sensors

MDPI

Article

Reusable Mesh Signature Scheme for Protecting Identity Privacy of IoT Devices

Ke Gu [1], WenBin Zhang [1], Se-Jung Lim [2,*], Pradip Kumar Sharma [3], Zafer Al-Makhadmeh [4] and Amr Tolba [4,5]

1 School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha 410114, China; gk4572@163.com (K.G.); 442520480@qq.com (W.Z.)
2 Liberal Arts & Convergence Studies, Honam University, Gwangju, 62399, Korea
3 Department of Multimedia Engineering, Dongguk University, Seoul, 04620, Korea pradip@dongguk.edu
4 Computer Science Department, Community College, King Saud University, Riyadh 11437, Saudi Arabia; zalmakhadmee@ksu.edu.sa (Z.A.-M.); atolba@ksu.edu.sa (A.T.)
5 Mathematics and Computer Science Department, Faculty of Science, Menoufia University, Shebin-El-kom 32511, Egypt
* Correspondence: sejunglim@126.com

Received: 30 December 2019; Accepted: 25 January 2020; Published: 30 January 2020

Abstract: The development of the Internet of Things (IoT) plays a very important role for processing data at the edge of a network. Therefore, it is very important to protect the privacy of IoT devices when these devices process and transfer data. A mesh signature (MS) is a useful cryptographic tool, which makes a signer sign any message anonymously. As a result, the signer can hide his specific identity information to the mesh signature, namely his identifying information (such as personal public key) may be hidden to a list of tuples that consist of public key and message. Therefore, we propose an improved mesh signature scheme for IoT devices in this paper. The IoT devices seen as the signers may sign their publishing data through our proposed mesh signature scheme, and their specific identities can be hidden to a list of possible signers. Additionally, mesh signature consists of some atomic signatures, where the atomic signatures can be reusable. Therefore, for a large amount of data published by the IoT devices, the atomic signatures on the same data can be reusable so as to decrease the number of signatures generated by the IoT devices in our proposed scheme. Compared with the original mesh signature scheme, the proposed scheme has less computational costs on generating final mesh signature and signature verification. Since atomic signatures are reusable, the proposed scheme has more advantages on generating final mesh signature by reconstructing atomic signatures. Furthermore, according to our experiment, when the proposed scheme generates a mesh signature on 10 MB message, the memory consumption is only about 200 KB. Therefore, it is feasible that the proposed scheme is used to protect the identity privacy of IoT devices.

Keywords: anonymity; mesh signature; IoT device; privacy; identity

1. Introduction

1.1. Background

The Internet of Things (IoT) is an important environment for processing data at the edge of a network [1], where a huge amount of data is generated in IoT. Thus, we are always surrounded by IoT data in our homes, cars and offices. IoT devices are responsible for acquiring, storing and transferring data, as shown in Figure 1. By collecting, processing and analyzing the data through IoT devices, consumers and organizations can gain valuable insights, the data can further help them

make better decisions for the future. However, since data usually comes from multiple IoT devices on different formats, after sensors acquire data from IoT devices, such as smart appliances, smart TVs, and wearable health devices, data must be preprocessed. In IoT, data may be transmitted, saved and retrieved at any time. For example, we build a system to collect location data of any things, such as a things track system. In the system, location data enables you to track your packages, pallets and devices in real time, rather than directing you to specific destinations. Therefore, as IoT devices keep "connected" and communicate with each other by introducing various new ways, IoT enables us to automatically complete certain tasks through some platforms, further making our life easier. Currently many IoT devices are located on the edge of a network and lack of protection measures to resist various attacks. Therefore, these devices are more vulnerable to some attacks, such as device theft, device manipulation, identity theft, data eavesdropping and so on. Once an IoT system is invaded, it may have a serious impact on the security of personal life or enterprise. For example, attackers may track a person by attacking his/her mobile phone; further, when a physical defense system based on IoT devices was successfully attacked in a building, it leads to that the attackers can more easily access some confidential areas in the building. Obviously the current vulnerabilities of IoT system can make attackers easier to implement these attacks. Therefore, when IoT devices process their data, their privacy is easily disclosed. It is very important to protect the privacy of IoT devices when these devices process and transfer data. Thus, the privacy of IoT devices needs to be focused. The privacy protection of IoT devices refers to the privacy protection measures to prevent the unnecessary disclosure of personal information. For the privacy protection technology of IoT devices, many scholars have done a lot of research. The current privacy protection technology mainly focuses on data publishing, data mining, wireless sensor network and other fields. In data publishing field, it is mainly divided into data distortion-based technology, data encryption-based technology and restricted publishing technology, among which the restricted publishing technology is mainly realized by data anonymity. For example, when IoT devices sign and publish their data, and the data anonymity technology may prevent disclosure of their identities. Additionally, IoT devices also need to publish a large amount of data, thus it is also very important for IoT devices to decrease the number of signatures generated by them in the same data. A mesh signature (MS) [2] allows a user to hide his specific identity information in a list of tuples that consist of public key and message when the user signs any message. Thus, mesh signature can only tell us that one of potential signers signed the message. Furthermore, a mesh signature consists of some atomic signatures, where the atomic signatures may be reused. Therefore, a mesh signature is a good choice for protecting the identities of IoT devices when these devices issue their data. For example, in some IoT devices that belong to one network group sign and publish their data through mesh signatures, no one can know the specific identities of the publishing IoT devices, and further the old mesh signatures are easily modified and reconstructed by partly generating some new atomic signatures so as to decrease the number of signatures.

A mesh signature is the extension of a ring signature [3]. Compared with ring signature, mesh signature can modularize the construction of signature, namely a user first must sign or collect enough atomic signatures which are seen as the basic elements of mesh signature, then the user may construct an access structure to mesh the atomic signatures and generate the final mesh signature. Boyen first proposed the notion of mesh signature in the Cryptology-EUROCRYPT, 2007, and a revised version [4] in the Journal of Cryptology, 2015. In the notion of mesh signatures, access structure is used to construct different combinations of atomic signatures; and mesh signature does not disclose that which atomic signature was used, thus atomic signatures can be reusable when a new mesh signature needs to be generated. Compared with a ring signature, a mesh signature has the modularity, which may provide much richer predicate expression of language. In [2,4], according to the context of mesh signature, the mesh signature may use a tree as the access structure to represent the relationship of atomic signatures. In the tree, its interior nodes denote the logic relationships, such as "And", "Or", and "Threshold gates", and its leaf nodes denote the specific atomic signatures. Thus, the construction of mesh signature is similar to another anonymous signature, attribute-based signature (ABS) [5]. Compared with other

kind of anonymous signatures (ring signature, attribute-based signature and group signature [6]), the mesh signature consists of some atomic signatures, where the atomic signatures can be reusable. Thus, the merit is very suitable for IoT devices. As IoT devices can generate a large amount of data every day, if each IoT device both needs to sign and then publish its data, then the signing cost is very heavy for itself, which needs to consume a lot of energy. However, for many IoT devices, some publishing data are the same. Thus, if each IoT device may reuse some "old" signatures by itself on the same data, then it will save the signing cost so as to decrease the number of signatures generated by IoT devices. Therefore, for a large amount of data published by the IoT devices, mesh signature is suitably used for publishing the same data.

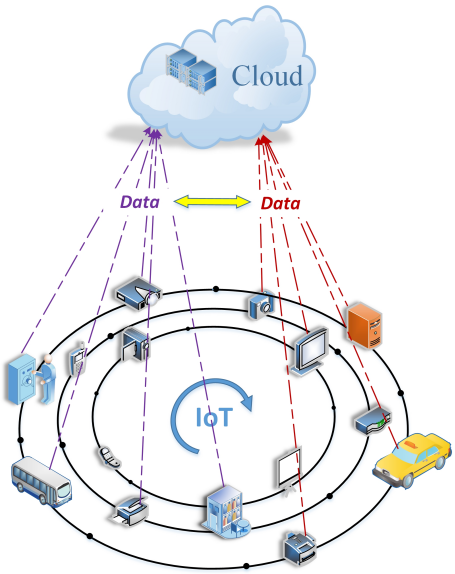

Figure 1. Data collection framework in IoT.

We have the following example to show that how the structure of mesh signature is used to protect the identities of IoT devices. For example, IoT device 1, IoT device 2 and IoT device 3 belong to a online group at the edge of the network, where the public verification key of IoT device 1 is VK_{d1}, the public verification key of IoT device 2 is VK_{d2} and the public verification key of IoT device 3 is VK_{d3}. These devices both need to send their data to the IoT data collector, as shown in Figure 2. When the IoT device 1 issues a tuple of messages {$Msg1$,$Msg2$,$Msg3$} to the IoT data collector, it does not want to disclose that these messages are only published by itself. Therefore, this device may create such mesh signature,

$$\sigma_1 = \underbrace{[VK_{d1} : Msg1]}_{atomic\ signature-1}\ And\ \underbrace{[VK_{d2} : Msg2]}_{atomic\ signature-2}\ And\ \underbrace{[VK_{d3} : Msg3]}_{atomic\ signature-3}.$$

Then this device issues these messages by the names of three devices, thus its specific identity can be hidden into these names. Additionally, another feature of mesh signature is that it is modularized and its atomic signatures can be reusable, which is suitable for the same data published by the IoT devices. For example, IoT device 1 may flexibly create a new mesh signature on other messages {$Msg4$,$Msg2$,$Msg5$},

$$\sigma_2 = \underbrace{[VK_{d1} : Msg4]}_{atomic\ signature-4}\ And\ \underbrace{[VK_{d2} : Msg2]}_{atomic\ signature-2}\ And\ \underbrace{[VK_{d3} : Msg5]}_{atomic\ signature-5},$$

where the $atomic\ signature-2$ that binds to IoT device 2 is reused. As mesh signature has perfect anonymity, it does not disclose any fact that how the two signatures σ_1 and σ_2 are made up as long as the signatures σ_1 and σ_2 are valid.

However, although mesh signatures may be used in many security fields [3,7–20], few researchers focused on the improvement of mesh signatures because of their complexity. Currently the generation of mesh signatures consists of two main steps: 1) generating some atomic signatures; 2) generating a final mesh signature based on previous atomic signatures. Because atomic signatures can be reused, randomization technology is employed so that any adversary cannot know which atomic signatures were reused. Compared with other similar anonymous signature schemes, the generation of mesh signatures is relatively complicated in the existing schemes. In this paper, we focus on improving mesh signatures, where we construct a novel mesh signature scheme for IoT devices.

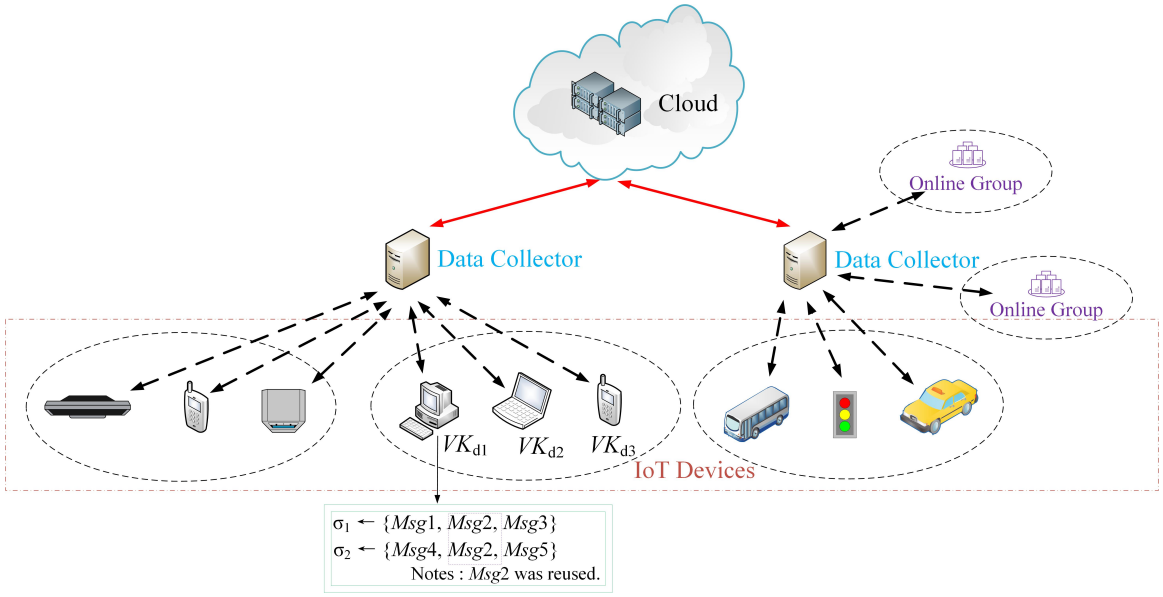

Figure 2. A example of mesh signature in IoT.

1.2. Our Contributions

In this paper, we present an improved mesh signature for protecting the identities of IoT devices. Also, we give a syntax of mesh signature in IoT. In this paper, our detailed contributions are as follows:

- We present a syntax for mesh signature in IoT. Compared with the works of [2,4], we further clearly describe the frame of mesh signature in IoT. Under the proposed syntax, we present a fully anonymous mesh signature scheme for IoT devices, where the IoT devices may be seen as the signers to sign their data and their specific identities can be hidden. Additionally, the atomic signatures on the same data can be reusable so as to decrease the number of signatures generated by IoT devices.
- In our proposed scheme, we have limitedly defined the access structure of language expression by monotone-span programs, thus the proposed mesh signature can resist the collusion attacks and its access structure still support generalized monotone predicates. Also, under the security frame proposed by [2,4], our proposed scheme is secure in the standard model, where the security of our scheme can be reduced to the CDH assumption. Also, the proposed scheme has the anonymity with enough security to protecting the identities of IoT devices.
- Compared with the original mesh signature scheme [2], the proposed scheme preserves the original modularity. Although generating atomic signatures in the proposed scheme needs more computational cost, the proposed scheme has less computational costs on generating final mesh signature and signature verification. Since atomic signatures are reusable, the proposed scheme has more advantages on generating final mesh signature by reconstructing atomic signatures.

According to our experiment, it is feasible that the proposed scheme is used to protect the identity privacy of IoT devices.

1.3. Organization

The rest of this paper is organized as follows. In Section 2, we discuss the related works about the privacy protection of IoT devices. In Section 3, we review the complexity assumptions and the related technologies on which we build. In Section 4, we show a syntax for MS in IoT. In Section 5, we propose an improved mesh signature scheme for protecting the identities of IoT devices. In Section 6, we analyze the efficiency and security of the proposed scheme. Finally, we draw our conclusions in Section 7.

2. Related Work

Currently, many signature schemes have been used to protect the privacy (identities) of IoT devices. Li [21] proposed an attribute-based signature to receive WiFi beacons and use Doppler Effect and multipath signal to produce signatures. In their scheme, because these generated signatures do not need sensor attachments, the related identities are still anonymous. Karati [1] proposed a secure certificateless signature scheme to protect industrial-IoT Environments. The proposed signature scheme is proved to be secure under bilinear strong Diffie–Hellman (BSDH) assumptions, which can resist the Type-I and Type-II attacks. Furthermore, they analyzed the performance of their scheme, which is superior to other similar schemes. Sun [22] proposed a decentralized multi-authority attribute-based signature scheme for IoT devices. Compared with other similar signature schemes, their proposed scheme has more perfect privacy and can resist authority corruption. Furthermore, their scheme employs an extra cloud server to sign messages so as to decrease the signing cost. Xie [23] proposed a novel group signature based on lattice for anonymous authentication in IoT. In their scheme, a user may dynamically join a network group, and their proposed scheme easily revoke a group membership when the user quits the group. Also, their scheme can effectively resist the frameability attack, where other users cannot forge any user's signature. Furthermore, their scheme is proved to be secure under lattice problem. Mughal [24] proposed a lightweight shortened signature scheme to secure the communication between devices in human centered IoT. In their scheme, the signing and verification procedures need less costs. Also, for different document protection requirements, their scheme provides the parameter selection function to make signature/verification. Their scheme is enough secure to resist traffic analysis attacks. Additionally, compared with other similar signature schemes, their scheme provides an experimental environment to test that whether their scheme can secure the communication procedure between cell phones (or smart devices). The obtained results show their scheme is effective. Cui [25] also proposed an attribute-based signature to protect industrial-IoT Environments under constrained resources. Their scheme employs a server to decrease the signing and verification cost, where a signing procedure can be immediately ceased when a signer is revoked. Li [26] proposed an effective ring signcryption scheme to protect the data transmission procedure from sensors to servers in IoT under public key infrastructure. They proved that their scheme is indistinguishable under adaptive chosen ciphertext attacks and unforgeable under adaptive chosen message attacks, whose security can be reduced to the computational Diffie–Hellman (CDH) assumption.

Additionally, many new anonymous signature schemes were also proposed, where the group signature [27–31], ring signature [32–34] and attribute-based signature [35–37] all belong to anonymous signatures. Libert et al. [28] proposed an effective group signature. Their proposed scheme has linear size public keys, linear size revocation list and constant signature size. Furthermore, the verification time is constant. We [31] proposed a traceable identity-based group signature, which employs verifier-local revocation to revoke users. Under the proposed security frame, the security of our scheme can be reduced to the CDH assumption. Yuen et al. [32] proposed a linkable ring signature, which is based on the logic operations, such as "and", "or" and "threshold". In their scheme, a sub-linear size $O(d \cdot \sqrt{n})$ signature can be generated, where d is a threshold and n is the number of potential

signers in a ring. Liu et al. [33] also proposed a perfect anonymous linkable ring signature scheme, where the generated signature size is still linear with the number of possible signers in a ring. Au et al. [34] proposed a novel identity-based linkable ring signature scheme, which is revocable-iff-linked. Kaafarani et al. [35] proposed some traceable attribute-based signatures, which are decentralized. Their schemes provide anonymity under adaptive chosen-ciphertext attack. We [37] proposed an attribute-based signature, which supports monotone predicates. Compared with other similar schemes, our scheme is efficient by decreasing the signing and verification cost. Boyen first proposed the original mesh signature in [2] , which may be seen as the extension of ring signature. Compared with other kind of anonymous signatures, the most advantage of mesh signature is that it can modularize the construction of signature and provide much richer predicate expression of language. In 2015, Boyen proposed a revised version in [4] . He considered that the construction of mesh signature is more flexible than that of ring signature, thus they proposed the notion of mesh signature, in which the access structure is used to construct different combinations of atomic signatures; and mesh signature does not disclose that which atomic signature was used, thus atomic signatures can be reusable when a new mesh signature needs to be generated. However, as the modularity of mesh signature is open to the construction of access structure of language expression, original mesh signature [2,4] has a security weakness that this scheme cannot satisfy the strict unforgeability because multiple illegal signers may collusively pool their obtained atomic signatures together and then generate final mesh signature which none of them could produce.

3. Preliminaries

3.1. Bilinear Maps

Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be groups of prime order q and g be a generator of $\mathbb{G}_1$. We say $\mathbb{G}_2$ has an admissible bilinear map, $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ if the following two conditions hold. The map is bilinear; for all a, b, we have $e\left(g^a, g^b\right) = e(g,g)^{a \cdot b}$. The map is non-degenerate; we must have that $e\left(g,g\right) \neq 1$.

3.2. Computational Diffie–Hellman Assumption

Definition 3.1 Computational Diffie–Hellman (CDH) Problem: *Let $\mathbb{G}_1$ be a group of prime order q and g be a generator of $\mathbb{G}_1$; for all $(g, g^a, g^b) \in \mathbb{G}_1$, with $a, b \in \mathbb{Z}_q$, the CDH problem is to compute $g^{a \cdot b}$.*
Definition 3.2 *The $(\hbar, \varepsilon)$-CDH assumption holds if no $\hbar$-time algorithm can solve the CDH problem with probability at least ε.*

3.3. Monotone-Span Programs

Let $\mathrm{Y} : \{0,1\}^n \to \{0,1\}$ be a monotone boolean function. A monotone span program [5] for Y over a field $\mathbb{F}$ is an $l \times t$ matrix Λ with entries in $\mathbb{F}$, along with a labeling function $\varpi : [l] \to [n]$ that associates each row of Λ with an input variable of Y, that, for every $(x_1, x_2 \ldots\ldots x_n) \in \{0,1\}^n$, satisfies the following:

$$\mathrm{Y}\left(x_1, \ldots\ldots x_n\right) = 1 \Leftrightarrow \exists\, \vec{\eta} \in \mathbb{F}^{1 \times l} : \vec{\eta} \cdot \Lambda = \{1,0,0 \ldots\ldots 0\} \; and \left(\forall i : x_{\varpi(i)} = 0 \Rightarrow \eta_i = 0\right);$$

in other words, $\mathrm{Y}(x_1, x_2 \ldots\ldots x_n) = 1$ if and only if the rows of Λ indexed by $\{i \mid x_{\varpi(i)} = 1\}$ span the vector $[1,0,0 \ldots\ldots 0]$, where we call l the length and t the width of the span program, and $l + t$ the size of the span program.

4. A Syntax for MS in IoT

In this section, we present a syntax for mesh signature in IoT, where each IoT device is seen as a signer, they need to issue their data to the IoT data collector. Intuitively, a mesh signature is the combination of some atomic signatures, which satisfies the condition that the monotone boolean

expression Y over access structure (or expression structure) is true. Therefore, in our proposed syntax we set that the monotone boolean expression Y is associated with a list of tuples that consist of public key and message and its value is true if one IoT device possesses some corresponding atomic signatures on the verified messages under the public verification keys, as shown in Figure 3.

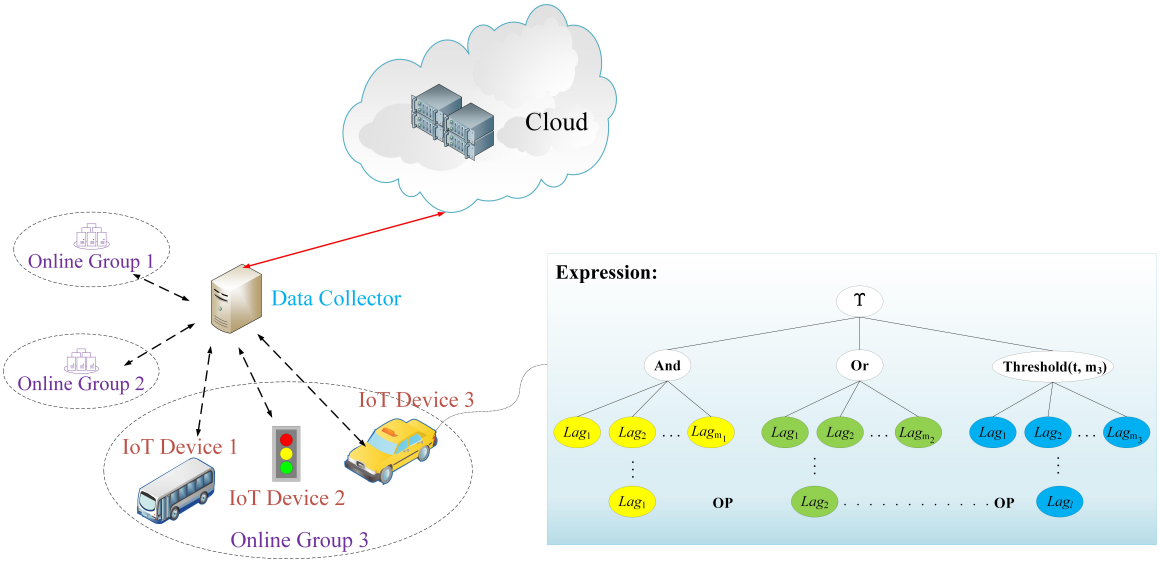

Figure 3. Syntax for mesh signature in IoT.

In Figure 3, when one IoT device belonging to a network group needs to issue its data set to the IoT data collector, the whole language expression $Expression$ is represented by the form $Expression ::= \{Lag_1$ **OP** Lag_2.....**OP** $Lag_l\}$, where Lag_i is sub-expression belongs to the whole expression, **OP** denotes the operation on the sub-expressions, l is the number of involved IoT devices belonging to the same network group (or the number of atomic clauses in a mesh structure). The more detailed and generalized form is as follows:

$$\begin{aligned} Expression ::= & \{Lag_1 \textbf{ OP } Lag_2 \text{.....} \textbf{OP } Lag_l\} \\ = & \textbf{ And}\{Lag_1 \text{ , } Lag_2 \text{..... } Lag_{m_1}\} \\ & \mid \textbf{Or}\{Lag_1 \text{ , } Lag_2 \text{..... } Lag_{m_2}\} \\ & \mid \textbf{Threshold}_{t,m_3}\{Lag_1 \text{ , } Lag_2 \text{..... } Lag_{m_3}\}, \end{aligned}$$

where we set $l = m_1 + m_2 + m_3$. Then we consider the monotone boolean expression Y over access structure is true only if $Y(Lag_1, Lag_2 \text{.....} Lag_l) = 1$. Thus, for the previous-mentioned example, $\sigma_1 = \underbrace{[VK_{d1} : Msg1]}_{atomic\ signature-1}$ *And* $\underbrace{[VK_{d2} : Msg2]}_{atomic\ signature-2}$ *And* $\underbrace{[VK_{d3} : Msg3]}_{atomic\ signature-3}$, the form of the atomic signature $[VK_i : Msg_i]$ is set to Lag_i, which means this "Msg_i" is signed under VK_i.

Definition 4.1 Improved Mesh signature in IoT: *Let* **MS**=(***System-Setup***, ***Generate-Key***, ***Mesh-Sign***, ***Mesh-Verify***) *be a mesh signature scheme in IoT. In* **MS**, *all detailed algorithms are as follows:*

1) ***System-Setup****: The authority system runs the randomized algorithm, and inputs a security parameter* 1^k. *In addition, the algorithm outputs all related public system parameters MRK and a master system private key msk on the parameter* 1^k.
2) ***Generate-Key****: The authority system runs the randomized algorithm, and inputs (MRK, msk), and then outputs the IoT device's private/public key pair* (sk_i, pk_i) *to the device i, where* $i \in \{1, 2 \text{.....} n\}$ *(we set that n is the number of the IoT devices).*
3) ***Mesh-Sign****: The randomized algorithm generates a mesh signature. The IoT device i issues its message set (data)* $\mathfrak{M} \in \{0,1\}^*$ *and then signs the message set, thus the device i runs the algorithm: (a) the*

algorithm inputs (MRK, sk_i, PK_List, $\mathfrak{M}$), and then outputs a monotone boolean expression Υ *and the atomic signatures* σ_i*; (b) the algorithm inputs (MRK, sk_i, σ_i,* Υ*), and then outputs a mesh signature* Φ*, where PK_List is a list of all the public keys of the devices involved with this signing; (c) the algorithm run by the device i sends the message set* $\mathfrak{M}$*, the boolean expression* Υ *and the mesh signature* Φ *to the IoT data collector.*

4) ***Mesh-Verify****: The IoT data collector verifies the standard mesh signature* Φ *on* Υ *and* $\mathfrak{M}$*. The IoT data collector runs the deterministic algorithm, and inputs (MRK, PK_List,* $\mathfrak{M}$*,* Υ*,* Φ*), and then outputs the result, accept or reject.*

5. Improved Mesh Signature Scheme for IoT Devices

In the section, we propose an improved mesh signature scheme for protecting the identities of IoT devices. Currently the generation of mesh signatures consists of two main steps: 1) generating some atomic signatures; 2) generating a final mesh signature based on previous atomic signatures. Because atomic signatures can be reused, in our construction the randomization technology is also employed so that any adversary cannot know which atomic signatures were reused. Compared with the original mesh signature [2,4], we have limitedly defined the access structure of language expression by monotone-span programs, thus improved mesh signature can still support generalized monotone predicates over access structure. Let **MS**=(***System-Setup***, ***Generate-Key***, ***Mesh-Sign***, ***Mesh-Verify***) be a mesh signature scheme in IoT. In **MS**, all detailed algorithms are described as follows (shown in Figure 4):

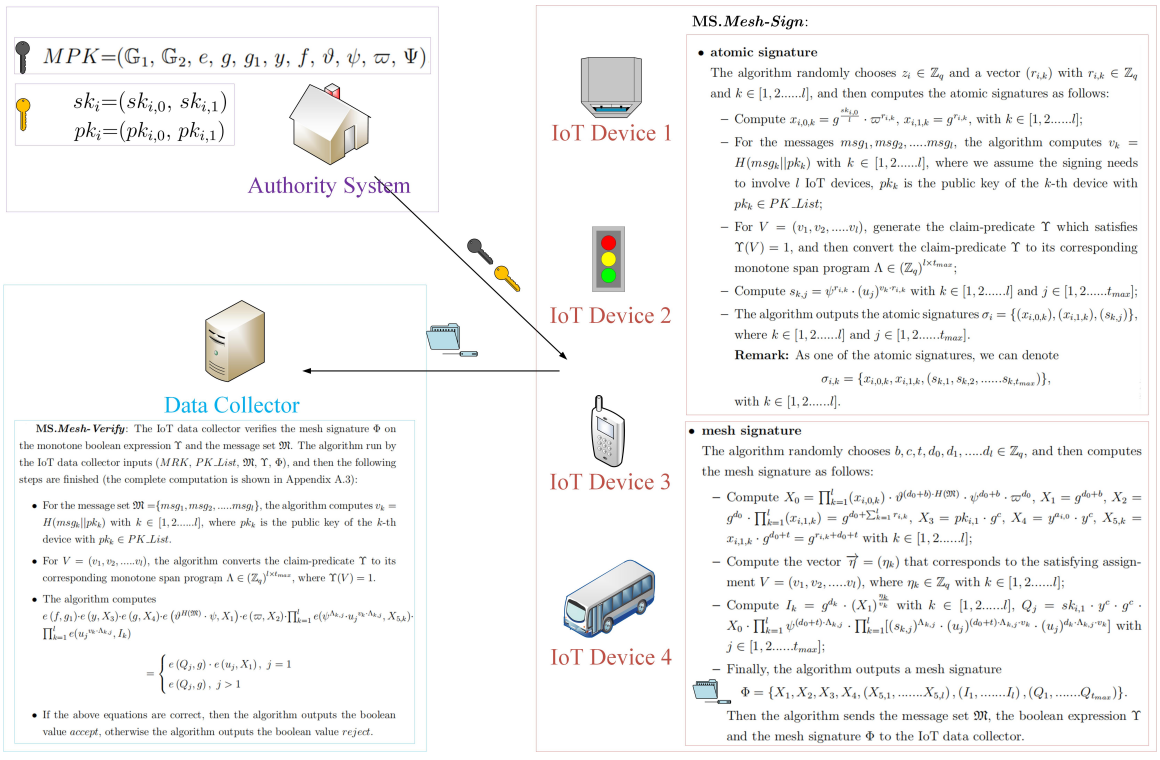

Figure 4. Improved mesh signatures for IoT devices.

1) **MS.*System-Setup***: The system runs this setup algorithm, and inputs the parameter 1^k (used as the security level). Also, we set that $\mathbb{G}_1$ and $\mathbb{G}_2$ are the groups of prime order q, g is a generator of $\mathbb{G}_1$, and that $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ denotes the bilinear map. In addition, we set that $H : \{0,1\}^* \to \mathbb{Z}_{1^k \cdot q}$ denotes one hash function and it can be used to output integers in $\mathbb{Z}_{1^k \cdot q}$.

Additionally, we assume that the monotone span programs related to claim-predicates have their width at most t_{max} in our construction.

Then the following parameters are outputted in the system. The algorithm randomly chooses $a \in \mathbb{Z}_q$ and sets $g_1 = g^a$. Five group elements y, f, ϑ, ψ and $\varpi \in \mathbb{G}_1$ are randomly picked. Also, the algorithm generates a t_{max}-length vector $\Psi = (u_i)$, whose element u_i is randomly picked from $\mathbb{G}_1$. Finally the algorithm outputs the public parameters MPK=($\mathbb{G}_1$, $\mathbb{G}_2$, e, g, g_1, y, f, ϑ, ψ, ϖ, Ψ), where $msk = a$ is a master private key in the system.

2) MS.*Generate-Key*: The system runs the algorithm and then generates IoT device's private/public key pair. For the device i, the algorithm inputs (MRK, msk), and then it randomly picks $a_{i,0}, a_{i,1} \in Z_q$, sets $sk_{i,0} = a_{i,0}$ and computes $sk_{i,1} = f^{msk} \cdot y^{a_{i,1}} = f^a \cdot y^{a_{i,1}}$, $pk_{i,0} = g^{a_{i,0}}$ and $pk_{i,1} = g^{a_{i,1}}$, where we set sk_i=($sk_{i,0}$, $sk_{i,1}$) as the private key of the device i and pk_i=($pk_{i,0}$, $pk_{i,1}$) as the public key of the device i.

3) MS.*Mesh-Sign*: The IoT device i signs a message set $\mathfrak{M} \in \{0,1\}^*$, where the message set $\mathfrak{M}$ =$\{msg_1, msg_2, ...msg_l\}$. The device i runs the algorithm, and then inputs (MRK, sk_i, PK_List, $\mathfrak{M}$) where PK_List is a list of the public keys of the IoT devices involved with this signing, and then the following steps are finished:

- **atomic signature**
  The algorithm randomly chooses $z_i \in \mathbb{Z}_q$ and a vector $(r_{i,k})$ with $r_{i,k} \in \mathbb{Z}_q$ and $k \in [1, 2.....l]$, and then computes the atomic signatures as follows:
  - Compute $x_{i,0,k} = g^{\frac{sk_{i,0}}{l}} \cdot \varpi^{r_{i,k}}$, $x_{i,1,k} = g^{r_{i,k}}$, with $k \in [1, 2.....l]$;
  - For the messages $msg_1, msg_2, ...msg_l$, the algorithm computes $v_k = H(msg_k || pk_k)$ with $k \in [1, 2.....l]$, where we assume the signing needs to involve l IoT devices, pk_k is the public key of the k-th device with $pk_k \in PK\_List$;
  - For $V = (v_1, v_2, ...v_l)$, generate the claim-predicate Y which satisfies $Y(V) = 1$, and then transform the claim-predicate Y to its corresponding monotone span program $\Lambda \in (\mathbb{Z}_q)^{l \times t_{max}}$;
  - Compute $s_{k,j} = \psi^{r_{i,k}} \cdot (u_j)^{v_k \cdot r_{i,k}}$ with $k \in [1, 2.....l]$ and $j \in [1, 2.....t_{max}]$;
  - The algorithm outputs the atomic signatures $\sigma_i = \left\{(x_{i,0,k}), (x_{i,1,k}), (s_{k,j})\right\}$, where $k \in [1, 2.....l]$ and $j \in [1, 2.....t_{max}]$.

    Remark: As one of the atomic signatures, we can denote
    $$\sigma_{i,k} = \left\{x_{i,0,k}, x_{i,1,k}, (s_{k,1}, s_{k,2}, ...s_{k,t_{max}})\right\},$$
    with $k \in [1, 2.....l]$.
- **mesh signature**
  The algorithm randomly chooses $b, c, t, d_0, d_1, ...d_l \in \mathbb{Z}_q$, and then computes the mesh signature as follows:
  - Compute $X_0 = \prod_{k=1}^{l}(x_{i,0,k}) \cdot \vartheta^{(d_0+b) \cdot H(\mathfrak{M})} \cdot \psi^{d_0+b} \cdot \varpi^{d_0}$, $X_1 = g^{d_0+b}$, $X_2 = g^{d_0} \cdot \prod_{k=1}^{l}(x_{i,1,k}) = g^{d_0+\sum_{k=1}^{l} r_{i,k}}$, $X_3 = pk_{i,1} \cdot g^c$, $X_4 = y^{a_{i,0}} \cdot y^c$, $X_{5,k} = x_{i,1,k} \cdot g^{d_0+t} = g^{r_{i,k}+d_0+t}$ with $k \in [1, 2.....l]$;
  - Compute the vector $\overrightarrow{\eta} = (\eta_k)$ related to the satisfying assignment $V = (v_1, v_2, ...v_l)$, where $\eta_k \in \mathbb{Z}_q$ with $k \in [1, 2.....l]$;
  - Compute $I_k = g^{d_k} \cdot (X_1)^{\frac{\eta_k}{v_k}}$ with $k \in [1, 2.....l]$, $Q_j = sk_{i,1} \cdot y^c \cdot g^c \cdot X_0 \cdot \prod_{k=1}^{l} \psi^{(d_0+t) \cdot \Lambda_{k,j}} \cdot \prod_{k=1}^{l}\left[(s_{k,j})^{\Lambda_{k,j}} \cdot (u_j)^{(d_0+t) \cdot \Lambda_{k,j} \cdot v_k} \cdot (u_j)^{d_k \cdot \Lambda_{k,j} \cdot v_k}\right]$ with $j \in [1, 2.....t_{max}]$;
  - The algorithm finally generates and outputs a mesh signature
    $$\Phi = \{X_1, X_2, X_3, X_4, (X_{5,1}, ...X_{5,l}), (I_1, ...I_l), (Q_1, ...Q_{t_{max}})\}.$$
    Then the algorithm sends the message set $\mathfrak{M}$, the boolean expression Y and the mesh signature Φ to the IoT data collector.

4) MS.*Mesh-Verify*: The IoT data collector verifies the mesh signature Φ on the monotone boolean expression Y and the message set $\mathfrak{M}$. The algorithm run by the IoT data collector inputs (MRK, PK_List, $\mathfrak{M}$, Y, Φ), and then the following steps are finished (the complete computation is shown in Appendix A.3):

- For the message set $\mathfrak{M} = \{msg_1, msg_2, ...msg_l\}$, the algorithm computes $v_k = H(msg_k||pk_k)$ with $k \in [1, 2.....l]$, where pk_k is the public key of the k-th device with $pk_k \in PK\_List$.
- For $V = (v_1, v_2, ...v_l)$, the algorithm transforms the claim-predicate Υ to the monotone span program $\Lambda \in (\mathbb{Z}_q)^{l \times t_{max}}$, where $\Upsilon(V) = 1$.
- The algorithm computes

$$e(f, g_1) \cdot e(y, X_3) \cdot e(g, X_4) \cdot e\left(\vartheta^{H(\mathfrak{M})} \cdot \psi, X_1\right) \cdot e(\varpi, X_2) \cdot \prod_{k=1}^{l} e(\psi^{\Lambda_{k,j}} \cdot u_j^{v_k \cdot \Lambda_{k,j}}, X_{5,k}) \cdot \prod_{k=1}^{l} e(u_j^{v_k \cdot \Lambda_{k,j}}, I_k)$$
$$= \begin{cases} e(Q_j, g) \cdot e(u_j, X_1), & j = 1 \\ e(Q_j, g), & j > 1 \end{cases}$$

If the equation is correct, the algorithm outputs *accept*, otherwise it outputs *reject*.

6. Analysis of Our Scheme

6.1. Security Analysis

In our proposed mesh signature scheme, we need to consider the two notions "one-more unforgeability" and "full anonymity". First, any IoT device cannot forge a new mesh signature on any corrupted or fresh information. Second, the anonymity of IoT device will be preserved even if some atomic signatures are reused to generate a new mesh signature, namely mesh signature and its atomic signatures must be anonymous, where we need to use the technology of randomization to randomize the generated signatures. Under the security frame proposed by [2,4], our scheme is proven to be unforgeable and anonymous.

Theorem 6.1 *Our proposed scheme is* $(\hbar, \varepsilon, q_k, q_a, q_m)$*-unforgeable, where we assume that the* $(\hbar', \varepsilon')$*-CDH assumption can hold in* $\mathbb{G}_1$*, and:*

$$\varepsilon' = (1 - \tfrac{q_k}{q}) \cdot [1 - q_a + q_a \cdot (1 - \tfrac{1}{q})^l] \cdot (1 - \tfrac{q_m}{q}) \cdot \varepsilon,$$
$$\hbar' = \hbar + O(q_k \cdot [3 \cdot C_{exp} + C_{mul}] + q_a \cdot [(2 \cdot l \cdot t_{max} + 3) \cdot C_{exp} + (l \cdot t_{max} + 1) \cdot C_{mul}] + q_m \cdot [(4 \cdot l \cdot t_{max} + 3 \cdot l + 13) \cdot C_{exp} + (4 \cdot l \cdot t_{max} + 4 \cdot l + 8) \cdot C_{mul}]),$$

q_k denotes the queries number of "Generate-Key" oracle, q_a denotes the queries number of "Atomic Signature" oracle, q_m denotes the queries number of "Mesh Signature" oracle, C_{mul} denotes the time of a multiplication in $\mathbb{G}_1$, C_{exp} denotes the time of an exponentiation in $\mathbb{G}_1$. (This proof is provided to Appendix A.1)

Theorem 6.2 *Our proposed scheme is* $(\hbar, \varepsilon, q_k, q_a, q_m)$*-anonymous, where we assume that the* $(\hbar', \varepsilon')$*-CDH assumption can hold in* $\mathbb{G}_1$*, and:*

$$\varepsilon' = (1 - \tfrac{q_{k_1}}{q}) \cdot (1 - \tfrac{q_{k_2}}{q}) \cdot [1 - q_{a_1} + q_{a_1} \cdot (1 - \tfrac{1}{q})^l] \cdot [1 - q_{a_2} + q_{a_2} \cdot (1 - \tfrac{1}{q})^l] \cdot (1 - \tfrac{q_{m_1}}{q}) \cdot (1 - \tfrac{q_{m_2}}{q}) \cdot (\varepsilon - \tfrac{1}{2}),$$
$$\hbar' = \hbar + O((q_{k_1} + q_{k_2}) \cdot [3 \cdot C_{exp} + C_{mul}] + (q_{a_1} + q_{a_2}) \cdot [(2 \cdot l \cdot t_{max} + 3) \cdot C_{exp} + (l \cdot t_{max} + 1) \cdot C_{mul}] + (q_{m_1} + q_{m_2}) \cdot [(4 \cdot l \cdot t_{max} + 3 \cdot l + 13) \cdot C_{exp} + (4 \cdot l \cdot t_{max} + 4 \cdot l + 8) \cdot C_{mul}]),$$

q_{k_1} and q_{k_2} denote the queries numbers of "Generate-Key" oracle in the query phases 1 and 2 respectively, q_{a_1} and q_{a_2} denote the queries numbers of "Atomic Signature" oracle in the query phases 1 and 2 respectively, q_{m_1} and q_{m_2} denote the queries numbers of "Mesh Signature" oracle in the query phases 1 and 2 respectively, C_{mul} denotes the time of a multiplication in $\mathbb{G}_1$, C_{exp} denotes the time of an exponentiation in $\mathbb{G}_1$. (This proof is provided to Appendix A.2)

6.2. Efficiency Analysis

In the proposed scheme, the length of the atomic signatures is $(2 \cdot l + l \cdot t_{max}) \cdot |\mathbb{G}_1|$, the length of the mesh signature is $(4 + 2 \cdot l + t_{max}) \cdot |\mathbb{G}_1|$, where $|\mathbb{G}_1|$ is the size of element in $\mathbb{G}_1$. Because $x_{i,0,k}$, $x_{i,1,k}$, $\psi^{r_{i,k}}$ in $s_{k,j}$ may be pre-computed (To make our analysis simple, we set the time of integer and hash computations is ignored.), signing a message set for the atomic signatures only computes at most $l \cdot t_{max}$ exponentiations in $\mathbb{G}_1$ and $l \cdot t_{max}$ multiplications in $\mathbb{G}_1$. Also, because X_1, X_2, X_3, X_4, $X_{5,k}$, g^{d_k}

in I_k, $\prod_{k=1}^{l}(x_{i,0,k}) \cdot \psi^{d_0+b} \cdot \omega^{d_0}$ in X_0, $sk_{i,1} \cdot y^c \cdot g^c$ in Q_j may be pre-computed, signing a message set for the mesh signature only computes at most $4 \cdot l \cdot t_{max} + l + 1$ exponentiations in $\mathbb{G}_1$ and $4 \cdot l \cdot t_{max} + l + 1$ multiplications in $\mathbb{G}_1$. In the verify algorithm, because the value $e(f, g_1)$ can be pre-computed and cached, the verification needs $(2 \cdot l + 1) \cdot t_{max} + 5$ pairing computations, $2 \cdot l \cdot t_{max}$ exponentiations in $\mathbb{G}_1$, $2 \cdot l \cdot t_{max} + 5$ multiplications in $\mathbb{G}_1$. Furthermore, we compare our proposed scheme with the original mesh signature scheme [2] in detail. Table 1 shows the performance comparison according to our theoretical analysis (In this comparison, we assume that the order of assigned structure tree in [2] is set to t_{max}.), where C_{mul} denotes the time of a multiplication in $\mathbb{G}_1$, C_{exp} denotes the time of an exponentiation in $\mathbb{G}_1$ and C_{pair} denotes the time of a pairing computation. According to Table 1, we can know although generating atomic signatures in our scheme needs more computational cost, our scheme has less computational costs on generating final mesh signature and signature verification. Since atomic signatures are reusable, our scheme has more advantages on generating final mesh signature by reconstructing atomic signatures.

Table 1. Complexity of Two Schemes.

	Atomic Signatures	Mesh Signature	Verification
Original scheme [2]	C_{exp}	$(6 \cdot (l+1) \cdot t_{max}) \cdot C_{exp} +$ $(4 \cdot l \cdot t_{max} + t_{max}) \cdot C_{mul}$	$((l+1) \cdot t_{max} + 1) \cdot C_{pair} +$ $3 \cdot (l+1) \cdot t_{max} \cdot C_{exp} +$ $3 \cdot l \cdot t_{max} \cdot C_{mul}$
Our scheme	$l \cdot t_{max} \cdot (C_{exp} + C_{mul})$	$(4 \cdot l \cdot t_{max} + l + 1) \cdot$ $(C_{exp} + C_{mul})$	$((2 \cdot l + 1) \cdot t_{max} + 5) \cdot C_{pair} +$ $2 \cdot l \cdot t_{max} \cdot C_{exp} +$ $(2 \cdot l \cdot t_{max} + 5) \cdot C_{mul}$

Additionally, we make some experiments to test and evaluate the actual performance of our scheme. In the tests, we employ the paring based cryptography (PBC) library to simulate our scheme, where the experimental computer is under Intel Core i5 2.7 GHz and RAM 8GB. In our experiments, we use the Type A parings in PBC library to construct the parings, where the lengths of the parameters p and q are respectively set as 160 bits and 512 bits. Furthermore, the parameter l is set to {1, 10, 20, 30, 40, 50}, and then we test our scheme and the original scheme [2] 10 times on average under the different settings of l. Table 2 shows the actual performance comparison of our scheme and the original scheme. Similar to our theoretical analysis, our scheme has less computational costs on generating final mesh signature and signature verification, compared with the original scheme.

Table 2. Actual Performance of Two Schemes.

		Computational Costs (ms)					
l		1	10	20	30	40	50
Original scheme [2]	Atomic Signatures	1.958	1.746	1.590	1.605	1.566	1.629
	Mesh Signature	91.495	583.225	1038.55	1617.78	2003.15	2270.82
	Verification	61.457	339.048	593.315	1001.73	1263.15	1473.34
Our scheme	Atomic Signatures	7.890	78.850	164.900	236.100	313.600	387.250
	Mesh Signature	37.752	353.003	881.153	981.836	1551.64	1675.29
	Verification	37.910	441.830	591.710	1000.43	1150.84	1128.29

Since our scheme is used to protect the identity privacy of IoT devices, we further test our memory consumption through signing different sizes of messages. Figure 5 shows the change of memory consumption by signing different sizes of messages, where the sizes of messages are set to 100 KB, 1 MB, 10 MB, 20 MB, 50 MB respectively. In Figure 5, when our scheme generates a mesh signature on 10 MB message, the memory consumption is only about 200 KB. Therefore, it is feasible that our scheme is used to protect the identity privacy of IoT devices.

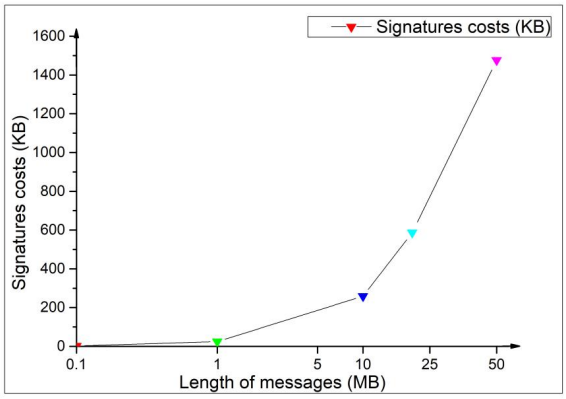

Figure 5. Memory consumption under different sizes of messages.

7. Conclusions

IoT devices are responsible for acquiring, storing, and transferring data. Currently, many IoT devices are located on the edge of a network and lack of protection measures to resist various attacks [38–43]. Therefore, these devices are more vulnerable to some attacks, such as device theft, device manipulation, identity theft, data eavesdropping and so on. Thus, the privacy of IoT devices needs to be focused. It is very important to protect the identities of IoT devices when these devices process and transfer data [44–52]. Then we present a syntax about mesh signature in IoT. Under the proposed syntax, we present a fully anonymous mesh signature scheme for IoT devices, where the IoT devices may be seen as the signers to sign their data and their specific identities can be hidden. In our proposed scheme, the generation of mesh signatures consists of two main steps: 1) generating some atomic signatures; 2) generating a final mesh signature based on previous atomic signatures. Additionally, as IoT devices can generate a large amount of data every day, if each IoT device both needs to sign and then publish its data, then the signing cost is very heavy for itself. Thus, if each IoT device reuses some "old" signatures by itself on the same data, it will save the signing cost so as to decrease the number of signatures generated by IoT devices. In our proposed scheme, the atomic signatures on the same data can be reusable so as to decrease the number of signatures. Although the atomic signatures can be reused, the randomization technology is employed so that any adversary cannot know which atomic signatures were reused. Thus, the merit is very suitable for IoT devices. Furthermore, in our proposed scheme we have limitedly defined the access structure of language expression by monotone-span programs, thus the proposed mesh signature can resist the collusion attacks and its access structure still support generalized monotone predicates. Compared with the original mesh signature scheme, our proposed scheme has its advantage, which has linear size length of signature.

Author Contributions: conceptualization, K.G. and W.Z.; methodology, S.L., A.T. and P.K.S.; formal analysis, K.G., S.L., A.T. and Z.A.-M.; writing–original draft preparation, K.G. and W.Z.; writing–review and editing, Z.A.-M., A.T. and S.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research is funded by the National Natural Science Foundation of China (No.61504013, No.61772280, No.61772454, No.61811530332, No.61811540410) and the Hunan Provincial Natural Science Foundation (No.2018JJ2445). The authors extend their appreciation to the Deanship of Scientific Research at King Saud University, Saudi Arabia for funding this work through Research Group No.RG-1439-088. Se-Jung Lim is the corresponding author.

Acknowledgments: The authors extend their appreciation to the Deanship of Scientific Research at King Saud University, Saudi Arabia for funding this work through Research Group No.RG-1439-088.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Appendix A.1 Unforgeability

(**Theorem 6.1**).

Proof: We set that **MS** is our proposed mesh signature scheme. Also we set that $\mathcal{A}$ is an adversary with the tuple ($\hbar$, ε, q_k, q_a, q_m) that can make attack to **MS**. To make interaction with the adversary $\mathcal{A}$, an algorithm $\mathcal{B}$ is constructed. For (g, g^a, g^b)$\in \mathbb{G}_1$, $\mathcal{B}$ may make interaction with $\mathcal{A}$ to compute $g^{a \cdot b}$. Then the algorithm $\mathcal{B}$ can be assumed to solve the CDH problem with probability at least ε' and in time at most $\hbar'$, which is contrary to the ($\hbar'$, ε')-CDH assumption. Therefore, we may build a simulation procedure as follows:

Setup: The parameter 1^k is inputted. Also, we set that $\mathbb{G}_1$ and $\mathbb{G}_2$ are the groups of prime order q, g is a generator of $\mathbb{G}_1$, and that $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ denotes the bilinear map. In addition, we set that $H : \{0,1\}^* \to \mathbb{Z}_{1^k \cdot q}$ denotes one hash function and it can be used to output integers in $\mathbb{Z}_{1^k \cdot q}$. Additionally, we assume that the monotone span programs related to claim-predicates have their width at most t_{max} in our construction.

Then the following parameters are outputted. The algorithm sets $g_1 = g^a$ and $f = g^b$ with $a, b \in \mathbb{Z}_q$ ($\mathcal{B}$ does not know a and b), chooses $\omega, \beta, \iota, \varphi, \varrho \in \mathbb{Z}_q$, and then sets $y = f^\omega = g^\beta$, $\vartheta = g^\iota$, $\psi = g^\varphi$ and $\varpi = g^\varrho$. In addition, the algorithm chooses $\ell_j \in \mathbb{Z}_q$ for all js with $j \in [1, 2.....t_{max}]$, and then sets $u_j = g^{\ell_j}$ for all js with $j \in [1, 2.....t_{max}]$. Then this system outputs all the parameters MRK=($\mathbb{G}_1$, $\mathbb{G}_2$, e, g, g_1, y, f, ϑ, ψ, ϖ, $\Psi = (u_j)$), where $msk = a$ is seen as the master key of the system.

Queries: $\mathcal{A}$ makes the following key and signature queries, then $\mathcal{B}$ gives its answers as follows:

- ***Generate-Key***(): Given the public parameters MRK, for the device i, the algorithm randomly chooses $a_{i,0}, a_{i,1} \in Z_q$, sets $sk_{i,0} = a_{i,0}$ and computes $sk_{i,1} = y^{a_{i,1}}$, $pk_{i,0} = g^{a_{i,0}}$ and $pk_{i,1} = g^{a_{i,1}} \cdot g_1^{-\frac{1}{\omega}}$, where sk_i=($sk_{i,0}$, $sk_{i,1}$) is the private key of the device i and pk_i=($pk_{i,0}$, $pk_{i,1}$) is the public key of the device i, and then the private/public key pair is passed to the adversary $\mathcal{A}$.

  Remark: To the correctness of sk_i and pk_i, they may be changed as follows:

  $$sk_{i,1} = y^{a_{i,1}} = f^a \cdot f^{-a} \cdot y^{a_{i,1}} = f^a \cdot f^{\omega \cdot \frac{-a}{\omega}} \cdot y^{a_{i,1}} = f^a \cdot y^{-\frac{a}{\omega}} \cdot y^{a_{i,1}} = f^a \cdot y^{a_{i,1} - \frac{a}{\omega}},$$

  $$pk_{i,1} = g^{a_{i,1}} \cdot g_1^{-\frac{1}{\omega}} = g^{a_{i,1}} \cdot g^{-\frac{a}{\omega}} = g^{a_{i,1} - \frac{a}{\omega}}.$$

  Setting $a'_{i,1} = a_{i,1} - \frac{a}{\omega}$, then $sk_{i,1} = f^a \cdot y^{a'_{i,1}}$ and $pk_{i,1} = g^{a'_{i,1}}$. Therefore, sk_i and pk_i is a valid private/public key pair.

  If $a_{i,1} - \frac{a}{\omega} = 0 \bmod q$, the above procedure cannot occur and aborts. Otherwise, a private/public key pair is outputted to $\mathcal{A}$.
- ***Atomic-Sign***(): Given the public parameters MRK, the public key list PK_List and the message $\mathfrak{M}$, where PK_List is a list of the public keys of the devices involved with this query (with respect to the device i), the algorithm finishes the following steps:
  - The algorithm randomly chooses $sk_{i,0}, z_i \in \mathbb{Z}_q$ and a vector ($r_{i,k}$) with $r_{i,k} \in \mathbb{Z}_q$ and $k \in [1, 2.....l]$, computes $x_{i,0,k} = g^{\frac{sk_{i,0}}{l}} \cdot \varpi^{r_{i,k}}$ and $x_{i,1,k} = g^{r_{i,k}}$ with $k \in [1, 2.....l]$, and then saves $sk_{i,0}$ where $sk_{i,0} = a_{i,0}$;
  - The message $\mathfrak{M}$ is divided to $msg_1, msg_2, ...msg_l$; then the algorithm computes $v_k = H(msg_k || pk_k)$ with $k \in [1, 2.....l]$, where we assume the signing needs to involve l devices, pk_k is the public key of the k-th device with $pk_k \in PK\_List$;
  - For $V = (v_1, v_2, ...v_l)$, generate the claim-predicate Y which satisfies $\Upsilon(V) = 1$, and then transform the claim-predicate Y to the monotone span program $\Lambda \in (\mathbb{Z}_q)^{l \times t_{max}}$;
  - Compute $s_{k,j} = \psi^{r_{i,k}} \cdot (u_j)^{v_k \cdot r_{i,k}}$ with $k \in [1, 2.....l]$ and $j \in [1, 2.....t_{max}]$;
  - The algorithm outputs the atomic signatures $\sigma_a = \left\{(x_{i,0,k}), (x_{i,1,k}), (s_{k,j})\right\}$ to $\mathcal{A}$, where $k \in [1, 2.....l]$ and $j \in [1, 2.....t_{max}]$.

If $v_k = H(msg_k||pk_k) = 0 \bmod q$ with $k \in [1, 2.....l]$, then the above procedure cannot ocur and aborts. Otherwise, the atomic signatures are passed to the adversary $\mathcal{A}$.

- ***Mesh-Sign***(): Given the public parameters MRK, the atomic signatures $\sigma_a = \left\{(x_{i,0,k}), (x_{i,1,k}), (s_{k,j})\right\}$ on the public key list PK_List and the message $\mathfrak{M}$ (with respect to the device i), and the monotone boolean expression Y, the algorithm finishes the following steps:
  - Choose $b, c, t, d_0, d_1, ...d_l, a_{i,1} \in \mathbb{Z}_q$ randomly, compute $X_0 = \prod_{k=1}^{l}(x_{i,0,k}) \cdot \vartheta^{(d_0+b)\cdot H(\mathfrak{M})} \cdot \psi^{d_0+b} \cdot \varpi^{d_0}$, $X_1 = g^{d_0+b}$, $X_2 = g^{d_0} \cdot \prod_{k=1}^{l}(x_{i,1,k}) = g^{d_0+\sum_{k=1}^{l} r_{i,k}}$, $X_3 = g^{a_{i,1}} \cdot g_1^{-\frac{1}{\omega}} \cdot g^c$, $X_4 = y^{sk_{i,0}} \cdot y^c$ according to the corresponding $sk_{i,0}$, $X_{5,k} = x_{i,1,k} \cdot g^{d_0+t} = g^{r_{i,k}+d_0+t}$ with $k \in [1, 2.....l]$;
  - Compute the vector $\overrightarrow{\eta} = (\eta_k)$ related to the satisfying assignment $V = (v_1, v_2, ...v_l)$, where $\eta_k \in \mathbb{Z}_q$ with $k \in [1, 2.....l]$;
  - Compute $I_k = g^{d_k} \cdot (X_1)^{\frac{\eta_k}{v_k}}$ with $k \in [1, 2.....l]$, $Q_j = y^{a_{i,1}} \cdot y^c \cdot g^c \cdot X_0 \cdot \prod_{k=1}^{l} \psi^{(d_0+t)\cdot\Lambda_{k,j}} \cdot \prod_{k=1}^{l}[(s_{k,j})^{\Lambda_{k,j}} \cdot (u_j)^{(d_0+t)\cdot\Lambda_{k,j}\cdot v_k} \cdot (u_j)^{d_k\cdot\Lambda_{k,j}\cdot v_k}]$ with $j \in [1, 2..... .t_{max}]$;
  - The algorithm finally generates and outputs a mesh signature

$$\sigma_m = \{X_1, X_2, X_3, X_4, (X_{5,1}, ...X_{5,l}), (I_1, ...I_l), (Q_1, ...Q_{t_{max}})\}.$$

Similarly, setting $a'_{i,1} = a_{i,1} - \frac{a}{\omega}$, σ_m is a valid mesh signature. If $a_{i,1} - \frac{a}{\omega} = 0 \bmod q$, the above procedure cannot occur and aborts. Otherwise, a mesh signature σ_m is outputted to $\mathcal{A}$.

Forgery: If $\mathcal{B}$ finally does not abort, then $\mathcal{A}$ can return its forgery with probability at least ε, (MRK, PK_List^*, $\mathfrak{M}^*$, Y^*, Φ^*), where Y^* can be converted to the corresponding monotone span program $\Lambda^* \in (\mathbb{Z}_q)^{l\times t_{\max}}$, the vector $\overrightarrow{\eta}^* = (\eta_k^*)$ is related to the satisfying assignment $V^* = (v_1^*, v_2^*, ...v_l^*)$ with $k \in [1, 2.....l]$. It succeeds if

(a) $accept \leftarrow$***Mesh-Verify***(MRK, PK_List^*, $\mathfrak{M}^*$, Y^*, Φ^*);
(b) $\mathcal{A}$ did not query ***Generate-Key*** on any public key belongs to PK_List^*, and it did not query ***Mesh-Sign*** on the related inputs PK_List^*, $\mathfrak{M}^*$ and Y^*.

Then we may get the following:

$$\Phi^* = \{X_1^*, X_2^*, X_3^*, X_4^*, \left(X_{5,1}^*, X_{5,2}^*, ...X_{5,l}^*\right), (I_1{}^*, I_2{}^*, ...I_l{}^*), (Q_1{}^*, Q_2{}^*, ...Q_{t_{max}}^*)\},$$

where $\mathrm{Y}^*(V^*) = 1$, and

$$\begin{aligned}
&X_1^* = g^{d_0^*+b^*},\\
&X_2^* = g^{d_0^*+\sum_{k=1}^{l} r_{i,k}^*},\\
&X_3^* = g^{a_{i,1}^*+c^*},\\
&X_4^* = y^{a_{i,0}^*+c^*},\\
&X_{5,k}^* = g^{r_{i,k}^*+d_0^*+t^*},\\
&I_k^* = g^{d_k^*} \cdot (X_1^*)^{\frac{\eta_k^*}{v_k^*}},\\
&Q_j^* = f^a \cdot y^{a_{i,1}^*+c^*} \cdot g^{c^*+a_{i,0}^*} \cdot \vartheta^{(d_0^*+b^*)\cdot H(\mathfrak{M}^*)} \cdot \psi^{d_0^*+b^*} \cdot \varpi^{d_0^*+\sum_{k=1}^{l} r_{i,k}^*} \cdot \prod_{k=1}^{l} \psi^{(d_0^*+t+r_{i,k}^*)\cdot\Lambda_{k,j}^*} \cdot \\
&\prod_{k=1}^{l}(u_j)^{(d_0^*+t^*+r_{i,k}^*)\cdot\Lambda_{k,j}^*\cdot v_k^*} \cdot \prod_{k=1}^{l}(u_j)^{d_k^*\cdot\Lambda_{k,j}^*\cdot v_k^*}.
\end{aligned}$$

Finally, the algorithm $\mathcal{B}$ computes and outputs

$$\frac{Q_j^*}{(X_3^*)^{\beta}\cdot(X_4^*)^{\frac{1}{\beta}}\cdot(X_1^*)^{\iota\cdot H(\mathfrak{M}^*)}\cdot(X_1^*)^{\varphi}\cdot(X_2^*)^{\varrho}\cdot\prod_{k=1}^{l}(X_{5,k}^*)^{\varphi\cdot\Lambda_{k,j}^*}\cdot\prod_{k=1}^{l}(X_{5,k}^*)^{\ell_j\cdot\Lambda_{k,j}^*\cdot v_k^*}\cdot\prod_{k=1}^{l} g^{\ell_j\cdot d_k^*\cdot\Lambda_{k,j}^*\cdot v_k^*}} = f^a = g^{a\cdot b},$$

which solves the given CDH problem.

Then, we compute the probability that $\mathcal{B}$ does not abort. For the complete simulation procedure of $\mathcal{B}$, we must assure that all key queries can have $a_{i,1} - \frac{a}{\omega} \neq 0 \bmod q$, all atomic signature queries

can have $v_k = H(msg_k||pk_k) \neq 0 \bmod q$ for all $k \in [1, 2.....l]$, and all mesh signature queries can have $a_{i,1} - \frac{a}{\omega} \neq 0 \bmod q$. Therefore, if $\mathcal{B}$ will not abort, then we must assure that the following three conditions hold:

(a) $a_{i,1} - \frac{a}{\omega} \neq 0 \bmod q$ in related key queries;
(b) $v_k = H(msg_k||pk_k) \neq 0 \bmod q$ for all $k \in [1, 2.....l]$ in related atomic signature queries;
(c) $a_{i,1} - \frac{a}{\omega} \neq 0 \bmod q$ in related mesh signature queries.

To make our analysis easier to understand, we define the events E_j, R_j and T_j as

E_j :$a_{i,1} - \frac{a}{\omega} \neq 0 \bmod q$, with j=1, 2.....q_k, q_k denotes the queries number of "Generate-Key" oracle;

R_j :$v_k = H(msg_k||pk_k) \neq 0 \bmod q$ for all $k \in [1, 2.....l]$, with j=1, 2.....q_a, q_a denotes the queries number of "Atomic Signature" oracle;

T_j :$a_{i,1} - \frac{a}{\omega} \neq 0 \bmod q$, with j=1, 2.....q_m, q_m denotes the queries number of "Mesh Signature" oracle.

The probability that $\mathcal{B}$ is completely simulated is $\Pr(not\_abort) = \Pr\left(\bigcap_{j=1}^{q_k} E_j \wedge \bigcap_{j=1}^{q_a} R_j \wedge \bigcap_{j=1}^{q_m} T_j\right)$. It is easy to see that the events $\bigcap_{j=1}^{q_k} E_j$, $\bigcap_{j=1}^{q_a} R_j$, $\bigcap_{j=1}^{q_m} T_j$ are independent. Then we may compute

$$\Pr(\bigcap_{j=1}^{q_k} E_j) = 1 - \Pr(\bigcup_{j=1}^{q_k} \neg E_j) = 1 - q_k \cdot \frac{1}{q} = 1 - \frac{q_k}{q};$$

$$\Pr(\bigcap_{j=1}^{q_a} R_j) = 1 - \Pr(\bigcup_{j=1}^{q_a} \neg R_j) = 1 - q_a \cdot [1 - (1 - \frac{1^k}{1^k \cdot q})^l] = 1 - q_a + q_a \cdot (1 - \frac{1}{q})^l;$$

$$\Pr(\bigcap_{j=1}^{q_m} T_j) = 1 - \Pr(\bigcup_{j=1}^{q_m} \neg T_j) = 1 - q_m \cdot \frac{1}{q} = 1 - \frac{q_m}{q}.$$

Therefore,

$$\begin{aligned}\Pr(not\_abort) &= \Pr\left(\bigcap_{j=1}^{q_k} E_j \wedge \bigcap_{j=1}^{q_a} R_j \wedge \bigcap_{j=1}^{q_m} T_j\right) \\ &= \Pr(\bigcap_{j=1}^{q_k} E_j) \cdot \Pr(\bigcap_{j=1}^{q_a} R_j) \cdot \Pr(\bigcap_{j=1}^{q_m} T_j) \\ &= (1 - \tfrac{q_k}{q}) \cdot [1 - q_a + q_a \cdot (1 - \tfrac{1}{q})^l] \cdot (1 - \tfrac{q_m}{q}).\end{aligned}$$

Therefore, we can get that $\varepsilon' = (1 - \frac{q_k}{q}) \cdot [1 - q_a + q_a \cdot (1 - \frac{1}{q})^l] \cdot (1 - \frac{q_m}{q}) \cdot \varepsilon$.

If $\mathcal{B}$ is completely simulated, then $\mathcal{A}$ generates a valid mesh signature forgery with probability at least ε, and $\mathcal{B}$ may be used to compute $g^{a \cdot b}$. The time cost of $\mathcal{B}$ mainly includes the time of the exponentiations and multiplications in queries. We assume that the time of other lightweight computations is ignored (such as integer addition, integer multiplication and hash computation), then the time cost of $\mathcal{B}$ is

$$\hbar' = \hbar + O(q_k \cdot [3 \cdot C_{exp} + C_{mul}] + q_a \cdot [(2 \cdot l \cdot t_{max} + 3) \cdot C_{exp} + (l \cdot t_{max} + 1) \cdot C_{mul}] + q_m \cdot [(4 \cdot l \cdot t_{max} + 3 \cdot l + 13) \cdot C_{exp} + (4 \cdot l \cdot t_{max} + 4 \cdot l + 8) \cdot C_{mul}]).$$

Thus, Theorem 6.1 follows.

Appendix A.2 Anonymity

(**Proof of Theorem 6.2**[1]).
Proof: We set that **MS** is our proposed mesh signature scheme. Also we set that $\mathcal{A}$ is an adversary

[1] This proof is similar to that of Theorem 6.1, the difference between them is to add the queries of phase 2.

with the tuple ($\hbar$, ε, q_k, q_a, q_m) that can make attack to **MS**. To make interaction with the adversary $\mathcal{A}$, an algorithm $\mathcal{B}$ is constructed. For (g, g^a, g^b)$\in \mathbb{G}_1$, $\mathcal{B}$ may make interaction with $\mathcal{A}$ to compute $g^{a \cdot b}$. Then the algorithm $\mathcal{B}$ can be assumed to solve the CDH problem with probability at least ε' and in time at most $\hbar'$, which is contrary to the ($\hbar'$, ε')-CDH assumption. Therefore, we may build a simulation procedure as follows:

1.Setup: The parameter 1^k is inputted. Also, we set that $\mathbb{G}_1$ and $\mathbb{G}_2$ are the groups of prime order q, g is a generator of $\mathbb{G}_1$, and that $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ denotes the bilinear map. In addition, we set that $H : \{0,1\}^* \to \mathbb{Z}_{1^k \cdot q}$ denotes one hash function and it can be used to output integers in $\mathbb{Z}_{1^k \cdot q}$. Additionally, we assume that the monotone span programs related to claim-predicates have their width at most t_{max} in our construction.

Then the following parameters are outputted. The algorithm sets $g_1 = g^a$ and $f = g^b$ with $a, b \in \mathbb{Z}_q$ ($\mathcal{B}$ does not know a and b), chooses $\omega, \beta, \iota, \varphi, \varrho \in \mathbb{Z}_q$, and then sets $y = f^\omega = g^\beta$, $\vartheta = g^\iota$, $\psi = g^\varphi$ and $\varpi = g^\varrho$. In addition, the algorithm chooses $\ell_j \in \mathbb{Z}_q$ for all js with $j \in [1, 2.....t_{max}]$, and then sets $u_j = g^{\ell_j}$ for all js with $j \in [1, 2.....t_{max}]$. Then this algorithm outputs all the parameters MRK=($\mathbb{G}_1$, $\mathbb{G}_2$, e, g, g_1, y, f, ϑ, ψ, ϖ, $\Psi = (u_j)$), where $msk = a$ is seen as the master key of the system.

2.Queries Phase 1: $\mathcal{A}$ makes the following key and signature queries, then $\mathcal{B}$ gives its answers as follows:

- ***Generate-Key***(): Given the public parameters MRK, for the device i, the algorithm randomly chooses $a_{i,0}, a_{i,1} \in Z_q$, sets $sk_{i,0} = a_{i,0}$ and computes $sk_{i,1} = y^{a_{i,1}}$, $pk_{i,0} = g^{a_{i,0}}$ and $pk_{i,1} = g^{a_{i,1}} \cdot g_1^{-\frac{1}{\omega}}$, where sk_i=($sk_{i,0}$, $sk_{i,1}$) is the private key of the device i and pk_i=($pk_{i,0}$, $pk_{i,1}$) is the public key of the device i, and then the private/public key pair is passed to the adversary $\mathcal{A}$. Similarly, setting $a'_{i,1} = a_{i,1} - \frac{a}{\omega}$, then $sk_{i,1} = f^a \cdot y^{a'_{i,1}}$ and $pk_{i,1} = g^{a'_{i,1}}$. Therefore, sk_i and pk_i is a valid private/public key pair.

  If $a_{i,1} - \frac{a}{\omega} = 0 \bmod q$, the above procedure cannot occur and aborts. Otherwise, a private/public key pair is outputted to $\mathcal{A}$.
- ***Atomic-Sign***(): Given the public parameters MRK, the public key list PK_List and the message $\mathfrak{M}$, where PK_List is a list of the public keys of the devices involved with this query (with respect to the device i), the algorithm finishes the following steps:
  - The algorithm randomly chooses $sk_{i,0}, z_i \in \mathbb{Z}_q$ and a vector $(r_{i,k})$ with $r_{i,k} \in \mathbb{Z}_q$ and $k \in [1, 2.....l]$, computes $x_{i,0,k} = g^{\frac{sk_{i,0}}{l}} \cdot \varpi^{r_{i,k}}$ and $x_{i,1,k} = g^{r_{i,k}}$ with $k \in [1, 2.....l]$, and then saves $sk_{i,0}$ where $sk_{i,0} = a_{i,0}$;
  - The message $\mathfrak{M}$ is divided to $msg_1, msg_2, ...msg_l$; then the algorithm computes $v_k = H(msg_k || pk_k)$ with $k \in [1, 2.....l]$, where we assume the signing needs to involve l devices, pk_k is the public key of the k-th device with $pk_k \in PK\_List$;
  - For $V = (v_1, v_2, ...v_l)$, generate the claim-predicate Y which satisfies $\mathrm{Y}(V) = 1$, and then transform the claim-predicate Y to the monotone span program $\Lambda \in \left(\mathbb{Z}_q\right)^{l \times t_{max}}$;
  - Compute $s_{k,j} = \psi^{r_{i,k}} \cdot (u_j)^{v_k \cdot r_{i,k}}$ with $k \in [1, 2.....l]$ and $j \in [1, 2.....t_{max}]$;
  - The algorithm outputs the atomic signatures $\sigma_a = \left\{(x_{i,0,k}), (x_{i,1,k}), (s_{k,j})\right\}$ to $\mathcal{A}$, where $k \in [1, 2.....l]$ and $j \in [1, 2.....t_{max}]$.

  If $v_k = H(msg_k || pk_k) = 0 \bmod q$ with $k \in [1, 2.....l]$, then the above procedure cannot occur and will abort; otherwise the atomic signatures are passed to the adversary $\mathcal{A}$.
- ***Mesh-Sign***(): Given the public parameters MRK, the atomic signatures $\sigma_a = \left\{(x_{i,0,k}), (x_{i,1,k}), (s_{k,j})\right\}$ on the public key list PK_List and the message $\mathfrak{M}$ (with respect to the device i), and the monotone boolean expression Y, the algorithm finishes the following steps:
  - Choose $b, c, t, d_0, d_1, ...d_l, a_{i,1} \in \mathbb{Z}_q$ randomly, compute $X_0 = \prod_{k=1}^{l}(x_{i,0,k}) \cdot \vartheta^{(d_0+b) \cdot H(\mathfrak{M})} \cdot \psi^{d_0+b} \cdot \varpi^{d_0}$, $X_1 = g^{d_0+b}$, $X_2 = g^{d_0} \cdot \prod_{k=1}^{l}(x_{i,1,k}) = g^{d_0+\sum_{k=1}^{l} r_{i,k}}$, $X_3 = g^{a_{i,1}} \cdot g_1^{-\frac{1}{\omega}} \cdot g^c$, $X_4 =$

$y^{sk_{i,0}} \cdot y^c$ according to the corresponding $sk_{i,0}$, $X_{5,k} = x_{i,1,k} \cdot g^{d_0+t} = g^{r_{i,k}+d_0+t}$ with $k \in [1, 2.....l]$;
- Compute the vector $\overrightarrow{\eta} = (\eta_k)$ related to the satisfying assignment $V = (v_1, v_2, ...v_l)$, where $\eta_k \in \mathbb{Z}_q$ with $k \in [1, 2.....l]$;
- Compute $I_k = g^{d_k} \cdot (X_1)^{\frac{\eta_k}{v_k}}$ with $k \in [1, 2.....l]$, $Q_j = y^{a_{i,1}} \cdot y^c \cdot g^c \cdot X_0 \cdot \prod_{k=1}^{l} \psi^{(d_0+t)\cdot\Lambda_{k,j}} \cdot \prod_{k=1}^{l}[(s_{k,j})^{\Lambda_{k,j}} \cdot (u_j)^{(d_0+t)\cdot\Lambda_{k,j}\cdot v_k} \cdot (u_j)^{d_k\cdot\Lambda_{k,j}\cdot v_k}]$ with $j \in [1, 2..... .t_{max}]$;
- The algorithm finally generates and outputs a mesh signature

$$\sigma_m = \{X_1, X_2, X_3, X_4, (X_{5,1}, ...X_{5,l}), (I_1, ...I_l), (Q_1, ...Q_{t_{max}})\}.$$

Similarly, setting $a'_{i,1} = a_{i,1} - \frac{a}{\omega}$, σ_m is a valid mesh signature. If $a_{i,1} - \frac{a}{\omega} = 0 \bmod q$, the above procedure cannot occur and aborts. Otherwise, a mesh signature σ_m is passed to $\mathcal{A}$.

3.Challenge: The adversary $\mathcal{A}$ sends its forgeries (MRK, $PK\_List^* \cup \{pk_0^*\} \cup \{pk_1^*\}$, $\mathfrak{M}^*$, Υ^*, Φ^*) to the challenger. The following conditions are satisfies:

(a) The adversary did not make query to ***Generate-Key*** on pk_0^* (and pk_1^*);
(b) The adversary did not make query to ***Atomic-Sign*** on pk_0^* (and pk_1^*);
(c) The adversary did not make query to ***Mesh-Sign*** on pk_0^* (and pk_1^*).

The challenger randomly chooses a bit $x \in \{0, 1\}$, and then the following is outputted as

$$\sigma^* \leftarrow \textbf{\textit{Mesh-Sign}}(MRK, sk_x^*, PK\_List^* \cup \{pk_0^*\} \cup \{pk_1^*\}, \mathfrak{M}) \text{ to } \mathcal{A}.$$

4.Queries Phase 2: $\mathcal{A}$ makes the following key and signature queries, then $\mathcal{B}$ gives its answers as follows:

- ***Generate-Key***(): Given the public parameters MRK, for the device i, the algorithm randomly chooses $a_{i,0}, a_{i,1} \in Z_q$, sets $sk_{i,0} = a_{i,0}$ and computes $sk_{i,1} = y^{a_{i,1}}$, $pk_{i,0} = g^{a_{i,0}}$ and $pk_{i,1} = g^{a_{i,1}} \cdot g_1^{-\frac{1}{\omega}}$, where sk_i=($sk_{i,0}$, $sk_{i,1}$) is the private key of the device i and pk_i=($pk_{i,0}$, $pk_{i,1}$) is the public key of the device i, and then the private/public key pair is passed to the adversary $\mathcal{A}$. Similarly, setting $a'_{i,1} = a_{i,1} - \frac{a}{\omega}$, then $sk_{i,1} = f^a \cdot y^{a'_{i,1}}$ and $pk_{i,1} = g^{a'_{i,1}}$. Therefore, sk_i and pk_i is a valid private/public key pair.

  If $a_{i,1} - \frac{a}{\omega} = 0 \bmod q$, the above procedure cannot occur and aborts. Otherwise, a private/public key pair is outputted to $\mathcal{A}$.
- ***Atomic-Sign***(): Given the public parameters MRK, the public key list PK_List and the message $\mathfrak{M}$, where PK_List is a list of the public keys of the devices involved with this query (with respect to the device i), the algorithm finishes the following steps:
  - The algorithm randomly chooses $sk_{i,0}, z_i \in \mathbb{Z}_q$ and a vector $(r_{i,k})$ with $r_{i,k} \in \mathbb{Z}_q$ and $k \in [1, 2.....l]$, computes $x_{i,0,k} = g^{\frac{sk_{i,0}}{l}} \cdot \varpi^{r_{i,k}}$ and $x_{i,1,k} = g^{r_{i,k}}$ with $k \in [1, 2.....l]$, and then saves $sk_{i,0}$ where $sk_{i,0} = a_{i,0}$;
  - The message $\mathfrak{M}$ is divided to $msg_1, msg_2, ...msg_l$; then the algorithm computes $v_k = H(msg_k||pk_k)$ with $k \in [1, 2.....l]$, where we assume the signing needs to involve l devices, pk_k is the public key of the k-th device with $pk_k \in PK\_List$;
  - For $V = (v_1, v_2, ...v_l)$, generate the claim-predicate Υ which satisfies $\Upsilon(V) = 1$, and then transform the claim-predicate Υ to the monotone span program $\Lambda \in (\mathbb{Z}_q)^{l \times t_{max}}$;
  - Compute $s_{k,j} = \psi^{r_{i,k}} \cdot (u_j)^{v_k \cdot r_{i,k}}$ with $k \in [1, 2.....l]$ and $j \in [1, 2.....t_{max}]$;
  - The algorithm outputs the atomic signatures $\sigma_a = \left\{(x_{i,0,k}), (x_{i,1,k}), (s_{k,j})\right\}$ to $\mathcal{A}$, where $k \in [1, 2.....l]$ and $j \in [1, 2.....t_{max}]$.

  If $v_k = H(msg_k||pk_k) = 0 \bmod q$ with $k \in [1, 2.....l]$, the above procedure cannot be occur and will abort; otherwise the atomic signatures are passed to the adversary $\mathcal{A}$.

- ***Mesh-Sign***(): Given the public parameters MRK, the atomic signatures $\sigma_a = \left\{(x_{i,0,k}), (x_{i,1,k}), (s_{k,j})\right\}$ on the public key list PK_List and the message $\mathfrak{M}$ (with respect to the device i), and the monotone boolean expression Y, the algorithm finishes the following steps:
  - Choose $b, c, t, d_0, d_1, ...d_l, a_{i,1} \in \mathbb{Z}_q$ randomly, compute $X_0 = \prod_{k=1}^{l}(x_{i,0,k}) \cdot \vartheta^{(d_0+b)\cdot H(\mathfrak{M})} \cdot \psi^{d_0+b} \cdot \varpi^{d_0}$, $X_1 = g^{d_0+b}$, $X_2 = g^{d_0} \cdot \prod_{k=1}^{l}(x_{i,1,k}) = g^{d_0+\sum_{k=1}^{l} r_{i,k}}$, $X_3 = g^{a_{i,1}} \cdot g_1^{-\frac{1}{\omega}} \cdot g^c$, $X_4 = y^{sk_{i,0}} \cdot y^c$ according to the corresponding $sk_{i,0}$, $X_{5,k} = x_{i,1,k} \cdot g^{d_0+t} = g^{r_{i,k}+d_0+t}$ with $k \in [1, 2.....l]$;
  - Compute the vector $\overrightarrow{\eta} = (\eta_k)$ related to the satisfying assignment $V = (v_1, v_2, ...v_l)$, where $\eta_k \in \mathbb{Z}_q$ with $k \in [1, 2.....l]$;
  - Compute $I_k = g^{d_k} \cdot (X_1)^{\frac{\eta_k}{v_k}}$ with $k \in [1, 2.....l]$, $Q_j = y^{a_{i,1}} \cdot y^c \cdot g^c \cdot X_0 \cdot \prod_{k=1}^{l} \psi^{(d_0+t)\cdot\Lambda_{k,j}} \cdot \prod_{k=1}^{l}\left[(s_{k,j})^{\Lambda_{k,j}} \cdot (u_j)^{(d_0+t)\cdot\Lambda_{k,j}\cdot v_k} \cdot (u_j)^{d_k\cdot\Lambda_{k,j}\cdot v_k}\right]$ with $j \in [1, 2..... .t_{max}]$;
  - The algorithm finally generates and outputs a mesh signature

$$\sigma_m = \{X_1, X_2, X_3, X_4, (X_{5,1}, ...X_{5,l}), (I_1, ...I_l), (Q_1, ...Q_{t_{max}})\}.$$

  Similarly, setting $a'_{i,1} = a_{i,1} - \frac{a}{\omega}$, σ_m is a valid mesh signature. If $a_{i,1} - \frac{a}{\omega} = 0 \bmod q$, the above procedure cannot occur and aborts. Otherwise, a mesh signature σ_m is passed to $\mathcal{A}$.

5.Guess: If $\mathcal{B}$ finally does not abort, then the adversary $\mathcal{A}$ can output its result $x' \in \{0, 1\}$ with probability at least ε and succeeds if $x' = x$. Then we may get the following:

$$\Phi^* = \{X_1^*, X_2^*, X_3^*, X_4^*, \left(X_{5,1}^*, X_{5,2}^*, ...X_{5,l}^*\right), (I_1{}^*, I_2{}^*, ...I_l{}^*), \left(Q_1{}^*, Q_2{}^*, ...Q_{t_{max}}^*\right)\},$$

where $Y^*(V^*) = 1$, and

$$\begin{aligned}
&X_1^* = g^{d_0^*+b^*},\\
&X_2^* = g^{d_0^*+\sum_{k=1}^{l} r_{i,k}^*},\\
&X_3^* = g^{a_{i,1}^*+c^*},\\
&X_4^* = y^{a_{i,0}^*+c^*},\\
&X_{5,k}^* = g^{r_{i,k}^*+d_0^*+t^*},\\
&I_k^* = g^{d_k^*} \cdot (X_1^*)^{\frac{\eta_k^*}{v_k^*}},\\
&Q_j^* = f^a \cdot y^{a_{i,1}^*+c^*} \cdot g^{c^*+a_{i,0}^*} \cdot \vartheta^{(d_0^*+b^*)\cdot H(\mathfrak{M}^*)} \cdot \psi^{d_0^*+b^*} \cdot \varpi^{d_0^*+\sum_{k=1}^{l} r_{i,k}^*} \cdot \prod_{k=1}^{l} \psi^{(d_0^*+t+r_{i,k}^*)\cdot\Lambda_{k,j}^*} \cdot\\
&\prod_{k=1}^{l}(u_j)^{(d_0^*+t^*+r_{i,k}^*)\cdot\Lambda_{k,j}^*\cdot v_k^*} \cdot \prod_{k=1}^{l}(u_j)^{d_k^*\cdot\Lambda_{k,j}^*\cdot v_k^*}.
\end{aligned}$$

Therefore, the algorithm $\mathcal{B}$ computes and outputs

$$\frac{Q_j^*}{(X_3^*)^{\beta}\cdot(X_4^*)^{\frac{1}{\beta}}\cdot(X_1^*)^{\iota\cdot H(\mathfrak{M}^*)}\cdot(X_1^*)^{\varphi}\cdot(X_2^*)^{\varrho}\cdot\prod_{k=1}^{l}(X_{5,k}^*)^{\varphi\cdot\Lambda_{k,j}^*}\cdot\prod_{k=1}^{l}(X_{5,k}^*)^{\ell_j\cdot\Lambda_{k,j}^*\cdot v_k^*}\cdot\prod_{k=1}^{l} g^{\ell_j\cdot d_k^*\cdot\Lambda_{k,j}^*\cdot v_k^*}} = f^a = g^{a\cdot b},$$

which solves the given CDH problem.

Then we compute the probability that $\mathcal{B}$ does not abort. For the complete simulation procedure of $\mathcal{B}$, we must assure that all key queries can have $a_{i,1} - \frac{a}{\omega} \neq 0 \bmod q$ in related queries Phases 1 and 2, all atomic signature queries can have $v_k = H(msg_k||pk_k) \neq 0 \bmod q$ for all $k \in [1, 2.....l]$ in related Queries Phases 1 and 2, and all mesh signature queries can have $a_{i,1} - \frac{a}{\omega} \neq 0 \bmod q$ in related Queries Phases 1 and 2. Therefore, if $\mathcal{B}$ will not abort, then we must assure that the following three conditions hold:

(a) $a_{i,1} - \frac{a}{\omega} \neq 0 \bmod q$ in the related key queries of Queries Phases 1 and 2;
(b) $v_k = H(msg_k||pk_k) \neq 0 \bmod q$ for all $k \in [1, 2.....l]$ in the related atomic signature queries of Queries Phases 1 and 2;
(c) $a_{i,1} - \frac{a}{\omega} \neq 0 \bmod q$ in the related mesh signature queries of Queries Phases 1 and 2.

To make our analysis easier to understand, we define the following events E_{j_1}, R_{j_1}, T_{j_1}, E_{j_2}, R_{j_2} and T_{j_2} as

E_{j_1} :$a_{i,1} - \frac{a}{\omega} \neq 0 \bmod q$, with j_1=1, 2.....q_{k_1}, q_{k_1} denotes the queries number of "Generate-Key" oracle in related Queries Phase 1;

R_{j_1} :$v_k = H(msg_k||pk_k) \neq 0 \bmod q$ for all $k \in [1, 2.....l]$, with j_1=1, 2.....q_{a_1}, q_{a_1} denotes the queries number of "Atomic Signature" oracle in related Queries Phase 1;

T_{j_1} :$a_{i,1} - \frac{a}{\omega} \neq 0 \bmod q$, with j_1=1, 2.....q_{m_1}, q_{m_1} denotes the queries number of "Mesh Signature" oracle in related Queries Phase 1;

E_{j_2} :$a_{i,1} - \frac{a}{\omega} \neq 0 \bmod q$, with j_2=1, 2.....q_{k_2}, q_{k_2} denotes the queries number of "Generate-Key" oracle in related Queries Phase 2;

R_{j_2} :$v_k = H(msg_k||pk_k) \neq 0 \bmod q$ for all $k \in [1, 2.....l]$, with j_2=1, 2.....q_{a_2}, q_{a_2} denotes the queries number of "Atomic Signature" oracle in related Queries Phase 2;

T_{j_2} :$a_{i,1} - \frac{a}{\omega} \neq 0 \bmod q$, with j_2=1, 2.....q_{m_2}, q_{m_2} denotes the queries number of "Mesh Signature" oracle in related Queries Phase 2.

Therefore, the probability that $\mathcal{B}$ is completely simulated is

$$\Pr(not\_abort) = \Pr\left(\bigcap_{j_1=1}^{q_{k_1}} E_{j_1} \wedge \bigcap_{j_1=1}^{q_{a_1}} R_{j_1} \wedge \bigcap_{j_1=1}^{q_{m_1}} T_{j_1} \wedge \bigcap_{j_2=1}^{q_{k_2}} E_{j_2} \wedge \bigcap_{j_2=1}^{q_{a_2}} R_{j_2} \wedge \bigcap_{j_2=1}^{q_{m_2}} T_{j_2}\right).$$

It is easy to see that the events $\bigcap_{j_1=1}^{q_{k_1}} E_{j_1}$, $\bigcap_{j_1=1}^{q_{a_1}} R_{j_1}$, $\bigcap_{j_1=1}^{q_{m_1}} T_{j_1}$, $\bigcap_{j_2=1}^{q_{k_2}} E_{j_2}$, $\bigcap_{j_2=1}^{q_{a_2}} R_{j_2}$ and $\bigcap_{j_2=1}^{q_{m_2}} T_{j_2}$ are independent. Then we may compute

$$\Pr(\bigcap_{j_1=1}^{q_{k_1}} E_{j_1}) = 1 - \Pr(\bigcup_{j_1=1}^{q_{k_1}} \neg E_{j_1}) = 1 - q_{k_1} \cdot \frac{1}{q} = 1 - \frac{q_{k_1}}{q};$$

$$\Pr(\bigcap_{j_1=1}^{q_{a_1}} R_{j_1}) = 1 - \Pr(\bigcup_{j_1=1}^{q_{a_1}} \neg R_{j_1}) = 1 - q_{a_1} \cdot [1 - (1 - \frac{1^k}{1^k \cdot q})^l] = 1 - q_{a_1} + q_{a_1} \cdot (1 - \frac{1}{q})^l;$$

$$\Pr(\bigcap_{j_1=1}^{q_{m_1}} T_{j_1}) = 1 - \Pr(\bigcup_{j_1=1}^{q_{m_1}} \neg T_{j_1}) = 1 - q_{m_1} \cdot \frac{1}{q} = 1 - \frac{q_{m_1}}{q};$$

$$\Pr(\bigcap_{j_2=1}^{q_{k_2}} E_{j_2}) = 1 - \Pr(\bigcup_{j_2=1}^{q_{k_2}} \neg E_{j_2}) = 1 - q_{k_2} \cdot \frac{1}{q} = 1 - \frac{q_{k_2}}{q};$$

$$\Pr(\bigcap_{j_2=1}^{q_{a_2}} R_{j_2}) = 1 - \Pr(\bigcup_{j_2=1}^{q_{a_2}} \neg R_{j_2}) = 1 - q_{a_2} \cdot [1 - (1 - \frac{1^k}{1^k \cdot q})^l] = 1 - q_{a_2} + q_{a_2} \cdot (1 - \frac{1}{q})^l;$$

$$\Pr(\bigcap_{j_2=1}^{q_{m_2}} T_{j_2}) = 1 - \Pr(\bigcup_{j_2=1}^{q_{m_2}} \neg T_{j_2}) = 1 - q_{m_2} \cdot \frac{1}{q} = 1 - \frac{q_{m_2}}{q}.$$

Therefore,

$$\begin{aligned}\Pr(not\_abort) &= \Pr\left(\bigcap_{j_1=1}^{q_{k_1}} E_{j_1} \wedge \bigcap_{j_1=1}^{q_{a_1}} R_{j_1} \wedge \bigcap_{j_1=1}^{q_{m_1}} T_{j_1} \wedge \bigcap_{j_2=1}^{q_{k_2}} E_{j_2} \wedge \bigcap_{j_2=1}^{q_{a_2}} R_{j_2} \wedge \bigcap_{j_2=1}^{q_{m_2}} T_{j_2}\right) \\ &= (1 - \tfrac{q_{k_1}}{q}) \cdot (1 - \tfrac{q_{k_2}}{q}) \cdot [1 - q_{a_1} + q_{a_1} \cdot (1 - \tfrac{1}{q})^l] \cdot [1 - q_{a_2} + q_{a_2} \cdot (1 - \tfrac{1}{q})^l] \cdot (1 - \tfrac{q_{m_1}}{q}) \cdot (1 - \tfrac{q_{m_2}}{q}).\end{aligned}$$

Therefore, we can get that $\varepsilon' = (1 - \frac{q_{k_1}}{q}) \cdot (1 - \frac{q_{k_2}}{q}) \cdot [1 - q_{a_1} + q_{a_1} \cdot (1 - \frac{1}{q})^l] \cdot [1 - q_{a_2} + q_{a_2} \cdot (1 - \frac{1}{q})^l] \cdot (1 - \frac{q_{m_1}}{q}) \cdot (1 - \frac{q_{m_2}}{q}) \cdot (\varepsilon - \frac{1}{2})$.

If $\mathcal{B}$ is completely simulated, then $\mathcal{A}$ generates a valid mesh signature forgery with probability at least ε, and $\mathcal{B}$ may be used to compute $g^{a \cdot b}$. The time cost of $\mathcal{B}$ mainly includes the time of

the exponentiations and multiplications in queries. We assume that the time of other lightweight computations is ignored, then the time cost of $\mathcal{B}$ is

$$\hbar' = \hbar + O((q_{k_1} + q_{k_2}) \cdot [3 \cdot C_{exp} + C_{mul}] + (q_{a_1} + q_{a_2}) \cdot [(2 \cdot l \cdot t_{max} + 3) \cdot C_{exp} + (l \cdot t_{max} + 1) \cdot C_{mul}] + (q_{m_1} + q_{m_2}) \cdot [(4 \cdot l \cdot t_{max} + 3 \cdot l + 13) \cdot C_{exp} + (4 \cdot l \cdot t_{max} + 4 \cdot l + 8) \cdot C_{mul}]).$$

Thus, Theorem 6.2 follows.

Appendix A.3 Correctness

In the proposed scheme, the mesh signature is

$$\Phi = \{X_1, X_2, X_3, X_4, (X_{5,1}, ...X_{5,l}), (I_1, ...I_l), (Q_1, ...Q_{t_{max}})\},$$

where

$$\begin{aligned}
X_0 &= \prod_{k=1}^{l}(x_{i,0,k}) \cdot \vartheta^{(d_0+b)\cdot H(\mathfrak{M})} \cdot \psi^{d_0+b} \cdot \varpi^{d_0} \\
&= \prod_{k=1}^{l}(g^{\frac{sk_{i,0}}{l}} \cdot \varpi^{r_{i,k}}) \cdot \vartheta^{(d_0+b)\cdot H(\mathfrak{M})} \cdot \psi^{d_0+b} \cdot \varpi^{d_0} \\
&= g^{sk_{i,0}} \cdot \varpi^{\sum_{k=1}^{l} r_{i,k}} \cdot \vartheta^{(d_0+b)\cdot H(\mathfrak{M})} \cdot \psi^{d_0+b} \cdot \varpi^{d_0} \\
&= g^{a_{i,0}} \cdot \vartheta^{(d_0+b)\cdot H(\mathfrak{M})} \cdot \psi^{d_0+b} \cdot \varpi^{d_0+\sum_{k=1}^{l} r_{i,k}}, \\
X_1 &= g^{d_0+b}, \\
X_2 &= g^{d_0} \cdot \prod_{k=1}^{l}(x_{i,1,k}) = g^{d_0+\sum_{k=1}^{l} r_{i,k}}, \\
X_3 &= pk_{i,1} \cdot g^{c}, \\
X_4 &= y^{a_{i,0}} \cdot y^{c}, \\
X_{5,k} &= x_{i,1,k} \cdot g^{d_0+t} = g^{r_{i,k}+d_0+t}, \\
I_k &= g^{d_k} \cdot (X_1)^{\frac{\eta_k}{v_k}}, \\
Q_j &= sk_{i,1} \cdot y^{c} \cdot g^{c} \cdot X_0 \cdot \prod_{k=1}^{l} \psi^{(d_0+t)\cdot\Lambda_{k,j}} \cdot \prod_{k=1}^{l}[(s_{k,j})^{\Lambda_{k,j}} \cdot (u_j)^{(d_0+t)\cdot\Lambda_{k,j}\cdot v_k} \cdot (u_j)^{d_k\cdot\Lambda_{k,j}\cdot v_k}] \\
&= f^{a} \cdot y^{a_{i,1}} \cdot y^{c} \cdot g^{c} \cdot g^{a_{i,0}} \cdot \vartheta^{(d_0+b)\cdot H(\mathfrak{M})} \cdot \psi^{d_0+b} \cdot \varpi^{d_0+\sum_{k=1}^{l} r_{i,k}} \cdot \prod_{k=1}^{l} \psi^{(d_0+t)\cdot\Lambda_{k,j}} \cdot \prod_{k=1}^{l}[(s_{k,j})^{\Lambda_{k,j}} \cdot \\
&(u_j)^{(d_0+t)\cdot\Lambda_{k,j}\cdot v_k} \cdot (u_j)^{d_k\cdot\Lambda_{k,j}\cdot v_k}] \\
&= f^{a} \cdot y^{a_{i,1}+c} \cdot g^{c+a_{i,0}} \cdot \vartheta^{(d_0+b)\cdot H(\mathfrak{M})} \cdot \psi^{d_0+b} \cdot \varpi^{d_0+\sum_{k=1}^{l} r_{i,k}} \cdot \prod_{k=1}^{l} \psi^{(d_0+t)\cdot\Lambda_{k,j}} \cdot \prod_{k=1}^{l}[(\psi^{r_{i,k}} \cdot \\
&(u_j^{v_k\cdot r_{i,k}}))^{\Lambda_{k,j}} \cdot (u_j)^{(d_0+t)\cdot\Lambda_{k,j}\cdot v_k} \cdot (u_j)^{d_k\cdot\Lambda_{k,j}\cdot v_k}] \\
&= f^{a} \cdot y^{a_{i,1}+c} \cdot g^{c+a_{i,0}} \cdot \vartheta^{(d_0+b)\cdot H(\mathfrak{M})} \cdot \psi^{d_0+b} \cdot \varpi^{d_0+\sum_{k=1}^{l} r_{i,k}} \cdot \prod_{k=1}^{l} \psi^{(d_0+t)\cdot\Lambda_{k,j}} \cdot \prod_{k=1}^{l} \psi^{r_{i,k}\cdot\Lambda_{k,j}} \cdot \\
&\prod_{k=1}^{l}(u_j)^{v_k\cdot r_{i,k}\cdot\Lambda_{k,j}} \cdot \prod_{k=1}^{l}(u_j)^{(d_0+t)\cdot\Lambda_{k,j}\cdot v_k} \cdot \prod_{k=1}^{l}(u_j)^{d_k\cdot\Lambda_{k,j}\cdot v_k} \\
&= f^{a} \cdot y^{a_{i,1}+c} \cdot g^{c+a_{i,0}} \cdot \vartheta^{(d_0+b)\cdot H(\mathfrak{M})} \cdot \psi^{d_0+b} \cdot \varpi^{d_0+\sum_{k=1}^{l} r_{i,k}} \cdot \prod_{k=1}^{l} \psi^{(d_0+t+r_{i,k})\cdot\Lambda_{k,j}} \cdot \\
&\prod_{k=1}^{l}(u_j)^{(d_0+t+r_{i,k})\cdot\Lambda_{k,j}\cdot v_k} \cdot \prod_{k=1}^{l}(u_j)^{d_k\cdot\Lambda_{k,j}\cdot v_k}.
\end{aligned}$$

Therefore, we have that

$$\begin{aligned}
&e(f, g_1) \cdot e(y, X_3) \cdot e(g, X_4) \cdot e\left(\vartheta^{H(\mathfrak{M})} \cdot \psi, X_1\right) \cdot e(\varpi, X_2) \cdot \prod_{k=1}^{l} e(\psi^{\Lambda_{k,j}} \cdot u_j^{v_k\cdot\Lambda_{k,j}}, X_{5,k}) \cdot \prod_{k=1}^{l} e(u_j^{v_k\cdot\Lambda_{k,j}}, I_k) \\
&= e(f, g^{a}) \cdot e\left(y, g^{a_{i,1}+c}\right) \cdot e\left(g, y^{c+a_{i,0}}\right) \cdot e\left(\vartheta^{H(\mathfrak{M})} \cdot \psi, g^{d_0+b}\right) \cdot e\left(\varpi, g^{d_0+\sum_{k=1}^{l} r_{i,k}}\right) \cdot \prod_{k=1}^{l} e(\psi^{\Lambda_{k,j}} \cdot u_j^{v_k\cdot\Lambda_{k,j}}, g^{r_{i,k}+d_0+t}) \cdot \prod_{k=1}^{l} e(u_j^{v_k\cdot\Lambda_{k,j}}, g^{d_k} \cdot (X_1)^{\frac{\eta_k}{v_k}}) \\
&= e\left(f^{a} \cdot y^{a_{i,1}+c} \cdot g^{c+a_{i,0}} \cdot \vartheta^{(d_0+b)\cdot H(\mathfrak{M})} \cdot \psi^{d_0+b} \cdot \varpi^{d_0+\sum_{k=1}^{l} r_{i,k}} \cdot \prod_{k=1}^{l} \psi^{(r_{i,k}+d_0+t)\cdot\Lambda_{k,j}} \cdot \prod_{k=1}^{l}(u_j)^{(r_{i,k}+d_0+t)\cdot\Lambda_{k,j}\cdot v_k} \cdot \prod_{k=1}^{l}(u_j)^{d_k\cdot\Lambda_{k,j}\cdot v_k}, g\right) \cdot \prod_{k=1}^{l} e(u_j^{\eta_k\cdot\Lambda_{k,j}}, X_1) \\
&= e\left(Q_j, g\right) \cdot \prod_{k=1}^{l} e(u_j^{\eta_k\cdot\Lambda_{k,j}}, X_1),
\end{aligned}$$

when $j = 1$, we have that $\sum_{k=1}^{l} \Lambda_{k,j} \cdot \eta_k = 1$, so

$$\begin{aligned}
&e(f, g_1) \cdot e(y, X_3) \cdot e(g, X_4) \cdot e\left(\vartheta^{H(\mathfrak{M})} \cdot \psi, X_1\right) \cdot e(\varpi, X_2) \cdot \prod_{k=1}^{l} e(\psi^{\Lambda_{k,j}} \cdot u_j^{v_k\cdot\Lambda_{k,j}}, X_{5,k}) \cdot \prod_{k=1}^{l} e(u_j^{v_k\cdot\Lambda_{k,j}}, I_k) \\
&= e(Q_j, g) \cdot e(u_j, X_1);
\end{aligned}$$

when $j > 1$, we have that $\sum_{k=1}^{l} \Lambda_{k,j} \cdot \eta_k = 0$, so

$$\begin{aligned}
&e(f, g_1) \cdot e(y, X_3) \cdot e(g, X_4) \cdot e\left(\vartheta^{H(\mathfrak{M})} \cdot \psi, X_1\right) \cdot e(\varpi, X_2) \cdot \prod_{k=1}^{l} e(\psi^{\Lambda_{k,j}} \cdot u_j^{v_k\cdot\Lambda_{k,j}}, X_{5,k}) \cdot \prod_{k=1}^{l} e(u_j^{v_k\cdot\Lambda_{k,j}}, I_k) \\
&= e(Q_j, g).
\end{aligned}$$

References

1. Karati, A.; Islam, S.H.; Karuppiah, M. Provably Secure and Lightweight Certificateless Signature Scheme for IIoT Environments. *IEEE Trans. Ind. Inf.* **2018**, *14*, 3701–3711. doi: 10.1109/TII.2018.2794991.
2. Boyen, X. Mesh Signatures-How to Leak a Secret with Unwitting and Unwilling Participants. , Advances in Cryptology—EUROCRYPT 2007; Springer-Verlag: Berlin/Heidelberg, Germany, 2007.
3. Rivest, R.; Shamir, A.; Tauman, Y. How to leak a secret. In C. Boyd, editor, Asiacrypt 2001, LNCS 2248, Springer: Berlin/Heidelberg, Germany, 2001; pp. 552–565.
4. Boyen, X. Unconditionally Anonymous Ring and Mesh Signatures. *J. Cryptology* **2016**, *29*, 729–774.
5. Maji, H.K.; Prabhakaran, M.; Rosulek, M. Attribute-Based Signatures, Topics in Cryptology-CT-RSA 2011, LNCS 6558, Springer-Verlag, 2011, pp. 376-392.
6. Chaum, D.; van Heyst, E. Group Signatures. In Eurocrypt'91, LNCS 547, pp. 257-265, Springer, 1991.
7. Liu, J.K.; Wei, V.K.; Wong, D.S. Linkable spontaneous anonymous group signature for ad hoc groups. In *ACISP 2004: Information Security and Privacy*; Springer: Berlin/Heidelberg, Germany, 2004; 325–335.
8. Chow, S.S.M.; Liu, J.K.; Wong, D. S. Robust receipt-free election system with ballot secrecy and verifieability. *NDSS* **2008**, *8*, 81–94.
9. Tsang, P.P.; Wei, V.K. Short linkable ring signatures for e-voting, e-cash and attestation. In *ISPEC 2005: Information Security Practice and Experience*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 48–60.
10. Susilo, W.; Mu, Y. Non-Interactive Deniable Ring Authentication. In *ICISC 2003: Information Security and Cryptology - ICISC 2003*; Springer: Berlin/Heidelberg, Germany, 2004; pp. 386–401.
11. Laguillaumie, F.; Vergnaud, D. Multi-designated Verifiers Signatures. In *ICICS 2004, volume 3269 of Lecture Notes in Computer Science*Springer: Berlin/Heidelberg, Germany, 2004; pp. 495–507.
12. Gu, K.; Wu, N.; Yin, B.; Jia, W. Secure Data Query Framework for Cloud and Fog Computing. *IEEE Trans. Netw. Serv. Manage.* **2019**, doi: 10.1109/TNSM.2019.2941869.
13. Gu, K.; Wu, N.; Yin, B.; Jia, W. Secure Data Sequence Query Framework Based on Multiple Fogs. *IEEE Trans. Netw. Serv. Manage.* **2019**, doi: 10.1109/TETC.2019.2943524.
14. Gu, K.; Wang, K.; Yang, L. Traceable Attribute-Based Signature. *J. Inf. Secur. Appl.* **2019**, *49*, 102400.
15. Gu, K.; Dong, X.; Wang, L. Efficient Traceable Ring Signature Scheme without Pairings. *Adv. Math. Commun.* **2019**, doi:10.3934/amc.2020016.
16. Yu, F;. Liu, L.; Xiao, L.; Li, K.; Cai, S. A robust and fixed-time zeroing neural dynamics for computing time-variant nonlinear equation using a novel nonlinear activation function. *Neurocomputing* **2019**, *350*, 108–116.
17. Yu, F.; Liu, L.; He, B.; Huang, Y.; Shi, C.; Cai, S.; Song, Y.; Du, S.; Wan, Q. Analysis and FPGA Realization of a Novel 5D Hyperchaotic Four-Wing Memristive System, Active Control Synchronization, and Secure Communication Application. *Complexity* **2019**, *2019*, 4047957.
18. Yu, F.; Zhang, Z.; Liu, L.; Shen, H.; Huang, H.; Shi, C.; Cai, S.; Song, Y.; Du, S.; Xu, Q. Secure communication scheme based on a new 5D multistable four-wing memristive hyperchaotic system with disturbance inputs. *Complexity* **2020**, *2020*, 5859273.
19. Chen, Y.; Wang, J.; Xia, R.; Zhang, Q.; Cao, Z.; Yang, K. The visual object tracking algorithm research based on adaptive combination kernel. *J. Ambient Intell. Humanized Comput.* **2019**, *10*, 4855–4867.
20. Li, W.; Chen, Z.; Gao, X.; Liu, W.; Wang, J. Multi-Model Framework for Indoor Localization under Mobile Edge Computing Environment. *IEEE Internet of Things J.* **2019**, *6*, 4844–4853.
21. Li, Y.; Zhu, T. Gait-Based Wi-Fi Signatures for Privacy-Preserving. In Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security (ASIA CCS '16), Xi'an, China; pp. 571–582, doi: 10.1145/2897845.2897909
22. Sun, J.; Su, Y.; Qin, J.; Hu, J.; Ma, J. Outsourced Decentralized Multi-authority Attribute Based Signature and Its Application in IoT. *IEEE Trans. Cloud Comput.* **2019**, doi: 10.1109/TCC.2019.2902380
23. Xie, R.; He, C.; Xu, C.; Gao C. Lattice-based dynamic group signature for anonymous authentication in IoT. *Ann. Telecommun.* **2019**, *74*, 531–542. doi:10.1007/s12243-019-00705-x.
24. Mughal, M.A.; Luo, X.; Ullah, A.; Ullah, S.; Mahmood, Z. A Lightweight Digital Signature Based Security Scheme for Human-Centered Internet of Things. *IEEE Access* **2018**, *6*, 31630–31643. doi: 10.1109/ACCESS.2018.2844406.

25. Cui, H.; Deng, R.H.; Liu, R.H.; Yi, X.; Li, Y. Server-Aided Attribute-Based Signature With Revocation for Resource-Constrained Industrial-Internet-of-Things Devices. *IEEE Trans. Ind. Inf.* **2018**, *14*, 3724–3732. doi: 10.1109/TII.2018.2813304
26. Li, F.; Zheng, Z.; Jin, C. Secure and efficient data transmission in the Internet of Things. *Telecommun. Syst.* **2016**, *62*, 111–122; doi:10.1007/s11235-015-0065-y.
27. Libert, B.; Peters, T.; Yung, M. Scalable Group Signatures with Revocation. In *Advances in Cryptology-EUROCRYPT 2012*; Springer-Verlag: Berlin/Heidelberg, Germany, 2012; pp. 609–627.
28. Libert, B.; Peters, T.; Yung, M. Scalable Group Signatures with Almost-for-Free Revocation. In *Advances in Cryptology-CRYPTO2012*; Springer-Verlag: Berlin/Heidelberg, Germany, 2012; pp. 571-589.
29. Ibraimi, L.; Nikova, S.; Hartel, S.; Jonker, W. An Identity-Based Group Signature with Membership Revocation in the Standard Model. Available online: http:/doc.utwente.nl/72270/1/Paper.pdf (accessed on 28 January 2020).
30. Emura, K.; Miyaji, A.; Omote, K. An *r*-Hiding Revocable Group Signature Scheme: Group Signatures with the Property of Hiding the Number of Revoked Users. *Eur. J. Appl. Math.* **2014**, *2014*, 983040.
31. Gu, K.; Yang, L.; Wang, Y.; Wen, S. Traceable Identity-Based Group Signature. *RAIRO-Theor. Inf. Appl.* **2016**, *50*, 193–226.
32. Yuen, T.H.; Liu, J.K.; Au, M.H.; Susilo, W.; Zhou, J. Efficient linkable and/or threshold ring signature without random oracles. *Comput. J.* **2013**, *56*, 407-–421.
33. Liu, J.K.; Au, M.H.; Susilo, W.; Zhou, J. Linkable Ring Signature with Unconditional Anonymity. *IEEE Trans. Knowl. Data Eng.* **2014**, *26*, 157—165.
34. Au, M.H.; Liu, J.K.; Susilo, W.; Yuen, T.H. Secure ID-Based Linkable and Revocable-iff-Linked Ring Signature with Constant-Size Construction. *Theor. Comput. Sci.* **2013**, *469*, 1–14.
35. Kaafarani, A.E.; Ghadafi, E.; Khader, D. Decentralized Traceable Attribute-Based Signatures. In *Topics in Cryptology – CT-RSA 2014*; Springer-Verlag: Berlin/Heidelberg, Germany, 2014; pp.327–348.
36. Ghadafi, E. Stronger Security Notions for Decentralized Traceable Attribute-Based Signatures and More Efficient Constructions. In *Topics in Cryptology — CT-RSA 2015.*; Springer-Verlag: Berlin/Heidelberg, Germany, 2015; pp.391–409.
37. Gu, K.; Jia, W.; Wang, G.; Wen, S. Efficient and secure attribute-based signature for monotone predicates. *Acta Inf.* **2017**, *54*, 521–541.
38. Song, T.; Li, R.; Mei, B.; Yu, J.; Xing, X.; Cheng, X. A privacy preserving communication protocol for IoT applications in smart homes. *IEEE Internet of Things J.* **2017**, *4*, 1844–1852.
39. Dwivedi, A.D.; Srivastava, G.; Dhar, G.; Singh, R. A decentralized privacy-preserving healthcare blockchain for IoT. *Sensors* **2019**, *19*, 326.
40. Sharma, S.; Chen, K.; Sheth, K. Toward practical privacy-preserving analytics for IoT and cloud-based healthcare systems. *IEEE Internet Comput.* **2018**, *22*, 42–51.
41. Zhou, J.; Cao, Z.; Dong, X.; Vasilakos, A. Security and privacy for cloud-based IoT: Challenges. *IEEE Commun. Mag.* **2017**, *55*, 26–33.
42. Gope, P.; Sikdar, B. Lightweight and privacy-preserving two-factor authentication scheme for IoT devices. *IEEE Internet of Things J.* **2018**, *6*, 580–589.
43. Li, X.; Liu, S.; Wu, F.; Kumari, S; Rodrigues, J. Privacy preserving data aggregation scheme for mobile edge computing assisted IoT applications. *IEEE Internet of Things J.* **2018**, *6*, 4755–4763.
44. Shen, M.; Tang, X.; Zhu, L.; Du, X.; Guizani, M. Privacy-preserving support vector machine training over blockchain-based encrypted IoT data in smart cities. *IEEE Internet of Things J.* **2019**, *6*, 7702–7712.
45. Lu, R. A new communication-efficient privacy-preserving range query scheme in fog-enhanced IoT. *IEEE Internet of Things J.* **2018**, *6*, 2497–2505.
46. Huang, P.; Guo, P.; Li, M.; Fang, Y. Practical Privacy-preserving ECG-based Authentication for IoT-based Healthcare. *IEEE Internet of Things J.* **2019**, *6*, 9200–9210.
47. Jiang, L.; Chen, L.; Giannetsos, T.; Luo, B.; Liang, K.; Han, J. Toward Practical Privacy-Preserving Processing Over Encrypted Data in IoT: An Assistive Healthcare Use Case. *IEEE Internet of Things J.* **2019**, *6*, 10177–10190.
48. Ma, Z.; Liu, Z.; Liu, X.; Ma, J.; Li, F. Privacy-Preserving Outsourced Speech Recognition for Smart IoT Devices. *IEEE Internet of Things J.* **2019**, *6*, 8406–8420.
49. Zhao, Y.; Yang, L.T.; Sun, J. Privacy-Preserving Tensor-Based Multiple Clusterings on Cloud for Industrial IoT. *IEEE Trans. Ind. Inf.* **2018**, *15*, 2372–2381.

50. Gan, X.; Li, X.; Huang, Y.; Fu, L.; Wang, X. When Crowdsourcing Meets Social IoT: An Efficient Privacy-Preserving Incentive Mechanism. *IEEE Internet of Things J.* **2019**, *6*, 9707-9721.
51. Gochoo, M.; Tan, T.H.; Huang, S.C.; Batjargal, T.; Hsieh, J.; Alnajjar, F. S.; Chen Y. Novel IoT-Based Privacy-Preserving Yoga Posture Recognition System Using Low-Resolution Infrared Sensors and Deep Learning. *IEEE Internet of Things J.* **2019**, *6*, 7192–7200.
52. Xu, C.; Ren, J.; Zhang, D.; Zhangm Y. Distilling at the edge: A local differential privacy obfuscation framework for IoT data analytics. *IEEE Commun. Mag.* **2018**, *56*, 20–25.