LAMANCO: A Lightweight Anonymous Mutual Authentication Scheme for *N*-Times Computing Offloading in IoT

Fei Wang, *Member, IEEE*, Yongjun Xu, *Member, IEEE*, Liehuang Zhu, *Member, IEEE*, Xiaojiang Du, *Senior Member, IEEE*, and Mohsen Guizani, *Fellow, IEEE*

***Abstract*—Nowadays in many application scenarios of Internet of Things (IoT), low latency is achieved at the cost of computing-complexity which is beyond the capabilities of IoT devices. Offloading the computing intensive tasks to more powerful edge devices is expected to provide new generation computing-intensive and delay-sensitive services. In the three hierarchy architecture user/IoT-edge-cloud, private and secure mutual authentication are necessary between user, IoT device, and edge device. However, in the emerging computing paradigms, such as mobile transparent computing, edge computing, fog computing, and several threats, such as edge device compromise, privacy leaking, and denial of service (DoS) might crash the security of the system. Here, we propose a lightweight anonymous mutual authentication scheme for *n*-times computing offloading (CO) in IoT. In our novel scheme, through a smartcard as token and an edge device as a security proxy, a user is able to subscribe or renew *n*-times CO service and consume it securely in daily use. Moreover, both IoT and edge devices authenticate each other anonymously without leaking user's sensitive information, which will preserve the privacy even when an edge device is comprised. Finally, our scheme is based on lightweight one-way hash function and MAC function, therefore the adversary is not able to perform a DoS attack. To evaluate the solution, a security analysis and a performance analysis are presented. Compared with similar schemes, our approach achieves all designed security features and achieves a 1.66× and 2.87× of computing speed on IoT and edge devices, respectably.**

***Index Terms*—Authentication, computing offloading (CO), Internet of Things (IoT), privacy preserving.**

Manuscript received May 11, 2018; revised September 27, 2018 and November 22, 2018; accepted December 11, 2018. Date of publication December 19, 2018; date of current version June 19, 2019. This work was supported in part by the National Natural Science Foundation of China under Grant 61602447 and in part by the Innovation Foundation of the Chinese Academy of Sciences under Grant CXJJ-17-M116. This work also thanks the K. C. Wong Education Foundation for the support. *(Corresponding author: Fei Wang.)*

F. Wang and Y. Xu are with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China (e-mail: wangfei@ict.ac.cn; xyj@ict.ac.cn).

L. Zhu is with the School of Computer Science, Beijing Institute of Technology, Beijing 100081, China (e-mail: liehuangz@bit.edu.cn).

X. Du is with the Department of Computer and Information Sciences, Temple University, Philadelphia, PA 19122 USA (e-mail: dxj2005@gmail.com).

M. Guizani is with the College of Engineering, Qatar University, Doha, Qatar (e-mail: mguizani@qu.edu.qa).

Digital Object Identifier 10.1109/JIOT.2018.2888636

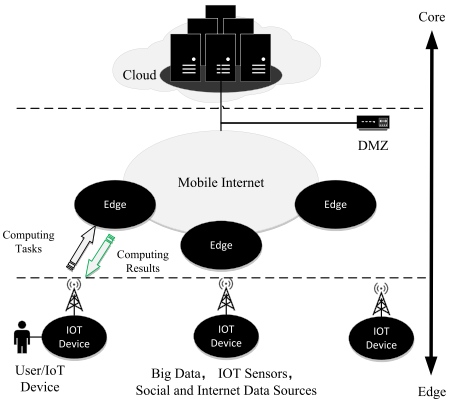


Fig. 1. User/IoT-edge-cloud architecture.

I. Introduction

NOWDAYS billions of Internet of Things (IoT) devices are providing a variety of services ranging from augmented/virtual realities to content delivery and caching, distributed data analysis, and artificial intelligence. Due to the user's need for fluent experience and low delay, these services are not only computing-intensive but also delay-sensitive. However, in many application scenarios of IoT, low latency is achieved at the cost of computing-complexity which is beyond the capabilities of IoT devices. Emerging computing paradigms, such as mobile transparent computing, edge computing, and fog computing will provide the necessary support [1]–[3].

Researchers proposed slightly different designs of system architecture for these paradigms [26]–[29]. But here we adopt a three hierarchy architecture of edge computing, which add a mobile edge computing (MEC) layer between the IoT devices and Cloud as shown in Fig. 1, where the term MEC was already standardized by European Telecommunications Standards Institute and Industry Specification Group and accepted by Intel, Vodafone, IBM, Huawei, NTT DOCOMO, and so on. Also, MEC is acknowledged as a prime emerging technology for 5G networks [4].

MEC employs more resourceful and more advanced edge devices [30], e.g., small-scale servers, smartphones, laptops, and small data centers to assist IoT devices as shown in Fig. 1. By offloading the computing-intensive tasks to the

edge devices, it is expected to converge the data collection at IoT devices and the data processing at edge devices to provision computing-intensive and delay-sensitive services. We call this technique computing offloading (CO). In such hierarchy architecture, users are only able to see different IoT devices along with different services, which means the edge devices are transparent to them.

Security issues need to be addressed when designing edge computing system especially when it comes to CO between IoT devices and edge devices. Several threats, such as message forgery, edge device compromise, privacy leaking, and denial of service (DoS) might crash the security of the system and cause unpredictable chaos.

Authentication is a fundamental service which needs to be employed to prevent that a marvelous edge user or edge device pretends to be legitimate. In order to keep user's sensitive information to himself/herself and avoid that an adversary achieves user's location, interests, or habits, anonymity must be guaranteed to users as long as they behave legitimately. However, once malicious behaviors are detected, the centered authority has the right to trace and reveal any user's clear identity, such that evidences will be saved and accusation will be solved. Moreover, some schemes are based on virtual identities or instant identities, adversary might be able to link multiple messages or transactions to the same user through such identities even user's clear identity is blinded from adversary. So the unlinkability of virtual identities must be available. Due to that one-time CO task might incur large storage, computing resources on edge devices, a secure charging mechanism is necessary. In such a mechanism, users can anonymously subscribe CO service before enjoying the IoT service (or renew it after having used them up), while the edge devices are able to charge user simultaneously. As edge devices are ubiquitous, the adversary can easily compromise some of them and pry into user's uploaded information. At last, there exists scenarios that a large number of IoT devices generate a large number of CO task request in a short time interval, or even worse that adversary forges a large number of requests and jams it in communication channel, which both cause DoS.

In the literature, variety of authentication techniques have been proposed for mobile cloud environment [5]–[8], fog computing [9], [10], and vehicular ad-hoc network [11], [12], however most of traditional authentication schemes are not targeting the mutual anonymous authentication for CO in user/IoT-edge-cloud architecture.

Here, we propose a lightweight anonymous mutual authentication scheme for n-times CO in IoT (LAMANCO). In our novel scheme, through a smart card as token and an edge device as security proxy, user is able to subscribe or renew n-times CO service and consume it securely in daily use. Moreover, both IoT and edge devices authenticate each other anonymously without leaking user's sensitive information, which will preserve the privacy even with some edge devices being comprised. At last, our scheme is based on lightweight one-way hash function and MAC function, therefore the adversary is not able to perform DoS attack. As far as we are concerned, our scheme is the first mutual anonymous authentication scheme for CO subscription or renewing, which directly targets user/IoT-edge-cloud architecture. The advantages of our proposed LAMANCO scheme are as follows.

1) *User Anonymity:* Both users and IoT devices acquire pseudo identities during registration which are used to cover clear identities. Therefore, user/IoT devices are anonymous to edge devices or adversary whether it is in CO subscription/renewing or CO requesting phase.
2) *Conational Message Tracing:* Clear identity and corresponding information of users, IoT devices and edge devices are all stored in CA. CA is able to trace any illegitimate message in open channel when the system administrator needs to solve accusations.
3) *Unlinkability:* Even with multiple messages collected from open channels, the adversary cannot link several messages to one user or IoT device, or find some common information from messages generated from one user or IoT device.
4) *Secure n-Times CO:* In the proposed scheme, the cloud will generate and share with user initial hash chain value $h(n1)$, a hash chain whose length is based on the times user prepaying for CO service. The user will compute the latest hash-chain value and use it to acquire CO service from edge device securely.
5) *Resilience to Edge Compromise:* When an adversary steal and break into an edge device, he acquires clear identity of the edge device. However, he/she no longer obtains secret information from tamper proof devices. So he cannot create fake subscription/renewing replies or impersonate as a normal edge device to gain more from CA.
6) *Resilience to DoS:* Our novel scheme uses only lightweight hash functions and message authentication code functions, which are based on elliptical curve cryptography (ECC) and symmetric cryptographic system that are used for the anonymous mutual authentication phase. This makes LAMANCO lightweight enough and save nearly a half of computation cycle compared with closely related schemes.

The remainder of this paper is organized as follows. Section II presents the state-of-the-art research. In Section III, the system model is defined. In Sections IV and V, we describe our main idea and the full scheme. Sections V and VI analyze the security features and performance of our scheme. At last in Section VIII, we conclude this paper and give a discussion of future work.

II. Related Work

A bulk of research work has been proposed for mutual authentication in different scenarios for user-IoT or user–server architecture [5]–[8], [23]–[25] and basic cryptographic technology such as key management [31]–[33] are practical enough in IoT environment. However, security and privacy issues were rarely studied to directly satisfy the requirements of edge computing under user/IoT-edge-cloud architecture.

Some research works [5]–[8] are in the context of mobile cloud computing environment, which focus on remote cloud user–server authentication, where user and cloud server authenticate mutually. However, these schemes are not

suitable for CO in edge computing context, because they do not take intermediate layer into consideration. Gope and Das [8] proposed an anonymous mutual authentication scheme for mobile cloud computing, in which the user is able to enjoy n times ubiquitous services securely, it provides clues to solve the service charging problem in user/IoT-edge-cloud architecture.

Many schemes focus on anonymous mutual vehicle-to-vehicle authentication [11], [12]. Based on ECC, they have exploited the user of tamper proof device and achieved advanced security feature of anonymity, unlinkability, conditional traceability, resilience to RSU compromise, and DoS resilience. However, they are hardly directly applied to IoT-edge CO because they cannot support the service subscription and billing mechanism.

Some research works provide solution for mutual authentication for three-layer hierarchy architecture, but does not protect the user's anonymity. Ibrahim [9] presented a secure and efficient mutual authentication scheme for edge-fog-cloud network architecture, which cannot preserve user's anonymity and suffers from the attack of adversary linking multiple messages to break user's sensitive information. Conditional traceability is of course not considered.

Few research work really achieve anonymous mutual authentication for three-layer hierarchy architecture. Recently, Amor *et al.* [10] proposed a solution, which lies on that the fog user and the fog server authenticate each other anonymously. But it does not include in a secure CO subscription/renewing mechanism to bill the users. Moreover, once the intermediate layer device is compromised at the key exchange phase, user's privacy is easily achieved.

To the best of our knowledge, until the time of writing this paper, the proposed scheme is the first that aims to directly solve the user/IoT-edge mutual anonymous authentication while supporting the CO charging.

TABLE I
NOTATIONS USED IN PROPOSED SCHEME

Symbol	Description
CA	centered authority
ID_i	identity of ith IoT device $Device_i$
EID_j	identity of ith edge device $Edge_j$
$DTPD_i$	TPD device of $Device_i$
$ETPD_j$	TPD device of $Edge_j$
$SC_{i,u}$	smart card distributed to user u
$SCID_{i,u}$	identity of TD_i
PID_i	initial pseudo identity of $Device_i$
$PID_{i,ts}$	instant virtual pseudo identity of $Device_i$ at ts
$PEID_j$	initial pseudo identity of $Edge_j$
ts	current timestamp
$pw_{i,u}$	biological password of driver u of $Device_i$
TS_{ms}	transaction sequence number for offloading
$RPID_i$	CO request identity
$SSID_i$	CO subscription/renewing identity
$CPID_i$	CO confirmation identity
$h(.)$	hash function h: $\{0,1\}^*\times V\rightarrow \mathbf{Z}_q^*$, $\mathbf{Z}_q^* = \{x\in\{1,\ldots,q-1\} \mid \gcd(x,q)=1\}$
$h_k^1(.)$	hash function h_k^1: $\{0,1\}^*\rightarrow\{0,1\}^n$
$H(.)$	hash function H: $\{0,1\}^*\rightarrow\mathbf{G}^*$, $\mathbf{G}^* = \mathbf{G}\backslash\{0\}$ [14]
$mac_k(.)$	MAC using k as a key, such as HMAC [13]
$Enc_k(.)$	encryption function using k as key, like AES [15]
$Dec_k(.)$	decryption function using k as key, like AES [15]
$\|\|$	Operation of message concatenation operation
?=	Operation of checking if equal

TABLE II
SYSTEM PARAMETERS

Symbol	Description
α	private key $\alpha\in Z_q^*$
β	public key $\beta=\alpha P$
$S_{ID_{CA}}$	identity secret key $S_{ID_{CA}} = \alpha H(ID_{CA})$
k_m	system key $k_m = \{k_m^1, k_m^2\}$, $k_m^1\in\{0,1\}^a$, $k_m^2\in\{0,1\}^b$
a	key length of $Enc_k(.)$
b	key length of $h_k^1(.)$

III. SYSTEM MODEL

We consider a three-layer hierarchy architecture for MEC. The notations used in proposed scheme and necessary system parameters are shown in Tables I and II, respectively.

In the cloud layer, a CA is employed by the cloud service provider. The CA is responsible for system initialization, user/IoT device registration, edge device registration, secure credential storage, and after-event tracing.

In the MEC layer, many edge devices reside and form a mobile Internet. Edge devices act as a security proxy of CA and are securely connected to the CA. User needs to subscribe for CO service before enjoying IoT services supported by edge devices. If the paid times are used up, he/she needs to renew the service. The edge devices will receive and deal with the subscription or renewing requests. When users try to use IoT service which needs to offload computing to edge devices, the corresponding edge devices will receive the CO task request, authenticate the request, fulfill the tasks, and charging the user by minoring available CO service times accordingly.

In the user/IoT layer, each user has his/her frequent used IoT devices (e.g., virtual reality helmet, distributed micro data center, surveillance home camera, etc.). To enable the IoT devices with a CO service support by edge devices, a user needs to register himself/herself along with his/her IoT devices to CA and acquire a personal smart card for future subscription/renewing. It is noted that, edge devices are dynamically selected by IoT devices based on geographical location, available resources, and historical task fulfillment statistics. They are transparent to users, of which the selection cannot be affected by the user.

We assure that communication connection between the Cloud layer and the MEC layer is wired and secure which means the adversary can hardly eavesdrop the messages. As for the connection between the MEC layer and for user/IoT layer, it is wireless and easy to be controlled by misbehaving users or adversaries.

IV. PROPOSED SCHEME

Our scheme employs mainly three methods.

Method 1 (n-Times Computing Offloading): We allow an IoT user to obtain the ubiquitous CO service from edge device, to a specific time-period (n-times) which depends on that the IoT user has paid. During subscription, the cloud generates a nonce, and generates and sends the initial hash chain value $h(n1)$ to the user via edge device. The cloud and the user both maintain a hash chain until renewing it. Every time when the user wants to obtain the CO service, it needs to compute

the latest hash-chain value and sends it to the cloud. The cloud will verify the hash-chain value and synchronize with the cloud database. If the hash chain is used up, the user will start a renewing process and get a new nonce $n1_{\text{new}}$ through a secure manner.

Method 2 (Anonymous Subscription/Renewing via Edge as Security Proxy): To preserve users' privacy information in CO subscription/renewing and CO task requesting, we introduce the two-factor authentication technique, where both IoT and edge devices employ tamper-proof devices to store system key, which we call it "km," and other supporting secret information. This supports a fast user verification, edge-aided CO subscription, and mutual authentication for CO. During user/IoT device and edge registration, the CA will distribute initial pseudo identities for user's smart card. Both IoT and edge devices (SCID, PID, and PEID), where the identities will be used to generate dynamic instant pseudo identities. During subscription, the user plugs his/her smart card into the IoT device and inputs the password (usually the fingerprint, iris image, etc.). Subsequently, the IoT device will verify the user's identity, generates instant presubscription identity preSID based on initial pseudo identities of user and device, public identity of edge device, and send it to the intended edge device. The intended device then verifies the preSID and generates a formal subscription identity SSID and sends to the CA. The CA completes the SSID verification using the prestored information from the device and edge registration. Then the CA generates and sends the initial hash chain value and maintain a hash chain.

Method 3 (Anonymous User/IoT Device-Edge Mutual Authentication for CO): We let the CO request composed in the message, at the cost of message signing and verifying, the edge user and edge server will accomplish the mutual authentication and service count processes simultaneously. The user verification process is the same as in the anonymous edge-aided subscription. After that, the IoT device will generate instant pseudo identity RPID for CO task request coREQ, compute the corresponding signature sigREQ and send tuple <RPID, sigREQ, TS_{ms}, coREQ> to the edge server. The edge server will verify the signature, generate the instant pseudo identity CPID for mutual confirming and corresponding signature sigCPID, and send tuple <CPID, sigCPID, TS, *ts*> back to the IoT device, where new transaction sequence number TSnew is securely stored in the TS. The IoT device will verify the sigCPID and update the transaction sequence number. To resist to potential DoS attack, the scheme uses a lightweight one-way hash function and MAC function to fulfill the signing/verifying the above processes. That is more lightweight than point multiplication, field exponentiation, or pairing operation.

V. Proposed LAMANCO Scheme

The proposed LAMANCO scheme has four phases. In Phase I, the certificated authority (CA) on cloud initializes the system and generates the necessary secure parameters. The user then, along with his/her IoT device, fulfill the registration to the CA with the edge devices also need to register to the CA. We call it Initialization and Registration Phase. Phase II is the CO Subscription/Renewing Phase, the user anonymously performs the subscribing or renewing for CO services to a specific period, *n*-times in our scheme. The edge device acts as a security proxy to securely communicate with the CA and keeps the credentials. Phase III is Anonymous Mutual Authentication Phase, in which user/both IoT and edge devices authenticate each other anonymously and then computing-intensive tasks can be offloaded to the edge devices. Phase IV is proposed for CA to conditionally trace marvelous messages after bad events in Phase II or Phase III, we call it the Conditional Tracing Phase.

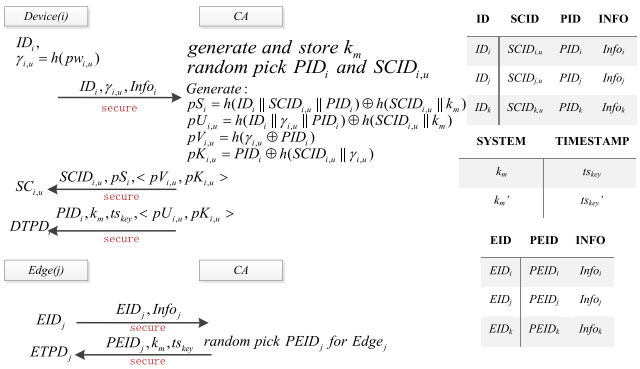


Fig. 2. Initialization and registration phase.

A. Initialization and Registration Phase

Suppose $\mathbf{G}$, a cyclic additive group of order q; $P \in \mathbf{G}$ a generator of $\mathbf{G}$; $e : \mathbf{G} \times \mathbf{G} \rightarrow V$ be a bilinear map which satisfies: bilinear, $e(x_1 + x_2, y) = e(x_1, y)e(x_2, y)$ and $e(x, y_1 + y_2) = e(x, y_1 + y_2)$; nondegenerate, $x \in \mathbf{G}$ and $y \in \mathbf{G}$ such that $e(x, y) \neq 1$ [11].

As shown in Fig. 2, initially the CA will generate public keys, private key, and a set of public parameters, then the CA publishes $\{\beta, \text{ID}_{\text{CA}}\}$, and keeps α, k_m, and $S_{\text{ID}_{\text{CA}}}$ secret. Both IoT and edge devices are free to register themselves to the CA.

As for the IoT device registration, Device_j, along with its user, submits its real identity ID_i, $\gamma_{i,u} = h(pw_{i,u})$ and Info_i (e.g., device serial number, owner, and subscribed service to the CA). Then CA randomly picks $\text{PID}_i \in \mathbf{Z}_q^*$ for Device_i, picks $\text{SCID}_{i,u}$ for user's smart card, and keeps record <ID_i, $\text{SCID}_{i,u}$, PID_i, Info_i>. The CA computes pS_i, pU_i, $pV_{i,u}$, $pK_{i,u}$ and writes necessary parameters to user's smart card or tamper proof device of IoT devices. As for the edge device registration, Device_jsubmits its real identity EID_j, and Info_j, and the CA will pick $\text{PEID}_j \in \mathbf{Z}_q^*$ for Edge_j, also keeps tuple <EID_j, PEID_j, Info_j>.

B. Computing Offloading Subscription/Renewing Phase

The subscription and renewing phase are illustrated in Fig. 3. When a user/IoT device is in edge covering area and needs to acquire CO services. He/she plugs a smart card into IoT device, input password and proceeds as follows.

$SC_i \rightarrow DTPD_i$:

1) Compute $\gamma_{i,u}^* = h(pw_{i,u})$and restore PID_i.

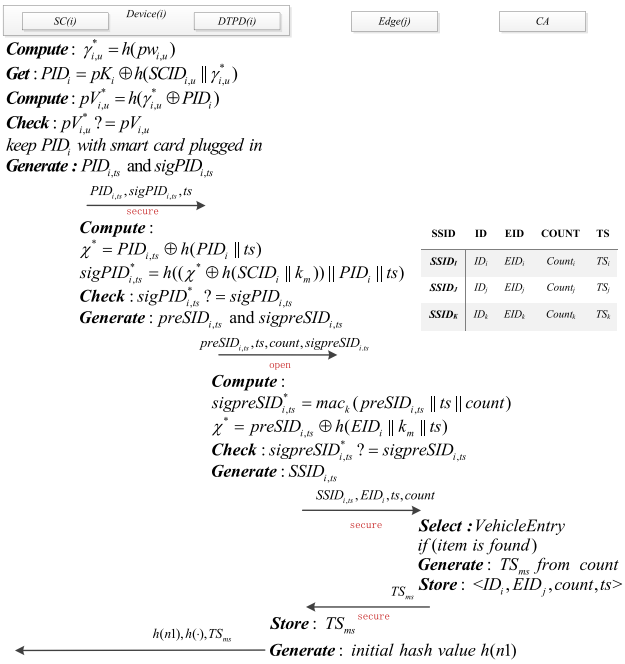


Fig. 3. Subscription/renewing phase.

2) Compute $pV^*_{i,u}$ and check $pV^*_{i,u} ? = pV_{i,u}$, if equal, the user verification is passed.
3) $\text{PID}_{i,ts} = h(\text{ID}_i||\text{SCID}_{i,u}||\text{PID}_i) \oplus h(\text{PID}_i||ts)$and its signature $\text{sigPID}_{i,ts} = h(pS_i||\text{PID}_i||ts)$ as login token, send tuple <$\text{PID}_{i,ts}$, $\text{sigPID}_{i,ts}$, ts> to DTPD_i.

$DTPD_i \rightarrow Edge_j$:

1) Compute $\text{sigPID}^*_{i,ts}$ and authenticate user by checking $\text{sigPID}^*_{i,ts} ? = \text{sigPID}_{i,ts}$.
2) Generate $\text{preSID}_{i,ts} = h(\text{ID}_i||\text{SCID}_{i,u}||\text{PID}_i) \oplus h(\text{EID}_i||k_m||ts)$ as instant presubscription identity and $\text{sigpreSID}_{i,ts} = \text{mac}_k(\text{preSID}_{i,ts}||ts||\text{count})$.
3) Send tuple <$\text{prePID}_{i,ts}$, ts, count> to Edge_i. Here, count is the intended subscription or renewing times.

$Edge_j \rightarrow CA$:

1) Compute $\text{SSID}_{i,ts}$ by replacing part $h(\text{EID}_j||k_m||ts)$ of $\text{prePID}_{i,ts}$ with $h(\text{EID}_j||\text{PEID}_j||\text{count}||ts)$.
2) Send tuple <$\text{SSID}_{i,ts}$, EID_j, ts, count> to CA.

$CA \rightarrow Edge_j$:

1) Select *VehicleEntry*< $\text{ID}_i, \text{SCID}_{i,u}, \text{PID}_i, \text{Info}_i$> join with <$\text{EID}_j, \text{PEID}_j, \text{Info}_j$> where $\text{SSID}_{i,ts} = h(\text{ID}_i||\text{SCID}_{i,u}||\text{PID}_i) \oplus h(\text{EID}_j||\text{PEID}_j||\text{count}||ts)$.
2) Generate TS_{ms} from count, where TS_{ms} is a sequence number, the value of it depends on the number of requests (T_m) held by smart card.
3) Send TS_{ms} to EDGE_j.

$Edge_j \rightarrow Device_i$:

1) Store a copy of TS_{ms} for further CO.
2) Generate initial hash value $h(n1)$.
3) Send tuple $< h(n1), h(\cdot), \text{TS}_{\text{ms}} >$ toDevice_i.

Both the user and the edge device need to maintain a hash chain $h^1(n1), h^2(n1), h^3(n1), h^{n+1}(n1)$. This hash chain satisfies that $h^1(n1) = h(n1)$ and $h^{i+1}(n1) = h(h^i(n1)), i = 1, 2, 3, \ldots, n$. When a user wants to acquire services of edge device offloading computing tasks, he/she needs to produce the lasted hash value and use it in an authentication request.

C. Anonymous Mutual Authentication Phase

In this phase, the user/IoT device achieves CO support from the edge device without communicating with CA. The anonymous mutual authentication phase is illustrated in Fig. 4. The scheme is performed as follows.

$SC_i \rightarrow DTPD_i$:

1) Perform user verification as subscription phase.
2) Compute $x = h^{n-k+1}(n1), k = k + 1$, where x is the hashed value and k is the location in hash chain.
3) Generate $\text{RPID}_{i,ts}$ and its signature sigRPID_i, then send tuple <$\text{RPID}_{i,ts}$, sigPID_i, TS_{ms}, x> to DTPD.

$DTPD_i \rightarrow Edge_j$:

1) Compute sigRPID^*_i and authenticate user by checking $\text{sigRPID}^*_i ? = \text{sigRPID}_i$.
2) Generate sigREQ_i based on message authentication code and hash technique as followings: $\text{sigREQ}_i = \text{mac}_{k_m}(\text{RPID}_{i,ts}||h(\text{coREQ}||k_m)||x||\text{TS}_{\text{ms}})$, where coREQ represents computing requesting request.
3) Send <$\text{RPID}_{i,ts}$, sigREQ_i, TS_{ms}, coREQ> to Edge_i.

$Edge_j \rightarrow DTPD_i$:

1) Compute $x^* = h^{n-k+1}(n1), k = k + 1$.
2) Compute sigREQ^*_i and authenticate Device_i by checking $\text{sigREQ}^*_i ? = \text{sigREQ}_i$. If Device_i passed authentication, the Edge_j will accept coREQ.
3) Increate $\text{TS}_m = \text{TS}_m + 1$, $\text{TS}_{\text{new}} = Tm$.
4) Compute $\theta_i = \text{sigREQ}^*_i \oplus h(\text{TS}_{\text{new}}||k_m)$.
5) Generate $\text{CPID}_{j,ts}$ as CO confirmation identity and TS as securely transmission form of TS_{new}.
6) Generate corresponding $\text{sigCPID}_{j,ts}$.
7) Send back <$\text{CPID}_{j,ts}$, sigCPID_j, TS, ts>.

$Device_i$:

1) Generate $\text{CPID}^*_{j,ts}$, TS^*_{new}, θ^*_i and sigCPID^*_j. Then Device_i will authenticates Edge_j by checking $\text{sigCPID}^*_j ? = \text{sigCPID}_j$.
2) Confirm coREQ when the check is passed.
3) Set $\text{TS}_{\text{ms}} = \text{TS}^*_{\text{new}}$ to make it in synchronization.

D. Conditional Tracing Phase

Once malicious behaviors are detected, the authority needs to collect enough evidence and trace accused user's clear identity. In the proposed scheme, the tuples between Device_i and Edge_j are most dangerous because they are usually exposed in wireless communication environments, and are easy to replay, forge, or modify. Moreover, the tuples between Edge_j and CA also suffer from threats that adversary might compromise the edge device and try to gain sensitive information from users. The tuples include <$\text{preSID}_{i,ts}$, ts, count> and <$\text{SSID}_{i,ts}$, EID_j, ts, count> in subscription/renewing phase. While the threatened tuples are <$\text{RPID}_{i,ts}$, sigREQ_i, TS_{ms}, coREQ> that Edge_j receive and <$\text{CPID}_{j,ts}$, sigCPID_j, TS, ts> that Edge_j sends in anonymous mutual authentication phase. To trace these deliberated malicious tuples, the CA first selects corresponding records in tables where the Device_i's registration information, Edge_j's registration information, and the CO subscription/renewing information are stored. Then the CA computes the key anonymous identity in the tuple and check it, if equal the clear

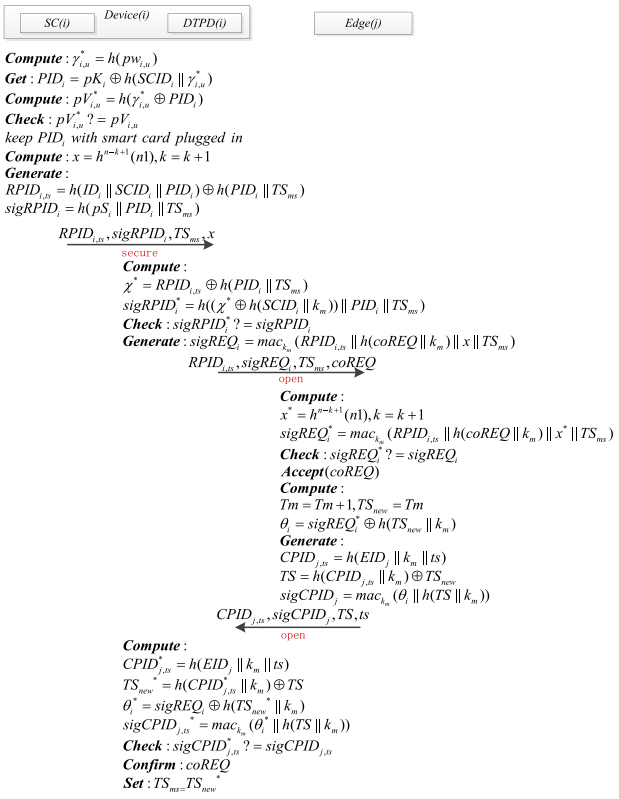


Fig. 4. Anonymous mutual authentication phase.

identity of Device$_i$ or Edge$_j$ can be traced. For example, if the CA wants to trace <RPID$_{i,ts}$, sigREQ$_i$, TS$_{\text{ms}}$, coREQ>, first it computes and checks sigREQ$_i$ to make sure the message is from valid user. After that, it will select <ID$_i$, SCID$_{i,u}$, PID$_i$, Info$_i$> from registration information table stored in the CA where RPID$_{i,ts}$ = h(ID$_i$||SCID$_i$||PID$_i$) $\oplus$ h(PID$_i$||TS$_{\text{ms}}$), then the user and the IoT devices are tracked.

VI. Security Analysis

In this section, we first give preliminaries about soundness of symbolic approach realized in automatous way, then we discuss the security analysis of the proposed LAMANCO and compare it with some existing schemes most closely similar to CO context.

A. Preliminaries

In the last two decades, researchers employed computational and symbolic approaches to analyze protocols. The computational approach is computationally sound because computational complexity and probability theory are applied such that the security of the protocols can be reduced to one of cryptographic hardness assumptions. However, it is hard to perform proof through programming and highly error prone, when researchers need to design and analyze moderately complex protocols with many security entities. Benefitting from work [16]–[18] that have solved the three problems in a symbolic approach: 1) unclear computational soundness; 2) fixed number of participants; and 3) exponentially increased time complexity along with the number of participants. We now can use the symbolic approach in cryptographic scheme analyzing, which is amenable enough to realize in automatous way based on a relevant tool [19].

TABLE III
Security Comparisons

Schemes / Security Properties	[5]	[6]	[9]	[10]	LAMANCO
Message Integrity	-	-	-	-	+
User Anonymity	+	-	-	+	+
Conditional Traceability	-	-	-	+	+
Unlinkability	-	-	-	+	+
Secure n-times CO	-	-	-	-	+
Resilience to Edge Compromise	No Edge	No Edge	-	-	+
Resilience to DoS	-	-	-	+	+
Three Hierarchy Architecture Support	-	-	+	+	+
Cloud Offline Authentication	-	-	+	+	+

B. Experiments and Analysis

The comparison results between LAMANCO and [5], [6], [9], and [10] are demonstrated in Table III.

1) Message Integrity: Messages between the user/IoT and the edge device are protected by the MAC. Once the message is forged or modified, both kinds of entities are able to detect them assisted by a tamper proof device. Results show that if "event endAuthCO (RPID_*i_ts*, sigREQ_*i*, TS_ms, coREQ)" has been executed, then "event beginAuthCO(RPID_*i_ts*, sigREQ_*i*, TS_ms, coREQ)" must have been executed. Thus, the proposed scheme is able to protect the integrity of messages <RPID$_{i,ts}$, sigREQ$_i$, TS$_{\text{ms}}$, coREQ> and <CPID$_{j,ts}$, sigCPID$_j$, TS, *ts*>.

2) User Anonymity: The utilization of instant pseudo identity RPID$_{i,ts}$ preserves the clear identity of user/IoT device from leaking. Moreover, even only through the tamper proof device on edge device, thus user's anonymity is preserved even if the smart card is stolen. As shown in results of ProVerif, the adversary is unable to obtain any information about ID$_i$ because "querying not attacker" result of ID$_i$ returns true.

3) Conditional Traceability: k_m is the important parameter to disclose user/IoT device's clear identity. Due to that k_m is securley stored in tamper proof device and not utilized directly in open communication, only CA can perform the conditional tracking. We queried the value of k_m and the results show that adversary cannot achieve useful information of k_m.

4) Unlinkability: In LAMANCO, instant pseudo identity changes as time passes, an adversary can never achieve the linking between numerous messages and one user/IoT device. We use keyword "choice[RPID_*i_ts*, *r*0]" to test the anonymity. The result of "RESULT Observational equivalence is true (bad not derivable)" shows that $RPID_{i,ts}$ cannot be distinguished from an *r0*. We also use keyword "! processes" to test the unlinkability. The "true" result means no sensitive information about user/IoT device can be acquired,

even in multiple processes running(adversary collects multiple messages).

5) Secure n-Times Offloading: N-times offloading is the key to charge user for consume CO services and easily attract adversary's eyes. An adversary or a misbehaving user will try to cheat for more times of CO service without paying legitimately. In the proposed LAMANCO, we use anonymous subscription identity $\text{SSID}_{i,ts}$ to represent a valid subscription request, which is generated on one-way hash function from ID_i,$\text{SCID}_{i,u}$, PID_i, EID_j, PEID_j, count, and ts. The CA could detect replay attacks by checking ts. An adversary or a misbehaving user will fail to cheat for subscription because they can never pass the verification of Edge_j.

6) Resilience to Edge Compromise: When an adversary compromises or corrupts an edge device Edge_j, then he knows a clear identity of Edge_j, but never gets more secret information, such as PEID_j or k_m due to that these parameters are stored in tamper-proof device. In addition, the adversary cannot get clear identities of user because the adversary can never achieves PID_i or k_m from the messages he receives from user/IoT device too, although he could collect subscription/renewing or CO requests through a compromised edge device. The $\text{prePID}_{i,ts}$ or $\text{RPID}_{i,ts}$ are generated by applying a one-way hash function to anonymous identity and system key, which are computationally hard to link to one security entity.

7) Resilience to Denial of Service: User/IoT and edge devices need no exponential computation, paring computation, or public key computation, but only a few lightweight hash functions and message authentication code functions. Moreover, the edge device acts like a security proxy which is able to reject invalid subscription requests or CO service requests early in interactions, which prevents the CA from facing requests directly. Hence, the proposed scheme is computationally efficient and resilient to DoS.

8) Three Layer Hierarchy Architecture Support: LAMANCO, along with [9] and [10] are able to support three-layer hierarchy architecture. However, only LAMANCO considers the edge device, other than fog server, as an intermediate layer.

9) Cloud Offline Authentication Support: After the initialization and registration, all schemes including [9] and [10] and LAMANCO use an intermediate layer as a security proxy, which is able to support authentication with an offline cloud server.

VII. Performance Analysis

In the performance analysis, we consider a scenario in which wearable devices (e.g., smartwatch and smart band) act as IoT devices and smart phones or wireless routers act as edge devices. Our scheme uses only hash and MAC, two simple cryptographic primitives, to build the core phases. This makes the scheme very lightweight. As for $h_k^1(.)$, we select SHA-1 from many hash functions like SHA-1, SHA-2, SHA-224, and SHA-256. SHA-1 that takes an arbitrary length message in blocks of 512 bits as input and produces an SHA-1 output of 160 bits, which will be reinvoked as input with the next 512-bit block. As for $\text{mac}_k(.)$, we use HMAC. The performance analysis adopts Tate pairing [33], in which $\mathbf{G}$ is 161 bits and order q is 160 bits. Moreover, we also utilize AES-128 as $\text{Enc}_k(.)$ and $\text{Dec}_k(.)$. Hence, the smartphones or wireless routers are already strong in computing power. We calculated the time cost benchmark for different operations in protocol on a computer with Intel Core 2 Duo CPU@2.4 GHz. To achieve more precise results, we ran 1000 times of computation for each kind of operation. Suppose T_{mul}, T_{exp}, T_{par}, T_{mod}, T_h, T_{mac}, T_{enc}, and T_{dec} denote time cost for one-point multiplication, one field exponentiation, one pairing operation, one hash function operation, one MAC operation, and one encryption operation, which are 5.4 ms, 4.9 ms, 40.7 ms, 11.7 ms, 6.0 us, 16.7 us, 91.9 us, and 107.4 us, respectively. We assure that SCID_i, ID_i, and EID_i are all of size 3 bytes, which are enough to support more than 16 million entities for IoT devices, users, or edge devices.

A. Memory Requirements

We analyze the memory requirements as follows. For the user, he/she needs to store $<\text{SCID}_{i,u}, pS_i, pV_{i,u}, pK_{i,u}>$, in which is the length of $\text{SCID}_{i,u}$ is 3 bytes, while $pS_i, pV_{i,u}$, and $pK_{i,u}$ are all short hashed values of a string. Therefore, the memory of a smart card is enough. For the IoT device, it needs to store $<\text{PID}_i, k_m, ts_{\text{key}}, pU_{i,u}, pK_{i,u}>$ in tamper proof device, in which $\text{PID}_{i,u}$ is 3 bytes, the system key k_m is of fixed length, ts_{key} is only 4 bytes, $pU_{i,u}$ and $pK_{i,u}$ are hashed value of strings. For the edge device, it also needs to store $<\text{EPID}_j, k_m, ts_{\text{key}}>$ in tamper proof device, the memory consumed is less than the IoT device. For the CA, the CA needs to store all versions of the system key, the user/IoT and edge device registration information and CO subscription information. All versions of system keys k_m are stored for the CA to conditionally trace misbehaving user/IoT device or edge device. Only for the maintaining period or the current system key is leaked, the CA needs to update the system key, thus the storage consuming is affordable. The number of registration information records are linear with the number of user/IoT devices or edge devices, and only used for conditional tracing, which consumes small memory and situation is the same as subscription information.

B. Computation Complexity

The comparisons of our scheme along with other two close scheme are shown in Table IV.

By analyzing our scheme, the user/IoT device performs no computation in Initialization and Registration Phase, we only need to submit necessary information, and receive security parameters such as pseudo identity, system key, and others. While even for the CA, apart from generating initial cryptographic parameters, it only needs six hash invocations operations for each user/IoT device registration.

To the best of our knowledge, our scheme is the only one considering CO Subscription/Renewing Phase. In this phase, a user/IoT device needs ten hash invocation and 1 MAC operation to fulfill the CO precharging. In order to make sure the request is legitimate, the edge device only needs two hash

TABLE IV
COMPUTATION COMPARISONS BETWEEN THE SCHEMES

Schemes	Maged's solution		Arij's solution		LAMANCO		
Phases	Initialization &Registration	Authentication	Initialization &Registration	Authentication	Initialization &Registration	Subscription Or Renewing	Authentication
User/IoT or Fog User		1 hash 1 sym enc. 1 sym dec.	1 sym enc. 1 sym dec.			10 hash 1 mac	15 hash 2 mac
Edge Device or Fog Server	2 sign verif. 1 asym dec.	1 sym enc. 1 sym dec.	1 sign verif. 1 asym dec.			2 hash 1 mac	6 hash 2 mac
Centered Authority	2 sign gen. 2 hash 1 asym enc.		1 asym dec. 1 sign gen. 1 asym enc.		6 hash	2 N hash	

*Note: N is the average query times from user/IoT registration table

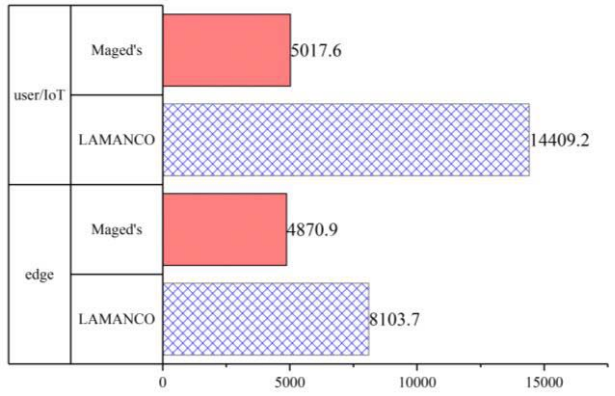


Fig. 5. Comparison of number of requests processed in 1 s.

invocations and 1 mac operation to authenticate the request and deliver it to the CA, while the CA needs $2N$ hash invocation operations, where N is the average query times to find user/IoT and edge devices' identities. Hence, the invalid subscription/renewing requests are rejected by the edge device and that the CA has nearly unlimited computation resources, the computation consumption is affordable.

In the Anonymous Mutual Authentication Phase, the CA needs not to participate. The user/IoT device needs 15 hash invocations and 2 macs, whereas edge device needs 8 hash and mac. Both the hash invocation and mac are lightweight enough to make the phase efficient.

C. Computation Time

We calculated the computation time based on the number of basic operations and the time cost for each kind of basic operation, and we compared the computation time between schemes. When the edge device needs to authenticate user/IoT device in LAMANCO, it takes the user/IoT device 82.7 us (11 T_h and 1 T_{mac}) to generate and sign the service request, and it takes the edge device 28.7 us (2 T_h and 1 T_{mac}) to verify it. When the user/IoT device needs to authenticate mutually, it takes both edge device and user/IoT device 40.7 us (4 T_h and 1 T_{mac}) more.

D. Comparison With Closely Related Work

Although, up to the time this piece of work is proposed, there is no contribution which has completely the same user/IoT-edge-cloud CO architecture. We discuss closely related work in [9] and [10]. In these two schemes, the authentication phases are both constructed on symmetric or asymmetric cryptography and have heavy computation complexity. For example, in Maged's scheme [9], it takes a fog user 205.3 us (1T_{enc}, 1 T_{dec}, and 1 T_h) and it takes a fog server 199.3 us (1 T_{enc} and 1 T_{dec}) to complete the mutual authentication, which are 1.66× of user/IoT device's consumption and 2.87× of edge device's consumption respectably compared with LAMANCO. As shown in Fig. 5, LAMANCO is much more lightweight in the number of processed requests in 1 s. Although in [10], Amor *et al.* claimed that confidential communication is guaranteed and no computation is needed in the authentication phase, it did not take message integrity nor edge compromise resilience into account. This can hardly be suitable for the CO subscription/renewing nor task request interactions between user/IoT and edge devices. Other schemes [20]–[22] are based on public key cryptosystems which employ identity-based cryptography, group signature, and so on. In these schemes, quantities of point multiplication, field exponentiation, and bilinear pairing operations will increase the computation complexity dramatically, which are apparently not fit for smart cards nor edge devices.

VIII. CONCLUSION

CO services are offered to support computing-intensive and delay-sensitive IoT service where it is hard to support anonymous mutual authentication between large quantities of IoT devices and edge devices. We proposed a lightweight anonymous mutual authentication scheme for n-times CO in IoT. Our scheme does not need a CA when the user/IoT devices are acquiring CO services from edge devices. A user subscribes or renews n-times CO service before daily use through a smartcard as token and an edge device as a security proxy. Moreover, user/IoT and edge devices can mutually authenticate each other in a CO requesting phase in case the messages are replayed or forged. On the other hand, our scheme uses tamper proof device to store pseudo identity, which protects user's sensitive information even with some edge devices being comprised and suffer from brute force. At last the scheme is resilient to the DoS attack due to the implementation of the authentication phase based on only the lightweight one-way hash function and MAC function.

In order to analyze the security features of our protocol, we use automated verifying tool ProVerif under symbolic approach. Our results strongly suggest that the proposed scheme is robust, anonymous, and adaptable in CO applications. To evaluate the performance, we calculate benchmarks of the proposed scheme and analyze the results. Our results strongly suggest that LAMANCO has 1/1.66 computation complexity in user/IoT device and 1/2.87 computation complexity in edge device. In conclusion, our framework provides insights into how user/IoT device's sensitive information can be simultaneously protected, while also keeping low computing consumption in mutual authentication. The novel scheme should be of considerable practical use to IoT manufactures that aim to provide user with fluent experience and improve the security.

Proposing an anonymous mutual authentication scheme for CO which targets directly to user/IoT-edge-cloud architecture is a new topic. Considering the resource consuming differences in CO tasks, we observe another two problems: 1) how to authenticate requests without leaking habits and 2) how to subscribe/renew and charge tasks accordingly in a secure manner. In the future, we plan to focus on these problems and improve this paper to a more advanced version.

REFERENCES

[1] N. Abbbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: A survey," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 450–465, Feb. 2018.

[2] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016.

[3] J. Ren, H. Guo, C. Xu, and Y. Zhang, "Serving at the edge: A scalable IoT architecture based on transparent computing," *IEEE Netw.*, vol. 31, no. 5, pp. 96–105, Aug. 2017.

[4] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computing: A key technology towards 5G," Sophia Antipolis, France, ETSI, White Paper, 2015.

[5] I. A. Gomaa, E. A. Elrahman, and M. Abid, "Virtual identity approaches evaluation for anonymous communication in cloud environments," *Int. J. Adv. Comput. Sci. Appl.*, vol. 7, no. 2, pp. 267–276, 2016.

[6] T.-H. Chen, H.-L. Yeh, and W.-K. Shih, "An advanced ECC dynamic ID-based remote mutual authentication scheme for cloud computing," in *Proc. IEEE 5th FIRA Int. Conf. Multimedia Ubiquitous Eng.*, 2011, pp. 155–159.

[7] L. Zhu, M. Li, Z. Zhang, and Z. Qin, "ASAP: An anonymous smart-parking and payment scheme in vehicular networks," *IEEE Trans. Depend. Secure Comput.*, to be published, doi: 10.1109/TDSC.2018.2850780.

[8] P. Gope and A. K. Das, "Robust anonymous mutual authentication scheme for *n*-times ubiquitous mobile cloud computing services," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1764–1772, Oct. 2017.

[9] M. H. Ibrahim, "Octopus: An edge-fog mutual authentication scheme," *Int. J. Netw. Security*, vol. 18, no. 6, pp. 1089–1101, 2018.

[10] A. B. Amor, M. Abid, and A. Meddeb, "A privacy preserving authentication scheme in an edge-fog environment," in *Proc. IEEE/ACS 14th Int. Conf. Comput. Syst. Appl.*, 2017, pp. 1225–1230.

[11] F. Wang, Y. Xu, H. Zhang, Y. Zhang, and L. Zhu, "2FLIP: A two-factor lightweight privacy-preserving authentication scheme for VANET," *IEEE Trans. Veh. Technol.*, vol. 65, no. 2, pp. 896–911, Feb. 2016.

[12] L. Zhu *et al.*, "PRIF: A privacy-preserving interest-based forwarding scheme for social Internet of Vehicles," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2457–2466, Aug. 2018.

[13] M. Bellare, R. Canetti, and H. Krawczyk, "Message authentication using hash functions: The HMAC construction," *RSA Lab. CryptoBytes*, vol. 2, no. 1, pp. 12–15, 1996.

[14] F. Hess, "Efficient identity based signature schemes based on pairings," in *Proc. Selected Areas Cryptography*, 2003, pp. 310–324.

[15] J. Daemen and V. Rijmen, "AES proposal: Rijndael," in *Proc. 1st Adv. Encryption Stand. Candidate Conf. Nat. Inst. Stand. Technol. (NIST)*, pp. 1–37, Nov. 1998.

[16] R. Canetti and S. Gajek. *Universally Composable Symbolic Analysis of Diffie–Hellman Based Key Exchange*. Accessed: May 2010. [Online]. Available: http://eprint.iacr.org/2010/303.pdf

[17] R. Canetti and J. Herzog, "Universally composable symbolic analysis of mutual authentication and key-exchange protocols," in *Proc. Theory Cryptography Conf.*, 2006, pp. 380–403.

[18] Z. Zhang, L. Zhu, L. Liao, and M. Wang, "Computationally sound symbolic security reduction analysis of the group key exchange protocols using bilinear pairings," *Inf. Sci.*, vol. 276, no. 20, pp. 93–112, 2012.

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

[20] 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.

[21] Y. Qiu, J. Zhou, J. Baek, and J. Lopez, "Authentication and key establishment in dynamic wireless sensor networks," *Sensors*, vol. 10, no. 4, pp. 3718–3731, 2010.

[22] Q.-Q. Xie, S.-R. Jiang, L.-M. Wang, and C.-C. Chang, "Composable secure roaming authentication protocol for cloud-assisted body sensor networks," *Int. J. Netw. Security*, vol. 18, no. 5, pp. 816–831, 2016.

[23] C. Xu, J. Ren, D. Zhang, and Y. Zhang, "Distilling at the edge: A local differential privacy obfuscation framework for IoT data analytics," *IEEE Commun. Mag.*, vol. 56, no. 8, pp. 20–25, Aug. 2018.

[24] C. Xu, J. Ren, Y. Zhang, Q. Zhan, and K. Ren, "DPPro: Differentially private high-dimensional data release via random projection," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 12, pp. 3081–3093, Dec. 2017.

[25] Z. Guan *et al.*, "Achieving efficient and secure data acquisition for cloud-supported Internet of Things in smart grid," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 1934–1944, Dec. 2017.

[26] D. Zhang, R. Shen, J. Ren, and Y. Zhang, "Delay-optimal proactive service framework for block-stream as a service," *IEEE Wireless Commun. Lett.*, vol. 7, no. 4, pp. 598–601, Aug. 2018, doi: 10.1109/LWC.2018.2799935.

[27] D. Zhang *et al.*, "Two time-scale resource management for green Internet of Things networks," *IEEE Internet Things J.*, to be published, doi: 10.1109/JIOT.2018.2842766.

[28] X. Peng *et al.*, "BOAT: A block-streaming app execution scheme for lightweight IoT devices," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 1816–1829, Jun. 2018.

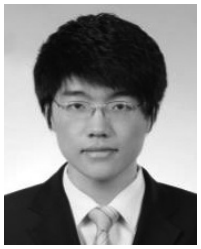
[29] D. Zhang *et al.*, "Utility-optimal resource management and allocation algorithm for energy harvesting cognitive radio sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3552–3565, Dec. 2016, doi: 10.1109/JSAC.2016.2611960.

[30] Y. Cheng, X. Fu, X. Du, B. Luo, and M. Guizani, "A lightweight live memory forensic approach based on hardware virtualization," *Inf. Sci.*, vol. 379, pp. 23–41, Feb. 2017.

[31] X. Du, M. Guizani, Y. Xiao, and H.-H. Chen, "Transactions papers a routing-driven elliptic curve cryptography based key management scheme for heterogeneous sensor networks," *IEEE Trans. Wireless Commun.*, vol. 8, no. 3, pp. 1223–1229, Mar. 2009.

[32] X. Du and H.-H. Chen, "Security in wireless sensor networks," *IEEE Wireless Commun. Mag.*, vol. 15, no. 4, pp. 60–66, Aug. 2008.

[33] X. Du, Y. Xiao, M. Guizani, and H.-H. Chen, "An effective key management scheme for heterogeneous sensor networks," *Ad Hoc Netw.*, vol. 5, no. 1, pp. 24–34, Jan. 2007.



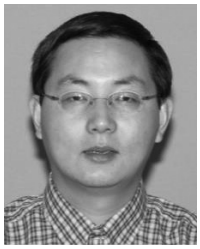
Fei Wang (M'12) was born in Shandong, China, in 1988. He received the B.S. degree in computer science from the Beijing Institute of Technology, Beijing, China, and the graduation degree and the Ph.D. degree in computer architecture from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, in 2011 and 2017, respectively.

From 2011 to 2017, he was a Research Assistant with the Institute of Computing Technology, Chinese Academy of Sciences, where he has been an Assistant Professor since 2017. He has authored over ten articles and holds two patents. His current research interests include privacy and security protection in Internet of Things, cyber-physical system architecture, and multisensor data fusion.

Yongjun Xu (M'06) received the B.Eng. degree in computer communication from the Xi'an Institute of Posts and Telecommunications, Xi'an, China, in 2001 and the graduation degree and the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2006.

He became an Associate Professor with the Institute of Computing Technology, Chinese Academy of Sciences in 2008, where he is an Associate Professor. His current research interests include cyber-physical systems and multisensor data fusion.

Liehuang Zhu (M'11) is currently a Professor with the Department of Computer Science, Beijing Institute of Technology, Beijing, China. He has been selected into the Program for New Century Excellent Talents of the University from the Ministry of Education, Beijing. His current research interests include Internet of Things, cloud computing security, and Internet and mobile security.



Xiaojiang Du (M'04–SM'09) is currently a Tenured Professor with the Department of Computer and Information Sciences, Temple University, Philadelphia, PA, USA. His current research interests include wireless communications, wireless networks, security, and systems. He has authored over 260 journal and conference papers and 1 book (Springer).

Prof. Du was the recipient of over $5 million in research grants from the U.S. National Science Foundation, the U.S. Army Research Office, the U.S. Air Force Research Laboratory, NASA, the State of Pennsylvania, and Amazon. He serves on the Editorial Boards of three international journals. He is a Life Member of the ACM.

Mohsen Guizani (S'85–M'89–SM'99–F'09) received the B.S. (with Distinction) and M.S. degrees in electrical engineering and the M.S. and Ph.D. degrees in computer engineering from Syracuse University, Syracuse, NY, USA, in 1984, 1986, 1987, and 1990, respectively.

He served as the Associate Vice President of graduate studies with Qatar University, Doha, Qatar, and the Chair of the Computer Science Department, Western Michigan University, Kalamazoo, MI, USA, and the Computer Science Department, University of West Florida, Pensacola, FL, USA. He is currently a Professor and the Electrical and Computer Engineering Department Chair with the University of Idaho, Moscow, ID, USA. He also served in academic positions with the University of Missouri–Kansas City, Kansas City, MO, USA, the University of Colorado–Boulder, Boulder, CO, USA, and Syracuse University. He has authored 9 books and over 500 publications in refereed journals and conferences. He has guest edited a number of special issues in IEEE journals and magazines. His current research interests include wireless communications and mobile computing, computer networks, mobile cloud computing, security, and smart grid.

Dr. Guizani was a recipient of three teaching awards and four research awards throughput his career and the 2017 IEEE Communications Society Recognition Award for his contribution to outstanding research in wireless communications. He is currently the Editor-in-Chief of *IEEE Network Magazine*, serves on the Editorial Boards of several international technical journals and is the founder and the Editor-in-Chief of *Wireless Communications and Mobile Computing* (Wiley). He also served as a member, the Chair, and the General Chair of a number of international conferences. He was the Chair of the IEEE Communications Society Wireless Technical Committee and the TAOS Technical Committee. He served as the IEEE Computer Society Distinguished Speaker from 2003 to 2005. He is a Senior Member of the ACM.