Hindawi
Security and Communication Networks
Volume 2021, Article ID 6694058, 19 pages
https://doi.org/10.1155/2021/6694058

Research Article

Fast and Universal Inter-Slice Handover Authentication with Privacy Protection in 5G Network

Zhe Ren,[1] **Xinghua Li**,[1] **Qi Jiang**,[2,3] **Qingfeng Cheng**,[4] **and Jianfeng Ma**[1]

[1]*State Key Laboratory of Integrated Services Networks, and the School of Cyber Engineering, Xidian University, Xi'an 710126, China*
[2]*Network Communication Research Centre, Peng Cheng Laboratory, Shenzhen 518055, China*
[3]*Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin 541004, China*
[4]*State Key Laboratory of Mathematical Engineering and Advanced Computing, Strategic Support Force Information Engineering University, Zhengzhou 450001, China*

Correspondence should be addressed to Xinghua Li; xhli1@mail.xidian.edu.cn

Received 6 November 2020; Revised 14 December 2020; Accepted 11 January 2021; Published 31 January 2021

Academic Editor: Chengzhe Lai

In a 5G network-sliced environment, mobility management introduces a new form of handover called inter-slice handover among network slices. Users can change their slices as their preferences or requirements vary over time. However, existing handover-authentication mechanisms cannot support inter-slice handover because of the fine-grained demand among network slice services, which could cause challenging issues, such as the compromise of service quality, anonymity, and universality. In this paper, we address these issues by introducing a fast and universal inter-slice (FUIS) handover authentication framework based on blockchain, chameleon hash, and ring signature. To address these issues, we introduce an anonymous service-oriented authentication protocol with a key agreement for inter-slice handover by constructing an anonymous ticket with the trapdoor collision property of chameleon hash functions. In order to reduce the computation overhead of the user side in the process of authentication, a privacy-preserving ticket validation with a ring signature is designed to finish in the consensus phase of the blockchain in advance. Thanks to the edge computing capabilities in 5G, distributed edge nodes help to store the anonymous ticket information, which guarantees that the legal users can finish authentication swiftly during handover. Our scheme's performance is evaluated through simulation experiments to testify the efficiency and feasibility in a 5G network-sliced environment. The results show that compared to other authentication schemes of the same type, the overall inter-slice handover delay has been reduced by 97.94%.

1. Introduction

Compared to the previous mobile communication systems, 5G provides its users a more flexible network environment where the Internet can be realized anytime and anywhere, with a faster speed, higher broadband, and lower latency. Services of all kinds and different application scenes have been developed after the concept of 5G was introduced, such as intelligent transportation networks, VR games, and telemedicine. However, different services and application scenes acquire different demands on the network. For example, telemedicine requires the network to be reliable with low latency, but VR games require higher broadband. Consequently, when handling different service demands, the same network cannot satisfy the demands based on different application scenes. For the sake of its satisfaction, network slicing was introduced into the 5G system to slice the network. Therefore, network slicing has become the highlight in both fields of academics [1–3] and enterprises [4, 5]. According to a new report [6], the market share of network slicing is said to witness an increase from around 112,300,000 US dollars in 2017 to around 302,200,000 US dollars in 2022, with a CAGR of 21.9%.

As shown in Figure 1, inter-slice handover would take place when services alternate according to the varying

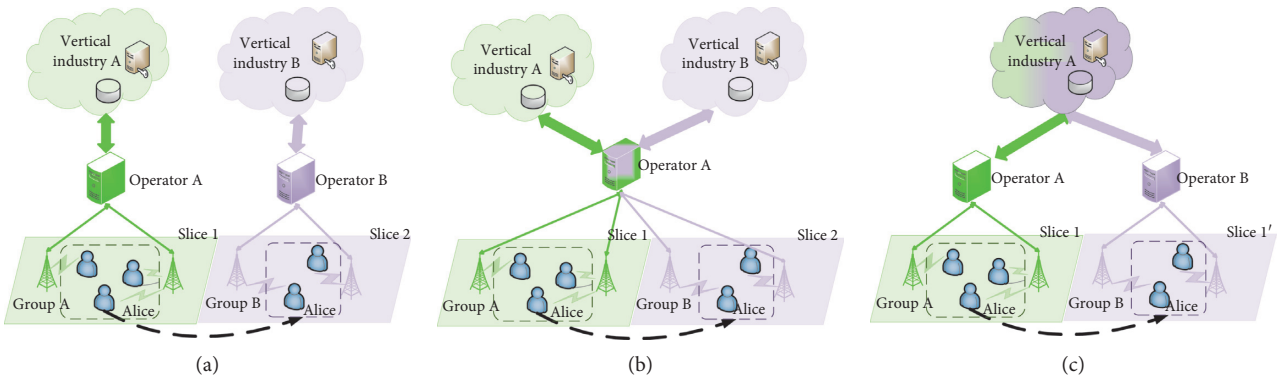

Figure 1: Types of inter-slice handover. (a) Different vertical industries with different operators. (b) Different vertical industries with same operators. (c) Same vertical industries with different operators.

preference of users, such as the service quality of slices, service charge or the change of users' locations [7]. Specifically speaking, on the one hand, based on the service charge of different time intervals, a user would alternate the services among slices in order to reduce the costs to the most extent; on the other hand, the user has to undergo inter-slice handover between different slices when users travel through different geographic areas. Consequently, to prohibit the unauthenticated users from occupying the unsubscribed resources, specific slice authentication should finish before users apply for slice services.

It is worth noting that different network slices may be operated by different slice service providers, just like the different situations in Figure 1. Because of different demands on safety, different operators and service providers would design different authentication protocols [8], but different authentication protocols are incompatible with each other, which will cause the compromise of service quality when users undergo inter-slice handover among different slices. Hence, it is necessary to realize efficient inter-slice handover between slices to meet the needs of certain high-real-time services, such as 5G-assisted drones and automatic driving.

Furthermore, customized slices would provide special services for specific groups and devices such as cars, mobile phones, and IoT devices. However, the slice identifier Single-Network Slice Selection Assistance Information (S-NSSAI) could be linked to drivers for specific purposes. The privacy compromise would happen to drivers who have access to this service if the users' identity and S-NSSAI are transmitted without protection, which would cause the leakage of information including user's identity and types of services [9–11]. For the sake of protecting user privacy, the real identity of users and types of services should be concealed when inter-slice handover authentication happens.

From the perspective of handover authentication, current works [12–15] only focus on when User Equipment (UE) moves from one domain covered by the source Access Point (AP) to another domain covered by the target AP, and the handover authentication is triggered with the decline of the source AP's signal and enhancement of the target AP's signal. However, these schemes cannot support the fine-grained and service-oriented authentication after logical network slices, and edge computing is built for the individual service, which leads to the lack of sufficient security and privacy guarantees. From the perspective of network slicing privacy, the current work [16–18] only focuses on users' privacy protection under the single slice, which does not take the inter-slice handover into consideration and cannot support inter-slice handover as well.

In order to solve the issues mentioned above, we propose a fast and universal inter-slice (FUIS) handover authentication framework with privacy protection. With the support of 5G edge computing, to achieve fast and universal inter-slice handover authentication, a ticket is constructed, which is cached among the edge nodes in the system, making users finish authentication with edge nodes conveniently. Meanwhile, to realize the authentication crossing the operators and slices, the blockchain is introduced in this work to achieve shared authentication information among edge nodes with the feature of data consistency and tamper resistance. Furthermore, the authentication of tickets can be finished in the consensus phase. The main contributions of this paper are as follows:

(i) We propose the FUIS to realize fast and universal inter-slice authentication. In order to achieve the inter-slice handover authentication crossing different operators and slice services, a universal authentication framework is proposed based on the blockchain for the sake of accomplishing and sharing authentication information among edge nodes. Furthermore, users can apply the universal authentication method when they undergo inter-slice handover among different slices.

(ii) A distributed anonymous authentication ticket is designed in this work. Specifically speaking, to achieve anonymous authentication when the user undergoes inter-slice handover, we use the chameleon hash function to construct an implicit link relationship between the ticket and user. To achieve privacy-preserving ticket validation in the consensus phase of the blockchain, the ring signature is applied to conceal the ticket's authoritative source

and protect the types of services behind the ticket. As for authentication efficiency, users can undergo fast inter-slice handover authentication with the ticket data cached among edge nodes.

(iii) Combining ProVerif to analyze the safety of this scheme, the results indicate that the anonymous authentication can be realized. Meanwhile, extensive experiments based on NS-3 simulation show that our proposed protocol in this work has excellent performance concerning the overall inter-slice handover delay. Compared to the slice authentication [16], the overall inter-slice handover delay is reduced by 97.94%.

The remainder of the paper is organized as follows: firstly, previous studies on handover authentication and slice authentication are reviewed in Section 2; secondly, knowledge on blockchain and cryptology are introduced in Section 3; thirdly, the system model is presented in Section 4; then, FUIS protocols proposed in this work will be described in detail in Section 5; next, security attributes of this scheme are analyzed in Section 6; afterward, detailed descriptions on the prototype system of this work, including the experiment results of registration, authentication, and key agreement, are provided in Section 7; last but not least, the conclusion to this work will be made in Section 8.

2. Related Work

2.1. Handover Authentication in the Mobile Communication Network. Due to the limitation of the coverage area of Access Points (APs), during mobility, it is common for users to undergo handover among APs. With the development of heterogeneous networks, handover authentication evolved from a single-type Access Network (AN) to different ANs, which makes handover authentication face more severe security challenges. At present, some works focus on the latter situation. Since servers for authentication were placed far from users, so in order to complete the authentication of a base station, more interactions should be made to authentication servers, which will cause the delay for hundreds of milliseconds [19]. The delay is unacceptable for many real-time services.

For the sake of reducing the authentication delay, Choi and Jung [20] first simplified the process by adopting the idea of direct authentication, which achieved bidirectional authentication and key agreement between users and base stations only in three rounds of interactions without the participation of the AAA server. However, even though this scheme simplified the process of authentication, it would cost much since more technologies related to cryptology are concerned with this process. Meanwhile, in order to erase the existence of AAA servers in the process of authentication, the scheme designed by Cai et al. [21] and Haddad et al. [22] was proposed based on the security context. By adopting this scheme, a predetermined alternative target AP should be selected, and authentication information should be sent to the target AP. This scheme did solve the problem of costing too much on computation. However, this scheme would increase the information interaction between users and base stations or between base stations, which depended on the confidential relationship between base stations.

To realize more efficient handover authentication, Vassilakis et al. [23] introduced Mobile Edge Computing (MEC) servers to assist authentication, making MEC servers reduce the delay of authentication caching users' information. However, the core network still needed to exchange users' information with the MEC server, which will increase communication overhead. From another aspect, since plenty of users' information is stored on the MEC server, the cached information is vulnerable. Leakage of users' information will happen when MEC servers come under attack. To make for the deficiency of bad security conditions, Lee et al. [24], Yazdinejad et al. [25], and Yazdinejad et al. [25] applied the blockchain to, on the one hand, share users' information among different APs, and, on the other hand, finish faster handover authentication through shared information. However, the above schemes only focused on the legitimacy of users' identities but neglected the existence of slices. Furthermore, authentication for specific slices cannot be provided; consequently, service-oriented authentication cannot be achieved, which makes current handover authentication schemes inapplicable to inter-slice handover authentication.

2.2. Authentication in 5G Slices. With the support of network slicing, third-party service providers are able to rent slices within the 5G network. Besides, Service Level Agreement (SLA) could be achieved with operators concerning indexes such as service quality and data bandwidth. According to the work of Lu et al. [26], the key security issue is how to perform access authentication and authorization for a specific network slice.

In the Internet of Things (IoT) field, Ni et al. [16] presented a service-oriented authentication framework supporting network slicing. This framework allows users to acquire anonymous authentication tickets authorized by operators and IoT server (ISV) in the registration phase. When users request services, these anonymous authentication tickets could be applied to authentication with the ISV. Although this work proposed an authentication framework supporting network slicing, this framework did not consider the inter-slice handover. Since, on the one hand, tickets should be transmitted from users to the ISV to undergo verification in the authentication phase; on the other hand, the bilinear pairing cryptology primitive was adopted in this framework, and the overhead of the authentication phase was of great amount; for example, the costs of the user side would reach 332.544 ms when processing handover authentication, which did not satisfy the demand of the real-time service. Furthermore, only an abstract ISV authentication server was constructed in this model, which did not take the situation where many ISV servers would be adopted in arrangement of network slices into the account, thereupon the ticket in [16] was not a universal feature, and an extra mechanism was required to finish inter-slice handover authentication.

Besides, in order to realize the secure cooperation among Network Slice Components (NSCs), Sathi et al. [27] proposed a new re-encrypted scheme based on agencies. This scheme provided anonymous services for NSCs groups under the Service Provider (SP) by applying bilinear features on the elliptic curve. According to the protocols proposed by Sathi et al. [27], NSC under different SP cannot distinguish the identities of SP, which can contribute to the isolation of slices. However, the scheme mentioned above only concentrates on the cooperation among NSCs. It cannot be applied to the user's communication under different slices. In order to achieve secure cross-slice communication among users, Liu et al. [28] proposed a two-hybrid combined signature scheme, PKI-CLC Heterogeneous Signcryption (PCHS) and CLC-PKI Heterogeneous Signcryption (CPHS), which can guarantee the security of cross-slice communication to different users under Certificateless Public Key Cryptography (CLC) and PKI environments. However, Sathi et al. and Liu et al. [27, 28] do not consider the third-party slice service, so it is impossible to achieve the authentication and authorization to third-party slices. For the sake of achieving them, Behrad et al. [9] proposed an authentication mechanism named 5G-Slice Specific Authentication and Access Control (5G-SSAAC), which could reduce the load of the core network by entrusting the third-party slice providers with users' identity authentication and access control, whereas Behrad et al. [9] only raised a protocol framework without presenting the concrete realization of the protocol. Based on the work done by Behrad et al. [9], they [29] designed a new network function within the 5G RAN (Ratio Access Network); specifically speaking, the protocols for users to link third-party slices were designed, which makes the third-party slice providers choose corresponding Authentication and Access Control (AAC) according to their security demands.

Concentrating on power infusion in the 5G smart grid slice, Zhang et al. and Kamil et al. [17, 18] designed the schemes for batch authentication. Specifically speaking, Zhang et al.'s study [17] was based on hash-then-homomorphic technology, and Kamil and Ogundoyin's study [18] was based on noncertificate aggregate signature technology. In order to make a supplement to protect privacy among peer-to-peer users, Sathi et al. [30] proposed a grouping anonymous mutual authentication scheme of antitopological learning attacks in the formulation phase of the slice. Simultaneously, a group anonymous one-way authentication scheme is proposed to protect users' service access behavior. However, although the above schemes were all based on the authentication under network slices, they do not take the users' demand for fast authentication in the process of inter-slice handover into consideration. Consequently, the previous work cannot satisfy the requirement of quick inter-slice handover authentication.

3. Preliminaries

3.1. Blockchain. Originated from Bitcoin proposed by Satoshi Nakamoto [31], as its underlying technology, the essence of the concept of the blockchain is a distributed database. Taking Bitcoin as an example, the miner would pack the transaction existing in the current blockchain network and strive for bookkeeping. Once acquiring the bookkeeping, the miner would pack transaction data into one block and link this block to the previous one. After that, the information on this chain would be broadcast to the blockchain network. When being confirmed by six blocks in succession, all transactions are confirmed and tamper resistant. Furthermore, the transaction data are immutable and are distributed into every node to be saved.

According to public degrees, the blockchain can be subdivided into three categories: public blockchain, consortium blockchain, and private blockchain. For the purpose of achieving consistency in the distributed memory, a consensus algorithm is designed in the blockchain system. Commonly speaking, Proof of Work (PoW), Proof of Stake (PoS), and Byzantine Fault Tolerance (BFT) are all related to the consensus algorithm. However, the participants of blockchains (i.e., miners) always add one transaction in the blockchain with the intention of proving their workload. In order to achieve this, some "complicated but unhelpful" algorithm problems should be solved. Under this consensus mechanism, the computing capability of miners is wasted.

Considering the waste on miners' computing capability, Berkeley Open Infrastructure for Network Computing (BOINC) manifested that the underlying blockchains are required to be upgraded and hoped that blockchains can facilitate the development of BOINC in their recently released White Book, in which BOINC proposed a consensus mechanism named Proof of Valuable Computing (PoVC) that can introduce the computation resources to more applicable scenes with practical significance.

The blockchain module in the FUIS also uses this method as a reference. In detail, the computation task for ticket validation is transferred to miners to accomplish in advance; accordingly, the overhead of inter-slice handover authentication will be reduced a lot.

3.2. Chameleon Hash. Chameleon hash is a special type of hash function [32]; it can satisfy the collision resistance of the hash function for most of its users. Nevertheless, if others have some knowledge about chameleon hash, the hash's collision resistance can be compromised easily. In other words, to any m, it is easy to find 'm' to make $\mathrm{CH}(m) = \mathrm{CH}(m')$. However, although it seems to break the hash's collision resistance, for most users, the hash function is still secure.

Based on Elliptic Curve Cryptography (ECC), the chameleon hash function is introduced. Users choose initial values (m^*, r^*), within which $m^*, r^* \in \mathbb{Z}_q^*$. To the chameleon hash function, if we input (m, r), a hash value can be calculated as $\mathrm{CH}_Y(m, r) = mP + rY$, in which (P, Y) can be applied to acquire the hash value and can be called as the hash key. Furthermore, (k, x) is the trapdoor, in which $x \in \mathbb{Z}_q^*, Y = xP$ and $k = m^* + r^*x$. The chameleon hash function has the following properties:

(1) Collision resistance: for those who do not know the trapdoor, it is difficult to find $m', r' \in \mathbb{Z}_q^*$, $(m, r) \neq (m', r')$ to make $\mathrm{CH}_Y(m, r) = \mathrm{CH}_Y(m', r')$

(2) Collision based on the trapdoor: giving $r' \in \mathbb{Z}_q^*$, for those who know the trapdoor, it is easy to find $m' = k - r'x \pmod{q}$ to make $\mathrm{CH}_Y(m^*, r^*) = \mathrm{CH}_Y(m', r')$

3.3. Ring Signature. The ring signature [33] is a special type of signature, such as the group signature; it also can achieve an anonymous signature. In other words, the verifier is only aware that the signer belongs to a certain group without knowing the signer's concrete identity. Comparing to the group signature, a user can produce a ring signature without negotiating with other users who only need to collect the public key of other users to form a ring and add his private key to this ring. This process is highly anonymous and cannot be used to disclose the identities of signers.

The ring signature based on ECC is introduced by work [34], which can be used to protect the data privacy on the blockchain. The ring signature function can be defined as $\mathrm{Sig}_{\mathrm{Ring}}([\mathrm{RG}, sk], m) \longrightarrow \mathrm{RC}$, in which RG represents the public key collection of ring members generating the ring signature, sk represents the private key of certain ring member's public key, m represents the information being signed, and the output result RC represents the generated ring signature. Meanwhile, the verification function of the ring signature can be defined as $\mathrm{Verify}_{\mathrm{Ring}}(\mathrm{RG}, \mathrm{RC}) \longrightarrow \{\mathrm{True}, \mathrm{False}\}$ in which RG and RC have the same meaning as the above function, and the output can be either true or false.

4. System Model

4.1. Network Model. As shown in Figure 2, the system structure consists of 5 parts, namely, the mobile network user, edge controller, mobile network operator, slice service provider, and blockchain.

4.1.1. Mobile Network User. Users are diversified in the 5G system. It consists of different elements such as mobile terminals, mini-type devices for IoT, and intelligent car.

4.1.2. Edge Controller. Edge computing is an essential concept in the 5G system. Edge controller, with the purpose of placing the devices near base stations, is empowered by a wired connection to establish a communication system with the assistance of core network and base stations, making the edge of the system have stronger abilities to compute and store. Consequently, edge controllers are served as miners to maintain the blockchain.

4.1.3. Mobile Network Operator. Mobile network operators, responsible for the operation of the 5G network and leasing business of network slices, are comprised of several functional network modules in its core network such as Access and Mobility Management Function (AMF), Session Management Function (SMF), and Authentication Server Function (AUSF). Specifically speaking, the AMF is principally responsible for the registration of users, management of connection, management of accessibility, management of mobility, and identity authentication; the SMF is for the management of sessions, such as their establishment, modification, and releasing; the AUSF is mainly for the access authentication.

4.1.4. Slice Service Provider. There are two responsibilities for slice service providers: to rent slices to mobile network operators and to provide dedicated services to specific users. To protect the slices' resources from being occupied by unauthenticated users, AAA Server of the 3rd Vertical Industry (A3VI) is provided to slice service providers to guarantee the users who could undergo specific slice authentication.

4.1.5. Blockchain. An anonymous ticket will be assigned to the user when the specific slice registration is processed. This ticket is published on the blockchain by slice service providers, and validation of the ticket is carried out by miners. After finishing the validation, the ticket will be stored on the blockchain. Data of anonymous tickets will be cached on each of the edge controllers for the sake of facilitating users to achieve inter-slice authentication swiftly.

4.2. Adversary Model. With stronger extensibility and more flexible openness, the 5G network is more easily attacked by security threats from both internal and external perspectives. According to the previous work [35–39], the principal attacking objects of the 5G network are identity privacy, completeness, and accessibility of data. For instance, attackers can either acquire the data package by launching intercept attacks or acquire session keys by launching Man-in-the-Middle (MitM) attacks. These external attacks threatening the service security and users' privacy are major security threats to the service facing slice network framework. In this case, the trustiness of protocol participants is defined.

Because mobile network operators, slice service providers, and edge controllers are the operators and users of the whole network, it can be believed that they are not motivated to sabotage network facilities. However, it will be possible for them to record and analyze users' service data out of their curiosity and sincerity. For users, as beneficiaries of network the slicing service and 5G network, although users will not attack the network facilities on purpose, they may avoid charges by disguising into other users. Therefore, it can be presumed that users are malicious.

4.3. Design Goals. In the FUIS, air interface messages can be sniffed by attackers. Based on the attacks proposed in previous work [8, 10, 11, 14, 38, 40], it can be believed that attackers are highly possible to launch classical protocol attacks in the 5G network, such as impersonation attacks, reply attacks, and MitM attacks. Therefore, there are five

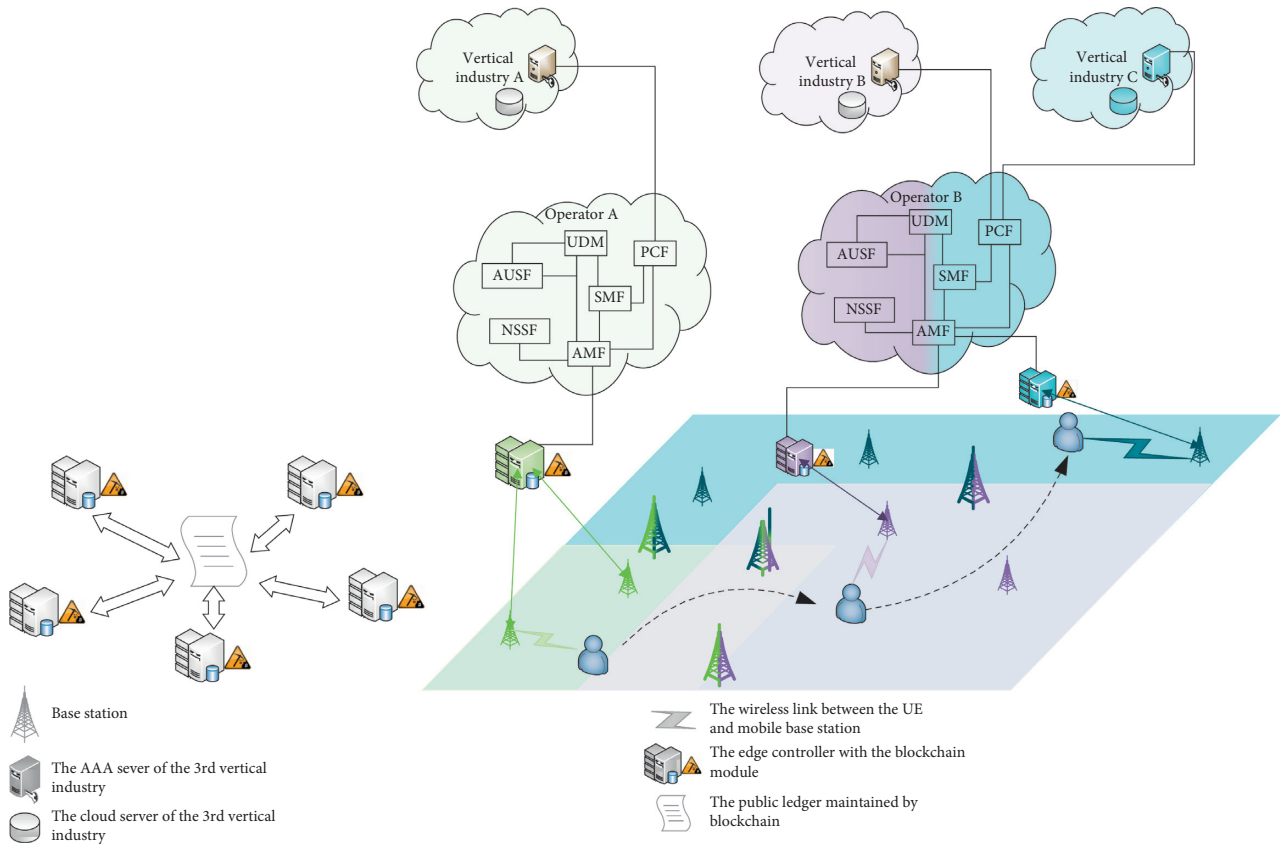

Figure 2: The edge-assisted 5G network model supporting network slices.

design goals that need to be considered when designing inter-slice handover authentication protocols.

4.3.1. Inter-Slice Handover Authentication. To guarantee that the service resource of slices is not occupied by illegal subscribers, the inter-slice handover authentication should be executed when inter-slice handover needs to be processed due to the reason of UE or network. This process can assure users that they subscribe to the slice services and have the right to access the network supported by these slices.

4.3.2. Identity Anonymity in Network Slices. For the purpose of avoiding the compromise of users' identities in the process of authentication and service, users are expected to finish authentication by applying their pseudoidentities without manifesting their real identities in the process of inter-slice handover authentication. Meanwhile, the unlinkability of certain users' sessions in any two slices to external interceptors should be guaranteed by the pseudoidentities.

4.3.3. Fast Authentication. Considering there is an authentication ticket, we hope that the validation can be finished in advance before the formal authentication. Besides, we must guarantee there is no message leakage about the service type of the ticket during validation. And, to satisfy the low-latency of the 5G network, users only need to interact with the nearest edge controllers when undergoing authentication. During identity authentication, users should avoid consuming waiting time in communication and computation with the far-end AAA server.

4.3.4. Traceability. The anonymity of users is like a "double-edged sword." More specifically, some users would perpetrate without considering liabilities by taking advantage of anonymity. Consequently, it is necessary to establish a tracing mechanism to disclose users' real identities in the system, that is, the tracing mechanism is compiled to assure AAA servers/supervisors to trace and disclose users' identities that violate regulations.

4.3.5. Key Escrow Freeness. Moreover, the long-term key applied for users' authentication was assigned by the AAA server/Private Key Generator (PKG) in previous plans, which would make long-term keys to be intercepted by attackers. Also, a single point of failure could also attribute to the disclosure of the long-term key. For the sake of avoiding this dilemma, the long-term keys of users are expected to be decided by themselves.

4.3.6. Key Agreement with Forward Secrecy. When users switch the services under the slices, the service data between the user and new slice could be intercepted. Therefore, an independent session key should be negotiated when inter-

slice handover authentication is processed. To resist some potential attacks, such as the master keys of users or AAA servers are crashed and the disclosure of temporary random numbers during negotiation, the protocol we designed needs to realize Perfect Forward Secrecy, Master-Key Forward Secrecy, and Known Randomness Secrecy.

5. The Proposed FUIS

For the ease of reference, some important symbols and explanations are provided in Table 1.

5.1. Overview of the FUIS. We design an inter-slice handover protocol with privacy protection. Under this protocol, users need to register with slice service providers in advance. During the registration process, users should calculate a chameleon hash value CH_{UE} and provide this value CH_{UE} and related registration information to slice service providers through operators. Meanwhile, users should preserve the trapdoor value, which would be applied during inter-slice handover authentication. After authenticating the received registration information from users, operators could produce a public key ring RG_i^{Ope} and produce the ticket PST_i by generating a ring signature based on the chameleon hash value CH_{UE}. It can be used to manifest the authentication of users from operators. Furthermore, after saving the ID_{UE} of users, operators can send the PST_i to slice service providers; meanwhile, the ticket PST_i would be processed by A3VI. Likewise, the ring signature is used on the PST_i to produce the ticket of ST_i. Finally, A3VI sends ticket A and information testifying the ticket's validity to the blockchain network. In a moment, they would be validated by miners and saved on the chain, during which the identities of users and service types of tickets would be kept confidential. After being uploaded on the chain, the ticket number would be returned to A3VI, and A3VI would return the ticket number to users.

During the inter-slice handover authentication process, when the user undergoes inter-slice handover to the next new slice, the interactions should be made only with edge controllers. After showing the ticket number to edge controllers, edge controllers could use the ticket number to check the blockchain for acquiring the ticket's information (including the value of chameleon hash). Afterward, users could apply the trapdoor into calculating the hash collision value recorded by the tickets on the chain. In that case, authentication could be finished to prove that the user is the rightful owner of tickets on the chain. During this process, the user only provides a pseudonym and can complete the anonymous authentication in the meantime. Finally, based on A3VI, the user and slice service provider could negotiate with each other to establish a session key to serve encrypted communication later.

5.2. The Detailed FUIS

5.2.1. System Initialization. To guarantee the participants of the protocol, such as users, operators, and slice service providers, can process calculation and protocol interactions under the same standard, the initialization of the system can be finished as follows.

E_p is an elliptic curve on the finite field $GF(p)$. P, whose order is the prime number q, is a point on the curve. $\mathbb{G}$ represents the addition cycle subgroup preceding E_p generated by P. The general system is initialized through the following four steps:

(1) The operator chooses secure hash functions:

$$H_0: \{0,1\}^* \times \mathbb{Z}_q^* \longrightarrow \mathbb{Z}_q^*$$
$$H_1: \{0,1\}^* \times \mathbb{G}^2 \times \{0,1\}^* \longrightarrow \mathbb{Z}_q^*$$
$$H_2: \{0,1\}^* \times \mathbb{Z}_q^* \times \mathbb{G}^2 \times \{0,1\}^* \times \mathbb{Z}_q^* \times \mathbb{G}^2 \times \{0,1\}^* \longrightarrow \{0,1\}^\lambda$$
$$H_3: \mathbb{G} \times \{0,1\}^\lambda \times \{0,1\}^* \times \mathbb{Z}_q^* \times \mathbb{G}^2 \times \{0,1\}^* \longrightarrow \{0,1\}^\lambda$$

(2) The operator assigns the chameleon hash function $CH_Y(m,r)$ to users.

(3) The generation of the public/private key pair, (pk_{Ope}, sk_{Ope}): pk_{Ope} is for encryption and verification; meanwhile, sk_{Ope} is for signature. After the generation of the public and private key pair, operators apply for registration to CA to acquire the certificate $Cert_{Ope}$. Likewise, the A3VI of slice service operators can also generate the public/private key pair (pk_{A3VI}, sk_{A3VI}) to register the certificate $Cert_{A3VI}$. Users can also generate the public/private key pair (pk_{UE}, sk_{UE}) and register the certificate $Cert_{UE}$.

(4) Finally, the operator publishes the system public parameter $PK = \{q, P, \mathbb{G}, H_0, H_1, H_2, H_3, CH_Y(m,r)\}$.

5.2.2. Slice Service Registration. Before applying the slice's service, users would register the corresponding slice services in advance, as shown in Figure 3.

Step 1: the user will choose parameter $x_{UE}, s_{UE}, m_{UE}^* \in \mathbb{Z}_q^*$ and calculate $Y_{UE} = x_{UE}P, r_{UE}^* = H_0(ID_{UE} \| s_{UE})$ to make $CH_{UE} = CH_{Y_{UE}}(m_{UE}^*, r_{UE}^*)$. After generating the chameleon hash value CH_{UE}, the user would choose a session value $N_i \in \mathbb{Z}_q^*$ and a symmetric key key_1. Afterward, the symmetric encryption algorithm AES is applied to calculate $UText = AES_{ENS}(key_1, CH_{UE} \| N_i \| ID_{UE} \| ID_{A3VI})$ and uses the public key from operators to encrypt key_1 as $E_1 = Enc(pk_{Ope}, key_1)$. Meanwhile, with the purpose of maintaining the nonrepudiation and integrity of the message, users apply their private keys to generate the signature $\sigma = Sig(sk_{UE}, hash(UText \| E_1))$. Finally, the user will send **message 1** $\langle \mathbf{UText}, \mathbf{E}_1, \sigma \rangle$ to the AMF of operators.

Step 2: after receiving the information $\langle UText, E_1, \sigma \rangle$, the AMF firstly uses the public key of the user pk_{UE} to verify the signature σ. If it fails, the request from the user will be rejected, and the connection will be terminated. If it succeeds, the AMF will send **message 2** $\langle \mathbf{UText}, \mathbf{E}_1 \rangle$ to the SMF.

Table 1: Explanations on symbols in the FUIS.

Notation	Meaning
$x \in_R X$	x is chosen at random from a set X
$l_1 \| l_2$	The concatenation of two-bit strings l_1 and l_2
H_i	A secure hash function, for $i = 0, 1, 2, 3$
$\mathrm{GF}(p)$	A finite field of the prime power order p
E_P	An elliptic curve over $\mathrm{GP}(p)$
$\mathbb{G}$	The subgroup of the prime order q in E_P
$\mathbb{Z}_q^*$	A finite field of integers modulo prime q
ID_E	The identity of the entity E
$\mathrm{CH}(\cdot)$	A secure chameleon hash function
T_{Curr}	The current time used as a timestamp
T_{Exp}	The expiration time of the ticket
$\mathrm{AES}_{\mathrm{ENC}}(\mathrm{key}, m)/\mathrm{AES}_{\mathrm{DEC}}(\mathrm{key}, m)$	The AES algorithm to encrypt and decrypt the message m with the key
$\mathrm{ENC}(\mathrm{pk}, m)$	The asymmetric encryption of the public key pk
$\mathrm{Sig}(\mathrm{sk}, m)$	The signature on the message m of the private key sk
RG^E	A set of public keys for the ring signature, for $E = \{\mathrm{A3VI}, \mathrm{ope}\}$
$\mathrm{Sig}_{\mathrm{Ring}}([\mathrm{RG}, \mathrm{sk}], m)$	A ring signature with the RG and the signer's secret key sk
$\mathrm{Verify}_{\mathrm{Ring}}(\mathrm{RG}, m)$	Verify the algorithm for the ring signature

Step 3: the SMF uses the private key $\mathrm{sk}_{\mathrm{Ope}}$ from operators to decrypt E_1 to acquire the key key_1. After obtaining key_1, the SMF could decrypt UText and get $\mathrm{CH}_{\mathrm{UE}}, N_i, \mathrm{ID}_{\mathrm{UE}}$, $\mathrm{ID}_{\mathrm{A3VI}}$. Next, the SMF sends the **message 3** $\langle \mathbf{CH_{UE}}, \mathbf{N_i}, \mathbf{ID_{UE}}, \mathbf{ID_{A3VI}} \rangle$ to the AUSF to proceed.

Step 4: after receiving the information, the AUSF firstly chooses the public key group of the ring signature $\mathrm{RG}_i^{\mathrm{Ope}} = \{\mathrm{pk}_1, \mathrm{pk}_2, \ldots, \mathrm{pk}_i, \ldots \mathrm{pk}_n\}$, in which the formation of $\mathrm{RG}_i^{\mathrm{Ope}}$ is the public key of other operators forming the 5G network. After ascertaining $\mathrm{RG}_i^{\mathrm{Ope}}$, the AUSF produces the ticket $\mathrm{PST}_i = \mathrm{Sig}_{\mathrm{Ring}}([\mathrm{RG}_i^{\mathrm{Ope}}, \mathrm{sk}_{\mathrm{Ope}}], H_0(\mathrm{CH}_{\mathrm{UE}} \| N_i))$. Afterward, the AUSF chooses a key key_2 and uses the AES algorithm to calculate $C\mathrm{Text} = \mathrm{AES}_{\mathrm{ENS}}(\mathrm{key}_2, \mathrm{CH}_{UE} \| N_i \| \mathrm{RG}_i^{\mathrm{Ope}} \| \mathrm{PST}_i)$, in the meantime, the public key of A3VI is applied to encrypt key_1 as $E_2 = \mathrm{Enc}(\mathrm{pk}_{\mathrm{A3VI}}, \mathrm{key}_2)$. Furthermore, for protecting the nonrepudiation and integrity of the message, the AUSF uses the private key from operators to produce the signature $\beta = \mathrm{Sig}(\mathrm{sk}_{\mathrm{Ope}}, \mathrm{hash}(C\mathrm{Text} \| E_2))$. Finally, users send **message 4** $\langle \mathbf{CText}, \mathbf{E}_2, \beta \rangle$ to the slice service provider whose identity is $\mathrm{ID}_{\mathrm{A3VI}}$.

Step 5: after the slice service provider receives the message $\langle C\mathrm{Text}, E_2, \beta \rangle$, firstly, A3VI uses the public key $\mathrm{pk}_{\mathrm{A3VI}}$ provided by slice service providers to verify the signature β. After being verified successfully, A3VI uses its private key $\mathrm{sk}_{\mathrm{A3VI}}$ to decrypt E_2 to get the key key_2. After getting key_2, A3VI could decrypt $C\mathrm{Text}$ and get $\mathrm{CH}_{UE}, N_i, \mathrm{RG}_i^{\mathrm{Ope}}, \mathrm{PST}_i$. In the meantime, A3VI chooses a public key group of ring members $\mathrm{RG}_i^{\mathrm{A3VI}} = \{\mathrm{pk}_1, \mathrm{pk}_2, \ldots, \mathrm{pk}_i, \ldots, \mathrm{pk}_n\}$, in which pk_i is the public key $\mathrm{pk}_{\mathrm{A3VI}}$ of A3VI itself and $\mathrm{RG}_i^{\mathrm{A3VI}}$ is the public key of other A3VIs in the 5G network. After ascertaining $\mathrm{RG}_i^{\mathrm{A3VI}}$, A3VI generates tickets $\mathrm{ST}_i = \mathrm{Sig}_{\mathrm{Ring}}([\mathrm{RG}_i^{\mathrm{A3VI}}, \mathrm{sk}_{\mathrm{A3VI}}], \mathrm{PST}_i)$. Finally, A3VI sends information $\mathrm{data}_{Tx} = (\mathrm{CH}_{UE}, N_i, T_{\mathrm{Exp}}, \mathrm{RG}_i^{\mathrm{Ope}}, \mathrm{RG}_i^{\mathrm{A3VI}}, \mathrm{ST}_i)$ to the network of the blockchain. During the mining, miners would verify data_{Tx} to ensure the ticket ST_i can prove the ticket owner has been authorized legally to access the corresponding slice service. The methods of verification are as follows: firstly, miners use $\mathrm{CH}_{\mathrm{UE}}$ and N_i to generate $H_0(\mathrm{pkn}_i \| N_i)$; next, $\mathrm{Verify}_{\mathrm{Ring}}(\mathrm{RG}_i^{\mathrm{Ope}}, \mathrm{Verify}_{\mathrm{Ring}}(\mathrm{RG}_i^{\mathrm{A3VI}}, \mathrm{ST}_i))$ will be calculated by miners. If equation $H_0(\mathrm{pkn}_i \| N_i) = \mathrm{Verify}_{\mathrm{Ring}}(\mathrm{RG}_i^{\mathrm{Ope}}, \mathrm{Verify}_{\mathrm{Ring}}(\mathrm{RG}_i^{\mathrm{A3VI}}, \mathrm{ST}_i))$ is established, it can be assumed that the ticket is an authorized ticket signed by legitimate A3VI and operators during the registration. Besides, it also can be acknowledged that the owner of tickets can visit the corresponding slice services. Finally, miners record $\mathrm{Message}_{\mathrm{TXID}} = (\mathrm{CH}_{\mathrm{UE}}, T_{\mathrm{Exp}})$ on the chain, as shown in Figure 4. If the equation cannot be established, miners will dispose data_{Tx}. When $\mathrm{Message}_{\mathrm{TXID}}$ is recorded on the chain successfully, A3VI will receive a transaction number $\mathrm{TXID}_{\mathrm{ST}_i}$ recorded on the chain. This number can be applied to ascertain the location where the data is recorded on the chain. Afterward, A3VI would send **message 5** $\langle \mathbf{TXID_{ST_i}}, \mathbf{T_{Exp}} \rangle$ to the AUSF and send **message 6** $\langle \mathbf{TXID_{ST_i}}, \mathbf{T_{Exp}} \rangle$ to UE. After receiving TXID_{ST_i}, the AUSF would keep $\mathrm{ID}_{\mathrm{UE}} \| \mathrm{ID}_{\mathrm{A3VI}} \| \mathrm{TXID}_{ST_i} \| T_{\mathrm{Exp}}$ in the local storage, for the sake of facilitating the anonymous tracking to users who has malicious behavior in the future. After receiving $\mathrm{TXID}_{\mathrm{ST}_i}$, UE would confirm whether the ticket has been on the blockchain with $\mathrm{TXID}_{\mathrm{ST}_i}$. Specifically speaking, users would use $\mathrm{TXID}_{\mathrm{ST}_i}$ to locate transactions. For the sake of guaranteeing that the transactions are created by A3VI, users would check whether the Sigscript included in the input script contains the ring signature of $\mathrm{pk}_{\mathrm{A3VI}}$ or

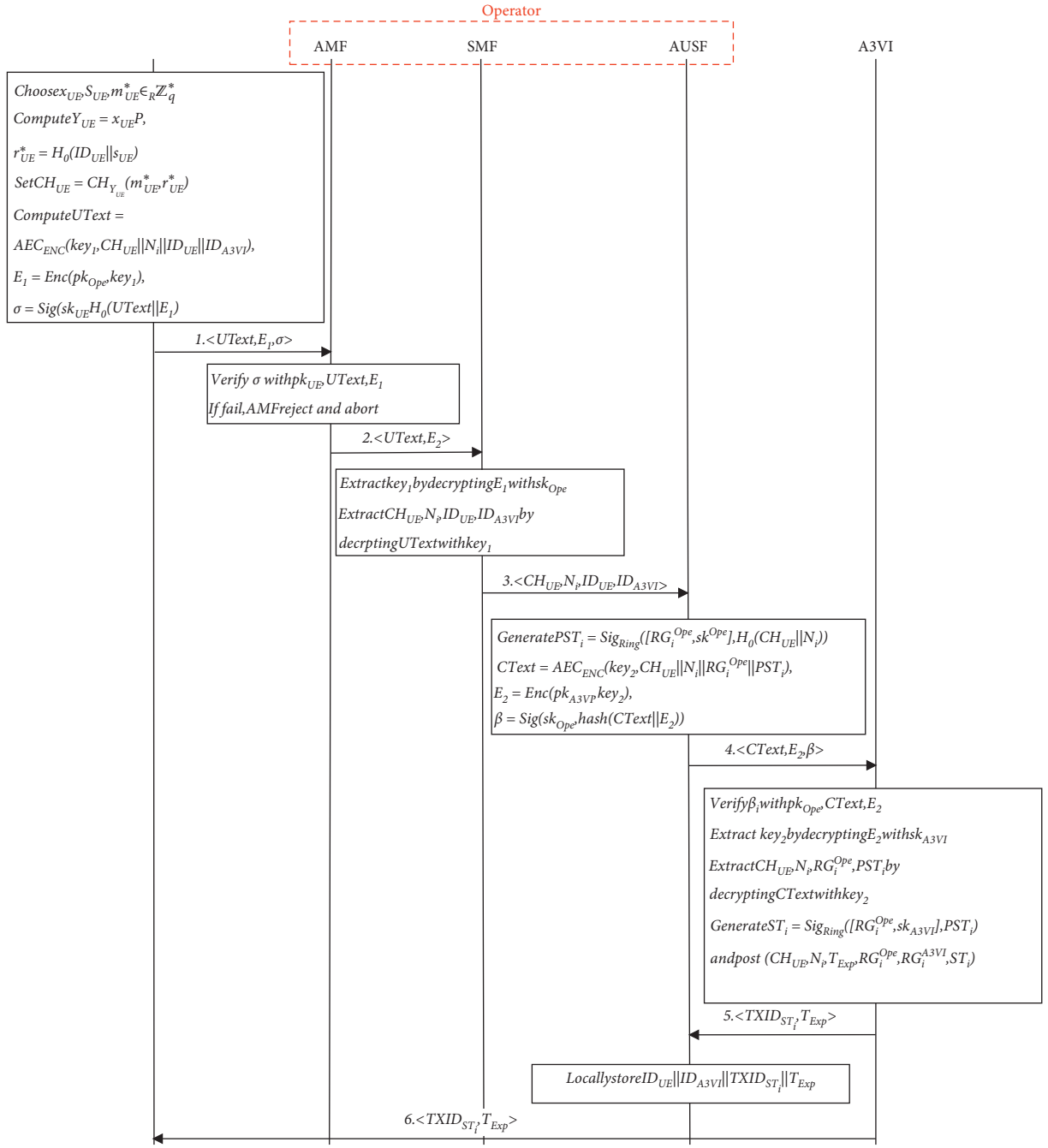

Figure 3: The flow diagram of slice service registration.

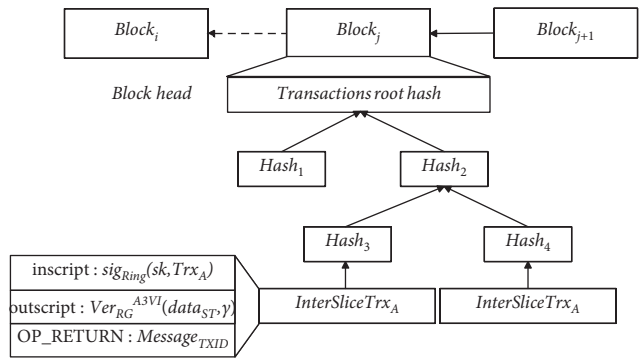

Figure 4: Data structure of ticket information on the blockchain.

not. Meanwhile, the output script Pkscript would be checked to examine whether it includes a ring signature of pk_{A3VI} or not, in other words, examining and using RG_i^{A3VI} to verify the legitimacy of the ring signature γ. Finally, verification on whether OP_RETURN saved the CH_{UE} sent initially or not is processed. After UE verifying the ticket is on the chain, the phase of ticket authentication is finished.

5.2.3. Inter-Slice Handover Authentication with Key Agreement. As shown in Figure 5, the operators' network is divided into several virtual network slices: Slice 1, Slice 2, . . ., Slice n, Default Slice. With the purpose of protecting user privacy, accessing types and concrete slice ID are acknowledged by edge nodes through calculating $HSST_i = H_1(SST_i \| ID_{A3VI})$, among which $i \in \{1, 2, \ldots, n\}$ by the operator's network. Afterward, the AMF sends $HT = (HSST_1, HSST_2, \ldots, HSST_n)$ to edge controllers, who will save the HT. When required by users, edge controllers will use HT to choose slices to realize the privacy-protected handover effect. It should be noticed that if some slices are changed in the situation, the corresponding $HSST_i$ is supposed to be updated in the edge controller.

Each user has a user's identifier ID_{UE}. With the purpose of connecting to the 5G network of operators, users should execute every registration and authentication process mentioned in 3GPP TS.33.201 [41]. Specifically speaking, users would execute primary authentication with the AUSF from operators, take registration to operators, and acquire subscription information, including subscribed S-NSSAIs. After finishing primary authentication, UE would obtain the allowed NSSAIs of subscription information and establish NAS (nonaccess Stratum) of the nonaccess layer to acquire the security context.

When the user undergoes inter-slice handover from slice 1 to slice 2 due to the network or their own preferences, the UE needs to find the ticket number of slice 2 and prepare for the future inter-slice handover authentication.

As shown in the authentication section in Figure 6, when user UE needs to undergo inter-slice handover, Hidden Allowed $S-NSSAI = H_1(SST_i \| ID_{A3VI})$ should be calculated first. Then, a pseudonym $PID_{UE} \in \{0, 1\}^*$ and two random numbers $\alpha_{UE}, \beta_{UE} \in_R \mathbb{Z}_q^*$ are selected. After that, $A_{UE} = \alpha_{UE} Y_{UE}$ and $B_{UE} = \beta_{UE} Y_{UE}$ are calculated. To prove that the user is the owner of the ticket with $TXID_{ST_i}$ on the blockchain, the user lets $\gamma_{UE} = H_1(PID_{UE} \| A_{UE} \| B_{UE} \| T_{Curr})$ and calculates $m_{UE} = k_{UE} - r_{UE} x_{UE}$, where T_{Curr} is a timestamp, $r_{UE} = \alpha_{UE} \gamma_{UE}$. Finally, UE sends the message $\langle$Hidden Allowed $S-NSSAI, PID_{UE}, A_{UE}, B_{UE}, m_{UE}, T_{Curr}, TXID_{ST_i}\rangle$ to the edge controller EC.

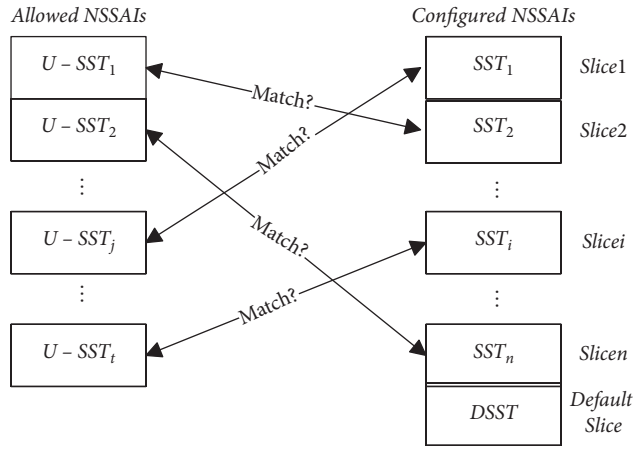

Figure 5: Mapping of allowed NSSAIs to configured NSSAIs for data forwarding.

The edge controller EC conducts matching according to the locally cached HT and Hidden Allowed $S-NSSAI$. It also selects Slice i specified by the user for subsequent data packet transmission. Simultaneously, the ticket number $TXID_{ST_i}$ is used to query on the blockchain for the sake of acquiring CH_{UE}. Then, $\gamma_{UE} = H_1(PID_{UE} \| A_{UE} \| B_{UE} \| T_{Curr})$ is calculated, and the equation $m_{UE} P + \gamma_{UE} A_{UE} = CH_{UE}$ is verified to check whether it can be established or not. If the equation cannot be established, the user will not be the ticket's legal owner and cannot access the slice service he applied. Furthermore, the EC terminates the protocol interaction. Otherwise, the EC notifies UE to start the key agreement and at the same time sends a message $\langle ACK, PID_{UE}, m_{UE}, A_{UE}, B_{UE}\rangle$ to A3VI of the slice service provider, where $ACK = \{1\}$.

As shown in Figure 6, after the authentication, A3VI selects the parameters $\alpha_{A3VI}, \beta_{A3VI} \in_R \mathbb{Z}_q^*$ and uses its own public and private key pair $pk_{A3VI}, sk_{A3VI} = (Y_{A3VI}, x_{A3VI})$, where $Y_{A3VI} = x_{A3VI} P$, to calculate $A_{A3VI} = \alpha_{A3VI} Y_{A3VI}$ and $B_{A3VI} = \beta_{A3VI} Y_{A3VI}$. After that, the received A_{UE} and B_{UE} is used to calculate $K_{A3VI} = x_{A3VI}(\alpha_{A3VI} + \beta_{A3VI})(A_{UE} + B_{UE})$. Finally, a temporary session key $SK_{A3VI} = H_2(PID_{UE} \| m_{UE} \| K_{A3VI})$ is generated. After generating the temporary session key SK_{A3VI}, A3VI sends a message $\langle A_{A3VI}, B_{A3VI}\rangle$ to the UE. After receiving the message, UE calculates $K_{UE} = x_{UE}(\alpha_{UE} + \beta_{UE})(A_{A3VI} + B_{A3VI})$ and then calculates the temporary session key $SK_{UE} = H_2(PID_{UE} \| m_{UE} \| K_{UE})$. Finally, the user calculates $ACK_{UE} = H_3(K_{UE} \| SK_{UE})$ and sends ACK_{UE} to A3VI. After A3VI receives ACK_{UE}, it uses its own K_{A3VI} and SK_{A3VI} to conduct verification. If the verification is passed, the session key $SK = SK_{UE} = SK_{A3VI}$ will be used for encrypted communication between UE and A3VI.

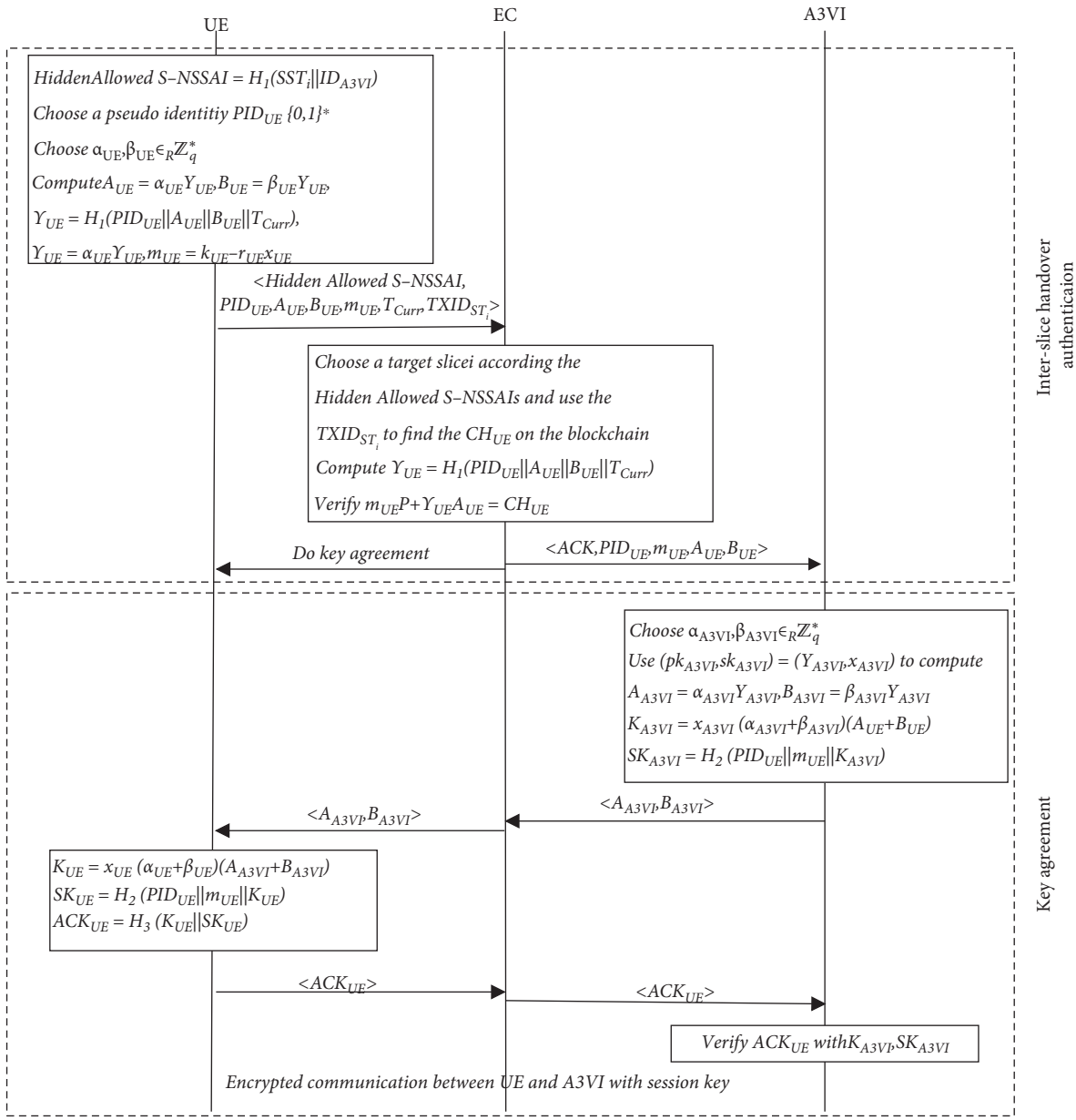

Figure 6: The flow diagram of inter-slice handover authentication with the key agreement.

6. Security Analysis

Here, we first give some preliminaries about the formal model. Then, we implement the core part of the FUIS, including the registration phase, authentication phase, and key agreement phase. We also give our analysis of essential security properties.

6.1. Formal Analysis. Compared with the correctness of a general system, the security of a cryptographic protocol is more subtle because a correct system only needs to consider the correct completion of the expected task. In contrast, besides the expected tasks, designing a secure protocol especially needs to take various attacks into account. At present, there are two main methods for the security analysis of cryptographic protocols. One is based on formal models, and the other is based on computation models. The widely used ProVerif [42], which used pi calculations, is an automated analysis tool under the former method. The advantage of this method is that it is easy to implement an analysis through programming. Here, we briefly describe some basic operations of ProVerif to facilitate our understanding of the security analysis in this section.

query <query>: the statement, which can be written in two forms, tells the system what we want to prove.

Query attacker: **M** means that the attacker can obtain M at a certain stage (noting that M is not a secret value).

query inj $-$**event**: $\mathbf{f}(\mathbf{x}_1, \ldots, \mathbf{x}_n) == > \mathbf{inj} - \mathbf{event}$: $\mathbf{f}'(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ is an injective agreement; when the query is

true, it means when the incident $f(x_1, \ldots, x_n)$ is being executed, the incident $f\prime(x_1, \ldots, x_n)$ has been executed.

! < **proc ess** > means to execute < **process** > unlimitedly , and also, the form <process > | < process > | < process > | $\cdots$ can be used to manifest the execution of ⟨process⟩.

The results of ProVerif are shown in Figures 7–9. Next, several subsections will be subdivided for the specific analysis of the security goals set before.

6.1.1. Inter-Slice Handover Authentication. In ProVerif, a corresponding statement can be used to capture identity verification. In order to prove that the user can complete the authentication after inter-slice handover, **event** AuthStrated (Hidden_S_NSSAI, PID, A, B, *m*, T, TXID) and **event** AuthFinished (PID, *m*, A, B) can be defined. Besides, the following query inj-event (AuthFinished(PID, *m*, A, B)) = = >inj-event(AuthStrated(Hidden_S_NSSAI, PID, A, B, *m*, T, TXID)) can be performed whose result is shown in Figure 7. If the query result is true, which indicates that when the protocol execution ends, A3VI believes that it has indeed completed the interaction with the user UE, the user's authentication to A3VI is established.

6.1.2. Key Agreement with Forward Secrecy. In order to prove that the UE and A3VI successfully establish the session key, **event** KA_A3VI_Finished(A,B,K,SK), **event** KA_UE_Finished(A,B,K,SK), **event** KA_UE_ACK(ACK), and **event** KA_A3VI_ACK_Vefify(ACK) can be defined. And, the following **query** inj-event(KA_A3VI_ACK_Vefify(ACK)) = =>(inj-event(KA_UE_ACK(ACK)) = =>(inj-event(KA_UE_Finished(A,B,K K,SK)) = =>inj -event(KA_A3VI_Finished(A,B,K,SK)))) is processed. The result of the query is shown in Figure 8, which manifests that the session key is successfully established between the UE and A3VI.

In order to illustrate further, the key agreement process has two security features: perfect forward secrecy and master-key forward secrecy. **Phase** in ProVerif is used to leak the master keys of UE and A3VI deliberately because there are random numbers $\alpha_{\mathrm{UE}}, \beta_{\mathrm{UE}}, \alpha_{\mathrm{A3VI}}$, and β_{A3VI} in the session key agreement material K_{UE} and K_{A3VI} calculation process in Section 5.2.3. The result in Figure 9 shows that even if the master keys of UE and A3VI are leaked, the forward security of the session key can be guaranteed.

6.1.3. Key Randomness Secrecy. Similarly, to show that the key agreement process has the known randomness secrecy security feature, $\alpha_{\mathrm{UE}}, \beta_{\mathrm{UE}}, \alpha_{\mathrm{A3VI}}$, and β_{A3VI} are leaked deliberately. The result is consistent with that shown in Figure 9. The attacker still cannot obtain the session key agreement material K_{UE} and K_{A3VI} and the session keys $\mathrm{SK}_{\mathrm{UE}}$ and $\mathrm{SK}_{\mathrm{A3VI}}$.

6.2. Informal Analysis

6.2.1. Identity Anonymity in Network Slices. Although the user uses his identity information in the registration stage,

```
--Query inj-event(AuthFinished(PID,m_3479,A_3477,B_3478)) = = > inj-ev
ent(AuthStrated(Hidden_S_NSSAI_3476,PID,A_3477,B_3478,m_3479,T,TXID
_3480))
Completing...

200 rules inserted. The rule base contains 194 rules. 9 rules in the queue.

Starting query inj-event(AuthFinished(PID,m_3479,A_3477,B_3478)) = = >
int-event(AuthStrated(Hidden_S_NSSAI_3476,PID,A_3477,B_3478,m_3479,
T,TXID_3480))
RESULT inj-event(AuthFinished(PID,m_3479,A_3477,B_3478)) = = >
int-event(AuthStrated(Hidden_S_NSSAI_3476,PID,A_3477,B_3478,
m_3479,T,TXID_3480)) is true.
```

Figure 7: Authentication results in ProVerif.

```
--Query inj-event(KA_A3VI_ACK_Verify(ACK))= = > inj-event(KA_UE_ACK
(ACK)) = = > (inj-event(KA_UE_Finished(A,B,K,SK)) = = > inj-event(KA_A3VI_
Finished(A,B,K,SK))))
Completng...
200 rules inserted. The rule base contains 196 rules. 9 rules in the queue.
Starting query inj-event(KA_A3VI_ACK_Verify(ACK)) = = > (inj-event(KA_U
E_ACK_(ACK)) = = > (inj-event(KA_UE_Finished(A,B,K,SK)) = = > inj-event(KA
_A3VI_Finished(A,B,K,SK))))
RESULT inj-event(KA_A3VI_ACK_Verify(ACK)) = = > ((inj-event(KA_UE_ACK
(ACK)) = = > inj-event(KA_UE_Finished(A,B,K,SK)) = = > inj-event(KA_A3VI
_Finished(A,B,K,SK)))) is true.
```

Figure 8: Key agreement results in ProVerif.

```
RESULT not attacker_ID(ID_UE[]) is true.
RESULT not attacker_nonce(N[]) is true.
RESULT not attacker_point(CH_UE[]) is true.
RESULT not attacker_bitstring(SK_A3VI[]) is true.
RESULT not attacker_point(K_A3VI[]) is true.
RESULT not attacker_bitstring(SK_UE[]) is true.
RESULT not attacker_point(K_UE[]) is true.
```

Figure 9: Testification results of forward security.

the generated anonymous authentication ticket does not contain any user's personal information. Furthermore, because of the encryption protection in the registration stage, as shown in Figure 9, the attack cannot be performed in the open channel and cannot process eavesdropping on the user's $\mathrm{ID}_{\mathrm{UE}}$ and $\mathrm{CH}_{\mathrm{UE}}$ and session value N_i.

The anonymity of the user in the authentication phase is guaranteed from two perspectives. The first perspective is that the user presents a pseudonym $\mathrm{PID}_{\mathrm{UE}}$ during authenticating, which has nothing to do with the user's identity $\mathrm{ID}_{\mathrm{UE}}$. Next, when A3VI publishes data to the chain, it only stores $(\mathrm{CH}_{\mathrm{UE}}, T_{\mathrm{Exp}})$ without revealing any information related to the identity $\mathrm{ID}_{\mathrm{UE}}$. The second perspective is that when a user applies for a ticket, the operator and slice service provider A3VI both use the ring signature method to generate the authorization ticket, so the miners cannot know the ticket belongs to which operator and slice service provider A3VI when verifying the legitimacy of the ticket, which ensures the anonymity of the ticket type and, meanwhile, ensures the service privacy of the user when using the ticket.

6.2.2. Traceability. Assuming that the user commits malicious behavior when using the pseudonym $\mathrm{PID}_{\mathrm{UE}}$, the operator can use the information <Hidden Allowed

$S-\text{NSSAI}, \text{PID}_{\text{UE}}, A_{\text{UE}}, B_{\text{UE}}, m_{\text{UE}}, T_{\text{Curr}}, \text{TXID}_{\text{ST}_i}>$ and combine the following methods to track the user's real identity. Firstly, A3VI calculates $\gamma_{\text{UE}} = H_1(\text{PID}_{\text{UE}} \| A_{\text{UE}} \| B_{\text{UE}} \| T_{\text{Curr}})$ and uses γ_{UE} to calculate $\text{CH}_{\text{UE}} = m_{\text{UE}}P + \gamma_{\text{UE}}A_{\text{UE}}$. Then, $\text{TXID}_{\text{ST}_i}$ is used to query CH_{UE} cached on the chain. Besides, $\text{TXID}_{\text{ST}_i}$ is used in the operator's local database to ensure the same value as the locally calculated CH_{UE}. After that, the query result $\text{ID}_{\text{UE}} \| \text{ID}_{\text{A3VI}} \| \text{TXID}_{\text{ST}_i}$ can be acquired, and ID_{UE} can be output.

6.2.3. Key Escrow Freeness. From the introduction in Section 5.2.2, it can be known that the users' private key x_{UE} is completely selected by themselves. Therefore, the FUIS is a key escrow-free inter-slice handover authentication protocol.

7. Performance Evaluation

In this section, the computation overhead and communication overhead of the FUIS protocol will be analyzed and tested to illustrate its performance in a specific implementation. Besides, NS3-5G-LENA will be used to conduct a more comprehensive analysis of the FUIS time delay during inter-slice handover. Finally, the EC side and AUSF side's storage overhead will be analyzed to further prove the scheme's feasibility. Note that we leave the storage overhead of the FUIS in Appendix B.

7.1. Computation Overhead. Before the start of the experiment, it can be assumed that the public key certificates of UE, operator, and A3VI have been exchanged. We simulate the FUIS protocol and record the computation time and running time of each stage. UE, EC, A3VI, and operators all run on a desktop computer. The configuration of this computer is Intel® Core™ i7-8700 CPU @ 3.20 GHz and 16 GB memory. The computer's operating system is 64 bit Windows 10, the C++ compiler with Visual Studio 2019, and the version of Python is 3.7. Two libraries called PyCryptodome3.9.8 and sslcrypto5.3 are principally used to implement cryptography primitives. The elliptic curve adopted is secp256r1. The detailed computation overhead of the FUIS is shown in Appendix A. And, we mainly describe the computation overhead comparison below.

For illustrating the advantages of the FUIS in handover between slices, the FUIS with the known anonymous authentication protocols ES^3A [16, 43, 44], CPAL, and LCCH are compared under the same settings. The computation overhead of the user side and authentication side (authentication side of our scheme is the EC, the key agreement is completed by A3VI, and the authentication side of the rest schemes is the AAA server) is firstly compared in the authentication and key agreement phase. As shown in Figure 10(a), the computation overhead increases linearly with the increase of users' number. It can be easily observed that the FUIS shows an obvious advantage in comparison. The entire computation overhead of the FUIS on the user side is still maintained at a low level when the number of users reaches 100. Figure 10(b) shows the computation overhead of the authentication side in the authentication and key agreement phase. Similar to the user side, compared to other schemes, the FUIS also shows a lower computation overhead on the authentication side.

Figures 10(a) and 10(b) show that the overheads of the FUIS on the user side and that on the EC side are similar. This is because the ECC-based chameleon hash function is used once in the authentication phase, and ECC-based scalar multiplication is used twice in the key agreement phase. The scalar multiplication operation of the chameleon hash function is equivalent to 1 multiscalar multiplication. Therefore, the overheads on the user side and EC side are almost identical.

To further analyze why the FUIS is superior to the other three compared schemes, Table 2 is drawn based on time-consuming cryptographic primitives' statistics. Before analysis, it should be emphasized that the bilinear pairing operation is very time consuming. As shown in Table 2, the FUIS does not have any pairing operations. On the contrary, ES^3A, CPAL, and LCCH contain several pairing operations on both the user side and authentication side. Therefore, from the cryptographic primitive statistics results, the solution has obvious advantages in terms of computation overhead.

In addition, to show that FUIS is suitable for some IoT devices with low computing capability, different IoT devices are stimulated for testing by changing CPU frequency. Although the computing capability is not entirely determined by CPU frequency, the CPU frequency is indeed an important factor affecting computing capability. In the test, the following CPU frequencies are selected to perform UE operations, including 160 MHz, 480 MHz, 640 MHz, 800 MHz, 960 MHz, 1280 MHz, and 1600 MHz. The reason why these frequencies are chosen is that these frequencies can cover most of the IoT devices in the market. Figure 10(c) shows the computation overhead of user authentication and key agreement under different CPU frequencies. It can be seen from the figure that the CPU frequency change has little effect on the user's authentication and key agreement process, and the computation overhead remains relatively low. Only devices with very low computing capability increase the user's computation overhead during the registration phase.

7.2. Communication Overhead. In this section, the communication overhead of the FUIS is calculated in detail. In the service registration phase, the user will send the message $\langle U\text{Text}, E_1, \sigma \rangle$ to the AMF, which requires 463 bytes, and the AUSF will send the message $< C\text{Text}, E_2, \beta >$ to A3VI, which requires 945 bytes, and A3VI will send the message $\langle \text{TXID}_{Tx}, T_{\text{Exp}} \rangle$ to AUSF and UE, respectively, for which a total of 138 bytes are needed. In the inter-slice handover authentication phase, user UE needs to send the message $\langle \text{Hidden Allowed } S-\text{NSSAI}, \text{PID}_{\text{UE}}, A_{\text{UE}}, B_{\text{UE}}, m_{\text{UE}}, T_{\text{Curr}}, \text{TXID}_{\text{ST}_i} \rangle$ to the edge controller EC, which requires 308 bytes. The EC needs to send the message $\langle \text{ACK}, \text{PID}_{\text{UE}}, m_{\text{UE}}, A_{\text{UE}}, B_{\text{UE}} \rangle$ to the slice service provider's A3VI, which needs 180 bytes. In the key agreement phase, A3VI needs to

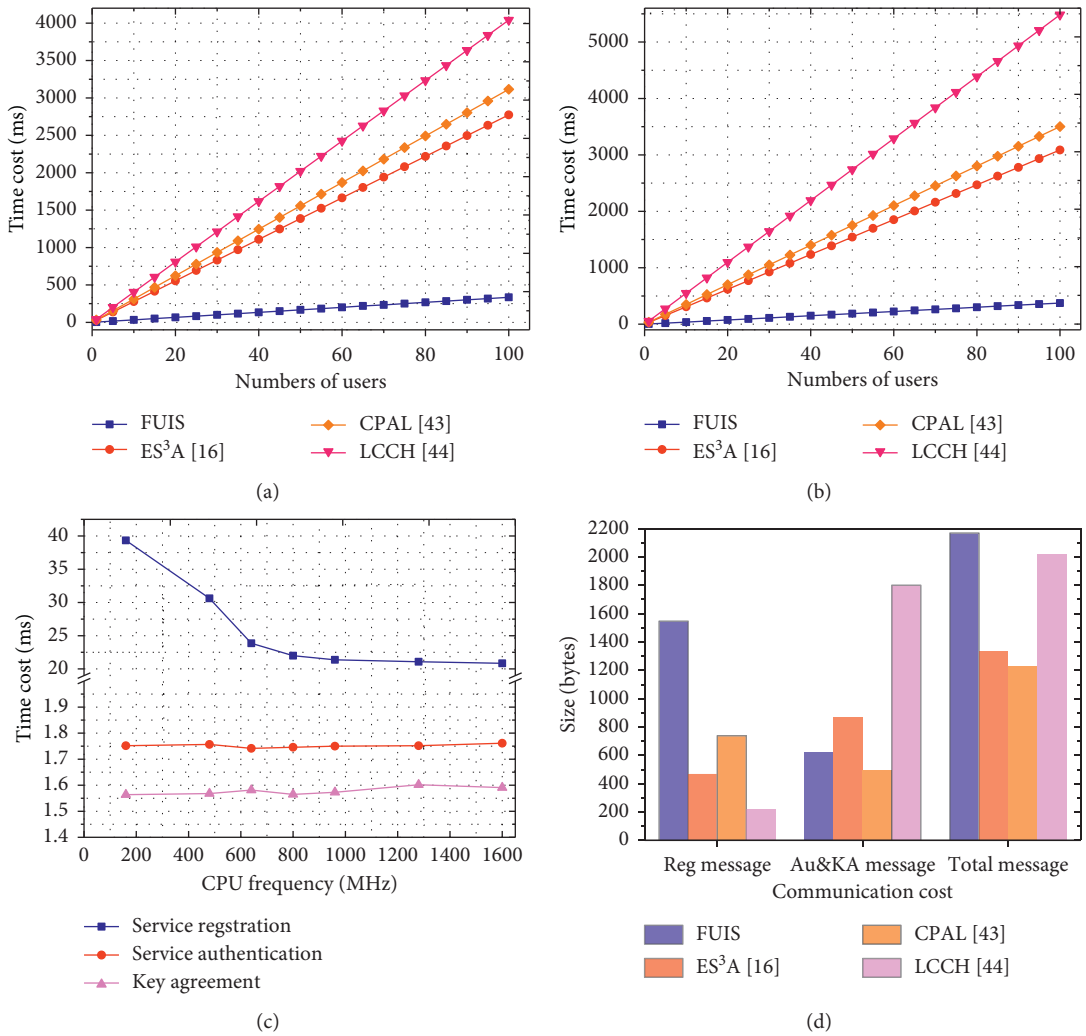

Figure 10: Comparison of computation and communication overhead (Au: authentication, KA: key agreement, and Reg: registration). (a) Cost on users for Au and KA. (b) Cost on the EC for Au and KA. (c) Cost on users with different devices. (d) Communication cost.

Table 2: Comparison of time-consuming cryptographic primitives in Au and KA.

Scheme	User	Auth-side
FUIS	3SM	SM + MSM
ES^3A [16]	$6\text{Exp}_1 + 3\text{Exp}_2 + 2\text{Exp}_T + 7\text{BP}$	$10\text{Exp}_1 + 5\text{Exp}_2 + 4\text{Exp}_T + 8\text{BP}$
CPAL [46]	$18\text{Exp}_1 + 7\text{Exp}_T + 7\text{BP}$	$17\text{Exp}_1 + 4\text{Exp}_T + 7\text{BP}$
LCCH [47]	$27\text{Exp}_1 + \text{Exp}_2 + 9\text{Exp}_T + 9\text{BP}$	$23\text{Exp}_1 + 11\text{Exp}_T + 13\text{BP}$

Notes: in schemes of ES^3A, CPAL, and LCCH, SM represents scalar multiplication, MSM represents multiscalar multiplication elliptic curve, F_p – 256BN is set in experiment, $\text{Exp}_1, \text{Exp}_2$, and Exp_T represent modular exponentiation under cyclic groups G_1, G_2, and G_T, and BP represents bilinear mapping.

send the message $\langle A_{\text{A3VI}}, B_{\text{A3VI}} \rangle$ to the UE, which requires 128 bytes, and the user sends the message ACK_{UE} to A3VI, which requires 8 bytes. Consequently, the communication overhead of the FUIS requires a total of 2170 bytes.

Similarly, the FUIS is compared with the schemes ES^3A, CPAL, and LCCH. Generally speaking, comparing with ES^3A's 1336 bytes, CPAL's 1232 bytes, and LCCH's 2016 bytes, this solution has the largest communication overhead. However, the communication overhead is mainly concentrated on the registration phase. This is because ring signatures have been introduced in the registration phase. Nevertheless, the communication overhead in the authentication phase and key agreement phase is better than that of the ES^3A and LCCH schemes, as shown in Figure 10(d). We also show the communication overhead under different ring sizes in Appendix C.

7.3. Overall Handover Delay. To further test our proposed protocol, we simulate the protocol based on NS3. The network topology is shown in Figure 2 in Section 4.1.

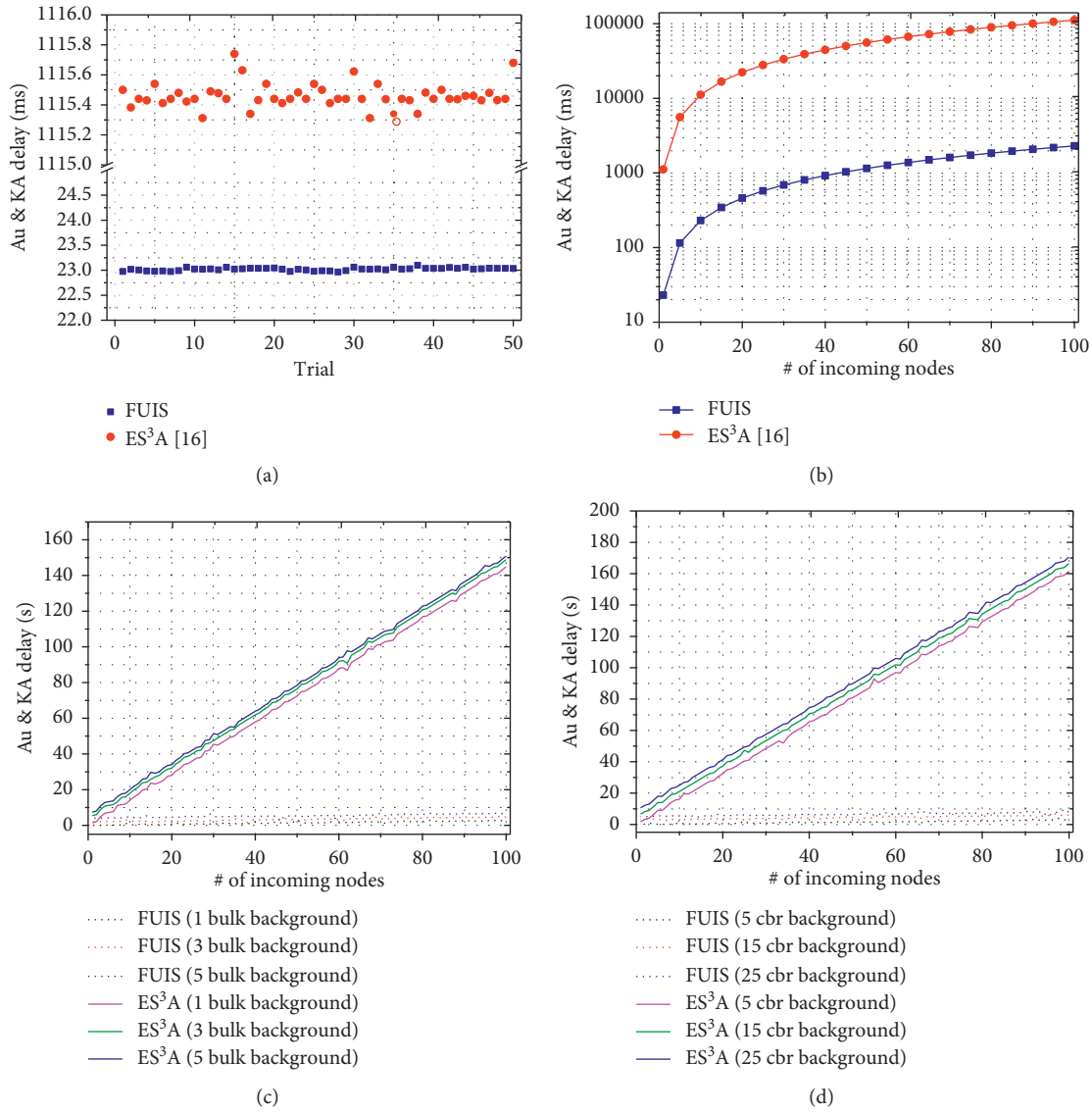

FIGURE 11: The delay comparison of Au and KA between FUIS and ES^3*A*. (a) Delay simulation in 50 times. (b) Delay simulation with 100 users. (c) Delay comparison with bulk transmissions' background traffic. (d) Delay comparison with CBR transmissions' background traffic.

Considering that the scheme ES^3*A* and scheme FUIS are the same type of the authentication protocol, we only test these two schemes in this section.

The basic configuration in NS3 is shown as follows: in the wireless part, the channel frequency is 28Ghz, channel bandwidth is 100 M, and numerology = 4. The data transmission rate configured in the wired part is 100 Gb/s, the MTU is 2500, and the channel delay is 0.0005 s.

We run ES^3*A* and FUIS for 50 times, respectively. The communication delay of each scheme during the inter-slice handover (that is, the authentication phase and key agreement phase) is recorded (here, the time when the terminal scans nearby base stations is not recorded). The recorded time delay is shown in Figure 11(a). ES^3*A* spends an average of 1115.461 ms in the inter-slice handover process, and the FUIS spends an average of 23.021 ms in the inter-slice handover process. The experimental results show that the proposed FUIS reduces the overhead by 97.94% in the handover delay.

In addition, the handover delay of ES^3*A* and FUIS with multiuser access is compared in three scenarios. The three scenarios are no background traffic, bulk background traffic (such as web browsing), and CBR (Constants' Bit Rate, CBR) background traffic (such as video services), respectively. In the first case, no background traffic is prescribed. The number of concurrent authentications in the 5G slice changes from 1 to 100, and the value is shown in Figure 11(b) (for a more intuitive comparison, the vertical axis uses the right number scale). Then, bulk data packets and CBR data packets are set as background traffic, and then, ES^3*A* and FUIS are retested with background traffic. When it has bulk background traffic, the number of bulks is set as 1, 3, and

5. Similarly, the number of concurrent authentications in the 5G slice will change from 1 to 100, and the value is shown in Figure 11(c). Finally, some terminals are set up to access the channel for CBR transmission. Similarly, ES^3A and FUIS are retested, and the value is shown in Figure 11(d).

As shown in Figures 11(b)–11(d), the FUIS has obvious advantages over ES^3A in terms of handover delay. This conclusion is consistent with the conclusion in Section 7.1. This situation is more obvious when the number of terminals increases. It can be seen from the experimental data obtained before. When the number of terminals is 50, the handover delay of ES^3A is 55.77 seconds without background traffic; in the case of 1 bulk background traffic, the handover delay of ES^3A is 72.42 seconds; in the case of 5 CBR background traffic, the handover delay of ES^3A is 80.76 seconds. Under the same circumstances, the handover delay of the FUIS is only 1.15 seconds, 1.33 seconds, and 1.43 seconds, respectively. It can be assumed that a car crosses a slice at a speed of 45 km/h, and the overlapping buffer interval between slices is 100 meters. Then, the time left for the car-machine system to complete the handover between slices is only 8 s. Therefore, in comparing the two solutions, only the FUIS can meet the needs of such fast handover.

Next, we further analyze the experimental results obtained in Figures 11(b)–11(d). It is easy to know that the handover delay shows a linear growth trend. Regarding this, then the composition of the handover delay can be analyzed as follows. The handover delay consists of two parts: communication delay and computation delay. Communication delay includes propagation delay and transmission delay. In the topology mentioned before, there are both wireless access networks and wired networks. Among them, there is the channel competition in the wireless network, and the channel competition will take some time. Considering the wired network's huge bandwidth, the transmission delay of the wired network is almost negligible. The computation delay includes calculation time, packet encapsulation, and de-encapsulation time. This part has been analyzed in detail in Section 7.1.

8. Conclusions

In this paper, we proposed an authentication framework called the FUIS based on chameleon hashing, ring signature, and blockchain technologies towards different inter-slice handover scenes. Under this framework, an inter-slice handover authentication protocol is introduced to achieve a fast and universal handover for users between slices. Specifically speaking, in the registration phase, users send the results calculated by the chameleon hash function to operators. Operators and slice providers will generate the tickets by applying the ring signature based on the hash value; finally, the tickets will be saved in blockchains. The verification computation of tickets produced in the registration phase is transferred to the consensus phase in advance through handover authentication models. Furthermore, the authentication service provided by the AAA server is transferred to the edge controller for helping users to verify that they are legitimate owners of the tickets on the chain and finishing authentication. In order to achieve this goal, users should interact with the nearest edge controllers and use their trapdoor to compute the collision of chameleon hash when they undergo inter-slice handover among slices. The results show that the protocol proposed in this work has great computation overhead performance; it can also achieve fast authentication when user handover takes place among slices. However, the communication overhead in the experiment keeps medium; this is because the ring signature, a time-consuming primitive tool, is applied in our protocol design [45–47].

Appendix

A. The Detailed Computation Overhead of the FUIS

The computation overhead of the FUIS will determine the performance of the protocol in actual situations. In order to evaluate the experimental overhead, the time-consuming cryptographic primitive operations at each stage of the protocol will be counted, including scalar multiplication in $\mathbb{G}$, AES encryption, and decryption. Please note that point addition, integer multiplication, and hash operations will not be evaluated because these operations are not resource-consuming cryptographic primitives compared to the first two operations. SM stands for scalar multiplication, MSM stands for multiscalar multiplication, and AES stands for AES encryption or decryption. After calculation, the numbers of time-consuming cryptographic operations of each stage are counted in Table 3, among which every operation is based on ECC. The generation of the ECC public and private key requires 1SM, ECDSA signature requires 1SM, ECDSA verification requires 1MSM, calculation of the chameleon hash function value requires 1MSM, ECC encryption requires 1SM, ECC decryption requires 1SM, and ring signature requires $(4n-2)$SM, where $n \geq 1$. Taking the registration of A3VI in the slice service as an example, A3VI requires one signature verification, one ECC decryption, one AES decryption, and one ring signature. Therefore, the computation overhead of A3VI is $3\text{SM} + \text{AES} + (3n-2)\text{MSM}$, where n is the number of ring participants.

According to the settings in Section 7.1, after the simulation experiment, every entity's computation overheads in FUIS protocols in different phases are shown in Table 4, where $n = 10$, which means the number of the ring signature is 10. Based on the results, it can be known that the computation overheads are acceptable for mobile phones and general devices of the Internet of Things. Besides, for the sake of saving time, the users can calculate $A_{\text{UE}}, B_{\text{UE}}, A_{\text{A3VI}}$, and B_{A3VI} in advance. Furthermore, the ring members RG_i^{Ope} and $\text{RG}_i^{\text{A3VI}\oplus}$ can also be chosen by operator and A3VI in advance to promote the performances of the FUIS in registration and authentication phases further.

Table 3: Computation overhead of the FUIS.

Phase	UE	EC	Operator	A3VI
Initialization of the system	SM	—	SM	SM
Registration of the slice service	3SM + 1MSM + AES	—	4SM + 2AES + $(3n-1)$MSM	3SM + AES + $(3n-2)$MSM
Inter-slice handover authentication	2SM	MSM	—	—
Key agreement	SM	—	—	3SM

Table 4: Execution time of the FUIS (ms).

Phase	UE	EC	Operator	A3VI
Initialization of the system	1.342	—	1.097	1.085
Registration of the slice service	19.549	—	987.767	975.2699
Inter-slice handover authentication	1.7585	2.029	—	—
Key agreement	1.5876	—	—	3.6951

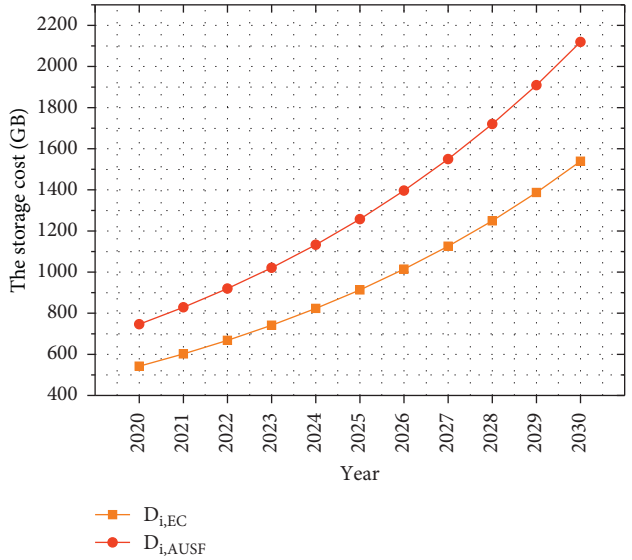

Figure 12: The storage cost of the EC and AUSF within the next decade (the total number of UEs worldwide increases from 7.6 billion to 24.1 billion).

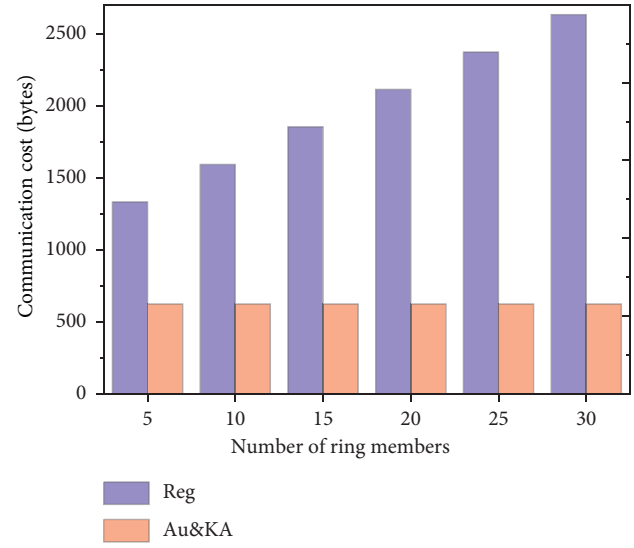

Figure 13: Communication Overhead of Reg, Au, and KA under different ring sizes.

B. The Storage Overhead of the FUIS

Based on the introduction in Section 5.2.2, it can be known that the edge controller will cache the blockchain data, and the AUSF will back up the mapping relationship between the ticket, slicing service provider, and user's real identity. Specifically speaking, the edge controller will cache the transaction data OP_RETURN after A3VI is uploaded. The storage overhead of the EC and AUSF is calculated here. Taking the third Bitcoin reward halving on May 12, 2020, as an example, there are currently $N_b = 630000$ blocks, and each block head is $b_{\text{header}} = 80$ bytes. Because of the difficulty of mining automatically controlled by Bitcoin, one block can be produced approximately every 10 minutes, so $R_b = 52560$ blocks will be supplemented every year.

Based on the test in Section 7.2, the size of an EC's data $\text{TXID}_{\text{ST}_i}, T_{\text{Exp}}$ to be recorded on the blockchain is $d_{\text{EC}} = 69$ bytes. The size of a AUSF's data to back up on the operator's core network is $d_{\text{AUSF}} = 95$ bytes.

According to a report published in 2020 [48], as of the end of 2019, the number of activated IoT devices in the world is $N_{IoT} = 7.6$ billion units, and it is growing at a compound annual growth rate of $R_{IoT} = 11\%$. Then, it can be computed that, after i years, the amount of data that the edge controller EC and AUSF need to store is approximately as follows:

$$D_{i,\text{EC}} = b_{\text{header}}\left(N_b + R_b \times i\right) + d_{\text{EC}} \times N_{IoT}\left(1 + R_{IoT}\right)^i, \tag{B.1}$$

$$D_{i,\text{AUSF}} = b_{\text{header}}\left(N_b + R_b \times i\right) + d_{\text{AUSF}} \times N_{IoT}\left(1 + R_{IoT}\right)^i. \tag{B.2}$$

As shown in Figure 12, the number of users will reach 24.1 billion in ten years, and the storage on the EC side will reach 1539.363 GB. With the rapid development of storage technology and the popularity of IoT devices, more and

more IoT devices will have stronger storage capabilities such as mobile phones. As for the storage of the AUSF is centralized, the storage capacity of the AUSF should be qualified. In addition, the AUSF can also clean up expired storage data according to T_{Exp} to optimize its storage overhead.

C. The Communication Overhead under Different Ring Sizes

Due to the characteristics of ring signatures, the communication overhead of the FUIS increases. As shown in Figure 13, the communication overhead in the registration phase will increase with the growth of ring members.

Data Availability

The complete code and data are published at https://github.com/JK211/FUIS. The project contains protocol code, related test code, and experiment data, which can be cloned directly to the local to reproduce the protocol conveniently. The code in nr/example in 5G-LENA is principally referred to simulate the delay for Section 7.3. Since 5G-LENA is not a fully open source yet, this part of the code is not disclosed. About the use of 5G-LENA, refer to https://5g-lena.cttc.es/download/.

Conflicts of Interest

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

Acknowledgments

The authors would like to thank Dr. Mengfan Xu, Dr. Jiao Liu, Dr. Yunwei Wang, and Dr Yanbing Ren in the same lab for their guidance and help during the preparation of the manuscript, Dr. Xiaohan Zhang from the same experimental group for his careful guidance in the experimental part, and Ziyang Zhang from Northwestern Polytechnical University for his guidance and help in paper writing. The authors also acknowledge the support of the project "The Verification Platform of Multi-tier Coverage Communication Network for Oceans" (LZC0020) and Guangxi Key Laboratory of Trusted Software (program no. KX202035). This work was funded in part by the National Natural Science Foundation of China (Grant nos. U1708262, U1736203, 61872449, and 62072352).

References

[1] X. Foukas, G. Patounas, A. Elmokashfi, and M. K. Marina, "Network slicing in 5G: survey and challenges," *IEEE Communications Magazine*, vol. 55, no. 5, pp. 94–100, 2017.

[2] X. Li, M. Samaka, H. A. Chan et al., "Network slicing for 5G: challenges and opportunities," *IEEE Internet Computing*, vol. 21, no. 5, pp. 20–27, 2017.

[3] H. Zhang, N. Liu, X. Chu, K. Long, A.-H. Aghvami, and V. C. M. Leung, "Network slicing based 5G and future mobile networks: mobility, resource management, and challenges," *IEEE Communications Magazine*, vol. 55, no. 8, pp. 138–145, 2017.

[4] K. Campbell, The 5G Economy, https://www.qualcomm.com/media/documents/files/ihs-5g-economic-impact-study-2019.pdf, 2020.

[5] Huawei, 5G. S.-Guaranteed Network Slicing White Paper, https://www-file.huawei.com/-/media/corporate/pdf/white%20paper/5g-service-guaranteed-network-slicing-whitepaper.pdf?la=zh, 2020.

[6] Network Slicing Market, 2020, https://www.marketsandmarkets.com/Market-Reports/network-slicing-market-120515704.html.

[7] M. M. Sajjad, C. J. Bernardos, D. Jayalath et al., "Inter-slice mobility management in 5g: motivations, standard principles, challenges and work directions," https://arxiv.org/abs/2003.11343.

[8] H. Yang, S. Bae, M. Son et al., "Hiding in plain signal: physical signal overshadowing attack on LTE," in *Proceedings of the 28th USENIX Security Symposium (USENIX)*, pp. 55–72, 2019.

[9] S. Behrad, E. Bertin, S. Tuffin et al., "5G-SSAAC: Slice-specific Authentication and Access Control in 5G," in *Proceedings of the IEEE Conference on Network Softwarization (NetSoft)*, pp. 281–285, IEEE, Paris, France, June 2019.

[10] D. Rupprecht, K. Kohls, T. Holz et al., "Breaking LTE on layer two," in *Proceedings of the IEEE Symposium on Security & Privacy (S&P)*, San Francisco, CA, USA, May 2019.

[11] D. Rupprecht, K. Kohls, T. Holz et al., "IMP4GT: IMPersonation attacks in 4G NeTworks," in *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, San Francisco, CA, USA, February 2020.

[12] X. Duan and X. Wang, "Authentication handover and privacy protection in 5G hetnets using software-defined networking," *IEEE Communications Magazine*, vol. 53, no. 4, pp. 28–35, 2015.

[13] C.-I. Fan, "ReHand: secure region-based fast handover with user anonymity for small cell networks in mobile communications," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 927–942, 2019.

[14] Q. Jiang, N. Zhang, J. Ni et al., "Unified biometric privacy preserving three-factor authentication and key agreement for cloud-assisted autonomous vehicles," *IEEE Transactions on Vehicular Technology*, vol. 69, pp. 9390–9401, 2020.

[15] Y. Zhang, R. Deng, E. Bertino et al., "Robust and universal seamless handover authentication in 5G HetNets," *IEEE Transactions on Dependable Secure Computing*, 2019, In press.

[16] J. Ni, X. Lin, and X. S. Shen, "Efficient and secure service-oriented authentication supporting network slicing for 5G-enabled IoT," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 3, pp. 644–657, 2018.

[17] Y. Zhang, J. Li, D. Zheng, P. Li, and Y. Tian, "Privacy-preserving communication and power injection over vehicle networks and 5G smart grid slice," *Journal of Network and Computer Applications*, vol. 122, pp. 50–60, 2018.

[18] I. A. Kamil and S. O. Ogundoyin, "A big data anonymous batch verification scheme with conditional privacy preservation for power injection over vehicular network and 5G smart grid slice," *Sustainable Energy, Grids and Networks*, vol. 20, Article ID 100260, 2019.

[19] D. He, C. Chen, S. Chan et al., "Secure and efficient handover authentication based on bilinear pairing functions," *IEEE Transactions on Wireless Communications*, vol. 11, no. 1, pp. 48–53, 2011.

[20] J. Choi and S. J. I. C. L. Jung, "A handover authentication using credentials based on chameleon hashing," *IEEE Communications Letters*, vol. 14, no. 1, pp. 54–56, 2009.

[21] L. Cai, S. Machiraju, and H. Chen, "Capauth: a capability-based handover scheme," in *Proceedings of the IEEE INFOCOM*, pp. 1–5, IEEE, San Diego, CA, USA, March 2010.

[22] Z. Haddad, M. Mahmoud, I. A. Saroit et al., "Secure and efficient uniform handover scheme for LTE-A networks," in *Proceedings of the 2016 IEEE Wireless Communications and Networking Conference(WCNC)*, pp. 1–6, IEEE, Doha, Qatar, April 2016.

[23] V. G. Vassilakis, H. Mouratidis, E. Panaousis et al., "Security Requirements Modelling for Virtualized 5G Small Cell Networks," in *Proceedings of the 2017 24th International Conference on Telecommunications (ICT)*, pp. 1–5, IEEE, Limassol, Cyprus, May 2017.

[24] H. Lee and M. d. Ma, "Blockchain-based mobility management for 5G," *Future Generation Computer Systems*, vol. 110, pp. 638–646, 2019.

[25] A. Yazdinejad, R. M. Parizi, A. Dehghantanha et al., "Blockchain-enabled authentication handover with efficient privacy protection in SDN-based 5G networks," *IEEE Transactions on Network Science Engineering*, 2019, In press.

[26] R. Lu, L. Zhang, J. Ni et al., "5G vehicle-to-everything services: gearing up for security and privacy," *Proceedings of the IEEE*, vol. 108, no. 2, pp. 373–389, 2019.

[27] V. N. Sathi, M. Srinivasan, P. K. Thiruvasagam et al., "A novel protocol for securing network slice component association and slice isolation in 5G networks," in *Proceedings of the 21st ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems(MSWiM)*, pp. 249–253, 2018.

[28] J. Liu, L. Zhang, R. Sun, X. Du, and M. Guizani, "Mutual heterogeneous signcryption schemes for 5G network slicings," *IEEE Access*, vol. 6, pp. 7854–7863, 2018.

[29] S. Behrad, E. Bertin, S. Tuffin, and N. Crespi, "A new scalable authentication and access control mechanism for 5G-based IoT," *Future Generation Computer Systems*, vol. 108, pp. 46–61, 2020.

[30] V. N. Sathi, M. Srinivasan, P. K. Thiruvasagam et al., "Novel protocols to mitigate network slice topology learning attacks and protect privacy of users' service access behavior in softwarized 5G networks," *IEEE Transactions on Dependable Secure Computing*, p. 1. In press, 2020.

[31] S. Nakamoto, *Bitcoin: A Peer-To-Peer Electronic Cash System*, 2019.

[32] H. Krawczyk and T. Rabin, *Chameleon Hashing and Signatures*, 1998.

[33] R. L. Rivest, A. Shamir, and Y. Tauman, "How to leak a secret," in *Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security*, pp. 552–565, Springer, Gold Coast, Australia, December 2001.

[34] X. Li, Y. Mei, J. Gong, F. Xiang, and Z. Sun, "A blockchain privacy protection scheme based on ring signature," *IEEE Access*, vol. 8, pp. 76765–76772, 2020.

[35] D. Basin, J. Dreier, L. Hirschi et al., "A formal analysis of 5G authentication," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security(CCS)*, pp. 1383–1396, ACM, Toronto, Canada, October 2018.

[36] S. R. Hussain, M. Echeverria, I. Karim et al., "5GReasoner: a property-directed security and privacy analysis framework for 5G cellular network protocol," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security(CCS)*, pp. 669–684, ACM, London, UK, November 2019.

[37] A. Koutsos, "The 5G-AKA authentication protocol privacy," in *Proceedings of the 2019 IEEE European Symposium on Security and Privacy (EuroS&P)*, pp. 464–479, IEEE, Stockholm, Sweden, June 2019.

[38] A. Shaik, R. Borgaonkar, S. Park et al., "New vulnerabilities in 4G and 5G cellular access network protocols: exposing device capabilities," in *Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*, pp. 221–231, ACM, Miami, FL, USA, May 2019.

[39] C. Cremers and M. Dehnel-Wild, "Component-based formal analysis of 5G-AKA: channel assumptions and session confusion," in *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, USA, February 2019.

[40] S. R. Hussain, M. Echeverria, O. Chowdhury et al., "Privacy attacks to the 4G and 5G cellular paging protocols using side channel information," in *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, USA, February 2019.

[41] 3GPP, *3GPP T. S. 33.501, Security Architecture and Procedures for 5G system (Release 16)*, 3rd Generation Partnership Project, 2020.

[42] B. Blanchet, "Automatic verification of correspondences for security protocols*," *Journal of Computer Security*, vol. 17, no. 4, pp. 363–434, 2009.

[43] C. Lai, H. Li, X. Liang, R. Lu, K. Zhang, and X. Shen, "CPAL: a conditional privacy-preserving authentication with access linkability for roaming service," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 46–57, 2014.

[44] J. K. Liu, C.-K. Chu, S. S. Chow et al., "Time-bound anonymous authentication for roaming networks," *IEEE Transactions on Information Forensics*, vol. 10, pp. 178–189, 2014.

[45] R. A. Addad, T. Taleb, H. Flinck, M. Bagaa, and D. Dutra, "Network slice mobility in next generation mobile systems: challenges and potential solutions," *IEEE Network*, vol. 34, no. 1, pp. 84–93, 2020.

[46] F. Z. Yousaf, "Network slicing with flexible mobility and QoS/QoE support for 5G Networks," in *Proceedings of the 2017 IEEE International Conference on Communications Workshops (ICC Workshops)*, pp. 1195–1201, IEEE, Paris, France, May 2017.

[47] J. Cao, H. Li, M. Ma, Y. Zhang, and C. Lai, "A simple and robust handover authentication between HeNB and eNB in LTE networks," *Computer Networks*, vol. 56, no. 8, pp. 2119–2131, 2012.

[48] Global IoT Market Will Grow to 24.1 Billion Devices in 2030, Generating $1.5 Trillion Annual Revenue, 2020, https://transformainsights.com/news/iot-market-24-billion-usd15-trillion-revenue-2030.