Journal of Network and Computer Applications 237 (2025) 104135

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

Research paper

BAS-NDN: BlockChain based mobile producer authentication scheme for Named Data Networking

Guangquan Xu [a,b], Chenghe Dong [a], Cong Wang [c,*], Feng Feng [d]

[a] Tianjin Key Laboratory of Advanced Networking (TANK), College of Intelligence and Computing, Tianjin University, Tianjin, 300354, China
[b] School of Big Data, Qingdao Huanghai University, Qingdao, 266555, China
[c] College of Artificial Intelligence, Tianjin University of Science and Technology, Tianjin, 300457, China
[d] School of Information Engineering, Ningxia University, Yinchuan, 750021, China

ARTICLE INFO

Keywords:
Named Data Network
Producer mobility
Prefix authentication
Certificateless signcryption
Security

ABSTRACT

Named Data Network (NDN) is a content-centric, name-based communication architecture, with a push-based communication model naturally supports consumer mobility. However, the management of producer prefix authentication during mobility is challenging due to NDN's name-based mechanism, which facilitates direct interaction between producers and the forwarding plane. The current solutions fail to balance security and efficiency. To address insecure interactions arising from producer mobility, we introduce a protocol for blockchain-based mobile producer authentication (BAS-NDN). Our protocol relies on a novel elliptic curve-based certificateless signcryption scheme, which is easy to deploy, provides both signature and encryption, and avoids complex certificate management and key escrow problems. This makes it suitable for secure and efficient mobile management in NDN. In addition, the proposed scheme efficiently authenticates the producer's prefixes by enforcing the producer to publish routing updates that use only valid prefixes. This design renders it resistant to prefix hijacking attacks. Through analyzing under the random oracle model, it is also resistant to both Type I and Type II adversaries present in certificateless signcryption. Finally, experimental analysis indicates that our scheme provides significant performance benefits.

1. Introduction

Named Data Networking (NDN) is an emerging Internet paradigm that addresses the requirements of future networks, including 5G, by catering to various wireless access technologies such as WiFi and Zigbee. Unlike current Internet Protocol (IP) networks (Zhang et al., 2014) that protect data containers, NDN ensures content security by naming the data instead of relying on location, transforming it into a first-class entity. This design separates the user's trust in the data from the host, thereby overcoming the shortcomings of traditional IP networks, including insufficient bandwidth, security vulnerabilities, non-scalability, and mobility challenges (Jacobson, 2006). In particular, NDN provides network-layer mobility support by separating the temporal and spatial aspects between request resolution and content delivery. Communication in NDN is triggered by consumers interested in specific content. Subsequently, NDN forwards the Interest to producers, who generate the relevant data with designated name prefixes and return it to consumers.

In NDN, mobility is categorized into two types: consumer mobility, representing the content requester, and producer mobility, signifying the content provider. Mobility is defined as changing location and moving to a new Point of Attachment (PoA). The NDN communication model is consumer-driven, with a request/response mechanism between the consumer and producer that is connectionless. Therefore, NDN naturally encourages seamless consumer mobility (Fang et al., 2018). However, during producer mobility, content distribution turns into a challenging task due to the tight integration of routing locators and content identifiers. Complications may arise when the producer moves between different PoAs, ultimately compromising the functionality of the communication network. The currently proposed tracking-based approach is more suitable for addressing the producer mobility issue since it provides seamless movement of the producer through the continuous updating of the forwarding table for every mobility event.

In trace-based schemes, producer mobility in NDN is provided by the stateful forwarding plane, ensuring low switching latency, minimal signaling, and mitigating packet loss. However, trace-based protocols enable producers to insecurely interact with routes, which leads to significant security risks to all network entities, including consumers,

* Corresponding author.
E-mail addresses: chenghe_d@163.com (C. Dong), wangcongjcdd@tust.edu.cn (C. Wang).

https://doi.org/10.1016/j.jnca.2025.104135
Received 20 October 2024; Received in revised form 11 December 2024; Accepted 3 February 2025
Available online 11 February 2025

producers, and the network itself. Therefore, protocols should only allow producers to publish legitimate routing updates, denoted as Interest Updates (IUs), thereby only valid prefixes are authorized to publish relevant content. Networks lacking robust security measures are susceptible to breaches by adversaries, who can forge IUs of legitimate producers to launch prefix hijacking attacks, deliver false information in the network, and possibly deny consumers access to requested content, or even perform black hole attacks to harm users.

Blockchain (BC), as a distributed system, can ensure privacy security and access control for applications like networks and data storage (Sultan et al., 2024; Xia et al., 2024). Moreover, plenty of authentication schemes with privacy-preserving have been designed, they are rooted in traditional public key cryptography (PKC) or identity-based cryptography (IBC). However, PKC and IBC encounter challenges such as complex certificate management and key escrow problems, respectively. To overcome these problems, Al-Riyami and Paterson (2003) introduced the concept of certificateless cryptography (CLC), wherein a user's private key is a combination of key generation center (KCG) and the user itself. Therefore, we propose a secure and efficient BC-based distributed producer mobile certificateless authentication scheme aimed at resolving the secure mobility issue within NDN. This scheme authenticates the producers' prefixes, ensuring that they exclusively publish genuine and valid IUs.

1.1. Related work

The centralized prefix authentication used in IP-based mobility protocols is no longer applicable for NDN, i.e., a central entity is required to authenticate when a host changes its PoAs. In 2000, Das et al. (2000) proposed using a single network key to generate a host's session key. However, this suffers from a single point-of-failure problem and fails to identify malicious routers and users attempting to launch denial or replay attacks. Huston et al. (2011) also discussed prefix hijacking attacks in inter-domain and intra-domain IP routing respectively. Digital signatures and certificates are a common mechanism for authenticating IP prefixes, where the address owner solicits a signed certificate to secure the authorization to publish the IP address. This also applies to the trace-based NDN mobility protocol, however, it is also subject to denial or replay attacks.

To avoid the above attack, Compagno et al. (2017) proposed a distributed prefix authentication protocol that employs a unidirectional hash chain to ensure producer mobility in NDN and resist prefix hijacking attacks. In addition, to achieve forward security, producers are forced to maintain synchronized hash chain values with the network to verify prefix identity. However, their protocol fails to ensure that each router has the latest hash chain value, and cannot prevent prefix hijacking attacks, Dos and replay attacks. In 2017, Kim and Ko (2017) proposed an anchor-based mobility support method, which uses anchor nodes to forward an updated packet about the generated prefix, but the inefficient forwarding path of this scheme increases the delay and interest packet loss. Ren et al. (2016) introduced Software Defined Controller (SDC) to support producer mobility, however, their method delays the packet transmission and creates bottlenecks in SDC. In the same year, Gao and Zhang (2016) introduced a scalable mobility management scheme for NDN that utilizes Binding Update and Binding Acknowledgment to notify the server about producers' location. However, global mapping server updates may cause problems such as high bandwidth, interest packet loss or retransmission. Later, Farahat and Hassanein (2017) proposed an active caching scheme for NDN producer mobility management, which optimally caches content for interest packet matching. However, interest packets are lost or retransmitted in real-time communication. In 2018, Gohar et al. (2018) proposed a cluster-based approach to device mobility management, however, the approach increases signaling and bandwidth usage, incurring content delivery delays. In the same year, Rui et al. (2018) proposed a real-time delivery scheme that caches the content of interest and uses a special data packet to get the name of the next route. However, this packet increases signaling and bandwidth. Recently, Hlaing et al. (2021) proposed an identity-based proxy re-encryption scheme that allows content injection and revocation of malicious users. Dulal et al. (2022) proposed a scheme that implements fine-grained access control in NDN. However, both schemes suffer from the key escrow problem and lack attention to producer mobility.

In recent year, several studies use diversity tools to support producer mobility. Yan et al. (2020) proposed a scheme to support hybrid network mobility in NDN, where binding update lists were created by the previous route and the intermediate router. However, excessive updating of producers may lead to the loss of interest packets and increase the lookup time. Later, Kar et al. (2022) proposed an NDN producer mobility management technique, which was applied to telemedicine systems, inspiring a wide range of IoT applications. In 2023, Khalid et al. (2023) proposed an SDN controller-based protocol to address mobility problems. The SDN controller can monitor the network topology of each node and share route updates with other nodes during mobility events. However, this scheme requires a significant communication overhead to track the frequent changes in the location of mobile nodes.

Most of the existing schemes to support NDN producer mobility are flawed, as demonstrated by the above schemes. Table 1 briefs the method and problems in supporting producer mobility programs. Therefore, designing a lightweight producer authentication scheme that is resistant to prefix hijacking attacks, to fulfill the requirement of secure producer mobility in NDN, remains a significant challenge.

1.2. Motivation & Contribution

Due to the tight integration of routing location fits and content identifiers in NDN, it imposes a natural challenge to manage producer mobility. Several solutions have been proposed to support producer mobility, however, most of them fail to balance security and efficiency. For example, both Das et al. (2000) and Huston et al.'s (2011) schemes are not resistant to replay attacks, and Compagno et al. (2017) is not resistant to prefix hijacking attacks; Yan et al. (2020), Kar et al. (2022), and Khalid et al.'s (2023) schemes suffer from packet loss and high bandwidth problems. In addition, no specific authentication scheme is given in the blockchain-based solution proposed by Conti et al. Since the certificateless signcryption technique does not require certificates and simplifies the management of secure communication in NDN, it ensures a secure and efficient producer authentication scheme. However, there are two types of adversaries inherent in the approach: Type I and Type II. Type I adversary is a malicious producer that is able to replace the public key without the master key. Type II adversary is a malicious KGC that possesses the master key but is unable to replace the public key. Most of the existing certificateless signcryption schemes (Gong et al., 2022; Chen et al., 2022, 2023; Dai and Xu, 2023) cannot resist Type I or Type II adversaries. In Gong et al. (2022), Gong et al. proposed a certificateless signcryption scheme using bilinear map, but it cannot resist Type I attacks and failure to provide unforgeability. Chen et al. (2022) presented a certificateless online/offline signcryption scheme, which reduces computational costs. However, this scheme also is insecure against Type I attacks. Later, Chen et al. (2023) and Dai and Xu (2023) proposed certificateless signcryption schemes without pairing for medical IoT and vehicle sensor networks, respectively. However, both schemes are susceptible to Type I attacks, which allow the adversary to forge a valid ciphertext without secret key. Therefore, designing a secure and efficient producer authentication scheme in NDN remains a challenge.

In order to design a certificateless signcryption scheme for mobile producer authentication in NDN, the primary contributions of this study are as follows:

Table 1
Schemes supporting produce mobility.

Scheme	Method	Drawback
Das et al. (2000)	Uses a single network key to generate a host's session key.	Single point-of-failure problem, fails to identify malicious routers, and suffers denial or replay attacks.
Compagno et al. (2017)	Employs a unidirectional hash chain to ensure producer mobility.	Fails to ensure that each router has the latest hash chain value, and cannot prevent prefix hijacking attacks, Dos and replay attacks.
Kim and Ko (2017)	Uses anchor nodes to forward an updated packet.	Increases delay and interest packet loss.
Ren et al. (2016)	Introduces Software Defined Controller (SDC).	Delays packet transmission and creates bottlenecks in SDC.
Gao and Zhang (2016)	Uses Binding Update and Binding Acknowledgment to notify the server about producers' location.	High bandwidth, interest packet loss or retransmission.
Farahat and Hassanein (2017)	Optimally caches content.	Interest packets are lost or retransmitted.
Gohar et al. (2018)	Cluster-based approach to device mobility management.	Increases signaling and bandwidth usage.
Rui et al. (2018)	Uses a special data packet to get the name of the next route.	Increases signaling and bandwidth.
Hlaing et al. (2021)	Allows content injection and revocation of malicious users.	Suffer from the key escrow problem and lack attention to producer mobility.
Dulal et al. (2022)	Implements fine-grained access control.	Suffer from the key escrow problem.
Yan et al. (2020)	Creats binding update lists	Excessive updating of producers lead to the loss of interest packets and increase the lookup time.
Khalid et al. (2023)	SDN controller monitors the network topology of each node and shares route updates	Requires a significant communication overhead

Table 2
List of acronyms.

Abbreviations	Full names
AS	Authorization Server
BC	Blockchain
BSA-NDN	BlockChain based mobile producer Authentication Scheme for Named Data Networking
CH	Cluster Head
CLC	Certificateless Cryptography
CS	Content Store
ECC	Elliptic Curve Cryptography
ECDLP	Elliptic Curve Discrete Logarithm Problem
EUF-CLSC-CMA	Unforgeability under adaptively Chosen Message Attacks in Certificateless Signcryption
FIB	Forwarding Interest Base
GBAs	Global Blockchain Administrators
IBC	Identity-Based Cryptography
ICN	Information-Centric Networking
IND-CLSC-CCA2	Indistinguishable under adaptively Chosen Ciphertext Attacks in Certificateless Signcryption
IP	Internet Protocol
IU	Interest Update
KGC	Key Generation Center
LIA	Local Immutable Ledger Administrator
LIL	Local Immutable Ledger
NDN	Named Data Networking
P	Mobile Producer
PIT	Pending Interest Table
PKC	Public Key Cryptography
PoA	Point of Attachment
SDC	Software Defined Controller

- We propose a new lightweight BC-based authentication certificateless scheme (BAS-NDN) that ensures the security of producer mobility in a distributed way. In particular, BAS-NDN is a novel solution that utilizes blockchain to provide secure and reliable authentication for producer movement in NDN, and we give the initial syntactic and security model for BAS-NDN.
- The scheme exploits the features of a time-based consensus algorithm, distributed trust, and throughput management in BC for secure and efficient producer switching within NDN. Moreover, our scheme is proven to be secure against Type I and Type II adversaries under the random oracle model, which relies on the elliptic curve discrete logarithm problem (ECDLP).
- We analyze the security of BAS-NDN against prefix hijacking attacks. Theoretical analysis and experimental evaluation show that our scheme performs comparably to most existing hash chain-based prefix attestation schemes. Our scheme provides excellent security with minimal computational and communication overheads.

1.3. Organization

The structure of this paper is as follows. Section 2 provides key concepts that are essential for BAS-NDN. Section 3 presents the system model and adversary Model. In Section 4, we introduce our scheme. Section 5 and Section 6 present the security analysis and performance analysis of the system, respectively. Finally, the conclusion of our work is summarized in Section 7.

2. Preliminaries

We briefly review the relevant notions that are fundamental to our work in this section. And Table 2 is given to make general readers read much easily.

2.1. NDN overview

The content-centric nature of NDN is gaining significant attention. Each content is uniquely identified by a name instead of IP, allowing users to focus on the content rather than the physical location of the network (Xylomenos et al., 2021). Unlike IP-based networks, which suffer from inefficiency, limited scalability, and security vulnerabilities, NDN addresses these issues effectively. NDN communication involves

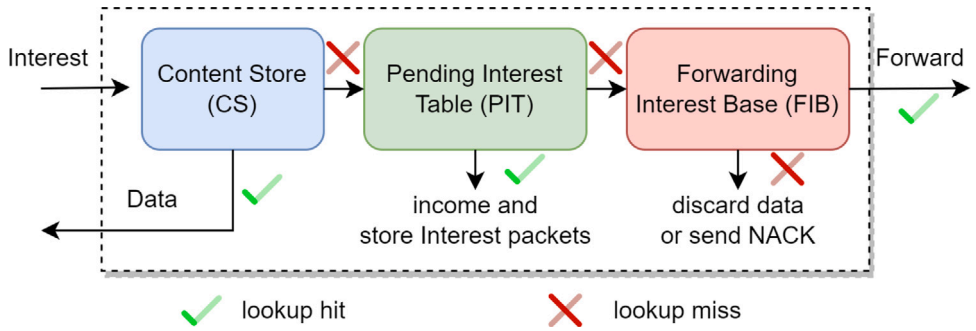

Fig. 1. Interest and data processing in NDN.

two primary packet types: Interest packets, initiated by consumers to request content, and Data packets, transmitted by producers in response (Li and Ma, 2023). To forward these packets, NDN manages three distinct data structures: the Content Store (CS), the Pending Interest Table (PIT), and the Forwarding Interest Base (FIB). As shown in Fig. 1, the CS stores cached content locally. The FIB serves as an output interface and contains message names or prefixes, which look up the most matched source of content for Interest packets. The PIT is used to income and store Interest packets awaiting responses. Due to NDN's utilization of the Publish-Subscribe Internet model, producers publish content using explicit name prefixes such as "YouTube/". NDN builds the name-based routing table, when a router receives an Interest packet (e.g., YouTube/forwarder.apk) from a consumer, it initially verifies the content already exists in the CS; if it is available, the content is forwarded immediately. Otherwise, the router searches the PIT for pending interests related to the same content and forwards the interest to the target route. Upon a PIT match, the interest of the same name is collapsed, and the router looks up the path that satisfies the request based on FIB and forwards the interest to the producer via the network interface. The packet is subsequently transmitted by following the path of interest. If the FIB still fail to locate the source name, the interest packet is discarded, resulting in the consumer receiving a Negative Acknowledgment.

The security of NDN is ensured by a content-centric model. Therefore, content is shared along with the producer's signature, and authenticated by the producer to enable the consumer to verify its integrity and validate the data source. Based on the above idea, we design a secure and efficient authentication scheme for producer mobility.

2.2. Mobility significance in NDN

Mobility is considered to be the most important component of managing NDN (Naeem et al., 2018), which allows mobile devices to change location between different PoAs seamlessly. Without supporting mobility, the core functions of NDN such as forwarding, scaling, and content retrieval, cannot be realized. However, with the popularity of mobile devices, designing an efficient mobility-enabled scheme comes with several challenges, including efficiency, routing consistency, and security (Ma et al., 2021).

NDN retrieves data based on content names rather than IP addresses, thereby mobile devices can access data without the need to repeatedly obtain an IP address. In addition, as NDN supports multiple network interfaces by sending interests across multi-attributed networks (Mars et al., 2019), consumer requests can be freely multiplexed by different interfaces. Unlike current communication oriented to TCP/IP connections (Saxena et al., 2016), communication in NDN does not require re-establishing a connection to a data source when a mobile device moves. Therefore, NDN mobility facilitates seamless device repositioning across various PoAs, ensuring minimal switching and latency without interrupting the content. This mobility can be classified into two types: consumer mobility and producer mobility.

Consumer mobility is naturally supported by NDN's consumer-driven, where the request/response model between consumers and producers is connectionless. Upon a consumer attaches to a new PoA, it is allowed to reacquire data. However, the tight coupling of route locators and content identifiers poses challenges for producer mobility. In particular, when a producer initiates a mobility event, the network should ensure that the producer is reachable, and routes need to adjust the forwarding information to relocate the interest of matching prefixes. Thus, producer mobility requires the support of a name resolution system to ensure routing consistency (Cisco, 2020). Moreover, consumers cannot track the producer's location changes, potentially resulting in blocked consumer communication. As a result, producer mobility suffers from the risks of increased overhead (Xylomenos et al., 2014), prolonged switching delay (Abrar et al., 2022) and heightened packet loss (Chen et al., 2014).

Research on producer mobility is limited at present. After producer mobility, the name prefixes in the FIB need to be updated for communication. Indirection-Based and DNS-Based techniques aim to facilitate producer mobility in NDN, which use the primary router to maintain source and destination prefixes, however, they suffer from single points of failure, high bandwidth usage, interest packet loss, and interest retransmission. Locator/Identifier Split-Based and Control Data Plane Split-Based schemes support producer mobility by modifying the interest or date packets, yet they have high signaling and bandwidth problems. Trace-Based schemes utilize the NDN stateful forwarding platform to overcome the above problems, however, they allow producers to directly interact with forwarded messages and lack robust security protocol, which may cause serious damage to all entities in NDN. Therefore, we propose a producer authentication scheme for ensuring network security during producer mobility in NDN.

2.3. Blockchain

BC (Kosba et al., 2016) is a secure shared distributed data ledger. Recently, BC has received extensive attention from experts across various fields (Dorri et al., 2017) due to its decentralized, transparent, highly autonomous, and tamper-resistant features. Within BC, transactions or datasets are packaged into cryptographic hash blocks in the chain, subsequently subjected to a consensus system to decide how to add them to the chain. Initially employed in digital currency cryptosystems, the BC technique is now gradually proving its applicability across various domains. Fotiou and Polyzos (2016) presented a distributed BC-based scheme for secure content access in Information-Centric Networking (ICN, where NDN serves as a specific implementation). In our scheme, BC is used for distributed storage and data sharing to authenticate the interactions between mobile producers and network-forwarded messages to enhance network security.

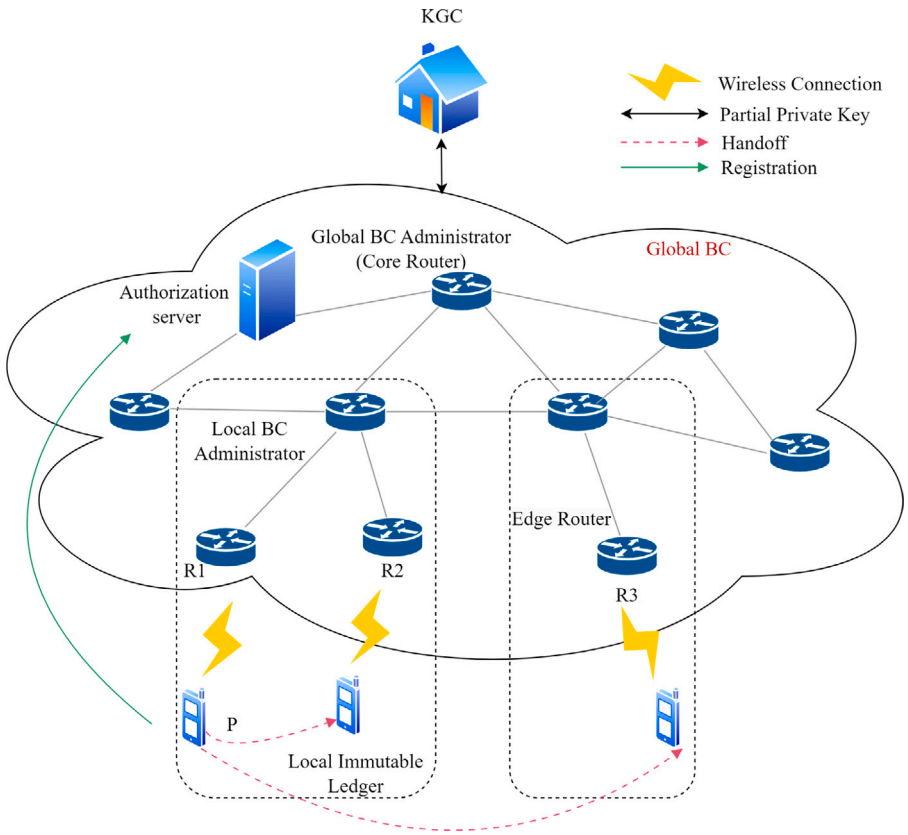

Fig. 2. BAS-NDN system model.

2.4. Mathematical theory

Elliptic Curve Cryptography (ECC). Let $E(F_p)$ represent an elliptic curve E over a finite field F_p, where defined by a prime number p. The curve is described by the equation $y^2 = x^3 + ax + b \bmod p$ satisfying $4a^3 + 27b^2 \neq 0 \bmod p$, where a and b are elements in F_p. The set of points on $E(F_p)$ include the point at infinity, denoted as $\mathcal{O}$. The group of points that forms the additive elliptic curve group G be defined as $G = \{(x, y) : y^2 = x^3 + ax + b, 4a^3 + 27b^2 \neq 0\} \cup \mathcal{O}$, where P is a generator of G. The security of ECC is based on the ECDLP, which ensures that it is difficult to compute x in the equation $Q = xP$, $x \in Z_p^*$ is a random number. This problem is considered difficult in probabilistic polynomial time (PPT) and forms the basis of ECC security.

Square Diffie–Hellman (Square-DH) Assumption. Consider a group G with a generator Q and a prime number p. Given the 2-tuple $(Q, dQ) \in G^2$, it is computationally hard to determine d^2Q, where $d \in Z_p^*$ is a random number.

3. System model and adversary model

3.1. System model

To perform fast intra- and inter-cluster switching in NDN (Hajjar et al., 2015), edge routers within a cluster use a private ledger that stores local transactions, with a structure resembling that of the BC. To meet the requirements of switching delay and scalability, as shown in Fig. 1, edge routers managed by a cluster head (CH) form a Local Immutable Ledger (LIL), and the CH is also known as the Local Immutable Ledger Administrator (LIA). In addition, certain designated routes within the NDN network are used to manage the global BC, functioning as the global BC administrator responsible for storing transactions generated by mobile producers from various clusters for inter-cluster mobility.

As shown in Fig. 2, our scheme involves six entities: Key Generation Center (KGC), Core Router, Edge Router, Cluster Head (CH), Mobile Producer (P), and Authorization Server (AS).

- KGC: The trusted authority is tasked with initializing system settings and generating partial private keys that validate the authenticity of network entities.
- Core Router: It is NDN router as global BC administrator and is assigned to manage the global BC.
- Edge Router: Positioned between NDN and producers, we regard the network as consisting of edge routers and core routers that form a unified autonomous system.
- CH: We use the memory-based algorithm (Muhammad, 2013) to select cluster head. It provides easy connectivity to other cluster heads and manages the entry/exit of cluster members and attached mobile producers. It also serves as an LIA to manage the local immutable ledger.
- P: It is considered a mobile device that holds P's identity ID_p, granting it the ability to generate one or more prefixes for publishing content. Each prefix is associated with a public/private key pair, ensuring the authentication of P's prefixes.
- AS: It handles P's initial authentication, verifies its ownership of the declares prefix, generates and distributes the source transaction, marking the start of the global BC.

3.2. Adversary model

An adversary can manipulate mobile devices, e.g., by holding a valid SIM card to connect to the network and forging legitimate IUs using the producer's prefixes. In addition, a malicious but legitimate P may launch a double-spend attack to disrupt the network, whereby P sends signatures for prefix authentication to multiple edge routers simultaneously. Finally, the adversary may also compromise CHs or core routers.

In a certificateless cryptosystem, adversaries are categorized into two types: Type I and Type II. Type I adversaries have the ability to replace a user's public key, while Type II adversaries possess the master private key. Furthermore, a certificateless signcryption scheme must ensure two critical security properties–message confidentiality and ciphertext unforgeability–when faced with attacks from both types of adversaries, because signcryption combines digital signature and encryption into one operation.

- Confidentiality: For two types of adversaries, after interacting with a challenger $\mathcal{B}$ through a sequence of Oracle queries, it still cannot determine, with non-negligible probability, which of two known messages corresponds to the given ciphertext. Then the scheme is deemed indistinguishable under adaptively chosen ciphertext attacks (IND-CLSC-CCA2). For further information on security games, refer to Gong et al. (2022), Chen et al. (2022, 2023), Dai and Xu (2023) and Zhang et al. (2024).
- Unforgeability: For two types of adversaries, after an interactive game has been executed, they remain unable to forge a ciphertext that would be accepted by receivers with a non-negligible probability. Then the scheme is considered to have existential unforgeability under adaptively chosen message attacks (EUF-CLSC-CMA).

3.3. Blockchain-based data storage

The BC functions as a distributed transactional database that provides various functions for data storage (Dukkipati et al., 2018) and participates in sharing among all nodes. The proposed scheme uses two main primitives: retrieval transaction and add transaction. The key features of the BC can be summarized as follows.

- Transaction: Each prefix authentication request by P (i.e., IU) is represented as a BC transaction.
- Block: Multiple valid transactions form a block, which needs to be verified before it can be chained. Each block contains a hash of the previous block to ensure invariance.
- Mining: All miners verify transactions before creating a block to be added to the BC. In the proposed scheme, only BC administrators are able to mine, i.e., local ledger and global BC administrators. Therefore, only the core routers and CHs are responsible for mining and broadcasting new blocks to the network.
- Genesis block: The genesis block represents the first block, generated when P registers on the network, with the AS responsible for verifying the initial transaction and creating the block.

The BC serves as a database of transactions shared among NDN routers that are responsible for verifying the prefix of P (i.e., IU), consisting of transactions added sequentially. The two basic primitives of BC are as follows:

- Retrieval transaction: All routers (R_i) maintain a local copy of the global BC. For each new transaction (i.e., producer-initiated prefix authentication request) received by R_i, the router retrieves pertinent information for authentication the producer,using the previous transaction ID (i.e., pre_Tx_{ID}).
- Add transaction: Once the above phase of transaction authentication is completed, the transactions are added to the BC. When a sufficient number of transactions accumulated in the mining pool, they will be packaged and created as a block, which will be obtained by all miners by competitive mining. The miner who mines the new block first adds to the BC and broadcasts it to the other R_i. When R_i receives a new block, it first validates the block and adds it to the local BC copy. Thus, each R_i has a copy of the latest version of the global BC.

4. Our scheme

4.1. BAS-NDN

The proposed scheme is outlined in six algorithms, as described below:

(1) System initialization: Before the producer mobility, the system needs to be initialized to generate the required public parameters. This process is executed by the KGC. With the security parameter λ as input, the KGC produces the system parameters pp by running the Algorithm 1, while keeping the master secret key s secret. Subsequently, the KGC provides the producer or AS with the partial private key by executing the Algorithm 2.

Algorithm 1 Setup(λ)

Input: Security parameter λ
Output: Public parameter pp
1: Select an appropriate elliptic curve $\mathbb{E}$ over a finite field F_q and sets $G =< P >$ as a cyclic group with a prime order q.
2: Choose $s \in Z_q$ as the master secret key and compute its public key $P_{pub} = sP$.
3: Choose five hash functions:
$H_1 : \{0,1\}^{l_0} \times G \times G \rightarrow Z_q^*$,
$H_2 : \{0,1\}^{l_0} \times \{0,1\}^{l_1} \times G \rightarrow Z_q^*$,
$H_3 : \{0,1\}^{l_0} \times \{0,1\}^{l_1} \times G \times G \rightarrow \{0,1\}^*$,
$H_4 : G \times \{0,1\}^* \times G \rightarrow Z_q^*$,
$H_5 : G \times \{0,1\}^* \rightarrow Z_q^*$.
Here l_0 and l_1 are security parameters defined by λ, the P's identity length is $\{0,1\}^{l_0}$ and AS's identity length is $\{0,1\}^{l_1}$.
4: The public parameters are output as $pp = \{G, q, P, P_{pub}, H_1, H_2, H_3, H_4, H_5\}$.

Algorithm 2 Partial Private-Key-Extract()

Input: Public parameter pp, master key s, and a P's identity ID_p/AS's identity ID_{AS}
Output: Partial public key T_i and partial private key d_{ID_i}
1: Select $\alpha_i \in Z_q^*$ to compute $T_i = \alpha_i P$.
2: Compute $h_i = H_1(ID_i, T_i, P_{pub})$.
3: Compute $d_{ID_i} = \alpha_i + s \cdot h_i$.
4: Return (d_{ID_i}, T_i) to ID_p or ID_{AS}.

(2) Key Generation: In the proposed scheme, P owns the key pair (pk_p, sk_p), which is associated with its prefix(es). This key pair is generated by the producer running the KeyGen Algorithm 3 with the KGC's partial private key. P uses private key sk_p to sign the prefix, and public key pk_p is used by the router and AS to authenticate P's prefix.

Algorithm 3 KeyGen

Input: Public parameter pp, partial public key T_i and partial private key d_{ID_i}
Output: Public key pk_i and private key sk_i.
1: Select $x_{ID_i} \in Z_q^*$ to compute $X_i = x_{ID_i} P$.
2: Output private key $sk_i = (x_{ID_i}, d_{ID_i})$ and $pk_i = (X_i, T_i)$.

(3) Signcryption: Signcryption is an emerging cryptographic method, which enables simultaneous signing and encryption in a single step, to achieve data integrity and data source verifiability. In BAS-NDN, the signature of P's prefix is generated by encrypting the prefix and some additional information $info$, comprising the identifiers of edge routers and CHs (i.e., R_i and ID_{CH_i}) where the producer is currently located, along with the ID associated with the BC (i.e., Tx_{ID} and pre_Tx_{ID}). The Tx_{ID} serves as a hash pointer to the interest message, connecting it to the prior transaction (i.e., pre_Tx_{ID}), where $Tx = H(message)$. When the producer sends the first interest (i.e., registered interest) to publish prefix(es), set the $pre\_Tx_{ID} = \perp$. NDN nodes receiving such interests forward them to the AS. The mobile producer performs Signcryption Algorithm 4 and outputs the signcryptext σ,

which can be used to verify the prefix. The producer signs and encrypts the prefix with its own key pair. Finally, both the ciphertext C and the signature φ are sent to the AS.

Algorithm 4 Signcryption()

Input: Public parameter pp, public key pk_p, private key sk_p and P's prefix $prefix$
Output: Signcryptext σ
1: Select $u \in Z_q^*$ to compute $U = uP$.
2: Compute $f_p = H_2(ID_p, ID_{AS}, pk_p)$.
3: Compute $\omega = (f_p \cdot (X_{AS} + T_{AS} + h_{AS}P_{pub})) \cdot u$ and $Y = H_3(ID_p, ID_{AS}, \omega, U)$.
4: Compute the ciphertext $C = Y \oplus (prefix + info)$.
5: Compute $r_1 = H_4(U, C, pk_p)$ and $r_2 = H_5(U, prefix + info)$.
6: Calculate $\varphi = u + r_1 \cdot (r_2 \cdot x_{ID_p} + d_{ID_p})$.
7: Output the signcryptext $\sigma = (U, C, \varphi)$ and transmit it to the AS.

(4) Verification: When P first connects to the network, the AS executes Unsigncryption Algorithm 5 to decrypt the prefix announced by P. If the validation passes, the AS generates P's initial transaction and broadcasts it across the network. All miners who receive this transaction add it to the subsequent block. As shown in Fig. 4, P's initial transaction contains the following: (1) P's public key pk_p, (2) P's prefix $prefix$, (3) prefix's signcryptext σ, (4) current transaction Tx_{ID}, (5) previous transaction pre_Tx_{ID} and (6) payload that improves the scalability and efficiency of the BC.

After the initial transaction is broadcasted, BC stores pk_p, σ, and $prefix$ to allow each NDN router to authenticate the IUs generated by P. Note that, routers are able to obtain the prefix plaintext in the BC and need only to verify the signature's validity by running the Verify Algorithm 6.

Algorithm 5 Unsigncryption()

Input: Public parameter pp, P's public key pk_p, AS's private key sk_p and signcryptext $\sigma = (U, C, \varphi)$
Output: P's prefix $perfix$ or $\perp$
1: Compute $\omega' = f_p \cdot (x_{ID_{AS}} + d_{ID_{AS}}) \cdot U$ and $prefix + info = C \oplus H_4(ID_p, ID_{AS}, \omega', U)$.
2: Compute $r_1' = H_4(U, C, pk_p)$ and $r_2' = H_5(U, prefix + info)$.
3: Verify $\varphi P = U + r_1'(r_2' \cdot X_i + T_i + h_i P_{pub})$.
4: If the verification is successful, the $prefix$ will be accepted; otherwise $\perp$ will be return.

Algorithm 6 Verify()

Input: Public parameter pp, P's public key pk_p and signcryptext $\sigma = (U, C, \varphi)$
Output: $True$ or $\perp$
1: Compute $r_1' = H_4(U, C, pk_p)$ and $r_2' = H_5(U, prefix + info)$.
2: Verify $\varphi P = U + r_1'(r_2' \cdot X_i + T_i + h_i P_{pub})$.
3: If the verification is successful, the $prefix$ will be accepted; otherwise, $\perp$ will be return.

4.2. Secure producer mobility

This section outlines the authentication process when a mobile producer reaches a new PoA and publishes an IU. The BC provides facilitates the validation of the producer's IU for each NDN router.

To ensure secure authenticate of the IU while in the move, the producer signs $prefix$ alongside the associated edge routers R_i and the CHs ID_{CH_i} using the private key sk_p, which is intended to have invariant with the initial prefix registration, while being able to track P's activity trajectory. The tuple $(prefix, \sigma, pre\_Tx_{ID}, Tx_{ID})$ is then sent to the network, where $Tx_{ID} = H(IU)$. Note that, a new edge router or CH ID may be signed on each IU authentication event of the mobile producer. In addition, the scheme implements backward security, which allows the BC to link each P's IU authentication with its previous authentication in an invariant manner, ultimately pointing to the initial prefix registration.

In BAS-NDN, each router R_i performs the same authentication, i.e., Algorithm 6, when it receives a new IU request. When an IU is received, the edge router first utilizes pre_Tx_{ID} to access the P's previous transaction and obtain pk_p, which is utilized to validate the digital signature σ contained in the current transaction. If this validation passes, the edge router forwards the transaction to the CH. Subsequently, the CH broadcasts the transaction across the network for inclusion in the BC. Fig. 3 illustrates the process of initial registration and IU authentication for the BAS-NDN.

4.3. Scalable BC for BAS-NDN

NDN comprises multiple core routers alongside edge routers. To facilitate the scalability of the BC, core routers and CHs are designated as Global Blockchain Administrators (GBAs) responsible for managing the global BC. To maintain the integrity of GBAs, the blocks are protected by the proposed scheme. In addition, each CH acts as a LIA to manage the LIL and handle local transactions generated and disseminated within the cluster. The transaction structure is shown in Fig. 4. The first field serves as a hash pointer to the previous transaction of the producer, interlinking all transactions generated by mobile P. The second is the initial prefix, public key, and signature of the P, respectively.

All transactions follow the source transaction, which is generated by the AS first authenticating the prefix of P and stored as a block within the BC. The block is composed of two components: the successfully authenticated transaction and a block header that includes the hash of the preceding block, the GBA ID and its signature to ensure block-level immutability. If an attacker attempts to corrupt any block, the hash of subsequent blocks would become inconsistent with the global BC, which effectively prevents attacks. Similar to Bitcoin, multiple transactions can be bundled together into a single block, which can accommodate up to Tx^{Max} transactions, the value of it is related to BC throughput.

Compared to resource-dense consensus algorithms such as Proof of Work or Proof of Stake, BAS-NDN uses a more efficient time-based consensus algorithm (Dorri et al., 2017), which randomly assigns a block generator (i.e., GBA) among all BC administrators (i.e., miners). GBA generates a limited number of blocks on the same node. To deter malicious BC administrators from generating an excessive number of spurious blocks, a random waiting period must elapse before a new block is generated. The waiting period is restricted to twice the maximum end-to-end delay observed between NDN routers, which ensures that each BC administrator has sufficient time to propagate new blocks. When a BC administrator receives a block created from another administrator during the waiting period, it deletes the duplicate transaction in the transaction pool if the block contains a pre-existing transaction.

4.4. Verification

The BC administrator verifies the received block before adding it to the local BC. The block is deemed valid only if each transaction within it is verified. The verification process for a single transaction is outlined in Algorithm 7. The connection between transactions associated with the producer is established through the hash pointer referencing the previous transaction. Therefore, the BC administrator initially verifies the links between consecutive transactions of the producer through pre_Tx_{ID}. Then, pk_p in $Tx - 1_{ID}$ is used to verify the signature stored in Tx_{ID}. Note that, pk_p in Tx_{ID} is retrieved from $Tx - 1_{ID}$ to ensure its association with the initial transaction, where pk_p is verified by the AS.

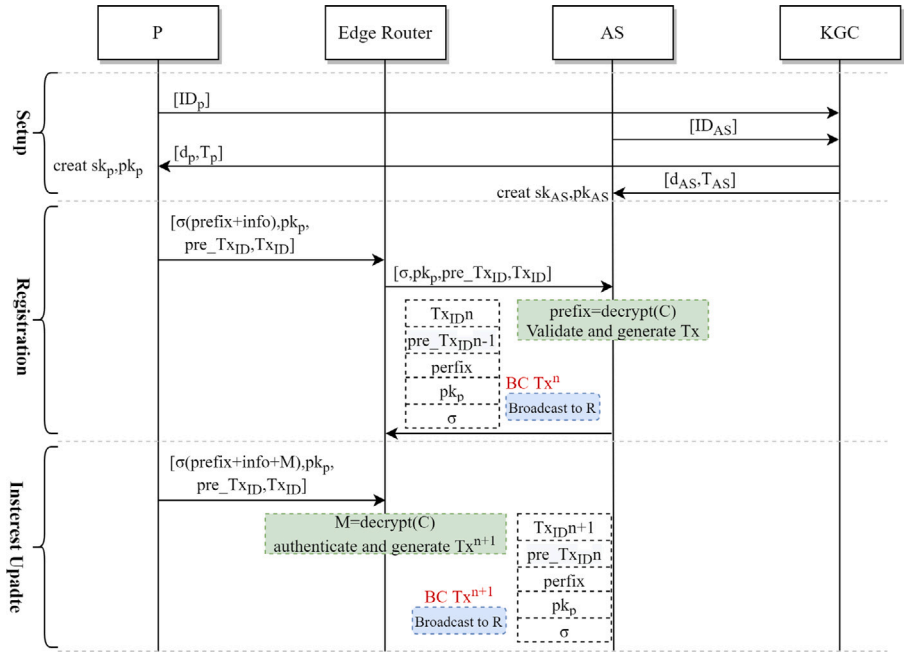

Fig. 3. Initial registration and prefix authentication process.

Algorithm 7 BlockChain Tx Verify()

Input: P's $prefix$, P's public key pk_p, Transaction Tx_{ID}, Previous Transaction pre_Tx_{ID} and Additional information $info$
Output: $True$ or $\perp$

```
if pre_TX_ID ≠ Tx − 1_ID then
    return ⊥
else
    if φP ≠ U + r'(H5(U, C, pk_p^{x−1}) · X_i + T_i + h_i P_pub) then
        return ⊥
    else
        check prefix
        return True
    end if
end if
```

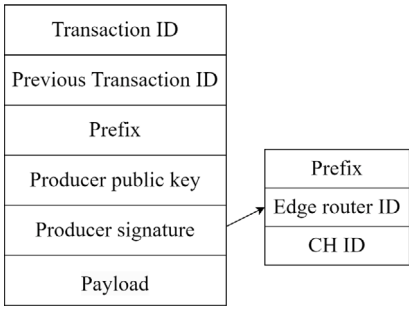

Fig. 4. Transaction structure.

4.5. Intra/Inter cluster handoff

P issues two transaction IDs: one associated with the LIL, denoted as Tx_{ID_i}, and the other associated with the global BC, denoted as Tx_{ID_j}. While P moves within the cluster, the CH maintains both the LIL and global BC transactions for P. Thus, only the CH can control P's location change. The LIL records all transactions of P, where each block consists of a block header and a policy header. The block header stores a hash of the preceding block, and the policy header establishes rules for processing transactions in the form of access control lists. Mobile P connects to the nearest cluster by checking the CH ID, and after being authenticated by that cluster, it only updates and sends the local Tx_{ID_i} to be authenticated again at the same cluster. Thus, P can seamlessly switch within clusters using only the LIL, until it moves to another cluster, which reduces communication overhead within the core network.

For the inter-cluster move event of P, it updates the global Tx_{ID_j} utilized in the previous cluster. Since the base station of the new cluster cannot locate any information related to P in the LIL, P can only be involved in the verification using Tx_{ID_j}. After it passes, the transaction is processed by the CH and subsequently recorded in the global BC.

As shown in Fig. 5, when the CH receives a transaction, it first determines whether P is moving in the same cluster. The CH looks up the pointer to the previous transaction in the IL based on Tx_{ID_i}. If it

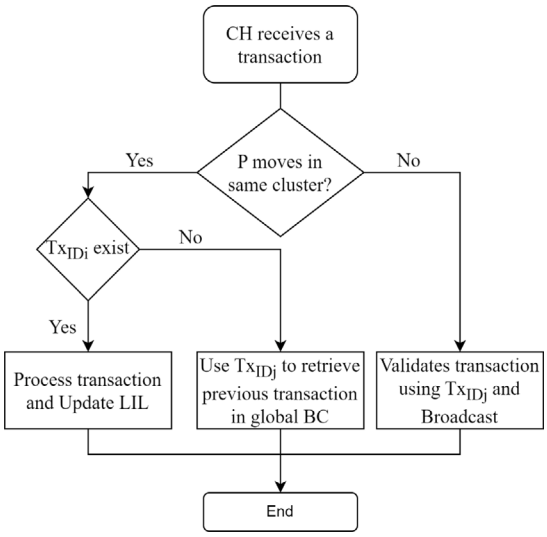

Fig. 5. Cluster handoff.

exists, the CH will process the transaction and update the LIL; otherwise, it uses Tx_{ID_j} to retrieve the previous transaction in the global BC. If P moves among clusters, the new CH validates the transaction using Tx_{ID_j} and broadcasts it to all other GBAs. Each router verifies the transaction and stores it in an unprocessed local transaction pool. When the size of the transaction pool reaches Tx^{Max}, the GBA consolidates these transactions as a block and incorporates it to the global BC using the time-based consensus algorithm.

5. Security analysis

5.1. Prefix hijacking attack mitigation

An adversary must complete the initial registration before inserting an error transaction into the BC. This involves generating a legitimate signature and prefix. However, the adversary cannot independently complete registration without P's private key, which is never transmitted over the network. Therefore, the adversary can only replay the valid initial registration interest.

The signature uses P's private key, which is never transmitted over the network. Therefore, an adversary cannot independently complete a valid registration without P's private key and can only replay a legitimate initial registration interest. However, since the hash pointer of the current transaction is immutable with respect to the previous transaction, the AS discard the replayed interest upon receipt, causing the adversary's authentication to fail. As a result, the security features of BAS-NDN are effective in preventing the adversary's prefix hijacking attack because the BC Ledger is immutable without the private key, the PoA information, and the hash pointer, i.e., pre_TX_{ID}.

5.2. Two types of adversaries

In certificateless cryptography, there exist two types of adversaries: Type I and Type II adversaries. In this section, we demonstrate that the BAS-NDN ensures data confidentiality and unforgeability against attacks launched by these adversaries.

Theorem 1 (*Confidentiality*). *Within the random oracle model, if an adversary compromises the IND-CCA2 security of our scheme with a significant advantage, we can construct an algorithm C that leverages the adversary to solve the ECDLP with a non-negligible probability. Lemmas 1 and 2 illustrate resistance to both Type I and Type II adversaries.*

Lemma 1. *In probability polynomial time (PPT), if an adversary $\mathcal{A}_I$ breaches the IND-CCA2-I security of the proposed scheme with a non-negligible advantage ϵ_1, then we can construct an algorithm $\mathcal{B}$ that employs $\mathcal{A}_I$ to solve an instance of the square-DH with non-negligible advantage*

$$\varepsilon \geq (1-\frac{2}{q_{pp}})(1-\frac{2+q_r}{q_p})(1-\frac{2}{q_{us}})\frac{1}{q_{H_3}}(1-\frac{2}{q_{sc}^2})\epsilon_1,$$

where q_{pp} denotes the number of queries to partial-private-key oracle, q_p represents the query time to private-key-extract oracle, q_r is the number of queries to public-key-replacement oracle, q_{H_3} indicates the number of queries to the H_3 oracle, q_{us} and q_{sc} correspond to the query counts for unsigncryption oracle and signcryption oracle, respectively.

Proof. Assuming that an adversary $\mathcal{A}_I$ successfully compromises the proposed scheme, we can construct a challenger $\mathcal{B}$ to solve the square-DH by utilizing $\mathcal{A}_I$ as a subroutine. Given an instance of the square-DH problem represented as (P, aP), we attempt to compute a^2P through an interactive game between $\mathcal{B}$ and $\mathcal{A}_I$. $\mathcal{B}$ creates a game simulation environment and addresses a series of queries posed by $\mathcal{A}_I$.

(1) Setup: The challenger $\mathcal{B}$ executes this algorithm to generate the system parameters $pp = \{G, q, P, P_{pub}, H_1 \sim H_5\}$, where $P_{pub} = aP$. Subsequently, $\mathcal{B}$ sends pp to $\mathcal{A}_I$.

(2) Phase 1: Let ID_p^* and ID_r^* represent the identities of the challenged entities, specifically the producer and the receiver, respectively. $\mathcal{A}_I$ is permitted to adaptively issue the following oracle queries.

- H_1 Oracle Queries: When $\mathcal{A}_I$ issues this query with (ID_i, T_i, P_{pub}), $\mathcal{B}$ maintains a list L_{H_1} which is initially empty. If the query tuple is found in L_{H_1}, the corresponding h_i is returned. If not, $\mathcal{B}$ randomly selects $h_i \in Z_q^*$ and provides it to $\mathcal{A}_I$. Finally, $(h_i, ID_i, T_i, P_{pub})$ is added to L_{H_1}.
- H_2 Oracle Queries: For this hash query, $\mathcal{A}_I$ uses (ID_i, ID_j, pk_i), and $\mathcal{B}$ keeps an initially empty list list L_{H_2}. If (ID_i, ID_j, pk_i) exists in L_{H_2}, the corresponding f is returned. Otherwise, $\mathcal{B}$ randomly selects $f_i \in Z_q^*$ and returns it to $\mathcal{A}_I$. Lastly, (ID_i, ID_j, pk_i, f_i) is inserted to L_{H_2}.
- H_3 Oracle Queries: $\mathcal{A}_I$ submits (ID_i, ID_j, ω, U), $\mathcal{B}$ maintains an initially empty list L_{H_3}. If it is present in L_{H_3}, Y_{ij} is returned. Otherwise, $\mathcal{B}$ randomly chooses $Y_{ij} \in Z_q^*$ and provides it to $\mathcal{A}_I$. Finally, $(Y_{ij}, ID_i, ID_j, \omega, U)$ is added to L_{H_3}.
- H_4 Oracle Queries: $\mathcal{A}_I$ issues this query with (U, C, pk_i), $\mathcal{B}$ maintains an empty list L_{H_4}. If $U = aP$ and $ID_i = ID_p^*$, then $\mathcal{B}$ set $r_1 = (t^* + h_p)^{-1}$, where t^* is derived from L in the Creating-User oracle. If $U \neq aP$ and (U, C, pk_i) exists in L_{H_4}, the corresponding r_1 is returned. If neither condition is met, $\mathcal{B}$ randomly chooses $r_2 \in Z_q^*$ and returns it to $\mathcal{A}_I$. Finally, (r_2, U, C, pk_i) is inserted to L_{H_4}.
- H_5 Oracle Queries: $\mathcal{A}_I$ submits $(U, prefix)$, $\mathcal{B}$ maintains a list L_{H_5} which is initially empty. If it exists in L_{H_5}, the corresponding r_2 is returned. Otherwise, $\mathcal{B}$ randomly chooses $r_2 \in Z_q^*$ and returns it to $\mathcal{A}_I$. Finally, $(r_2, U, prefix)$ is inserted to L_{H_5}.

(3) Creating-User Oracle: In this case, $\mathcal{A}_I$ submits a query with ID_i, if $ID_i \neq ID_p^*$ and $ID_i \neq ID_r^*$, $\mathcal{B}$ randomly chooses $d_{ID_i}, x_{ID_i} \in Z_1$ and set $T_i = d_{ID_i}P - h_i P_{pub}$ and $X_{ID_i} = x_{ID_i}P$. If $ID_i = ID_r^*$, $\mathcal{B}$ randomly chooses $x_r^* \in Z_q$, then sets $d_{ID_i} = \perp$, $x_{ID_i} = x_r^*$ and $T_i = t_r^* \cdot aP$, where $t_r^* \in Z_q^*$ is a random number. Note that, $X_i = x_r^* P$. When $ID_i = ID_p^*$, $\mathcal{B}$ randomly chooses $x_p^*, t^* \in Z_q$ to set $X_i = x_p^* P$, $T_i = t^* aP$, $x_{ID_i} = x_p^*$ and $d_{ID_i} = \perp$. It also sets $t_i = t^*$. Finally, $\mathcal{B}$ inserts $(ID_i, T_i, X_i, x_{ID_i}, d_{ID_i}, h_i, t_i)$ in the list L which is initially empty. Note that, when $ID_i \in \{ID_p^*, ID_r^*\}$, the corresponding x_p^* or x_r^* can be returned to $\mathcal{A}_I$.

(4) Public-Key Oracle: When $\mathcal{A}_I$ submits a query with the identity ID_i, $\mathcal{B}$ first checks the presence of ID_i in L. If it is found, $\mathcal{B}$ sets public key $pk_i = (X_i, T_i)$ and sends to $\mathcal{A}_I$. Otherwise, $\mathcal{B}$ issues a creating-user query with ID_i to obtain the public key pk_i.

(5) Partial-Private-Key Oracle: When $\mathcal{A}_I$ issues this query with ID_i, if $ID_i \neq ID_p^*$ and $ID_i \neq ID_r^*$, $\mathcal{B}$ retrieves d_{ID_i} from the list L and returns it to $\mathcal{A}_I$. Otherwise, $\mathcal{B}$ outputs $\perp$.

(6) Public-Key-Replacement Oracle: When $\mathcal{A}_I$ makes a query using $(ID_i, pk_i' = (X_i', T_i'))$, $\mathcal{B}$ verifies if ID_i is present in L. If it is found, $\mathcal{B}$ replaces (X_i, T_i) with the new pair (X_i', T_i').

(7) Private-Key-Extract Oracle: When $\mathcal{A}_I$ initiates this query with ID_i, if $ID_i \notin \{ID_p^*, ID_r^*\}$, $(ID_i, x_{ID_i}, d_{ID_i})$ exists in L without a preceding public key-replacement oracle for ID_i. Then (x_{ID_i}, d_{ID_i}) is returned to $\mathcal{A}_I$. Otherwise, $\mathcal{B}$ terminates it.

(8) Signcryption Oracle: When $\mathcal{A}_I$ presents a signcryption query with $(ID_i, ID_j, prefix)$, where ID_i and ID_j represent the producer and the receiver, respectively. $\mathcal{B}$'s responds as follows:

(a) If $ID_i \notin \{ID_p^*, ID_r^*\}$, $\mathcal{B}$ retrieves (x_{ID_i}, d_{ID_i}) from L and executes the signcryption algorithm to obtain the signcryptext σ.

(b) If $ID_i = ID_p^*$ and $ID_j = ID_r^*$, $\mathcal{B}$ retrieves (x_{ID_j}, d_{ID_j}) from L, and does as follows:

i. It randomly selects $r \in Z_q$ to compute $U = -r_1 \cdot (t^* + h_p)aP$, where $(t^* + h_p)$ is from the list L_{H_4}.
ii. Next, it determines $\omega = (f_p \cdot (x_{ID_j} + d_{ID_j})) \cdot U$ and $Y = H_4(ID_i, ID_j, \omega, U)$.
iii. It then calculates $C = prefix \oplus Y$ by using the producer prefix $prefix$.
iv. The value $r_1 = H_4(U, C, pk_i)$ is set, and (U, C, T_i, X_i, r_1) is added to L_{H_4}.
v. It sets $\varphi = r_1 \cdot r_2 \cdot x_s^*$, where $r_2 = H_5(U, prefix)$.
vi. Finally, the signcryptext $\sigma = (U, C, \varphi)$ is returned to $\mathcal{A}_I$.

Note that,

$$\begin{aligned}\varphi P &= r_1 \cdot r_2 x_s^* P \\ &= -r_1 \cdot (t^* + h_p)aP + r_1 r_2 x_s^* P + r_1(t^* + h_p)aP \\ &= U + r_1 \cdot (r_2 X_s + T_s + h_p \cdot P_{pub}).\end{aligned}$$

(c) If $ID_i = ID_r^*$ and $ID_j = ID_p^*$, this case is similar to above case $ID_i = ID_p^*$ and $ID_j = ID_r^*$.
(d) If $ID_i \in \{ID_p^*, ID_r^*\}$ and If $ID_j \in \{ID_r^*, ID_p^*\}$, $\mathcal{B}$ aborts the game.

(9) Unsigncryption Oracle: Upon receiving the tuple (ID_i, ID_j, σ), $\mathcal{B}$ responses as follows:

- If $ID_j \notin \{ID_p^*, ID_r^*\}$, $\mathcal{B}$ retrieves (x_{ID_j}, d_{ID_j}) from L. It then utilizes the unsigncryption algorithm to decrypt σ and returns $prefix$ to $\mathcal{A}_I$.
- If $ID_j \in \{ID_r^*, ID_p^*\}$ while $ID_i \notin \{ID_p^*, ID_r^*\}$, $\mathcal{B}$ interprets σ as (U, C, φ) and fetches (x_{ID_i}, d_{ID_i}). And it recovers $u = \sigma - r_1 \cdot (r_2 \cdot x_{ID_i} + d_{ID_i})$, where $r_1 = H_4(U, C, pk_i)$ and $r_2 = H_5(U, prefix)$. Next, it computes

  $$\omega = (f_i \cdot (X_j + T_j + h_j P_{pub})) \cdot u.$$

  Finally, $prefix$ is recovered by computing $C \oplus H_4(ID_i, ID_j, \omega', U)$.
- Otherwise, $\mathcal{B}$ terminates the process and returns $\perp$.

(10) Challenge: $\mathcal{A}_I$ provides two prefixes of equal length $(prefix_1, prefix_2)$ along with identities (ID_p^*, ID_r^*), where ID_p^* and ID_r^* represent the producer and the receiver in the challenged, respectively. $\mathcal{B}$ randomly chooses a bit $b \in \{0, 1\}$, ensuring that both ID_p^* and ID_r^* are present in $L_{H_i} \in \{1, 2, 3, 4, 5\}$. It also selects a random value $V^* \in G$ and assigns $U^* = -aP$. It then computes $Y^* = H_3(ID_p^*, ID_r^*, \omega, U)$, $C^* = prefix_b \oplus Y^*$ and $\varphi = (t^* + h_p)^{-1} \cdot r_2 x_p^*$, where t^* and f are from L and L_{H_2}. In conclusion, $\mathcal{B}$ transmits $\sigma^* = (\varphi^*, C^*, U^*)$ to $\mathcal{A}_I$. Note that,

$$\begin{aligned}\varphi P &= (t^* + h_p)^{-1} \cdot r_2 x_p^* P \\ &= -aP + (t^* + h_p)^{-1} \cdot r_2 x_p^* P + aP \\ &= U^* + r_1 \cdot (r_2 X_p + T_p + h_p P_{pub}).\end{aligned}$$

Here $r_1 = (t^* + h_p)^{-1}$ is obtained by the list L_{H_4}.

(11) Phase 2: During this phase, the queries mirror those from Phase 1, with the exception that σ^* cannot be queried for unsigncryption oracle, and ID_p^* and ID_r^* cannot be used in the private key oracle and partial-private-key oracle, respectively.

(12) Guess: Ultimately, $\mathcal{A}_I$ provides its guess $b' \in \{0, 1\}$. Based on the calculation procedure for ω during the decryption phase, it can be represented as follows:

$$\begin{aligned}\omega^* &= (f_p \cdot (x_{ID_r} + d_{ID_r})) \cdot U^* \\ &= -(f_p \cdot (x_{ID_r})aP) - (f_p(t_r^* + h_r \cdot a)aP) \\ &= -f_p(x_{ID_r}aP + (t_r^* + h_r)a^2 P).\end{aligned}$$

Thus, $\mathcal{B}$ can solve the square-DH problem by retrieving ω from the hash list L_{H_3}.

$$a^2 P = \frac{-\omega^* - x_{ID_r}aP}{t_r^* + h_r}$$

Probability Analysis: Based on the previous game, it is evident that if the game does not terminate, $\mathcal{B}$ can solve the square-DH problem. The probability of an abort occurring during the Partial-Private-Key query phase is $2/q_{pp}$, and the probability of aborting in Private-Key-Extract query is $\frac{2+q_r}{q_p}$. In signcryption query phase, the probability of an abort is $2/q_{sc}^2$, and for unsigncryption query aborts, this probability is $2/q_{us}$. In addition, we must ensure that ω^* has been issued a H_3-oracle query. Thus, the probability of successfully solving the square-DH problem is at least

$$\varepsilon \geq (1 - \frac{2}{q_{pp}})(1 - \frac{2 + q_r}{q_p})(1 - \frac{2}{q_{us}})\frac{1}{q_{H_3}}(1 - \frac{2}{q_{sc}^2})\epsilon_1. \quad \square$$

Lemma 2. *If there exists an adversary $\mathcal{A}_{II}$ capable of compromising the IND-CCA2-II security of the proposed scheme with a non-negligible advantage ϵ_2, it becomes to devise an algorithm $\mathcal{B}$ that can solve a square-DH instance with a substantial advantage*

$$\varepsilon \geq (1 - \frac{2}{q_p})(1 - \frac{2}{q_{us}})\frac{1}{q_{H_3}}(1 - \frac{2}{q_{sc}^2})\epsilon_2.$$

Proof. Assuming the existence of an adversary $\mathcal{A}_{II}$ that can successfully compromise our scheme, we can construct a challenger $\mathcal{B}$ designed to solve the square-DH by utilizing $\mathcal{A}_{II}$ as a subroutine. Given an instance (P, aP) of the square-DH problem, we attempt to compute $a^2 P$ through an interactive game between $\mathcal{B}$ and $\mathcal{A}_{II}$. Similar to Lemma 1, $\mathcal{A}_{II}$ and $\mathcal{B}$ execute a series of queries:

(1) $\mathcal{B}$ invokes the Setup to create the system parameters $pp = \{G, q, P, P_{pub}, H_1 \sim H_5\}$, where $P_{pub} = sP$ and $s \in Z_q$. $\mathcal{B}$ provides both pp and the master secret key s to $\mathcal{A}_{II}$.
(2) Phase 1: Let ID_p^* and ID_r^* represent the identities of the challenged entities, specifically the producer and the receiver. $\mathcal{A}_{II}$ can adaptively issue the following oracle queries.

- Oracle Queries for $H_1/H_2/H_3/H_5$: $\mathcal{A}_{II}$ issues these query, $\mathcal{B}$ responses as Lemma 1.
- Oracle Queries for H_4: If $\mathcal{A}_I$ submits a query with (U, C, pk_i), $\mathcal{B}$ maintains an empty list L_{H_4}. If $U = aP$ and $ID_i = ID_p^*$, then $\mathcal{B}$ assigns $r_1 = -(r_2 \cdot \alpha_i)^{-1}$. If $U \neq aP$ and the tuple exists in L_{H_4}, then r_2 is returned. Otherwise, $\mathcal{B}$ randomly selects $r_1 \in Z_q^*$ and returns it to $\mathcal{A}_I I$. Finally, (r_1, U, C, pk_i) is inserted to L_{H_4}.

(3) Creating-User Oracle: $\mathcal{A}_{II}$ issues this query with ID_i, if $ID_i \notin \{ID_p^*, ID_r^*\}$, $\mathcal{B}$ randomly selects $t_i, x_{ID_i} \in Z_q$ and sets $T_i = t_i P$, $X_i = x_{ID_i} P$, and $d_{ID_i} = t_i + sh_i$, where h_i is acquired by querying the H_1-Oracle. If $ID_i = ID_r^*$, $\mathcal{B}$ randomly selects $t_i, \alpha_i \in Z_q$ to set $T_i = t_i P$, $X_i = \alpha_i aP$, $d_{ID_i} = t_i + sh_i$ and $x_{ID_i} = \perp$. When $ID_i = ID_s^*$, $\mathcal{B}$ randomly chooses $\alpha_i, d_{ID_i}^* \in Z_q$ to set $T_i = d_{ID_i}^* P - h_i sP$, $X_i = \alpha_i aP$, $d_{ID_i} = d_{ID_i}^*$, $t_i = d_{ID_i}^* - h_i s$ and $x_{ID_i} = \perp$. Lastly, $\mathcal{B}$ adds $(ID_i, T_i, X_i, x_{ID_i}, d_{ID_i}, h_i, t_i, \alpha_i)$ to the initially empty list L.
(4) Public-Key/Private-Key-Extract Oracle: They are same as Lemma 1.
(5) Signcryption Oracle: $\mathcal{A}_{II}$ submits a signcryption oracle with $(ID_i, ID_j, prefix)$, where ID_i and ID_j signify the producer and the receiver. $\mathcal{B}$ is structured as follows:

(a) If $ID_i \notin \{ID_p^*, ID_r^*\}$, $\mathcal{B}$ retrieves (x_{ID_i}, d_{ID_i}) from L and executes signcryption algorithm to obtain the signcryptext σ.

(b) If $ID_i = ID_p^*$ and $ID_j = ID_r^*$, $\mathcal{B}$ retrieves (x_{ID_j}, d_{ID_j}) and computes

i. A random value $u \in Z_q$ is chosen to compute $U = uaP$.
ii. Next, it computes $\omega = (f_p \cdot (x_{ID_j} + d_{ID_j})) \cdot U$ and $Y = H_4(ID_i, ID_j, \omega, U)$, where $f_p = H_2(ID_i, ID_j, pk_i)$.
iii. It then calculates $C = prefix \oplus Y$ using the producer prefix $prefix$.
iv. The value $r_1 = r_s^* \cdot u$ is established, and (U, C, T_i, X_i, r_1) is added to L_{H_4}.
v. It sets $\varphi = -u \cdot d_{ID_p}(f_p\alpha_p)^{-1}$.
vi. Finally, the signcryptext $\sigma = (U, C, \varphi)$ is returned to $\mathcal{A}_{II}$.

(c) If $ID_i = ID_r^*$ and $ID_j = ID_p^*$, this scenario is analogous to above where $ID_i = ID_p^*$ and $ID_j = ID_r^*$.

(d) If $ID_i \in \{ID_p^*, ID_r^*\}$ and $ID_j \in \{ID_r^*, ID_p^*\}$, $\mathcal{B}$ aborts the process.

(6) Unsigncryption Oracle: Upon receiving a query in (ID_i, ID_j, σ), $\mathcal{B}$ responses as follows:

- If $ID_j \notin \{ID_p^*, ID_r^*\}$, $\mathcal{B}$ retrieves (x_{ID_j}, d_{ID_j}) corresponding to ID_j. It decrypts σ by invoking unsigncryption algorithm and subsequently returns $prefix$ to $\mathcal{A}_{II}$.
- If $ID_j \in \{ID_r^*, ID_p^*\}$ and $ID_i \notin \{ID_p^*, ID_r^*\}$, $\mathcal{B}$ parses σ as (U, C, φ) and retrieves (x_{ID_i}, d_{ID_i}). And it recovers $u = \sigma - r_1 \cdot (r_2 \cdot x_{ID_i} + d_{ID_i})$, where $r_1 = H_4(U, C, pk_i)$ and $r_2 = H_5(U, prefix)$. Next, it computes

$$\omega = (f_i \cdot (X_j + T_j + h_j P_{pub})) \cdot u.$$

Finally, $prefix$ is recovered by computing $C \oplus H_4(ID_i, ID_j, \omega', U)$.
- In all other cases, $\mathcal{B}$ terminates the process and outputs $\perp$.

(7) Challenge: $\mathcal{A}_{II}$ generates two prefixes of equal length $(prefix_1, prefix_2)$ and two identities (ID_p^*, ID_r^*). $\mathcal{B}$ randomly chooses a bit $b \in \{0,1\}$ and ensures that both ID_p^* and ID_r^* appear in $L_{H_i}\, i \in \{1,2,3,4,5\}$. It also randomly selects $V^* \in G$ and defines $U^* = aP$. It then computes $Y^* = H_3(ID_i, ID_r^*, \omega, U)$, $C^* = prefix_b \oplus Y^*$ and $\varphi = -d_{ID_p} \cdot (r_2\alpha_p)^{-1}$, where α_p is form the lists L. Finally, $\mathcal{B}$ transmits $\sigma^* = (\varphi^*, C^*, U^*)$ to $\mathcal{A}_{II}$. Note that,

$$\begin{aligned}\varphi^* P &= -d_{ID_p} \cdot (r_2\alpha_p)^{-1} P \\ &= aP - (r_2\alpha_p)^{-1}((r_2\alpha_p)aP + d_{ID_p}P) \\ &= U^* + r_s^* \cdot (r_2 X_p + T_p + h_p P_{pub}).\end{aligned}$$

Here $r_s^* = -(r_2\alpha_p)^{-1}$ is obtained by the list L_{H_4}.

(8) Phase 2: During this phase, the queries are analogous to Phase 1, with the notable exception that σ^* cannot be used in unsigncryption oracle. ID_p^* and ID_r^* are not permitted in the private key extract oracle.

(9) Guess: Ultimately, $\mathcal{A}_{II}$ returns its guess $b' \in \{0,1\}$. According to the calculation process of ω^*, we can know:

$$\begin{aligned}\omega^* &= (f_p \cdot (x_{ID_r} + d_{ID_r})) \cdot U^* \\ &= (f_p \cdot (\alpha_r \cdot a + d_{ID_r})) \cdot aP \\ &= f_p\alpha_r a^2 P + f_p(t_r^* + h_r s)aP.\end{aligned}$$

Consequently, $\mathcal{B}$ is able to resolve the square-DH problem by obtaining ω from the list L_{H_3}.

$$a^2 P = \frac{\omega^* - (t_r^* + h_r s)aP}{\alpha_r}$$

Probability Analysis: Based on the aforementioned game, it is evident that if the game without aborting, $\mathcal{B}$ is capable of solving the square-DH problem. The probability of an abort occurring during the private key extract query is $2/q_p$, and the probability of signcryption query aborting is $2/q_{sc}^2$. For the unsigncryption query, the probability of termination is $2/q_{us}$. Furthermore, it is essential to verify that ω^* has been issued the H_3-oracle query. Therefore, the probability of solving the square-DH problem is at least

$$\epsilon \geq (1 - \frac{2}{q_p})(1 - \frac{2}{q_{us}})\frac{1}{q_{H_3}}(1 - \frac{2}{q_{sc}^2})\epsilon_2. \quad \square$$

Based on Lemmas 1 and 2, it follows that if $\mathcal{A}_I$ and $\mathcal{A}_{II}$ succeed with a non-negligible probability in the above games, the Challenger $\mathcal{B}$ is able to solve the square-DH problem. This situation contradicts the established difficulty of solving the square-DH. Thus, our scheme is IND-CCA2 safe in the ROM.

Theorem 2 (*Unforgeability*)**.** *If an adversary manages to compromise the EUF-CMA security of the scheme with a certain advantage, we can build an algorithm C that can utilize this adversary to effectively solve the Square-DH with a non-negligible probability. Lemmas 3 and 4 demonstrate resistance to Type I and Type II adversaries.*

Lemma 3. *In P.P.T, if $\mathcal{A}_I$ who breaks the EUF-CMA-I security of the proposed scheme with a non-negligible advantage ϵ_3, we can develop an algorithm $\mathcal{B}$ to solve an instance of the square-DH with non-negligible advantage*

$$\epsilon \geq (1 - \frac{2 + q_r}{q_{pp}})(1 - \frac{2}{q_p})(1 - \frac{2}{q_{sc}^2})\epsilon_3.$$

Proof. Given an instance of the square-DH problem represented as (P, aP), we attempts to compute a^2P by conducting an interactive game between $\mathcal{B}$ and $\mathcal{A}_I$. Similar to Lemma 1, $\mathcal{A}_I$ and $\mathcal{B}$ execute a series of queries:

(1) Setup: This phase is similar to Lemma 1.
(2) $H_1 \sim H_5$-Oracle Queries: They are similar to Lemma 1. $\mathcal{A}_I$ issues a hash query, $\mathcal{B}$ returns the corresponding value by checking $L_{H_1} \sim L_{H_5}$.
(3) Creating-User Oracle: Let ID_p^* and ID_r^* be the identities of the challenged. When $\mathcal{A}_I$ submits this query with ID_i, if $ID_i \notin \{ID_p^*, ID_r^*\}$, $\mathcal{B}$ randomly selects $d_{ID_i}, x_{ID_i} \in Z_q$ and sets $T_i = d_{ID_i}P - h_i P_{pub}$ and $X_{ID_i} = x_{ID_i}P$. If $ID_i \in \{ID_p^*, ID_r^*\}$, $\mathcal{B}$ randomly picks $\zeta_i, \pi_i \in Z_q$ to define $X_i = \zeta_i P$, $T_i = \pi_i aP - h_i aP$, $x_{ID_i} = \zeta_i$, and $d_{ID_i} = \perp$. Ultimately, $\mathcal{B}$ adds $(ID_i, T_i, X_i, x_{ID_i}, d_{ID_i}, h_i, \zeta_i)$ to the initially empty list L. Note that, when $ID_i \in \{ID_p^*, ID_r^*\}$, $d_{ID_i} = \pi_i a$ is unknown for $\mathcal{B}$.
(4) Public-Key/Partial-Private-Key/Public-Key-Replacement/ Private-Key-Extract Oracle: These oracle are similar to Lemma 1.
(5) Signcryption Oracle: $\mathcal{A}_I$ submits a signcryption request in $(ID_i, ID_j, prefix)$, where ID_i and ID_j represent the producer and the receiver. $\mathcal{B}$ responds as follows:

(a) If $ID_i \notin \{ID_p^*, ID_r^*\}$, $\mathcal{B}$ retrieves (x_{ID_i}, d_{ID_i}) from L and runs the signcryption algorithm to obtain the signcryptext σ.
(b) If $ID_i = ID_p^*$ and $ID_j \neq ID_r^*$, $\mathcal{B}$ retrieves (x_{ID_j}, d_{ID_j}) and does as follows:

i. It randomly selects $r_1 \in Z_q$ to calculate $U = -r_1 \cdot \pi_p aP$.
ii. Next, it computes $\omega = (f_p \cdot (x_{ID_j} + d_{ID_j})) \cdot U$ and $Y = H_4(ID_i, ID_j, \omega, U)$.
iii. Then it calculates $C = prefix \oplus Y$.
iv. And it determines $r_1 = H_4(U, C, pk_p)$ and adds (U, C, T_p, X_p, r_1) to L_{H_4}.
v. The value $\varphi = r_1 \cdot r_2 \cdot \zeta_p$ is set, where $r_2 = H_5(U, prefix)$.

vi. Finally, the signcryptext $\sigma = (U, C, \varphi)$ is returned to $\mathcal{A}_I$.

Note that,

$$\begin{aligned}\varphi P &= r_1 \cdot r_2\zeta_p P \\ &= -r_1 \cdot \pi_p aP + r_1(r_2\zeta_p P + (\pi_p aP - h_p aP) \\ &+ h_p P_{pub}) \\ &= U + r_1 \cdot (r_2 X_s + T_s + h_p \cdot P_{pub}).\end{aligned}$$

(c) If $ID_i = ID_r^*$ and $ID_j \neq ID_p^*$, this case is analogous to previous situation with $ID_i = ID_p^*$ and $ID_j \neq ID_r^*$.

(d) If $ID_i \in \{ID_p^*, ID_r^*\}$ and If $ID_j \in \{ID_r^*, ID_p^*\}$, $\mathcal{B}$ aborts the game.

(6) Forgery Phase: In the end, $\mathcal{A}_I$ outputs a forged signcryptext $\sigma^* = (\varphi^*, U^*, C^*)$ corresponding to ID_p^* and ID_r^*. The forged signcryptext σ^* satisfy the following relation:

$$\varphi^* P = U^* + r_1^*(r_2 \cdot X_p + T_p + h_p P_{pub}).$$

Then we have

$$\begin{aligned}\varphi^* aP &= aU^* + a \cdot r_1^*(r_2 \cdot X_p + T_p + h_p P_{pub}) \\ &= aU^* + r_1^*(ar_2 \cdot X_i + a\pi_p aP),\end{aligned}$$

$$r_1^*\pi_p a^2 P = (\varphi^* - r_1^* r_2\zeta_p)aP - aU^*,$$

$$a^2 P = (r_1^*\pi_p)^{-1}((\varphi^* - r_1^* r_2\zeta_p)aP - aU^*),$$

where aU^* can be computed by ω^*:

$$aU^* = \frac{\omega^* - f_p\zeta_r U^*}{f_p\pi_r}.$$

Thus, the Square-DH problem can be solved:

$$a^2 P = \frac{(\varphi^* - r_1^* r_2\zeta_p)aP - \frac{\omega^* - f_p\zeta_r U^*}{f_p\pi_r}}{r_1^*\pi_p}.$$

Probability Analysis: Based on the game described above, it is evident that if the game without abort, $\mathcal{B}$ is capable of solving the square-DH problem. The probability of an abort occurring during the private key extract query is $2/q_p$, and the probability of partial private key query aborting is $\frac{2+q_r}{q_{pp}}$. In the signcryption query, the probability of an abort is $2/q_{sc}^2$. Additionally, it is crucial to confirm that ω^* has been issued the H_3-oracle query. Therefore, the probability of solving the square-DH problem can be expressed

$$\varepsilon \geq (1 - \frac{2+q_r}{q_{pp}})(1 - \frac{2}{q_p})(1 - \frac{2}{q_{sc}^2})\epsilon_3. \quad \square$$

Lemma 4. *Within the ROM, if $\mathcal{A}_{II}$ who breaks the EUF-CMA-II security of the proposed scheme with advantage ϵ_4, it is possible to construct an algorithm $\mathcal{B}$ capable of solving the square-DH with non-negligible advantage*

$$\varepsilon \geq (1 - \frac{2}{q_p})(1 - \frac{2}{q_{sc}^2})\epsilon_4.$$

Proof. Consider an instance of the square-DH problem represented as (P, aP), we attempt to compute $a^2 P$ by conducting an interactive game between $\mathcal{B}$ and $\mathcal{A}_{II}$.

(1) Setup: This phase is similar to Lemma 2.

(2) $H_1 \sim H_5$-Oracle Queries: $\mathcal{A}_I$ issues a hash query, $\mathcal{B}$ returns the corresponding value by checking the list $L_{H_1} \sim L_{H_5}$.

(3) Creating-User Oracle: When $\mathcal{A}_{II}$ issues this query with ID_i, if $ID_i \notin \{ID_p^*, ID_r^*\}$, $\mathcal{B}$ randomly chooses $t_i, x_{ID_i} \in Z_q$ to set $T_i = t_i P$, $X_i = x_{ID_i} P$ and $d_{ID_i} = t_i + sh_i$. When $ID_i \in \{ID_p^*, ID_r^*\}$, $\mathcal{B}$ randomly chooses $\zeta_i, d_{ID_i} \in Z_q$ to set $X_i = \zeta_i aP$, $T_i = d_{ID_i} P - h_i sP$, $x_{ID_i} = \perp$. Finally, $\mathcal{B}$ inserts $(ID_i, T_i, X_i, x_{ID_i}, d_{ID_i}, h_i, \zeta_i)$ in the list L which is initially empty.

(4) Public-Key/Private-Key-Extract/Signcryption Oracle: These oracles are similar to Lemma 3.

(5) Forgery Phase: Finally, $\mathcal{A}_I$ outputs a forged signcryptext $\sigma^* = (\varphi^*, U^*, C^*)$ with the producer's identity ID_p^* and the receiver's identity ID_r^*. The forged signcryptext should satisfy the following relation.

$$\begin{aligned}\varphi^* aP &= aU^* + a \cdot r_1^*(r_2 \cdot X_p + T_p^* + h_p^* P_{pub}) \\ &= aU^* + r_1^*(ar_2 \cdot \zeta_p a^2 P + ad_{ID_p} P),\end{aligned}$$

$$r_1^* r_2 \cdot \zeta_p a^2 P = (\varphi^* - r_1^* d_{ID_p})aP - aU^*,$$

$$a^2 P = \frac{(\varphi^* - r_1^* d_{ID_p})aP - aU^*}{r_1^* r_2 \cdot \zeta_p},$$

where aU^* can be obtained similar to Lemma 3.

Therefore, it means that the square-DH problem can be solved. However, this contradicts the difficulty of solving the square-DH problem.

Probability Analysis: According to the above game, we can see that if the game does not abort, $\mathcal{B}$ can solve the square-DH. The probability of aborting during the private key extract query is $2/q_p$, in the signcryption query, the probability of aborting is $2/q_{sc}^2$. Therefore, the probability of solving the square-DH problem is at least

$$\varepsilon \geq (1 - \frac{2}{q_p})(1 - \frac{2}{q_{sc}^2})\epsilon_4. \quad \square$$

6. Performance analysis

In this section, we first evaluate the computation cost of authentication, which mainly comes from the verification time of the signature. We also compare BAS-NDN with the recently certificateless signcryption schemes. Finally, we estimate network throughput and additional storage costs.

6.1. Computation costs

To assess the computational overhead incurred by our scheme at the router, we evaluate the time for IU authentication, which is a combination of the time of BC retrieving the relevant transaction and the time of verifying the prefix. Since this scheme uses private chains, the time to retrieve content (i.e., pre_Tx) is negligible, therefore the time delay for IU authentication is mainly based on the verification time of the signature. Furthermore, according to the model introduced in Conti et al. (2019), we calculate how this scheme affects the overall throughput of the router as the producer mobility increases. The throughput refers to the router's ability to process and forward data, i.e., the total number of interest packets successfully transmitted within a given time frame.

The simulation uses Ryzen 7 7700@8 × 3800 MHz as the reference hardware and the cryptographic system (Bernstein and Lange) as the benchmark to obtain the signature-based verification time. Table 3 and Fig. 6 show the time required to verify a message (59 bytes in size) with different public key signature schemes. The producer publishes an IU and verifies it at the router, which is signed by the producer's private key. The results indicate that our scheme has low computational overhead. Subsequently, we analyze the impact of scheme authentication delay on the throughput of edge routers.

We use the model presented in Conti et al. (2019) to calculate the change in router throughput as producer mobility increases, by setting κ as the ratio of the total number of standard datagrams received by IUs at the router interface. The throughput of the router is quantified as λ (packets/second).

$$\lambda = \frac{1 - \kappa}{\tau_{process} + (\kappa * \tau_{auth})},$$

Table 3
Comparative of public key signature.

Cryptosystem type								
RSA 512-bit	RSA 1024-bit	RSA 2048-bit	RSA 3072-bit	DSA 512-bit	DSA 1024-bit	DSA 2048-bit	Pairing-based	ECC-based
9.25 μs	19.32 μs	61.39 μs	147.55 μs	10.70 μs	23.54 μs	62.76 μs	1315.79 μs	7.25 μs

Table 4
Parameterization of the cryptosystem.

Cryptosystem type	Curve	$\|Z_p\|$	Point size
Pairing-based	$y^2 = x^3 + x \pmod p$	160 bits	$\|G_{pb}\| = 1024$ bits
ECC-based	$y^2 = x^3 + ax + b \pmod p$	160 bits	$\|G_{ecc}\| = 320$ bits

Table 5
Time for basic operations.

Nation	Operation	Time (ms)
T_{psm}	Scalar multiplication on bilinear pairs	1.88
T_{sm}	Point addition on elliptic curves	1.81
T_p	Bilinear pair operation	3.86
T_{inv}	Modulo inverse operation	0.05

Table 6
Comparison of schemes in terms of performance.

Schemes	Signcryption costs	Unsigncryption costs
Gong et al. (2022)	$3T_{psm} + T_{inv}$	$2T_{psm} + T_p$
Chen et al. (2022)	$3T_{psm}$	$2T_{psm} + 5T_p$
Chen et al. (2023)	$4T_{sm} + T_{inv}$	$5T_{sm}$
Dai and Xu (2023)	$3T_{sm}$	$5T_{sm}$
Our scheme	$3T_{sm}$	$4T_{sm}$

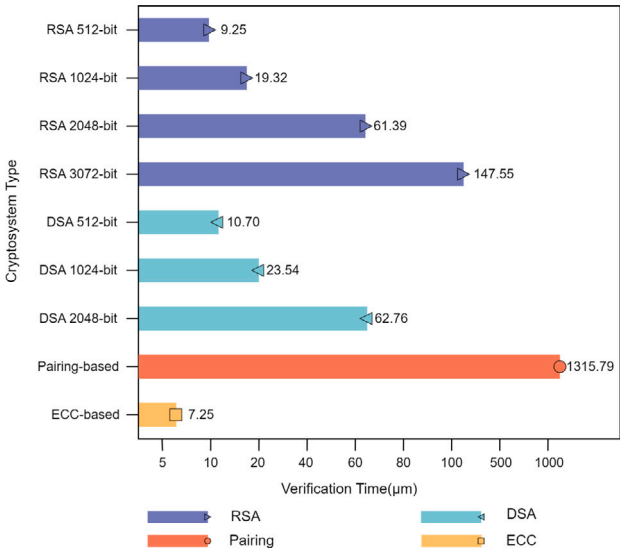

Fig. 6. Comparative of public key signature.

where $\tau_{process}$ refers to the average time taken by the router to process a normal packet and τ_{auth} denotes the delay in verifying the prefix. Let the maximum throughput of the edge router is 0.5 MB/s. Consequently, we find $\tau_{process} = 2$ μs. τ_{auth} using the data provided in Table 3. As shown in Fig. 7, the ECC-based scheme provides approximately 80% of the router's raw throughput (i.e., without IUs authentication) when the mobility rate reaches 5%, which is the highest of the signature-based schemes. For increases in mobility traffic ranging from 10% to 15%, the ECC-based scheme can maintain 66% to 55% throughput, which is the most efficient program available.

The comparison experiments are deployed on an Asus laptop configured with Intel Core i7-13700H CPU, 16 GB RAM. We use Charm-Crypto 0.50 cryptographic library and Python to code simulation experiments in Ubuntu 16.0 system. For comparison purposes, we standard the parameter length $|Z_p| = 160$ bits and consider the time-consuming operations, as detailed in Tables 4 and 5.

Due to the computation cost of certificateless signcryption schemes is mainly decided by the signcryption and unsigncryption algorithms, we focus only on these two phases. The cost overview of the related certificateless signcryption schemes (Gong et al., 2022; Chen et al., 2022, 2023; Dai and Xu, 2023) is shown in Table 6. Specifically, the scheme (Gong et al., 2022) takes $3T_{psm} + T_{inv} = 5.69$ ms and $2T_{psm} + T_p = 6.90$ ms to generate and decrypt a ciphertext, respectively. While in the scheme (Chen et al., 2022), the time in the signcryption and unsigncryption phase is $3T_{psm} = 5.64$ ms and $2T_{psm} + 5T_p = 23.05$ ms, respectively. The above two scheme are using bilinear pairing groups. For the scheme (Chen et al., 2023), the time required for the signcryption and unsigncryption phase is $4T_{sm} + T_{inv} = 7.29$ ms and $5T_{sm} = 9.05$ ms, respectively. In our scheme, the time is $3T_{sm} = 5.43$ ms and $4T_{sm} = 7.24$ ms during the signcryption and unsigncryption phase, respectively. In the scheme (Dai and Xu, 2023), its time in the signcryption and unsigncryption phase is $3T_{sm} = 5.43$ ms and $5T_{sm} = 9.05$ ms, respectively. As shown in Fig. 8, it can be concluded that our scheme outperforms several other schemes in computation cost, and it has the least running time.

6.2. Additional storage costs

The storage cost associated with the proposed scheme is related to the blockchain size. Each router is responsible for storing both the blockchain and the forwarding state. Following the initial registration of the producer, the blockchain size increases as new transactions are generated. This is related to the producer issues legitimate IUs in each movement event. Therefore, the additional storage cost incurred by the scheme can be expressed as follows:

$$Storage\_cost = N_\kappa \times size\_Tx, \tag{1}$$

where N_κ is the number of IU generated by the mobile event and $size\_Tx$ denotes the size of a single transaction required for prefix authentication. Let the size of Tx be 59 bytes (Bernstein and Lange). Fig. 9 shows the change in storage costs as the number of mobility events increases. In a mobile EPC network, the number of mobile subscribers is around 1 million. Consider the worst case, if each mobile producer is creating an IU message, i.e., triggering a mobile event, the storage cost per router is about 60 MB. Fortunately, modern routers are equipped with sufficient storage capacity to accommodate this data. Furthermore, the network owner can periodically delete blocks that are outdated or no longer needed.

6.3. Discussion

In terms of computation costs, we first evaluate the time for IU authentication, which is mainly based on the verification time of the signature. As shown in Table 3 and Fig. 6, the results indicate that our scheme exhibits low computational overhead. Subsequently, we calculate how this scheme affects the overall throughput of the router as the producer mobility increases. As illustrated in Fig. 7, the ECC-based scheme achieves the highest raw throughput among the signature-based schemes, approximately 80%, when the mobility rate reaches 5%. Additionally, we compare the proposed certificateless signcryption scheme with several recent schemes in the main computational cost with signcryption and unsigncryption algorithms. Table 6 and

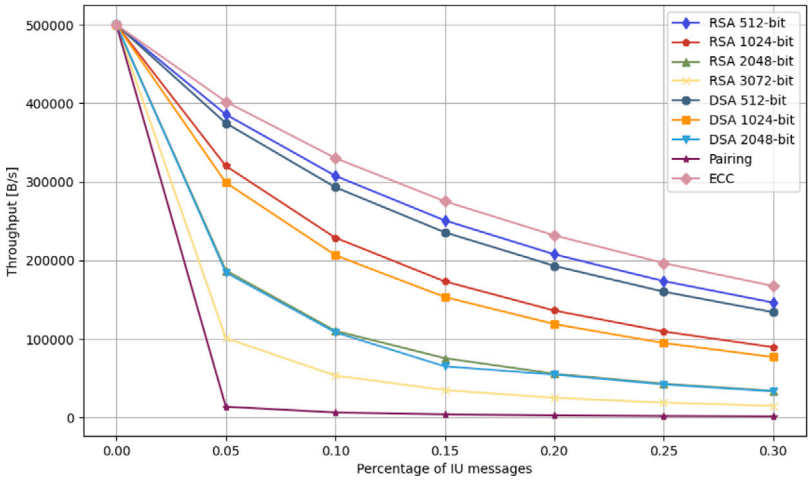

Fig. 7. Comparative of edge router throughput.

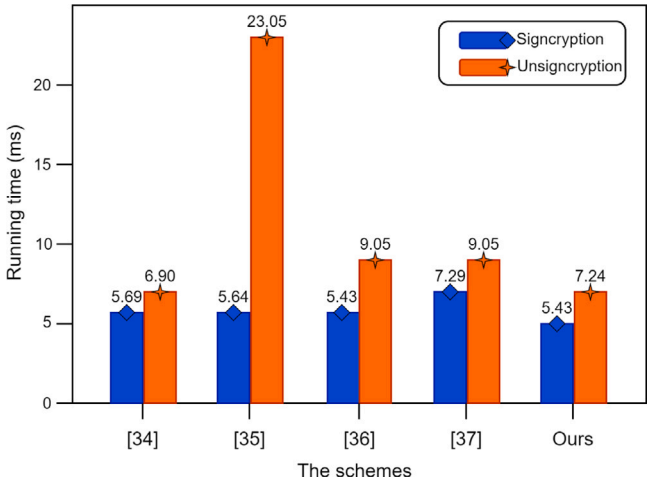

Fig. 8. Running time comparison of signcryption and unsigncryption.

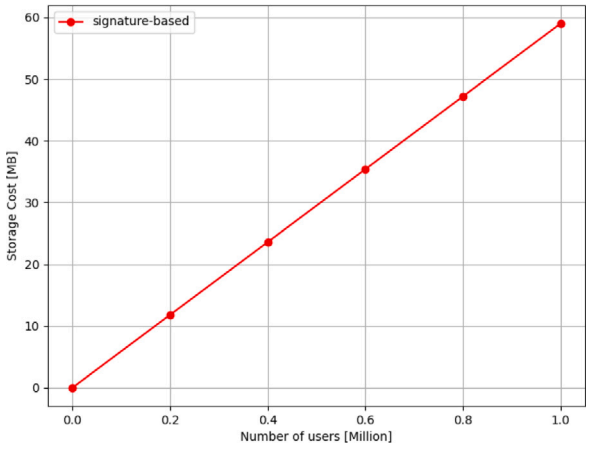

Fig. 9. Additional storage cost.

Fig. 8 show that our scheme outperforms several other schemes in computation cost, and has the least running time.

In terms of storage costs, a router's storage cost is mainly determined by the number of producer mobility events. As shown in Fig. 9, although 1 million producers generate mobility events in the network, the storage cost per router is about 60 MB, a capacity that modern routers can easily handle.

7. Conclusion

In this work, we observed that a consumer-driven NDN network is less capable of supporting producer mobility, and that existing trace-based approaches allow insecure interactions between producers and routers. To address these problems, we proposed a novel BC-based authentication certificateless scheme to ensure secure producer mobility within NDN. We proved the security of our scheme in the ROM. The security and performance analysis shows that the proposed scheme performs significantly better when compared to state-of-the-art certificateless signcryption schemes, and it efficiently mitigates prefix hijacking attacks. Besides, the proposed scheme can maintain the router's original throughput up to 80%, enabling prefix authentication at the line rate. Regarding storage requirements, the scheme is capable of accommodating billions of mobile producers while only consuming tens of megabytes per router.

CRediT authorship contribution statement

Guangquan Xu: Validation, Project administration, Funding acquisition, Conceptualization. **Chenghe Dong:** Writing – review & editing, Writing – original draft, Validation, Methodology, Formal analysis, Data curation, Conceptualization. **Cong Wang:** Writing – review & editing, Methodology, Conceptualization. **Feng Feng:** Validation, Project administration, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported in part by the National Key R&D Program of China under No. 2023YFB2703800, the National Science Foundation

of China under Grants U22B2027, 62172297, Hainan Province Science and Technology Special Fund (Grant No. ZDYF2024GXJS008), Guangxi Science and Technology Plan Project (Guangxi Science and Technology Base and Talent Special Project) under grant AD23026096 (Application Number 2022AC20001).

Data availability

The authors do not have permission to share data.

References

Abrar, A., Arif, A.S.C.M., Zaini, K.M., 2022. Producer mobility support in information-centric networks: research background and open issues. Int. J. Commun. Netw. Distrib. Syst. 28 (3), 312–336.

Al-Riyami, S.S., Paterson, K.G., 2003. Certificateless public key cryptography. In: ASIACRYPT, vol. 2894. pp. 452–473.

Bernstein, D.J., Lange, T., eBACS: ECRYPT Benchmarking of Cryptographic Systems. https://bench.cr.yp.to/.

Chen, X., He, D.B., Khan, M.K., Luo, M., Peng, C., 2023. A secure certificateless signcryption scheme without pairing for internet of medical things. IEEE Internet Things J. 10 (10), 9136–9147.

Chen, Y.S., Hsu, C.S., Huang, D.Y., 2014. A pipe-assisted mobility management in named data networking networks. In: 16th Asia-Pacific Network Operations and Management Symposium. pp. 1–4.

Chen, J., Wang, L., Wen, M., Zhang, K., Chen, K., 2022. Efficient certificateless online/offline signcryption scheme for edge iot devices. IEEE Internet Things J. 9 (11), 8967–8979.

Cisco, 2020. Cisco annual internet report (2018–2023) white paper.

Compagno, A., Zeng, X., Muscariello, L., Carofiglio, G., Augé, J., 2017. Secure producer mobility in information-centric network. In: Proceedings of the 4th ACM Conference on Information-Centric Networking. ICN '17, ACM, pp. 163–169.

Conti, M., Hassan, M., Lal, C., 2019. BlockAuth: BlockChain based distributed producer authentication in ICN. Comput. Netw. 164, 106888.

Dai, C., Xu, Z.W., 2023. Pairing-free certificateless aggregate signcryption scheme for vehicular sensor networks. IEEE Internet Things J. 10 (6), 5063–5072.

Das, S., Misra, A., Agrawal, P., 2000. TeleMIP: telecommunications-enhanced mobile IP architecture for fast intradomain mobility. IEEE Pers. Commun. 7 (4), 50–58.

Dorri, A., Kanhere, S.S., Jurdak, R., Gauravaram, P., 2017. LSB: a lightweight scalable blockchain for IoT security and privacy. CoRR abs/1712.02969.

Dukkipati, C., Zhang, Y., Cheng, L.C., 2018. Decentralized, BlockChain based access control framework for the heterogeneous internet of things. In: Proceedings of the Third ACM Workshop on Attribute-Based Access Control. ABAC'18, pp. 61–69.

Dulal, S., et al., 2022. Building a secure mhealth data sharing infrastructure over NDN. In: Proc. 9th ACM Conf. Inf. Centric Netw.. pp. 114–124.

Fang, C., Yao, H., Wang, Z., Wu, W., Jin, X., Yu, F.R., 2018. A survey of mobile information centric networking: research issues and challenges. IEEE Commun. Surv. Tutor..

Farahat, H., Hassanein, H.S., 2017. Proactive caching for producer mobility management in named data networks. In: 13th International Wireless Communications and Mobile Computing Conference. IWCMC, pp. 171–176.

Fotiou, N., Polyzos, G.C., 2016. Decentralized name-based security for content distribution using BlockChains. In: IEEE Conference on Computer Communications Workshops. pp. 415–420.

Gao, S., Zhang, H., 2016. Scalable mobility management for content sources in named data networking. In: 13th IEEE Annual Consumer Communications Networking Conference. CCNC, pp. 79–84.

Gohar, M., Khan, N., Ahmad, A., Najam-Ul-Islam, M., Sarwar, S., Koh, S.-J., 2018. Cluster-based device mobility management in named data networking for vehicular networks. Mob. Inf. Syst. 1–7.

Gong, B., Wu, Y., Wang, Q., Ren, Y.-h., Guo, C., 2022. A secure and lightweight certificateless hybrid signcryption scheme for internet of things. Future Gener. Comput. Syst. 127, 23–30.

Hajjar, M., Aldabbagh, G., Dimitriou, N., 2015. Using clustering techniques to improve capacity of LTE networks. In: 21st Asia-Pacific Conference on Communications. APCC, pp. 68–73.

Hlaing, H.H., Funamoto, Y., Mambo, M., 2021. Secure content distribution with access control enforcement in named data networking. Sensors 21 (13), 4477.

Huston, G., Rossi, M., Armitage, G., 2011. Securing BGP x2014; a literature survey. IEEE Commun. Surv. Tutor. 13 (2), 199–222.

Jacobson, V., 2006. A new way to look at networking. Google Tech Talk..

Kar, P., Chen, R., Qian, Y., 2022. An efficient producer mobility management technique for real-time communication in ndnbased remote health monitoring systems. Smart Heal. 26.

Khalid, A., Rehman, R.A., Burhan, M., 2023. CBILEM: A novel energy aware mobility handling protocol for SDN based NDN-MANETs. Ad Hoc Netw. 140.

Kim, D., Ko, Y.B., 2017. On-demand anchor-based mobility support method for named data networking. In: 19th International Conference on Advanced Communication Technology. ICACT, pp. 19–23.

Kosba, A., Miller, A., Shi, E., Wen, Z., Papamanthou, C., 2016. Hawk: the BlockChain model of cryptography and privacy-preserving smart contracts. In: 2016 IEEE Symposium on Security and Privacy. pp. 839–858.

Li, B., Ma, M., 2023. An advanced hierarchical identity-based security mechanism by blockchain in named data networking. J. Netw. Syst. Manag. 31 (1), 1–32.

Ma, J., Li, T., Cui, J., Ying, Z., Cheng, J., 2021. Attribute-based secure announcement sharing among vehicles using blockchain. IEEE Internet Things J. 8 (13), 10873–10883.

Mars, D., Mettali Gammar, S., Lahmadi, A., Azouz Saidane, L., 2019. Using information centric networking in internet of things: A survey. Wirel. Pers. Commun. 105 (1), 87–103.

Muhammad, S., 2013. Cluster-based mobility support in content centric networking. Res. Not. Inf. Sci. (RNIS) 14, 441–444.

Naeem, M.A., Nor, S.A., Hassan, S., Kim, B.-S., 2018. Performances of probabilistic caching strategies in content centric networking. IEEE Access 58807–58825.

Ren, F., Qin, Y., Zhou, H., Xu, Y., 2016. Mobility management scheme based on software defined controller for content-centric networking. In: IEEE Conference on Computer Communications Workshops. pp. 193–198.

Rui, L., Yang, S., Huang, H., 2018. A producer mobility support scheme for real-time multimedia delivery in named data networking. Multimedia Tools Appl. 77 (4), 4811–4826.

Saxena, D., Raychoudhury, V., Suri, N., Becker, C., Cao, J., 2016. Named data networking: a survey. Comput. Sci. Rev. 19, 15–55.

Sultan, N.H., Varadharajan, V., Dulal, S., Camtepe, S., Nepal, S., 2024. NDN-RBE: An accountable privacy aware access control framework for NDN. Comput. J. 67 (4), 1572–1589.

Xia, Q., et al., 2024. PRIDN: A privacy preserving data sharing on named data networking. IEEE Trans. Inf. Forensics Secur. 19, 677–692.

Xylomenos, G., Vasilakos, X., Tsilopoulos, C., Siris, V.A., Polyzos, G.C., 2021. Caching and mobility support in a publish–subscribe internet architecture. IEEE Commun. Mag. 50 (7), 52–58.

Xylomenos, G., Ververidis, C.N., Siris, V.A., Fotiou, N., Tsilopoulos, C., Vasilakos, X., Katsaros, K.V., Polyzos, G.C., 2014. A survey of information-centric networking research. IEEE Commun. Surv. Tutor. 16 (2), 1024–1049.

Yan, Z., Park, Y.J., Leau, Y.B., Ren-Ting, L., Hassan, R., 2020. qNetwork mobility support in named data networking. In: International Conference on Information Networking. ICOIN, pp. 16–19.

Zhang, L., Afanasyev, A., Burke, J., Jacobson, V., Claffy, K., Crowley, P., Papadopoulos, C., Wang, L., Zhang, B., 2014. Named data networking. ACM SIGCOMM Comput. Commun. Rev. 44 (3), 66–73.

Zhang, J., Dong, C., Liu, Y., 2024. Efficient pairing-free certificateless signcryption scheme for secure data transmission in IoMT. IEEE Internet Things J. 11 (3), 4348–4361.

Guangquan Xu is a Ph.D. and full professor at the Tianjin Key Laboratory of Advanced Networking (TANK), College of Intelligence and Computing, Tianjin University, China. He is also a joint-professor of School of Big Data, Huanghai University, Qingdao. He received his Ph.D. degree from Tianjin University in March 2008. He is an IET Fellow, members of the CCF and IEEE. He is the director of Network Security Joint Lab and the Network Attack & Defense Joint Lab. He has published 100+ papers in reputable international journals and conferences, including USENIX Security, IEEE TIFS, IEEE TDSC, JSAC and so on. He served as a TPC member for FCS2020, ICA3PP2021, IEEE UIC 2018, SPNCE2019, IEEE UIC2015, IEEE ICECCS 2014, and editors for IEEE IoT J, DCN, , and so on. His research interests include cybersecurity and trust management. (Email:losin@tju.edu.cn)

Chenghe Dong was born in Heilongjiang Province, China, in 1999. She is currently pursuing the Ph.D. degree with Tianjin University, Tianjin and her M.S. degree in North China University of Technology, Beijing, in 2024. Her main research interests include computer networks, applied cryptography, Blockchain.(Email:chenghe_d@163.com)

Cong Wang was born on February 10, 1988 in Hunan, China. She received her M.S. and Ph.D. degree in Computer Science, Tianjin University, China in 2012 and 2017 respectively. She is currently a teacher in Tianjin University of Science & Technology. Her research interest includes network security and authentication scheme design, Internet of Things. (Email:wangcongjcdd@tust.edu.cn)

Feng Feng is currently a full professor and Dean of the School of Information Engineering, Ningxia University, Yinchuan 750021, China. His research topics are about Information systems integration and application, Internet of Things, and so on. Dr. Feng published more than 60 papers, and received many projects, including 1 National Nature Science Foundation of China, 2 key projects in Ningxia Province, 3 projects about Ningxia Natural Science Foundation, and so on. He also obtained many awards, including the "Ningxia Science and Technology Progress Award", the "Ningxia University Teaching Achievement Award" (three times). He is a distinguished expert of "Yinchuan Innovation and Development Think Tank". (Email:feng_f@nxu.edu.cn)