An Efficient Privacy Preserving Message Authentication Scheme for Internet-of-Things

Jiannan Wei, *Member, IEEE*, Tran Viet Xuan Phuong, and Guomin Yang, *Senior Member, IEEE*

***Abstract*—As an essential element of the next generation Internet, Internet of Things (IoT) has been undergoing an extensive development in recent years. In addition to the enhancement of people's daily lives, IoT devices also generate/gather a massive amount of data that could be utilized by machine learning and big data analytics for different applications. Due to the machine-to-machine communication nature of IoT, data security and privacy are crucial issues that must be addressed to prevent different cyber attacks (e.g., impersonation and data pollution/poisoning attacks). Nevertheless, due to the constrained computation power and the diversity of IoT devices, it is a challenging problem to develop lightweight and versatile IoT security solutions. In this article, we propose an efficient, secure, and privacy-preserving message authentication scheme for IoT. Our scheme supports IoT devices with different cryptographic configurations and allows offline/online computation, making it more versatile and efficient than the previous solutions.**

***Index Terms*—Hop-by-hop authentication, integrity, Internet of Things (IoT), source privacy.**

I. Introduction

THE Internet of Things (IoT) provides a self-establishing network of highly coupled heterogeneous objects, such as smart devices, radio frequency identification (RFID) tags, sensors. It allows devices to simplify the retrieval as well as the exchange of data without human involvement in various applications [1] and has a considerable position in the growth of information technology after the computer science and the Internet. IoT brings a pervasive digital appearance by engaging society and industries, and enables a series of interactions between human to human, human to thing, and more importantly, thing to thing. The development of IoT has led to enormous applications, such as smart home systems [2], intelligent transportation systems [3],[4], machine learning and big data [5].

The machine-to-machine (M2M) [6] communication among massive numbers of IoT devices will dominate future communication network traffic. The integrity and authenticity of the massive amount of data collected and transmitted by the IoT devices are crucial in some applications, such as machine learning and big data analytics. Maliciously injected or modified data can cause biased or wrong decision-making and prediction. Therefore, in order to ensure the correctness and accuracy of machine learning and big data analysis, the integrity and authenticity of the collected data must be retained [7].

There are two approaches to achieve secure message delivery in IoT—the symmetric-key-based approach and the public-key-based approach. The symmetric-key approach incurs less computation overhead compared with the public-key approach since symmetric-key operations are much more efficient than their public-key counterparts. However, key management is a major issue for symmetric-key-based approach in a large scale heterogeneous IoT network. Also, if the message is only authenticated using a shared key between the sender and the receiver, the intermediate forwarding nodes in the IoT network cannot verify the integrity of the message. If the message has been altered or damaged during transmission, then the problem can only be discovered by the receiver. On the other hand, public-key-based approach can solve these problems since anyone can use the public key to verify the integrity and authenticity of a message. However, public-key operations are very computation intensive, and privacy is another concern for public-key-based approach since the authentication token is publicly verifiable using the sender's public key. It is worth noting that the privacy of a data source is also important in some situations, e.g., when a wearable device is attached to a human. If the attacker can identify the sources of the data streams, then they could also cutoff a data stream (e.g., via a denial-of-service attack) and eventually affect the accuracy of the decision or prediction produced by machine learning.

In order to address the abovementioned problems in IoT and M2M communications, a secure, efficient, and privacy-preserving message authentication scheme that can support hop-by-hop verification is desirable. Li *et al.* [8] proposed a novel source anonymous message authentication (SAMA) scheme which could be used for such a purpose. Their scheme was believed to achieve message authentication and message source privacy with a lower cost than the previous approaches.

Manuscript received November 5, 2019; revised January 8, 2020; accepted January 27, 2020. Date of publication February 10, 2020; date of current version October 23, 2020. This work was supported by National Natural Science Foundation of China under Grant 61702268. Paper no. TII-19-4882. *(Corresponding author: Jiannan Wei.)*

Jiannan Wei is with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: jnwei@njust.edu.cn).

Tran Viet Xuan Phuong and Guomin Yang are with the School of Computing and Information Technology, University of Wollongong, Wollongong, NSW 2522, Australia (e-mail: txuan@uow.edu.au; gyang@uow.edu.au).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TII.2020.2972623

A. Motivation and Our Contribution

In this work, we first review Li *et al.*'s [8] SAMA scheme and point out a security problem in their scheme. In [8], a rigorous security analysis has been provided to prove that no attacker is able to forge a valid signature for a new message (e.g., a message injected by the attacker). However, we find that the integrity of the signing group (named Ambiguous Set in [8]) has been neglected in the design of the scheme. Specifically, we show in this article that based on a valid message-signature pair for a particular signer group, an attacker without knowing any secret key can change the signer group and produce a new valid signature for the modified group. According to the first design goal of Li *et al.*'s [8] SAMA scheme, which says "the message receiver should be able to verify whether a received message is sent by the node that is claimed, or by a node in *a particular group*," such an attack should be prevented. Our first contribution is to fix the security issue in Li *et al.*'s [8] SAMA scheme without introducing additional computation or communication cost.

The SAMA scheme proposed in [8] considers that all the sensor nodes in the network use the same cryptographic system (i.e., the modified ElGamal signature scheme [9], [10]). However, such an assumption may not hold in large scale IoT networks where different sensor nodes or smart devices may use different security systems or parameters. For example, some devices may choose to use the ElGamal-type system, but others may prefer the RSA system [11]. The SAMA scheme proposed in [8] cannot handle such a situation.

Motivated by the above consideration, in this article, we propose a new SAMA scheme with better practicality compared with [8]. Our scheme allows the devices in an IoT network to use different (more precisely, RSA-type and ElGamal-type) systems. Such an approach makes the solution more versatile and provides a stronger guarantee on the identity and location privacy of a data source since it can hide the type of security system used by a device.

Moreover, considering the low computation power of the IoT devices, we also apply the offline/online paradigm in the design of our system. Efficiency is extremely important in practical IoT scenarios, such as industrial automation, environmental monitoring, smart grids. In our scheme, a smart device can perform some expensive public-key operations offline (e.g., when it is idle), and only does the online computation when the message to be sent is ready. Interestingly, we find that by allowing both RSA- and ElGamal-type systems in our scheme, we are able to reduce the computation cost compared with the pure ElGamal scheme proposed in [8]. This may look counterintuitive since it is known that the ElGamal system [implemented using elliptic curve cryptography (ECC)] is much faster than the RSA system. The reason of this counterintuitive fact is that in our hybrid scheme, for most of the RSA nodes, we only need to do RSA signature verification, which is very fast since the RSA public exponent e can be very small. The proposed new SAMA scheme is compared with the previous scheme in terms of its execution time during signature generation and verification. We also implement our scheme in a laptop and in a Raspberry Pi to demonstrate its practicality.

B. Related Work

In order to prevent various types of attacks in data transmission, both symmetric-key and public-key approaches have been proposed in the literature. In [12], two different message authentication protocols were proposed. The first protocol, named TESLA, is based on message authentication code (MAC), and the design utilizes a one-way key chain and timed release of keys by the sender. However, the TESLA protocol requires synchronization among devices, which is difficult to implement in a large scale network. The second protocol in [12], named EMSS, is based on cryptographic hash function and public-key technique, and can achieve the security property of nonrepudiation. In [13], an interleaved hop-by-hop authentication scheme was proposed to prevent the injected false data packet attack by attackers or compromised nodes in the network. Their scheme is symmetric-key-based, and the basic idea is that multiple sensor nodes have to endorse a message (or report) using MACs in order to achieve message authentication. A similar approach was also proposed in an independent work by Ye *et al.* [14]. In [15], a polynomial-based approach was proposed to achieve lightweight and compromise-resilient message authentication, where messages are authenticated and verified via evaluating polynomials. Li *et al.* [8] proposed a ring signature [16] based solution to achieve message authentication. Their scheme utilizes a ring signature scheme derived from the modified ElGamal signature scheme [10], and can achieve better features and performance in several aspects compared with the previous solutions. However, as we will demonstrate later, the ring signature scheme proposed in [8] has a security flaw—it allows an attacker to arbitrarily form a ring and forge a valid ring signature from an existing one. Such an attack has been considered in the literature of ring signature (e.g., [17]) and in this work we introduce a technique similar to that of [17] to fix the flaw without introducing any computation or communication overhead.

There are also a number of research works on privacy-preserving user authentication (and key agreement) protocols for IoT and wireless sensor networks (WSNs) in recent years (e.g., [18]–[26]). These works focus on remote user authentication, which is related but different from the privacy preserving hop-by-hop message authentication considered in this article. Moreover, due to the concerns on the physical security of sensor nodes and IoT devices, the research on constructing lightweight and physically secure authentication protocols for IoT and WSNs has also become a popular topic in recent years. To ensure physical layer security, physically unclonable functions and wireless channel characteristics (such as the link quality indicator) are popular choices to enable security even if a sensor node is captured by an adversary.

Several novel lightweight authentication protocols with physical security for IoT and WSNs can be found in [27]–[29].

C. Article Organization

For the rest of the article, in Section II, we present the system and threat models for cloud-based IoT. We review and fix a security issue in Li *et al.*'s [8] SAMA scheme in Section III. In

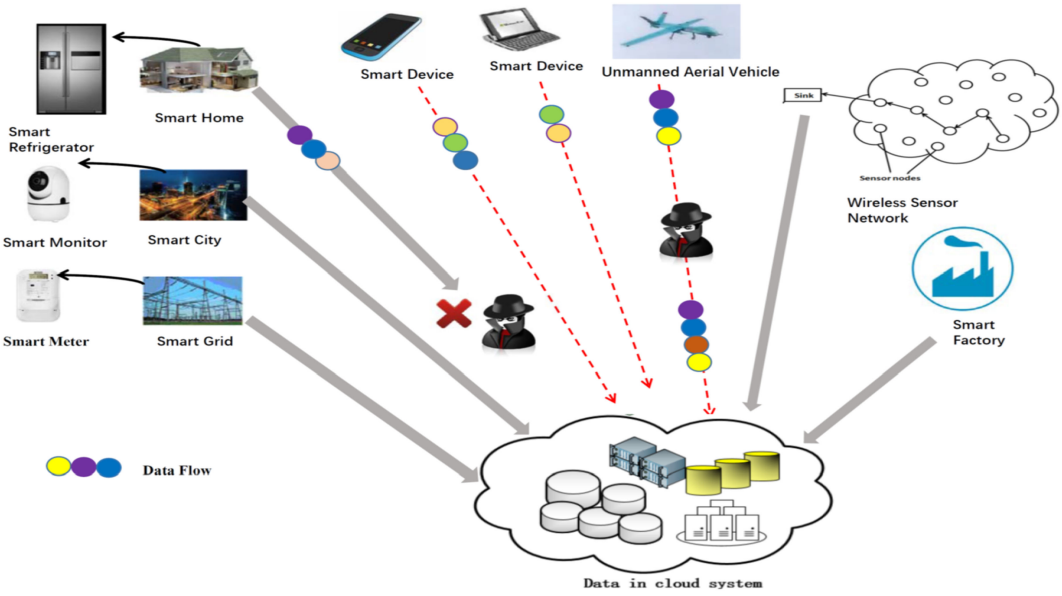

Fig. 1. Cloud-centric IoT network architecture.

Section IV, we present a new SAMA scheme and analyze its security and efficiency. Section V concludes this article.

II. Network and Threat Models

A. Network Model

We consider the cloud-centric network model for the proposed SAMA scheme for IoT data authenticity in this article. The network architecture is shown in Fig. 1. Specifically, we consider an IoT network where smart devices may be deployed by different entities for collecting various types of data and hence may use different system parameters. We also assume that all the system parameters, certificates, and keys have been installed in each node before it is deployed. After the deployment, terminals will collect data and send the data to the cloud or data center for analysis. If a subnetwork (e.g., WSN) is *ad hoc*, then each node may also serve as a router (or forwarder) for others in order to allow hop-by-hop message transmission.

B. Threat Model

Since IoT consists of a large number (hundreds of thousands) of nodes which are connected via Internet, there are many possible attacks against the message transmission in the network.

In a *passive attack*, an attacker may eavesdrop the communication channels between different nodes and perform traffic analysis. Specifically, the attacker may try to identify the content of the message or the identity of the message sender. We should note that since difference nodes may be used to collect different types of data, the disclosure of the sender identity may directly leak the message type, even if the message is encrypted.

In an *active attack*, an attacker may intercept and modify a message (i.e., perform a man-in-the-middle attack) during transmission, or inject fake messages into the network. Moreover, a node may be corrupted and controlled by an attacker. When a node is compromised, we assume that the adversary can access all the information (including the secret keys) stored in the node.

C. Security Goals

In this article, we assume that all the data are in encrypted form and hence only focus on the integrity, authenticity, and source privacy in data transmission. We summarize the security goals as follows.

1) *Authenticity:* The receiver and each forwarder in the routing path can verify that the message is sent by a legitimate data source, which can be a specific node or a node in a particular group.
2) *Integrity:* The receiver and each forwarder in the routing path can verify that the message has not been altered during transmission.
3) *Identity and location privacy:* The identity and location of the message sender is well protected. As mentioned before, the identity and location of a node may disclose some information about the data sent by that node.

III. Improving Li et al.'s SAMA Scheme

Li *et al.* [8] proposed a privacy-preserving message authentication scheme for WSNs. Their scheme can provide message source privacy and achieve better efficiency than the previous approaches in terms of computation and communication overhead. In this section, we point out a security issue that has been neglected in Li *et al.*'s [8] scheme. Specifically, we show that based on a valid message-signature pair for a particular signer group chosen by the real signer, an attacker without knowing any secret key can change the signer group and produce a valid signature for the new group. We then provide a solution to fix the problem and a suggestion to improve its efficiency.

A. SAMA Scheme

Notations: Let $E(\mathbb{F}_p)$ denote an elliptic curve (EC) over a finite field $\mathbb{F}_p$ where p is a large prime number. The set of points $(x, y) \in \mathbb{F}_p$ on the curve $E(\mathbb{F}_p)$ together with a special point $\mathcal{O}$, called the point at infinity, form an additive Abelian Group $\mathbb{G}$. Let G denote a generator of $\mathbb{G}$ whose order is a large number N. Each user selects a random integer $d_U \in [1, N-1]$ as his private key and publishes $Q_A = d_A G$ as his public key.

SAMA Scheme: Suppose that Alice wishes to send a message m anonymously from her network node to any other nodes. Alice first creates a signer group $\mathcal{S} = \{Q_1, Q_2, \ldots, Q_n\}$, where Alice's public key is Q_t, for some value $t(1 \le t \le n)$.

Authentication Generation Algorithm: Given a message m to be transmitted, Alice's private key d_t, and the chosen signer group $\mathcal{S} = \{Q_1, Q_2, \ldots, Q_n\}$, Alice performs the following three steps:

1) select random and pairwise different $k_i \in [1, N-1]$ for each $1 \le i \le n$, $i \ne t$ and compute $(r_i, y_i) = k_i G$;
2) choose a random $k_t \in [1, N-1]$ and compute $(r_t, y_t) = k_t G - \Sigma_{i \ne t} r_i h_i Q_i$ such that $r_t \ne 0$ and $r_t \ne r_i$ for any $i \ne t$, where $h_i \xleftarrow{l} h(m, r_i)$ and $\xleftarrow{l}$ denotes the l left-most bits of the hash;
3) compute $s = k_t + \Sigma_{i \ne t} k_i + r_t d_t h_t \bmod N$.

The SAMA of the message m is defined as

$$\mathcal{S}(m) = \{m, \mathcal{S}, r_1, y_1, \ldots r_n, y_n, s\}.$$

Verification of the SAMA: For Bob to verify an alleged SAMA $(m, \mathcal{S}, r_1, y_1, \ldots r_n, y_n, s)$, he first performs the following steps to verify the public keys in $\mathcal{S}$:

1) check that $Q_i \ne \mathcal{O}$ for $i = 1, \ldots, n$;
2) check that Q_i $(i = 1, \ldots, n)$ lies on the curve;
3) check that $NQ_i = \mathcal{O}$ for $i = 1, \ldots, n$.

After that Bob performs the following verifications:

1) verify that (r_i, y_i) $(i = 1, \ldots, n)$ are points on the curve and s is an integer in $[1, N-1]$, if not, the signature is invalid;
2) calculate $h_i \xleftarrow{l} h(m, r_i)$;
3) calculate $(x_0, y_0) = sG - \Sigma_{i=1}^{n} r_i h_i Q_i$.
4) the signature is valid if the first coordinate of $\Sigma_i (r_i, y_i)$ equals x_0, invalid otherwise.

The correctness of the scheme can be verified as follows:

$$\begin{aligned}(x_0, y_0) &= sG - \sum_{i=1}^{n} r_i h_i Q_i \\ &= \left(k_t + \sum_{i \ne t} k_i + r_t d_t h_t\right) G - \sum_i r_i h_i Q_i \\ &= \sum_{i \ne t} k_i G + \left(k_t G - \sum_{i \ne t} r_i h_i Q_i\right) \\ &= \sum_i (r_i, y_i).\end{aligned}$$

B. Security Issue

In this section, we show that there is a security issue in the above SAMA scheme. We show that an attacker who has intercepted an SAMA $\mathcal{S}(m) = \{m, \mathcal{S}, r_1, y_1, \ldots r_n, y_n, s\}$ can change the signer group $\mathcal{S}$ and create a new SAMA $\mathcal{S}'(m)$ that can still be verified successfully and the attacker does not need to know any secret key in order to do this.

Suppose that the attacker intercepts an SAMA $\mathcal{S}(m) = \{m, \mathcal{S}, r_1, y_1, \ldots, r_n, y_n, s\}$, then the attacker performs the following steps:

1) choose a new user $Q_{n+1} \notin \mathcal{S}$;
2) randomly select $k_{n+1} \in [1, N-1]$ and compute $(r_{n+1}, y_{n+1}) = k_{n+1} G$;
3) arbitrarily choose $j(1 \le j \le n)$ and compute $(r'_j, y'_j) = (r_j, y_j) - r_{n+1} h_{n+1} Q_{n+1}$ where $h_{n+1} \xleftarrow{l} h(m, r_{n+1})$;
4) compute $s' = s + k_{n+1} \bmod N$;
5) output the modified SAMA for $\mathcal{S}' = \{Q_1, Q_2, \ldots, Q_n, Q_{n+1}\}$ as

$$\begin{aligned}\mathcal{S}'(m) = \{&m, \mathcal{S}', r_1, y_1, \ldots, r_{j-1}, y_{j-1}, r'_j, y'_j, \\ &r_{j+1}, y_{j+1}, \ldots, r_n, y_n, r_{n+1}, y_{n+1}, s'\}.\end{aligned}$$

We can verify that the modified SAMA is valid as follows:

$$\begin{aligned}&s'G - \sum_{i=1}^{n+1} r_i h_i Q_i \\ &= sG + k_{n+1} G - \sum_{i=1}^{n} r_i h_i Q_i - r_{n+1} h_{n+1} Q_{n+1} \\ &= \sum_{i=1}^{n} (r_i, y_i) + k_{n+1} G - r_{n+1} h_{n+1} Q_{n+1} \\ &= \sum_{i=1, i \ne j}^{n} (r_i, y_i) + (r_j, y_j) + (r_{n+1}, y_{n+1}) - r_{n+1} h_{n+1} Q_{n+1} \\ &= \sum_{i=1, i \ne j}^{n+1} (r_i, y_i) + (r'_j, y'_j).\end{aligned}$$

It is easy to see that the above procedures can be repeated in order to add any new users into the signer group $\mathcal{S}$. Therefore, the goal of ensuring that the message is from *a particular group* cannot be achieved, which is undesirable when the real signer has some preferences/strategies on the selection of the users to be included in the set $\mathcal{S}$.

C. Solution for the Problem

To solve the problem, we suggest to slightly modify the SAMA scheme as follows: instead of using $h_i = h(m, r_i)$ in the signature generation and verification, we use $h_i = h(m, r_1, \ldots, r_n, i)$. The idea behind the modification is to allow the whole signer group to be included in the signature of each possible signer. The rest of the SAMA scheme remains unchanged.

Security Analysis: First of all, it is easy to verify that such a simple modification does not affect the original security analysis (namely, unforgeability for a new message and anonymity) in [8]. On the other hand, the modified scheme can effectively prevent an attacker from modifying the signer group chosen by the real signer. Specifically, we can simply threat

$m' = (m, r_1, \ldots, r_n,)$ as the "real message" being signed. If the adversary can change the signer group (i.e., the message m'), then it can break the unforgeability of the SAMA scheme. However, based on the result of [8], this cannot happen except with a negligible probability.

D. Suggestion for Better Efficiency

We notice that the SAMA scheme and its fixed version can utilize the online/offline paradigm to further improve its online efficiency. Specifically, the computation of k_iG for $1 \leq i \leq n$ can be performed offline (i.e., when the message is unavailable and the node is idle). In this way, the online computation cost can be reduced by half.

IV. New SAMA Scheme With Better Practicality and Efficiency

The SAMA scheme proposed by Li *et al.* [8] assumes all the nodes use the same system parameters. However, such an assumption may not be true in an IoT network if the nodes are deployed by different users or organizations. Under such a scenario, nodes cannot freely choose other nodes in the network to form a ring. In order to address such a practicality issue, in this section, we propose a new SAMA scheme that can be used by nodes with different system parameters. Moreover, we also apply the offline/online paradigm to improve the efficiency of the scheme.

In our new SAMA scheme presented below, for simplicity, we assume the message m is in encrypted form using a key shared between the sender and the receiver (or the public key of the receiver). In other words, we assume that only the sender and receiver can access the plain message. Based on such an assumption, we only focus on the authenticity, integrity, identity, and location privacy in the design of our new scheme.

A. Our Construction

Our construction is based on the 1-out-of-n signatures from a variety of keys proposed by Abe *et al.* [30]. Since the RSA-based signature and the discrete logarithm (DL)-based signature (e.g., DSA [31], Schnorr [32], or modified ElGamal [10]) are the most widely used digital signature schemes nowadays, we assume a terminal node in the IoT network will use one of these signature schemes.

Setup: Each RSA-based node is equipped with a public key (e_i, N_i) and a private key (d_i, N_i), whereas each DL-based node is equipped with a public key (g, p, q, y_i) and a private key x_i where $y_i = g^{x_i} \bmod p$. Let $N_{\min}$ denote the smallest RSA modulus among all the RSA nodes.

Signing Process: Without loss of generality, suppose that the ring selected by the message sender (either a RSA node or a DL node) consists of n RSA nodes (with indices from 1 to n) and n DL nodes (with indices from $n+1$ to $2n$). Let L denote all the public keys of the ring members, and $H_{\text{RSA}} : \{0,1\}^* \to \mathbb{Z}_{N_{\min}}$, $H_{DL} : \{0,1\}^* \to \mathbb{Z}_q$ two cryptographic hash functions.

1) Signing by an RSA Node: Suppose the real signer j in the ring is an RSA node, it generates a ring signature as follows.

1) Offline sign
   a) For each DL node i in the ring, select $s_i \in \mathbb{Z}_q$ and compute $v_i = g^{s_i} \bmod p$.
   b) For each RSA node $i \neq j$, choose $s_i \in \mathbb{Z}_{N_i}$ and compute $v_i = {s_i}^{e_i} \bmod N_i$.

   Store the $2n-1$ offline signatures $\{s_i, v_i\}$ for $i \neq j$.
2) Online sign
   a) (Initialization:) Randomly choose $\beta_j \in \mathbb{Z}_{N_j}$ and computes $c_{j+1} = H_{RSA}(L, m, \beta_j)$.
   b) (Forward the sequence:) For $i = j+2, \ldots, n$, compute $c_i = H_{RSA}(L, m, c_{i-1} + v_{i-1} \bmod N_{i-1})$.
   c) (Connecting RSA to DL:) $c_{n+1} = H_{DL}(L, m, c_n + v_n \bmod N_n)$.
   d) (Forward the sequence:) For $i = n+2, \ldots, 2n$, compute $c_i = H_{DL}(L, m, v_{i-1} y_{i-1}^{c_{i-1}} \bmod p)$.
   e) (Connecting DL to RSA:) $c_1 = H_{RSA}(L, m, v_{2n} y_{2n}^{c_{2n}} \bmod p)$.
   f) (Forward the sequence:) For $i = 2, \ldots, j$, compute $c_i = H_{RSA}(L, m, c_{i-1} + v_{i-1} \bmod N_{i-1})$.
   g) (Forming the ring:) Compute $s_j = (\beta_j - c_j)^{d_j} \bmod N_j$.

   The final ring signature is $\sigma = (c_1, s_1, s_2, \ldots, s_{2n})$.

2) Signing by a DL Node: Similarly, if the real signer j is a DL node, it generates a ring signature as follows.

1) Offline sign
   a) For each DL node $i \neq j$ in the ring, select $s_i \in \mathbb{Z}_q$ and compute $v_i = g^{s_i} \bmod p$.
   b) (Initialization:) Randomly choose $\alpha \in \mathbb{Z}_q$ and compute $c_{j+1} = H_{DL}(L, m, g^{\alpha} \bmod p)$.
   c) For each RSA node i, choose $s_i \in \mathbb{Z}_{N_i}$ and compute $v_i = {s_i}^{e_i} \bmod N_i$.

   Store the $2n-1$ offline signatures $\{s_i, v_i\}$ for $i \neq j$ and the initialization secret α.
2) Online sign
   a) (Forward the sequence:) For $i = j+2, \ldots, 2n$, compute $c_i = H_{DL}(L, m, v_{i-1} y_{i-1}^{c_{i-1}} \bmod p)$.
   b) (Connecting DL to RSA:) $c_1 = H_{RSA}(L, m, v_{2n} y_{2n}^{c_{2n}} \bmod p)$.
   c) (Forward the sequence:) For $i = 2, \ldots, n$, compute $c_i = H_{RSA}(L, m, c_{i-1} + v_{i-1} \bmod N_{i-1})$.
   d) (Connecting RSA to DL:) $c_{n+1} = H_{DL}(L, m, c_n + v_n \bmod N_n)$.
   e) (Forward the sequence:) For $i = n+2, \ldots, j$, compute $c_i = H_{DL}(L, m, v_{i-1} y_{i-1}^{c_{i-1}} \bmod p)$.
   f) (Forming the ring:) Compute $s_j = (\alpha - x_j c_j) \bmod q$.

The final ring signature is $\sigma = (c_1, s_1, s_2, \ldots, s_{2n})$.

3) Verifying a Message: Upon receiving an SAMA message $(L, m, c_1, s_1, s_2, \ldots, s_{2n})$, the receiver or forwarder in the routing path verifies the authenticity and integrity of the message as follows:

a) for $i = 1, \ldots, n-1$ compute $c_{i+1} = H_{RSA}(L, m, c_i + s_i^{e_i} \ mod \ N_i)$;
b) compute $c_{n+1} = H_{DL}(L, m, c_n + s_n^{e_n} \ mod \ N_n)$;

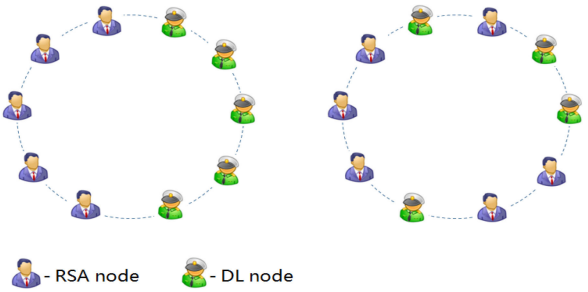

Fig. 2. Our new SAMA scheme with mixed keys.

c) for $i = n+1, \ldots, 2n-1$ compute $c_{i+1} = H_{DL}(L, m, g^{s_i} y_i^{c_i} \bmod p)$;
d) if $c_1 = H_{RSA}(L, m, g^{s_{2n}} y_{2n}^{c_{2n}} \bmod p)$, output accept; otherwise, output reject.

Remark: In our construction given above, for simplicity, we assume that half of the smart terminal nodes are RSA-based, and the other half are DL-based. It is easy to see that the real signer can set different numbers for the RSA nodes and the DL nodes to be included in the ring based on the preferences of the real signer.

Also, in the above construction, we explicitly assume a ring structure as shown in Fig. 2(a). However, the real signing node can choose any structure for the ring, e.g., it can evenly mix the RSA nodes and the DL nodes in the ring [i.e., the structure of the ring can be "$\cdots$ RSA – DL – RSA – DL $\cdots$" as shown in Fig. 2(b)].

B. Security Analysis

In this section, we provide security analysis for our new SAMA scheme presented above. As mentioned in the security goals given in Section II-C, we assume that the message is encrypted either using a symmetric key shared between the sender and the receiver or the public key of the receiver if message confidentiality is a concern. We mainly focus on the message authenticity and integrity, and the identity and location privacy of the proposed scheme as follows.

1) Message Authenticity and Integrity: The message authenticity and integrity of the new SAMA scheme can be guaranteed by the unforgeability of its underlying ring signature scheme which follows the 1-out-of-n ring signature paradigm proposed by Abe *et al.* [30]. Based on the result of [30], we can ensure that the employed ring signature is existentially unforgeable under adaptive chosen message and chosen public key attacks. During the multihop message transmission, each forwarder and the final receiver will first verify the ring signature before forwarding the message to the next node or accepting the message. Hence, if an attacker wants to inject a fake message (i.e., breaking message authenticity) or modify a message (i.e., breaking message integrity) during the transmission, the attacker must first forge a valid ring signature for the fake or modified message in order to bypass the verification, which contradicts the unforgeability of the ring signature.

2) Identity Privacy: The identity privacy of our new SAMA scheme can be guaranteed by the anonymity of the ring signature scheme. From the signing and verification procedures given above, we can see that the signature generated by any member in the ring will have the same format $\sigma = (c_1, s_1, s_2, \ldots, s_{2n})$, no matter the real signing node is RSA-based or DL-based. Also, a universal verification algorithm is used regardless of the type of the real signer. Therefore, we can conclude that from a ring signature, no one (including the forwarding nodes in the routing path) can distinguish the real signer from the dummy ones in the ring.

3) Location Privacy: In order to ensure location privacy, the real signer should choose as many geographically dispersed ring members as possible to hide its location. Since the real signer is perfectly hidden among all the ring members, eavesdroppers and forwarding nodes in the routing path are difficult to identify the location of the real sender given an SAMA message. Nevertheless, the selection of ring members should also be reasonable and consistent with the network structure and routing paths, e.g., a sensor node should pick ring members who would use the node as a forwarder when sending messages to the destination. How to strengthen the location privacy, e.g., by allowing arbitrary nodes to be used in the ring, is still an open problem and we leave it as our future work.

TABLE I
COMPUTATION COST (n DL NODES AND n RSA NODES)

Scheme	OfflineSign	OnlineSign	Verication
SAMA	-	$(4n-1)S_E$	$(2n+1)V_E$
Online/Offline SAMA	$2nS_E$	$(2n-1)S_E$	$(2n+1)V_E$
IMAEP	-	$(2n+3)S_E$	$2nV_E + 2P$
Our Scheme (RSA signer)	$nS_E + (n-1)V_R$	$nS_E + S_R$	$nV_E + nV_R$
Our Scheme (DL signer)	$nS_E + nV_R$	nS_E	$nV_E + nV_R$

S_E: DL sign exp, V_E: DL verify exp, P: Pairing, S_R: RSA sign exp, V_R: RSA verify exp.

TABLE II
COMMUNICATION OVERHEAD

Scheme	Complexity (2n Nodes)	No. of bits (n = 10)
SAMA	$2n\lvert\mathbb{G}_{DL}\rvert + \log q$	3,360
IMAEP	$(2n+2)\lvert\mathbb{G}_{DL}\rvert$	3,520
Our Scheme	$n\lvert\mathbb{G}_{DL}\rvert + (n+1)\lvert\mathbb{G}_{RSA}\rvert$	12,864

C. Efficiency Analysis and Experimental Results

In Table I, we present the computation costs of our scheme, the SAMA scheme proposed by Li *et al.* [8], and a recent ID-based message authentication with enhanced privacy (IMAEP) scheme proposed in [33]. In the table, we assume the ring consists of n RSA nodes and n DL nodes for our scheme and $2n$ nodes for the other schemes. We use S_R and V_R to represent the signature generation and verification exponentiation operations performed by an RSA node. Similarly, S_E and V_E represent the signing and verification exponentiation operations performed by a DL node. P denotes the pairing operation used by the IMAEP scheme in [33]. Tables III and IV give the real computation time based on different values of n on a laptop.

In terms of the communication overhead, Table II shows a comparison of the communication overhead among the proposed scheme, the original SAMA scheme, and the IMAEP scheme. Since the size of a RSA group element is significantly larger than that of a DL group element when the latter is implemented using ECC, the communication overhead of our proposed scheme is

TABLE III
COMPUTATION TIME (ms), 80-B SECURITY, n DL NODE, AND n RSA NODE

	n=10			n=20			n=30			n=40		
Scheme	OffS	OnS	Verify	OffS	OnS	Verify	OffS	OnS	Verify	OffS	OnS	Verify
SAMA	–	160.03	148.29	–	288.34	274.59	–	402.71	367.25	–	656.66	645.93
Online/Offline SAMA	79.63	80.40	148.29	132.95	155.38	274.59	174.35	228.36	367.25	294.15	362.50	645.93
IMAEP	–	100.16	166.89	–	173.94	304.92	–	248.19	415.87	–	384.55	694.56
Our Scheme (RSA signer)	55.04	42.84	95.46	99.36	78.37	169.59	128.51	102.38	209.54	177.93	135.40	294.79
Our Scheme (DL signer)	45.45	28.97	78.53	90.73	62.56	159.57	117.06	77.41	203.33	195.87	130.52	311.39

TABLE IV
COMPUTATION TIME (ms), 80-B SECURITY, n DL NODE, AND n RSA NODE

	n=50			n=100			n=150			n=200		
Scheme	OffS	OnS	Verify	OffS	OnS	Verify	OffS	OnS	Verify	OffS	OnS	Verify
SAMA	–	682.60	607.45	–	1399.06	1461.48	–	2203.27	2183.22	–	3027.33	3068.17
Online/Offline SAMA	306.27	376.32	607.45	603.58	795.47	1461.48	963.55	1239.71	2183.22	1208.38	1818.95	3068.17
IMAEP	–	399.31	661.07	–	831.80	1591.83	–	1295.49	2373.60	–	1896.93	3417.14
Our Scheme (RSA signer)	214.09	147.59	331.89	457.89	358.79	776.91	659.15	509.91	1196.53	898.52	754.42	1594.62
Our Scheme (DL signer)	238.22	141.17	339.89	444.53	324.57	769.84	694.00	460.57	1221.94	892.59	695.40	1586.39

TABLE V
COMPUTATION TIME (ms), 80-B SECURITY, i RSA NODE, AND j DL NODE

	i=10, j=390			i=100, j=300			i=200, j=200			i=300, j=100			i=390, j=10		
Signer	OffS	OnS	Verify	OffS	OnS	Verify	OffS	OnS	Verify	OffS	OnS	Verify	OffS	OnS	Verify
RSA	1233.64	1285.88	2494.34	1039.54	1004.68	2012.36	898.52	754.42	1594.62	730.70	500.73	1146.45	600.69	288.45	734.03
DL	1221.66	1262.10	2468.61	1040.66	955.01	1992.06	892.59	695.40	1586.39	727.38	396.97	1127.28	585.89	135.87	729.48

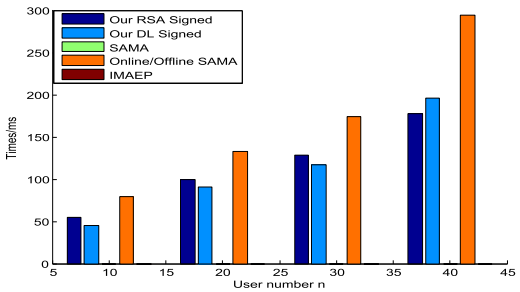

Fig. 3. Computational cost for offline sign ($n = 10, 20, 30, 40$), refer to Table III.

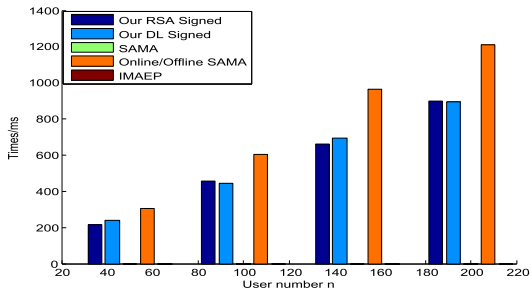

Fig. 4. Computational cost for offline sign ($n = 50, 100, 150, 200$), refer to Table IV.

higher compared with other schemes, as shown in the table. In particular, when the number of RSA nodes in the ring grows, the communication overhead of our scheme will increase.

We conducted real experiments to test the efficiency of the proposed SAMA scheme. The first experiment is conducted on a Lenovo ThinkPad X1 Carbon laptop. The device configuration is Intel(R) Core i7-7500 U CPU@2.70 GHz and 8-GB RAM with Window 10 operating system. We implement theEC version of the DL-based signature using the popular PBC library [34] for 80-b security level (i.e., 1024-bit RSA and 160-b ECC). The running time of every operation is calculated by taking the average of ten consecutive executions. Our simulation is based on the simpler architecture shown in Fig. 2(a). Since the number of operations remains the same in other settings [e.g., Fig. 2(b)], we can expect a similar result under those settings.

We also investigate the impact on computational efficiency for different configurations on the number of RSA and DL nodes. The result is presented in Table V and Figs. 9 and 10, where i indicates the number of RSA nodes and j denotes the number of DL nodes (In Table V, the total number of nodes is

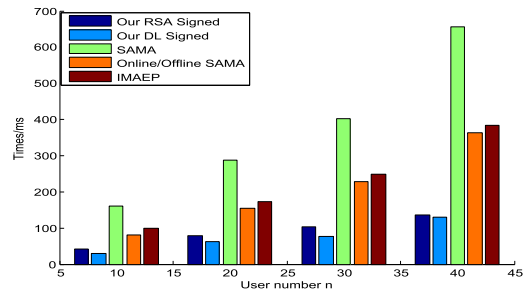

Fig. 5. Computational cost for online sign ($n = 10, 20, 30, 40$), refer to Table III.

$i + j = 400$). We should note that in the offline signing phase, the signing node needs to perform the verification (rather than signature generation) operation for RSA, since we use the public key (rather than secret key) of the RSA nodes in the ring. As a result, in opposite to the communication cost, computation cost of our scheme becomes lower when there are more RSA nodes in the ring.

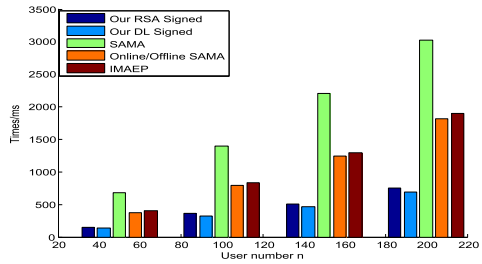

Fig. 6. Computational cost for online sign (n = 50, 100, 150, 200), refer to Table IV.

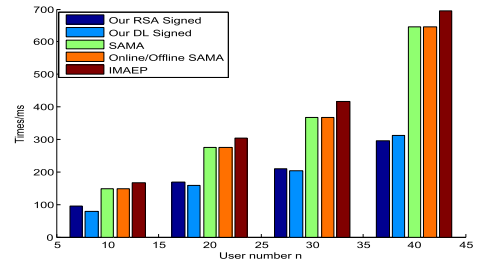

Fig. 7. Computational cost for verify (n = 10, 20, 30, 40), refer to Table III.

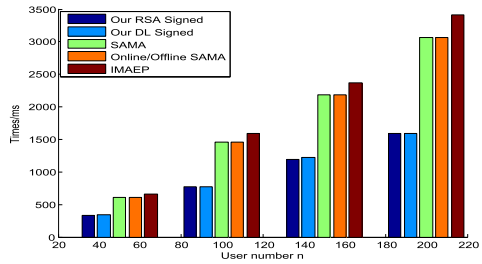

Fig. 8. Computational cost for verify (n = 50, 100, 150, 200), refer to Table IV.

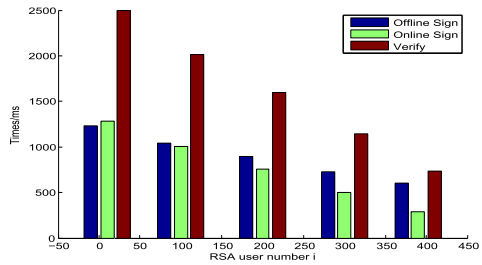

Fig. 9. Computation time for different number of RSA and DL nodes—RSA signer, refer to Table V.

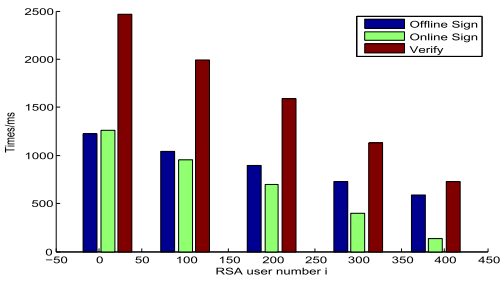

Fig. 10. Computation time for different number of RSA and DL nodes—DL signer, refer to Table V.

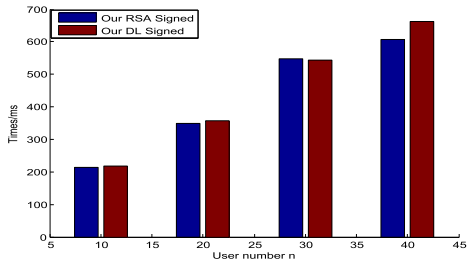

Fig. 11. Computational cost for offline sign on Raspberry Pi (n = 10, 20, 30, 40), refer to Table VI.

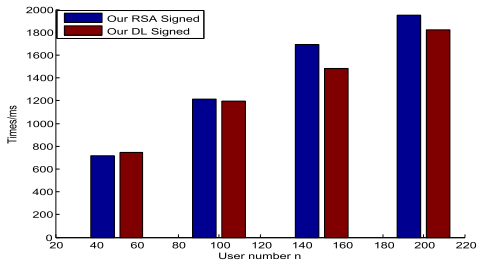

Fig. 12. Computational cost for offline sign on Raspberry Pi (n = 50, 100, 150, 200), refer to Table VII.

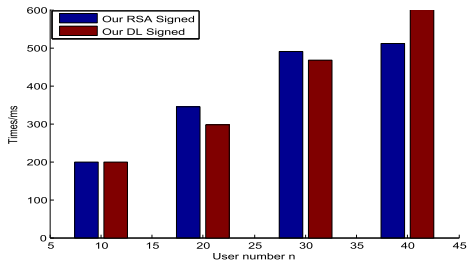

Fig. 13. Computational cost for online sign on Raspberry Pi (n = 10, 20, 30, 40), refer to Table VI.

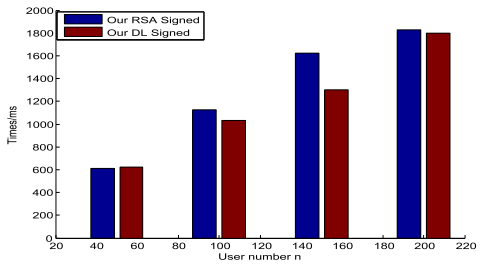

Fig. 14. Computational cost for online sign on Raspberry Pi (n = 50, 100, 150, 200), refer to Table VII.

From the above comparison (Tables III–IV and Figs. 3–8), we can see that, in general, the computation overhead of our new scheme is lower than that of (online/offline) SAMA and IMAEP. Although the signing cost of RSA is much higher than that of ECC, in our scheme, we only need to perform one RSA signing operation in the online phase if the real signing node is RSA-based. Moreover, if the real signer is DL-based, there is no RSA signing operation involved. For the signature verification cost, the new scheme also performs better than the other two schemes. Overall, we can see that the computation cost of the new scheme is much better than that of the (offline/online) SAMA scheme

TABLE VI
COMPUTATION TIME (ms) ON RASPBERRY PI, 80-B SECURITY, n DL NODE, AND n RSA NODE

	n=10			n=20			n=30			n=40		
Signer	OffS	OnS	Verify	OffS	OnS	Verify	OffS	OnS	Verify	OffS	OnS	Verify
RSA	214	200	226	350	344	390	548	490	482	607	512	532
DL	217	200	278	357	298	300	544	467	398	662	600	538

TABLE VII
COMPUTATION TIME (ms) ON RASPBERRY PI, 80-B SECURITY, n DL NODE, AND n RSA NODE

	n=50			n=100			n=150			n=200		
Signer	OffS	OnS	Verify	OffS	OnS	Verify	OffS	OnS	Verify	OffS	OnS	Verify
RSA	715	614	673	1211	1123	1123	1694	1621	1546	1952	1830	1980
DL	743	621	732	1194	1034	1023	1484	1304	1434	1822	1800	1979

TABLE VIII
COMPUTATION TIME (ms) ON RASPBERRY PI, 80-B SECURITY, i RSA NODE, AND j DL NODE

	i=10, j=390			i=100, j=300			i=200, j=200			i=300, j=100			i=390, j=10		
Signer	OffS	OnS	Verify	OffS	OnS	Verify	OffS	OnS	Verify	OffS	OnS	Verify	OffS	OnS	Verify
RSA	2380	2283	2890	2140	2019	2450	1952	1830	1980	1546	1501	1568	1473	1304	1211
DL	2333	2222	2912	2193	2001	2561	1822	1800	1979	1435	1343	1366	1322	1210	1123

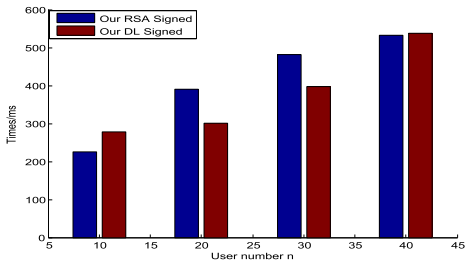

Fig. 15. Computational cost for verify on Raspberry Pi ($n = 10$, 20, 30, 40), refer to Table VI.

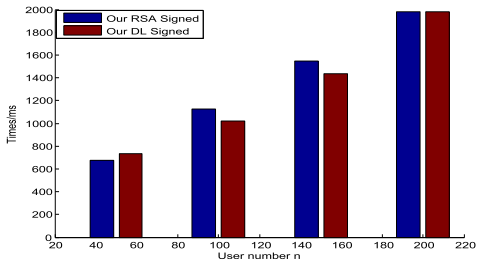

Fig. 16. Computational cost for verify on Raspberry Pi ($n = 50$, 100, 150, 200), refer to Table VII.

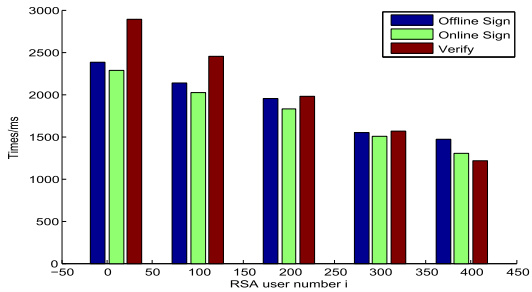

Fig. 17. Computation time for different number of RSA and DL nodes—RSA signer on Raspberry Pi, refer to Table VIII.

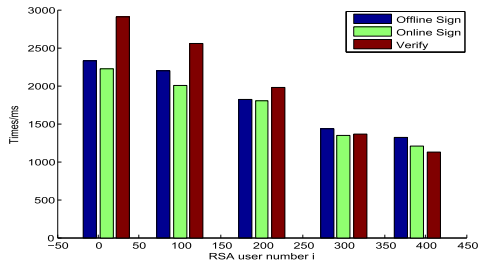

Fig. 18. Computation time for different number of RSA and DL nodes—DL signer on Raspberry Pi, refer to Table VIII.

and the IMAEP scheme, and the advantage is more significant when the ring size is large. On the other hand, the communication cost of our scheme is higher than that of the other two schemes, especially when the number of RSA nodes grows.

Experimental Results on A Raspberry Pi: To further test the efficiency of our proposed scheme in IoT devices, we also conducted experiments in a Raspberry Pi 3 with an ARMv7 processor and 2048-MB RAM. The results are presented in Tables VI–VIII and Figs. 11–18. We can see that for a ring consisting of 100 RSA nodes and 100 DL nodes, the computation time is about 1 s for offline and online signing, as well as verification, which indicates our scheme is practical to be implemented in real IoT devices.

V. CONCLUSION

In this article, we revisited a privacy-preserving message authentication scheme and showed a security weakness in the scheme. We also provided a solution to fix the problem without introducing any overhead. In order to provide better practicality in IoT consisting of different types of smart devices, we also proposed a new privacy-preserving message authentication scheme that allows IoT devices to use different security systems and parameters. Moreover, we applied the offline/online computation technique to improve the efficiency and scalability of the proposed scheme, which makes it more practical compared with the previous solution.

REFERENCES

[1] L. Da Xu, W. He, and S. Li, "Internet of Things in industries: A survey," *IEEE Trans. Ind. Informat.*, vol. 10, no. 4, pp. 2233–2243, Nov. 2014.

[2] T. Song, R. Li, B. Mei, J. Yu, X. Xing, and X. Cheng, "A privacy preserving communication protocol for IoT applications in smart homes," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 1844–1852, Dec. 2017.

[3] W. He, G. Yan, and L. Da Xu, "Developing vehicular data cloud services in the IoT environment," *IEEE Trans. Ind. Informat.*, vol. 10, no. 2, pp. 1587–1595, May 2014.

[4] J. Wei, X. Wang, N. Li, G. Yang, and Y. Mu, "A privacy-preserving fog computing framework for vehicular crowdsensing networks," *IEEE Access*, vol. 6, pp. 43 776–43 784, 2018.

[5] M. Mohammadi, A. Al-Fuqaha, S. Sorour, and M. Guizani, "Deep learning for IoT big data and streaming analytics: A survey," *IEEE Commun. Surv. Tut.*, vol. 20, no. 4, pp. 2923–2960, Oct.–Dec. 2018.

[6] J. Shen, T. Zhou, X. Liu, and Y.-C. Chang, "A novel latin-square-based secret sharing for M2M communications," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3659–3668, Aug. 2018.

[7] P. McDaniel, N. Papernot, and Z. B. Celik, "Machine learning in adversarial settings," *IEEE Secur. Privacy*, vol. 14, no. 3, pp. 68–72, May/Jun. 2016.

[8] J. Li, Y. Li, J. Ren, and J. Wu, "Hop-by-hop message authentication and source privacy in wireless sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 5, pp. 1223–1232, May 2014.

[9] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," in *Proc. Adv. Cryptology*, 1985, pp. 10–18.

[10] D. Pointcheval and J. Stern, "Security proofs for signature schemes," in *Proc. Adv. Cryptology*, 1996, pp. 387–398.

[11] R. L. Rivest, A. Shamir, and L. M. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[12] A. Perrig, R. Canetti, J. D. Tygar, and D. Song, "Efficient authentication and signing of multicast streams over lossy channels," in *Proc. IEEE Symp. Secur. Privacy,*, 2000, pp. 56–73.

[13] S. Zhu, S. Setia, S. Jajodia, and P. Ning, "An interleaved hop-by-hop authentication scheme for filtering of injected false data in sensor networks," in *Proc. IEEE Symp. Secur. Privacy*, 2004, pp. 259–271.

[14] F. Ye, H. Luo, S. Lu, and L. Zhang, "Statistical en-route filtering of injected false data in sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 4, pp. 839–850, Apr. 2005.

[15] W. Zhang, N. Subramanian, and G. Wang, "Lightweight and compromise-resilient message authentication in sensor networks," in *Proc. 27th IEEE Conf. Comput. Commun.*, 2008, pp. 1418–1426.

[16] R. L. Rivest, A. Shamir, and Y. Tauman, "How to leak a secret," in *Proc. Adv. Cryptology*, 2001, pp. 552–565.

[17] E. Fujisaki and K. Suzuki, "Traceable ring signature," in *Proc. Int. Workshop Public Key Cryptography*, 2007, pp. 181–200.

[18] D. He, N. Kumar, and N. Chilamkurti, "A secure temporal-credential-based mutual authentication and key agreement scheme with pseudo identity for wireless sensor networks," *Inf. Sci.*, vol. 321, pp. 263–277, 2015.

[19] Q. Jiang, J. Ma, F. Wei, Y. Tian, J. Shen, and Y. Yang, "An untraceable temporal-credential-based two-factor authentication scheme using ECC for wireless sensor networks," *J. Netw. Comput. Appl.*, vol. 76, pp. 37–48, 2016.

[20] S. Roy, S. Chatterjee, A. K. Das, S. Chattopadhyay, S. Kumari, and M. Jo, "Chaotic map-based anonymous user authentication scheme with user biometrics and fuzzy extractor for crowdsourcing Internet of Things," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2884–2895, Aug. 2018.

[21] X. Li, J. Niu, M. Z. A. Bhuiyan, F. Wu, M. Karuppiah, and S. Kumari, "A robust ECC-based provable secure authentication protocol with privacy preserving for industrial internet of things," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3599–3609, Aug. 2018.

[22] M. N. Aman, K. C. Chua, and B. Sikdar, "Secure data provenance for the Internet of Things," in *Proc. 3rd ACM Int. Workshop IoT Privacy, Trust, Secur.*, 2017, pp. 11–14.

[23] X. Li, J. Niu, S. Kumari, F. Wu, A. K. Sangaiah, and K.-K. R. Choo, "A three-factor anonymous authentication scheme for wireless sensor networks in Internet of Things environments," *J. Netw. Comput. Appl.*, vol. 103, pp. 194–204, 2018.

[24] D. Wang, W. Li, and P. Wang, "Measuring two-factor authentication schemes for real-time data access in industrial wireless sensor networks," *IEEE Trans. Ind. Informat.*, vol. 14, no. 9, pp. 4081–4092, Sep. 2018.

[25] Z. Cai, Z. He, X. Guan, and Y. Li, "Collective data-sanitization for preventing sensitive information inference attacks in social networks," *IEEE Trans. Dependable Secure Comput.*, vol. 15, no. 4, pp. 577–590, Jul./Aug. 2018.

[26] Y. Liu, Y. Zhang, J. Ling, and Z. Liu, "Secure and fine-grained access control on e-healthcare records in mobile cloud computing," *Future Gener. Comput. Syst.*, vol. 78, pp. 1020–1026, 2018.

[27] P. Gope, A. K. Das, N. Kumar, and Y. Cheng, "Lightweight and physically secure anonymous mutual authentication protocol for real-time data access in industrial wireless sensor networks," *IEEE Trans. Ind. Informat.*, vol. 15, no. 9, pp. 4957–4968, Sep. 2019.

[28] M. N. Aman, M. H. Basheer, and B. Sikdar, "Two-factor authentication for IoT with location information," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3335–3351, Apr. 2019.

[29] P. Gope and B. Sikdar, "Lightweight and privacy-preserving two-factor authentication scheme for IoT devices," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 580–589, Feb. 2019.

[30] M. Abe, M. Ohkubo, and K. Suzuki, "1-out-of-n signatures from a variety of keys," *IEICE Trans.*, vol. 87-A, no. 1, pp. 131–140, 2004.

[31] *Digital Signature Standard (DSS)*, FIPS Standard FIPS PUB 186-4, Jul. 2013.

[32] C. Schnorr, "Efficient signature generation by smart cards," *J. Cryptology*, vol. 4, no. 3, pp. 161–174, 1991.

[33] J. Li, Y. Liu, Z. Zhang, B. Li, H. Liu, and J. Cheng, "Efficient ID-based message authentication with enhanced privacy in wireless ad-hoc networks," in *Proc. Int. Conf. Comput., Netw. Commun.*, 2018, pp. 322–326.

[34] B. Lynn, "PBC library—the pairing-based cryptography library," *Version* 0.5.14, 2013.

Jiannan Wei (Member, IEEE) received the M.S. degree in computer science from Zhengzhou University, Zhengzhou, China, and the Ph.D. degree in information security from the School of Computing and Information Technology, University of Wollongong, Wollongong, NSW, Australia, in 2012 and 2016, respectively.

She was previously an Academic Visitor with the Monash Blockchain Technology Center, Monash University, Melbourne, VIC, Australia, in 2019. She is currently an Assistant Professor with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China. Her research interests include applied cryptography, blockchain, and network security.

Tran Viet Xuan Phuong received the bachelor's degree in information technology from Vietnam National University, Hanoi, Vietnam, the M.S. degree in computer science from the Japan Advanced Institute of Science and Technology, Nomi, Japan, and the Ph.D. degree in information security from the School of Computing and Information Technology, University of Wollongong, Wollongong, NSW, Australia, in 2010, 2012, and 2016, respectively.

She was previously a Postdoc Research Associate with Old Dominion University, Norfolk, VA, USA, and is currently a Research Fellow with CSIRO's Data61 and University of Wollongong. Her main research interest include applied cryptography, network security, and lightweight Internet of Things computation.

Guomin Yang (Senior Member, IEEE) received the Ph.D. degree in computer science from the Computer Science Department, City University of Hong Kong, Kowloon, Hong Kong, in 2009.

Formerly, he worked as a Research Scientist with the Temasek Laboratories, National University of Singapore, Singapore. He is currently an Associate Professor with the School of Computing and Information Technology, University of Wollongong, Wollongong, NSW, Australia. His research interests include cryptography and network security. Dr. Yang was a recipient of the prestigious Australian Research Council Discovery Early Career Researcher Award (DECRA) Fellowship in 2015.