Computer Science and Information Systems 21(3):725–757 https://doi.org/10.2298/CSIS230328013G

A Practical and UC-Secure Decentralized Key Management and Authentication Scheme Based on Blockchain for VNDN

Xian Guo*, Sarah Almadhehagi, Tao Feng, Di Zhang, Yongbo Jiang, and Junli Fang

School of Computer and Communication, Lanzhou University of Technology
730050 Lanzhou, China
{Xian Guo}iamxg@163.com

Abstract. In Vehicular Named Data Networking (VNDN), most of the existing key management mechanisms still rely on the hierarchical key trust model or the Public Key Infrastructure (PKI), in which the centralized certificate authority is used as a Trusted Third Party (TTP) to provide a signature for the user's public key. Thus, the TTP becomes vulnerable to attacks and maybe there exists a singlepoint failure problem. In addition, the in-network caching in the VNDN may incur a threat to the system and make it is vulnerable to the DoS attack caused by Interest flooding aimed at the Content-Store. To tackle these security issues, we firstly propose an efficient decentralized key management solution based on blockchain for VNDN. Secondly, based on the proposed key management scheme, a lightweight mutual authentication scheme and a key agreement protocol for V2X are respectively proposed in this paper. Finally, we analyze the security attributes of our solutions in the Universally Composable (UC) framework. Our analysis results show that our novel schemes can meet the security requirements of our solutions. In addition, our experimental results show our new schemes have higher efficient, lower computation and communication costs than other related schemes.

Keywords: Vehicular Named Data Networking (VNDN), Key management, Blockchain, Authentication, Key agreement, Universally Composable (UC).

1. Introduction

The implementation of a new network architecture with an appropriate security mechanism should strive to minimize the cost of computing, storage, and transmission and while to ensure the confidentiality, integrity, and authenticity of a transmitted message. The present Internet of Vehicles (IoV) [1] uses an IP-based communication pattern for this information delivery method, where messages are shared with other vehicles that have IP addresses. Most of IoV applications are point-to-multipoint (P2M) in design, where communications produced by one vehicle are broadcasted and shared with other vehicles nearby. Moreover, because IP-based communication is more vulnerable to threat and data loss, it violates the security requirements of IoV. Thus, the current network architecture based on IP network cannot fully adapt to the dynamic topology environment of the vehicular networks. An alternative communication architecture that suits IoV applications and services must consider the requirements such as security and Quality of

* Corresponding author

Service. The design principle of the Information-Centric Networking (ICN), e.g. Named Data Networking (NDN) [2] differs significantly from the traditional IP network. The NDN focuses on the transmitted content itself instead of the location of the content generator. Therefore, the integrating of vehicular networks with the NDN architecture, which is often called Vehicular Named Data Networking (VNDN) [3], is considered a hot research area. Due to the sensitivity of the data shared between the vehicles, the NDN fully complies with the nature of IoV where transmitted data is of more important than the location of the data generator. The VNDN enables a vehicle to access critical information within the network. Based on this critical information obtained from a network, vehicles can take correct decisions in time to ensure transportation security and safety.

In VNDN, the authenticity and integrity of the information spread are important issues. In order to guarantee the integrity and authenticity of the data transmitted on a network, the content producer will sign the content carried in a data packet using its private key, which securely and effectively links a name to the data. A consumer (or router) can then verify the signature using the public key of the producer and assert the data's provenance in this way. The customer now trusts the integrity and authenticity of the data and doesn't need to worry about where or how it is acquired. In addition, this scheme promotes fine-grained trust and enables a consumer to verify if a public key owner is a reliable producer of a specific piece of data in a certain situation. Therefore, the secret key of a content producer is the same critical material as in traditional IP network. Thus, it is very important to securely manage these cryptographic materials so as to avoid attacks to the transmitted data. Most of the existing NDN key managements use a hierarchical key trust model [4], in which the root key is used as a trust anchor to provide a key signature for the user's public key. To verify the authenticity of the public key, we can retrieve the key chain by using the key's name. This method can avoid generation of a false message. However, in the application of VANET, most of the existing key management schemes are based on traditional Public Key Infrastructure (PKI) [5], which adopts the centralized management model, the root key may be vulnerable to attacks and a tampered threat. So, a scheme based on PKI can lead to a single point of failure. In addition, PKIs such as X.509 PKIX [6] and Web of Trust [7], are designed to certify that a public key indeed belongs to a user with a name and the principle behind the name, and depend on Certificate Authorities (CAs). Due to the unconstrained privileges of CAs, they become central points of failure of the entire network. If a CA is compromised, the attacker can bind a name to an unauthorized public key and produce false data, which may result in severe security problems. Therefore, the key management in NDN is still an important security problem.

In recent years, using the distributed, tamper-proof, traceable and publicly verifiable characteristics of blockchain [8]to explore blockchain-based decentralization solutions has become a research hotspot in various fields [9,10,11,12]. The blockchain make use of distributed storage, consensus mechanisms, smart contracts, and cryptography [13]. So, the communication quality of messages and the pace of convergence are crucial challenges in peer-to-peer (P2P) networks. The blockchain technology over IP still has some significant issues, such as a lack of hierarchical access efficiency. These issues can be effectively resolved by an adoption of blockchain technology over NDN, which provides a decentralized system and streamlines the design. The integration [14,15] of the NDN network with the blockchain technology improves the actual security of the stored infor-

mation and the forwarding process, and further enhances the overall performance of the network, and maximizes forwarding effectiveness. So, the combining of blockchain with VNDN [16] can address some security challenges in the VNDN system, which can clearly improve the efficiency and security of VNDN. The blockchain-based VNDN facilitates trust establishment between different entities in VNDN, and can ensure the transmission of the interest packet and the data packet in a secure way, and prevents attackers from leveraging the built-in design features, such as the utilization of the distributed and uncontrolled in-network caching, to launch attacks at the cache store, and from supplying illegitimate or unrestricted interests to launch an Interest flooding attack.

To solve the security issues of the traditional key managements, some schemes depend on the features of the blockchain technology are proposed to manage the user's keys [17,18,19]. The scheme in [17] introduces a key management mechanism for VNDN. The proposed scheme is used to solve the mutual trust establishment issue between different entities. However, this scheme does not support key agreement suited to fast and dynamic applications in VNDN. In addition, the scheme in [18] proposes a key management based on blockchain for NDN to tackle the issue of reduction of the mutual trust between entities, but this scheme does not support key registration and key update. As for the scheme [19], it proposes a decentralized public key management for NDN based on blockchain. However, this scheme does not support efficiently key management in terms of updating and revoking of the public key. In addition, conducting security evaluations based on the informal analysis method is a common method in the existing studies [17,18,19]. The informal security analysis cannot accurately reflect security concerns. Actually, using of the blockchain technology has potential advantages and defects. On the one hand, a trust chain for a public key can be built using the tamper-proof attribute of blockchain. However, this attribute poses certain challenges for key management, including key update and revocation.

The Universal Composable (UC) framework [20,21] allows modular design and analysis of complex cryptography protocols and ensures security when any multiple protocol instances are executed concurrently. It has become the theoretical basis and methodological guidance for the design and analysis of various composable protocols. Ran Canetti et al. [20,21] propose the concept of the 'Ideal Functionality' that used to capture the security properties of basic cryptographic primitives and propose the security evaluation methods in the UC framework. These ideal functionalities are widely used as standard ideal functionalities to modularly design and analyze various composable protocols. The UC framework is suitable to the design and analysis of the complex system combined the VNDN and the blockchain.

Given the limitations of the above existing schemes of key management, we explore the possibility of implementing a decentralized key management mechanism for VNDN by combining the blockchain technology with the NDN in this paper. In addition, we present a lightweight mutual authentication and key agreement solution based on the proposed key management mechanism to resolve privacy and security issues in the system. To tackle the above shortcomings of the blockchain-based key managements, smart contracts [22] are used to automatically manage the user's keys and implement the various functions of registering, updating, and revoking for the user's keys. Finally, we analyze our novel schemes in UC framework.

The remainder of the paper is organized as follows: We discuss the related works in Section 2.The preliminary knowledge adopted in this study is described in Section 3. Our system model is described in Section 4. In Section 5, we detail our novel solution. In Section 6, we analyze the scheme's security requirements by using the UC framework. In Section 7, we evaluate the scheme's performance. In Section 8, we conclude our works in this paper.

2. Related Work

The centralized management model was used by many existing key management schemes based on traditional PKI [5]. Consequently, PKI-based schemes can lead to a single point of failure. Moreover, PKIs, such as X.509 [6] and web of trust [7], rely on certificate authorities (CAs) to prove that a public key belongs exactly to a name. Given that CA privileges are unaffected, they become the central point of network failure. An attacker can bind a key name to an unauthorized public key if a CA is compromised, causing serious security issues.

In 2021, Hao Liu et al. proposed blockchain-based key management and green routing scheme for VNDN [17]. A blockchain-based key management scheme was presented to solve the mutual trust problem between domain nodes. The number of signature verifications is reduced using the scheme. Moreover, the process of key acquisition and verification is accelerated, and the NDN is suitable for IoV.

A blockchain-based key management scheme in NDN was proposed by Junjun Lou et al. [18] to address the lack of mutual trust between sites without trust anchors. The NDN public key content objects and the scheme for storing, verifying, and revocation are redesigned.

In 2018, Kan Yang et al. proposed BC-PKM [19] for NDN. It utilizes the decentralized and tamper-proof design of blockchains to register, query, update, validate, and revoke the public keys of important principals. This system takes advantage of the decentralized and tamper-proof nature of blockchains. Adversaries that compromise less than half of the public key miners can resist various attacks by using this scheme. The BC-PKM scheme was proposed to address the compromised CA problem.

In 2021, Anhao Xiang et al. [23] proposed a lightweight anonymous device authentication scheme. This scheme is based on NDN and is a representative implementation of ICN. It provides security features, such as mutual authentication, session key agreement, defense against cyberattacks, anonymity, and resilience against device capture attacks. The results of security analysis and performance evaluation indicated that the proposed scheme has lower computational and communication overheads than other state-of-the-art schemes. Compagno et al. [24] proposed an authentication protocol for the ICN network. The proposed OnboardICNg is a symmetric key cryptographic protocol. Given its design, it can reduce the number of packets forwarded in the network.

For NDN-VANET, Xian Guo et al. [25] proposed a receiver-forwarding decision scheme based on Bayesian to solve the broadcast storm problem caused by blind flooding of interest packets. A received interest packet can be adaptively forwarded by their solution. Experimental simulations demonstrate that the BRFD algorithm greatly reduces the redundancy of interest packets.

In 2023, an intelligent forwarding solution with privacy awareness PABRFD for NDN-VANET was proposed by Xian Guo et al. [26] by including homomorphic encryption (HE) into the enhanced BRFD. In PABRFD, the security and privacy concerns of information transferred among car nodes are addressed using a secure Bayesian classifier. They explicitly demonstrate that the new approach can meet security demands, and they put their solution into practice using the HE standard libraries CKKS and BFV. The experimental findings demonstrate that PABRFD can meet their anticipated performance needs.

Table 1. Notations of the paper

Notation	Description
/ndn/VNDN/V	The name prefix of the vehicle's nodes
/ndn/VNDN/RSU/node	The name prefix of the blockchain nodes
HF	The hash function
Sig	The signature algorithm
sK	The secret value
T_Z	The timestamp of the entity Z
K_Z	The authentication key value of the entity Z
PbK	The public key
PrK	The private key
enc	The public encryption algorithm
$Senc$	The symmetric encryption algorithm
SEK_{X-Y}	The session key between entities X and Y
M_r	The request message
M_s	The response message
SEK'_Z	The session key parameter of the entity Z

3. Preliminaries

3.1. Bivariate Polynomial Theory

Blundo et al. [27] proposed a bivariate polynomial in a key distribution scheme. This scheme uses the symmetry of the polynomial to establish the key and ensure communication security.In Eq. (1), a polynomial with two variables (x, y) and degree n has the properties of a bivariate polynomial:

$$f(x,y) = \sum_{i,j=0}^{n} a_{ij} x^i y^j \,. \tag{1}$$

A unique identifier is assigned to each node of the network. For example, node b has a polynomial segment $g_b\,(y)$, which can be obtained by the following function:

$$G_b\,(y) = f\,(b,y)\,. \tag{2}$$

Node b stores n coefficient $g_j\ (0 \leq j \leq n)$:

$$g_j = \sum_{i=0}^{n} a_{ij} b^i \ (0 \leq j \leq n) \,. \tag{3}$$

where b is the ID of the node, and g_i is the coefficient of y^j in the polynomial $f\,(b, y)$. Nodes b and p need to establish a symmetric key between each other by exchanging node IDs first. Then, let $y = b$; node p calculates $f\,(b, y)$. Let $y = p$; node b calculates $f\,(b, y)$. Given that $f\,(b, p) = f\,(p, b)$, nodes b and p compute the same value that can serve as the session key between them.

3.2. NDN Architecture

The rise of NDN [28] has attracted the attention of researchers. Using caching technology to improve the cache utilization in the network by caching data packets, simplifying the coordination between nodes, and reducing the delay for users to obtain data have become urgent concerns. NDN relies on the content of the data instead of the IP address. The data are retrieved by using the name of the data. In NDN, network communication is driven by receivers (that is, data consumers), mainly involving two kinds of packets: an interest packet and a data packet. The interest packet is a data packet sent when an NDN user initiates a request, whereas a data packet is a response from a network service provider to a user's request to satisfy a request. An NDN nod maintains the three data structures Forwarding Information Base (FIB), Pending Interest Table (PIT), and Content Store (CS). In case something goes wrong, it's a good idea to have a backup strategy. Interest packet outgoing interfaces are stored in a forwarding table known as FIB. The arrival interfaces of pending interest packets are stored in the PIT table, which also functions as a forwarding table for data packets.

3.3. Blockchain Technology

A blockchain [29] is a distributed public digital ledger that records transactions in distributed decentralized networks. It uses a peer-to-peer (P2P) network. The blockchain data structure is interpreted as a chain record of transaction blocks. It is organized and can be kept as a ledger or in an ordinary database. Each block is determined by a hash created using the SHA256 cryptographic hash algorithm on the block header.

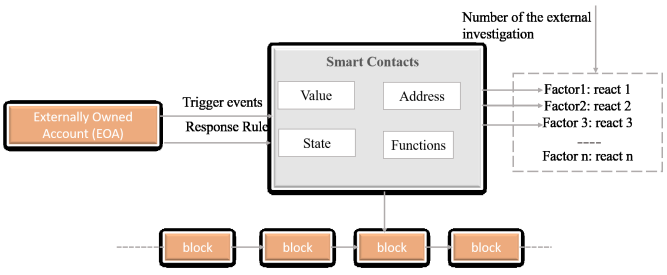

Fig. 1. The model of smart contracts

3.4. Smart Contracts

A public blockchain platform called Ethereum [22] enables anyone to create smart contracts and tokens. On Ethereum, decentralized virtual machines handle P2P agreements using their currencies. A Turing-complete scripting language is also available on Ethereum's programmable blockchain. The smart contract is triggered and executed when used. The model of smart contracts is shown in Figure 1.

3.5. Blockchain Over NDN

A few research projects have been created recently to incorporate the blockchain system into the NDN architecture, including [14,15]. These preliminary investigations mainly focus on how to develop forwarding and synchronization techniques for enabling all of the fundamental and essential functions of NDN as well as how to improve connections between NDN and blockchain.

The blockchain over NDN system is depicted in Figure 2. This system consists of users, NDN routers, and miners who can either be producers or consumers when constructing blockchain blocks depending on whether they are receiving blocks or transactions in real time. Also, although typically not in real-time, the user can assume the roles of a producer when generating a transaction or a consumer while receiving a block.

When a transaction is generated, the user assumes the role of the consumer and sends the transaction to the nearby router by encasing it in an interest packet. Each router forwards this interest packet to all of the outbound interfaces as soon as it receives it. When the interest reaches the miners, they can parse it to retrieve the transaction. Miners won't send any responses, and after the timeout, each router's PIT entries will be deleted.

3.6. UC Framework

The UC framework [20,21] is a paradigm for analyzing and guaranteeing the security of cryptographically composable protocols. The structure of the UC framework is formulated with some entities, such as an environment machine that interacts with the execution protocol and adversaries.The notion of this framework is based on emulation. For any real-world protocol π and an adversary A, an ideal protocol ϕ and an adversary S exist. In the UC framework, a protocol π performs UC simulation on protocol ϕ for an ideal functionality F if an ideal adversary S on the network exists for any possible adversary A on the network. In particular, for any environment ε, the probability that ε can tell the difference between the execution of the protocolπ with the adversary A and the execution of the protocol ϕ with the adversary S is neglible at most. However, the UC simulation ensures the strong correspondence between protocols.

The ideal functionality F is considered the most important part of the UC framework. It is a trusted party for achieving the security requirements of cryptographic protocols. It guarantees the secrecy, authenticity, or delivery of the communicated information. Most of the ideal functionalities have already been formulated, such as the signature functionality F_{SIG}, the public key encryption functionality F_{PKE}, the secure message transmission functionality F_{SMT}, the key exchange functionality F_{KE}, and the message authentication functionality F_{AUTH}.

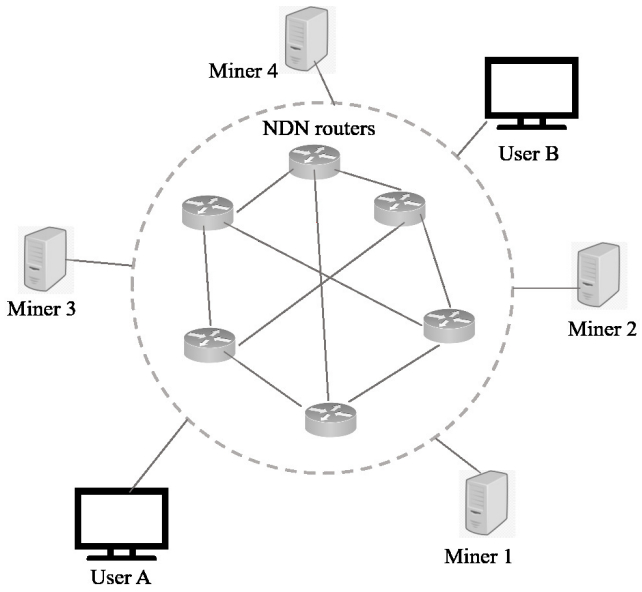

Fig. 2. Blockchain over NDN

3.7. Notations

The notations in the paper are described in Table 1.

4. System Model

4.1. System model

Our system divides into three levels: the first level includes vehicles, the second levle includes Vehicle Trusted Authority (VTA), the third level includes NDN–blockchain network (NDN-BCN). The system model is shown in Figure 3.

1) Vehicles: Vehicles are the main components in our considered VNDN network. Each vehicle is equipped with an onboard unit (OBU), which supports wireless communication for V2X. Every OBU has a tamper-proof device (TPD) to ensure that no private information is exposed or leaked. A hardware security module (HSM) is also installed in RSU's agent. Thus, a vehicle can securely store cryptographic materials. In our scheme, each vehicle can be a content consumer or a producer by sending a generated interest packet or a data packet of the owned content.
2) VTA: In our scheme, the VTA acts as a trusted authority. It is responsible for registering all the information of vehicles in advance, deploying a BCN and smart contract, initializing some system cryptography parameters, generating a transaction, and sending the transaction to the smart contract to register, update, and revoke vehicle information.

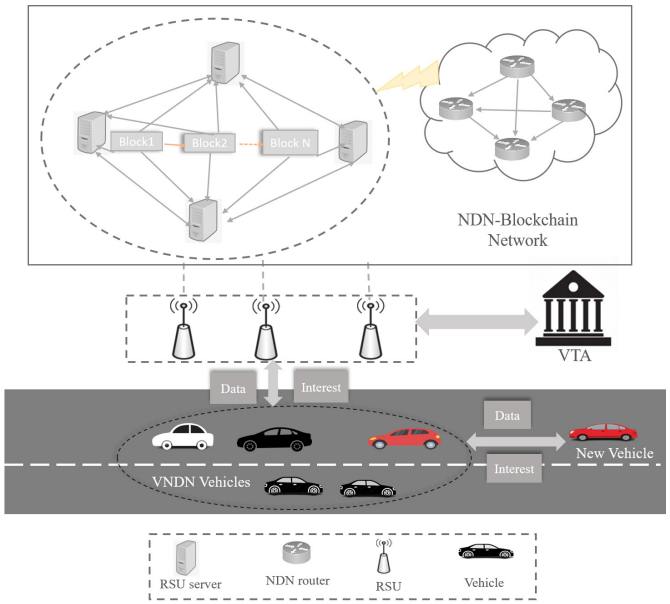

Fig. 3. System model

3) NDN-BCN: The blockchain in our scheme is implemented with the NDN protocol. NDN-BCN is a P2P network consisting of an RSU set, SRSU. Its function is to maintain and manage a public ledger for storing public keys and the identifier information of vehicles without the need for central PKI. Each RSU node plays a router role with the name prefix ndn/VNDN/RSU_i ($RSU_i \in SRSU$).

4.2. Threat Model

In our threat model, the attacks can be divided into two types: passive and active. The passive attack only eavesdrops a communication channel to gather sensitive information without modifying the messages eavesdropped. Two types of passive attacks are the release of message contents and traffic analysis. We denoted a passive attacker by the symbol Atc_1 in our proposed system.

The active attacks based on the Dolev-Yao attacker model [30]] can intercept messages, modify messages, or replay old messages to obtain confidential information. We denoted an active attacker by the symbol Atc_2 in our proposed system. The Atc_2 can intercept a transmitted interest/data packet, and try to obtain the intended information to conduct some attack. The Atc_2 even can masquerade or impersonate a legal new vehicle or RSU node to generate an interest packet or a data packet. Sometimes, the Atc_2 can compromise a RSU to obtain the legal identity of some vehicles. The Atc_2 also can flood the system with interest packets to crash the network or halt communication between different entities in the VNDN.

5. Proposed Scheme

Our proposed scheme consists of six stages: system initialization, registration stage, authentication stage, key agreement stage, public key update stage, and public key revocation stage. These stages are illustrated as follows:

5.1. System Initialization

In this stage, the VTA determines the algorithms and the parameters needed in our solution, such as hash function (e.g., SHA-256), EC-Schnorr signature algorithm, AES, and a bivariate polynomial function. The VTA deploys smart contracts on the blockchain to manage the public keys of the users.

Step 1: Building NDN-BCN and deploying of smart contracts

The VTA initializes a P2P BCN with an NDN prefix name. It also needs to create an Ethereum account for each NDN-BCN node by using an Ethereum wallet, such as MetaMask. The VTA creates smart contracts by using its account private key and address. Once the VTA has successfully deployed the smart contracts, a contract address is automatically created by NDN-BCN.

Step 2: Initialization of System Algorithms and Parameters

The VTA needs to determine a system parameter set $X = (p, a, b, G, n, h)$ to realize the public key cryptography system. These parameters determine an elliptic curve E on a finite field F_p, where p is a large prime number, and a and b are the parameters that define the elliptic curve E of a form $y^2 \pmod p = x^3 + ax + b \pmod p$. G is a generator denoted by a point (G_x, G_y) chosen from the elliptic curve, n is the order of the generator, h is SHA-256 hash function, and $h : \{0,1\}^* \longrightarrow F_p$ is a secure hash function. All the system parameters (p, a, b, G, n, h) are public to all the entities in the network.

Step 3: Bivariate polynomial initialization

A bivariate polynomial is introduced into the authentication stage in our solution to prevent a DoS attack and conduct a secure session key agreement. Moreover, two session keys are generated based on the polynomial for V2X. In the system initialization stage, VTA must set the relevant parameters of the bivariate polynomial. First, the VTA generates m bivariate polynomials $f(x, y)$ of degree n. They are denoted as $f_1, f_2, f_3, \ldots, f_m$. Then, the VTA generates n authentication key values K set $setK$. Each polynomial corresponds to an authentication key value K, and a polynomial fragment F is generated for the authentication value K using Eq. (3). Finally, the VTA randomly assigns an authentication key value $K(K \in setK)$ and a corresponding polynomial segment F to an RSU or a vehicle V when it needs to register on the VTA.

5.2. Registration Stage

The vehicle's information can be registered offline on the VTA in advance when a new vehicle needs to join the VDNN network and wants to communicate with other vehicles or access network services (RSU). Thus, the registration processes are illustrated as follows to ensure the legitimacy of the new vehicle:

Step 1: The vehicle owner submits all the registration information, such as the name of the vehicle owner, phone number, driving license, and physical vehicle number ID. The owner registers the vehicle's information to the VTA via a secure channel.

Step 2: The VTA generates a unique ID and a validity period (VP) for vehicle V. Then, the VTA selects $L(1 \leq L \leq m)$ polynomials from m bivariate polynomials. A corresponding authentication key value $K_v(K_v \in setK)$ and polynomial fragment F are distributed to the vehicle owner. According to the cryptography parameters (p, a, b, G, n, h), the user of the vehicle V computes a pair key (private key PrK and public key PbK). The vehicle V sends the public key PbK to the VTA. Afterward, the VTA generates a unique name $(PbKname)$ for the received public key PbK.

Step 3: The VTA binds the tuple $(ID, PbKname, PbK, andVP)$ and encodes it as hexadecimal codes. Then, it is compressed in JSON format. A new transaction is generated by the VTA. The generated hexadecimal code is embedded in its data field. Then, the VTA sends the transaction to the NDN-BCN for execution. It triggers the smart contract function register_ PK for vehicle registration. After the smart contract is successfully executed and deployed to the NDN-BCN, the transaction is stored in the NDN-BCN.

Step 4: After the registration is successfully executed, the secret materials $(ID, PrK, andVP)$ and the polynomial parameters are sent by the VTA to the vehicle. They are stored in the HSM, which is installed on the vehicle. Then, the vehicle V is implemented with the NDN prefix name /ndn/VNDN/V. The operation of the registration is described in Algorithm 1.

Algorithm 1 Aregister_PK function

Input ID, $PbKname$, PbK, VP
Output bool

1: **if** $msg.sender \neq VTA$ **then**
2: revert (); // Define an error and pass it while reverting a transaction
3: **else**
4: $users[ID] \leftarrow user(ID, PbKname, PbK, VP)$;
5: emit registerPK $(ID, PbKname, PbK, VP)$;
6: **end if**
return true;

5.3. Authentication Stage

After the registration stage is successfully completed, the authentication protocol is vital to realizing the system's security. This stage is described as follows and illustrated in Figure 4.

Step 1: When a new vehicle wants to participate and share information in VNDN, the RSU in the vehicle's communication range should first authenticate the legitimacy of the vehicle before allowing it to join the VNDN network. Given the broadcasted message of the RSU, the new vehicle V can obtain the public key PbK_{RSU} and the authentication key value K_{RSU} of the RSU, which are in the same communication range as the new vehicle V. At the beginning, the vehicle V sends a request message M_r in an interest packet to the RSU. The vehicle V first calculates a shared secret value sK_v between the vehicle V and RSU, as shown in Eq. (4), by using the symmetric bivariate polynomial function in Eq. (3). Then, the vehicle V computes a hash value HF on the message

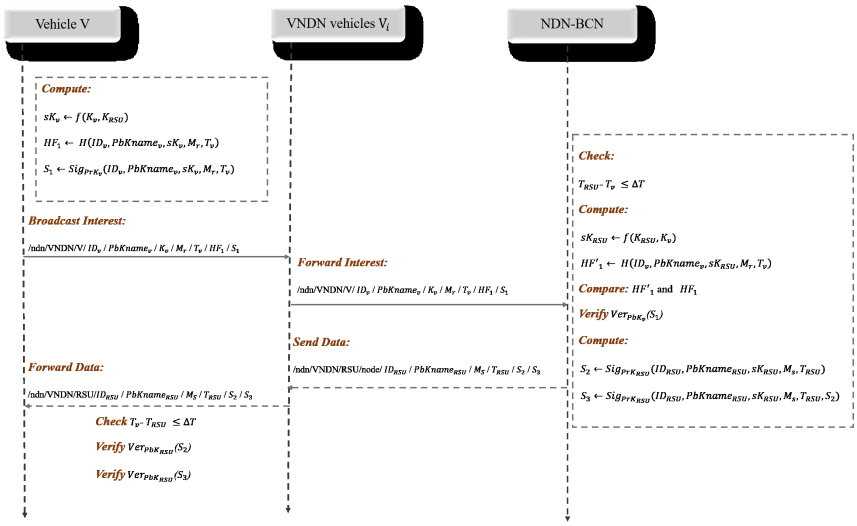

Fig. 4. An illustration of the authentication stage

$(ID_v, PbKname_v, sK_v, M_r, T_v)$ with HF_1 to guarantee the integrity of the message request, where the algorithm $H : \{0,1\}^* \longrightarrow \{0,1\}^h$ is a collusion-resistant hash function. The hash algorithm is described in Eq. (5), where M_r is the request message, and T_v is the vehicle current time stamp.

$$sk_v \leftarrow f\left(k_v, K_{RSU}\right). \tag{4}$$

$$HF_1 \leftarrow H\left(ID_v, PbKname_v, sK_v, M_r, T_v\right). \tag{5}$$

Step 2: The vehicle V needs to sign the request message to guarantee the authenticity of the message. The vehicle V signs the message $(ID_v, PbKname_v, sK_v, M_r, T_v)$ with its private key PrK_v, as shown in Eq. (6).

$$S_1 \leftarrow Sig_{PrK_v}\left(ID_v, PbKname_v, sK_v, M_r, T_v\right). \tag{6}$$

Afterward, the vehicle V generates an interest packet with the name /ndn/VNDN/V/ID_v /$PbKname_v$/K_v/M_r/T_v/HF_1/S_1 , where K_v is the authentication key value assigned to the vehicle V by the VTA. Then, this interest packet is broadcasted to the VNDN network, and a new PIT entry is created with the name prefix /ndn/VNDN/V/ID_v/$PbKname_v$/K_v /M_r/T_v/HF_1/S_1 in the router that received the interest packet copies.

Step 3: Upon receipt of the interest packet, the RSU calculates a shared secret value sK_{RSU} by using the same method used by the vehicle V, as shown in Eq. (7). Then, the RSU extracts HF_1 and S_1 from the interest packet to check the integrity and authenticity of the interest packet. If $T_{RSU} - T_v \leq \triangle T$, the RSU computes a commitment value $HF_1^{'}$ based on the secret key sK_{RSU}, as described in Eq. (8). If the $HF_1^{'}$ does not match the received HF_1, the session stops at this step.

$$sK_{RSU} \leftarrow f\left(K_{RSU}, K_v\right). \tag{7}$$

$$HF_1^{'} \leftarrow H\left(ID_v, PbKname_v, sK_{RSU}, M_r, T_v\right). \tag{8}$$

If the $HF_1^{'}$ matches the received HF_1, then the RSU checks the legitimacy of the new vehicle's public key by calculating the hash value of its public key in the received interest packet. It is compared with the hash value of the public key retrieved from the blockchain by using the public key's name $PbKname_v$. If these hash values are equal, the RSU believes that this new vehicle V is the owner of this public key, and it has been registered on the VTA before. After the legitimacy of the public key PbK_v is verified, the RSU uses PbK_v to verify the message signature S_1 to guarantee the authenticity of the requested message.

If the verification is successful, then RSU believes that the vehicle V is a legitimate vehicle. Moreover, the vehicle V is allowed to participate and share information within VNDN.

Step 4: The new vehicle in the VNDN also needs to authenticate the RSU that communicates with it to guarantee the legitimacy of this RSU and prevent an adversary from masquerading as an RSU. Thus, the RSU is required to sign a message M_s in a data packet forwarded to the vehicle V. The RSU signs the message $(ID_{RSU}, PbKname_{RSU}, sK_{RSU}, M_s, T_{RSU})$ with its private key PrK_{RSU} with S_2 to ensure the authenticy of the message, as shown in Eq. (9). Then, the RSU prepares a data packet with the message. The RSU also signs the data packet with S_3 to guarantee the authenticity of the data packet, as shown in Eq. (10). Then, the data packet with the name /ndn/VNDN/RSU/node/ID_{RSU}/$PbKname$/K_{RSU}/M_s/T_{RSU}/S_2/S_3 is sent. The data packet is appended in PIT and forwarded to the vehicle V.

$$S_2 \leftarrow Sig_{PrK_RSU}\left(ID_{RSU}, PbKname_{RSU}, sK_{RSU}, M_s, T_{RSU}\right). \tag{9}$$

$$S_3 \leftarrow Sig_{PrK_RSU}\left(ID_{RSU}, PbKname_{RSU}, sK_{RSU}, M_s, T_{RSU}, S_2\right). \tag{10}$$

Step 5: When the vehicle V receives the data packet, it first checks if $T_{RSU} - T_v \leq \triangle T$. Then, it retrieves the public key PbK_{RSU} of the RSU from the blockchain by using the name $PbKname_{RSU}$ to check the legitimacy and validity of the RSU's public key PbK_{RSU} by using the hash algorithm. After the legitimacy of PbK_RSU is verified, the vehicle V extracts S_2 and S_3 from the data packet. Then, it uses PbK_{RSU} to verify the message signature S_2 and the signature S_3 on the data packet to guarantee the authenticity of the response message and the data packet. If they are correct, the vehicle V believes that the RSU is legal. Otherwise, the vehicle V stops at this step. Finally, the lightweight mutual authentication between the vehicle V and RSU is completed.

5.4. Key Agreement Stage

In this stage, a vehicle $V_i (i = 1, 2, 3, \ldots n)$ in the VNDN helps complete a communication process between the vehicle V_i and an RSU node. Two session keys are established for V2X to guarantee the confidentiality of the content in an interest packet and a data packet containing the relevant parameters of the polynomial used in our solution. These keys are used for future communications based on the symmetry of the bivariate polynomial, as described in Section 3. The key agreement protocol is described in Figure 5. In this stage, the communications for V2X are based on symmetric key encryption. The key agreement process is described as follows:

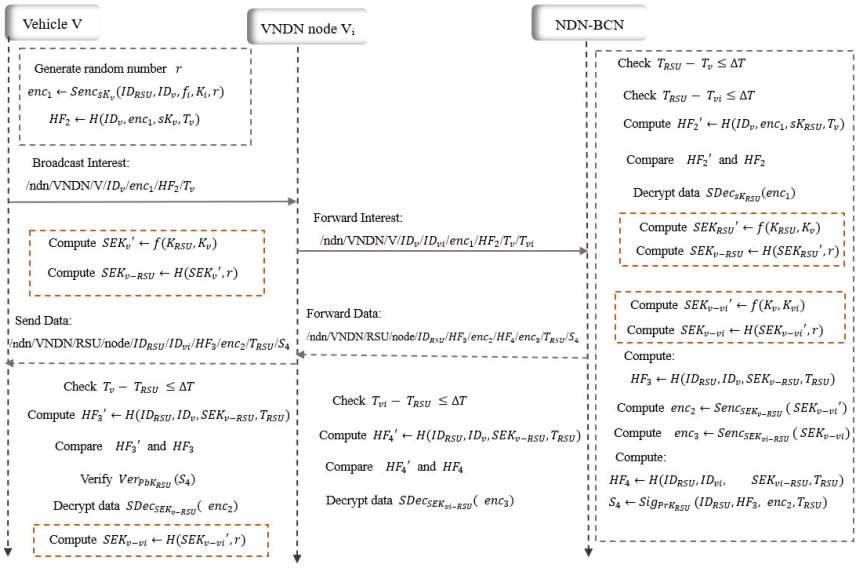

Fig. 5. An illustration of the key agreement stage

Step 1: After the above authentication process between the vehicle V and an RSU is completed, an agreement process of a session key SEK must be established between the vehicle V and RSU as SEK_{v-RSU}. The vehicle V first generates a random number r. Then, it uses the secret value sK_v described in Eq. (4) to encrypt the relevant message $(ID_{RSU}, ID_v, (f_i, K_i), r)$ described in Eq. (11), where (f_i, K_i) represents a random polynomial, which is chosen from the polynomial parameters assigned to the vehicle V by the VTA, as described in the registration stage. $Senc_{sK_v}(m)$ is a symmetric key encryption algorithm on message m using the secret value sK_v of the vehicle V. Similar to the case in the HF_1 generation process in the authentication stage, the vehicle V also needs to compute a hash authentication function HF_2 to prevent DoS attack, as described in Eq. (12).

$$enc_1 \leftarrow Senc_{sK_v}(ID_{RSU}, ID_v, f_i, K_i, r). \tag{11}$$

$$HF_2 \leftarrow H(ID_v, enc_1, sK_v, T_v). \tag{12}$$

Afterward, the vehicle V generates an interest packet with the name /ndn/VNDN/V/ID_v /enc_1/HF_2/T_v. A PIT entry is created with the name prefix /ndn/VNDN/V/ID_v/enc_1/HF_2 /T_v. Then, this interest packet is broadcasted to the VNDN network.

Step 2: Upon the receipt of the broadcasted interest packet, a trusted neighbor vehicle V_i, which is authenticated while it joins the VNDN network, helps complete the communication process. The vehicle V_i generates a new interest packet with a name prefix /ndn/VNDN/V/ID_v/ID_{vi}/enc_1/HF_2/T_v/T_{vi}. The new interest packet is added to the PIT. The interest packet is forwarded to the RSU located in its communication range.

Step 3: After the RSU receives the interest packet /ndn/VNDN/V/ID_v/ID_{vi}/enc_1/HF_2 /T_v/T_{vi} from the vehicle V_i, the RSU extracts enc_1 and HF_2 from the interest packet. If

$T_{RSU} - T_v \leq \triangle T$ and $T_{RSU} - T_{vi} \leq \triangle T$, then the RSU computes a hash function $HF_2^{'}$ based on the message in the received interest packet. If the $HF_2^{'}$ is equal to the received HF_2, the RSU decrypts enc_1 to obtain the relevant parameters for a session key establishment by using the secret value sK_{RSU} described in Eq. (7).

Step 4: First, the RSU computes a session key parameter $SEK_{RSU}^{'}$ using the corresponding polynomial parameters f_i and the fragments F simultaneously to generate a session key. The vehicle V computes a session key parameter $SEK_v^{'}$ using the same method used by the RSU. Then, the RSU uses the hash function H with the $SEK_{RSU}^{'}$ and r to compute an agreement session key SEK_{v-RSU}, as in Eq. (13). The vehicle V uses the hash function H with the $SEK_v^{'}$ and r to compute an agreement session key SEK_{v-RSU}, as in Eq. (14). On the basis of the information on the vehicle V and the vehicle V_i that the RSU obtained during the communication, the RSU helps the vehicle V_i to generate an agreement session key SEK_{v-vi} between it and the vehicle V. Thus, the RSU should compute a session key parameter $\left(SEK_{v-vi}^{'}\right)$. Then, it computes the generated agreement session key SEK_{v-vi}, as in Eq. (15).

$$SEK_{v-RSU} \leftarrow H\left(SEK_{RSU}^{'}, r\right). \tag{13}$$

$$SEK_{v-RSU} \leftarrow H\left(SEK_{v}^{'}, r\right). \tag{14}$$

$$SEK_{v-vi} \leftarrow H\left(SEK_{v-vi}^{'}, r\right). \tag{15}$$

Step 5: The RSU needs to communicate with the vehicle V_i to confirm that the session key is successfully established. The RSU generates a hash value HF_3, as described in Eq. (16). Moreover, the RSU computes enc_2 by using the session key SEK_{vi-RSU}, as described in Eq. (17). Meanwhile, the RSU generates a hash value HF_4, as described in Eq. (18), to guarantee the integrity of the message between the RSU and the vehicle V_i. The RSU also computes enc_3 by using SEK_{v-RSU} to encrypt the message $\left(SEK_{v-vi}^{'}\right)$, as in Eq. (19).

$$HF_3 \leftarrow H\left(ID_{RSU}, ID_v, SEK_{v-RSU}, T_{RSU}\right). \tag{16}$$

$$enc_2 \leftarrow Senc_{SEK_{v-RSU}}\left(SEK_{v-vi}^{'}\right). \tag{17}$$

$$HF_4 \leftarrow H\left(ID_{RSU}, ID_{vi}, SEK_{vi-RSU}, T_{RSU}\right). \tag{18}$$

$$enc_3 \leftarrow Senc_{SEK_{vi-RSU}}\left(SEK_{v-vi}\right). \tag{19}$$

Afterward, the RSU prepares a data packet with the message $(ID_{RSU}, HF_3, enc_2, enc_3, HF_4)$, which is included in the content. Then, the RSU signs the data packet by using its private key PrK_{RSU} with S_4. The RSU sends the data packet with a name prefix /ndn/VNDN/RSU/ID_{RSU}/HF_3/enc_2/HF_4/enc_3/T_{RSU}/S_4 to the vehicle V_i. The data packet with the name /ndn/VNDN/RSU/ID_{RSU}/HF_3/enc_2/HF_4/enc_3 /T_{RSU}/S_4 is appended to the PIT.

Step 6: After the vehicle V_i receives the data packet from the RSU, it extracts HF_4 and enc_3 from the data packet. If $T_{vi} - T_{RSU} \leq \triangle T$, the vehicle V_i first computes $HF_4^{'}$. Then, it compares the $HF_4^{'}$ with the received HF_4. If they are correct, the vehicle V_i

decrypts enc_3 by using SEK_{vi-RSU} to obtain the session key SEK_{v-vi}. The vehicle V_i forwards the data packet /ndn/VNDN/RSU/ID_{RSU}/ID_{vi}/HF_3/enc_2/T_{RSU}/S_4 to the vehicle V.

Step 7: When the vehicle V receives the data packet from the vehicle V_i, it first extracts HF_3 and enc_2 from the data packet. If $T_v - T_{RSU} \leq \triangle T$, the vehicle V computes the hash function $HF_3^{'}$ based on the message in the received data packet. If $HF_3^{'}$ is equal to the received HF_3, the vehicle V also verifies the signature S_4 of the received data packet by using the public key of the RSU. If it is correct, the vehicle V believes that the session key SEK_{v-RSU} is shared successfully between V and RSU. Then, the vehicle V decrypts enc_2 to obtain the session key parameters SEK_{v-vi}. The vehicle V computes the agreement session key SEK_{v-vi} between the vehicle V and the vehicle V_i by using the same method by the RSU using a hash function with $SEK_{v-vi}^{'}$ and r.

5.5. Public Key Update Stage

In this stage, the user requests to update the public key data. The users can apply for a key update if the VP of the current public key is about to expire. In the registration stage, the VTA conducts a detailed review of the vehicle user. Thus, at this stage, the VTA does not need to review the user's identity information in detail. The processes of the update public key are described in Algorithm 2.

Algorithm 2 update_PK function

Input $ID, new\_PbKname, new\_PbK, new\_VP$
Output bool

1: **if** $msg.sender \neq VTA$ **then**
2: revert (); // Define an error and pass it while reverting a transaction
3: **else**
4: $users[ID] \leftarrow user(ID, new\_PbKname, new\_PbK, new\_VP)$;
5: emit updatePK $(ID, new\_PbKname, new\_PbK, new\_VP)$;
6: **end if**
return true;

The user sends a request message to the VTA for an updated public key. The request includes $(ID, PbKname, PbK, andVP)$. The VTA performs a simple verification on the user. After the verification is passed, the user generates a new key pair (new_PbK, new_PrK) in the registration stage. Then, the user of the vehicle V sends the new public key n_PbK to the VTA. Then, the VTA generates a new PbKname and new VP for the new public key.

Afterward, the VTA binds the tuple $(ID, new\_PbKname, new\_PbK, new\_VP)$ and encodes it as hexadecimal codes. Then, it is compressed in JSON format. A new transaction is generated by the VTA, and the generated hexadecimal code is embedded in its data field. Then, the VTA sends the transaction to the NDN-BCN for execution. The smart contract function update_PK is triggered for the update of the public key of the vehicle V. After the smart contract is successfully executed and deployed to the NDN-BCN, the transaction is stored in the NDN-BCN.

After the updated public key data are successfully uploaded to the chain, the VTA also needs to update the polynomial previously assigned to the vehicle; that is, VTA selects new $L(1 \leq L \leq m)$ polynomials from m polynomials and distributes a new authentication key value $K_v(K_v \in setK)$ and polynomial fragment F to the user of the vehicle V. Finally, the VTA securely sends secret materials and the relevant parameters of the new polynomial to the vehicle user. Thus far, the public key update process of the vehicle user has been completed.

5.6. Public Key Revocation Stage

In the management mechanism proposed in this study, the security of the public and private keys also means the security of the VNDN system. Disclosing or stealing the user's private key seriously impacts the entire system. Therefore, an effective detection mechanism and a revocation mechanism for invalid public keys are urgently needed.

Step 1: When the RSU receives the broadcast interest packet from the vehicle user, the RSU takes the malicious user's ID and sends it to the VTA if the interest packet has serious authenticity deviation. The VTA defines this user as a malicious user.

Algorithm 3 revoke_PK function

Input ID
Output bool

1: **if** $msg.sender \neq VTA$ **then**
2: revert (); // Define an error and pass it while reverting a transaction
3: **else**
4: $users[ID] \leftarrow user\,(0)$;
5: number=number-1;
6: emit revokeID (ID);
7: **end if**
return true;

Step 2: After the VTA determines the identity of the malicious user, the VTA sends the revoked transaction to the NDN-BCN and executes the revoke_PK function in the contract to remove the user's identity and the binding $(PbKname, PbK)$. The transaction record is removed after the smart contract is successfully executed and mined. On the NDN-BCN, the malicious user's public key is identified as invalid and returned to the VTA for execution results. Thus far, the public key revocation process of the vehicle user has been completed. Algorithm 3 describes the revocation function operation of the user's public key.

6. Security Analysis

6.1. Informal Security Analysis

Our proposed scheme in this paper uses blockchain to manage the user's identity and public key. The decentralized key management solution can effectively resolve the problems

of the traditional solution based on the centralized PKI. And the proposed scheme satisfies the following security requirements:

1) **Entity Authentication:** In our proposed scheme, mutual authentication between different communication entities by using our authentication protocol is required to verify the legitimacy of these entities such as the vehicles or RSUs. So, our scheme can provide confidence that an entity can't perform the active attacks such as either a masquerade or an unauthorized replay.
2) **Data Confidentiality:** In our VNDN system, a sensitive data transmitted on the network is required to be encrypted by using a symmetric encryption to prevent a passive attack mentioned in section 4.2. So, a session key needs to be generated by using our key agreement protocol before two authenticated entities communicate. We denote the session key between a vehicle V and a RSU by the symbol SEK_{v-RSU} and denotes a session key between the vehicle V and its neighbor vehicles V_i by the symbolSEK_{v-vi}. These two session keys are created based on bivariate polynomial parameters, and are stored in the HSM. Of course, any adversary cannot obtain them to compromise the data confidentiality.
3) **Message Authentication:** The EC-Schnorr signature algorithm is used to ensure the integrity authenticity of a message transmitted between different entities in VNDN system. The private keys (PrK_v, PrK_{RSU}) used in the EC-Schnorr algorithm are stored in HSM. Thus, an adversary cannot obtain these cryptographic materials to forge a valid message signature. In addition to, our proposed scheme also can prevent the active attack mentioned in section 4.2 by flooding a large number of interest packets.

6.2. Security Analysis in the UC Framework

We follow the approach of the UC framework to analyze the security attributes of our protocols in our VNDN system. Our main protocols include an authentication protocol and a key agreement protocol. The notion of this framework is based on simulation. That is to say, for any real-world protocol π and an adversary A, if there exist an ideal protocol ϕ, ideal functionality F and a simulator S can simulate running of the protocol π, then we say that the protocol π . In other words, for any environment ε, if the probability that the ε can tell the difference between the execution of the protocol π with the adversary A and the execution of the ideal protocol ϕ with the adversary S is neglible at most, the protocol π is a UC-secure protocol. Therefore, the UC security ensures a composable protocol of any UC-secure protocols is also secure.

The ideal functionality F is considered the most important part of the UC framework. It is a trusted party for achieving the security requirements of cryptographic protocols. It guarantees the secrecy, authenticity, or delivery of the communicated information. Most of the ideal functionalities have already been formulated, such as the signature functionality F_{SIG}, the public key encryption functionality F_{PKE}, the secure message transmission functionality F_{SMT}, the key exchange functionality F_{KE}, and the message authentication functionality F_{AUTH}.The interested reader can refer to the related literatures. These ideal functionalities can be used as the standard ideal functionality in our security analysis.

A. Security Analysis of the Authentication Stage

The real authentication protocol π_{AUTH} in the real world is described in Figure 4. The difference with the ideal authentication protocol π_{AUTH} described in Figure 6 is that the ideal protocol should be executed in (F_{SIGN}, F_{REG})-hybrid model, where the two functions (F_{SIGN}, F_{REG}) in F_{AUTH} are selected to realize the composition theorem. . In particular, the authentication protocol π_{AUTH} uses the signature ideal functionality F_{SIGN} to achieve the signature process. The registration ideal functionality F_{REG} is used for registering the key material parameters such as K_v and K_{RSU} and so on. The detail of the ideal authentication protocol ϕ_{AUTH} is described in Figure 6.

Ideal Protocol φ_{AUTH} for $\mathcal{F}_{AUTH}$

Let $SID = (ID_v, ID_{RSU}, sid)$ where sid is the session identity of ID_v and ID_{RSU}.

1) The vehicle V is activated with input (Send, SID, Interest packet) to the ideal functionality $\mathcal{F}_{AUTH}$.
2) Upon receiving (Sent, SID, new Interest packet), the RSU sends the response (Send, SID, Data packet) to the functionality $\mathcal{F}_{AUTH}$, which may forward through VNDN nodes to the vehicle V.

Fig. 6. Ideal Authentication Protocol ϕ_{AUTH}

Theorem 1. *The UC framework can be used to prove the security attributes of the authentication protocol π_{AUTH} in the (F_{CERT}, F_{REG})-hybrid model, i.e., the real protocol π_{AUTH} can securely access the ideal protocol ϕ_{AUTH} for the functionality F_{AUTH}.*

Proof. Let π_{AUTH} be the real protocol in the real world. We say that the protocol π_{AUTH} securely realizes an ideal protocol ϕ_{AUTH} for the ideal functionality F_{AUTH} if an ideal process adversary S exists for any real-world adversary A. In particular, no environment ε can tell the difference between a real process and an ideal process with a nonnegligible probability.

An adversary S in the (F_{SIGN}, F_{REG})-hybrid model is responsible for delivering the message from the copies of F_{CERT} and F_{REG}. This message is forwarded to a real adversary A. The adversary S can be a simulator for the adversary A. Thus, adversary S can activate a real adversary A. Moreover, the simulator S forwards the instruction from environment ε to adversary A and copies the output of the adversary A to the environment ε. The detailed of adversary S is as follows:

1) When the vehicle V is activated with input (**Send**, SID, Interest packet), the adversary S simulates the forwarding process of the interest packet with the name prefix /ndn/VNDN/V/ID_v/$PbKname_v$/K_v/M_r/T_v/HF_1/S_1 for A from the vehicle V to the RSU. First, the message $(ID_v, PbKname_v, K_v, M_r, T_v, HF_1, S_1)$ is extracted from the interest packet. Upon receiving (**Send**, SID, Interest packet), the adversary S first wants to retrieve the key value parameters K_v and K_{RSU}. It sends (**Retrieve**, sid, V)

and (**Retrieve**, sid, RSU). Then, it obtains the responses (**Retrieve**, sid, V, K_v) and (**Retrieve**, sid, RSU, K_{RSU}) by the interaction with the F_{REG}. After the adversary S obtains the key values parameters, it computes the secret value sK_v according to the bivariate polynomial $sK_v \leftarrow f(K_v, K_{RSU})$. Once the secret values are obtained, the adversary S computes the hash value $HF_1 \leftarrow H(ID_v, PbKname_v, sK_v, M_r, T_v)$. Then, the adversary S interacts with F_{SIGN} to obtain S_1, when it receives (**Sign**, sid, V, $(ID_v, PbKname_v, sK_v, M_r, T_v)$) from F_{SIGN}. The adversary S forwards the message to the adversary A. When the adversary A outputs (**Signature**,sid, V,$(ID_v$, $PbKname_v, sK_v, M_r, T_v), S_1$), the adversary S assigns S_1 as the signature of the vehicle V. Then, the message is sent to the functionality F_{SIGN}.

2) When the adversary A delivers the interest packet /ndn/VNDN/V/ID_v/$PbKname_v$/K_v /M_r/T_v/HF_1/S_1 from the vehicle V, it is forwarded through the VNDN nodes to RSU. The adversary S simulates for the adversary A the receiving interaction with F_{SIGN}. When the adversary S receives the message (**Verify**,sid, V,$(ID_v, PbKname_v, sK_v$ $, M_r, T_v), S_1$) from F_{SIGN}, this message is forwarded to the adversary A. Then, the adversary S obtains the response (**Verify**,sid, V,$(ID_v, PbKname_v, sK_v, M_r, T_v), \sigma_1$) from the adversary A. Afterward, S sends the response message obtained from the adversary A to F_{SIGN}. If F_{SIGN} outputs (**Verified**,sid, V,$(ID_v, PbKname_v, sK_v, M_r, T_v)$ $, \sigma_1, f = 1$) to RSU, the adversary S sends (**Sent**, SID, Interest packet) to the authentication functionality F_{AUTH}. Otherwise, it does not do anything.
3) When the RSU is activated with the input (**Send**, SID, Data packet), the adversary S simulates for A the data packet /ndn/VNDN/RSU/node/ID_RSU/$PbKname_{RSU}$/ K_{RSU}/M_s/T_{RSU}/S_2/S_3 from RSU that is forwarded to the vehicle V. The adversary S also has sK_{RSU}. Then, it obtains S_2 and S_3 by the interaction with F_{SIGN}. When the adversary S receives the message (**Sign**, sid, RSU,$(ID_RSU, PbKname_{RSU}$, $sK_{RSU}, M_s, T_{RSU})$) from F_{SIGN}, the adversary S forwards the message to A. When the adversary A outputs (**Signature**,sid, RSU,$(ID_{RSU}, PbKname_{RSU}, sK_{RSU}, M_s$ $, T_{RSU}), S_2$), the adversary S assigns S_2 as the signature of the RSU and then sends the message to F_{SIGN}. Afterward, S also receives the message (**Sign**, sid, RSU, $(ID_{RSU}, PbKname_{RSU}, sK_{RSU}, M_s, T_{RSU}, S_2)$) from F_{CERT}. S forwards the message to A. When the adversary A outputs (**Signature**,sid, RSU,$(ID_{RSU}, PbKname_{RSU}$ $, sK_{RSU}, M_s, T_{RSU}, S_2), S_3$), the adversary S assigns S_3 as the signature of the data packet of RSU and then sends the message to F_{SIGN}.
4) When the adversary A delivers the data packet /ndn/VNDN/RSU/node/ID_{RSU}/ $PbKname_{RSU}$/K_{RSU}/M_s/T_{RSU}/S_2/S_3, RSU forwards it through the VNDN node to V. S also simulates for the adversary A the receiving interaction with F_{SIGN}. When the adversary S receives the messages (**Verify**,sid,RSU,$(ID_{RSU}, PbKname_{RSU}$ $, sK_{RSU}, M_s, T_{RSU}), S_2$) and (**Verify**,sid,RSU,$(ID_{RSU}, PbKname_{RSU}, sK_{RSU}$, $M_s, T_{RSU}, S_2), S_3$) from F_{SIGN}, it forwards these messages to the adversary A. Then, the adversary S obtains the responses (**Verify**,sid, RSU,$(ID_{RSU}, PbKname_{RSU}$ $, sK_{RSU}, M_s, T_{RSU}), \sigma_2$) and (**Verify**,sid, RSU,$(ID_{RSU}, PbKname_{RSU}, sK_{RSU}$, $M_s, T_{RSU}, S_2), \sigma_3$) from the adversary A. Afterward, S sends the response messages obtained from the adversary A to F_{SIGN}, if F_{SIGN} outputs (**Verified**,sid, RSU,$(ID_{RSU}, PbKname_{RSU}, sK_{RSU}, M_s, T_{RSU}), \sigma_2, f = 1$) and (**Verified**,sid, RSU,$(ID_{RSU}, PbKname_{RSU}, sK_{RSU}, M_s, T_{RSU}, S_2), \sigma_3, f = 1$) to RSU. Finally, the adversary S sends (**Sent**, SID, Data packet) to F_{AUTH}. Otherwise, it does not do anything.

According to the symmetry property of the polynomial function and the logic of F_{REG} that used to register the key values parameters, the adversary A cannot change K_v and K_{RSU}, which are registered in F_{REG}. Thus, the adversary A cannot modify the secret values sK_v and sK_{RSU} that are calculated based on the bivariate polynomial function. However, based on the resistant collusion of the hash function HF_1 and the signatures obtained from F_{SIGN}, the adversary A can attack the real protocol π_{AUTH} and forge the hash message and the signature message. Adversary A can create a forge interest packet as /ndn/VNDN/V/$ID_{v'}$/$PbKname_{v'}$/K_v/$M_r^{'}$/$T_{v'}$/HF_1/S_1 when it obtains the interest packet /ndn/VNDN/V/ID_v/$PbKname_v$/K_v/M_r/T_v/HF_1/S_1. Moreover, the adversary A can create a forge data packet /ndn/VNDN/RSU/node/$ID_{RSU'}$/$PbKname_{RSU'}$/$K_{RSU'}$/ $M_s^{'}$/$T_{RSU'}$/S_2/S_3. The verification process indicate that the authentication of the interest packet and data packet of the real protocol π_{AUTH} is different from that of the interest packet and data packet of the ideal protocol ϕ_{AUTH} for functionality F_{AUTH}.

Collusion in the hash function is hard to find. The probability of obtaining the collusion and forging the signature is negligible at most. However, the resistant collusion in the hash function and the logic of F_{SIGN} is unforgeability against chosen-message attacks. Therefore, we can say that environment ε can distinguish between the real protocol π_{AUTH} in the (F_{SIGN}, F_{REG})-hybrid model and the ideal protocol ϕ_{AUTH} for functionality F_{AUTH} with a negligible probability, at most.

B. Security Analysis of the Key Agreement Stage

The real key agreement protocol π_{KA} in the real world is described in Figure 5. The main difference with the ideal key agreement protocol ϕ_{KA} in the ideal process is that the protocol π_{KA} should be in the $(F_{PKE}, F_{SIGN}, F_{REG})$-hybrid model. In particular, the key agreement protocol π_{KA} uses the ideal functionality F_{PKE} for realizing the encryption algorithm. The ideal functionalities F_{SIGN} and F_{REG} are used in the authentication protocol. The detail of the ideal protocol ϕ_{KAG} is described in Figure 7.

Theorem 2. *The UC framework can be used to prove security attributes of the key agreement protocol π_{KA} in the $(F_{PKE}, F_{SIGN}, F_{REG})$-hybrid model, i.e., the real protocol π_{KA} can securely access the ideal protocol ϕ_{KA} for the functionality F_{KE}.*

Proof. Let π_{KA} be the real protocol in the real world. We say that the protocol π_{KA} securely realizes an ideal protocol ϕ_{KA} for the ideal functionality F_{KE} if an ideal-process adversary S exists for any real-world adversary A. In particular, no environment ε can tell the difference between the real process and ideal process with a nonnegligible probability.

The details of the adversary S is as follows:

1) When the vehicle V is activated with input (**CreateSEK**, SID_1), the adversary S first retrieves the key values parameters K_v and K_{RSU}. It sends (**Retrieve**, sid_1, V) and (**Retrieve**, sid_1, RSU). Then, it obtains the responses (**Retrieve**, sid_1, V, K_v) and (**Retrieve**, sid_1, RSU, K_{RSU}) through the interaction with the F_{REG}. After the adversary S obtains the key values parameters, it computes the secret values sK_v and sK_{RSU} based on the polynomial function f defined in the system initialization stage. Then, the simulator S selects random polynomials f_i and K_i and a random number r. It sends (**Encrypt**, sid_1, RSU, $(ID_{RSU}, ID_v, f_i, K_i, r)$) to the functionality F_{PKE}. When the vehicle V receives the response (**Ciphertext**, sid_1,

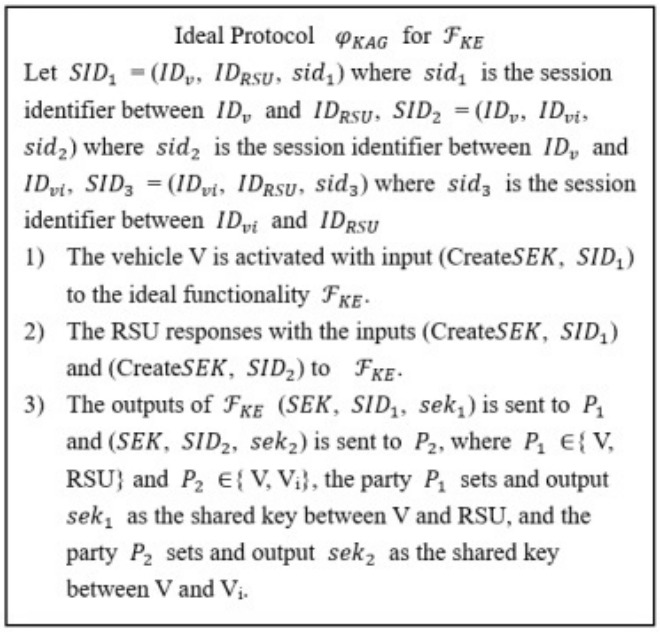

Fig. 7. Ideal Key Agreement Protocol ϕ_{KAG}

RSU, enc_1) from the functionality F_{PKE}, the adversary S computes the hash value $HF_2 \leftarrow H(ID_v, enc_1, sK_v, T_v)$ and simulates for the adversary A the forwarding process of the interest packet with the name prefix /ndn/VNDN/V/ID_v/enc_1/HF_2/T_v from the vehicle V through the vehicle V_i to the RSU.

2) When the adversary A delivers the interest packet /ndn/VNDN/V/ID_v/enc_1/HF_2/T_v from the vehicle V to the RSU, the simulator S simulates for the adversary A the receiving interaction with F_{PKE}. When the adversary S receives the message (**Decrypt**, sid_1, RSU, enc_1) from the simulated process of RSU, it forwards this message to the functionality F_{PKE}. Then, the adversary S obtains the response (**Plaintext**, sid_1, RSU, $(ID_{RSU}, ID_v, f_i, K_i, r)$) from the F_{PKE} to RSU. Afterward, the adversary S computes $SEK'_{(v-RSU)} \leftarrow f(K_{RSU}, K_v)$. Finally, the adversary S assigns the session key $SEK_{(v-RSU)}$ as $SEK_{v-RSU} \leftarrow H\left(SEK'_v, r\right)$ in the simulation of the real protocol π_{KA}.
3) When the RSU is activated with the inputs (**CreateSEK**, SID_1) and (**CreateSEK**, SID_2), the adversary S first retrieves $SEK_{(vi-RSU)}$. It sends (**Retrieve**, sid_3, V_i, RSU). Then, it obtains the response (**Retrieve**, sid_3, V_i, RSU, SEK_{vi-RSU}) through the interaction with the F_{KE} function. The adversary S computes HF_3, HF_4, and SEK'_{v-vi}. Then, S assigns the session key as $SEK_{v-vi} \leftarrow H\left(SEK'_{v-vi}, r\right)$ in the simulated of the real protocol π_{KA}. The adversary S sends (**Encrypt**, sid_1, V, SEK'_{v-vi}), and (**Encrypt**, sid_3, V_i, (SEK_{v-vi})) to the functionality F_{PKE}. When the RSU receives the response (**Ciphertext**, sid_1, V, enc_2) and (**Ciphertext**, sid_3, V_i, enc_3) from the functionality F_{PKE}, S obtains S_4 by interacting with F_{SIGN}. When

S receives the message (**Sign**, sid_1, V, $(ID_{RSU}, HF_3, enc_2, T_{RSU})$) from F_{SIGN}, S simulates for adversary A the data packet with the name prefix /ndn/VNDN/RSU/ID_{RSU} /HF_3/enc_2/HF_4/enc_3/T_{RSU}/S_4 from RSU to the vehicle V_i. The forwarding data packet /ndn/VNDN/RSU/ID_{RSU}/HF_3/enc_2/T_{RSU}/S_4 from the vehicle V_i to the vehicle V is also simulated.

4) When the adversary A delivers the data packet /ndn/VNDN/RSU/ID_{RSU}/HF_3/enc_2 /HF_4/enc_3/T_{RSU}/S_4 from RSU to the vehicle V_i and the forwarding data packet /ndn/VNDN/RSU/ID_{RSU}/HF_3/enc_2/T_{RSU}/S_4 from the vehicle V_i to the vehicle V, the adversary S simulates for the adversary A the receiving interaction with the functionality F_{PKE} and functionality F_{SIGN}. When the adversary S receives the messages (**Decrypt**, sid_1, V, enc_2) from the simulated process of the vehicle V and (**Decrypt**, sid_3, V_i, enc_3) from the simulated process of the vehicle V_i, it forwards these messages to the functionality F_{PKE}. Then, the adversary S obtains the responses (**Plaintext**, sid_1, V, $(SEK^{'}_{v-vi})$) from F_{PKE} to the vehicle V and (**Plaintext**, sid_3, V_i, (SEK_{v-vi})) from the functionality F_{PKE} to the vehicle V_i. When receiving (**Verify**, sid_1, V, $(ID_{RSU}, HF_3, enc_2, T_{RSU})$) from F_{SIGN}, the adversary S forwards this message to the adversary A and obtains the response (**Verify**, sid_1, V_i, $(ID_{RSU}, HF_3, enc_2, T_{RSU})$,$\sigma_4$) from the adversary A. Moreover, the responses are forwarded to the functionality F_{SIGN}.

5) For the outputs $(SEK, sid_1, P_1, SEK_{v-RSU})$ and $(SEK, sid_2, P_2, SEK_{v-vi})$ of the simulation of the real protocol π_{KA}, the adversary S forwards them to the functionality F_{KE}. If the vehicle V, vehicle V_i, and RSU are not corrupted, the vehicle V_i and RSU output (SEK, sid_1, P_1, sek_1). Moreover, the vehicle V and vehicle V_i output (SEK, sid_2, P_2, sek_2) according to the logic of the functionality F_{KE} in the ideal protocol ϕ_{KAG}.

According to the symmetry property of the polynomial function described in Section 3 and the logic of F_{REG} used to register the key values parameters, the adversary A cannot change K_v, K_{RSU}, and K_{vi}, which are registered in F_{REG}, and the logic of F_{PKE}. Thus, the adversary A cannot modify the secret values sK_v, sK_{vi}, and sK_{RSU}, which are computed based on bivariate polynomial functions. Moreover, the adversary A cannot modify the session key SEK_{vi-RSU} that is created in F_{KE}. However, the resistant collusion of the hash functions HF_2, HF_3, and HF_4 and the signatures that are obtained from F_{SIGN} indicate that the adversary A can attack the real protocol π_{KA} and forge the hash message and the signature message. When the adversary A obtains the interest packet /ndn/VNDN/V/ID_v/enc_1/HF_2/T_v, it can create a forge interest packet as /ndn/VNDN/V/SID_1/$ID_{v'}$ /enc_1/HF_2/$T_{v'}$. Moreover, the adversary A can create a forge data packet /ndn/VNDN/RSU/SID_1/SID_2/SID_3/$ID_{RSU'}$/HF_3/enc_2/HF_4/enc_3/$T_{RSU'}$/S_4.

In addition, the difference between the session keys SEK_{v-RSU} and SEK_{v-vi} in the real protocol π_{KA} and the session keys and in the ideal protocol ϕ_{KAG} that the environment ε can distinguish is a negligible. That because if the environment ε can tell the difference between the session keys in the real protocol and ideal protocol with just a non-negligible probability, the ciphertexts enc_1 and enc_2 can be distinguish by constructing an adversary A based on the environment ε , where this will conflict with the logic of the functionality F_{PKE}, as we know the UC-secure public key encryption that implemented the ideal functionality F_{PKE} is a CCA-secure encryption.

Collusion in the hash function is difficult to find, and the probability of obtaining the collusion and forging the signature is negligible at most. However, the resistant collusion in the hash function and the logic of F_{SIGN} and F_{PKE} are unforgeability against the chosen-message attacks. Therefore, we can say that environment ε can distinguish between the real protocol π_{KA} in the $(F_{SIGN}, F_{PKE}, F_{REG})$-hybrid model and the ideal protocol ϕ_{KAG} for functionality F_{KE} with a negligible probability at most.

7. Performance Analysis

7.1. Implementation of our scheme

The implementation of our scheme depends on the ndn-geth project to implement the blockchain nodes with the NDN protocols. A private blockchain is generated in virtual machine of the Ubuntu 20.04.4 LTS with 4 GB RAM on a laptop. To implement the NDN protocol on the laptop, first, the NFD and ndn-cxx are installed to obtain their requirements libraries. Then, the ndn-geth is cloned from GitHub. In this study, the blockchain nodes are maintained by the RSUs In the VNDN system. The private blockchain environment contains two nodes node1 and node2. After each node is generated, it is necessary to ensure that the NFD is running. the prefix name /ndn/VNDN/RSU1/node1 is assigned to the first node1, and the prefix name /ndn/VNDN/RSU2/node2 is assigned to the another node node2. These two nodes are configured with the same genesis file, Figure 8 shows the JISON configuration file of the prefix name and the block IDs of the NDN's host communication interface. These two nodes can communication by using the NDN protocol, and their information will be added the router list of the NDN. The peer-to-peer host configuration of NDN-blockchain is shown in Figure 9 and the router list of the NDN host is shown in Figure 10.

Fig. 8. JISON configuration file

7.2. Gas Consumption

In the proposed key management framework, the smart contracts are used to register, update and revoke the identity and public key data of vehicles and RSUs. In order to analyze the economic cost of this scheme, this paper uses the Ethereum test network (Rinkeby)

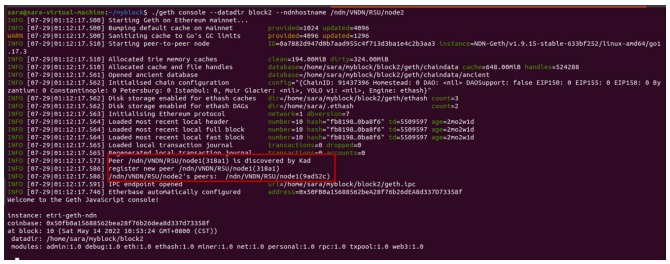

Fig. 9. Peer-to-peer host configuration NDN-blockchain

Fig. 10. Router list of the NDN host

to compile and deploy smart contracts. Rinkeby is an official test network provided by Ethereum. This test network can achieve the same functions as the real Ethereum network. The experiment was performed on July 2022. The gas price was set to 1 Gwei, where 1 Gwei = 0.000000001 Ether and 1 Ether = 1,065 USD. The transactions of the smart contracts are found at https://rinkeby.etherscan.io/ by using the account address of the VTA (0xb42819AEdB1b9928199F9E42202fF492c37072D8). The gas cost analysis of the smart contracts is shown in Table 2.

Table 2. The Cost Performance of The Smart Contracts

Functions	Gas Consumption	Ether's Cost	Cost (USD)
Contract Creation	1323932	0.001323932	1.4099
register_PK	166483	0.000166483	0.1773
update_PK	59910	0.000059910	0.0638
revoke_PK	35088	0.000035088	0.0374

As shown in Table 2, the contract creation is executed just once, and the cost is approximately $1.4099. The cost of the register_PK process is $0.1773, and the cost of the update_PK process is $0.0638. The cost of the revoke_PK process is approximately $0.0374. When the number of users increases, the cost of the computation increases in the process of registration, update, and revocation. The cost of the computation process with various users is shown in Figure 11.

We compare our scheme with the other existing key management schemes in the NDN and VNDN, as shown in Table 3. Some existing key management schemes in the NDN [5,6,7] are centralized schemes that depend on the CA. Other schemes [17,18,19] make use of the benefits provided by the blockchain.

Table 3 shows that the scheme [19] uses the blockchain to update and revoke a public key. However, it creates some problems in the key management because of the tamper-proof feature of the blockchain. Therefore, we use the smart contract to avoid these prob-

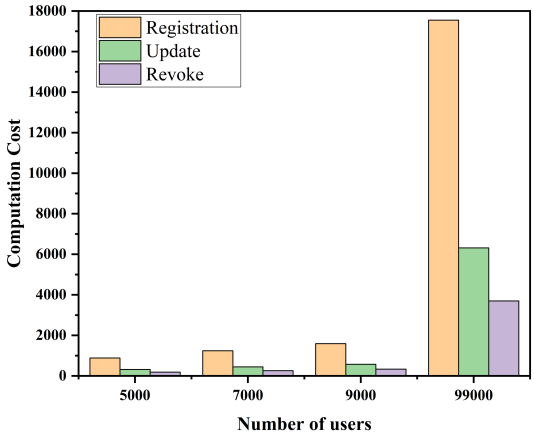

Fig. 11. The computation cost process of the variety of the users

lems by automatically managing the user's public key and identity and presenting the efficient key management in VNDN in an automatic way.

Table 3. Compression with the existing schemes

Existing sSchemes	Structure	Based On	Smart Contract	Key Update	Key Revocation	Authentication	Key Agreement
Chaoyi [4]	Centralized	Trusted Anchor	-	No	No	No	No
A. Albarqi [5]	Centralized	PKI	-	No	Yes	No	No
S. Santesson [6]	Centralized	PKI	-	No	Yes	No	No
Yingdi [7]	Decentralized	Web-of-Trust (WoT)	-	No	Yes	Yes	No
Hao [17]	Decentralized	Blockchain	No	No	No	Yes	No
Junjun [18]	Decentralized	Blockchain	No	No	Yes	Yes	No
Kan [19]	Decentralized	Blockchain	No	Yes	Yes	No	No
Our scheme	Decentralized	NDN with blockchain	Yes	Yes	Yes	Yes	Yes

7.3. Computation Cost

The computation cost of the authentication and key agreement stage in our scheme is analyzed in this section. The operation time of basic cryptographic algorithms used in our scheme is firstly tested in our experimental environment and is shown in Table 4. The notations used in the computation complexity measurement are illustrated in Table 5. We make a comparison between our proposed scheme and the scheme in [24]. The comparison results of the average computation complexity of two different stages in these solutions are shown in Table 6.

In our proposed scheme, the main costs of the authentication stage include the signing time and verification time used in the EC-Schnorr signature algorithm. Thus, the time cost of the authentication stage is $3(T_{sign} + T_{ver})$. In the key agreement stage, two different session keys are generated based on the bivariate polynomial function. The AES-based

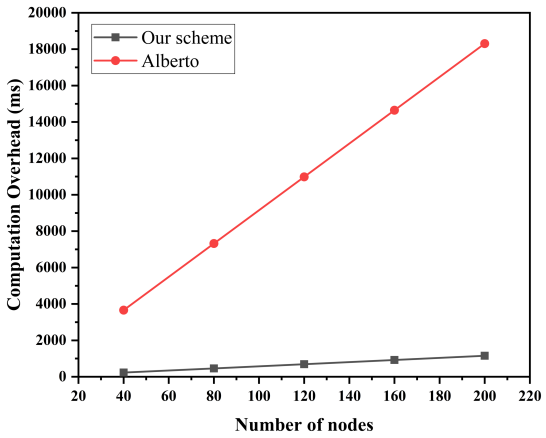

Fig. 12. The computation cost of the authentication stage over the number of nodes

Table 4. The execution times

Operations	Average Time (ms)
EC-Schnorr(signing)	0.144
EC-Schnorr (verification)	1.777
Hash function (SHA 256)	0.008
AES-128 (encryption)	0.019
AES-128 (Decryption)	0.418

Table 5. The Notation Explanation

Notations	Description
T_{sign} and T_{ver}	The average time of the signature and verification of our scheme's signature solution
T_{ssign} and T_{sver}	The average time of the signature and verification of other scheme's signature solution
T_{Senc} and T_{SDec}	The average time of the encryption and decryption of the symmetric cryptography
T_{Biv} and T_K	The average time of the bivariate polynomial and key derivation

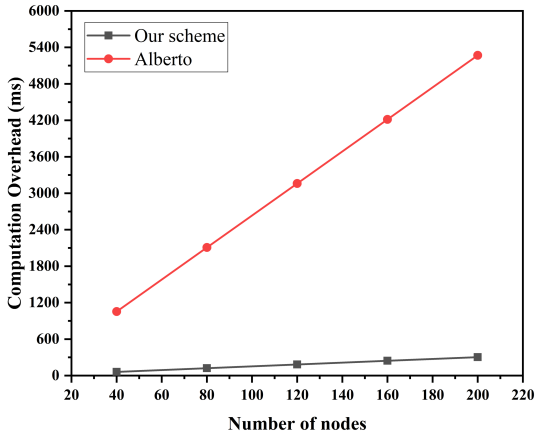

Fig. 13. The computation cost of the key agreement stage over the number of nodes

encryption is used to transmit securely the relevant parameters of the polynomial. Thus, the computation cost of the key agreement contains the cost of the AES algorithm and bivariate polynomial. The total time of the key agreement stage is $T_{Senc} + T_{SDec} + T_{Biv}$. The computation complexity of the authentication stage and key agreement stage over the different number of nodes is respectively shown in Figure 12 and Figure 13. The computation overhead against the DoS attack in the authentication stage is shown in Figure 14. The result in Figure 14 shows our scheme can efficiently mitigate DoS attacks aimed at the cache store by the cause of interest flooding.

Table 6. Computation complexity

Schemes	Authentication Stage Complexity		Key Agreement Stage	Complexity
Our Scheme	$3T_{sign} + 3T_{ver}$	5.765	$3T_{Senc} + 3T_{SDec} + 2T_{Biv}$	1.523
Alberto [24]	$2T_{ssign} + 2T_{sver}$	91.55	$4T_{Senc} + 4T_{SDec} + 2T_K$	26.35

7.4. Communication Cost

The communication cost of this study is evaluated during the authentication stage and key agreement stage in terms of the amount of interest and data packets shared in the network and the number of bytes sent and received by vehicle nodes and RSUs nodes. This thesis depends on IEEE 802.15.4 standard [31], which offers the fundamental lower network layers of a type of wireless personal area network (WPAN) which focuses on low-cost, low-speed ubiquitous communication between devices. The maximum size of the IEEE 802.15.4 frame is 127 bytes, the frame contains three basic fields such that header, payload, and footer. This study considers the 1 +0 encoding which is proposed for

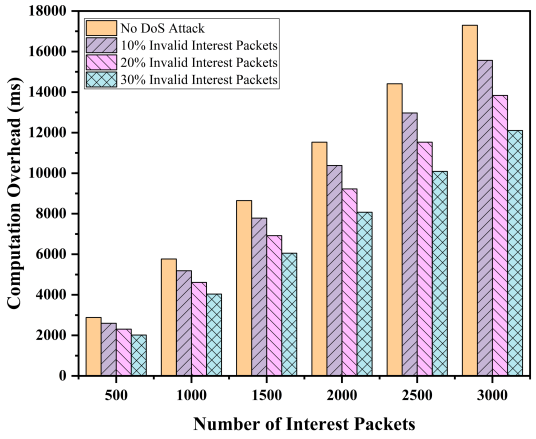

Fig. 14. The computation cost of the authentication protocol over DoS attacks

interest and data packets for NDN networks [32]. The size of the interest and data packets in each filed of the IEEE 802.15.4 frame is shown in Table 7.

Table 7. The size of NDN Interest (I)/Data packets(D)

Field	Field size	I	D
802.15.4 PHY header	6B	✓	✓
802.15.4 MAC header	23B	✓	✓
802.15.4 SEC header	5B	✓	✓
Packet type TL	1B	✓	✓
Name TL	1B	✓	✓
Name component TLVs	SN	✓	✓
Content TLV	1B (TL) +SC		✓
Signature info TL	1B		✓
Signature type TLV	1B (TL) + 1B (V)		✓
Signature value TLV	1B (TL) + 16B (V)		✓
802.15.4 Signature footer	16B	✓	✓
802.15.4 CRC footer	2B	✓	✓

As shown in Table 7, the number of bytes in the header and footer fields are 52 bytes. In the authentication stage, the vehicles who are on the same communication range help in the authentication process between a new vehicle and RSU in the VNDN. The communication between the new vehicle and other vehicles in the network is untrusted, therefore, the number bytes of the header and footer included in the 802.15.4 frame which is exchanged between the new vehicle and other vehicles in the network are 36 bytes because the signature field of the footer will not be included in the frame. On other hand,

the communication between the vehicles on the network who are already on the network and RSU is trusted, therefore, the number bytes of the header and footer included in the 802.15.4 frame which is exchanged between the vehicles and RSU are 52 bytes. According to the Table 7, the cost communication is computed, where SN is the total size of the name values, SC is the total size of the data packet content. In order to compute the cost communication, this study assumes values for the variables which are used in this study, where (ID, PbKname, M, T) are 4 bytes, the hash, signature and encryption are 16 bytes, the prefix name is 1 byte.

Table 8. Number of bytes for interest (I) and data (D) packets for authentication and key agreement stages

Stage	Communication path	Number of bytes (Interest/ Data)
Authentication stage	Vehicle V/ Vehicles Vi	91B (I)/ 92B(D)
	Vehicles Vi/ RSU	107B(I)/ 108B(D)
Key agreement stage	Vehicle V/ Vehicles Vi	79B(I)/ 104B(D)
	Vehicles Vi/ RSU	103B(I)/ 148B(D)

Table 9. The comparison of communication cost

Scheme	Number of bytes transmitted
This study	835 bytes
Alberto [24]	867 bytes

According to the above assumption, the number of bytes for interest (I) and data (D) packets which are transmitted in both authentication and key agreement stages are shown in Table 8. The comparison of cost communication of this study with the Alberto [24] scheme is shown in Table 5.8. As shown in Table 9, the number of bytes that are shared in this system are less than in scheme [24].

8. Conclusion

In this paper, a decentralized key management solution based on blockchain for the VNDN is firstly proposed. Smart contracts are used to manage the identity and public key information of entities in the VNDN. Then, a lightweight mutual authentication scheme between a vehicle V and RSU is proposed. In the key agreement stage, the session keys based on the bivariate polynomial between different entities are generated. Our scheme can prevent the passive attack by adopting the encryption scheme. Our scheme can also prevent the active attack, for example the DoS attack incurred by the interest flooding. In addition, we also explore the security analysis methods in the UC framework. The security requirements of the authentication stage and key agreement stage of the proposed scheme are analyzed in the UC framework. We also implement our scheme in the NDN-

blockchain platform and we experimentally analyze performance of our scheme. We will further improve our scheme in the future work.

Acknowledgments. This work is supported by NSFC No. 61461027; Gansu province science and technology plan project under grant No. 20JR5RA467; Innovation Promotion Education Fund of Ministry of Education No. $2018A05003$; Graduate Fine-designed Course of Lanzhou University of Technology.

References

1. Baofeng Ji, Xueru Zhang, Shahid Mumtaz, Congzheng Han, Chunguo Li, Hong Wen, and Dan Wang. Survey on the internet of vehicles: Network architectures and applications. *IEEE Communications Standards Magazine*, 4(1):34–41, 2020.
2. Alex Afanasyev, Jeff Burke, Tamer Refaei, Lan Wang, Beichuan Zhang, and Lixia Zhang. A brief introduction to named data networking. In *MILCOM 2018-2018 IEEE Military Communications Conference (MILCOM)*, pages 1–6. IEEE, 2018.
3. Chaker Abdelaziz Kerrche, Farhan Ahmad, Mohamed Elhoseny, Asma Adnane, Zeeshan Ahmad, and Boubakr Nour. Internet of vehicles over named data networking: Current status and future challenges. *Emerging Technologies for Connected Internet of Vehicles and Intelligent Transportation System Networks: Emerging Technologies for Connected and Smart Vehicles*, pages 83–99, 2020.
4. Chaoyi Bian, Zhenkai Zhu, Alexander Afanasyev, Ersin Uzun, and Lixia Zhang. Deploying key management on ndn testbed. *UCLA, Peking University and PARC, Tech. Rep*, 2013.
5. Aysha Albarqi, Ethar Alzaid, Fatimah Al Ghamdi, Somaya Asiri, Jayaprakash Kar, et al. Public key infrastructure: A survey. *Journal of Information Security*, 6(01):31, 2014.
6. Stefan Santesson, Michael Myers, Rich Ankney, Ambarish Malpani, Slava Galperin, and Dr. Carlisle Adams. X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP. RFC 6960, June 2013.
7. Yingdi Yu, Alexander Afanasyev, Zhenkai Zhu, and Lixia Zhang. An endorsement-based key management system for decentralized ndn chat application. *University of California, Los Angeles, Tech. Rep. NDN-0023*, 2014.
8. Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *Decentralized business review*, page 21260, 2008.
9. Moutaz Alazab, Salah Alhyari, Albara Awajan, and Ayman Bahjat Abdallah. Blockchain technology in supply chain management: an empirical study of the factors affecting user adoption/acceptance. *Cluster Computing*, 24:83–101, 2021.
10. Deepa Pavithran, Khaled Shaalan, Jamal N Al-Karaki, and Amjad Gawanmeh. Towards building a blockchain framework for iot. *Cluster Computing*, 23(3):2089–2103, 2020.
11. Hui Li, Lishuang Pei, Dan Liao, Xiong Wang, Du Xu, and Jian Sun. Bddt: use blockchain to facilitate iot data transactions. *Cluster Computing*, 24:459–473, 2021.
12. Tejasvi Alladi, Vinay Chamola, Nishad Sahu, Vishnu Venkatesh, Adit Goyal, and Mohsen Guizani. A comprehensive survey on the applications of blockchain for securing vehicular networks. *IEEE Communications Surveys & Tutorials*, 2022.
13. Ahmed Elkhalil, Jiashu Zhang, and Rashad Elhabob. An efficient heterogeneous blockchain-based online/offline signcryption systems for internet of vehicles. *Cluster Computing*, pages 1–18, 2021.
14. Khizra Asaf, Rana Asif Rehman, and Byung-Seo Kim. Blockchain technology in named data networks: A detailed survey. *Journal of Network and Computer Applications*, 171:102840, 2020.

15. Quang Tung Thai, Namseok Ko, Sung Hyuk Byun, and Sun-Me Kim. Design and implementation of ndn-based ethereum blockchain. *Journal of Network and Computer Applications*, 200:103329, 2022.
16. Hakima Khelifi, Senlin Luo, Boubakr Nour, Hassine Moungla, Syed Hassan Ahmed, and Mohsen Guizani. A blockchain-based architecture for secure vehicular named data networks. *Computers & Electrical Engineering*, 86:106715, 2020.
17. Hao Liu, Rongbo Zhu, Jun Wang, and Wengang Xu. Blockchain-based key management and green routing scheme for vehicular named data networking. *Security and Communication Networks*, 2021:1–13, 2021.
18. Junjun Lou, Qichao Zhang, Zhuyun Qi, and Kai Lei. A blockchain-based key management scheme for named data networking. In *2018 1st IEEE international conference on hot information-centric networking (HotICN)*, pages 141–146. IEEE, 2018.
19. Kan Yang, Jobin J Sunny, and Lan Wang. Blockchain-based decentralized public key management for named data networking. In *The international conference on computer communications and networks (ICCCN 2018)*, 2018.
20. Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*, pages 136–145. IEEE, 2001.
21. Michel Abdalla, Dario Catalano, Céline Chevalier, and David Pointcheval. Efficient two-party password-based key exchange protocols in the uc framework. In *Cryptographers' Track at the RSA Conference*, pages 335–351. Springer, 2008.
22. Zibin Zheng, Shaoan Xie, Hong-Ning Dai, Weili Chen, Xiangping Chen, Jian Weng, and Muhammad Imran. An overview on smart contracts: Challenges, advances and platforms. *Future Generation Computer Systems*, 105:475–491, 2020.
23. Anhao Xiang and Jun Zheng. A lightweight anonymous device authentication scheme for information-centric distribution feeder microgrid. *Computers, Materials & Continua*, 69(2), 2021.
24. Alberto Compagno, Mauro Conti, and Ralph Droms. Onboardicng: a secure protocol for onboarding iot devices in icn. In *Proceedings of the 3rd ACM Conference on Information-Centric Networking*, pages 166–175, 2016.
25. Xian Guo, Yuxi Chen, Laicheng Cao, Di Zhang, and Yongbo Jiang. A receiver-forwarding decision scheme based on bayesian for ndn-vanet. *China Communications*, 17(8):106–120, 2020.
26. Xian Guo, Baobao Wang, Yongbo Jiang, Di Zhang, and Laicheng Cao. Homomorphic encryption based privacy-aware intelligent forwarding mechanism for ndn-vanet. *Computer Science and Information Systems*, 20(1):1–24, 2023.
27. Carlo Blundo, Alfredo De Santis, Amir Herzberg, Shay Kutten, Ugo Vaccaro, and Moti Yung. Perfectly-secure key distribution for dynamic conferences. In *Annual international cryptology conference*, pages 471–486. Springer, 1992.
28. Junji Takemasa, Yuki Koizumi, and Toru Hasegawa. Data prefetch for fast ndn software routers based on hash table-based forwarding tables. *Computer Networks*, 173:107188, 2020.
29. Dylan Yaga, Peter Mell, Nik Roby, and Karen Scarfone. Blockchain technology overview. *arXiv preprint arXiv:1906.11078*, 2019.
30. Carlo Blundo, Alfredo De Santis, Luisa Gargano, and Ugo Vaccaro. On the information rate of secret sharing schemes. In *Advances in Cryptology—CRYPTO'92: 12th Annual International Cryptology Conference Santa Barbara, California, USA August 16–20, 1992 Proceedings*, pages 148–167. Springer, 2001.
31. Andreas F Molisch, Kannan Balakrishnan, Chia-Chin Chong, Shahriar Emami, Andrew Fort, Johan Karedal, Juergen Kunisch, Hans Schantz, Ulrich Schuster, and Kai Siwiak. Ieee 802.15. 4a channel model-final report. *IEEE P802*, 15(04):0662, 2004.

32. Marcel Enguehard, Ralph E Droms, and Dario Rossi. On the cost of geographic forwarding for information-centric things. *IEEE Transactions on Green Communications and Networking*, 2(4):1150–1163, 2018.

Xian Guo is a professor in School of Computer and Communication, Lanzhou University of Technology, China. He is a visiting scholar at University of Memphis. He received his PhD at Lanzhou University of Technology in 2011, and he received his BS at Northwest Normal University. His current research interests include cryptography, design and analysis of security protocol, and blockchain. E-mail: iamxg@ 163.com.

Sarah Almadhehagi is a graduate student in School Computer and Communication at Lanzhou University of Technology, China. She received her Bachelor degree from Taiz University of Computer Science, Yemen, in 2017, and started her master studying in 2020. Her research interests are blockchain, network and information security, design and analysis of security protocol, etc. E-mail: sara.almad2015@gmail.com.

Tao Feng is a professor in School of Computer and Communication at Lanzhou University of Technology, China. He received his PhD in Xidian University in 2008. His current research interests include network and information security, design and analysis of security protocol, etc.

Di Zhang is an associate professor in School of Computer and Communication at Lanzhou University of Technology, China. He received MS and PhD in Communication University of China, China, in 2013 and 2016, respectively. His current research interests include network and information security, and blockchain etc.

Yongbo Jiang is an associate professor in School of Computer and Communication at Lanzhou University of Technology, China. He received his PhD in Xidian University in 2013. His current research interests include network and information security, and Information-Centric Networking etc.

Junli Fang is a lecturer in School of Computer and Communication at Lanzhou University of Technology, China. She received her MS in Beijing Jiaotong University, China, in 2011. His current research interests include cryptography and blockchain, etc.

Received: March 26, 2023; Accepted: January 10, 2024.