4-2021

Robust and universal seamless handover authentication in 5G HetNets

Yinghui ZHANG

Robert H. DENG
Singapore Management University, robertdeng@smu.edu.sg

Elisa BERTINO

Dong ZHENG

Robust and Universal Seamless Handover Authentication in 5G HetNets

Yinghui Zhang, *Member, IEEE*, Robert H. Deng, *Fellow, IEEE*, Elisa Bertino, *Fellow, IEEE*, and Dong Zheng

Abstract—The evolving fifth generation (5G) cellular networks will be a collection of heterogeneous and backward-compatible networks. With the increased heterogeneity and densification of 5G heterogeneous networks (HetNets), it is important to ensure security and efficiency of frequent handovers in 5G wireless roaming environments. However, existing handover authentication mechanisms still have challenging issues, such as anonymity, robust traceability and universality. In this paper, we address these issues by introducing RUSH, a Robust and Universal Seamless Handover authentication protocol for 5G HetNets. In RUSH, anonymous mutual authentication with key agreement is enabled for handovers by exploiting the trapdoor collision property of chameleon hash functions and the tamper-resistance of blockchains. RUSH achieves universal handover authentication for all the diverse mobility scenarios, as exemplified by the handover between 5G new radio and non-3GPP access regardless of the trustworthiness of non-3GPP access and the consistency of the core network. RUSH also achieves perfect forward secrecy, master key forward secrecy, known randomness secrecy, key escrow freeness and robust traceability. Our formal security proofs based on the BAN-logic and formal verification based on AVISPA indicate that RUSH resists various attacks. Comprehensive performance evaluation and comparisons show that RUSH outperforms other schemes in both computation and communication efficiencies.

Index Terms—5G, LTE, handover authentication, blockchain, chameleon hashing, BAN logic, AVISPA

1 Introduction

The explosive growth of wireless data traffic driven by mobile Internet and smart terminals has triggered intensive research of the fifth generation (5G) wireless networks, which are designed to achieve lower cost and more energy-efficiency as well as improved quality of service in terms of communication delay, reliability, and security [1]. In particular, 5G will be backward-compatible tying any new air interface with the existing solutions to provide better user experiences [2] and will be a collection of highly flexible and multi-tier heterogeneous networks (HetNets). The 5G HetNets consist of densified small cell deployment and overlay coverage through 5G new radio access networks (RANs) and coexisting networks [3], such as Long Term Evolution (LTE), LTE-Advanced (LTE-A) specified by Third Generation Partnership Project (3GPP), and other non-3GPP radio access networks including wireless local area network (WLAN), code division multiple access (CDMA) 2000, and worldwide interoperability for microwave access (WiMAX).

Along with the promising benefits, the heterogeneity of 5G HetNets inevitably introduces new challenges. In 5G HetNets, a handover operation involves a user equipment (UE), access points (APs) and authentication servers which typically refer to authentication, authorization, and accounting (AAA) servers specified by the 3GPP committee. For the consistency of description, the authentication server function in a 5G core network (5GC) is also denoted by AAA. When a UE moves from its current AP (a.k.a source AP) to another AP (a.k.a target AP), the UE and the target AP must engage in mutual authentication and key agreement which are the two fundamental security requirements of handover authentication. The concrete form of an AP is determined by the type of the underlying access network. For instance, an AP can be an Evolved Node B (eNB) or a Home eNB (HeNB) in LTE-A, a base station in WiMAX or a WiFi hotspot. In 5G networks, a 5G New Radio NodeB (gNB) can act as an AP. More precisely, there are different tiers in 5G HetNets, in which each tier models base stations of a particular class such as femtocells, picocells, microcells, and macrocells [4]. In 5G HetNets, many different handover authentication scenarios exist which differ in the deployment of authentication servers, the type of access networks and the tier of access points. To simplify the description, as shown in Fig. 1, we directly use APs to represent access networks, in which UEs attach the closest access point to connect to core networks. The scenarios of intra-domain handover and

- *Y. Zhang is with the National Engineering Laboratory for Wireless Security, Xi'an University of Posts & Telecommunications, Xi'an 710121, China, and with the State Key Laboratory of Cryptology, Beijing 100878, China, and also wirh the School of Information Systems, Singapore Management University, Singapore. E-mail: yhzhaang@163.com.*
- *R.H. Deng is with the School of Information Systems, Singapore Management University, Singapore. E-mail: robertdeng@smu.edu.sg.*
- *E. Bertino is with the Department of Computer Science, Purdue University, West Lafayette, IN 47907 USA. E-mail: bertino@purdue.edu.*
- *D. Zheng is with the National Engineering Laboratory for Wireless Security, Xi'an University of Posts & Telecommunications, Xi'an 710121, China, and also with Westone Cryptologic Research Center, Beijing 100070, China. E-mail: zhengdong@xupt.edu.cn.*

Manuscript received 10 May 2018; revised 8 Dec. 2018; accepted 28 June 2019. Date of publication 9 July 2019; date of current version 12 Mar. 2021. (Corresponding author: Yinghui Zhang.)
Digital Object Identifier no. 10.1109/TDSC.2019.2927664

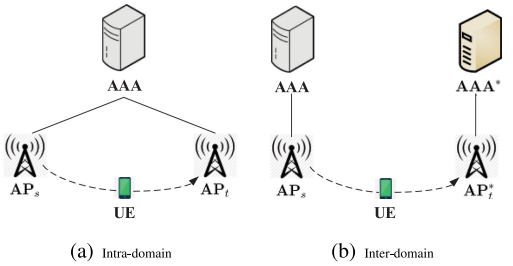


Fig. 1. Intra-domain and inter-domain handover scenarios.

inter-domain handover are illustrated in Figs. 1a and 1b, respectively. In the *intra-domain* handover, a UE moves from the source AP (i.e., AP_s) to the target AP (i.e., AP_t) and the same AAA authentication server is shared by AP_s and AP_t. In the *inter-domain* handover, a UE moves from AP_s managed by one AAA to the target access point AP_t^* managed by another AAA^*. Even in *intra-domain* handover, there exist a variety of handover scenarios because access networks can take diverse forms in 5G HetNets, such as 3GPP LTE-A radio access Evolved Universal Terrestrial Radio Access Network (E-UTRAN), trusted non-3GPP access, untrusted non-3GPP access and 5G new RANs.

In recent years, many handover authentication schemes have been proposed for various scenarios. However, most of these schemes are not suitable for *inter-domain* handover. Even the few schemes designed for *intra-domain* handover fail to simultaneously satisfy the requirements of *mutual authentication*, *key agreement*, *anonymity*, *traceability*, *robustness*, *perfect forward secrecy*, *master key forward secrecy*, *known randomness security* and *key escrow freeness*.

Mutual Authentication and Key Agreement. Mutual authentication and key agreement are the two fundamental goals of a handover authentication protocol in order to protect against basic security threats such as *impersonation*, *replay*, and *man-in-the-middle* attacks. A shared session key between the UE and AP_t is used to protect the confidentiality and integrity of subsequent communication. Towards basic mutual authentication and key agreement, substantial research has been done on handover authentication in various radio scenarios including 5G HetNets. Based on the design requirements, these schemes are mainly categorized into the following three types. (1) *AAA-based schemes:* The 3GPP committee proposed several solutions for seamless handover between 3GPP and non-3GPP access networks [5]. However, these protocols require the UE and AP_t to implement a full authentication procedure which involves the AAA server and suffers from efficiency and security drawbacks [6], [7]. In addition, different handover authentication procedures are required for diverse mobility scenarios which leads to a higher system complexity. Based on the idea of pre-authentication, handover authentication is enabled in WLAN [8] and 5G HetNets [3], [9], [10]. However, such schemes still need the involvement of the AAA server or similar entities and hence have performance limitations. (2) *Security Context Transfer (SCT)-based schemes:* To eliminate the involvement of the AAA server, SCT-based schemes [11], [12] allow AP_s and AP_t to exchange a random nonce without contacting the AAA server. However, SCT-based schemes rely on the existence of trust relationships among APs and hence are not suitable for 5G HetNets, where APs are located in different networks. (3) *Direct authentication schemes:* In direct handover authentication [13], the UE and AP_t achieve mutual authentication with key agreement without the involvement of the AAA server and the existence of trust relationship among AP_s and AP_t.

Anonymity. Although mutual authentication and key agreement are realized in the aforementioned schemes, the identity of a UE is transmitted in plaintext in authentication messages. Identity anonymity is important for achieving users' privacy protection in 5G HetNets [3]. To achieve anonymity, pseudo identities are adopted in [14], [15]. However, the pseudo identities are not chosen by UEs and the corresponding secret keys can be derived from the publicly transmitted message. A privacy-preserving handover authentication scheme for LTE-A networks can be found in [16].

Traceability and Robustness. Anonymity of UEs can be a double-edged sword. It may encourage some users to maliciously behave and harm others in the system without worrying about being punished. Hence, it is important to build mechanisms which can trace and reveal a UE's real identity in case malicious behaviors are detected. In previous schemes, traceability is enabled by the AAA server or other similar entities [14], [15], [17]. The traceability of such schemes is not *robust* since an innocent UE might be framed by a dishonest or compromised AAA server.

In robust handover authentication, if the AAA server is compromised and the UE, as an innocent user, is framed in the process of traceability, the UE should be capable of proving to any third party that it is framed by the AAA server. The scheme in [14] considered robustness of traceability under the strong assumption that the AAA server erases some temporary parameters honestly and cannot be compromised.

Perfect Forward Secrecy (PFS), Master Key Forward Secrecy (MKFS) and *Known Randomness Secrecy (KRS).* They are three important properties in key agreement [18], [19], [20]. PFS implies that even if the long-term secret keys of the UE and AP_t are compromised at any point in time, all the preceding session keys cannot be revealed; MKFS means that even if the master secret key of the AAA server is compromised at any point in time, all the preceding session keys between UE and AP_t cannot be revealed; and KRS, which is not supported in most of the existing schemes, ensures security of the current session key if the corresponding session-specific temporary random data are known to adversaries [21], [22].

Key Escrow Freeness (KEF). KEF requires that UEs' long-term secret keys are determined by themselves instead of the AAA server [23]. This property is critical to ensure secure communication without eavesdropping by a third party. However, many previous schemes such as [13], [15], [24] suffer from the key escrow problem in that UEs' long-term secret keys are completely controlled by the AAA server or a private key generator (PKG). Although the schemes in [17], [18], [25] realize KEF, none of them achieve KRS and the efficiency remains to be improved.

Inter-Domain Handover Authentication (i.e., Universality). This is a desirable property because it enables UEs' handover between domains managed by different operators without introducing a global PKG. In LTE-A networks, authentication servers are components of Evolved Packet Cores (EPCs) managed by operators. Therefore, inter-domain handover is also known as inter-EPC handover. In inter-EPC handover

authentication, there is no restriction on the types of source and target access networks. For examples, they can be 3GPP LET E-UTRAN, trusted non-3GPP access and untrusted non-3GPP access, even if the EPCs are different. Although the schemes in [20], [24] aim to realize inter-EPC handover authentication, a globally accessible entity, such as PKG and the AAA server, is required. Furthermore, inter-domain handover in 5GC means that a UE can handover between two different 5G new RANs. Particularly, in the non-standalone scenario of 5G, inter-domain handover should allow a UE to handover between a 5G new RAN and an access network of previous generations.

To the authors' knowledge, the aforementioned properties remain to be addressed for handover authentication in 5G HetNets. Specifically, neither robust traceability nor universality is obtained in existing schemes due to the lack of a global and uncompromisable entity to issue secret keys to UEs. Motivated by the recent explosion of interest around blockchain technologies [26], [27], we examine whether they make a good fit for the scenario of handover authentication in 5G HetNets. What is interesting and exciting is that blockchain-based distributed records have the properties of immutability and censorship resistance. In particular, the records are globally available and verifiable. Thus, it is quite natural to leverage blockchain technologies to address the challenging issues such as *traceability*, *robustness* and *universality* of handover authentication in 5G HetNets.

1.1 Our Contributions

In this paper, we introduce RUSH, a Robust and Universal Seamless Handover authentication protocol for 5G HetNets. RUSH leverages the trapdoor collision property of a chameleon hash function and the global availability and tamper-resistance of a blockchain, and achieves mutual authentication, key agreement, anonymity, traceability, robustness, perfect forward secrecy, master key forward secrecy, known randomness security, key escrow freeness, and universality. Our formal security proof based on the BAN-logic and formal verification based on the AVISPA tool indicate that RUSH is secure against various malicious attacks. In addition, comprehensive performance comparisons show that RUSH is very efficient in both computation and communication. Specifically, RUSH is characterized by the following attractive features:

- *Mutual Authentication with Key Agreement*. The trapdoor collision property of chameleon hash functions and the global availability and tamper-resistance of blockchains are exploited to ensure mutual authentication with key agreement.
- *Anonymity*. A pseudo identity chosen by a UE itself instead of its actual identity is used for generating trapdoor collisions for mutual authentication with key agreement.
- *Traceability*. An AAA server is able to trace and reveal the actual identity of any maliciously behaving UE by computing a chameleon hash based on authentication messages and comparing the hash value with the one recorded in the blockchain.
- *Robustness*. In order to prove that it is framed by the AAA server in the process of traceability, a UE first presents a signature initially issued by the AAA server to any third party. Then, it shows that the chameleon hash derived from the current authentication messages is not a valid message of the signature.
- *Perfect Forward Secrecy, Master Key Forward Secrecy, and Known Randomness Secrecy*. These properties are achieved because two random values and the long-term secret key from each participant are simultaneously involved in the generation of each session key based on the idea of Elliptic Curve Diffie-Hellman (ECDH) key agreement.
- *Key Escrow Freeness*. Long-term secret keys are chosen by UEs themselves instead of other entities. Note that the secret keys are indirectly acknowledged by the AAA server by recording the corresponding chameleon hash values in the blockchain.
- *Universality*. Since records in the blockchain are globally consistent, inter-domain handover authentication is ensured similar to the case of intra-domain handover authentication.

1.2 Related Work

Recently, substantial research has been done on handover authentication in various wireless scenarios including 5G HetNets. In order to realize handover authentication in wireless networks based on the IEEE 802.11 standard (e.g., WiFi), Mishra et al. [28] propose a proactive key distribution approach which distributes a new pairwise master key to neighbor APs using neighbor graphs. Furthermore, the idea of pre-authentication is adopted by Pack et al. [8] in WLAN and by Duan et al. [3], Alam et al. [9] and Sharma et al. [10] in 5G HetNets. All these schemes require the involvement of an AAA server or other similar entities to ensure the handover security, which increases the overall system complexity and hence has limited applications. To eliminate the involvement of the AAA server, security context based schemes [11], [12], [29], [30], [31], [32] are proposed. However, these schemes rely on the establishment of trust relationships among APs and hence are not suitable for 5G HetNets. To eliminate the involvement of the AAA server and the establishment of trust relationship among APs, Kim et al. [13] proposes a direct handover authentication scheme based on identity-based cryptography. Similar schemes can also be found in [15], [17], [20], [33], [34], [35]. However, all these schemes need to perform computationally expensive bilinear pairing operations and hence are inefficient. In order to improve the computational efficiency, Cao et al. [36] proposes a pairing-free identity-based authenticated key exchange scheme, which has been used for handover authentication in LTE scenario [24]. Note that a PKG has to be introduced to issue private keys to UEs and APs in all identity-based cryptographic solutions, and the key escrow problem remains to be solved in [15], [20], [24], [34], [36]. To achieve the PFS property, two handover authentication schemes are proposed [18], [37]. However, both schemes have high communication cost and large storage overheads. Additionally, Yoon et al. [38] shows that the scheme [18] cannot realize PFS as claimed. To overcome these issues, Zhang et al. proposes a direct handover authentication scheme based on a variant of the special double-trapdoor chameleon hash function [39]. However, the scheme cannot realize known randomness secrecy and universality.

Generally, because the communication scenario in 5G HetNets is highly integrated and complex [3], it is critical to design handover authentication protocols with robust security and high efficiency. However, in the above schemes, only the schemes in [10], [17], [25] provide formal security proofs. The key escrow problem is only addressed in [17], [18], [25], [35]. No previous scheme realizes known randomness secrecy. Without the existence of an uncompromisable and globally accessible entity issuing secret keys to UEs, neither robust traceability nor universality can be obtained. Recently, blockchain technologies have gained prominent popularity mostly due to the lack of a central authority and have thus been used for authentication in sensor networks [40] and data encryption in WiFi [41]. Zhang et al. propose two blockchain-based fair payment protocols called BPay [27] and BCPay [42] for outsourcing services in cloud computing. The protocol BPay [27] is compatible with the Bitcoin blockchain based on an iterative all-or-nothing checking-proof protocol and a top-down checking method. However, the efficiency remains to be improved. At the cost of losing the compatibility with the Bitcoin blockchain, the protocol BCPay [42] realizes robust fair payment based on a one round all-or-nothing checking-proof protocol and hence is very efficient in terms of the computation cost and the number of transactions. Furthermore, Zhang et al. [43] propose a trustworthy keyword search scheme over encrypted data with two-side verifiability via blockchains.

RUSH employs a chameleon hash function and a blockchain. We note that since the introduction of Bitcoin blockchain, significant progress has been made in designing highly efficient blockchains. For example, Kogias et al. [44] proposes a consensus protocol for constructing secure and efficient blockchains which achieve transaction rates higher than that of Paypal, with a block confirmation latency of 15-20 seconds. Tomescu et al. [45] present Catena, an efficiently-verifiable Bitcoin witnessing scheme. With the help of the opcode OP_RETURN [26], Catena enables any number of thin clients, such as mobile phones, to efficiently agree on an application-specific statement managed by an untrustworthy server. Its clients only need to download all Bitcoin block headers and a small proof for the statement in a block.

1.3 Organization

The rest of the paper is organized as follows. We first present the system model, adversary model and the main idea in Section 2. Then, the proposed handover authentication protocol RUSH is described in Section 3. In Section 4, we evaluate the security of RUSH. Performance related issues are presented in Section 5. Finally, concluding remarks are made in Section 6.

2 Models, Design Goals and Main Idea

In this section, we first present the system model, and then introduce the adversary model and design goals, followed by the main idea behind RUSH.

2.1 System Model

Considering the fact that 5G is backward-compatible and 3GPP has defined a new 5G core network as well as a new radio access technology called 5G new radio, it is possible to integrate elements of different generations in different configuration with 5G. In this paper, we mainly focus on the typical 5G HetNets architecture in which the access networks take the form of 3GPP LTE-A E-UTRAN, non-3GPP access and 5G new radio access. More exactly, there are different tiers in 5G HetNets, in which each tier models base stations of a particular class such as femtocells, picocells, microcells, and macrocells. The core network may be either EPC or 5GC. As shown in Fig. 2, a UE either accesses Internet services, including a blockchain, via EPC based on E-UTRAN and non-3GPP access networks such as WiMAX, WiFi and CDMA-2000 [46], or accesses Internet services via 5GC based on 5G new radio access networks. In E-UTRAN, the air interface technologies are based on eNB and HeNB. Non-3GPP access networks are categorized into trusted non-3GPP access networks and untrusted ones. For UEs, the trust relationship of a non-3GPP access network is determined by the Home Public Land Mobile Network (HPLMN) operator in the non-roaming scenario and by the Home Subscriber Server (HSS) or 3GPP AAA server in HPLMN in the roaming scenario. In 5G networks, the air interface technologies are based on gNB. *As shown in Fig. 2, we focus on handover in 5G HetNets, including intra-domain handover in E-UTRAN (❶), intra-domain handover in trusted non-3GPP access networks (❷), intra-domain handover in untrusted non-3GPP access networks (❸❹), intra-domain handover in 5G new radio access networks (❺), intra-domain handover among these heterogeneous access networks (❻❼) and inter-domain handover among these heterogeneous access networks (❽❾).*

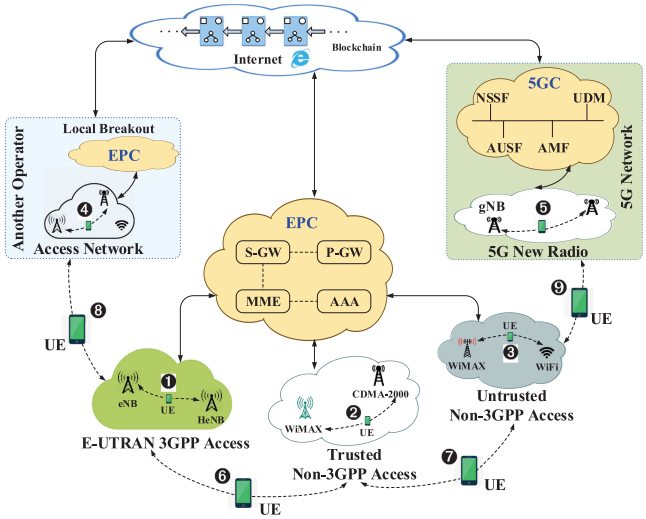


Fig. 2. The handover system architecture in 5G HetNets. ❶❷❸❹❺: Intra-domain handover between networks of the same type. ❻❼: Intra-domain handover between heterogeneous networks. ❽❾: Inter-domain handover.

The access to EPC is shown in Fig. 3. When a UE connects to EPC over E-UTRAN, the key component Mobility Management Entity (MME) of EPC performs mutual authentication with UE based on Evolved Packet System Authentication and Key Agreement (EPS-AKA). In the case of non-3GPP access networks, the non-3GPP access authentication is executed between the UE and the 3GPP AAA server across a STa/SWa reference point. Note that the authentication signaling passes through the Proxy AAA server in roaming scenarios. For a non-3GPP access network, if there is no pre-configured information in UEs, the access network is considered to be

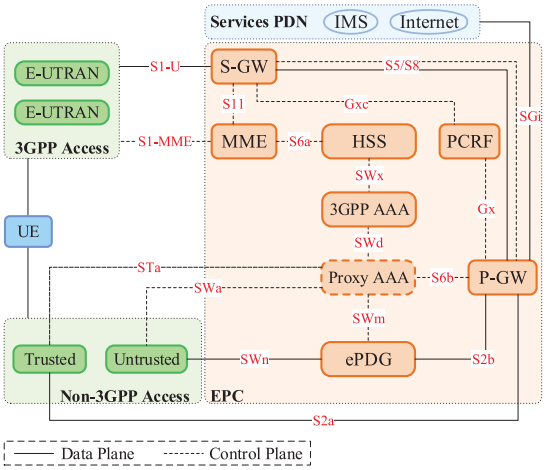


Fig. 3. The access of heterogeneous networks to EPC.

untrusted [47]. For access to EPC via a trusted non-3GPP access network, UEs and the AAA server implement Extensible Authentication Protocol Authentication and Key Agreement (EAP-AKA) or improved EAP-AKA (i.e., EAP-AKA') to accomplish the access authentication. For an untrusted non-3GPP access network, an IPsec tunnel is established between the UE and Evolved Packet Data Gateway (ePDG). As specified in 3GPP TS 33.402 [48], the access authentication signaling between the UE, ePDG and the 3GPP AAA server shall be based on EAP-AKA. During the authentication of the UE for accessing EPC via ePDG, ePDG shall initiate EAP-AKA based authentication between the UE and the 3GPP AAA server. ePDG shall extract the EAP messages received from the UE over Internet Key Exchange Protocol Version 2 (IKEv2), send them to the 3GPP AAA server and send the EAP message received from the 3GPP AAA server to the UE over IKEv2.

On the other hand, the access to 5GC is shown in Fig. 4, which can be found at 3GPP TS 33.501 [49]. In 5G, network slices are defined to be logical networks which comprise of control plane and user plane Network Functions (NFs) with different capabilities. The 5G network also needs to carry out basic functions including communicating with the UE, storing its subscription and credentials, enabling access to external networks, and managing the network access and mobility. As shown in Fig. 4, The Authentication Server Function (AUSF) acts as an authentication server and it has part of the HSS functionality of EPC. The Unified Data Management (UDM) has part of the home subscriber server functionality of EPC. The Access and Mobility Management Function (AMF) supports registration management, mobility management, access authentication and authorization, which has part of the MME functionality of EPC. The Session Management Function (SMF) mainly supports session management and it has part of the MME and Packet Data Network Gateway (P-GW) functionality of EPC. The Policy Control Function (PCF) has part of the policy and charging rules functionality of EPC. The Network Exposure Function (NEF) realizes translation of internal/external information. The NF Repository Function (NRF) supports service discovery and maintains NF profile. The Network Slice Selection Function (NSSF) enables the selection of network slice instances and determines the AMF set to be used to serve the UE. Similar to the case in EPC, the Application Function (AF) in 5G uses the services and information

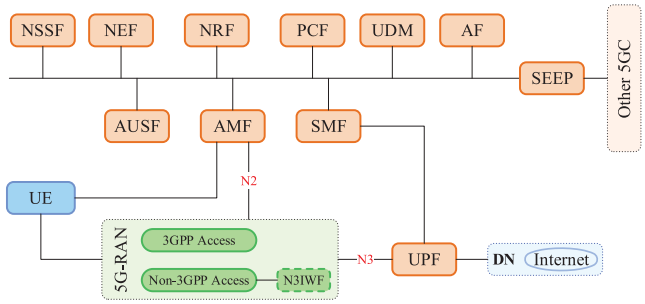


Fig. 4. The access of heterogeneous networks to 5GC.

offered by other 3GPP network functions based on the configured policies. The User Plane Function (UPF) has part of the S-GW and P-GW functionality of EPC. The Data Network (DN) means internet access or services from operators and third parties. The Security Edge Protection Proxy (SEPP) is used for secure connectivity to other operators. Note that NEF, NRF, NSSF and SEEP are not present in EPC. In addition, non-3GPP access networks are connected to 5GC via a Non-3GPP InterWorking Function (N3IWF) which interfaces to the control-plane function AMF and the user-plane function UPF via N2 interface and N3 interface, respectively.

2.2 Adversary Model and Design Goals

In RUSH, air interfaces are public and hence any adversary may compromise wireless transmissions between UEs and the access network. An adversary may try to launch typical protocol attacks including impersonation, replay, and man-in-the-middle attacks, etc. Therefore, a handover authentication protocol for 5G HetNets should realize *mutual authentication* between the UE and the target AP. After the UE comes to the area covered by the target AP, the communication between them may be eavesdropped by adversaries, and hence a session *key agreement* should be enabled by the UE and the target AP during handover authentication. To resist potential attacks due to the compromise of long-term secret keys of UEs and APs, the compromise of the master secret key of AAA servers, and the leakage of session-specific temporary random data, a handover authentication protocol should realize *perfect forward secrecy*, *master key forward secrecy* and *known randomness secrecy*, respectively. Additionally, user identity privacy is taken into account in our security model. In the process of handover authentication, the UE does not need to transmit its actual identity. It only transmits a pseudo identity chosen by itself for *anonymity*. Under the protection of anonymity, some users may maliciously behave and harm others in the system without worrying about being punished. Therefore, a handover authentication protocol should enable *traceability* by revealing the malicious user's real identity. It is noted that the AAA server is not fully trusted by users in our security model. Specifically, an innocent user might be framed by a dishonest or compromised AAA server. A secure handover authentication protocol should realize *robustness* which enables users to prove to any third party that it is framed by the AAA server. In order to realize secure communication without eavesdropping by any third party including the compromised AAA server, *key escrow freeness* should be realized for a UE to choose its long-term secret key by itself. Currently, in 5G networks, the trusted SEEP is involved in communication between two 5GCs of different operators. To

TABLE 1
Notations Used in RUSH

Notation	Meaning
$x \in_R X$	x is chosen at random from a set X.
$s_1 \parallel s_2$	The concatenation of two bit strings s_1 and s_2.
H_i	A secure hash function, for $i = 0, 1, 2, 3$.
$\mathbb{F}_t$	A finite field of prime power order t.
$E(\mathbb{F}_t)$	An elliptic curve over $\mathbb{F}_t$.
$\mathbb{G}$	The subgroup of prime order q in $E(\mathbb{F}_t)$.
P	The generator of $\mathbb{G}$.
$\mathbb{Z}_q$	A finite field of integers modulo prime q.
$\mathbb{Z}_q^*$	The multiplicative subgroup of the finite field $\mathbb{Z}_q$, and the corresponding set is $\{1, 2, \cdots, q-1\}$.
ID_A	The identity of the entity A.
$CH(\cdot)$	A secure chameleon hash function.
T_{Curr}	The current time used as a timestamp.
T_{Exp}	The expiration time of registration.
$sk_A = (k_A, x_A)$	The long-term secret key of the entity A (i.e., UE or AP).
$sk_{\text{AAA}}/vk_{\text{AAA}}$	The signing/verification key of the AAA server.
$\mathsf{sig}_{\text{AAA}}(\cdot)/\mathsf{ver}_{\text{AAA}}(\cdot)$	The ECDSA signing/verification algorithm of the AAA server.

eliminate trusted entities, we aim to realize *universality* for handover in 5G HetNets. In other words, inter-domain handover authentication should be supported. Moreover, the two fundamental requirements, mutual authentication and key agreement, should be proved formally. To guarantee the quality of service in 5G HetNets, handover authentication protocols should be as efficient as possible.

2.3 Overview of RUSH

In this section, we present the main idea behind RUSH for realizing the aforementioned design goals.

2.3.1 Inter-Domain Mutual Authentication with KEF

A secure chameleon hash function allows an entity who has a secret key (i.e., trapdoor) to compute a collision. Therefore, given a pre-registered hash value, the ability to generate a new collision of the same hash value implicitly indicates a UE (resp. AP) knows the trapdoor and hence is legitimate. In RUSH, a chameleon hash value is first generated by the UE (resp. AP) and then is recorded in a blockchain by an AAA server. The correctness of the chameleon hash value in the blockchain can be checked by the UE (resp. AP) which ensures the global consistency of the hash value. The trapdoor is completely chosen by the UE (resp. AP), which enables mutual authentication with the property of KEF. Immutability and finality of blockchain records allow handover authentication between heterogeneous access networks of different domains.

2.3.2 Key Agreement with PFS, MKFS and KRS

To achieve PFS and MKFS, we leverage the idea of ECDH key agreement by adopting two random values from each participant. To realize KRS, we let the long-term secret keys of both participants and the random values mentioned above contribute equally to the generation of a session key.

2.3.3 Anonymity, Traceability and Robustness

Existing solutions only realize limited anonymity because pseudo identities are specified in advance by an AAA server. In RUSH, a UE chooses a random pseudo identity in real time. To address the issue of traceability, RUSH allows the AAA server to trace and reveal the actual identity of the malicious UE based on public authentication messages. In fact, the AAA server just needs to calculate the chameleon hash value and compare it with the one in the blockchain. To protect an honest UE from being framed in the process of traceability, RUSH realizes robustness based on chameleon hash function and digital signature.

2.3.4 Fast Handover Authentication

In RUSH, mutual authentication and key agreement are achieved with three short message exchanges using ECC-based chameleon hash and ECDH.

3 RUSH: Robust and Universal Seamless Handover Authentication

For ease of reference, important notations are summarized in Table 1. RUSH consists of four phases including system initialization, network registration, handover preparation, and handover authentication.

3.1 System Initialization

Let λ be a security parameter, t a prime power, $E(\mathbb{F}_t)$ an elliptic curve over the finite field $\mathbb{F}_t$, and P a point of $E(\mathbb{F}_t)$ with prime order q. Denote by $\mathbb{G}$ the subgroup generated by P in the additive group of $E(\mathbb{F}_t)$. The AAA server initializes RUSH following the procedures below.

1) The AAA server chooses secure hash functions:
   - $H_0 : \{0,1\}^* \times \mathbb{Z}_q^* \rightarrow \mathbb{Z}_q^*$,
   - $H_1 : \{0,1\}^* \times \mathbb{G}^2 \times \{0,1\}^* \rightarrow \mathbb{Z}_q^*$,
   - $H_2 : \{0,1\}^* \times \mathbb{Z}_q^* \times \mathbb{G}^2 \times \{0,1\}^* \times \mathbb{Z}_q^* \times \mathbb{G}^2 \times \{0,1\}^* \times \mathbb{G} \rightarrow \{0,1\}^\lambda$,
   - $H_3 : \mathbb{G} \times \{0,1\}^\lambda \times \{0,1\}^* \times \mathbb{Z}_q^* \times \mathbb{G}^2 \times \{0,1\}^* \rightarrow \{0,1\}^\lambda$.
2) The AAA server specifies a chameleon hash function to be used by UEs and APs. Similar to each AP, a UE has a pair of public key and secret key associated with the chameleon hash function. The public key is called a hash key and the secret key is known as a trapdoor. In fact, the ECC variant of the chameleon hash [50] is adopted in RUSH. Specifically, an initial input (m^*, r^*), where $m^*, r^* \in \mathbb{Z}_q^*$, is first chosen by the UE. Then, given an input (m, r), where $m, r \in \mathbb{Z}_q^*$, the chameleon hash is defined as $CH_Y(m, r) = mP + rY$, where (P, Y) is the hash key and (k, x) is the trapdoor with $x \in \mathbb{Z}_q^*$, $Y = xP$ and $k = m^* + r^*x$. The chameleon hashing has the following properties:
   - *Collision Resistance.* It is infeasible for anyone except the holder of the trapdoor to find $m', r' \in \mathbb{Z}_q^*$ such that $(m, r) \neq (m', r')$ and $CH_Y(m, r) = CH_Y(m', r')$.
   - *Trapdoor Collisions.* Given an additional input component $r' \in \mathbb{Z}_q^*$, the holder of the trapdoor can easily find $m' = k - r'x \bmod q$ such that $CH_Y(m^*, r^*) = CH_Y(m', r')$, where (m^*, r^*) is the initial input.

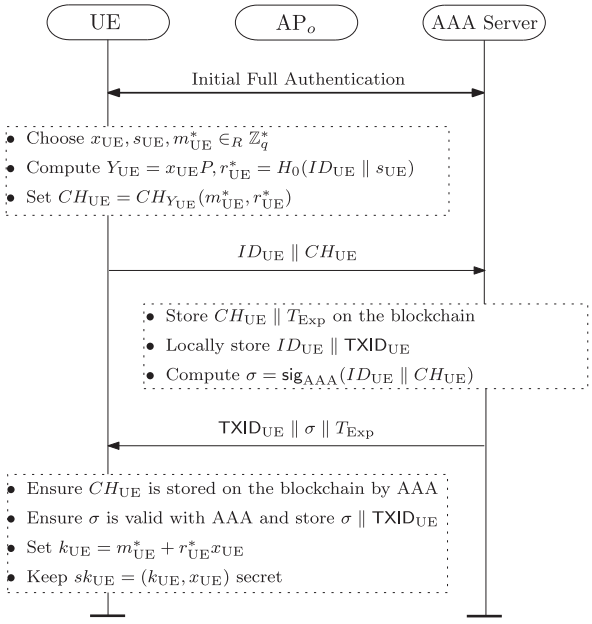


Fig. 5. The network registration phase of RUSH.

(3) Furthermore, the AAA server generates a signing and verification key pair (sk_{AAA}, vk_{AAA}) under an Elliptic Curve Digital Signature Algorithm (ECDSA), which is used to publish transactions in the blockchain.

4) Finally, the AAA server publishes the system public parameter $PK = \{q, P, \mathbb{G}, H_0, H_1, H_2, H_3, vk_{AAA}\}$, and secretly saves the master secret key sk_{AAA}.

3.2 Network Registration Phase

When a UE registers to the network with actual identity ID_{UE}, an initial full authentication, which is based on secure EPS-AKA and 5G-AKA respectively in LTE and 5G, is performed, and a secret key is shared by the UE, the original access point AP_o and the AAA server [47]. Then, all future communications between the UE and the AAA server will be via AP_o and protected with the shared secret key. Even if some security vulnerabilities have been found in EPS-AKA and 5G-AKA, corresponding solutions are also proposed [51], [52], [53], [54]. Furthermore, our RUSH focuses on the handover security in 5G HetNets and the initial full authentication can be realized based on existing security-enhanced EPS-AKA and 5G AKA protocols. In order to complete the network registration, as shown in Fig. 5, the following procedures are further performed between the UE and the AAA server.

1) The UE first chooses $x_{UE}, s_{UE}, m^*_{UE} \in_R \mathbb{Z}^*_q$, then computes $Y_{UE} = x_{UE}P, r^*_{UE} = H_0(ID_{UE} \| s_{UE})$ and $CH_{UE} = CH_{Y_{UE}}(m^*_{UE}, r^*_{UE})$, and sends $ID_{UE} \| CH_{UE}$ to the AAA server,

$$\text{UE} \rightarrow \text{the AAA server} : ID_{UE} \| CH_{UE}.$$

2) Upon receiving $ID_{UE} \| CH_{UE}$ from the UE, the AAA server creates a transaction TxHandoverUE shown in Fig. 6 to store $\text{data}_{UE} = CH_{UE} \| T_{Exp}$ on the blockchain by the opcode OP_RETURN of the transaction, where T_{Exp} is the expiration time of the registration of UE. In Fig. 6, $\mathsf{Tx}^{(0)}_{UE}$ is an unredeemed transaction with value d_0 ฿ of the AAA server. Let TXID_{UE} be the transaction identity. The AAA server locally stores $ID_{UE} \| \mathsf{TXID}_{UE}$, computes $\sigma = \mathsf{sig}_{AAA}(ID_{UE} \| CH_{UE})$ and sends $\mathsf{TXID}_{UE} \| \sigma \| T_{Exp}$ to the UE,

$$\text{the AAA server} \rightarrow \text{UE} : \mathsf{TXID}_{UE} \| \sigma \| T_{Exp}.$$

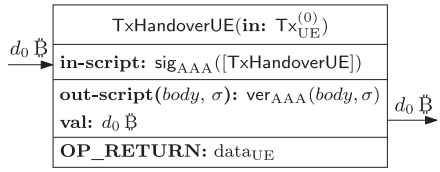


Fig. 6. The transaction TxHandoverUE.

3) Upon receiving TXID_{UE}, σ and T_{Exp} from the AAA server, the UE verifies that CH_{UE} is stored on the blockchain by the AAA server. Specifically, the UE locates the transaction with identity TXID_{UE}. To ensure the transaction is created by the AAA server, the UE checks if the transaction spends coins originating from the AAA server and its output script scriptPubKey is associated with vk_{AAA}.[1] Then, the UE checks if the data output by the opcode OP_RETURN of the transaction is CH_{UE}. If true, the UE further verifies the block corresponding to the transaction TXID_{UE} based on its local storage.[2] If the block is valid, the UE thinks that CH_{UE} is correctly stored on the blockchain by the AAA server. Furthermore, the UE ensures $\mathsf{ver}_{AAA}(ID_{UE} \| CH_{UE}, \sigma) = \mathsf{true}$ and locally stores $\sigma \| \mathsf{TXID}_{UE}$, sets $k_{UE} = m^*_{UE} + r^*_{UE} x_{UE}$ and keeps a long-term secret key $sk_{UE} = (k_{UE}, x_{UE})$.

On the other hand, each AP performs similar procedures for network registration. In particular, a target access point AP_t chooses $x_{AP_t}, s_{AP_t}, m^*_{AP_t} \in_R \mathbb{Z}^*_q$, and checks if $CH_{AP_t} = CH_{Y_{AP_t}}(m^*_{AP_t}, r^*_{AP_t})$ is stored in the blockchain by the AAA server, where $Y_{AP_t} = x_{AP_t}P$, $r^*_{AP_t} = H_0(ID_{AP_t} \| s_{AP_t})$. If it is, AP_t keeps a long-term secret key $sk_{AP_t} = (k_{AP_t}, x_{AP_t})$ where $k_{AP_t} = m^*_{AP_t} + x_{AP_t} r^*_{AP_t}$.

3.3 Handover Preparation Phase

When a UE is in the area covered by AP_s, the UE prepares for future handover authentication. Specifically, the UE broadcasts CH_{UE} to neighboring APs including AP_t. At the same time, the UE obtains the chameleon hash values of the neighboring APs through P-GW of the current LTE-EPC or UPF of the current 5GC. Cooperative communication technologies in 5G HetNets can be used to facilitate the preparation procedures [55]. Even if in a visited area, the chameleon hash values can also be obtained based on local breakout [56]. On the other hand, upon receiving the

1. Exactly, from a rational point of view, the UE only needs to ensure the output script scriptPubKey of the transaction is associated with vk_{AAA} in that nobody wants to transfer his coins to others without compensation.

2. To significantly reduce the local storage overhead of users (UEs and APs), blockchain techniques similar to Catena [45] can be used in RUSH. To be specific, each user locally stores all the block headers. In addition, a UE (resp. an AP) stores the output data of OP_RETURN in transactions associated with APs (resp. UEs) created by the AAA servers.

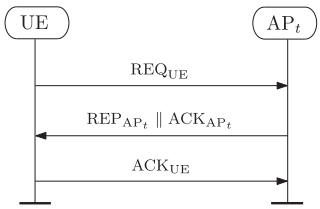


Fig. 7. The message exchange in handover authentication of RUSH.

message broadcast by the UE, the neighboring APs including AP_t temporarily store CH_{UE} upon ensuring that CH_{UE} has been put in the blockchain by the corresponding AAA server. Note that the neighboring APs can also download chameleon hash values from the blockchain in advance to reduce transmission overhead.

3.4 Handover Authentication Phase

Referring to Fig. 7, when a UE moves from the area covered by AP_s into the area covered by AP_t, a mutual authentication with key agreement between the UE and AP_t proceeds as follows.

1) The UE first chooses a pseudo identity $PID_{UE} \in_R \{0,1\}^*$ and α_{UE}, $\beta_{UE} \in_R \mathbb{Z}_q^*$, then computes $A_{UE} = \alpha_{UE} Y_{UE}, B_{UE} = \beta_{UE} Y_{UE}$. In order to prove the legitimacy of its identity, the UE sets $\gamma_{UE} = H_1(PID_{UE} \| A_{UE} \| B_{UE} \| T_{Curr})$ and computes

$$m_{UE} = k_{UE} - r_{UE} x_{UE},$$

where T_{Curr} is a timestamp and $r_{UE} = \alpha_{UE}\gamma_{UE}$. Finally, the UE sends a request message REQ_{UE} to AP_t,

$$UE \rightarrow AP_t : REQ_{UE} = PID_{UE} \| m_{UE} \| A_{UE} \| B_{UE} \| T_{Curr}.$$

2) Upon receiving REQ_{UE} from UE, AP_t uses T_{Curr} to prevent replay attacks and computes $\gamma_{UE} = H_1(PID_{UE} \| A_{UE} \| B_{UE} \| T_{Curr})$. Then, AP_t checks the legitimacy of the UE based on Equation (1).

$$m_{UE} P + \gamma_{UE} A_{UE} = CH_{UE}. \quad (1)$$

Note that $CH_{Y_{UE}}(m_{UE}, r_{UE}) = m_{UE}P + \gamma_{UE}A_{UE}$ with $r_{UE} = \alpha_{UE}\gamma_{UE}$. If Equation (1) does not hold, the UE is illegitimate and AP_t quits. Otherwise, AP_t chooses $\alpha_{AP_t}, \beta_{AP_t} \in_R \mathbb{Z}_q^*$ and computes

$$A_{AP_t} = \alpha_{AP_t} Y_{AP_t}, B_{AP_t} = \beta_{AP_t} Y_{AP_t}.$$

Then, AP_t chooses its own timestamp T_{Curr} and sets $\gamma_{AP_t} = H_1(ID_{AP_t} \| A_{AP_t} \| B_{AP_t} \| T_{Curr})$ and $m_{AP_t} = k_{AP_t} - r_{AP_t} x_{AP_t}$ with $r_{AP_t} = \alpha_{AP_t}\gamma_{AP_t}$. To make a key agreement with the UE, AP_t computes a secret K_{AU} based on Equation (2) and sets a response message REP_{AP_t} based on Equation (3). The pairwise transient key PTK_{AP_t} is calculated based on Equation (4).

$$K_{AU} = x_{AP_t}(\alpha_{AP_t} + \beta_{AP_t})(A_{UE} + B_{UE}), \quad (2)$$

$$REP_{AP_t} = ID_{AP_t} \| m_{AP_t} \| A_{AP_t} \| B_{AP_t} \| T_{Curr}, \quad (3)$$

$$PTK_{AP_t} = H_2(REQ_{UE} \| REP_{AP_t} \| K_{AU}). \quad (4)$$

Finally, AP_t computes $ACK_{AP_t} = H_3(K_{AU} \| PTK_{AP_t} \| REP_{AP_t})$, and sends REP_{AP_t} and ACK_{AP_t} to the UE,

TABLE 2
BAN-Logic Notations

Notation	Description
$P \models X$	The entity P believes the formula X.
$P \lhd X$	P sees X.
$P \Rightarrow X$	P has complete jurisdiction over X.
$P \sim X$	P has once said X.
$\sharp(X)$	X is fresh.
$\{X\}_K$	X is hided based on the secret K.
$P \overset{K}{\leftrightarrow} Q$	The entities P and Q share a secret key K.

$$AP_t \rightarrow UE : REP_{AP_t} \| ACK_{AP_t}.$$

3) Upon receiving REP_{AP_t} and ACK_{AP_t} from AP_t, the UE first checks freshness of T_{Curr} to prevent replay attacks and computes $\gamma_{AP_t} = H_1(ID_{AP_t} \| A_{AP_t} \| B_{AP_t} \| T_{Curr})$. Then, the UE checks the legitimacy of AP_t based on Equation (5).

$$m_{AP_t} P + \gamma_{AP_t} A_{AP_t} = CH_{AP_t}. \quad (5)$$

Note that $CH_{Y_{AP_t}}(m_{AP_t}, r_{AP_t}) = m_{AP_t}P + \gamma_{AP_t}A_{AP_t}$ with $r_{AP_t} = \alpha_{AP_t}\gamma_{AP_t}$. If Equation (5) does not hold, AP_t is illegitimate and the UE quits. Otherwise, the UE computes a secret K_{UA} based on Equation (6), and the pairwise transient key PTK_{UE} is calculated based on Equation (7).

$$K_{UA} = x_{UE}(\alpha_{UE} + \beta_{UE})(A_{AP_t} + B_{AP_t}), \quad (6)$$

$$PTK_{UE} = H_2(REQ_{UE} \| REP_{AP_t} \| K_{UA}). \quad (7)$$

Finally, the UE checks the validity of the pairwise transient key agreement based on $ACK_{AP_t} = H_3(K_{UA} \| PTK_{UE} \| REP_{AP_t})$. If successful, the UE sends an acknowledge message ACK_{UE} to AP_t,

$$UE \rightarrow AP_t : ACK_{UE} = H_3(K_{UA} \| PTK_{UE} \| REQ_{UE}).$$

4) Upon receiving ACK_{UE} from the UE, AP_t checks the validity of the pairwise transient key based on $ACK_{UE} = H_3(K_{AU} \| PTK_{AP_t} \| REQ_{UE})$.

4 Security Evaluation

In this section, based on the discussion in Section 2.2, we first formally prove the two fundamental security properties of RUSH based on the well-known Burrows-Abadi-Needham (BAN)-logic [57], which has been widely applied to prove the properties of mutual authentication and key agreement of security protocols [9], [10], [17]. Furthermore, we verify various security properties of RUSH based on the Automated Validation of Internet Security Protocols and Applications (AVISPA) tool and extensive analysis.

4.1 Formal Security Proof Based on the BAN-Logic

The notations and rules of the BAN-logic are first given in Tables 2 and 3, respectively. Then, according to the analytic procedure of the BAN-logic, we present the goals of RUSH and the assumptions on the initial state. Finally, we prove RUSH achieves various goals in detail.

TABLE 3
BAN-Logic Rules

Rule	Meaning
$\frac{P \models \sharp(X)}{P \models \sharp(X,Y)}$	The fresh-promotion rule.
$\frac{P \models \sharp(X), P \models Q \sim X}{P \models Q \models X}$	The nonce-verification rule.
$\frac{P \models Q \models (X,Y)}{P \models Q \models X}, \frac{P \models (X,Y)}{P \models X}$	The decomposition rule.
$\frac{P \models X, P \models Y}{P \models (X,Y)}$	The composition rule.
$\frac{P \models Q \Rightarrow X, P \models Q \models X}{P \models X}$	The jurisdiction rule.
$\frac{P \models P \overset{K}{\leftrightarrow} Q, P \{X\}_K}{P \models Q \sim X}$	The message-meaning rule.

4.1.1 The Goals of RUSH

The basic goal of RUSH is to realize mutual authentication with key agreement between the UE and AP_t. In particular, each entity not only believes the pairwise transient key (PTK) itself, but also has to believe that the other entity also believes the key. In the BAN-logic, the basic goals of RUSH can be described as:

Goal 1. $\mathrm{UE} \models \mathrm{UE} \overset{\mathrm{PTK}}{\longleftrightarrow} \mathrm{AP}_t$.
Goal 2. $\mathrm{AP}_t \models \mathrm{AP}_t \overset{\mathrm{PTK}}{\longleftrightarrow} \mathrm{UE}$.
Goal 3. $\mathrm{UE} \models \mathrm{AP}_t \models \mathrm{AP}_t \overset{\mathrm{PTK}}{\longleftrightarrow} \mathrm{UE}$.
Goal 4. $\mathrm{AP}_t \models \mathrm{UE} \models \mathrm{UE} \overset{\mathrm{PTK}}{\longleftrightarrow} \mathrm{AP}_t$.

4.1.2 Assumptions

To analyze RUSH, there are two necessary assumptions, which are reasonable because each message is chosen by the corresponding entity.

Assumption 1. $\mathrm{AP}_t \models \mathrm{UE} \Rightarrow \mathrm{REQ}_{\mathrm{UE}}$.
Assumption 2. $\mathrm{UE} \models \mathrm{AP}_t \Rightarrow \mathrm{REP}_{\mathrm{AP}_t}$.

4.1.3 Security Result

The security result of RUSH is shown in Theorem 1.

Theorem 1. *In RUSH, the UE and AP_t mutually authenticate each other and secretly share a session key while keeping the UE's identity anonymity.*

Proof. We first describe the messages involved in the handover authentication of RUSH. Then, based on the BAN-logic rules and assumptions, we prove RUSH realizes mutual authentication with key agreement and UE's identity anonymity. The details are as follows:

Message 1. $\mathrm{UE} \rightarrow \mathrm{AP}_t$: $\mathrm{REQ}_{\mathrm{UE}}$, where

$$\mathrm{REQ}_{\mathrm{UE}} = \mathrm{PID}_{\mathrm{UE}} \parallel m_{\mathrm{UE}} \parallel A_{\mathrm{UE}} \parallel B_{\mathrm{UE}} \parallel T_{\mathrm{Curr}}.$$

Message 2. $\mathrm{AP}_t \rightarrow \mathrm{UE}$: $\mathrm{REP}_{\mathrm{AP}_t} \parallel \mathrm{ACK}_{\mathrm{AP}_t}$, where

$$\mathrm{REP}_{\mathrm{AP}_t} = ID_{\mathrm{AP}_t} \parallel m_{\mathrm{AP}_t} \parallel A_{\mathrm{AP}_t} \parallel B_{\mathrm{AP}_t} \parallel T_{\mathrm{Curr}},$$
$$\mathrm{ACK}_{\mathrm{AP}_t} = H_3(K_{\mathrm{AU}} \parallel \mathrm{PTK}_{\mathrm{AP}_t} \parallel \mathrm{REP}_{\mathrm{AP}_t}).$$

Message 3. $\mathrm{UE} \rightarrow \mathrm{AP}_t$: $\mathrm{ACK}_{\mathrm{UE}}$, where

$$\mathrm{ACK}_{\mathrm{UE}} = H_3(K_{\mathrm{UA}} \parallel \mathrm{PTK}_{\mathrm{UE}} \parallel \mathrm{REQ}_{\mathrm{UE}}).$$

According to *Message 1*, we have:

Step 1. $\mathrm{AP}_t \lhd \mathrm{REQ}_{\mathrm{UE}}$
In RUSH, AP_t checks T_{Curr} to prevent replay attacks. Hence,
Step 2. $\mathrm{AP}_t \models \sharp(B_{\mathrm{UE}})$
According to *Step 2* and the fresh-promotion rule, we have:
Step 3. $\mathrm{AP}_t \models \sharp(\mathrm{REQ}_{\mathrm{UE}})$
Based on RUSH, AP_t sets $\gamma_{\mathrm{UE}} = H_1(\mathrm{PID}_{\mathrm{UE}} \parallel A_{\mathrm{UE}} \parallel B_{\mathrm{UE}} \parallel T_{\mathrm{Curr}})$ and checks if $m_{\mathrm{UE}}P + \gamma_{\mathrm{UE}}A_{\mathrm{UE}} = CH_{\mathrm{UE}}$. If it is, according to *Step 1*, we have:
Step 4. $\mathrm{AP}_t \models \mathrm{UE} \mid \sim \mathrm{REQ}_{\mathrm{UE}}$
According to *Step 3*, *Step 4*, and the nonce-verification rule, we have:
Step 5. $\mathrm{AP}_t \models \mathrm{UE} \models \mathrm{REQ}_{\mathrm{UE}}$
According to *Step 5*, and the decomposition rule, we have:
Step 6. $\mathrm{AP}_t \models \mathrm{UE} \models A_{\mathrm{UE}}$
According to *Step 6*, *Assumption 1*, and the jurisdiction rule, we have:
Step 7. $\mathrm{AP}_t \models A_{\mathrm{UE}}$
According to *Step 5*, and the decomposition rule, we have:
Step 8. $\mathrm{AP}_t \models \mathrm{UE} \models B_{\mathrm{UE}}$
According to *Step 8*, *Assumption 1*, and the jurisdiction rule, we have:
Step 9. $\mathrm{AP}_t \models B_{\mathrm{UE}}$
Based on *Step 7*, *Step 9*, and the composition rule, we have:
Step 10. $\mathrm{AP}_t \models A_{\mathrm{UE}} + B_{\mathrm{UE}}$
In RUSH, AP_t chooses $x_{\mathrm{AP}_t}, \alpha_{\mathrm{AP}_t}, \beta_{\mathrm{AP}_t} \in_R \mathbb{Z}_q^*$. Hence,
Step 11. $\mathrm{AP}_t \models x_{\mathrm{AP}_t}(\alpha_{\mathrm{AP}_t} + \beta_{\mathrm{AP}_t})$
In RUSH, AP_t computes $K_{\mathrm{AU}} = x_{\mathrm{AP}_t}(\alpha_{\mathrm{AP}_t} + \beta_{\mathrm{AP}_t})(A_{\mathrm{UE}} + B_{\mathrm{UE}})$. Based on *Step 10*, *Step 11*, and the composition rule, we have:
Step 12. $\mathrm{AP}_t \models K_{\mathrm{AU}}$
According to *Step 5*, *Assumption 1*, and the jurisdiction rule, we have:
Step 13. $\mathrm{AP}_t \models \mathrm{REQ}_{\mathrm{UE}}$
In RUSH, $\mathrm{REP}_{\mathrm{AP}_t}$ is generated by AP_t. Note that

$$\mathrm{PTK}_{\mathrm{AP}_t} = H_2(\mathrm{REQ}_{\mathrm{UE}} \parallel \mathrm{REP}_{\mathrm{AP}_t} \parallel K_{\mathrm{AU}}).$$

Based on *Step 12*, *Step 13*, and the composition rule, we have:
Step 14. $\mathrm{AP}_t \models \mathrm{PTK}_{\mathrm{AP}_t}$
That is, $\mathrm{AP}_t \models \mathrm{AP}_t \overset{\mathrm{PTK}}{\longleftrightarrow} \mathrm{UE}$ *(Goal 2)*
According to *Message 2*, we have:
Step 15. $\mathrm{UE} \lhd \mathrm{REP}_{\mathrm{AP}_t} \parallel \mathrm{ACK}_{\mathrm{AP}_t}$
In RUSH, the UE checks T_{Curr} to prevent replay attacks. Hence,
Step 16. $\mathrm{UE} \models \sharp(B_{\mathrm{AP}_t})$
According to *Step 16* and the fresh-promotion rule, we have:
Step 17. $\mathrm{UE} \models \sharp(\mathrm{REP}_{\mathrm{AP}_t})$
Based on RUSH, the UE sets $\gamma_{\mathrm{AP}_t} = H_1(ID_{\mathrm{AP}_t} \parallel A_{\mathrm{AP}_t} \parallel B_{\mathrm{AP}_t} \parallel T_{\mathrm{Curr}})$ and checks if $m_{\mathrm{AP}_t}P + \gamma_{\mathrm{AP}_t}A_{\mathrm{AP}_t} = CH_{\mathrm{AP}_t}$. If it is, according to *Step 15*, we have:

Step 18. $\mathrm{UE} \models \mathrm{AP}_t \mid\sim \mathrm{REP}_{\mathrm{AP}_t}$
According to *Step 17*, *Step 18*, and the nonce-verification rule, we have:
Step 19. $\mathrm{UE} \models \mathrm{AP}_t \models \mathrm{REP}_{\mathrm{AP}_t}$
According to *Step 19*, and the decomposition rule, we have:
Step 20. $\mathrm{UE} \models \mathrm{AP}_t \models A_{\mathrm{AP}_t}$
According to *Step 20*, *Assumption 2*, and the jurisdiction rule, we have:
Step 21. $\mathrm{UE} \models A_{\mathrm{AP}_t}$
According to *Step 19*, and the decomposition rule, we have:
Step 22. $\mathrm{UE} \models \mathrm{AP}_t \models B_{\mathrm{AP}_t}$
According to *Step 22*, *Assumption 2*, and the jurisdiction rule, we have:
Step 23. $\mathrm{UE} \models B_{\mathrm{AP}_t}$
Based on *Step 21*, *Step 23*, and the composition rule, we have:
Step 24. $\mathrm{UE} \models A_{\mathrm{AP}_t} + B_{\mathrm{AP}_t}$
In RUSH, the UE chooses $x_{\mathrm{UE}}, \alpha_{\mathrm{UE}}, \beta_{\mathrm{UE}} \in_R \mathbb{Z}_q^*$. Hence,
Step 25. $\mathrm{UE} \models x_{\mathrm{UE}}(\alpha_{\mathrm{UE}} + \beta_{\mathrm{UE}})$
In RUSH, UE computes $K_{\mathrm{UA}} = x_{\mathrm{UE}}(\alpha_{\mathrm{UE}} + \beta_{\mathrm{UE}})(A_{\mathrm{AP}_t} + B_{\mathrm{AP}_t})$. Based on *Step 24*, *Step 25*, and the composition rule, we have:
Step 26. $\mathrm{UE} \models K_{\mathrm{UA}}$
According to *Step 19*, *Assumption 2*, and the jurisdiction rule, we have:
Step 27. $\mathrm{UE} \models \mathrm{REP}_{\mathrm{AP}_t}$
In RUSH, $\mathrm{REQ}_{\mathrm{UE}}$ is generated by UE. Note that

$$\mathrm{PTK}_{\mathrm{UE}} = H_2(\mathrm{REQ}_{\mathrm{UE}} \parallel \mathrm{REP}_{\mathrm{AP}_t} \parallel K_{\mathrm{UA}}).$$

Based on *Step 26*, *Step 27*, and the composition rule, we have:
Step 28. $\mathrm{UE} \models \mathrm{PTK}_{\mathrm{UE}}$
Note that

$$\begin{aligned} K_{\mathrm{AU}} &= x_{\mathrm{AP}_t}(\alpha_{\mathrm{AP}_t} + \beta_{\mathrm{AP}_t})(A_{\mathrm{UE}} + B_{\mathrm{UE}}) \\ &= x_{\mathrm{UE}}(\alpha_{\mathrm{UE}} + \beta_{\mathrm{UE}})(A_{\mathrm{AP}_t} + B_{\mathrm{AP}_t}) = K_{\mathrm{UA}}. \end{aligned}$$

Hence, $\mathrm{UE} \models \mathrm{UE} \overset{\mathrm{PTK}}{\longleftrightarrow} \mathrm{AP}_t$ (*Goal 1*)
Note that $K_{\mathrm{AU}} = K_{\mathrm{UA}} \triangleq K$, according to *Step 26*, we have:
Step 29. $\mathrm{UE} \models \mathrm{UE} \overset{K}{\leftrightarrow} \mathrm{AP}_t$
According to *Message 2*, *Step 15*, *Step 29*, and the message-meaning rule, we have:
Step 30. $\mathrm{UE} \models \mathrm{AP}_t \mid\sim \mathrm{PTK}_{\mathrm{AP}_t} \parallel \mathrm{REP}_{\mathrm{AP}_t}$
According to *Step 17*, and the fresh-promotion rule, we have:
Step 31. $\mathrm{UE} \models \sharp(\mathrm{PTK}_{\mathrm{AP}_t} \parallel \mathrm{REP}_{\mathrm{AP}_t})$
According to *Step 30*, *Step 31*, and the nonce-verification rule, we have:
Step 32. $\mathrm{UE} \models \mathrm{AP}_t \models \mathrm{PTK}_{\mathrm{AP}_t} \parallel \mathrm{REP}_{\mathrm{AP}_t}$
According to *Step 32*, and the decomposition rule, we have:
Step 33. $\mathrm{UE} \models \mathrm{AP}_t \models \mathrm{PTK}_{\mathrm{AP}_t}$
Note that $K_{\mathrm{AU}} = K_{\mathrm{UA}}$, we obtain

$$\mathrm{UE} \models \mathrm{AP}_t \models \mathrm{AP}_t \overset{\mathrm{PTK}}{\leftrightarrow} \mathrm{UE} \qquad (\textit{Goal } 3)$$

According to *Message 3*, we have:

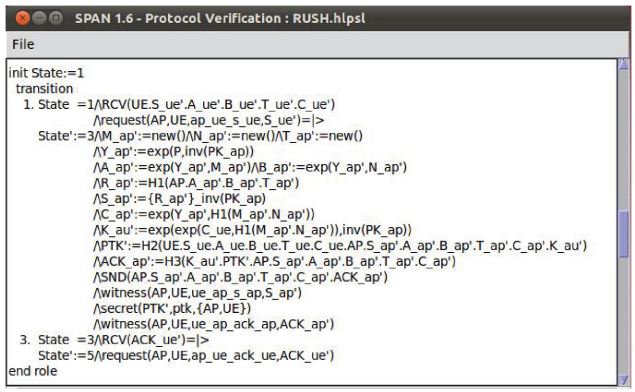


Fig. 8. The role AP's transition based on HLPSL in RUSH.

Step 34. $\mathrm{AP}_t \lhd \mathrm{ACK}_{\mathrm{UE}}$
Note that $K_{\mathrm{AU}} = K_{\mathrm{UA}} \triangleq K$, according to *Step 12*, we have:
Step 35. $\mathrm{AP}_t \models \mathrm{AP}_t \overset{K}{\leftrightarrow} \mathrm{UE}$
According to *Message 3*, *Step 34*, *Step 35*, and the message-meaning rule, we have:
Step 36. $\mathrm{AP}_t \models \mathrm{UE} \mid\sim \mathrm{PTK}_{\mathrm{UE}} \parallel \mathrm{REQ}_{\mathrm{UE}}$
According to *Step 3*, and the fresh-promotion rule, we have:
Step 37. $\mathrm{AP}_t \models \sharp(\mathrm{PTK}_{\mathrm{UE}} \parallel \mathrm{REQ}_{\mathrm{UE}})$
According to *Step 36*, *Step 37*, and the nonce-verification rule, we have:
Step 38. $\mathrm{AP}_t \models \mathrm{UE} \models \mathrm{PTK}_{\mathrm{UE}} \parallel \mathrm{REQ}_{\mathrm{UE}}$
According to *Step 38*, and the decomposition rule, we have:
Step 39. $\mathrm{AP}_t \models \mathrm{UE} \models \mathrm{PTK}_{\mathrm{UE}}$
Note that $K_{\mathrm{UA}} = K_{\mathrm{AU}}$, we obtain

$$\mathrm{AP}_t \models \mathrm{UE} \models \mathrm{UE} \overset{\mathrm{PTK}}{\leftrightarrow} \mathrm{AP}_t \qquad (\textit{Goal } 4)$$

In summary, four goals are achieved. We know that the UE and AP_t believe that they share PTK with each other and the partner is a legal entity. Obviously, the UE's identity is hidden in the whole process and hence anonymity is realized. □

4.2 Formal Verification

In this section, we model RUSH and check its security by AVISPA, which has been widely used for analyzing security protocols [10], [25], [58], [59]. In AVISPA, a modular and expressive formal language, named High-Level Protocol Specification Language (HLPSL), is adopted to specify protocols and validate intended security properties. It is noted that AVISPA integrates different back-ends such as On-the-Fly-Model-Checker (OFMC) and Tree Automata based on Automatic Approximations for the Analysis of Security Protocols (TA4SP) that have implemented a variety of automatic analysis techniques.

4.2.1 Specification of RUSH

In the specification of RUSH based on HLPSL, two basic roles `UE` and `AP` are used to represent the UE and AP_t, respectively. It is important to specify the state transition actions for the UE and AP_t. As illustrated in Fig. 8, we take AP_t as an example to show the state transition. In Fig. 8, SND() and RCV() represent

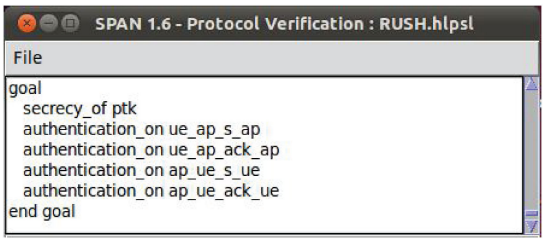


Fig. 9. Analysis goals of the model.

the sending channel and the receiving channel, respectively. For instance, RCV(UE.S_ue'.A_ue'.B_ue'.T_ue'.C_ue') means that AP receives the identity information of the UE and the authentication information $\mathrm{S\_ue}' \parallel \mathrm{A\_ue}' \parallel \mathrm{B\_ue}' \parallel \mathrm{T\_ue}' \parallel \mathrm{C}'_{\mathrm{ue}}$ from UE. In addition, AP sends its own identity, authentication and confirmation information to UE. UE also sends a confirmation $H_3(\mathrm{ACK\_ue}')$ to AP. Note that the Dolev-Yao intruder model is adopted, in which the intruder is afforded enormous capabilities. Concretely, the intruder may intercept, analyze and modify messages, and forward new ones in order to impersonate other agents. In AVISPA, two types of default HLPSL authentication goals strong authentication and weak authentication are considered. The difference lies in that the former allows to check replay attacks, while the latter cannot. As for secrecy, the goal specifies which values should be kept secret between the participants. As shown in Fig. 9, we verify one secrecy and two strong authentication, which are explained as follows.

- The UE authenticates AP_t on the collision $m_{\mathrm{AP}_t}(\mathrm{ue\_ap\_s\_ap})$ and the confirmation information $\mathrm{ACK}_{\mathrm{AP}_t}(\mathrm{ue\_ap\_ack\_ap})$: m_{AP_t} is computed based on secret keys of AP_t.
- AP_t authenticates the UE on the collision $m_{\mathrm{UE}}(\mathrm{ap\_ue\_s\_ue})$ and the confirmation information $\mathrm{ACK}_{\mathrm{UE}}(\mathrm{ap\_ue\_ack\_ue})$: m_{UE} is computed based on secret keys of UE.
- Secrecy of PTK (ptk): The generation of PTK involves secret keys and random values only known by the UE and AP_t. Hence, PTK should be a secret between the UE and AP_t.

4.2.2 Analysis of Results

We adopt the back-end OFMC for the execution test and checking, which can find replay attacks by session compilation. OFMC finds the replay attack even without the second parallel session between two participants. In fact, OFMC first simulates a run of the whole system and then gives the intruder the knowledge retrieved from previous runs. On the other hand, OFMC uses session compilation to check whether honest agents can execute the protocol by performing a search of a passive intruder, then gives the intruder the knowledge of some normal sessions between honest agents. The model checking result in Fig. 10 indicates that RUSH satisfies the goals of two strong authentication and one secrecy. Particularly, the OFMC based result indicates that RUSH can resist replay attacks and man-in-the-middle attacks.

4.3 Further Security Analysis

In this section, we further show that RUSH satisfies various security properties. Because the basic *Impersonation*, *Replay* and *Man-in-the-Middle* attacks are obvious and the resistance ability of RUSH has been ensured by the AVISPA verification, we only show other important properties in the following.

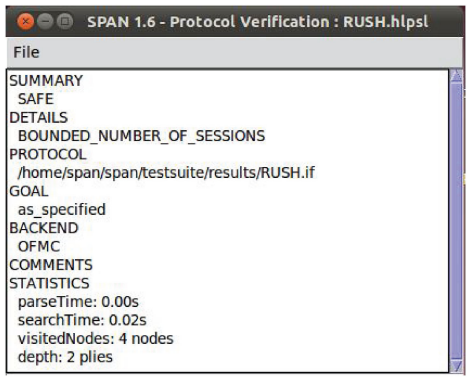


Fig. 10. Results reported by the OFMC back-end.

4.3.1 Mutual Authentication with Key Agreement

As for authentication, AP_t checks the legitimacy of the UE based on Equation (1), of which the soundness is shown as follows.

$$\begin{aligned} m_{\mathrm{UE}}P + \gamma_{\mathrm{UE}}A_{\mathrm{UE}} &= (k_{\mathrm{UE}} - r_{\mathrm{UE}}x_{\mathrm{UE}})P + \gamma_{\mathrm{UE}}A_{\mathrm{UE}} \\ &= k_{\mathrm{UE}}P = (m^*_{\mathrm{UE}} + r^*_{\mathrm{UE}}x_{\mathrm{UE}})P \\ &= CH_{\mathrm{UE}}. \end{aligned}$$

Similarly, Equation (5) is sound. As for key agreement, it's noted that

$$\begin{aligned} K_{\mathrm{AU}} &= x_{\mathrm{AP}_t}(\alpha_{\mathrm{AP}_t} + \beta_{\mathrm{AP}_t})(A_{\mathrm{UE}} + B_{\mathrm{UE}}) \\ &= x_{\mathrm{AP}_t}(\alpha_{\mathrm{AP}_t} + \beta_{\mathrm{AP}_t})(\alpha_{\mathrm{UE}}Y_{\mathrm{UE}} + \beta_{\mathrm{UE}}Y_{\mathrm{UE}}) \\ &= x_{\mathrm{UE}}x_{\mathrm{AP}_t}(\alpha_{\mathrm{AP}_t} + \beta_{\mathrm{AP}_t})(\alpha_{\mathrm{UE}} + \beta_{\mathrm{UE}})P \\ &= x_{\mathrm{UE}}(\alpha_{\mathrm{UE}} + \beta_{\mathrm{UE}})(\alpha_{\mathrm{AP}_t}Y_{\mathrm{AP}_t} + \beta_{\mathrm{AP}_t}Y_{\mathrm{AP}_t}) \\ &= x_{\mathrm{UE}}(\alpha_{\mathrm{UE}} + \beta_{\mathrm{UE}})(A_{\mathrm{AP}_t} + B_{\mathrm{AP}_t}) = K_{\mathrm{UA}}. \end{aligned}$$

According to the protocol, $\mathrm{PTK} = \mathrm{PTK}_{\mathrm{AP}_t} = \mathrm{PTK}_{\mathrm{UE}}$ is the pairwise transient key.

4.3.2 Anonymity

In the handover authentication phase of RUSH, the UE's pseudo identity $\mathrm{PID}_{\mathrm{UE}}$ is adopted instead of the actual identity ID_{UE}. In addition, in the network registration phase, the AAA server stores $CH_{\mathrm{UE}} \parallel T_{\mathrm{Exp}}$ in the blockchain, which does not leak ID_{UE} based on the properties of chameleon hash functions. In general, the UE's identity is hidden and the anonymity is achieved.

4.3.3 Traceability

Suppose the UE is detected to have behaved maliciously under the pseudo identity $\mathrm{PID}_{\mathrm{UE}}$. The AAA server is able to trace the UE and reveal the actual identity ID_{UE} as follows after specifying the malicious authentication message $\mathrm{REQ}_{\mathrm{UE}} = \mathrm{PID}_{\mathrm{UE}} \parallel m_{\mathrm{UE}} \parallel A_{\mathrm{UE}} \parallel B_{\mathrm{UE}} \parallel T_{\mathrm{Curr}}$.

- Compute $\gamma_{\mathrm{UE}} = H_1(\mathrm{PID}_{\mathrm{UE}} \parallel A_{\mathrm{UE}} \parallel B_{\mathrm{UE}} \parallel T_{\mathrm{Curr}})$.
- Set $CH_{\mathrm{UE}} = m_{\mathrm{UE}}P + \gamma_{\mathrm{UE}}A_{\mathrm{UE}}$.

TABLE 4
Comparison of Functionality

Scheme \ Functionality	Inter-domain	KEF	Formal Security Proof	Anonymity	Traceability	Robustness	PFS	MKFS	KRS
[13]	No	No	No	No	–	–	No	No	No
[34]	No	No	No	No	–	–	Yes	Yes	No
[24]	No	No	No	No	–	–	Yes	Yes	No
Scheme I [20]	No	No	No	No	–	–	Yes	Yes	No
Scheme II [20]	No	No	No	Yes	Yes	No	Yes	Yes	No
[15]	No	No	No	Yes	Yes	No	No	No	–
[14]	No	No	No	Yes	Yes	No	Yes	Yes	No
[25]	No	Yes	Yes	No	–	–	Yes	Yes	No
[35]	No	Yes	No	No	–	–	Yes	Yes	No
[18]	No	Yes	No	Yes	Yes	No	No	No	No
[17]	No	Yes	Yes	Yes	Yes	No	Yes	Yes	No
RUSH	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

† ***Inter-domain:*** *handover between domains of different core networks.* ***KEF:*** *the protocol is key escrow free.* ***Formal Security Proof:*** *security analysis is based on the BAN-logic or provable security techniques.* ***Anonymity:*** *the UE's identity is anonymous in handover authentication.* ***Traceability:*** *anonymous UE should be traced by revealing the UE's actual identity if malicious behaviors exist.* ***Robustness:*** *an honest UE cannot be framed in traceability even if the AAA server or PKG is compromised.* ***PFS:*** *perfect forward secrecy.* ***MKFS:*** *master key forward secrecy.* ***KRS:*** *known randomness secrecy, i.e., the session key is secret even if session-specific temporary information is leaked.*
‡ *The symbol "–" represents the functionality is not involved.*

- Get the transaction indexed by $\mathsf{TXID}_{\mathrm{UE}}$ from the blockchain and ensure that the transaction outputs CH_{UE}.
- Based on $\mathsf{TXID}_{\mathrm{UE}}$, get the item $ID_{\mathrm{UE}} \parallel \mathsf{TXID}_{\mathrm{UE}}$ from the local storage and output ID_{UE}.

4.3.4 Robustness

Suppose the AAA server is compromised and wants to frame an honest user UE^*. Specifically, based on a malicious behavior $\mathrm{REQ}_{\mathrm{UE}} = \mathrm{PID}_{\mathrm{UE}} \parallel m_{\mathrm{UE}} \parallel A_{\mathrm{UE}} \parallel B_{\mathrm{UE}} \parallel T_{\mathrm{Curr}}$, suppose the AAA server gets the identity ID_{UE} after performing traceability. However, the AAA server outputs $ID_{\mathrm{UE}^*} \neq ID_{\mathrm{UE}}$ to frame UE^*. In this case, UE^* can prove to a third party that it is framed by the AAA server as follows:

- Compute $CH_{\mathrm{UE}^*} = k_{\mathrm{UE}^*} P$.
- Present $(ID_{\mathrm{UE}^*} \parallel CH_{\mathrm{UE}^*}, \sigma^*)$, where $\sigma^* = \mathsf{sig}_{\mathrm{AAA}}(ID_{\mathrm{UE}^*} \parallel CH_{\mathrm{UE}^*})$ is issued by the AAA server in the network registration phase and it is locally stored by UE^*.

The third party does the following:

- Compute $\gamma_{\mathrm{UE}} = H_1(\mathrm{PID}_{\mathrm{UE}} \parallel A_{\mathrm{UE}} \parallel B_{\mathrm{UE}} \parallel T_{\mathrm{Curr}})$.
- Set $CH_{\mathrm{UE}} = m_{\mathrm{UE}} P + \gamma_{\mathrm{UE}} A_{\mathrm{UE}}$.
- Ensure CH_{UE^*} is stored on the blockchain by the AAA server.
- Think that UE^* is framed by the AAA server if and only if $\mathsf{ver}_{\mathrm{AAA}}(ID_{\mathrm{UE}^*} \parallel CH_{\mathrm{UE}^*}, \sigma^*) = \mathsf{true}$ and $CH_{\mathrm{UE}^*} \neq CH_{\mathrm{UE}}$.

4.3.5 Perfect Forward Secrecy (PFS) and Master Key Forward Secrecy (MKFS)

Suppose the long-term secret keys of the UE and AP_t are compromised. Based on Equations (2) and (6), it follows that the preceding PTK cannot be derived. The reason is that random values from the UE and AP_t are used in the generation of PTK. Similarly, if the master key of the AAA server is stolen, the preceding PTK still cannot be derived.

4.3.6 Known Randomness Secrecy (KRS)

Suppose session-specific temporary data are known to adversaries. Concretely, in RUSH, suppose random values α_{UE}, β_{UE}, α_{AP_t} and β_{AP_t} are leaked. According to Equations (2) and (6), it follows that the corresponding PTK cannot be derived. The reason is that two random values from each participant and two secret keys are involved in PTK.

4.3.7 Key Escrow Freeness (KEF)

According to the network registration phase of RUSH, we know that users' secret keys are completely chosen by themselves. Therefore, RUSH is a key escrow free handover authentication protocol.

4.3.8 Inter-EPC Handover Authentication

In RUSH, the UE can handover between domains of different EPCs similar to the case of intra-domain handover, which is enabled by the trapdoor collision property of chameleon hash functions and the global consistency of records in the blockchain.

5 Functionality and Performance Evaluation

In this section, we analyze the functionality and performance of RUSH and compare it with previous schemes.

5.1 Functionality Comparison

Table 4 shows the functionality comparison of RUSH and related approaches [13], [14], [15], [17], [18], [20], [24], [25], [34], [35]. Obviously, only RUSH supports handover between domains of different core networks without any globally accessible entities, which is realized based on the properties of chameleon hash functions and blockchains. Other schemes [20], [24] have to introduce a globally accessible entity such as a PKG to realize inter-EPC handover authentication. The key escrow problem is only addressed in RUSH and schemes [17], [18], [25], [35]. As for formal security proofs, only RUSH is proven secure based on the BAN-logic and schemes [17], [25]

TABLE 5
Comparison of Transmission Costs

Scheme \ Transmission Cost	$C_{\text{UE-AP}}$	$C_{\text{AP-AP}}$	$C_{\text{AP-AAA}}$
From E-UTRAN to trusted non-3GPP access [5]	5δ	0	$4e$
From E-UTRAN to untrusted non-3GPP access [5]	8δ	0	$4e$
From non-3GPP access to E-UTRAN [5]	3δ	0	$3e$
[8]	4δ	0	$2e$
[29]	4δ	2ε	0
[13], [17], [18], [20], [24], [25], [34], [35] and RUSH	3δ	0	0

are proved to be secure in the random oracle model. Note that both UE's identity anonymity and UE's traceability are realized in RUSH and schemes [14], [15], [17], [18], [20]. In particular, only RUSH achieves robustness, which prevents compromised authorities from framing honest UEs and hence is an indispensable property in practice. The scheme [14] aims to realize robustness by introducing many other entities to distribute the ability of the AAA server based on the secret sharing technique. However, it is assumed that the AAA server will honestly erase some random values and the number of collusive entities is less than a given number. In addition, similar to the scheme [15], the scheme [14] suffers from a security flaw because secret-related values are directly used in arithmetic multiplication operations. Finally, RUSH satisfies PFS, MKFS, and KRS which is not achieved by any other scheme.

5.2 Transmission Overhead

We use δ, ε and e to denote the costs of transmitting a single message between a UE and AP, between two APs, and between an AP and AAA server (or MME), respectively. As shown in Table 5, we compare the transmission cost of RUSH with those of the handover schemes [5], [8], [13], [17], [18], [20], [24], [25], [29], [34], [35]. In Table 5, $C_{\text{UE-AP}}$ is the transmission cost for all the messages between the UE and AP, $C_{\text{AP-AP}}$ is the transmission cost for all the messages between APs, and $C_{\text{AP-AAA}}$ is the transmission cost for all the messages between the AP and AAA server (or MME). From Table 5, similar to the schemes [13], [17], [18], [20], [24], [25], [34], [35], RUSH only requires three message exchanges between the UE and AP without needing to communicate with the AAA server or to establish communication between APs. Its cost is hence much lower than those of other schemes. In RUSH, the UE is able to authenticate itself to AP in one message after it is admitted to the network, and two messages suffice to realize mutual authentication with key agreement. In fact, the third message just plays a role of confirmation for key agreement between the UE and AP.

TABLE 6
Computation Costs of the Primitive Cryptography Operations (ms)

User \ Computation Cost	T_P	T_E	T_{SM}	T_{MSM}	T_{RV}
UE	2.87	0.225	0.2025	0.2532	0.127
AP_t	0.7616	0.0337	0.03	0.0375	0.019

5.3 Authentication Cost

We evaluate the performance of the handover authentication procedure of RUSH based on the computation cost and the communication overhead. Since a 224-bit ECC key offers more or less the same level of security as a 2048-bit RSA key [60], we let p be a prime of length $\ell_p = 2048$ bits and q a prime of length $\ell_q = 224$ bits, which keeps up with the strength requirement of secret keys in 5G HetNets.

5.3.1 Computation Cost

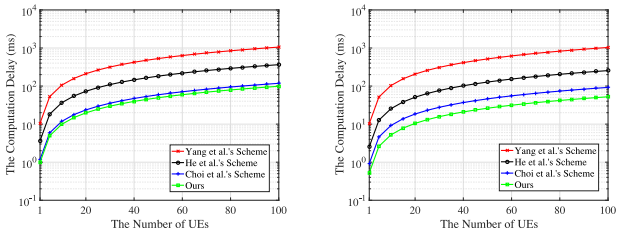
Table 6 shows the computation cost of the primitive cryptography operations, which has been investigated in [9] using OpenSSL with two cores on Intel i5-2500 @ 3.30 GHz as the UE and an Intel i7-6600U CPU @ 2.60 Ghz as AP_t. In Table 6, T_P, T_E, T_{SM}, T_{MSM} and T_{RV} represent the computation cost of a pairing operation, a modular exponentiation, an elliptic curve scalar multiplication, a multi elliptic curve scalar multiplication, and a RSA signature verification, respectively. Other operations such as modular multiplication and hash are neglected. Note that $T_{MSM} = 1.25T_{SM}$ [20], [61] and the ECDSA verification time is equal to T_{MSM} [62], [63]. The computation costs of schemes [13], [17], [18], [20], [24], [25], [34], [35] and RUSH are analyzed in Table 7. Fig. 11 compares the total computation costs of the UE and AP_t in the anonymous handover authentication schemes including Yang et al.'s scheme II [20], He et al.'s scheme [17], Choi et al.'s scheme [18] and RUSH. To give a more explicit comparison, the vertical axis adopts the log scale. Note that the schemes [14], [15]

TABLE 7
Comparison of Computation Costs in Secure Handover Authentication (ms)

Scheme \ Computation Cost	T_{UE}	T_{AP_t}	T_{tot}	$T_{\text{UE-opt}}$	$T_{\text{AP}_t\text{-opt}}$	$T_{\text{tot-opt}}$
[13]	$2T_P + 2T_{SM} = 6.145$	$2T_P + 2T_{SM} = 1.5832$	7.7282	$T_P = 2.87$	$T_P = 0.7616$	3.6316
[34]	$T_P + 3T_{SM} = 3.4775$	$T_P + 3T_{SM} = 0.8516$	4.3291	$T_P + 2T_{SM} = 3.275$	$T_P + 2T_{SM} = 0.8216$	4.0966
[24]	$3T_{SM} + T_{MSM} = 0.8607$	$3T_{SM} + T_{MSM} = 0.1275$	0.9882	$2T_{SM} + T_{MSM} = 0.6582$	$2T_{SM} + T_{MSM} = 0.0975$	0.7557
Scheme I [20]	$3T_{SM} + T_{MSM} = 0.8607$	$3T_{SM} + T_{MSM} = 0.1275$	0.9882	$2T_{SM} + T_{MSM} = 0.6582$	$2T_{SM} + T_{MSM} = 0.0975$	0.7557
Scheme II [20]	$3T_P + 5T_{SM} + 3T_{MSM} = 10.3821$	$T_{SM} + 4T_{MSM} = 0.18$	10.5621	$3T_P + 4T_{SM} + 3T_{MSM} = 10.1796$	$4T_{MSM} = 0.15$	10.3296
[25]	$3T_{SM} + 2T_{MSM} = 1.1139$	$3T_{SM} + 2T_{MSM} = 0.165$	1.2789	$T_{SM} + 2T_{MSM} = 0.7089$	$T_{SM} + 2T_{MSM} = 0.105$	0.8139
[35]	$3T_{SM} + 2T_{MSM} = 1.1139$	$3T_{SM} + 2T_{MSM} = 0.165$	1.2789	$T_{SM} + 2T_{MSM} = 0.7089$	$T_{SM} + 2T_{MSM} = 0.105$	0.8139
[18]	$4T_E + T_{RV} = 1.027$	$4T_E + T_{RV} = 0.1538$	1.1808	$3T_E + T_{RV} = 0.802$	$3T_E + T_{RV} = 0.1201$	0.9221
[17]	$2T_E + 4T_{SM} = 1.26$	$3T_P + T_E + T_{SM} + T_{MSM} = 2.386$	3.646	$T_E = 0.225$	$3T_P + T_{MSM} = 2.3223$	2.5473
Our RUSH	$3T_{SM} + T_{MSM} = 0.8607$	$3T_{SM} + T_{MSM} = 0.1275$	0.9882	$T_{SM} + T_{MSM} = 0.4557$	$T_{SM} + T_{MSM} = 0.0675$	0.5232

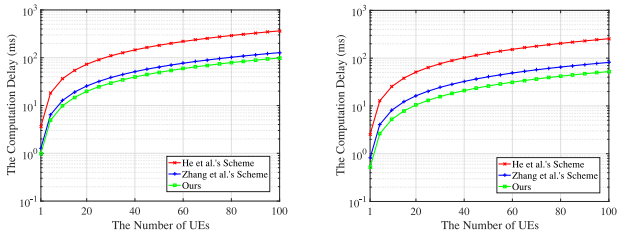
† *T_{UE} denotes the computation time of UE without optimization, T_{AP_t} the computation time of AP_t without optimization, and T_{tot} the total computation time without optimization.*

‡ *$T_{\text{UE}-\text{opt}}$ denotes the computation time of UE optimized by pre-computation, $T_{\text{AP}_t-\text{opt}}$ the computation time of AP_t optimized by pre-computation, and $T_{\text{tot}-\text{opt}}$ the total computation time optimized by pre-computation.*



(a) Not optimized (b) Optimized by pre-computation

Fig. 11. Computation costs of anonymous handover authentication.



(a) Not optimized (b) Optimized by pre-computation

Fig. 12. Computation costs of formally proved secure handover authentication.

suffer from a security flaw and hence are not considered in Fig. 11. Obviously, RUSH is the most efficient one and for handover of a single UE it only requires 0.5232 ms and 0.9882 ms with and without pre-computation, respectively. Fig. 12 compares the total computation costs of the UE and AP_t in the formally proved secure handover authentication schemes including He et al.'s scheme [17], Zhang et al.'s scheme [25] and RUSH. From Fig. 12 and Table 7, RUSH is obviously the most efficient one.

5.3.2 Communication Cost

As for the communication cost, we analyze and compare the anonymous handover authentication schemes including Yang et al.'s scheme II [20], He et al.'s scheme [17], Choi et al.'s scheme [18] and RUSH. In these schemes, the UE merely communicates with AP_t without contacting the AAA server or establishing trust relationships among APs. The message sizes are given in Table 8 and are compared in Fig. 13. As mentioned before, $\ell_p = 2048$ bits and $\ell_q = 224$ bits. Note that $\ell_{\mathrm{id}} = 4$ bytes and $\ell_{\mathrm{pid}} = 4$ bytes represent the length of the identity and the pseudo identity, respectively. The timestamp is of length $\ell_{\mathrm{ts}} = 4$ bytes and the output length of the adopted hash function $H_3 = \mathrm{SHA\text{-}1}$ is 160 bits. The total size of messages in RUSH is 224 bytes, which is far less than those of the other schemes.

5.3.3 Storage Cost

Based on the analysis in Section 3.2, we know each user locally stores all the block headers. In addition, a UE (resp. an AP) stores the output data of OP_RETURN in transactions associated with APs (resp. UEs) created by the AAA servers. Take the Bitcoin blockchain as an example, a block header is of $\ell_{\mathrm{header}} = 80$ bytes. The total number of blocks is $N_{0,\mathrm{header}} = 552048$ on December 1, 2018. Since blocks are mined on average every 10 minutes, $R_{\mathrm{block}} = 52560$ blocks are mined per year on average.

The size of an AP's data $CH_{\mathrm{AP}} \parallel T_{\mathrm{Exp}}$ on the blockchain is $\ell_{\mathrm{AP}} = 32$ bytes. Suppose the deployment of base stations has the annual rate of increase $R_{\mathrm{AP}} = 6\%$. According to the estimation from Mobile World Live [64], the total number of APs worldwide is about $N_{0,\mathrm{AP}} = 7.3$ million on December 1, 2018. After i years, the storage cost of the UE is approximately $S_{i,\mathrm{UE}}$ bytes, where

$$S_{i,\mathrm{UE}} = \ell_{\mathrm{header}} \cdot (N_{0,\mathrm{header}} + R_{\mathrm{block}} \cdot i) + \ell_{\text{AP-data}} \cdot N_{0,\mathrm{AP}} \cdot (1 + R_{\mathrm{AP}})^i.$$

Similarly, the size of a UE's data on the blockchain is $\ell_{\mathrm{UE}} = 32$ bytes. According to the Newzoo Global Mobile Market Report 2018 [65], the annual increase rate of mobile users worldwide is about $R_{\mathrm{UE}} = 3.8\%$ and the total number is about $N_{0,\mathrm{UE}} = 3.3$ billion on December 1, 2018. After i years, the storage cost of the AP is approximately $S_{i,\mathrm{AP}}$ bytes, where

$$S_{i,\mathrm{AP}} = \ell_{\mathrm{header}} \cdot (N_{0,\mathrm{header}} + R_{\mathrm{block}} \cdot i) + \ell_{\text{UE-data}} \cdot N_{0,\mathrm{UE}} \cdot (1 + R_{\mathrm{UE}})^i.$$

The storage cost of the UE and AP within the next decade is illustrated in Fig. 14. Obviously, because the total number of UEs will reach approximately 4.79 billion after ten years, the storage cost of the AP will increase to 137.65 GB. It is noted that the storage cost on the UE's side will be less than 475 MB over the next ten years. The storage cost will be further reduced with the advancements in blockchain technologies.

5.4 Discussion: The Case of Malicious UEs

With increasing number of mobile terminals in practice, malicious UEs may severely affect security and performance of the communication system. It is desirable for AP_t to effectively detect malicious UEs. However, in many existing schemes, AP_t cannot authenticate UEs until the third authentication message is received. In RUSH, the computation cost and the communication cost for AP_t to detect malicious UEs are significantly reduced in that AP_t only needs to perform partial computations after receiving the first message. In the case of malicious UEs, the computation costs of RUSH and the schemes [17], [18], [20], [25] are shown in Table 9 and Fig. 15. For the sake of clarity, the vertical axis adopts the log scale in Fig. 15. Note that anonymity is realized in Yang

TABLE 8
Comparison of the Message Sizes in Secure Anonymous Handover Authentication (byte)

Scheme \ Message	Message 1	Message 2	Message 3	Total Message
Scheme II [20]	$\ell_q + \ell_{\mathrm{id}} + \ell_{\mathrm{pid}} + \ell_{\mathrm{ts}} = 40$	$3\ell_q + \ell_{\mathrm{id}} + \ell_{\mathrm{ts}} = 92$	$7\ell_q + \ell_{\mathrm{ts}} = 200$	332
[18]	$2\ell_p + \ell_q + \ell_{\mathrm{pid}} + \ell_{\mathrm{ts}} = 548$	$2\ell_p + \ell_q + \ell_{\mathrm{id}} + \ell_{\mathrm{ts}} + \ell_h = 568$	$\ell_h = 20$	1136
[17]	$\ell_p + 3\ell_q + \ell_{\mathrm{pid}} + \ell_{\mathrm{ts}} = 348$	$\ell_p + \ell_{\mathrm{id}} + \ell_{\mathrm{ts}} + \ell_h = 284$	$\ell_h = 20$	652
RUSH	$3\ell_q + \ell_{\mathrm{pid}} + \ell_{\mathrm{ts}} = 92$	$3\ell_q + \ell_{\mathrm{id}} + \ell_{\mathrm{ts}} + \ell_h = 112$	$\ell_h = 20$	224

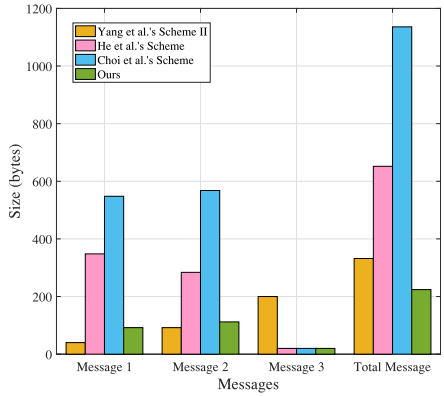


Fig. 13. Comparison of authentication message sizes.

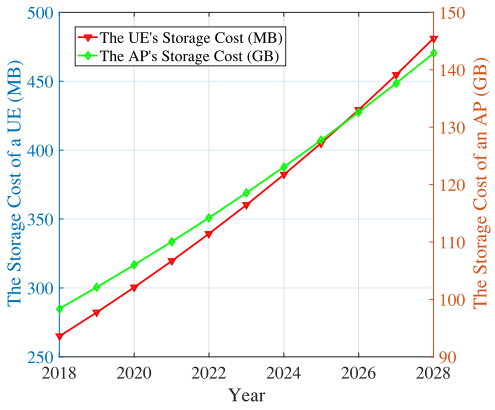


Fig. 14. The storage cost of the UE and AP within the next decade *(The total number of UEs worldwide increases from 3.3 billion to 4.79 billion).*

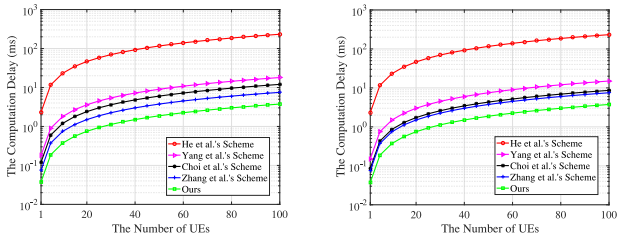


Fig. 15. Computation costs of AP_t for detecting malicious UEs.

TABLE 9
Comparison of Computation Costs for Detecting Malicious UEs (ms)

Scheme \ Computation Cost	T_{AP_t}	$T_{\mathrm{AP}_t\text{-opt}}$
Scheme II [20]	$T_{SM} + 4T_{MSM} = 0.18$	$4T_{MSM} = 0.15$
[25]	$2T_{MSM} = 0.075$	$2T_{MSM} = 0.075$
[18]	$3T_E + T_{RV} = 0.1201$	$2T_E + T_{RV} = 0.0864$
[17]	$3T_P + T_{MSM} = 2.3223$	$3T_P + T_{MSM} = 2.3223$
Our RUSH	$T_{MSM} = 0.0375$	$T_{MSM} = 0.0375$

et al.'s scheme [20], Choi et al.'s scheme [18], He et al.'s scheme [17], and RUSH, and formal security proof is given in [17], Zhang et al.'s scheme [25], and RUSH. It easily follows that RUSH is more efficient than the other schemes in the detection of malicious UEs.

6 Conclusions and Future Work

In this paper, towards secure and efficient handover authentication in 5G HetNets, we proposed a handover authentication protocol called RUSH based on chameleon hashing and blockchain technologies. RUSH is characterized by several unique and attractive properties including robust traceability, known randomness secrecy, and inter-domain handover without needing any globally accessible entity. Specifically, it supports the handover among LTE E-UTRAN, trusted non-3GPP access, untrusted non-3GPP access, and 5G new radio regardless of the consistency of EPCs and 5GCs. Our formal security proof based on the BAN-logic and formal verification based on the AVISPA tool indicated that RUSH can resist various malicious attacks. Our extensive efficiency analysis showed that RUSH is very efficient in terms of computation and communication costs. In future research, it would be interesting to design secure and mobile user-friendly handover authentication protocols in 5G HetNets.

Acknowledgments

The authors would like to thank the anonymous associate Editor and referees for their invaluable suggestions. This research is supported by the National Key R&D Program of China (No. 2017YFB0802000), the AXA Research Fund, the National Natural Science Foundation of China (Nos. 61772418, 61472472), the Key Research and Development Program of Shaanxi (No. 2019KW-053), Sichuan Science and Technology Program (No.2017GZDZX0002), and the Natural Science Basic Research Plan in Shaanxi Province of China (Nos. 2018JZ6001, 2015JQ6236). Yinghui Zhang is supported by New Star Team of Xi'an University of Posts & Telecommunications (2016-02).

References

[1] E. Hossain and M. Hasan, "5g cellular: Key enabling technologies and research challenges," *IEEE Instrum. Meas. Mag.*, vol. 18, no. 3, pp. 11–21, Jun. 2015.
[2] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. Soong, and J. C. Zhang, "What will 5g be?" *IEEE J. Select. Areas Commun.*, vol. 32, no. 6, pp. 1065–1082, Jun. 2014.
[3] X. Duan and X. Wang, "Authentication handover and privacy protection in 5g hetnets using software-defined networking," *IEEE Commun. Mag.*, vol. 53, no. 4, pp. 28–35, Apr. 2015.
[4] W. Bao and B. Liang, "Stochastic geometric analysis of user mobility in heterogeneous wireless networks," *IEEE J. Select. Areas in Communications*, vol. 33, no. 10, pp. 2212–2225, Oct. 2015.
[5] *Technical Specification Group Services and System Aspects; Architecture enhancements for non-3GPP accesses (Release 15), 3GPP TS 23.402 V15.2.0, 2017*, 3rd Generation Partnership Project Std.
[6] S. R. Hussain, O. Chowdhury, S. Mehnaz, and E. Bertino, "Lteinspector: A systematic approach for adversarial testing of 4g lte," in *Proc. Netw. Distrib. Syst. Secur.*, 2018, pp. 1–15.
[7] S. R. Hussain, S. Mehnaz, S. Nirjon, and E. Bertino, "Secure seamless bluetooth low energy connection migration for unmodified iot devices," *IEEE Trans. Mobile Comput.*, vol. 17, no. 4, pp. 927–944, Apr. 2018.
[8] S. Pack and Y. Choi, "Fast handoff scheme based on mobility prediction in public wireless lan systems," *IEE Proc.-Commun.*, vol. 151, no. 5, pp. 489–495, 2004.
[9] M. J. Alam and M. Ma, "Dc and comp authentication in lte-advanced 5g hetnet," in *Proc. IEEE Global Commun. Conf.*, 2017, pp. 1–6.
[10] V. Sharma, I. You, F.-Y. Leu, and M. Atiquzzaman, "Secure and efficient protocol for fast handover in 5g mobile xhaul networks," *J. Netw. Comput. Appl.*, vol. 102, pp. 38–57, 2018.

[11] H. Wang and A. R. Prasad, "Fast authentication for inter-domain handover," in *Proc. Int. Conf. Telecommun.*, 2004, pp. 973–982.
[12] J. Choi and S. Jung, "A secure and efficient handover authentication based on light-weight diffie-hellman on mobile node in fmipv6," *IEICE Trans. Commun.*, vol. 91, no. 2, pp. 605–608, 2008.
[13] Y. Kim, W. Ren, J.-Y. Jo, Y. Jiang, and J. Zheng, "Sfric: A secure fast roaming scheme in wireless lan using id-based cryptography," in *Proc. IEEE Int. Conf. Commun.*, 2007, pp. 1570–1575.
[14] A. Fu, N. Qin, Y. Wang, Q. Li, and G. Zhang, "Nframe: A privacy-preserving with non-frameability handover authentication protocol based on (t, n) secret sharing for lte/lte-a networks," *Wireless Netw.*, vol. 23, no. 7, pp. 2165–2176, 2017.
[15] D. He, C. Chen, S. Chan, and J. Bu, "Secure and efficient handover authentication based on bilinear pairing functions," *IEEE Trans. Wireless Commun.*, vol. 11, no. 1, pp. 48–53, Jan. 2012.
[16] R. Ma, J. Cao, D. Feng, H. Li, Y. Zhang, and X. Lv, "Ppsha: Privacy preserving secure handover authentication scheme for all application scenarios in lte-a networks," *Ad Hoc Netw.*, vol. 87, pp. 49–60, May 2019.
[17] D. He, D. Wang, Q. Xie, and K. Chen, "Anonymous handover authentication protocol for mobile wireless networks with conditional privacy preservation," *Sci. China Inf. Sci.*, vol. 60, no. 5, pp. 1–17, 2017.
[18] J. Choi and S. Jung, "A handover authentication using credentials based on chameleon hashing," *IEEE Commun. Lett.*, vol. 14, no. 1, pp. 54–56, Jan. 2010.
[19] Y. Zhang, X. Chen, J. Li, H. Li, and F. Li, "Attribute-based data sharing with flexible and direct revocation in cloud computing," *KSII Trans. Internet Inf. Syst.*, vol. 8, no. 11, pp. 4028–4049, 2014.
[20] G. Yang, Q. Huang, D. S. Wong, and X. Deng, "Universal authentication protocols for anonymous wireless communications," *IEEE Trans. Wireless Commun.*, vol. 9, no. 1, pp. 168–174, Jan. 2010.
[21] Z. Cheng, M. Nistazakis, R. Comley, and L. Vasiu, "On the indistinguishability-based security model of key agreement protocols-simple cases," *IACR Cryptology ePrint Archive, 2005/129*, pp. 1–39, 2005. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.59.5945&rep=rep1&type=pdf
[22] T. K. Mandt and C. H. Tan, "Certificateless authenticated two-party key agreement protocols," in *Proc. Annu. Asian Comput. Sci. Conf.*, 2006, pp. 37–44.
[23] L. Chen and C. Kudla, "Identity based authenticated key agreement protocols from pairings," in *Proc. 16th IEEE Proc. Comput. Secur. Found. Workshop*, 2003, pp. 219–233.
[24] J. Cao, M. Ma, and H. Li, "An uniform handover authentication between e-utran and non-3gpp access networks," *IEEE Trans. Wireless Commun.*, vol. 11, no. 10, pp. 3644–3650, Oct. 2012.
[25] Y. Zhang, X. Chen, J. Li, and H. Li, "Generic construction for secure and efficient handoff authentication schemes in eap-based wireless networks," *Comput. Netw.*, vol. 75, pp. 192–211, 2014.
[26] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008. [Online]. Available: http://bitcoin.org/bitcoin.pdf
[27] Y. Zhang, R. Deng, X. Liu, and D. Zheng, "Outsourcing service fair payment based on blockchain and its applications in cloud computing," *IEEE Trans. Serv. Comput.*, 2018. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8428431
[28] A. Mishra, M. H. Shin, N. L. Petroni, T. C. Clancy, and W. A. Arbaugh, "Proactive key distribution using neighbor graphs," *IEEE Wireless Commun.*, vol. 11, no. 1, pp. 26–36, Feb. 2004.
[29] C. Zhang, R. Lu, P.-H. Ho, and A. Chen, "A location privacy preserving authentication scheme in vehicular networks," in *Proc. IEEE Wireless Commun. Netw. Conf.*, 2008, pp. 2543–2548.
[30] L. Cai, S. Machiraju, and H. Chen, "Capauth: A capability-based handover scheme," in *Proc. 30th IEEE Int. Conf. Comput. Commun.*, 2010, pp. 1–5.
[31] A. Fu, S. Lan, B. Huang, Z. Zhu, and Y. Zhang, "A novel group-based handover authentication scheme with privacy preservation for mobile wimax networks," *IEEE Commun. Lett.*, vol. 16, no. 11, pp. 1744–1747, Nov. 2012.
[32] Z. Haddad, M. Mahmoud, I. A. Saroit, and S. Taha, "Secure and efficient uniform handover scheme for lte-a networks," in *Proc. Wireless Commun. Netw. Conf.*, 2016, pp. 1–6.
[33] D. He, J. Bu, S. Chan, C. Chen, and M. Yin, "Privacy-preserving universal authentication protocol for wireless communications," *IEEE Trans. Wireless Commun.*, vol. 10, no. 2, pp. 431–436, Feb. 2011.
[34] Y. Zhang, X. Chen, H. Li, and J. Cao, "Identity-based construction for secure and efficient handoff authentication schemes in wireless networks," *Secur. Commun. Netw.*, vol. 5, no. 10, pp. 1121–1130, 2012.
[35] Q. Han, Y. Zhang, X. Chen, H. Li, and J. Quan, "Efficient and robust identity-based handoff authentication in wireless networks," in *Proc. Int. Conf. Netw. Syst. Secur.*, 2012, pp. 180–191.
[36] X. Cao, W. Kou, and X. Du, "A pairing-free identity-based authenticated key agreement protocol with minimal message exchanges," *Inf. Sci.*, vol. 180, no. 15, pp. 2895–2903, 2010.
[37] J. Cao, H. Li, M. Ma, Y. Zhang, and C. Lai, "A simple and robust handover authentication between henb and enb in lte networks," *Comput. Netw.*, vol. 56, no. 8, pp. 2119–2131, 2012.
[38] E. J. Yoon, M. K. Khan, and K. Y. Yoo, "Cryptanalysis of a handover authentication scheme using credentials based on chameleon hashing," *IEICE Trans. Inf. Syst.*, vol. 93, no. 12, pp. 3400–3402, 2010.
[39] X. Chen, F. Zhang, W. Susilo, and Y. Mu, "Efficient generic on-line/off-line signatures without key exposure," in *Applied Cryptography and Network Security*. New York, NY, USA: Springer, 2007, pp. 18–30.
[40] A. Moinet, B. Darties, and J.-L. Baril, "Blockchain based trust & authentication for decentralized sensor networks," *arXiv: 1706.01730*, 2017. [Online]. Available: https://arxiv.org/pdf/1706.01730.pdf
[41] T. Sanda and H. Inaba, "Proposal of new authentication method in wi-fi access using bitcoin 2.0," in *Proc. 5th Global Conf. Consumer Electron.*, 2016, pp. 1–5.
[42] Y. Zhang, R. H. Deng, X. Liu, and D. Zheng, "Blockchain based efficient and robust fair payment for outsourcing services in cloud computing," *Inf. Sci.*, vol. 462, pp. 262–277, 2018.
[43] Y. Zhang, R. H. Deng, J. Shu, K. Yang, and D. Zheng, "Tkse: Trustworthy keyword search over encrypted data with two-side verifiability via blockchain," *IEEE Access*, vol. 6, pp. 31077–31087, Jun. 2018.
[44] E. K. Kogias, P. Jovanovic, N. Gailly, I. Khoffi, L. Gasser, and B. Ford, "Enhancing bitcoin security and performance with strong consistency via collective signing," in *Proc. 25th USENIX Secur. Symp.*, 2016, pp. 279–296.
[45] A. Tomescu and S. Devadas, "Catena: Efficient non-equivocation via bitcoin," in *Proc. 38th IEEE Symp. Secur. Privacy*, 2017, pp. 393–409.
[46] Technical Specification Group Services and System Aspects; Service requirements for the Evolved Packet System (EPS) (Release 15), 3GPP TS 22.278 V15.1.0, 2017, 3rd Generation Partnership Project Std.
[47] Technical Specification Group Services and System Aspects; 3GPP System Architecture Evolution (SAE); Security architecture (Release 15), 3GPP TS 33.401 V15.2.0, 2018, 3rd Generation Partnership Project Std.
[48] Technical Specification Group Services and System Aspects; 3GPP System Architecture Evolution (SAE); Security aspects of non-3GPP accesses (Release 14), 3GPP TS 33.402 V14.3.0, 2017, 3rd Generation Partnership Project Std.
[49] Technical specification group services and system aspects; security architecture and procedures for 5g system (release 15), 3gpp ts 33.501 v15.2.0, 2018.
[50] H. Krawczyk and T. Rabin, "Chameleon signatures," in *Proc. 7th Netw. Distrib. Syst. Secur.*, 2000, pp. 1–12.
[51] M. A. Abdrabou, A. D. E. Elbayoumy, and E. A. El-Wanis, "Lte authentication protocol (EPS-AKA) weaknesses solution," in *Proc. 7th Int. Conf. Intell. Comput. Inf. Syst.*, 2015, pp. 434–441.
[52] R. Arul, G. Raja, A. K. Bashir, J. Chaudry, and A. Ali, "A console grid leveraged authentication and key agreement mechanism for lte/sae," *IEEE Trans. Ind. Informat.*, vol. 14, no. 6, pp. 2677–2689, Jun. 2018.
[53] M. Dehnel-Wild and C. Cremers, "Security vulnerability in 5g-aka draft," Dept. Comput. Sci., Univ. Oxford, Tech. Rep, 2018. [Online]. Available: https://www.cs.ox.ac.uk/5G-analysis/5G-AKA-draft-vulnerability.pdf
[54] A. Koutsos, "The 5g-aka authentication protocol privacy," *arXiv: 1811.06922*, pp. 1–16, 2018. [Online]. Available: https://arxiv.org/pdf/1811.06922.pdf
[55] G. I. Tsiropoulos, A. Yadav, M. Zeng, and O. A. Dobre, "Cooperation in 5g hetnets: Advanced spectrum access and d2d assisted communications," *IEEE Wireless Commun.*, vol. 24, no. 5, pp. 110–117, Oct. 2017.
[56] Technical Specification Group Services and System Aspects; General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access (Release 15), 3GPP TS 23.401 V15.2.0, 2017, 3rd Generation Partnership Project Std.
[57] M. Burrows, M. Abadi, and R. M. Needham, "A logic of authentication," *Proc. Roy. Soc. London A: Math. Phys. Eng. Sci.*, vol. 426, no. 1871, pp. 233–271, 1989.

[58] R. Amin, N. Kumar, G. Biswas, R. Iqbal, and V. Chang, "A light weight authentication protocol for iot-enabled devices in distributed cloud computing environment," *Future Generation Comput. Syst.*, vol. 78, pp. 1005–1019, 2018.
[59] C. Miranda, G. Kaddoum, and E. Bou-Harb, "Cross-layer authentication protocol design for ultra-dense 5g hetnets," in *Proc. IEEE Int. Conf. Commun.*, 2018, pp. 1–7.
[60] NIST, "Digital signature standard (dss)," *Federal Inf. Process. Standards Publication 186-4*, 2013. [Online]. Available: http://www.gocs.de/pages/chiffrierverfahren/archiv/5-fips_186-3.pdf
[61] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*. Boca Raton, FL, USA: CRC Press, 1996.
[62] D. Johnson, A. Menezes, and S. Vanstone, "The elliptic curve digital signature algorithm (ecdsa)," *Int. J. Inf. Secur.*, vol. 1, no. 1, pp. 36–63, 2001.
[63] K. Okeya and K. Sakurai, "Fast multi-scalar multiplication methods on elliptic curves with precomputation strategy using montgomery trick," in *Proc. Int. Workshop Cryptographic Hardware Embedded Syst.*, 2002, pp. 564–578.
[64] J. Waring, "How many global base stations are there anyway?" Mobile World Live, 2017. [Online]. Available: https://www.mobileworldlive.com/blog/blog-global-base-station-count-7m-or-4-times-higher/
[65] J. Kooistra, "Newzoo global mobile market report 2018," Newzoo, 2018. [Online]. Available: https://newzoo.com/insights/trend-reports/newzoo-global-mobile-market-report-2018-light-version/

Yinghui Zhang is a professor of National Engineering Laboratory for Wireless Security (NELWS), Xi'an University of Posts & Telecommunications since 2018. He is also a research fellow with Singapore Management University. He has published more than 80 research articles in ACM ASIACCS, the *IEEE Transactions on Services Computing*, the *Computer Networks*, the *IEEE Internet of Things Journal*, the *Computers & Security*, the *IEEE Transactions on Industrial Informatics*, the *IEEE Transactions on Dependable and Secure Computing*, etc. His research interests include public key cryptography, cloud security and wireless network security. He is a member of the IEEE.

Robert H. Deng is an AXA chair professor of cybersecurity and a professor of information systems with the School of Information Systems, Singapore Management University since 2004. His research interests include data security and privacy, multimedia security, network and system security. He served/is serving on the editorial boards of many international journals in security, including the *IEEE Transactions on Information Forensics and Security*, the *IEEE Transactions on Dependable and Secure Computing*, and the *IEEE Security and Privacy Magazine*. He has been the chair of the Steering Committee of the ACM Asia Conference on Computer and Communications Security (ASIACCS) since 2012. He is a fellow of the IEEE.

Elisa Bertino is a professor of computer science with Purdue University, and serves as director of the Cyber Space Security Lab (Cyber2SLab). She is also an adjunct professor of computer science & info tech, RMIT, in Melbourne. Prior to joining Purdue in 2004, she was a professor and department head with the Department of Computer Science and Communication, University of Milan. She has been a visiting researcher with the IBM Research Laboratory (now Almaden) in San Jose, California, with the Microelectronics and Computer Technology Corporation, at Rutgers University, and with Telcordia Technologies. Her recent research focuses on database security, digital identity management, policy systems, and security for web services. She received the IEEE Computer Society 2002 Technical Achievement Award, the IEEE Computer Society 2005 Kanai Award, and the ACM SIGSAC Outstanding Contributions Award. She served as EiC of the the *IEEE Transactions on Dependable and Secure Computing*. She is a fellow of the ACM, IEEE, and AAAS.

Dong Zheng received the PhD degree in communication engineering from Xidian University, China, in 1999. He was a professor with the School of Information Security Engineering, Shanghai Jiao Tong University. He is currently a professor with National Engineering Laboratory for Wireless Security (NELWS), Xi'an University of Posts & Telecommunications. His research interests include cloud computing security, public key cryptography, and wireless network security. He has published more than 100 research articles including CT-RSA, the *IEEE Transactions on Industrial Electronics*, the *Information Sciences*.