer. The calculation is performed as follows: $P_i = x_iP$, $u = H_2(ID_i, P_i)$, $Q_i = R_i + uP_i$, and then the partial public key, PA_i, is composed of the R_i and the Q_i, i.e., $PA_i = (R_i, Q_i)$, and the public key of the producer would be viewed as the combination of the PA_i, the *params*, and the ID_i.

Signature: It is also executed by the producer. The *Name* and the *Content* in the Data packet are viewed as the message m. Therefore, the inputs would be the m, and the public key of the producer that includes the param, the ID_i and tSA_i, and the output is the signature σ. The calculation procedure is as follows: The producer randomly chooses $t \in Z_q^*$ and computes $T = tP = (T_x, T_y)$, where T_x, T_y denotes the x-coordinate, y-coordinate respectively. The it calculates $r = T_x$ mod q, $u = H_2(ID_i, P_i)$, $v = H_3(ID_i, m, h_i, PK_i, T)$, and $\tau = t^{-1}(v + r(s_i + ux_i))$mod q. After that, the combination of the T and the τ would be viewed as the signature, i.e., $\sigma = (T, \tau)$.

Verification: It is executed by the consumer. The inputs would be the message, the signature, and the public key of the producer, i.e., $(m, \sigma, PA_i, params, ID_i)$. The output is the verification result, whose value is True or False. The value of True indicates the m is data-authenticated and data-integral, and the value of *False* indicates the m is published by an unauthenticated producer or is modified. The verification procedure is as follows: The consumer calculates $h_i = H_1(ID_i, R_i)$, $v = H_3(ID, m, h_i, PK_i, T)$, and $r = T_x$ mod q. Then it checks whether $\tau T = vP + r(Q_i + h_iP_{\text{pub}})$ holds. If it does, the consumer outputs *True*, else it outputs *False*.

The Proof for the verification is shown as follows:

$$\tau = t^{-1}(v + r(s_i + ux_i)) \bmod q, T = tP, \tag{16}$$

$$\tau T = (t^{-1}(v + r(s_i + ux_i)))tP \bmod q, s_i = (r_i + h_is), \tag{17}$$

$$\tau T = vP + r(r_i + h_is + ux_i)P, \tag{18}$$

$$\tau T = vP + r(r_iP + h_isP + ux_iP), \ R_i = r_iP, P_i = x_iP, \tag{19}$$

$$\tau T = vP + r\left(R_i + h_isP + uP_{\text{pub}}\right), P_{\text{pub}} = sP, \tag{20}$$

$$\tau T = vP + r\left(R_i + uP_i + h_iP_{\text{pub}}\right), Q_i = R_i + uP_i, \tag{21}$$

$$\tau T = vP + r\left(Q_i + h_iP_{\text{pub}}\right). \tag{22}$$

5.1.2. Algorithm Security Analysis. The proposed CLDS-B scheme would be Existential Unforgeability against chosen-message attacks (EUF-CMA), safe against super adversaries if the Elliptic Curve Discrete Logarithm Problem (ECDLP) is difficult to handle in a random prediction machine.

Lemma 1. *Under the stochastic prediction machine model, if $\mathscr{A}_{\text{I}}$ is a Type I adversary that can attack the unforgeability of signatures in this scheme in polynomial time by a nonnegligible margin ϵ, there exists an ECDLP challenger $\mathscr{C}_{\text{I}}$ that can solve the ECDLP with $\varepsilon' \geq \varepsilon(1 - \frac{1}{q_1})^{q_2} \frac{1}{q_1}$ in polynomial time by a nonnegligible margin.*

q_1: the number of producers generated queries made by adversary $\mathscr{A}_{\text{I}}$ to the random prediction machine.

q_2 : the number of secret value queries made by adversary $\mathscr{A}_{\text{I}}$.

Proof. Assume that $\mathscr{C}_{\text{I}}$ is an attacker against ECDLP. $\mathscr{C}_{\text{I}}$ has parameters $(P, P_{\text{pub}} = sP)$, and its goal is to calculate s.

$\mathscr{C}_{\text{I}}$ makes its subroutine $\mathscr{A}_{\text{I}}$ an adversary to break through the proposed signature scheme under adaptive selected message attacks.

$\mathscr{C}_{\text{I}}$ maintains lists L_1, L_2, L_3, respectively, recording the hash queries to H_1, H_2, H_3.

$\mathscr{C}_{\text{I}}$ maintains lists $L_{\text{PSK}}, L_S, L_{PK}, L_{\text{Sign}}$ to record queries on partial private keys, secret values, public keys, and signatures. □

Initialization Phase: $\mathscr{C}_{\text{I}}$ runs Setup algorithm to calculate system parameters *params* and send $Params = (G, P, P_{\text{pub}}, H_1, H_2, H_3)$ to $\mathscr{A}_{\text{I}}$.

Inquiry Phase: $\mathscr{A}_{\text{I}}$ makes the following inquiries:

H_1-Query: The format of each item in the list is (ID_i, R_i, h_i). When a query is received, if $(ID_i, R_i, h_i) \in L_1$, then $\mathscr{C}_{\text{I}}$ returns h_i to $\mathscr{A}_{\text{I}}$. Otherwise $\mathscr{C}_{\text{I}}$ randomly picks $h_i \in Z_q^*$ and creates (ID_i, R_i, h_i) in L_1 and returns h_i to $\mathscr{A}_{\text{I}}$.

H_2-Query: The format of each item in the list is (ID_i, P_i, u). When a query is received, if $(ID_i, P_i, u) \in L_2$, then $\mathscr{C}_{\mathrm{I}}$ returns u to $\mathscr{A}_{\mathrm{I}}$. Otherwise $\mathscr{C}_{\mathrm{I}}$ randomly picks $u \in Z_q^*$, updates (ID_i, P_i, u) in L_2 and returns u to $\mathscr{A}_{\mathrm{I}}$.

H_3-Query: The format of each item in the list is $(ID_i, m, h_i, PK_i, T, v)$. When a query is received, if $(ID_i, m, h_i, PK_i, T, v) \in L_3$, then $\mathscr{C}_{\mathrm{I}}$ returns v to $\mathscr{A}_{\mathrm{I}}$. Otherwise $\mathscr{C}_{\mathrm{I}}$ randomly picks $v \in Z_q^*$, updates $(ID_i, m, h_i, PK_i, T, v)$ in L_3 and returns v to $\mathscr{A}_{\mathrm{I}}$.

Secret-Value-Generation-Query: The format of each item in the list is (ID_i, s_i, R_i). When a query is received, if $(ID_i, s_i, R_i) \in L_{\mathrm{PSK}}$, then $\mathscr{C}_{\mathrm{I}}$ returns R_i to $\mathscr{A}_{\mathrm{I}}$. Otherwise $\mathscr{C}_{\mathrm{I}}$ randomly picks $s_i \in Z_q^*$, calculates $R_i = s_iP - h_iP_{\mathrm{pub}}$, creates (ID_i, s_i, R_i) in L_{PSK} and returns (s_i, R_i) to $\mathscr{A}_{\mathrm{I}}$.

Partial-Private-Key-Query: The format of each item in the list is (ID_i, x_i). When a query is received, if $(ID_i, x_i) \in L_S$, then $\mathscr{C}_{\mathrm{I}}$ returns x_i to $\mathscr{A}_{\mathrm{I}}$. Otherwise $\mathscr{C}_{\mathrm{I}}$ randomly picks $x_i \in Z_q^*$, updates (ID_i, x_i) in L_S and returns x_i to $\mathscr{A}_{\mathrm{I}}$.

Partial-Public-Key-Query: The format of each item in the list is (ID_i, R_i, Q_i). When a query is received, if $(ID_i, R_i, Q_i) \in L_{PK}$, then $\mathscr{C}_{\mathrm{I}}$ returns (R_i, Q_i) to $\mathscr{A}_{\mathrm{I}}$. Otherwise $\mathscr{C}_{\mathrm{I}}$ randomly picks $x_i \in Z_q^*$, creates (ID_i, x_i) in L_S and returns x_i to $\mathscr{A}_{\mathrm{I}}$. Otherwise $\mathscr{C}_{\mathrm{I}}$ queries corresponding $(ID_i, s_i, R_i) \in L_{\mathrm{PSK}}$ and $(ID_i, x_i) \in L_S$ using ID_i in list L_{PSK} and list L_S. Then $\mathscr{C}_{\mathrm{I}}$ calculates $P_i = x_iP$ and $Q_i = R_i + uP_i$, updates (ID_i, R_i, Q_i) in L_{PK}, and returns (R_i, Q_i) to $\mathscr{A}_{\mathrm{I}}$.

Partial-Public-Key-Replacement-Query: $\mathscr{A}_{\mathrm{I}}$ selects a new public key $PK_i' = (R_i', Q_i')$, and sends (ID_i, PA_i') to $\mathscr{C}_{\mathrm{I}}$. When $\mathscr{C}_{\mathrm{I}}$ receives a public key replacement query from $\mathscr{A}_{\mathrm{I}}$, $\mathscr{C}_{\mathrm{I}}$ updates (ID_i, R_i', Q_i') in L_{PK}.

Signature-Query: The format of each item in the list is $(ID_i, m, x_i, SA_i, \sigma)$. When a query is received, if $(ID_i, m, x_i, SA_i, \sigma) \in L_{\mathrm{Sign}}$, then $\mathscr{C}_{\mathrm{I}}$ randomly picks $t \in Z_q^*$, calculates $T = tP$, $u = H_2(ID_i, P_i)$, $v = H_3(ID_i, m, h_i, PA_i, T)$ and $\tau = t^{-1}(v + r(s_i + ux_i))$ mod q, and returns $\sigma = (T, \tau)$ to $\mathscr{A}_{\mathrm{I}}$. Otherwise $\mathscr{C}_{\mathrm{I}}$ randomly picks $\tau \in Z_q^*$, calculates $T = \tau^{-1}(vP + r(Q_i + h_iP_{\mathrm{pub}}))$, and returns $\sigma = (T, \tau)$ to $\mathscr{A}_{\mathrm{I}}$.

Forgery Stage: $\mathscr{A}_{\mathrm{I}}$ outputs a forged signature $\sigma^* = (T^*, \tau^*)$ about the identity ID_i^*, message m^* after the above queries. In Game I, $\mathscr{A}_{\mathrm{I}}$ is not allowed to submit ID_i^* to Partial-Private-Key-query and (ID_i^*, m^*) has never been queried to Signature-query. If $ID_i^* \neq ID_i$, then $\mathscr{A}_{\mathrm{I}}$ terminates the game. If σ^* is valid, the following expression holds:

$$\tau^* = (t^*)^{-1}(v^* + r^*(s_i^* + u^*x_i^*)) \bmod q, \tag{23}$$

$$\text{Since } s_i^* = (r_i + h_is), R_i^* = r_iP, \tag{24}$$

$$\tau^*t^* = (v^* + r^*(r_i + h_is + u^*x_i^*)) \bmod q. \tag{25}$$

According to the bifurcation priming, different hash functions can be selected to get two forged signatures $(T^*, \tau^{(1)})$ and $(T^*, \tau^{(2)})$. The following equations can be obtained:

$$\tau^{(1)}t^* = \left(v^{(1)} + r^*(s_i^* + u^*x_i^*)\right) \bmod q, \tag{26}$$

$$\tau^{(2)}t^* = \left(v^{(2)} + r^*(s_i^* + u^*x_i^*)\right) \bmod q. \tag{27}$$

t^*, x_i^*, s are unknown variables. $\mathscr{C}_{\mathrm{I}}$ could calculate the value s from the equation.

If the following events occur, $\mathscr{C}_{\mathrm{I}}$ can successfully utilize attacker $\mathscr{A}_{\mathrm{I}}$ to solve ECDLP.

E_1: Adversary $\mathscr{A}_{\mathrm{I}}$ does not submit $ID_i^* = ID_i$ to Partial-Private-Key-query.

E_2: Adversary $\mathscr{A}_{\mathrm{I}}$ successfully outputs a signature with $ID_i^* = ID_i$.

E_3: Adversary $\mathscr{A}_{\mathrm{I}}$ successfully forges two different valid signatures.

During the query phase, $\mathscr{A}_{\mathrm{I}}$ makes q_2 partial private key queries in total, with a probability of $\frac{1}{q_1}$ finding the ID_i for each query. So $Pr(E_1) = (1 - \frac{1}{q_1})^{q_2}$.

The probability of that $\mathscr{A}_{\mathrm{I}}$ forges a signature with $ID_i^* = ID_i$ output during the forgery stage is $\frac{1}{q_1}$. So $Pr(E_2) = \frac{1}{q_1}$.

The probability of $\mathscr{A}_{\mathrm{I}}$ successfully outputting a valid forged signature is ε, and according to the bifurcation lemma, the probability of adversary $\mathscr{A}_{\mathrm{I}}$ successfully forging two different valid signatures is $Pr(E_3) \geq \varepsilon(1 - \frac{1}{q_1})^{q_2} \frac{1}{q_1}$.

The probability of event E_3 occurring is ε'. If ε is not negligible, then challenge $\mathscr{C}_{\mathrm{I}}$ can solve ECDLP with a probability of $\varepsilon' \geq \varepsilon(1 - \frac{1}{q_1})^{q_2} \frac{1}{q_1}$, which cannot be ignored either.

Lemma 2. *Under the stochastic prediction machine model, if $\mathscr{A}_{\mathrm{II}}$ is a Type* II *adversary that can attack the unforgeability of signatures in this scheme in polynomial time by a nonnegligible margin ε, then there exists an ECDLP challenger $\mathscr{C}_{\mathrm{II}}$ that can solve the ECDLP with $\varepsilon' \geq \varepsilon(1 - \frac{1}{q_3})^{q_4+q_5} \frac{1}{q_3}$ in polynomial time by a nonnegligible margin.*

q_3: the number of producers generated queries oracle queried by adversary $\mathscr{A}_{\mathrm{II}}$ to the random prediction machine.

q_4: the number of Secret-Value-queries oracle queried by adversary $\mathscr{A}_{\mathrm{II}}$.

q_5: the number of Public-Key-Replacement-queries oracle queried by adversary $\mathscr{A}_{\mathrm{II}}$.

The proof of Lemma 2 is similar to that of Lemma 1 with the difference that adversary $\mathscr{A}_{\mathrm{II}}$ has the KGC master key but cannot replace the producer's public key. Since $\mathscr{A}_{\mathrm{II}}$ does not know the privacy value XA_i of the producer, the objective of $\mathscr{C}_{\mathrm{II}}$ is to solve to get the privacy value XA_i.

Initialization Phase: $\mathscr{C}_{\mathrm{II}}$ runs Setup algorithm to calculate system parameters params and s, then send $Params = (G, P, P_{\mathrm{pub}}, H_1, H_2, H_3)$ and s to $\mathscr{A}_{\mathrm{II}}$.

Inquiry Phase: The inquiry process is consistent with Lemma 1.

Forgery Stage: $\mathscr{A}_{\mathrm{II}}$ outputs a forged signature $\sigma' = (T', \tau')$ about the identity ID_i^*, message m^* after the above queries. In Game II, $\mathscr{A}_{\mathrm{II}}$ is not allowed to submit ID_i^* to Secret-Value-query and Public-Key-Replacement-query, and (ID_i^*, m^*) has never been queried to Signature-query. If $ID_i^* \neq ID_i$, then $\mathscr{A}_{\mathrm{II}}$ terminates the game. If σ' is valid and the following equation holds:

$$\tau' t' = (v' + r'(s_i + u'x)) \bmod q. \tag{28}$$

According to the bifurcation priming, we can choose a different hash function to get a new forged signature $\sigma'' = (T', \tau'')$ and the following equation holds:

$$\tau'' t' = (v'' + r'(s_i + u'x)) \bmod q. \tag{29}$$

t', x are unknown variables. $\mathscr{C}_{\text{II}}$ could calculate the value x from the equations. If the following event occurs, $\mathscr{C}_{\text{II}}$ can successfully utilize attacker $\mathscr{A}_{\text{II}}$ to solve ECDLP.

E_1: Adversary $\mathscr{A}_{\text{II}}$ does not submit $ID_i^* = ID_i$ to Secret-Value-query and Public-Key-Replacement-query.

E_2: Adversary $\mathscr{A}_{\text{II}}$ successfully outputs a signature with $ID_i^* = ID_i$.

E_3: Adversary $\mathscr{A}_{\text{II}}$ successfully forges two different valid signatures.

During the query phase, since $\mathscr{A}_{\text{II}}$ makes q_4 secret value queries and q_5 public key replacement queries in total with the probability of finding the ID in each query to be $1/q$, $Pr(E_1) = (1 - \frac{1}{q_3})^{q_4} \times (1 - \frac{1}{q_3})^{q_5} = (1 - \frac{1}{q_3})^{q_4+q_5}$.

The probability of $\mathscr{A}_{\text{II}}$ forged signature with $ID_i^* = ID_i$ output during the forgery stage is $\frac{1}{q_3}$. So $Pr(E_2) = \frac{1}{q_3}$.

The probability of $\mathscr{A}_{\text{II}}$ successfully outputting a valid forged signature is ε, and according to the bifurcation lemma, the probability of adversary $\mathscr{A}_{\text{II}}$ successfully forging two different valid signatures is $Pr(E_3) \geq \varepsilon(1 - \frac{1}{q_3})^{q_4+q_5} \frac{1}{q_3}$.

The probability of event E_3 occurring is ε'. If ε is not negligible, then challenge $\mathscr{C}_{\text{II}}$ can solve ECDLP with a probability of $\varepsilon' \geq \varepsilon(1 - \frac{1}{q_3})^{q_4+q_5} \frac{1}{q_3}$, which cannot be ignored either.

5.2. Informal Security Analysis. The security of the CLDS-B would be analyzed for Four kinds of attacks as follows:

Data Authentication Attacks: In the CLDS-B, all the Data packets must encapsulate signatures, and then consumers would verify the signatures in Data packets to discard the Data packets with illegal signatures. Since consumers fetch public keys for signature verifications from ISEs, producers would register themselves to deliver their cryptographic information to ISEs. During the registration, producers would be authenticated at ISEs. Moreover, the producer's routing identifiers *ID*s that are encapsulated in the second field of the data names in Data packets are the element of public keys of signature verifications. Therefore, no imposture public keys could be provide to consumers. As a result, the Data packets that have been verified successfully must be published by the authenticated producers. Our CLDS-B scheme could resist the data authentication attack.

Data Integrity Attacks: In the CLDS-B, a signature is included in the Data packet. Any modification for the Data packet would cause the failure of verification for the signature. As a result, our CLDS-B could resist the data integrity attack.

KGC Internal Attacks: In the CLDS-B, the certificateless signature is employed instead of an identity-based signature. The ISE would take the role of the KGC to calculate the secret value *DA* for the producer. However, the producer's private key is the combination of the secret value *DA*, the privacy value *XA*, and the producer identifier *ID*. Therefore, the *DA* is just one element of the producer's private key. Since the *XA* is selected by the producer secretly, the ISE cannot calculate the *XA*, so that it cannot obtain the produces private key. As a result, any attempt by the ISE to fabricate a signature would be futile. The CLDS-B could prevent KGC internal attacks.

KGC Trust Attacks: In NDNs, both previous identity-based signature schemes and traditional certificateless signature schemes rely on a centralized trust center. However, these centralized schemes face the same security risk. The entire network's security is compromised if the centralized KGC is attacked. In contrast, our scheme leverages blockchain technology to implement decentralized smart contracts. The consensus algorithm of blockchain ensures that all the ISEs would maintain the consistent cryptographic information of producers. At the same time, the blocks at the blockchain have also record the digests of the previous blocks so that the modification of one block would result in all the following blocks must be modified. As a result, it is judged as impossible to tamper with the cryptographic information recorded at the blocks because the modification of all the following blocks in all the ISEs would take too many resources to be performed. Therefore, our scheme could prevent potential KGC trust attacks.

DDoS Attacks and Reply Attacks: Since our CLDS-B scheme still maintains the most mechanisms of the previous NDNs, the attacks that previous NDNs could prevent would also be prevented by our CLDS-B scheme. Since routers could reply to the Interests directly if they have cached the requested Data packets, the load for producers to reply would be mitigated greatly. Thus, producers would also be far away from the DDoS attacks in our CLDS-B scheme. Moreover, the Data packet has been viewed as the reply to the Interest packet so that the routes and consumers would discard the Data packets that do not match against any items of their PITs. Therefore, the Reply Attacks for the Data packet would be prevented.

6. Formal Security Verification

To demonstrate the security characteristics, we have formally validated the CLDS-B scheme by using Automatic Validation of Internet Security Protocols and Applications (AVISPA), an automatic formal verification tool to validate security protocols. Although it was originally designed for the Internet, AVISPA can be applied to NDN work without any difficulty.

6.1. Modeling Using HLPSL. The High-Level Protocol Specification Language (HLPSL) is used in the AVISPA system to describe secure communication and to verify authentication and integrity according to the CLDS-B scheme. By the

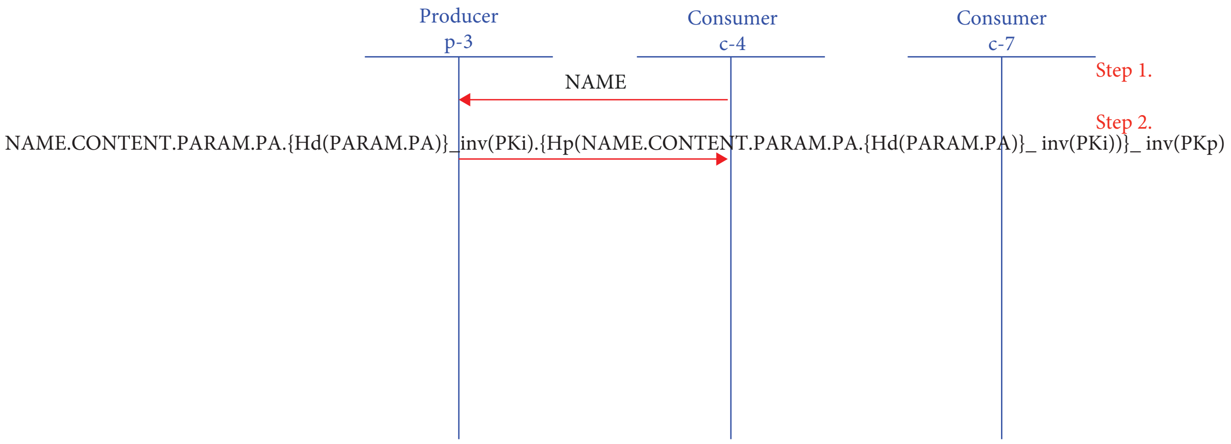

Producer
p-3
Consumer
c-4
Consumer
c-7
Step 1.
NAME
Step 2.
NAME.CONTENT.PARAM.PA.{Hd(PARAM.PA)}_inv(PKi).{Hp(NAME.CONTENT.PARAM.PA.{Hd(PARAM.PA)}_ inv(PKi))}_ inv(PKp)

Figure 6: Protocol validation for secure NDN communication.

CLDS-B scheme, there are two types of participants in the communication, the requester and the responder, where the requester refers to the consumer. Responders can be producers, ISEs, or routers, while producers and ISEs are able to publish packets, and routers act as packet caches. Different from IP networks, the NDNs only focus on the authentication of the packet publisher rather than that of the responder. Therefore, the role of the router is ignored in our formal verification model. Our verification is divided into two cases:

(a) Modeling the communication between the consumers and the producers.

(b) Modeling communication between consumers and ISEs.

Since the verification process is relatively similar for the two cases, only case (a) is presented in this section.

6.1.1. Basic Roles. Two basic roles, the consumer and the producer, have been defined in the verification model. The consumer will request the specified data via an Interest packet, while the producer will respond using a Data packet. These two basic roles would receive parameters from the combined role, declare their local variables, and perform transitions to simulate the interaction, as shown in Figure 4(a).

6.1.2. Parameters. Two more parameters have been defined besides the two basic roles. The first parameter, named *H*, is used for the Hash function in the signature. The other one, named *PK*, is used to simulate the producer's public key (i.e., *ID*, *PA*, and *params*). The *ID* in the public key is fetched from the second field of the data name in the Data packet so that it binds the data name to the producer's public key. Moreover, the *params* and *PA* have been delivered to the consumer through communication between the consumer and its ISE. Therefore, it is reasonably assumed that the consumer has the correct public key of the producer.

6.1.3. Local Variables. There are two local variables, named *Name* and *Content*, to model the payloads in the packet. The variable *Name* is used to model the name of the requested data in the Interest packet and the Data packet, while the variable *Content* is used to model the published data in the Data packet. The two variables are treated as new values at runtime so that they are generated locally by the new () operation in HLSPL.

6.1.4. Goals. The authentication goal is modeled in the validation. It works based on the producer's signature to guarantee packet integrity and data source authentication.

6.2. Verification Results. Both the CL-AtSe backend and the OFMC backend have been used to validate the CLDS-B scheme. The protocol verification at the sender side in a principal position is shown in Figure 6, and a snapshot of the intruder verification is shown in Figure 7. The verification result is "SAFE," as shown in Figures 8 and 9, which indicate the goals specified in HLSPL have been achieved, namely the data authentication and the data integrity.

7. Performance Evaluation

Since the CLDS-B has integrated the blockchain into the NDN, the performance evaluation would be performed from two respects, including the blockchain performance evaluation and the NDN performance evaluation.

The blockchain performance is analyzed as follows: Although the blockchain network is a multidomain network, its participants are the ISEs in each domain. Compared to that of producers and consumers in the NDNs, the scale of ISEs would be viewed as much smaller. Since the blockchain in our CLDS-B only takes charge of storing the cryptographic information of producers, the transactions in the blockchain network would be triggered only when the cryptographic information is created for a new producer or is updated. For one producer, it is one time to create its public key, and the frequency to its public key would be several weeks or months. Even if there are several thousands of producers, the transaction volume in the blockchain network would be small. To evaluate the performance of the blockchain network, we have implemented a blockchain experiment based on Hyperledger Fabric [37]. In our experiment, there are 11 ISEs that play three orders and eight peers, and

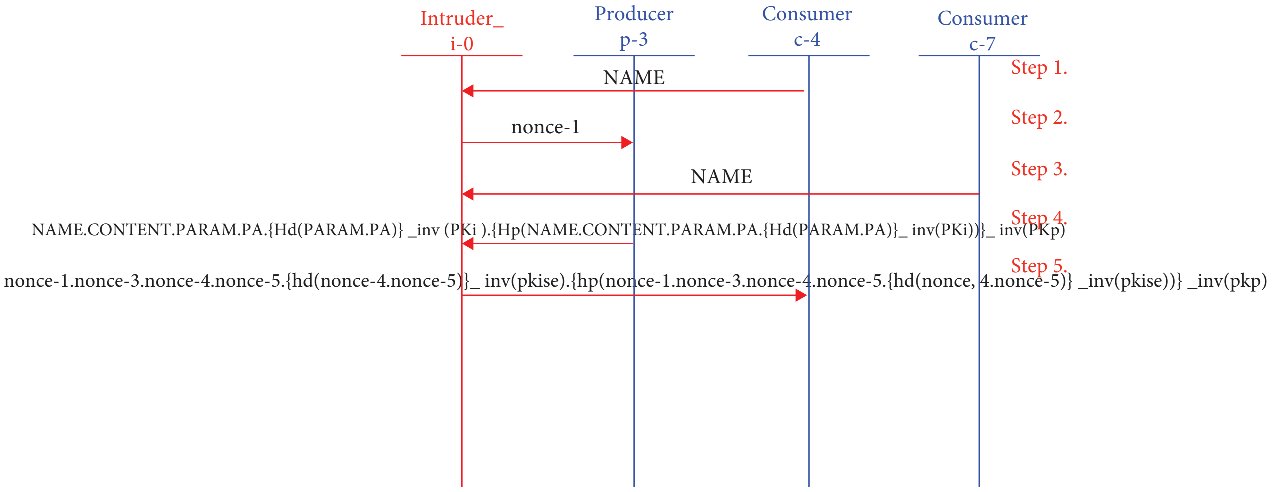
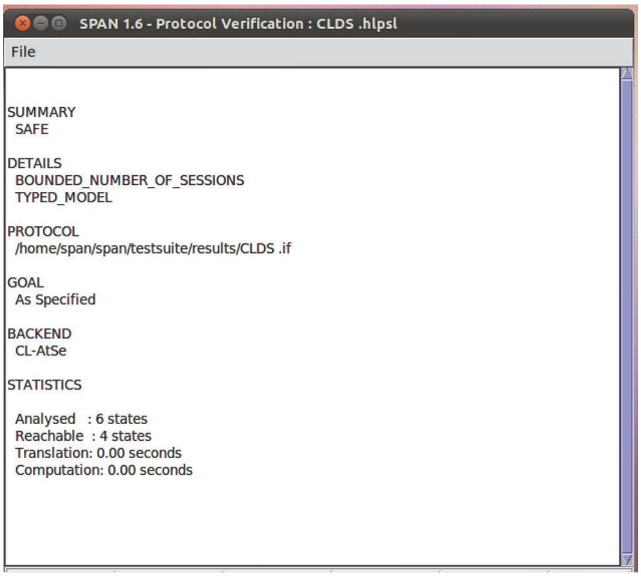
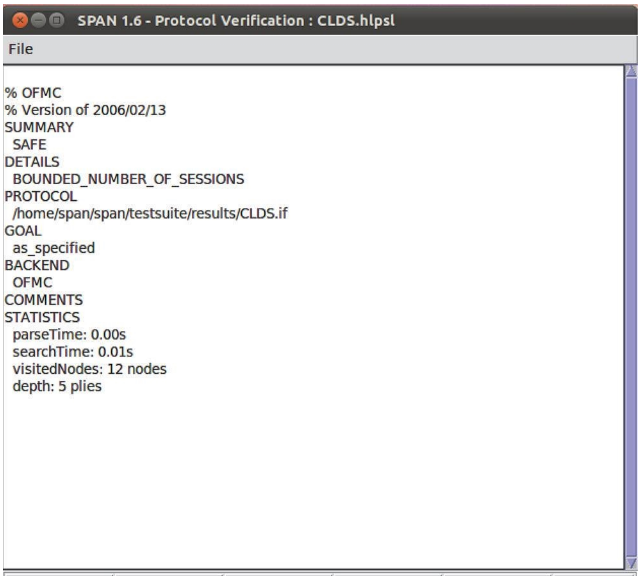

Intruder_
i-0
Producer
p-3
Consumer
c-4
Consumer
c-7
Step 1.
NAME
Step 2.
nonce-1
Step 3.
NAME
Step 4.
NAME.CONTENT.PARAM.PA.{Hd(PARAM.PA)} _inv (PKi).{Hp(NAME.CONTENT.PARAM.PA.{Hd(PARAM.PA)}_ inv(PKi))}_ inv(PKp)
Step 5.
nonce-1.nonce-3.nonce-4.nonce-5.{hd(nonce-4.nonce-5)}_ inv(pkise).{hp(nonce-1.nonce-3.nonce-4.nonce-5.{hd(nonce, 4.nonce-5)} _inv(pkise))} _inv(pkp)

Figure 7: Intruder authentication for NDN secure communications.



```
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL

PROTOCOL
  /home/span/span/testsuite/results/CLDS .if

GOAL
  As Specified

BACKEND
  CL-AtSe

STATISTICS

  Analysed   : 6 states
  Reachable  : 4 states
  Translation: 0.00 seconds
  Computation: 0.00 seconds
```

Figure 8: Validation result using backend of CL-AtSe.



```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/CLDS.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.01s
  visitedNodes: 12 nodes
  depth: 5 plies
```

Figure 9: Validation result using backend of OFMC.

the bandwidth among ISEs is 1 Mbps. The transaction size was between 1 and 3 KB. The experiment result shows that it is approximately 1 s to perform one transaction. This implies that the delay of the blockchain operation would be smaller compared to the secure NDN. Hence, in this section, we would primarily focus on the performance evaluation of the secure NDN.

The NDN simulator, named ndnSIM [38], is extended to simulate the secure NDN network of the proposed CLDS-B scheme. Moreover, the secure schemes of the AHISM-B, the HISM-B scheme, the HISM-B scheme, and the classic NDN scheme would also been simulated as the contrast schemes.

7.1. Network Topology and Configuration. The Abilene network topology [39] is used in our simulation, as shown in Figure 10. In the Abilene network, there are 12 routers with bi-directional links between the routers. We consider a network system with multiple producers and consumers, which are interconnected by some routers. In detail, the routers with the minimal number of links are selected to connect to four consumers respectively. In addition, the two routers with the second minimal number of links are selected to connect to two producers, respectively.

The network parameters are configured as follows: The propagation delay of one hop is d milliseconds, and the bandwidth is b Mbps. Two producers would publish Data packets independently. The population of the Data packet would follow a Zipf-Mandelbrot distribution with parameters q and s and the number of the Data packet published by each producer would be o packets. The cache size of each router is measured directly by the number of packets. It is configured to c packets. The arrival rate of Interest packets at all consumers follows a Poisson distribution with a mean value of r packets per second. The value of r would change in the simulation experiments to show the network performance at different traffic intensities. The data name and the

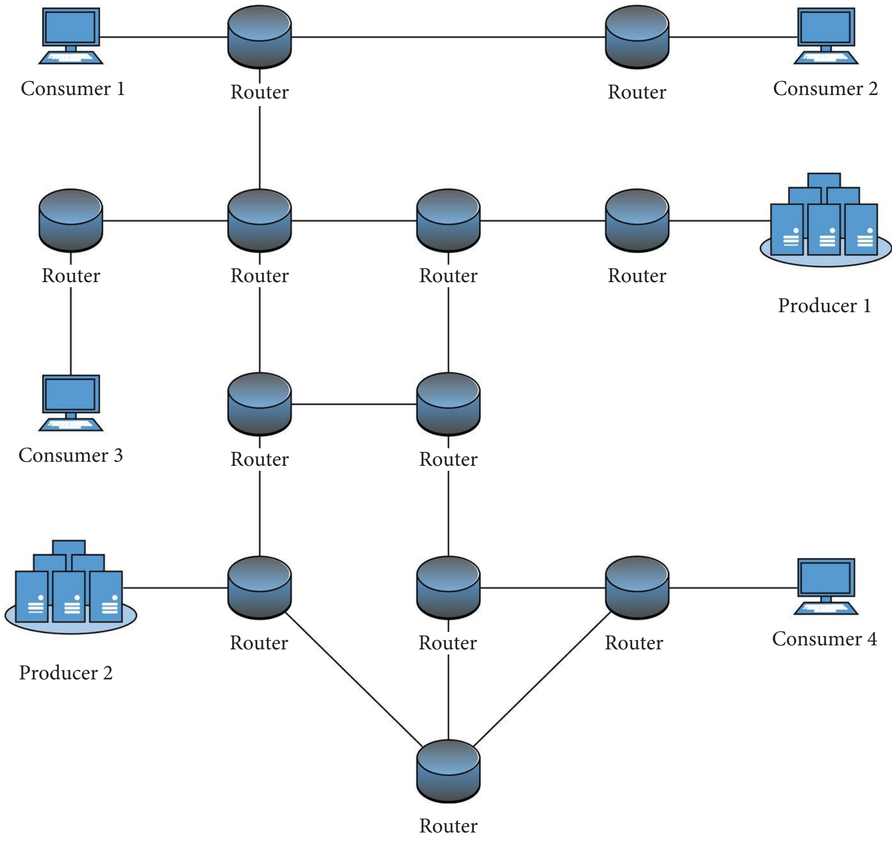


FIGURE 10: Network topology.

TABLE 2: Network parameters.

Parameter	Value
t	30
d	1
b	1
o	1,000
q	0.7
s	0.7
c	200
l_i	27
l_d	1,024

data encapsulated in the packet are simply considered to have a fixed length, so that the Interest packets and the Data packet have a fixed length. In the simulations, the length of the Interest packet is l_i bytes, and the length of the Data packet is l_d bytes. The simulation time of the experiment is t minutes. The parameter value is shown in Table 2.

7.2. Parameter Evaluation. In the secure NDN, the evaluation for the signature delay and the verification delay of the Data packets are critical since they are important components of the response delay of the Interest packets. The signature delay is the time required for the producer to sign a Data packet, while the verification delay is the time used by the consumer to verify the Data packet.

The evaluation of the signature delay and the verification delay is dependent on the cryptographic algorithm. In our evaluation, the AHISM-B scheme and the HISM-B scheme are assumed to use the HESS algorithm [25] to calculate and verify the producer's signature. With the HESS algorithm, the length of *param* and the length of the private key are both 128 bytes. The classic NDN scheme uses the elliptic curve digital signature algorithm (ECDSA) with a key size of 571 bits. The CLDS-B scheme uses Jia's algorithm [40] that is based on the ECC with its key length of 256 bits. All the above secure schemes would choose the secure hash algorithm SHA-256 to calculate the message digest. The HISM-B scheme also employs the RSA algorithm and the Message-Digest 5 (MD5) algorithm for its domain signature. The HESS algorithm, the RSA algorithm, the ECDSA algorithm, and Jia's algorithm are implemented over the operating system of Ubuntu 16 with the compiler of GCC to sign and verify the multiple Data packets.

The average signature and verification delay per packet in the four secure schemes are shown in Table 3. On the one hand, it is clear that the CLDS-B scheme has the shortest signature delay. On the other hand, the CLDS-B scheme consumes less verification delay than the classic NDN scheme and longer verification delay than both the AHISM-B scheme and the HISM-B scheme. Although it does not have the smallest verification delay, the CLDS-B scheme would present higher lever security. The ISE, which plays the role of the KGC, only generates a partial private key for the producers,

Table 3: Time parameters.

Scheme name	Signature delay (ms)	Verification delay (ms)
AHISM-B	4.492	3.465
HISM-B	4.574	3.522
Classic NDN	3.181	5.805
CLDS-B	1.296	4.966

Table 4: Average number of satisfied interest packet.

Scheme name	Average number of satisfied interest packets at each consumer				
	$r = 10$	$r = 20$	$r = 30$	$r = 40$	$r = 50$
CLDS-B	18,031	35,990	53,854	71,783	89,704
AHISM-B	18,032	35,990	53,860	71,788	89,707
HISM-B	18,032	35,990	53,859	71,786	89,704
Classic NDN	18,031	35,990	53,850	71,778	89,685
	$r = 60$	$r = 70$	$r = 80$	$r = 90$	$r = 100$
CLDS-B	107,659	125,525	143,314	161,233	179,060
AHISM-B	107,680	125,543	143,363	161,280	179,102
HISM-B	107,680	125,543	143,363	161,280	179,102
Classic NDN	107,650	125,497	143,291	161,196	179,024
	$r = 110$	$r = 120$	$r = 130$	$r = 140$	$r = 150$
CLDS-B	197,000	215,037	233,034	250,789	268,709
AHISM-B	197,057	215,127	233,125	250,899	268,817
HISM-B	197,066	215,121	233,146	250,882	268,804
Classic NDN	196,943	215,018	232,986	250,708	268,631

and it does not know the user's full private key. The producers' private keys are only known to the producers, avoiding the key escrow problem, which allows the CLDS-B to achieve the true nonrepudiation.

7.3. Performance. The performance of the CLDS-B scheme, the AHISM-B scheme, the HISM-B scheme, and the classic NDN scheme would be shown in this section. Our performance metrics mainly include the average number of Interest packets satisfied and the average response delay.

On one hand, the number of satisfied Interest packets is the number of Interest packets that have been replied by the requested Data packets. Therefore, the average number of satisfied Interest packets is the average value of the number of satisfied Interest packets at each consumer. On the other hand, the response delay refers to the time required from the moment to send an Interest packet at the consumer to the moment to receive a verified Data packet at the consumer. Therefore, the average response delay is the average value of the response delay at each consumer.

The average number of satisfied Interest packets is impacted by the average arrival rate of Interest packets, as shown in Table 4. The differences among the four schemes are not large at the low arrival rate of Interest packets. In detail, the number of satisfied Interest packets by the CLDS-B is slightly larger than that of the classic NDN scheme but slightly smaller than that of the AHISM-B scheme and HISM-B scheme. With the increase of the arrival rate, the differences become larger. The reason can be explained by the differences of the verification delay as follows: All Data packets must be verified at the consumer. If the verification delay is large, it could cause Data packets to be queued at the consumer. If the Date packet queue is long enough, some Data packets would be dropped directly. When the arrival rate of Interest packets becomes large, the packet loss rate would increase. Once the Data packets are dropped, the Interest packets need to be retransmitted to request the lost Data packets, which would aggravate the network congestion further. In summary, the larger verification delay is, the less average number of satisfied Interest packets. The verification delay of the CLDS-B is less than that of the classic NDN and is larger than that of the AHISM-B and the HISM-B. Therefore, the average number of satisfied Interest packets of the CLDS-B is larger than that of the classic NDN and is less than that of AHISM-B and the HISM-B.

The average response delay at each consumer is shown in Figures 11–14. As we expected, the curve of the CLDS-B is always below the cure of the classic NDN. It indicates that the CLDS-B has a shorter average response delay than that of the classic NDN scheme at all arrival rates of Interest packets. It is because the signature delay and verification delay of the CLDS-B scheme are much lower than those of the classic NDN scheme. With the increase of the arrival rate, the gap between the CLDS-B curve and the classic NDN curve is widened. It is because the longer signature delay and verification delay would cause a longer queue delay. When the arrival rate is larger, the queue delay would be aggravated. As a result, the CLDS-B would express a greater advantage compared to the classic NDN.

However, the curve of the CLDS-B intersects with both the curve of the AHISM-B and the curve of the HISM-B in Figures 11–13, and there is one intersection in Figure 14. Although it could maintain the shorter response delay when the arrival rate of Interest packets is less, the CLDS-B would show a longer response delay, unfortunately, with the increase of the arrival rate of Interest packets. The response delay is mainly composed by five types of the delay, including the propagation delay, the signature delay, the verification delay, the queue delay, and the retransmission delay. Since the four schemes share the same network topology and the bandwidth, the propagation delay in the four schemes would be very similar. When the arrival rate of the Interest packet is low, the queue delay and the retransmission delay would be zero or very little, so that the response delay is determined by the verification delay and the signature delay mainly. The signature delay of the CLDS-B is much smaller than that of the AHISM-B and HISM-B, and the verification delay of CLDS-B is a little larger than that of the AHISM-B and HISM-B. Although the number of verifications at consumers would be more than the number of signatures at the producer for a Data packet due to the router cache, the signature delay still has the most significant impact on the response delay. As a result, in the low arrival rate, the CLDS-B would express a smaller response delay than the AHISM-B and the HISM-B in Figures 11–13, and it would have a smaller response delay than the HISM-B in Figure 14.

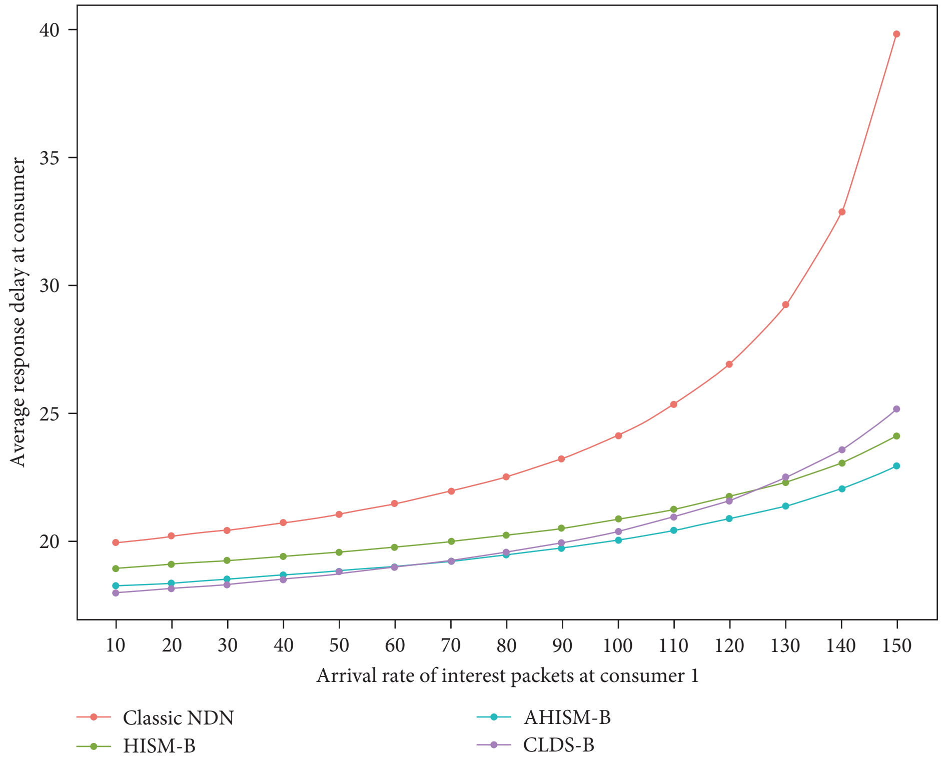


Figure 11: Average response delay at Consumer 1.

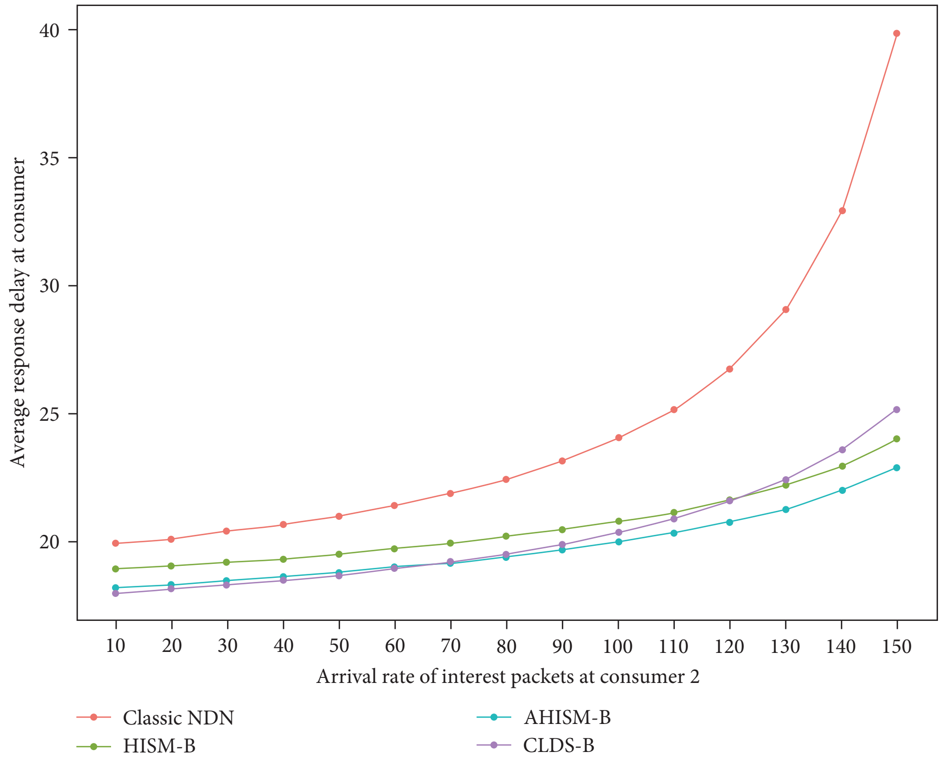


Figure 12: Average response delay at Consumer 2.

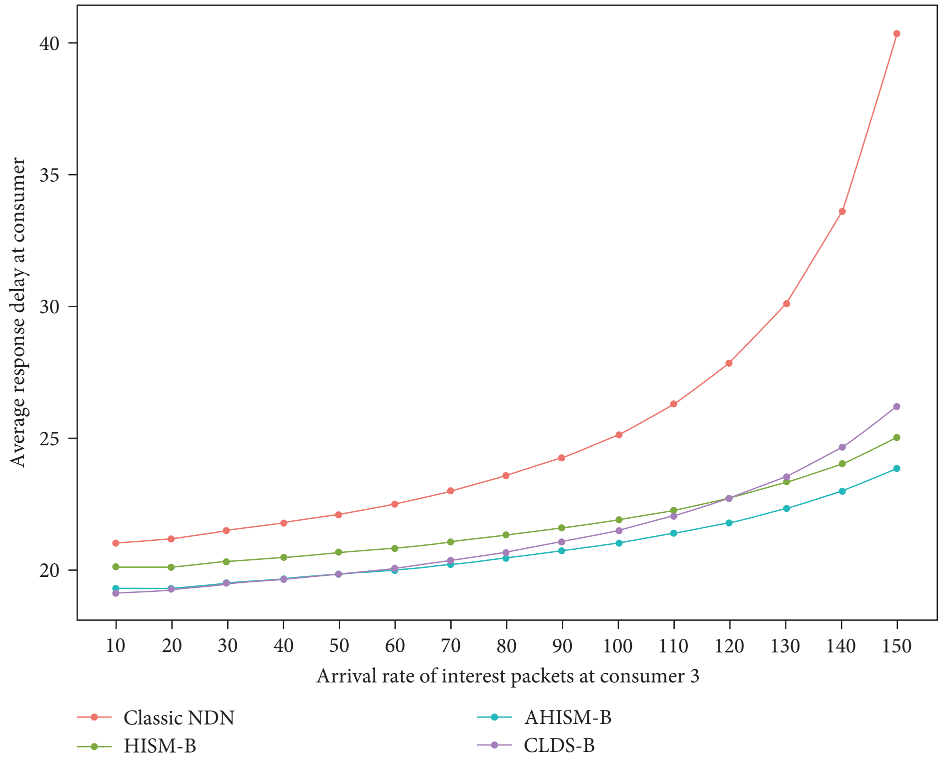


Figure 13: Average response delay at Consumer 3.

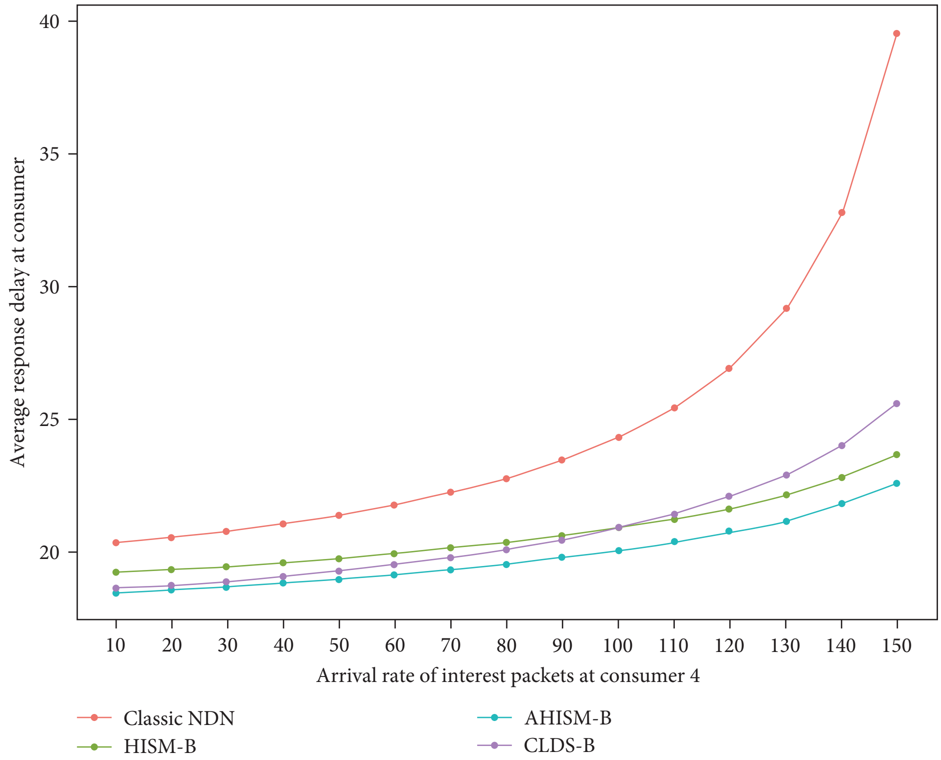


Figure 14: Average response delay at Consumer 4.

With the increase of the arrival rate, the queue delay would become large. In the NDN, routers can cache Data packets to alleviate the load of producers so that almost no queues would appear at producers. Therefore, the queue delay mainly focuses on the queue delay at consumers. The larger the verification delay is, the longer the queue would be at consumers. Since the verification delay of the CLDS-B is a little larger than that of the AHISM-B and HISM-B, the queue delay of the CLDS-B would be a little larger. Therefore, the response delay of the CLDS-B would be a little larger. When the arrival rate increases further, the queue would overflow, so the retransmission would be incurred. The retransmission delay causes the response delay gap among the three schemes to be a little bigger.

In conclusion, the proposed CLDS-B scheme performs better than the classic NDN scheme and shows slight inferiority to the AHISM-B scheme and the HISM-B scheme in terms of the number of satisfied Interest packets and the response delay. However, it has been verified and analyzed to provide stronger security to solve the key escrow problem. As a result, it would be a competitive solution in scenarios with a high-security level.

8. Conclusions

This paper proposes a new secure solution for NDN named Secure Mechanism supported by Certificateless Digital Signature and Blockchain (CLDS-B). The proposed scheme utilizes certificateless digital signatures to provide the data-oriented trust, ensuring identity verification and data integrity while eliminating key escrow issues. The use of blockchain technology allows for the management of encrypted information and enhances security through decentralization. The CLDS-B scheme outperforms the classical NDN scheme in terms of security and network performance, and it is competitive with other schemes. This proposed scheme is well-suited for high-security level scenarios and represents a competitive choice for such environments.

Data Availability

All datasets used in this submission are publicly available and can be accessed through the references cited in the text.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Mingxuan Zheng performed the experimental operations, Bing Li performed the manuscript writing, and Maode Ma reviewed the manuscript and provided improvement suggestions.

References

[1] L. Zhang, A. Afanasyev, J. Burke et al., "Named data networking," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 3, pp. 66–73, 2014.

[2] S. Mastorakis, A. Mtibaa, J. Lee, and S. Misra, "*ICedge*: when edge computing meets information-centric networking," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4203–4217, 2020.

[3] E. Bertino and M. Nabeel, "Securing named data networks: challenges and the way forward," in *SACMAT '18: Proceedings of the 23nd ACM on Symposium on Access Control Models and Technologies*, pp. 51–59, Association for Computing Machinery, Indianapolis, Indiana, USA, June 2018.

[4] D. Freet and R. Agrawal, "An overview of architectural and security considerations for named data networking (NDN)," in *MEDES: Proceedings of the 8th International Conference on Management of Digital EcoSystems*, pp. 52–57, Association for Computing Machinery, Biarritz, France, November 2016.

[5] M. E. Hellman, "An overview of public key cryptography," *IEEE Communications Magazine*, vol. 40, no. 5, pp. 42–49, 2002.

[6] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Advances in Cryptology. CRYPTO 1984. Lecture Notes in Computer Science*, pp. 47–53, Springer, Berlin, Heidelberg, 1985.

[7] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *Advances in Cryptology - ASIACRYPT 2003. ASIACRYPT 2003. Lecture Notes in Computer Science*, vol. 2894, pp. 452–473, Springer, Berlin, Heidelberg, 2003.

[8] B. Li and M. Ma, "An advanced hierarchical identity-based security mechanism by blockchain in named data networking," *Journal of Network and Systems Management*, vol. 31, Article ID 13, 2023.

[9] C. Bian, Z. Zhu, A. Afanasyev, E. Uzun, and L. Zhang, *Deploying Key Management on NDN Testbed*, UCLA, Peking University and PARC, NDN, Technical Report, 2013.

[10] B. Li, M. Ma, and R. Xia, "Hierarchical identity-based security mechanism using blockchain in named data networking," in *2020 3rd International Conference on Hot Information-Centric Networking (HotICN)*, pp. 148–153, IEEE, Hefei, China, December 2020.

[11] Y. Yu, *Public Key Management in Named Data Networking*, California, Los Angeles, CA, USA, NDN, Technical Report NDN-0029, 2015.

[12] T. Yu, H. Xie, S. Liu, X. Ma, X. Jia, and L. Zhang, "CertRevoke: a certificate revocation framework for named data networking," in *ICN '22: Proceedings of the 9th ACM Conference on Information-Centric Networking*, pp. 80–90, Association for Computing Machinery, Osaka, Japan, September 2022.

[13] Z. Zhang, A. Afanasyev, and L. Zhang, "NDNCERT: universal usable trust management for NDN," in *ICN '17: Proceedings of the 4th ACM Conference on Information-Centric Networking*, pp. 178-179, Association for Computing Machinery, Berlin, Germany, September 2017.

[14] P. F. Tehrani, E. Osterweil, T. C. Schmidt, and M. Wählisch, "SoK: public key and namespace management in NDN," in *ICN '22: Proceedings of the 9th ACM Conference on Information-Centric Networking*, pp. 67–79, Association for Computing Machinery, Osaka, Japan, September 2022.

[15] B. Hamdane, A. Serhrouchni, A. Fadlallah, and S. G. El Fatmi, "Named-data security scheme for named data networking," in *2012 Third International Conference on The Network of the Future (NOF)*, pp. 1–6, IEEE, Tunis, Tunisia, November 2012.

[16] R. Li, H. Asaeda, J. Li, and X. Fu, "A distributed authentication and authorization scheme for in-network big data sharing," *Digital Communications and Networks*, vol. 3, no. 4, pp. 226–235, 2017.

[17] S. S. Ullah, I. Ullah, H. Khattak et al., "A lightweight identity-based signature scheme for mitigation of content poisoning

attack in named data networking with internet of things," *IEEE Access*, vol. 8, pp. 98910–98928, 2020.

[18] B. Li, M. Ma, Y. Zhang, and F. Lai, "Access control supported by information service entity in named data networking," in *2022 5th International Conference on Hot Information-Centric Networking (HotICN)*, pp. 30–35, IEEE, Guangzhou, China, November 2022.

[19] K.-H. Yeh, C. Su, K.-K. R. Choo, and W. Chiu, "A novel certificateless signature scheme for smart objects in the internet-of-things," *Sensors*, vol. 17, no. 5, Article ID 1001, 2017.

[20] A. Karati, S. H. Islam, and M. Karuppiah, "Provably secure and lightweight certificateless signature scheme for IIoT environments," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3701–3711, 2018.

[21] Y. Zhang, R. H. Deng, D. Zheng, J. Li, P. Wu, and J. Cao, "Efficient and robust certificateless signature for data crowdsensing in cloud-assisted industrial IoT," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 9, pp. 5099–5108, 2019.

[22] A. Muhammad, N. U. Amin, I. Ullah et al., "An efficient scheme for industrial internet of things using certificateless signature," *Mathematical Problems in Engineering*, vol. 2021, Article ID 9960264, 11 pages, 2021.

[23] X. Chen, D. He, M. K. Khan, M. Luo, and C. Peng, "A secure certificateless signcryption scheme without pairing for internet of medical things," *IEEE Internet of Things Journal*, vol. 10, no. 10, pp. 9136–9147, 2023.

[24] X. Yang, H. Wen, R. Diao, X. Du, and C. Wang, "Improved security of a pairing-free certificateless aggregate signature in healthcare wireless medical sensor networks," *IEEE Internet of Things Journal*, vol. 10, no. 12, pp. 10881–10892, 2023.

[25] E. F. Cahyadi and M.-S. Hwang, "A comprehensive survey on certificateless aggregate signature in vehicular Ad Hoc networks," *IETE Technical Review*, vol. 39, no. 6, pp. 1265–1276, 2022.

[26] A. Imghoure, A. El-Yahyaoui, and F. Omary, "ECDSA-based certificateless conditional privacy-preserving authentication scheme in Vehicular Ad Hoc network," *Vehicular Communications*, vol. 37, Article ID 100504, 2022.

[27] Y. Wu, H. Xiong, and C. Jin, "A multi-use unidirectional certificateless proxy re-signature scheme," *Telecommunication Systems*, vol. 73, pp. 455–467, 2020.

[28] Y. H. Zhou, S. S. Dong, and Y. G. Yang, "A unidirectional certificateless proxy re-signature scheme based on lattice," *Transactions on Emerging Telecommunications Technologies*, vol. 33, no. 4, Article ID e4412, 2022.

[29] S. Bouakkaz and F. Semchedine, "A certificateless ring signature scheme with batch verification for applications in VANET," *Journal of Information Security and Applications*, vol. 55, Article ID 102669, 2020.

[30] Y. Ren, X. Li, S.-F. Sun, X. Yuan, and X. Zhang, "Privacy-preserving batch verification signature scheme based on blockchain for Vehicular Ad-Hoc networks," *Journal of Information Security and Applications*, vol. 58, Article ID 102698, 2021.

[31] L. Cao, Y. Liu, and S. Cao, "An authentication protocol in LTE-WLAN heterogeneous converged network based on certificateless signcryption scheme with identity privacy protection," *IEEE Access*, vol. 7, pp. 139001–139012, 2019.

[32] L. Chen, K. Zhang, S. Kumari, M. K. Khan, Z. Kong, and P. Chaudhary, "An efficient certificateless key exchange protocol for heterogeneous networks in human-centered IoT systems," *International Journal of Communication Systems*, vol. 36, no. 12, Article ID e4093, 2023.

[33] H. Huang, Y. Wu, F. Xiao, and R. Malekian, "An efficient signature scheme based on mobile edge computing in the NDN-IoT environment," *IEEE Transactions on Computational Social Systems*, vol. 8, no. 5, pp. 1108–1120, 2021.

[34] S. Hussain, S. S. Ullah, A. Gumaei, M. Al-Rakhami, I. Ahmad, and S. M. Arif, "A novel efficient certificateless signature scheme for the prevention of content poisoning attack in named data networking-based internet of things," *IEEE Access*, vol. 9, pp. 40198–40215, 2021.

[35] S. S. Ullah, S. Hussain, A. Gumaei et al., "A cost-effective approach for NDN-based internet of medical things deployment," *Computers, Materials & Continua*, vol. 70, no. 1, pp. 233–249, 2022.

[36] C. M. Rao, G. Prasuna, H. K. Chapala, N. Jeebaratnam, D. Navulla, and A. Verma, "Designing a reliable and cost-effective internet of medical things (IoMT) topology to minimize the maintenance and deployment cost," in *2023 Fifth International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, pp. 1–7, IEEE, Erode, India, February 2023.

[37] Q. Nasir, I. A. Qasse, M. Abu Talib, and A. B. Nassif, "Performance analysis of hyperledger fabric platforms," *Security and Communication Networks*, vol. 2018, Article ID 3976093, 14 pages, 2018.

[38] S. Mastorakis, A. Afanasyev, and L. Zhang, "On the evolution of ndnSIM: an open-source simulator for NDN experimentation," *ACM SIGCOMM Computer Communication Review*, vol. 47, no. 3, pp. 19–33, 2017.

[39] Y. Zhang, "Abilene traf_c matrices," 2015, http://www.cs.utexas.edu/users/yzhang/research/AbileneTM.

[40] 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.