This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSAC.2018.2815418, IEEE Journal on Selected Areas in Communications

Efficient and Secure Service-oriented Authentication Supporting Network Slicing for 5G-enabled IoT

Jianbing Ni, *Student Member, IEEE,* Xiaodong Lin, *Fellow, IEEE,* Xuemin (Sherman) Shen, *Fellow, IEEE*

***Abstract*—5G network is considered as a key enabler in meeting continuously increasing demands for future Internet of Things (IoT) services, including high data rate, numerous devices connection and low service latency. To satisfy these demands, network slicing and fog computing have been envisioned as promising solutions in service-oriented 5G architecture. However, security paradigms enabling authentication and confidentiality of 5G communications for IoT services remain elusive, but indispensable. In this paper, we propose an efficient and secure service-oriented authentication framework supporting network slicing and fog computing for 5G-enabled IoT services. Specifically, users can efficiently establish connections with 5G core network and anonymously access IoT services under their delegation through proper network slices of 5G infrastructure selected by fog nodes based on the slice/service types of accessing services. The privacy-preserving slice selection mechanism is introduced to preserve both configured slice types and accessing service types of users. In addition, session keys are negotiated among users, local fogs and IoT servers to guarantee secure access of service data in fog cache and remote servers with low latency. We evaluate the performance of the proposed framework through simulations to demonstrate its efficiency and feasibility under 5G infrastructure.**

Keywords: 5G network, Internet of Things (IoT), anonymous authentication, fog computing, network slicing.

I. Introduction

The rapid development of communication and sensing technologies is paving the way to the realization of Internet of Things (IoT) [1], which is the internetworking of numerous physical objects, such as sensors, mobile phones, vehicles, RFID tags and other electronic embedded devices, and network connectivity that allows these objects to communicate and exchange data. IoT enables a large number of applications in different domains, including smart city, smart healthcare, intelligent transportation, industrial automation and disaster response [2]. This trend leads to the production of large volumes of data. The amount of IP data handled by wireless networks increases by over a factor of 100: from under 3 exabytes in 2010 to over 190 exabytes by 2018, on pace to exceed 500 exabytes by 2020 [3]. In addition to the sheer volume of data, the number of connected devices will reach 50 billion by 2020 [4]. Due to the deluge of data brought by massive devices, even cutting-edge cellular networks cannot offer the bandwidth IoT services require [5].

This work was supported by the National Natural Science Foundation of China under Grant 61728102. Corresponding Author: Xiaodong Lin

Jianbing Ni and Xuemin (Sherman) Shen are with Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada N2L 3G1. Email: j25ni@uwaterloo.ca, xshen@bbcr.uwaterloo.ca.

Xiaodong Lin is with Department of Physics and Computer Science, Wilfrid Laurier University, Waterloo, Ontario, Canada N2L 3C5, E-mail: xlin@wlu.ca.

To satisfy the requirements of new IoT services, the fifth generation (5G) of mobile communication era is coming [3], [6], which aims to offer a 1000 times higher mobile data volume per unit area, 10–100 times more connecting devices and data rate, and 5 times reduced latency [7]. 5G will become the backbone of IoT, offering full connections to all "things", breaking through time-space constraints, to create all-dimensional, service-tailored and user-centric interconnections [8]. To support these desirable features, network slicing [9] has been introduced to provide customized reliable services based on limited network resources of 5G network with low capital expenditure and operating expense. By slicing a physical network into several logical networks, it enables on-demand customized services for different IoT applications sharing the same physical network at the same time [10]. The network elements and resources in each network slice can be configured and reused in parallel with isolation to reach the features of dedicated services, for example, to prevent data transmission in one slice from being negatively impacted by the services in other slices. In addition, to reduce response latency of IoT services, nodes at the edge of access network have been virtually upgraded with computing and storage resources, such as base stations, routers and switches. By integrating fog computing [11], service data can be cached on fog nodes proximate to user devices for low-latency and location-aware IoT applications. Supported by network slicing and fog computing, network resources are assigned to the dedicated services in a cost-efficient manner under the specific demands of IoT services on data volume, data rate and response latency [12].

While these innovations bring a variety of distinctive advantages to customized services in IoT, they trigger quite a few issues on security and privacy preservation. The standardized 5G security architecture in 3GPP is borrowed from 4G standards [13], such that data confidentiality, identity authenticity and user privacy are being put at risks due to the following reasons. First of all, both IoT service and 5G infrastructure, specifically fog nodes, are confronted with serious threats on privacy, integrity, availability and authentication, e.g., eavesdropping attack, impersonation attack, tracing attack and tampering attack [14]. Even the well-protected 4G LTE network has proved to be vulnerable to denial-of-service attacks [15]. Secondly, it is difficult to guarantee users accessing 5G network and IoT services honestly, rather than selfishly acquiring more benefits as they wish. An example is that a user prefers to pretend a legitimate one to access core network and enjoy IoT services without being charged. Thirdly, IoT service providers or fog nodes have their incentives to behave unfaithfully towards user

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSAC.2018.2815418, IEEE Journal on Selected Areas in Communications

privacy. For example, fog nodes might learn the interested services of users through the accessing network slices, or predict their shopping habits, daily activities, and other privacy witnessable references based on their subscribed IoT services [16]. In short, 5G-enabled IoT lacks of sufficient security and privacy guarantees after logical network slices and fog computing are built for individual services. These problems, if not property addressed, may impede the success of both 5G network and IoT applications.

Therefore, how to enhance security and privacy protection for IoT services powered by 5G is the focus of this paper. To secure 5G-enabled IoT services, a national demand is to design efficient service-oriented authentication protocols for numerous users with the severe demands of different IoT services. To preserve user privacy, it is critical to hide users' identities during service authentication. Thus, the challenge is to support anonymous service-oriented authentication with scalability of handing a large number of authentication requests. Furthermore, after users' identities are well protected, it is still possible for local fog nodes to identify users through some side-channel information, such as users' accessing services, which results in unwelcome advertisements for users [17]. However, it is of difficulty to protect service types against fog nodes since they are required to select proper network slices for service package forwarding. The types of network slices can expose the service types of the packages transmitting on these network slices. Therefore, protecting the accessing service types and the configured slice types during slice selection is another challenge we aim to overcome.

To address these issues, we propose an Efficient, Secure network-Sliced and Service-oriented Authentication framework (ES^3A) supporting privacy-preserving slice selection and service-oriented anonymous authenticated key agreement for 5G-enabled IoT. It enables both 5G operator and IoT service provider to generate anonymous delegation for subscribed users to access IoT services, and support slice selection for fog nodes without exposing slice/service types. Specifically, the contributions of this paper are four-fold:

- We propose an ES^3A to guarantee secure access of IoT services without exposing users' privacy in 5G-enabled IoT. With the integration of network slicing and fog computing, users are allowed to transmit their service data on proper slices with data isolation and customized service features, efficiently authenticate themselves under the delegation of 5G operator and IoT service provider, and secure access of service data maintained on remote servers and local fog nodes with low latency.
- A privacy-preserving slice selection mechanism is proposed to enable fog nodes choose proper network slices for package forwarding based on the matching of allowed service types and configured slice types. Both slice/service types and the features differentiating network slices are protected against fog nodes during slice selection to break the links between users and their accessing services.
- To support anonymous service-oriented authentication, we enhance the group signature [18] to realize service delegation of both 5G operator and IoT service provider, and anonymous batch verification for users' authentication messages. Neither service providers nor local fogs can learn any information about the subscribers of IoT services, but both are aware of whether users are legitimate to access IoT services.
- We design a service-oriented three-party key agreement mechanism to negotiate session keys among IoT servers, local fogs and users, based on Diffie-Hellman key agreement [19]. With the negotiated session keys, users are capable of securely accessing IoT service data maintained on both remote IoT servers and local fog nodes, which cache service contents to offer IoT services with low latency.

The remainder of this paper is organized as follows. We review the related work in section II and formalize the architecture of 5G-enabled IoT services, security threats and design goals in section III. In section IV, we propose our ES^3A and analyze its security, followed by the justification of performance in section V. Finally, we conclude our paper in section VI.

II. Related Work

5G network is emerging to link up fixed and mobile devices for a variety of applications with widely varied requirements in terms of bandwidth, latency, security and continuous network availability. To do so, Li et al. [20] proposed an end-to-end network slicing framework to horizontally slice computing and communication resources to form virtual computation platforms for supporting vertical industry applications. The concept of 5G network slice broker was introduced by Samdanis et al. [21] to enable virtual network operators to dramatically request and lease resources from infrastructure, and thereby reduce capital expenditure and operational expenditure costs. Zhou et al. [22] introduced hierarchical network slicing as a service, which helps 5G network operator to provide customized end-to-end cellular networks, and presented the architecture of service management across different levels of service models. Liang et al. [23] proposed an information-centric wireless network virtualization framework to balance the revenue obtained from serving users and the cost on leasing infrastructure. The virtual resource allocation and caching strategy is formulated based on the requests of IoT service providers. The above schemes present several network slicing strategies for mobile network, the protection of slice selection is ignored unfortunately, which leads to user privacy leakage.

To secure mobile network, the enhanced authentication profile (EAP) framework, specified in RFC 3748, has been deployed to achieve the authentication between users and core network in 4G/LTE network. In 5G security architecture, EAP framework is also leveraged to support the primary authentication between users and AUSF [13]. Subscriber privacy is a critical demand specified in standards of 5G security, but 5G security architecture does not protect user privacy [24]. A straightforward solution is to use a mark "anonymous" to replace a user's real identity, but leading to the infeasibility of identity authentication. Currently, several anonymous authentication protocols [25], [26], [27] have been designed to enable

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSAC.2018.2815418, IEEE Journal on Selected Areas in Communications

authentication and key agreement between roaming users and foreign network operators in traditional mobile networks. For 4G/LTE mobile network, Choi et al. [28] proposed a mutual authentication and secure key distribution protocol based on symmetric cryptography for machine-type communications (MTC) in long-term evaluation-advanced (LTE-A) network. The user devices in the same group can share a symmetric key to achieve secure communications with high efficiency and low cost. Li et al. [29] illustrated that separate authentication for each MTC device may result in signal congestion and difficulty on access-policy updating, and presented a group-based authentication and key agreement protocol with dynamic MTC-device access authority updating based on asynchronous secret sharing in LTE-A network. Lai et al. [30] proposed a lightweight group authentication protocol for resource-constrained MTC under both 3GPP and non-3GPP access networks, and an end-to-end secure authentication scheme to realize mutual authentication between MTC servers and user devices in 3GPP core network based on proxy signatures. The aforementioned protocols are designed to support secure network authentication for MTC in 4G/LTE architecture.

In 5G network, physical network resources are sliced to support different services. Rost et al. [31] demonstrated the scalability and flexibility brought by network slicing in 5G mobile network, and pointed out potential security vulnerabilities on the communication between 5G network operators and their users with the partition of network slices. Yang et al. [32] examined potential security issues in 5G network and designed a physical layer security mechanism to safeguard data confidentiality by leveraging intrinsic randomness of communication medium. A security and trust framework was proposed by Yan et al. [33] based on adaptive trust evaluation and sustainable trusted computing technologies to solve 5G network security issues. To protect real-time video reporting services in 5G-enabled vehicular network, Eiza et al. [34] studied the security and privacy issues in this service and utilized the anonymity technique to protect vehicles' identities, and ensure the traceability of misbehaving participating vehicles. Duan and Wang [35] illustrated security challenges on frequent handovers and authentication caused by stringent latency requirements in 5G hetnets, and designed an efficient authentication scheme by simplifying authentication handover through the sharing of security parameters among access points. However, service-oriented authentication with network slicing has not mentioned to build secure communications between user devices and IoT servers in 5G network.

Therefore, we propose an efficient and secure service-oriented authentication framework enabling network slicing and fog computing for 5G-enabled IoT services. To fit the scenario of 5G network, ES^{3}A achieves the properties that (1) privacy-preserving slice selection to allow the controller to select suitable network slices for package forwarding based on the service types without exposing the slice/service types; (2) anonymous service authentication to achieve the delegation from both 5G operator and IoT server, enable users to securely access the delegated IoT service without privacy leakage, and support batch verification to improve the computational efficiency for the IoT server on identity verification; and (3) service-oriented key agreement to build secure channels for information exchange between users, local fogs and remote IoT servers.

III. Problem Statement

In this section, we present the architecture of 5G-enabled IoT, discuss security threats, and identify our design goals.

A. Architecture of 5G-enabled IoT

5G aims to handle large volumes of data, connect numerous devices, reduce service latency and bring new levels of reliability for offering customized services based on specific quality-of-service demands [36]. By integrating network softwarization technologies [37], including network slicing, software-defined networking and fog computing, 5G is predicted to provide three categories of services, i.e., massive machine-type communication (mMTC, a.k.a., massive-scale IoT), enhanced mobile broadband (eMBB) and ultra-reliable low latency communication (UR-LLC) [38].

Following the 3GPP TS 23.501 [39], 5G architecture consists of four layers: user devices, access network, core network and IoT services, as shown in Fig. 1. Massive user devices are connected with data network through core network and access network (e.g., E-UTRAN, WLAN, WiMAX or other non-3GPP access networks), and communicate with IoT servers. The core network is separated user plane function (UPF) from control plane function (CPF) to allow independent scalability and flexible deployment. UPF includes data forwarding, traffic usage reporting, transport level packet marking in the uplink and downlink, etc. CPF controls packet processing in UPF by provisioning a set of rules in sessions, i.e., forwarding action rules for packets handling, packet detection rules for packets inspection, QoS enforcement rules to enforce QoS policies on the packets. As shown in Fig. 1, CPF includes several functions, including Access and Mobility Management Function (AMF), Session Management Function (SMF), Policy Control Function (PCF), Network Slice Selection Function (NSSF), Unified Data Management (UDM), Authentication Server Function (AUSF), etc., which have their individual functionalities. Specifically, AMF manages user registration, connection, reachability and mobility, access authentication and authorization; SMF includes the functionality of session management and roaming; PCF supports unified policy framework to govern network behaviour; UMD is responsible for authentication credential generation and subscription management; AUSF supports authentication server function and NSSF selects the set of network slice instances serving users and determines the Network Slice Selection Association Information (NSSAI) corresponding to applicable network slice instances. The core network is connected to the external data networks, such as the Internet, where IoT service providers offer a variety of appealing IoT services to users with fast, ubiquitous, and low-power connectivity.

To support different IoT services in 5G network, network slicing achieves the separation and prioritization of resources on a common infrastructure, including network capability, computing resources, virtual network functions and radio

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSAC.2018.2815418, IEEE Journal on Selected Areas in Communications

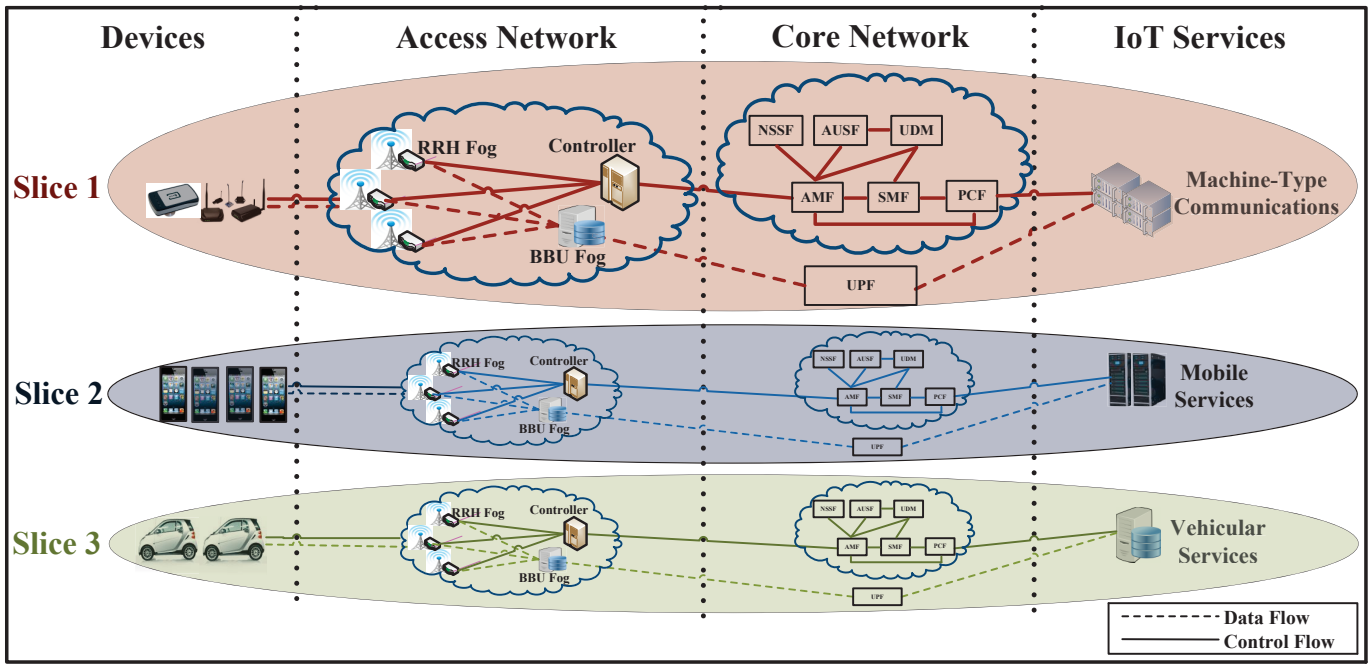


Fig. 1. Service-oriented Architecture of Fog-assisted 5G Network with Network Slicing

access technology settings [21]. 5G infrastructure in Fig. 1 is virtually sliced to support various types of IoT services. For example, slice 1 is established for machine-type communications by setting fully-fledged functions distributed across the network, which is isolated with the slices for mobile services and vehicular communications. The mobile phones need an end-to-end slice with a large bandwidth to enable high-data-rate and low-latency services, such as video streaming and augmented reality [10]. These network slices are managed by local controllers at the edge of access networks and control plane in core network, which select network slices for the coming packages based on slice/service types and other auxiliary information, and forward the packages to the IoT servers.

Access networks, either 3GPP or non-3GPP, are upgraded for delay-sensitive or location-aware IoT services by pushing computing and storage resources to the edge of access networks. Fog nodes are deployed on the top of cloud-radio access network (C-RAN) to offer high-speed baseband communications. In Fig. 1, the conventional base stations are divided into two components, baseband unit (BBU) and remote radio head (RRH) [40]. BBU fogs mainly provide centralized storage and communications for local services in access networks. RRH fogs are equipped with processing units, storage resources and computing capability for location-aware mobile applications and local data caching. These resources are virtualized to be isolated virtual machines (VMs) controlled by a local controller, which offers authentication, authorization, accounting, resource allocation and mobility management [41]. In terms of non-3GPP network, routers, switches and networking modules can be extended to fog nodes with consistent data management and local service hosting, under the management of local controllers.

B. Security Threats

5G enhances the diversity and scalability of IoT services, meanwhile faces with a variety of security and privacy threats from both internal and external attackers. Thirty-five types of cyber attacks have been identified as major threats on privacy, authentication, integrity and availability in 5G network [14], which bring serious security and privacy risks to its powered services. For example, an adversary may launch eavesdropping attacks to capture forwarding packages, man-in-the-middle attacks to acquire the session keys or stalking attacks to trace their locations. These outsider attacks, invading IoT service security and user privacy, are the major security threats for each entity in service-oriented IoT architecture.

IoT service providers, offering various services to users, have strong incentives to protect service data and facilities for monetary reasons. On the other hand, IoT service providers are greedy on their benefits by attracting more users accessing their services. Further, they are curious about the personal information about users for advertisement recommendation based on their accessed services and users' sensitive information sharing with their cooperators.

As intermediate nodes, RRH fogs, BBU fogs and controllers are able to capture the authentication messages and service data about local users, which can be used to predict mobility patterns and extract sensitive information about users. In addition, the controllers can learn the types of accessing services, so as to find the user's preferences and conduct statistic analysis on the services.

As the beneficiaries of 5G network and IoT services, users would neither actively break the network infrastructure or IoT services by cheating or via cyber attacks, nor share their secret credentials or keys to others. They have sufficient incentives to protect access credentials and secret keys securely.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSAC.2018.2815418, IEEE Journal on Selected Areas in Communications

Nonetheless, some users may pretend legitimate subscribers to access IoT services incapable to be acquired. They may also be curious about the services that other users subscribe and prefer to have free IoT services. Thereby, the users are honest-but-curious.

5G network operators honestly provide network connections to users with the responsibility of improving communication quality for their own income. Powerful intrusion detection mechanisms and great firewalls are deployed for the protection of core network. They are honest-but-curious as well, indicating that they are honest to build network and service connections for users, but curious on the exchanging data between users and IoT servers.

C. Design Goals

We aim to offer strong security guarantees for 5G-enabled IoT and privacy protection for users against the aforementioned security threats. The design goals of our ES^{3}A are as follows:

Privacy-preserving Slice Selection: To deter user privacy leakage, the accessing service types are protected against curious fog nodes or other outsider attackers. The mapping of allowed NSSAIs and configured slice types, i.e., part of the configured/allowed NSSAIs are applicable for the subscribed services, should be learnt by fog nodes to select proper slices for package forwarding.

Service-oriented Anonymous Authentication: Users utilize access credentials, generated by the AUSF and an IoT server, to authenticate themselves for the IoT service access. Without a valid access credential, an attacker is not able to succeed the verification of the controller or the IoT server. During service authentication, neither the local controller nor the IoT server can learn users' identities, which means that they cannot distinguish the source of an authentication message from two possible users.

Service-oriented Key Agreement: To protect service data, a unique session key is negotiated among the IoT server, the controller and the user. No attacker is able to learn the negotiated session key or corrupt the negotiation process. The session key is adopted to encrypt service data for the user to access the specified service offered by the IoT server.

IV. The Proposed ES^{3}A

We review the preliminaries and propose our efficient and secure service-oriented authentication framework for 5G-enabled IoT services.

A. Preliminaries

Notions. $s \in_R S$ denotes s is randomly chosen from a non-empty set S. Let p be a large prime and $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$ be three cyclic groups of the order p, satisfying the type 3 bilinear pairing $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ [18]. Here $\mathbb{G}_1 \neq \mathbb{G}_2$ and no efficiently computable homomorphism exists between $\mathbb{G}_1$ and $\mathbb{G}_2$ in either direction. The type 3 bilinear pairing has critical differences from the Type 1 and Type 2 bilinear parings, in which the Type 1 is that $\mathbb{G}_1 = \mathbb{G}_2$, and the type 2 pairing has the properties that $\mathbb{G}_1 \neq \mathbb{G}_2$ and there exists an efficiently computable homomorphism $\pi : \mathbb{G}_2 \rightarrow \mathbb{G}_1$, but no homomorphism in the other direction. g is a generator of $\mathbb{G}_1$ and h is a generator of $\mathbb{G}_2$.

Mathematical Assumptions. We revisit several mathematical assumptions on which the security of ES^{3}A relies.

Computational Diffie-Hellman (CDH) assumption for Product Groups [42]. The CDH problem for product groups is that given $g^\alpha, g^\beta \in \mathbb{G}_1$, $h^\alpha, h^\beta \in \mathbb{G}_2$, to compute $g^{\alpha\beta}$. If there is no algorithm to address the CDH problem for product groups with non-negligible probability in probabilistic polynomial time, we say that the CDH assumption for product groups holds.

Modified LRSW assumption 1 [18]. The modified LRSW 1 problem is that given $g^b \in \mathbb{G}_1, h^a, h^b \in \mathbb{G}_2$, where $a, b \in_R \mathbb{Z}_p$, and an oracle $\mathcal{O}$, on which input $m \in_R \mathbb{Z}_p$ and return the pair $P = (g_1, {g_1}^{a+bm})$, where $g_1 \in_R \mathbb{G}_1 \setminus 1_{\mathbb{G}_1}$, to generate such a new pair for a new m' that was not queried in $\mathcal{O}$. If there is no algorithm to solve the modified LRSW problem 1 with non-negligible probability in probabilistic polynomial time, we say the modified LRSW assumption 1 holds.

Modified LRSW assumption 2 [18]. The modified LRSW 2 problem is that given $h^a, h^b \in \mathbb{G}_2$, where $a, b \in_R \mathbb{Z}_p$, and an oracle $\mathcal{O}$, on which input $m \in_R \mathbb{Z}_p$ and return the pair $P = (g_1, {g_1}^{a+bm})$, where $g_1 \in_R \mathbb{G}_1 \setminus 1_{\mathbb{G}_1}$, to generate such a new pair for a new m' that was not queried in $\mathcal{O}$. If there is no algorithm to solve the modified LRSW problem 2 with non-negligible probability in probabilistic polynomial time, we say the modified LRSW assumption 2 holds.

Both modified LRSW assumption 1 and assumption 2 are proved to hold in the generic group model in [18].

PS Signature [18]. The PS signature proposed by Pointcheval and Sanders [18] is a public-key signature scheme based on the type 3 bilinear pairing. The existential unforgeability against chosen message attacks is proven under the modified LRSW assumption 2.

The secret key of the signer is $x, y_1, \cdots, y_r \in_R \mathbb{Z}_p$ and the public key is $(\widehat{X}, \widehat{Y}_1, \cdots, \widehat{Y}_r) \leftarrow (h^x, h^{y_1}, \cdots, h^{y_r})$. The signer can generate a signature σ on multi-block messages $(m_1, \cdots, m_r) \in \mathbb{Z}_p^r$ as $\sigma = (g_1, g_1^{x+\sum_{j=1}^r y_j m_j})$, where $g_1 \in_R \mathbb{G}_1 \setminus 1_{\mathbb{G}_1}$. Any verifier can verify the signature $\sigma = (\sigma_1, \sigma_2)$ by checking whether $\sigma_1 \neq \mathbb{G}_1 \setminus 1_{\mathbb{G}_1}$ and $\hat{e}(\sigma_1, \widehat{X} \prod_{j=1}^r \widehat{Y}_j^{m_j}) = \hat{e}(\sigma_2, h)$ hold.

The PS signature can be used to construct a sequence signature based on the public key sharing technique and a group signature by signing the committed messages and randomizing the signatures based on the Schnorr signature [18].

Network Slicing [39] Network slicing logically divides whole network resources into several slices that differ for supported features and network function optimisations. A network slice is identified by a Single-NSSAI (S-NSSAI), consisting of a slice/service type (SST) and some optimal information called a slice differentiator (SD) that complements the slice/service type(s) to differentiate amongst multiple network slices of the same slice/service type. The network slices have different S-NSSAIs with distinctive slice/service types deployed by 5G operator. The slice/service types of eMBB, URLLC and MIoT

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSAC.2018.2815418, IEEE Journal on Selected Areas in Communications

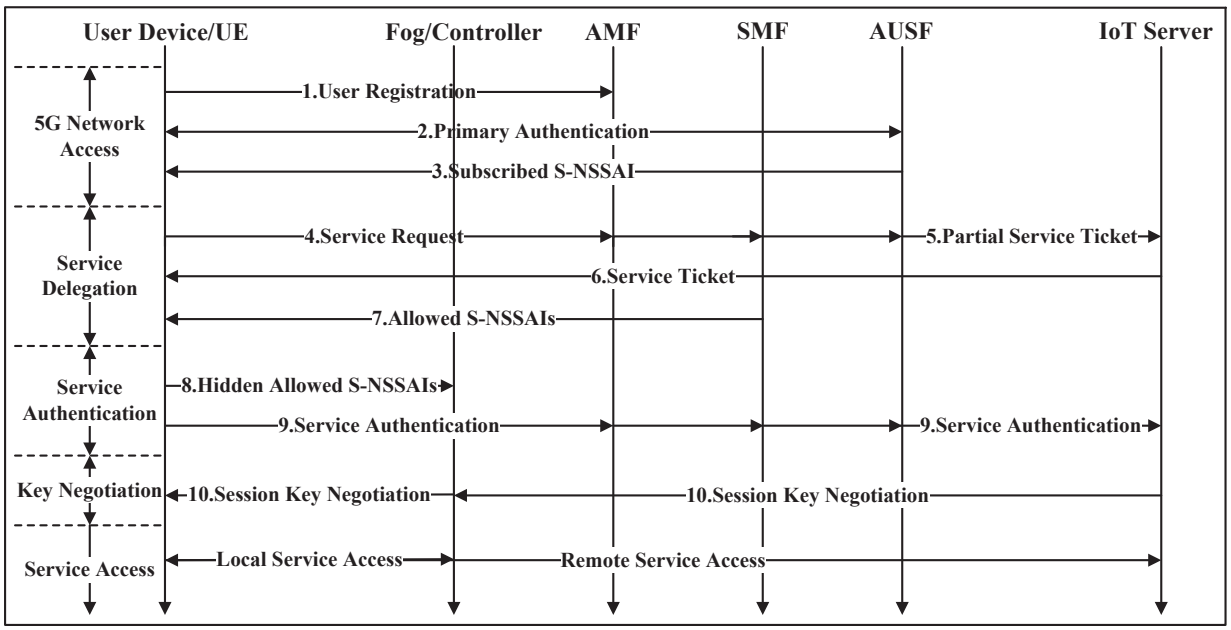


Fig. 2. Information Flow of ES^{3}A

have been specified in 3GPP TS 23.501 [39]. Other types can be defined by the 5G operator for different services. Each user is required to subscribe the desirable services with the generation of subscription information, which contains one or more S-NSSAIs, i.e., the subscribed NSSAIs. To access the subscribed services, the user provides a requested NSSAI to AMF in the user's registration procedure. AMF needs to query NSSF with the requested NSSAI and query UDM to retrieve user's subscription information including the subscribed S-NSSAIs, and the user obtains an allowed NSSAI, which includes one or more S-NSSAIs, from AMF, associated with the mapping of requested NSSAI to configured NSSAI for that service. The user can leverage the allowed NSSAI corresponding to network slices for service access. Under the delegation of NSSF, the controllers are able to check the allowed NSSAI for selecting a proper AMF that supports the required network slices. If the controller is unable to select an AMF based on the NSSAI, it routes the packages to a default AMF, which is set in the subscription information.

B. High-level Description of ES^{3}A

We show a high-level description of ES^{3}A, consisting of six phases: System Initialization, Network Slicing, 5G Network Access, Service Delegation, Service Authentication and Key Negotiation, and the information flow of ES^{3}A in Fig. 2.

System Initialization. A 5G operator bootstraps the whole network and builds 5G network for users. AUSF setups the system parameters $Params$, and generates the secret key (a_0, a_1) and the public key $(A_0, \widehat{A}_0, \widehat{A}_1)$ of 5G operator. The IoT server (ISV) and the local controller also generates their secret-public key pairs $(b, \widehat{B})$ and $(c, \widehat{C})$, respectively.

Network Slicing. Physical network resources are separated into several network slices by core network. The slice/service types SSTs and the features SAIs are specified for configured NSSAIs deployed on network slices. AMF computes PS_i with the secret key to hide each slice/service type SST_i, and generates ACF_i to protect the feature values A_iAI of SST_i. The controller maintains the protected configured NSSAI (PS_i, ACF_i) to select proper network slices for service packages.

5G Network Access. A user U_i generates the secret-public key pair (usk_i, upk_i) and performs the registration and authentication specified in 3GPP to access 5G network. U_i obtains network access credentials and subscription information, including subscribed S-NSSAIs, and establishes a Non Access Stratum (NAS) security context.

Service Delegation. To access the IoT service, U_i submits a NAS message and a request to AMF. SMF generates allowed S-NSSAIs with NSSF, and AUSF delegates access capability by generating a partial service ticket PST_i and forwards them to ISV. ISV generates a service ticket ST_i for U_i, including a session identifier $\mathcal{N}_i$, a service credential ϕ_i, a key negotiation tag (X_{i1}, X_{i2}). U_i verifies ϕ_i and obtains the allowed S-NSSAIs and the service credentials for anonymous service authentication.

Service Authentication. To authenticate to the service, U_i generates an authenticated key agreement message AKA_i and the hidden allowed S-NSSAIs PAS. According to the hidden allowed S-NSSAIs, the controller is able to find the proper network slice for message transmission without the knowledge of the detailed service types. ISV verifies AKA_i to allow U_i to access the service without learning the real identity of U_i, if the service credential in AKA_i is valid. In addition, a key negotiation tag (Y_{i1}, Y_{i2}) is generated for session key agreement.

Key Negotiation. The controller generates its key negotiation tag (Z_{i1}, Z_{i2}) and interacts with ISV and U_i to negotiate a session key sk_i for secure communications and service access.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSAC.2018.2815418, IEEE Journal on Selected Areas in Communications

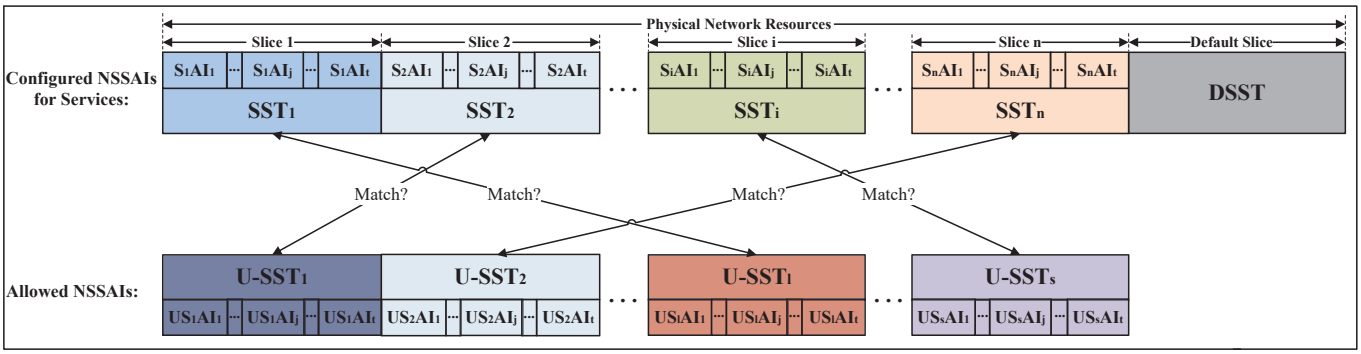


Fig. 3. Mapping of Allowed NSSAIs to Configured NSSAIs for Service Data Forwarding

C. The Detailed ES^{3}A

Now we describe our proposed ES^{3}A in detail.

System Initialization. A 5G network operator establishes the whole 5G mobile network, including radio access networks and 5G core network, to connect user devices and data network for supporting IoT services. Various network functionalities, i.e., AMF, SMF, AUSF, UDM, NSSF and PCF, are initialized to provide network connection and data network access. AUSF sets the security parameter κ, which denotes the security level. In general, $\kappa = 160$ or $\kappa = 256$. Let p be a large prime with κ bits and $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$ be three cyclic groups of the order p. $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ is the type 3 bilinear pairing. g is a generator of $\mathbb{G}_1$ with $g \neq 1_{\mathbb{G}_1}$, and h is a generator of $\mathbb{G}_2$ with $h \neq 1_{\mathbb{G}_2}$. AUSF also defines four collision-resistant hash functions: $H : \{0,1\}^* \rightarrow \mathcal{K}$, $H_0 : \{0,1\}^* \rightarrow \mathbb{Z}_p$, $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1$ and $H_2 : \{0,1\}^* \rightarrow \mathbb{G}_2$. $\mathcal{C} = AES_{ENC}(\mathcal{K}, \mathcal{M})$ and $\mathcal{M} = AES_{DEC}(\mathcal{K}, \mathcal{C})$ are the encryption and decryption algorithms of advanced encryption standard (AES) [19], respectively. $(\mathcal{SE}, \mathcal{SD})$ are the encryption and decryption algorithms of a deterministic symmetric encryption scheme. Such a scheme can be constructed from AES scheme with a fixed IV in CTR mode [19]. The system parameters $Params = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g, h, \hat{e}, H, H_0, H_1, H_2)$ are maintained on AUSF. AUSF randomly selects the secret key of core network $a_0, a_1 \in_R \mathbb{Z}_p$ to compute the corresponding public key $(A_0, \widehat{A}_0, \widehat{A}_1) = (g^{a_0}, h^{a_0}, h^{a_1})$. AUSF applies for the public-key certificate $cert_C$ to Public Key Infrastructure (PKI), and secretly maintains its secret key (a_0, a_1).

Substantial ISVs in data network offer a variety of IoT services, i.e., online video, automatic meter reading or augmented reality, through 5G mobile network. Suppose an ISV provide IoT service ISN, generate its secret key by randomly picking $b \in_R \mathbb{Z}_p$ and compute the corresponding public key as $\widehat{B} = h^b$. ISV applies for the public-key certificate $cert_S$ to PKI and secretly maintains its secret key b.

The local controller manages data forwarding and caches service data for users with location-aware or low-latency service access. It randomly picks $c \in_R \mathbb{Z}_p$ as the secret key and computes the corresponding public key as $\widehat{C} = h^c$. It also obtains the public key certificate $cert_L$ from PKI and maintains its secret key c.

Network Slicing. As depicted in Fig. 3, physical network resources are separated into several logical network slices, i.e., PNR={Slice$_1$, Slice$_2$, $\cdots$, Slice$_n$, Default Slice}, in which each Slice$_i$ $\in$PNR is configured with a S-NSSAI$_i$, consisting of a slice/service type SST$_i$ and SD, which is the values of several features $S_iAI = \{S_iAI_1, S_iAI_2, \cdots, S_iAI_t\}$, where t is the number of features in SD. S_iAI represents the features based on which the network is sliced for the same service, such as latency, security, capacity, throughput and scalability. A service type SST$_i$ and its features S_iAI are defined by the core network with random and unpredictable identifiers chosen from $\mathbb{Z}_p$. For example, the latency levels are denoted as (L_0, L_1, L_2, L_3) illustrating (low, medium, high, very high), respectively, and the values of all features are nonidentical.

To deter user privacy disclosure, the accessing service types should be protected against fog nodes. In doing so, 5G core network is required to encrypt both the slice/service types and their feature values of configured network slices. Specifically, to preserve the configured S-NSSAIs, for each SST$_i$, $i \in \{1, 2, \cdots, n\}$, AMF randomly chooses $\epsilon_i \in \mathbb{Z}_p$ to calculate $PS_{i1} = g^{\frac{\epsilon_i}{a_1 + \mathrm{SST_i}}}$ and $PS_{i2} = h^{\epsilon_i}$, sets $PS_i = (PS_{i1}, PS_{i2})$. Further, to protect feature values S_iAI_j in the S-NSSAI of slice i, for $j \in \{1, 2, \cdots, t\}$, AMF calculates $ACF_{ij} = \mathcal{SE}(H(S_iAI_j, \widehat{C}), S_iAI_j)$ and sets $ACF_i = (ACF_{i1}, ACF_{i2}, \cdots, ACF_{it})$. AMF forwards $NL = (PS_1, PS_2, \cdots, PS_n, ACF_1, ACF_2, \cdots, ACF_n)$ to the controller. The controller stores NL and leverages them to select network slices for the coming packages. Note that, if the IoT service or its features are changed, AMF is required to update SST$_i$ or S_iAI_j of the corresponding network slice. It can use the new SST$'_i$ or feature values to generate a new pair (PS'_i, ACF'_i) for the controller.

5G Network Access. A user U_i, with its identity information ID_i, which can be the ID number, telephone number, email address or home address, is registered at AMF to be capable of accessing 5G network and its supported data network. U_i randomly chooses a random value $u_i \in_R \mathbb{Z}_p$ as the secret key usk_i, computes its public key as $upk_i = g^{u_i}$, acquires the public-key certificate $cert_i$ from PKI and secretly keeps usk_i. To access 5G network, U_i performs the registration and authentication operations defined in 3GPP TS 33.501 [39]. Specifically, U_i registers at 5G operator providing her/his identity information ID_i if the real identity is needed, and obtains network access credentials and subscription information. The subscription information may include one or more S-NSSAIs

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSAC.2018.2815418, IEEE Journal on Selected Areas in Communications

i.e., subscribed S-NSSAIs. Both credential and subscription information of U_i are maintained on UDM. U_i connects with 5G network by performing primary authentication with AUSF using its network access credentials and establishes a NAS security context with AMF, if the primary authentication is successful.

Service Delegation. To access the IoT service ISN offered by ISV, U_i sends a session management NAS message to AMF, including a requested NSSAI, along with the information $(\mathcal{N}_i, a_{i1}, a_{i3}, a_{ic}, b_{ic})$ generated as follows.

- U_i chooses a random value $\mathcal{N}_i \in_R \mathbb{Z}_p$ as the session number;
- U_i picks a random $t_i \in_R \mathbb{Z}_p$ to calculate $Q_i = g^{t_i}$ and $\widehat{Q}_i = \widehat{A}_0^{t_i}$;
- To prevent the disclosure of $(Q_i, \widehat{Q}_i)$, U_i picks a random l_i to calculate $a_{i1} = g^{l_i}$, $a_{i2} = H(a_{i1}, A_0^{l_i})$, and $a_{i3} = AES_{ENC}(a_{i2}, \mathcal{N}_i||Q_i||\widehat{Q}_i||ISN)$;
- To prevent the corruption, U_i randomly chooses $k_i \in_R \mathbb{Z}_p$ to compute $T_{ic} = g^{k_i}$, $a_{ic} = H_0(\mathcal{N}_i, a_{i1}, a_{i3}, T_{ic})$ and $b_{ic} = k_i + a_{ic}u_i$.

AMF computes $T_{ic} = g^{b_{ic}} upk_i^{-a_{ic}}$ and verifies $a_{ic} \stackrel{?}{=} H_0(\mathcal{N}_i, a_{i1}, a_{i3}, T_{ic})$. If yes, AMF forwards the received message to SMF, along with a generated session management request message. SMF decrypts (a_{i1}, a_{i3}) to obtain $(Q_i, \widehat{Q}_i, ISN)$ as $a_{i2} = H(a_{i1}, a_{i1}^{a_0})$ and $\mathcal{N}_i||Q_i||\widehat{Q}_i||ISN = AES_{DEC}(a_{i2}, a_{i3})$. SMF also queries the subscription information from UDM and checks whether U_i's request is compliant with the subscription and with local policies. If not, the request would be rejected; otherwise, SMF checks subscribed S-NSSAIs and the requested NSSAI to determine the allowed S-NSSAIs with the SSTs U-SSTs=$\{$U-SST$_1$, U-SST$_2$, $\cdots$, U-SST$_s\}$ with NSSF, and sends the request to AUSF. After receiving the request, AUSF delegates U_i the access capability of IoT service ISN by generating a partial service ticket PST_i in the following way:

- AUSF verifies whether $\hat{e}(Q_i, \widehat{A}_0) \stackrel{?}{=} \hat{e}(g, \widehat{Q}_i)$ holds. If not, it returns failure and aborts.
- AUSF randomly selects $r_i \in_R \mathbb{Z}_p$ to compute

$$\sigma_i = (\sigma_{i1}, \sigma_{i2}) = (g^{r_i}, (g^{a_1} Q_i^{a_0})^{r_i}), \tag{1}$$

  which is used to generate a service credential of U_i to access ISN. This is the delegation of service access from the 5G operator. U_i also needs to obtain the delegation from ISV.
- To prevent the disclosure of σ_i, AUSF randomly picks $v_i \in_R \mathbb{Z}_p$ to calculate $c_{i1} = h^{v_i}$, $c_{i2} = H(c_{i1}, \widehat{B}^{v_i})$, and $c_{i3} = AES_{ENC}(c_{i2}, \mathcal{N}_i||\sigma_i)$;
- To prevent the corruption of σ_i, AUSF randomly picks $\alpha_i \in_R \mathbb{Z}_p$ to compute $V_{ic} = h^{\alpha_i}$, $c_{ic} = H_0(\mathcal{N}_i, c_{i1}, c_{i3}, V_{ic})$ and $s_{ic} = \alpha_i + c_{ic}a_0$.

AUSF sends $PST_i = (\mathcal{N}_i, c_{i1}, c_{i3}, c_{ic}, s_{ic})$ to ISV, and stores PST_i on UDM.

Upon receiving PST_i, ISV checks whether $\mathcal{N}_i$ has never been used and verifies the validity by calculating $V_{ic} = h^{s_{ic}} \widehat{A}_0^{-c_{ic}}$ and verifying $c_{ic} \stackrel{?}{=} H_0(\mathcal{N}_i, c_{i1}, c_{i3}, V_{ic})$. If either succeeds, ISV aborts and returns failure; otherwise, it delegates the access capability of ISN to U_i by performing the following steps:

- ISV decrypts (c_{i1}, c_{i3}) to obtain σ_i as $c_{i2} = H(c_{i1}, c_{i1}^b)$ and $\mathcal{N}_i||\sigma_i = AES_{DEC}(c_{i2}, c_{i3})$;
- ISV computes $G_i = H_0(\mathcal{N}_i, cert_C, ISN)$. Note that, ISV do not have any information about U_i;
- To delegate the access capability, ISV selects a random $\tau_i \in_R \mathbb{Z}_p$ to calculate

$$\phi_i = (\phi_{i1}, \phi_{i2}) = (\sigma_{i1}^{\tau_i}, (\sigma_{i2}\sigma_{i1}^{bG_i})^{\tau_i}), \tag{2}$$

  which is a service credential of U_i for the access of IoT service ISN;
- For session key negotiation, ISV randomly picks $x_i \in_R \mathbb{Z}_p$ and computes $X_{i1} = g^{x_i}$, $h_{i1} = H_1(\mathcal{N}_i, X_{i1}, cert_C, ISN)$ and $X_{i2} = h_{i1}^b$.
- To prevent the disclosure of ϕ_i, ISV picks a random $w_i \in_R \mathbb{Z}_p$ to calculate $d_{i1} = g^{w_i}$, $d_{i2} = H(d_{i1}, upk_i^{w_i})$, and $d_{i3} = AES_{ENC}(d_{i2}, \mathcal{N}_i||\phi_i||X_{i1}||X_{i2})$;
- To prevent the corruption, ISV randomly chooses $\beta_i \in_R \mathbb{Z}_p$ to compute $W_{ic} = h^{\beta_i}$, $d_{ic} = H_0(\mathcal{N}_i, d_{i1}, d_{i3}, W_{ic})$ and $e_{ic} = \beta_i + d_{ic}b$.

ISV sends the service ticket $ST_i = (\mathcal{N}_i, d_{i1}, d_{i3}, d_{ic}, e_{ic})$ to SMF.

After receiving ST_i, SMF forwards ST_i to U_i, along with the encrypted allowed S-NSSAIs for U_i.

Upon receiving ST_i, U_i verifies the signature (d_{ic}, e_{ic}) by computing $W_{ic} = h^{e_{ic}} \widehat{B}^{-d_{ic}}$ and verifying $d_{ic} \stackrel{?}{=} H_0(\mathcal{N}_i, d_{i1}, d_{i3}, W_{ic})$. If yes, U_i decrypts (d_{i1}, d_{i3}) by calculating $d_{i2} = H(d_{i1}, d_{i1}^b)$ and $\mathcal{N}_i||\phi_i||X_{i1}||X_{i2} = AES_{DEC}(d_{i2}, d_{i3})$. U_i obtains the service credential ϕ_i and the allowed S-NSSAIs. Then, U_i checks whether the service credential ϕ_i is valid or not by computing $G_i' = H_0(\mathcal{N}_i, cert_C, ISN)$ and checking

$$\hat{e}(\phi_{i2}, h) \stackrel{?}{=} \hat{e}(\phi_{i1}, \widehat{A}_1 \widehat{A}_0^{t_i} \widehat{B}^{G_i'}). \tag{3}$$

If ϕ_i is valid, ϕ_i is a qualified service credential for the service ISN and U_i can use ϕ_i to access the IoT service.

Service Authentication. To access ISN, U_i initializes secondary authentication using ϕ_i and allowed S-NSSAIs with U-SSTs=$\{$U-SST$_1$, U-SST$_2$, $\cdots$, U-SST$_s\}$. To allow the controller to choose a proper slice for package transmission, U_i generates a hidden allowed S-NSSAI PAS to match the slice types and features of configured slices. Specifically, for each U-SST$_l$ in U-SSTs, U_i randomly chooses $r_l \in \mathbb{Z}_p$ to compute $\lambda_l = (\widehat{A}_1 h^{\text{U-SST}_l})^{r_l}$ and $\mu_l = g^{r_l}$; further, for each $US_l AI_j$ in an allowed S-NSSAI, for $j \in \{1, 2, \cdots, t\}$, U_i calculates $UCF_{lj} = \mathcal{SE}(H(US_l AI_j, \widehat{C}), US_l AI_j)$, and sets $UCF_l = (UCF_{l1}, UCF_{l2}, \cdots, UCF_{lt})$. In doing so, $PAS = (\lambda_1, \lambda_2, \cdots, \lambda_s, \mu_1, \mu_2, \cdots, \mu_s, UCF_1, UCF_2, \cdots, UCF_s)$. U_i also uses the service credential ϕ_i to generate an authenticated key agreement message AKA_i as follows:

- U_i randomly selects $\nu_i, \omega_i, y_i, \theta_i \in_R \mathbb{Z}_p$, to generate a service authentication message $\phi_{i1}' = \phi_{i1}^{\nu_i}$, $\phi_{i2}' = \phi_{i2}^{\nu_i}$, $\phi_{i3}' = \hat{e}(\phi_{i1}, \widehat{A}_0)^{\nu_i \omega_i}$, $c_{is} = H_0(\mathcal{N}_i, \phi_{i1}', \phi_{i2}', \phi_{i3}', ISN)$ and $s_{is} = \omega_i + c_{is}t_i$;
- To verify the validity of (X_{i1}, X_{i2}), U_i calculates $h_{i1}' = H_1(\mathcal{N}_i, X_{i1}, cert_C, ISN)$ and checks whether

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSAC.2018.2815418, IEEE Journal on Selected Areas in Communications

$\hat{e}(h'_{i1}, \widehat{B}) = \hat{e}(X_{i2}, h)$ holds. If not, U_i aborts and returns failure;

- For session key negotiation, U_i computes $Y_{i1} = g^{y_i}$, $E_{i1} = X_{i1}^{y_i}$, and $h_{i2} = H_2(\mathcal{N}_i, E_{i1}, X_{i1}, Y_{i1}, cert_C, ISN)$;
- To prove the ownership of y_{i1}, U_i computes $Y_{i2} = h_{i2}^{y_i}$, $D_i = g^{\theta_i}$, $g_{i1} = H_0(\mathcal{N}_i, \phi'_{i1}, \phi'_{i2}, \phi'_{i3}, s_{is}, X_{i1}, X_{i2}, Y_{i1}, Y_{i2}, E_{i1}, D_i, cert_C, ISN)$, and $g_{i2} = \theta_i + g_{i1} y_i$.
- U_i sets $AKA_i = (\mathcal{N}_i, \phi'_{i1}, \phi'_{i2}, \phi'_{i3}, s_{is}, X_{i1}, X_{i2}, Y_{i1}, Y_{i2}, E_{i1}, g_{i1}, g_{i2}, cert_C, ISN)$.

Finally, U_i sends (PAS, AKA_i) to RRH fog, and the RRH fog forwards it to the local controller.

After receiving (PAS, AKA_i) from U_i, the controller determines the network slice based on the mapping of configured NSSAIs and allowed NSSAIs of U_i. With NL and PAS, the controller verifies $\hat{e}(PS_{i1}, \lambda_l) = \hat{e}(\mu_l, PS_{i2})$ for each $i \in \{1, 2, \cdots, n\}$ and $l \in \{1, 2, \cdots, s\}$. If no PS_i can make the equation hold, the controller transfers the packages through the default slice; if only one SST_{i^*} succeeds the equation, the controller uses the slice i^*; otherwise, multiple $\text{SST}_{i'}$ fit the service. Thus, the controller selects a network slice from multiple candidates by counting the number of identical feature values in ACF_{i^*} of the configured S-NSSAI SST_{i^*} and UCF_{l^*} of the allowed S-NSSAI U-SST_{l^*} and finding the configured S-NSSAI $\text{SST}_{i'}$ that has the largest number of identical feature values. Thereby, the controller forwards AKA_i and service packages through the slice i'.

When receiving AKA_i, ISV computes $D'_i = g^{g_{i2}} {Y_{i1}}^{-g_{i1}}$ and checks $g_{i1} \stackrel{?}{=} H_0(\mathcal{N}_i, \phi'_{i1}, \phi'_{i2}, \phi'_{i3}, s_{is}, X_{i1}, X_{i2}, Y_{i1}, Y_{i2}, E_{i1}, D'_i, cert_C, ISN)$. If not, it returns failure and aborts. Otherwise, they further independently verifies whether U_i has the available delegation of ISV and core network for service access by computing $c_{is} = H_0(\mathcal{N}_i, \phi'_{i1}, \phi'_{i2}, \phi'_{i3}, ISN)$ and checking whether

$$\phi'_{i3} = \hat{e}(\phi'_{i1}, \widehat{A}_1)^{c_{is}} \hat{e}(\phi'_{i2}, h)^{-c_{is}} \hat{e}(\phi'_{i1}, \widehat{A}_0)^{s_{is}} \hat{e}(\phi'_{i1}, \widehat{B})^{G_i c_{is}} \tag{4}$$

holds or not. If the delegation is invalid, ISV returns failure and aborts; otherwise, the controller and ISV initialize Key Negotiation phase, and SMF builds a new session for U_i, keeps the session number $\mathcal{N}_i$ for session management, and maintains AKA_i on UDM.

Batch Verification. Batch verification is supported to improve the verification efficiency for ISV. Suppose ISV receives m service authentication messages $\{AKA_1, \cdots, AKA_m\}$ from m users $\{U_1, \cdots, U_m\}$ for ISN service access. For each AKA_i, $i \in \{1, 2, \cdots, m\}$, ISV computes $c_{is} = H_0(\mathcal{N}_i, \phi'_{i1}, \phi'_{i2}, \phi'_{i3}, ISN)$, and chooses m random values $\{\rho_1, \rho_2, \cdots, \rho_m \in_R \mathbb{Z}_p\}$ to check whether

$$\begin{aligned} \prod_{i=1}^{m} {\phi'_{i3}}^{\rho_i} \quad = \quad & \hat{e}(\prod_{i=1}^{m} (\phi'_{i1})^{c_{is}\rho_i}, \widehat{A}_1) \cdot \hat{e}(\prod_{i=1}^{m} (\phi'_{i2})^{-c_{is}\rho_i}, h) \\ & \hat{e}(\prod_{i=1}^{m} (\phi'_{i1})^{s_{is}\rho_i}, \widehat{A}_0) \cdot \hat{e}(\prod_{i=1}^{m} (\phi'_{i1})^{G_i c_{is}\rho_i}, \widehat{B}). \end{aligned} \tag{5}$$

Key Negotiation. If U_i is eligible to access the IoT service ISN, the controller, U_i and ISV negotiate a session key. The controller executes the following steps to negotiate the session key for secure interactions with ISV and U_i:

- To verify the validity of (X_{i1}, X_{i2}), the controller computes $h''_{i1} = H_1(\mathcal{N}_i, X_{i1}, cert_C, ISN)$ and checks whether $\hat{e}(h''_{i1}, \widehat{B}) = \hat{e}(X_{i2}, h)$ holds or not. If not, it returns failure and aborts;
- To verify the validity of (Y_{i1}, Y_{i2}), the controller computes $h'_{i2} = H_2(\mathcal{N}_i, E_{i1}, X_{i1}, Y_{i1}, cert_C, ISN)$ and checks whether $\hat{e}(Y_{i1}, h'_{i2}) = \hat{e}(g, Y_{i2})$ holds or not. If not, it returns failure and aborts;
- For session key negotiation, the controller randomly selects $z_i \in_R \mathbb{Z}_p$ to compute $Z_{i1} = g^{z_i}$, $K_i = E_{i1}^{z_i}$, $E_{i2} = X_{i1}^{z_i}$, $E_{i3} = Y_{i1}^{z_i}$, $h_{i3} = H_1(\mathcal{N}_i, K_i, E_{i1}, E_{i2}, E_{i3}, X_{i1}, Y_{i1}, Z_{i1}, cert_C, ISN)$ and $Z_{i2} = h_{i3}^{c}$.
- The controller computes the session key $sk_i = H(\mathcal{N}_i, K_i, X_{i1}, Y_{i1}, Z_{i1}, cert_C, ISN)$.

The controller forwards session key negotiation messages $SKN_{is} = (\mathcal{N}_i, E_{i1}, E_{i2}, E_{i3}, Y_{i1}, Y_{i2}, Z_{i1}, Z_{i2}, cert_C, ISN)$ to ISV, and $SKN_{iu} = (\mathcal{N}_i, E_{i2}, E_{i3}, Z_{i1}, Z_{i2}, Y_{i1}, cert_C, ISN)$ to U_i through the dedicated network slice.

Upon obtaining SKN_{iu}, U_i verifies the validity of (Z_{i1}, Z_{i2}) by computing $K_i = E_{i2}^{y_i}$, $h'_{i3} = H_1(\mathcal{N}_i, K_i, E_{i1}, E_{i2}, E_{i3}, X_{i1}, Y_{i1}, Z_{i1}, cert_C, ISN)$ and verifying whether $\hat{e}(h'_{i3}, \widehat{C}) = \hat{e}(Z_{i2}, h)$ holds. If not, U_i returns failure and aborts; otherwise, U_i computes the session key $sk_i = H(\mathcal{N}_i, K_i, X_{i1}, Y_{i1}, Z_{i1}, cert_C, ISN)$.

Upon receiving SKN_{is}, to verify the validity of (Y_{i1}, Y_{i2}) and (Z_{i1}, Z_{i2}), ISV calculates $K_i = E_{i3}^{x_i}$, $h''_{i2} = H_2(\mathcal{N}_i, E_{i1}, X_{i1}, Y_{i1}, cert_C, ISN)$, $h''_{i3} = H_1(\mathcal{N}_i, K_i, E_{i1}, E_{i2}, E_{i3}, X_{i1}, Y_{i1}, Z_{i1}, cert_C, ISN)$, checks whether $\hat{e}(Y_{i1}, h''_{i2}) = \hat{e}(g, Y_{i2})$ and $\hat{e}(h''_{i3}, \widehat{C}) = \hat{e}(Z_{i2}, h)$ hold. If either does not hold, ISV returns failure and aborts. Otherwise, ISV further computes the session key $sk_i = H(\mathcal{N}_i, K_i, X_{i1}, Y_{i1}, Z_{i1}, cert_C, ISN)$. The session key sk_i is used to achieve secret communications among ISV, local fog and U_i in this session through the specified network slice.

D. Security Analysis

Next, we show that ES^{3}A realizes the design goals defined in III-C, namely, privacy-preserving slice selection, service-oriented anonymous authentication and service-oriented key agreement.

1) Privacy-preserving Slice Selection: In ES^{3}A, both the service types indicated in packages and types of network slices should be protected against fog nodes or outsider attackers for preserving the accessing services of users. Specifically, the slice type SST_i is hidden in PS_i using (a_1, ϵ_i), and thereby it is difficult to recover SST_i from $PS_{i1} = g^{\frac{\epsilon_i}{a_1 + \text{SST}_i}}$, if SST_i is unpredictable. Further, to protect the service type U-SST_l, U_i uses $\widehat{A}_1$ and r_l to encrypt U-SST_l. λ_l is formally defined as $\lambda_l = (\widehat{A}_1 h^{\text{U-SST}_l})^{r_l}$ and $\mu_l = g^{r_l}$. Thus, the protection of U-SST_l relies on the intractability of distinguishing a given λ_l^* from $\text{U-SST}_l^{(1)}$ and $\text{U-SST}_l^{(2)}$. If there is an adversary that is able to distinguish, it is possible to obtain $\widehat{A}_1^{r_l}$, which is the solution of co-CDH problem [43], that is, given $g, g^{r_l} \in \mathbb{G}_1$,$h, h^{a_1} \in \mathbb{G}_2$ for $r_l, a_1 \in_R \mathbb{Z}_p$, to compute $h^{a_1 r_l}$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSAC.2018.2815418, IEEE Journal on Selected Areas in Communications

Our another goal is to protect the feature values in a S-NSSAI, if these values are unpredictable. To hide these values, the deterministic AES encryption is leveraged in ES^{3}A. Specifically, a feature value S_iAI_j either in an allowed S-NSSAI or in configured S-NSSAI is encrypted as $\mathcal{SE}(H(S_iAI_j, \widehat{C}), S_iAI_j)$. Since the AES encryption is secure if S_iAI_j is unpredictable, the ciphertext does not disclose S_iAI_j. As the processing on S_iAI_j is deterministic, two identical S_iAI_j and US_lAI_j are easy to be detected by comparing the corresponding ciphertexts.

2) Service-oriented Anonymous Authentication: The user U_i proves the access capability of the IoT service based on the delegation from ISV without disclosing any information about its identity ID_i. Thus, in anonymous service authentication, we should take into account three issues: user authentication, user anonymity and ISV authentication.

User Authentication. U_i utilizes the access credential ϕ_i to generate the authenticated key agreement message AKA_i to access the IoT service ISN, in which ϕ_i is the key component generated by ISV and AUSF to authenticate U_i. U_i randomizes ϕ_i to generate a signature $(\phi'_{i1}, \phi'_{i2}, \phi'_{i3}, s_{is})$ using the random access identifier t_i. Since t_i is randomly chosen and only known by U_i, no attacker is able to generate a valid $(\phi'_{i1}, \phi'_{i2}, \phi'_{i3}, s_{is})$ without the correct t_i. If an attacker can generate a valid $(\phi'_{i1}, \phi'_{i2}, c_{is}, s_{is})$ with $t_i^* \neq t_i$, we can construct an extractor to output an authentication message ϕ_i^*. If ϕ_i^* is different from the existing service credential generated by ISV and AUSF, it is a successful forgery of the PS signature. Compared with the PS signature in [18], the AUSF's public key is $(A_0, \widehat{A}_0, \widehat{A}_1)$. Since the PS signature is unforgeable under the modified LRSW assumption 2, the unforgeability of the service credential can be reduced to the modified LRSW assumption 1. Therefore, if the modified LRSW assumption 1 holds, no attacker can pretend a delegated user to access the IoT service.

User Anonymity. During the service authentication, neither the controller nor ISV learns any information about U_i' identity. The authenticated key agreement message $AKA_i = (\mathcal{N}_i, \phi'_{i1}, \phi'_{i2}, \phi'_{i3}, s_{is}, X_{i1}, X_{i2}, Y_{i1}, Y_{i2}, E_{i1}, g_{i1}, g_{i2}, cert_C, ISN)$ is generated by U_i and sent to the controller and ISV. In AKA_i, $(\mathcal{N}_i, X_{i1}, X_{i2}, Y_{i1}, Y_{i2}, E_{i1})$ are random values, and (g_{i1}, g_{i2}) are the proof of the temporary secret key y_i, such that they contain no information about U_i. $(\phi'_{i1}, \phi'_{i2}, \phi'_{i3}, s_{is})$ is U_i's signature on Q_i using the random secret key t_i. This tag would not disclose the user's identity as well. Therefore, the user anonymity is preserved.

ISV Authentication. ISV's authenticity is verified by U_i in equation (3). ISV uses b to generate the service credential ϕ_i, which is a PS signature on the commitment Q_i. The generation of ϕ_i can be divided into two phases. In the first phase, AUSF uses its secret key to compute σ_i and ISV further generates ϕ_i from σ_i using its secret key b in the second phase. Since σ_i is a PS signature, the security of σ_i can be guaranteed. The computation from σ_i to ϕ_i follows the sequence aggregate signature in [18]. The sequence aggregate signature is constructed from the public key sharing technique by sharing (a_0, a_1) and b to 5G operator and ISV, respectively. The security of ϕ_i depends on the unforgeability of PS signature. Since the public keys are $(A_0, \widehat{A}_0, \widehat{A}_1, \widehat{B})$, the security of ISV authentication can be reduced to the modified LRSW assumption 1. Therefore, no attacker can compromise ISV authentication, as long as the modified LRSW assumption 1 holds.

3) Service-oriented Key Agreement: The security indicates that the session key cannot be identified, and thereby it is hard for an attacker to learn service data exchanged among U_i, ISV and controller. To ensure the security of session key, key negotiation messages cannot be corrupted under eavesdropping attacks, man-in-the-middle attacks and forgery attacks, and the session key was not exposed during information exchange, even the previous session keys may be exposed. Therefore, we show that the key negotiation message (X_{i1}, X_{i2}), $(X_{i1}, X_{i2}, Y_{i1}, Y_{i2}, E_{i1})$ or $(E_{i1}, E_{i2}, E_{i3}, Y_{i1}, Y_{i2}, Z_{i1}, Z_{i2})$ cannot be forged by attackers with any powerful capability, and it is impossible for attackers to distinguish $K_i = g^{x_i y_i z_i}$ with a random value.

Unforgeability. To prevent attackers from forging key negotiation messages, the unforgeability against chosen message attacks is essential for session key negotiation, in which the attackers are allowed to query any message to generate the session key. We prove that the key negotiation message is unforgeable against chosen message attacks, indicating that the attackers cannot forge a valid key negotiation message for key negotiation, even they can query any message they want.

The unforgeability of ISV's key negotiation message (X_{i1}, X_{i2}) can be reduced to CDH assumption for product groups, that is, given $g, g^\alpha, g^\beta \in \mathbb{G}_1^3$, $h, h^\alpha, h^\beta \in \mathbb{G}_2^3$, it is difficult to compute $h^{\alpha\beta}$. Suppose ISV's public key is h^β. An attacker queries the H_1 oracle to obtain the output of H_1 hash function, and queries to acquire (X_{i1}, X_{i2}) on any random value x_i from the oracle, $(X_{i1} = g^{x_i}, X_{i2} = h^{\beta\gamma})$, in which $\gamma \in_R \mathbb{Z}_p$. If the attacker is able to generate a valid forgery (X_{i1}^*, X_{i2}^*) for the output of H_1 hash function, $h_{i1}^* = h^\alpha$, we can use this forgery to compute $h^{\alpha\beta}$, which is the solution of the CDH problem for product groups. Since the CDH problem for product groups is intractable, it is hard to generate a valid forgery for the key negotiation message. Therefore, it is impossible to forge a valid key negotiation message, if the CDH assumption for product groups holds.

The unforgeability of the user's key negotiation message $(X_{i1}, X_{i2}, Y_{i1}, Y_{i2}, E_{i1})$ depends on the CDH assumption in $\mathbb{G}_1$ [43] and the CDH assumption for product groups. Specifically, given $Y_{i1} = g^{y_i}$ and $X_{i1} = g^{x_i}$, it is impossible to compute $E_{i1} = g^{x_i y_i}$. This is the typical CDH problem in $\mathbb{G}_1$ [43]. The unforgeability of $Y_{i2} = h_{i2}^{y_i}$ has the same format with ISV's key negotiation message, as $Y_{i2} = h_{i2}^{y_i}$ is the user's signature with the temporary secret key y_i. As for the controller's key negotiation message $(E_{i1}, E_{i2}, E_{i3}, Y_{i1}, Y_{i2}, Z_{i1}, Z_{i2})$, its unforgeability also can be reduced to the CDH problem in $\mathbb{G}_1$ and the CDH problem for product groups. In specific, $K_i = E_{i1}^{z_i}$, $E_{i2} = X_{i1}^{z_i}$ and $E_{i3} = Y_{i1}^{z_i}$ are the results of CDH problem in $\mathbb{G}_1$, and $Z_{i2} = h_{i3}^c$ has the same format of ISV's key negotiation message. In summary, both the user's and the controller's key negotiation messages are unforgeable, if the CDH assumption in $\mathbb{G}_1$ and the CDH assumption for

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSAC.2018.2815418, IEEE Journal on Selected Areas in Communications

TABLE I
COMPUTATIONAL OVERHEAD OF ES^{3}A (UINT: MS)

PHASES	U_i	Controller	Core Network	ISV
System Initialization	/	SM_1	SM_1+$2SM_2$	SM_2
Network Slicing	/	/	nSM_1+nSM_2+$ntAES_T$	/
5G Network Access	SM_1	/	/	/
Service Delegation	$5SM_1$+$5SM_2$+AES_T+$2BP$	/	$6SM_1$+$3SM_2$+$2AES_T$+$2BP$	$6SM_1$+$5SM_2$+$2AES_T$
Service Authentication	$(s+5)SM_1$+$(2s+1)SM_2$+$stAES_T$+$3BP$+Exp_T	$2snBP$	/	$2SM_1$+$4BP$+$4Exp_T$
Key Negotiation	SM_1+$2BP$	$5SM_1$+$4BP$	/	SM_1+$4BP$

product groups hold.

Confidentiality. The confidentiality of session keys is formally defined by distinguishing the negotiated session keys from random values. Even if an attacker can capture all information about key negotiation messages, $(X_{i1}, X_{i2}, Y_{i1}, Y_{i2}, Z_{i1}, Z_{i2}, E_{i1}, E_{i2}, E_{i3})$, and know all the previous session keys, it cannot distinguish K_i from a random value R. We show that the confidentiality of K_i depends on the Deterministic Diffie-Hellman problem in $\mathbb{G}_1$ [43], that is, given $g, g^\alpha, g^\beta, R \in \mathbb{G}_1$, to determine $g^{\alpha\beta} \stackrel{?}{=} R$. To show the confidentiality of ISV's session key, we set $X_{i1} = g^\alpha$, $Y_{i1} = g^\beta/g^\gamma$ and $Z_{i1} = g^\gamma$, where $\gamma \in_R \mathbb{Z}_p$. The attacker aims to distinguish between $g^{\alpha\beta}$ and R. The attacker can launch H_1 queries to obtain the hash values, reveal queries to have the session keys, and corrupt queries to acquire the secret key of any party. After the queries, the attacker is given a challenge R^*, and required to determine whether $R^* = g^{\alpha\beta}$ or $R^* = R$. If the attacker is able to make the correct decision with the probability more than 1/2, we can use the decision to solve the DDH problem in $\mathbb{G}_1$ [43]. In addition, the proof of the confidentiality of the user's session key and the controller's session key is similar to that of the indistinguishability of ISV's session key, and their confidentiality can be reduced to the DDH problem in $\mathbb{G}_1$ as well. Therefore, the session key is confidential, if the DDH assumption in $\mathbb{G}_1$ holds.

V. PERFORMANCE EVALUATION

In this section, we evaluate the computational and communication overhead of ES^{3}A to illustrate its performance in implementation.

A. Computational Overhead

The computational overhead is discussed through theoretical analysis and simulation on a smart phone and a laptop.

1) Theoretical Analysis: ES^{3}A is deployed on transport layer to guarantee service-oriented authentication and secure data transmission. The public-key certificates are exchanged first and then ES^{3}A is executed among ISV, 5G core network, fog nodes and user devices. The computational burden brought by ES^{3}A immediately impacts system performance. To evaluate the computational overhead, we count the number of time-consuming cryptographic operations in each phase, including scalar multiplication in $\mathbb{G}_1/\mathbb{G}_2$, bilinear pairing, exponentiation in $\mathbb{G}_T$, and AES encryption/decyrption. Other operations, such as point addition, integer multiplication and hash function, are not resource-consuming compared with the above four operations. Assume SM_1, SM_2 BP, Exp_T and AES_T denote the scalar multiplication in $\mathbb{G}_1$, scalar multiplication in $\mathbb{G}_2$, bilinear pairing $\hat{e}$, exponentiation in $\mathbb{G}_T$ and AES encryption/decyrption, respectively. The number of cost-sensitive operations executed by each entity in every phase of ES^{3}A is illustrated in Table I.

2) Simulation Setting: We conduct a simulation of ES^{3}A to record its execution time in practice. The operations of fog nodes, IoT server and ISV are executed on a laptop with Intel Core i5-4200U CPU @2.29GHz and 4.00GB memory. The operation system is 64-bit Windows 10 and the C++ compiler is Visual Studio 2008. The operations of users are executed on a smart phone, Huawei MT2-L01 with CPU Kirin 910 @1.6GHz with 1250M Memory. The operation system is Android 4.2.2 and the C++ compiler is NDK r8d. We use MIRACL library 5.6.1 to implement number-theoretic based methods of cryptography. The R-ATE pairing [44] is utilized to realize the type-3 bilinear pairing. The Barreto–Naehrig curve [44], i.e., $\mathbb{F}_p$-256BN, $E : y^2 = x^3 + 3$ defined over $\mathbb{F}_p$. z is an integer so that $n = 36z^4 + 36z^3 + 18z^2 + 6z + 1$ and $p = 36z^4 + 36z^3 + 24z^2 + 6z + 1$ are prime and $\#E(\mathbb{F}_p) = n$. The embedding degree $k = 12$ is the smallest positive integer with $n|p^4 - 1$. $E[n] \subsetneq E(\mathbb{F}_{p^{12}})$, where $E(n)$ denotes the set of all $n-$torsion points on E. Let $\mathbb{G}_1 = E(\mathbb{F}_p)$, $\mathbb{G}_2$ be the trace–0 order–n subgroup of $E(n)$, and $\mathbb{G}_T$ be the order–n subgroup of $\mathbb{F}^*_{p^{12}}$. p is a large prime with approximately 160 bits.

3) Simulation Results: The execution time of the operations in each phase of ES^{3}A for every entity is depicted in Table II, when $n = 10$, $s = 2$ and $t = 10$. The result indicates that the computation tasks are acceptable for smart phones. To reduce the computational overhead for user devices, $\hat{e}(\phi_{i1}, \widehat{A_0})$ can be pre-computed after U_i obtains the service credential and all bilinear pairing operations can be executed by fog nodes using fog-aided computation. To show the advantages of ES^{3}A on computational overhead, we compare ES^{3}A with several anonymous authentication protocols (YHWD [25], CPAL [26], LCCH [27], ASMC [45] and YCWL [46]) executed under the same setting. Since these schemes do not support key agreement, we compare the computational burden on the generation of authentication messages for U_i and the verification of authentication messages for ISV. The comparison results are shown in Fig. 4(a) and Fig. 4(b). Specifically, Fig. 4(a) shows the comparison results about computational overhead of users on the generation of the service authentication message AKA_i. ES^{3}A is the most efficient one in six protocols with the increasing number of users. The total time cost on AKA_i generation is pretty low even the number of users reaches 50. In Fig. 4(b), the time cost of ISV on the verification of service authentication in ES^{3}A is still lower than YHWD [25], CPAL [26], LCCH [27], ASMC [45] and YCWL [46].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSAC.2018.2815418, IEEE Journal on Selected Areas in Communications

Fig. 4(c) demonstrates the fact that batch verification in the service authentication phase can significantly improve the computational efficiency for ISV compared with the separate verification. For example, if ISV verifies service authentication messages of 50 users one by one, it costs around 3.4s, while the time cost is less than 0.5s if ISV uses batch verification.

To demonstrate whether ES^{3}A is suitable for service-oriented authentication of IoT devices that have lower computing capability than smart phones, we change CPU frequency of the smart phone by using a software to simulate different IoT devices. Although the computing capability is not only determined by CPU frequency, but it has huge impacts on the processing speed. In the simulation, we leverage different CPU frequency, including 208MHz, 418MHz, 624MHz, 798MHz, 1196MHz and 1596MHz, to execute the operations of users. Apple Watch has the processor speed 520MHz and Google Glass uses OMAP 4430 that has up to 1.2GHz frequency. Intel Atom® processor E3900 series, empowering real-time computing in digital surveillance, in-vehicle experiences and industrial automation, have 1.1–1.6 GHz CPU HFM frequency. The CPU frequency of these IoT devices is in our simulating range. Fig 4(d) illustrates the executing time of users' operations in service delegation, service authentication and key negotiation phases on the CPU with 208MHz–1.596GHz. The operations are still efficient even the CPU frequency is low. Therefore, ES^{3}A is applicable on many IoT devices, such as Apple Watch, Google Glass, smart vehicles, and industrial devices.

B. Communication Overhead

ES^{3}A does not possess high communication overhead on 5G network to achieve service-oriented authentication and key agreement for IoT services. The entities exchange and verify the public-key certificates. If the certificates are valid, they begin to perform authentication and key agreement procedure. U_i interacts with AUSF to achieve primary authentication and applies for the service delegation from both 5G core network and IoT server. In this process, U_i sends 1472-bit message to AMF and AUSF forwards 1152-bit PST_i to ISV. ISV computes the service ticket ST_i, which is 1120 bits, and sends it to U_i. To access ISN, U_i is required to send AKA_i to the controller and ISV, which is of binary length 4032 bits. To negotiate the session key, the controller forwards 1632-bit SKN_{is} to ISV and 1280-bit SKN_{iu} to U_i. Finally, U_i utilizes the session key to securely communicate with the fog and ISV for service access.

We also compare the communication overhead of ES^{3}A with YHWD [25], CPAL [26], LCCH [27], ASMC [45] and YCWL [46]. Fig. 4(e) shows the comparison result of communication overhead on service delegation. ES^{3}A is more efficient than YHWD [25], CPAL [26], ASMC [45] and YCWL [46] on service credential transmission. Fig 4(f) illustrates the result of comparison about communication overhead on service authentication. Only YHWD [25] is more efficient than ES^{3}A. Therefore, ES^{3}A possesses low communication bandwidth on the transmission of service credentials and service authentication messages compared with CPAL [26], ASMC [45] and YCWL [46].

TABLE II
EXECUTION TIME OF ES^{3}A (UNIT: MILLISECOND)

PHASES	U_i	Controller	Core Network	ISV
System Initialization	/	1.465	6.352	3.246
Network Slicing	/	/	48.531	/
5G Network Access	4.460	/	0.0242	/
Service Delegation	168.605	/	50.135	26.4285
Service Authentication	226.191	562.524	/	65.735
Key Negotiation	106.353	69.256	/	64.217

VI. CONCLUSIONS

In this paper, we have proposed an efficient and secure service-oriented authentication framework for IoT services in 5G network. Under this framework, a privacy-preserving slice selection mechanism is introduced to allow fog nodes to select proper network slices for data forwarding, and hide the accessing service types of users. Further, service-oriented anonymous authentication and key agreement are achieved to ensure the anonymity and authenticity of users and the confidentiality of service data. Specifically, users are able to anonymously authenticate to IoT servers based on the delegation of both 5G core network and IoT servers, and build secure data channels for the access of the service data cached on local fogs and maintained on remote IoT servers. We have demonstrated the security and privacy preservation of the proposed framework and its efficiency and practicality through simulation. In the future work, we will design network slicing-based privacy-preserving authenticated key agreement protocols for roaming services with efficient access delegation and revocation in 5G networks.

REFERENCES

[1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Commun. Surv. Tutor.*, vol. 17, no. 4, pp. 2347–2376, 2015.

[2] A. Athar, M. H. Rehmani, and A. Rachedi, "Cognitive-radio-based Internet of Things: Applications, Architectures, Spectrum Related Functionalities, and Future Research Directions", *IEEE Wirel. Commun.*, vol. 24, no. 3, pp. 17–25, 2017.

[3] 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. Sel. Areas Commun.*, vol. 32, no. 6, pp. 1065–1082, 2014.

[4] D. Evans, "The Internet of Things: How the Next Evolution of the Internet is Changing Everything," *Cisco White Paper*, 2011.

[5] F. Semiconductor, "The Internet of Things: Five Myths and Realities," *Mouser Electronics*, 2016.

[6] T. Taleb, A. Ksentini, and B. Sericola, "On Service Resilience in Cloud-native 5G Mobile Systems," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 3, pp. 483–496, 2016.

[7] X. Shen, "Device–to–device Communication in 5G Cellular Networks," *IEEE Netw.*, vol. 29, no. 2, pp. 2–3, 2015.

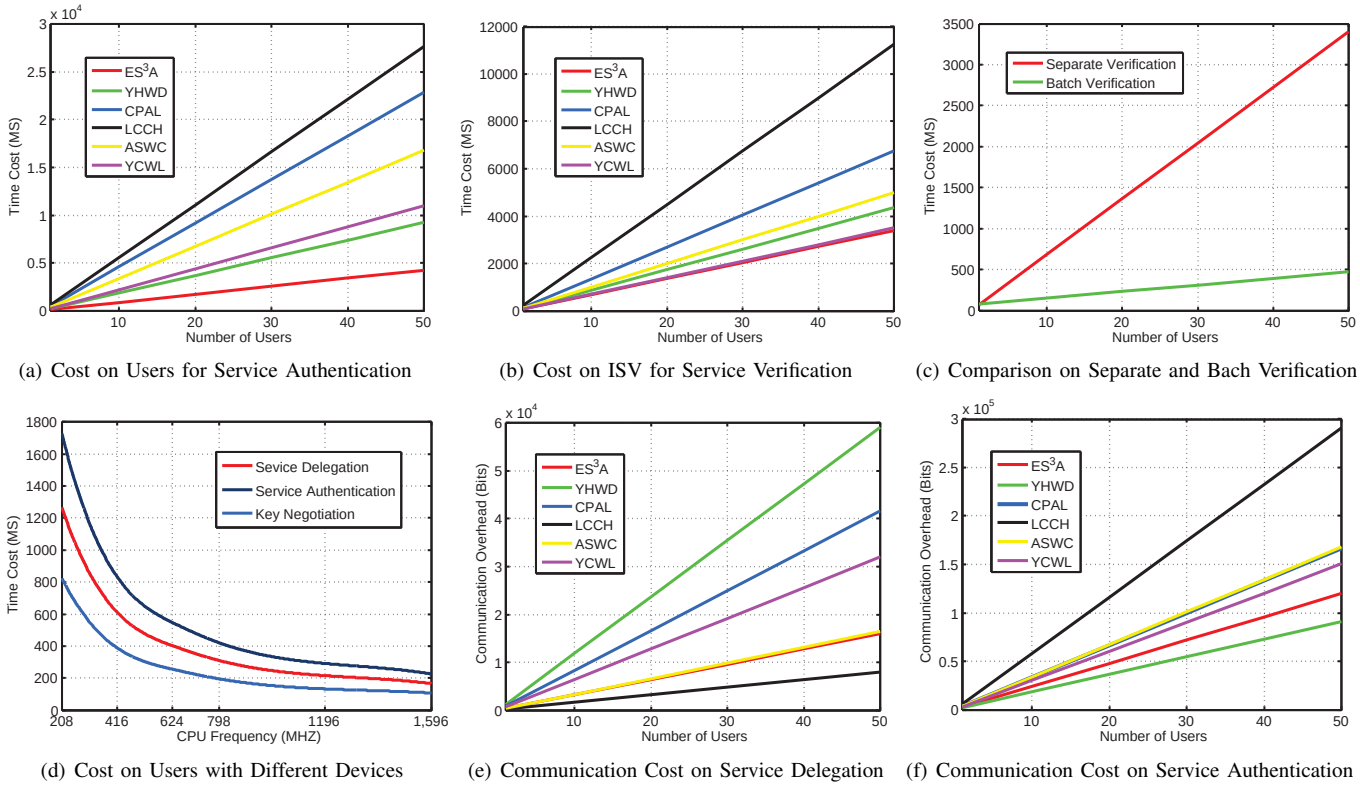
[8] M. R. Palattella, M. Dohler, A. Grieco, G. Rizzo, J. Torsner, T. Engel, and L. Ladid, "Internet of Things in the 5G Era: Enablers, Architecture, and Business Models," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 3, pp. 510–527, 2016.

[9] Nokia, "Dynamic End-to-end Network Slicing for 5G," *White Paper*, 2016.

[10] H. Zhang, N. Liu, X. Chu, K. Long, A. H. Aghvami, and V. C. Leung, "Network Slicing Based 5G and Future Mobile Networks: Mobility, Resource Management, and Challenges," *IEEE Commun. Mag.*, vol. 55, no. 8, pp. 138–145, 2017.

[11] L. M. Vaquero and L. Rodero-Merino, "Finding Your Way in the Fog: Towards A Comprehensive Definition of Fog Computing," *ACM SIGCOMM Comp. Commun. Rev.*, vol. 44, no. 5, pp. 27–32, 2014.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSAC.2018.2815418, IEEE Journal on Selected Areas in Communications



(a) Cost on Users for Service Authentication

(b) Cost on ISV for Service Verification

(c) Comparison on Separate and Bach Verification

(d) Cost on Users with Different Devices

(e) Communication Cost on Service Delegation

(f) Communication Cost on Service Authentication

Fig. 4. Comparison about Computational and Communication Overhead

[12] R. Morabito, I. Farris, A. Iera, and T. Taleb, "Evaluating Performance of Containerized IoT Services for Clustered Devices at the Network Edge", *IEEE Internet Thing J.*, vol. 4, no. 4, pp. 1019–1030, 2017.

[13] 3GPP, "3GPP TS 33.501, Security Architecture and Procedures for 5G System (Release 15)," *3rd Generation Partnership Project: Sophia-Antipolis*, France, 2017.

[14] M. A. Ferrag, L. Maglaras, A. Argyriou, D. Kosmanos, and H. Janicke, "Security for 4G and 5G Cellular Networks: A Survey of Existing Authentication and Privacy-preserving Schemes," *ArXiv Preprint*, arXiv:1708.04027, 2017.

[15] C. Bing, "4G LTE Protocols Used by Smartphones Can Be Hacked, Researchers Found," *Syberscoop*, Nov. 2016.

[16] J. Ni, K. Zhang, X. Lin, and X. Shen, "Securing fog Computing for Internet of Things applications: Challenges and solutions," *IEEE Commun. Surv. Tutor.*, vol. 20, no. 1, pp. 601–628, 2018.

[17] J. Ni, K. Zhang, Y. Yu, X. Lin, and X. Shen, "Providing Task Allocation and Secure Deduplication for Mobile Crowdsensing via Fog Computing", *IEEEE Trans. Dependable Sec. Comput.*, DOI: 10.1109/TDSC.2018.2791432, 2018.

[18] D. Pointcheval and O. Sanders, "Short Randomizable Signatures," in *Proc. of CT-RSA*, pp. 111–126, 2016.

[19] J. Katz and Y. Lindell, "Introduction to Modern Cryptography," CRC Press, 2014.

[20] Q. Li, G. Wu, A. Papathanassiou, and U. Mukherjee, "An End–to–end Network Slicing Framework for 5G Wireless Communication Systems," *ArXiv Preprint*, arXiv:1608.00572, 2016.

[21] K. Samdanis, X. Costa–Perez, and V. Sciancalepore, "From Network Sharing to Multi–tenancy: The 5G Network Slice Broker," *IEEE Commun. Mag.*, vol. 54, no. 7, pp. 32–39, 2016.

[22] X. Zhou, R. Li, T. Chen, and H. Zhang, "Network Slicing as a Service: Enabling Enterprises' Own Software–defined Cellular Networks," *IEEE Commun. Mag.*, vol. 54, no. 7, pp. 146–153, 2016.

[23] C. Liang, F. R. Yu, H. Yao, and Z. Han, "Virtual Resource Allocation in Information–centric Wireless Networks with Virtualization," *IEEE Trans. Veh. Technol.*, vol. 65, no. 12, pp. 9902–9914, 2016.

[24] 3GPP, "3GPP TS 33.899, Study on the Security Aspects of the Next Generation System (Release 14)," *3rd Generation Partnership Project: Sophia-Antipolis*, France, 2017.

[25] G. Yang, Q. Huang, D. S. Wong, and X. Deng, "Universal Authentication Protocols for Anonymous Wireless Communications," *IEEE Trans. Wirel. Commun.*, vol. 9, no. 1, pp. 168–174, 2010.

[26] C. Lai, H. Li, X. Liang, R. Lu, K. Zhang, and X. Shen, "CPAL: A Conditional Privacy–preserving Authentication with Access Linkability for Roaming Service," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 46–57, 2014.

[27] J. K. Liu, C. -K. Chu, S. S. M. Chow, X. Huang, M. H. Au, and J. Zhou, "Time–bound Anonymous Authentication for Roaming Networks," *IEEE Trans. Inf. Forensic Secur.*, vol. 10, no. 1, pp. 178–189, 2015.

[28] D. Choi, H. -K. Choi, and S. -Y. Lee, "A Group–based Security Protocol for Machine-type Communications in LTE–advanced," *Wirel. Netw.*, vol. 21, no. 2, pp. 405–419, 2015.

[29] J. Li, M. Wen, and T. Zhang, "Group-based Authentication and Key Agreement with Dynamic Policy Updating for MTC in LTE–A Networks," *IEEE Internet Things J.*, vol. 3, no. 3, pp. 408–417, 2016.

[30] C. Lai, R. Lu, D. Zheng, H. Li, and X. Shen, "GLARM: Group–based Lightweight Authentication Scheme for Resource-constrained Machine to Machine Communications," *Comput. Netw.*, vol. 99, pp. 66–81, 2016.

[31] P. Rost, C. Mannweiler, D. S. Michalopoulos, C. Sartori, V. Sciancalepore, N. Sastry, O. Holland, et al., "Network Slicing to Enable Scalability and Flexibility in 5G Mobile Networks," *IEEE Commun. Mag.*, vol. 55, no. 5, no. 72–79, 2017.

[32] N. Yang, L. Wang, G. Geraci, M. Elkashlan, J. Yuan, and M. D. Renzo, "Safeguarding 5G Wireless Communication Networks Using Physical Layer Security," *IEEE Commun. Mag.*, vol. 53, no. 4, pp. 20–27, 2015.

[33] Z. Yan, P. Zhang, and A. V. Vasilakos, "A Security and Trust Framework for Virtualized Networks and Software–defined Networking," *Secur. Commun. Netw.*, vol. 9, no. 16, pp. 3059–3069, 2016.

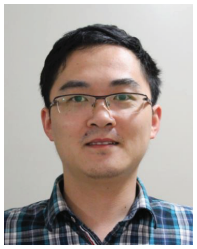
[34] M. H. Eiza, Q. Ni, and Q. Shi, "Secure and Privacy–aware Cloud–assisted Video Reporting Service in 5G–enabled Vehicular Networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 10, pp. 7868–7881, 2016.

[35] 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, 2015.

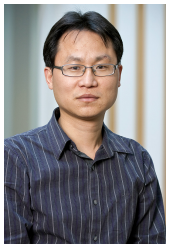
[36] M. Bagaa, T. Taleb, and A. Ksentini, "Efficient Tracking Area Management Framework for 5G Networks," *IEEE Trans. Wirel. Commun.*, vol. 15, no. 6, pp. 4117–4131, 2016.

[37] D. Bendouda, A. Rachedi, and H. Haffaf, "Programmable Architecture Based on Software Defined Network for Internet of Things: Connected

Dominated Sets Approach", *Futur. Gener. Comp. Syst.*, vol. 80, pp. 188–197, 2018.
[38] G. Brown, "Service-oriented 5G Core Networks," *Huawei White Paper*, Feb. 2017.
[39] 3GPP, "3GPP TS 23.501, System Architecture for the 5G System (Release 15)," *3rd Generation Partnership Project: Sophia-Antipolis*, France, 2017.
[40] P. Yang, N. Zhang, Y. Bi, L. Yu, and X. Shen, "Catalyzing Cloud-Fog Interoperation in 5G Wireless Networks: An SDN Approach," *IEEE Netw.*, vol. 31, no. 5, pp. 14–20 2017.
[41] H. Zhang, Y. Qiu, X. Chu, K. Long, and V. Leung, "Fog Radio Access Networks: Mobility Management, Interference Mitigation and Resource Optimization," *ArXiv Preprint*, arXiv:1707.06892, 2017.
[42] F. Vercauteren, "Final Report on Main Computational Assumptions in Cryptography," *ECRYPT II*, Jan. 2013.
[43] D. Boneh and X. Boyen, "Short Signatures without Random Oracles," in *Proc. of EUROCRYPT*, 2004, pp. 56–73.
[44] J. Fan, F. Vercauteren, and I. Verbauwhede, "Faster $\mathbb{F}_p$–Arithmetic for Cryptographic Pairings on Barreto-Naehrig Curves", in *Proc. of CHES*, 2009, pp. 240–253.
[45] M. H. Au, W. Susilo, Y. Mu, S. S. Chow, "Constant-Size Dynamic k-Times Anonymous Authentication", *IEEE Syst. J.*, vol. 7, no. 2, pp. 249–261, 2013.
[46] Y. Yang, H. Cai, Z. Wei, H. Lu, K. K. R. Choo, "Towards Lightweight Anonymous Entity Authentication for IoT Applications", in *Proc. of ACISP*, 2016, pp. 265–280



Jianbing Ni (S'16) received the B.E. degree and the M.S. degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2011 and 2014, respectively. He is currently working toward the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. His research interests are applied cryptography and information security, with current focus on cloud computing, 5G network, mobile crowdsensing and Internet of Things.



Xiaodong Lin (M'09-SM'12-F'17) received the Ph.D. degree in information engineering from Beijing University of Posts and Telecommunications, Beijing, China, and the Ph.D. degree in electrical and computer engineering (with Outstanding Achievement in Graduate Studies Award) from the University of Waterloo, ON, Canada. He was an Associate Professor of information security with the Faculty of Business and Information Technology, University of Ontario Institute of Technology (UOIT), Canada. Currently, he is an Associate Professor, Department of Physics and Computer Science, Wilfrid Laurier University, Waterloo, Ontario, Canada. His research interests include wireless network security, applied cryptography, computer forensics, software security, and wireless networking and mobile computing.

Xuemin (Sherman) Shen (M'97-SM'02-F'09) received Ph.D. degree from Rutgers University, New Jersey (USA) in electrical engineering, 1990. Dr. Shen is a University Professor, Department of Electrical and Computer Engineering, University of Waterloo, Canada. His research focuses on resource management in interconnected wireless/wired networks, wireless network security, social networks, smart grid, and vehicular ad hoc and sensor networks. Dr. Shen is a registered Professional Engineer of Ontario, Canada, an IEEE Fellow, an Engineering Institute of Canada Fellow, a Canadian Academy of Engineering Fellow, a Royal Society of Canada Fellow, and a Distinguished Lecturer of IEEE Vehicular Technology Society and Communications Society.

Dr. Shen is the Editor-in-Chief for IEEE Internet of Thing Journal and the vice president on publications of IEEE Communications Society. He received the Joseph LoCicero Award in 2015 and the Education Award in 2017 from the IEEE Communications Society. He has also received the Excellent Graduate Supervision Award in 2006, and the Outstanding Performance Award in 2004, 2007, 2010, and 2014 from the University of Waterloo, the Premier's Research Excellence Award (PREA) in 2003 from the Province of Ontario, Canada. Dr. Shen served as the Technical Program Committee Chair/Co-Chair for IEEE Globecom' 16, IEEE Infocom'14, IEEE VTC'10 Fall, the Symposia Chair for IEEE ICC'10, the Tutorial Chair for IEEE VTC'11 Spring, the Chair for IEEE Communications Society Technical Committee on Wireless Communications, and P2P Communications and Networking.