Achieving trustworthiness, Transferable, Cross-domain Dynamic Accumulator Authentication in Industrial Internet of Things*

Linjie Wang[1,2], Youliang Tian[1], and Jinbo Xiong[3]
{youliangtian@163.com, wanglinjie_66@hotmail.com, jinbo810@163.com}

[1] State Key Laboratory of Public Big Data, College of Computer Science Technology, Guizhou University, Guiyang 550025, China
[2] School of Data Science, Tongren University, Tongren 554300, China
[3] College of Mathematics and Informatics, Fujian Normal University, Fuzhou 350007, China

Abstract. The authentication problem is one of the most significant challenges in the industrial Internet of Things control system applications. To authenticate the relationships between devices is an effective solution to tackle authentication issue. However, the previous studies of Internet of Things focused on connecting sensors and setting the sharing condition from either private or public in the sense that the relationship authentication issue between devices is not well solved. In this paper, we propose CDAA, a Cross-domain Dynamic Accumulator Authentication with a general undirected graph representing the relationship between devices needed authentication for industrial Internet of Things supporting trustworthiness cross-domain, cryptographic accumulator, and transferability. Specifically, a dynamically updatable cross-domain authentication scheme is proposed based on the cryptographic accumulator and a standard digital signature scheme as the underlay of the CDAA. Finally, we give the analysis and comparison and the results show that the proposed scheme can address the authentication issue. The effectiveness and feasibility of proposed scheme are presented and analyzed through a comparison with traditional systems in practical application.

Keywords: Cross-domain · Authentication · Undirected graph · Accumulator · Bilinear pairings.

* This work is supported by The National Natural Science Foundation of China under Grant Nos. 61662009 and 61772008; Science and Technology Major Support Program of Guizhou Province (No. 20183001); Ministry of Education-China Mobile Research Fund Project under Grant No. MCM20170401; Foundation of Postgraduate of Guizhou Province under Grant No. YJSCXJH2019101; Science and Technology Planning Project of Tongren Municipality under Grant No. 202078.

EAI MOBIMEDIA 2020, August 27-28, Harbin, People's Republic of China

DOI 10.4108/eai.27-8-2020.2294325

1 Introduction

To date, ubiquitous wireless sensor devices have made the Internet of Things (IoT) attracted enormous attention from academics and industries, which is foreseen one of the most important technologies of this century [1] [2]. However, the industrial Internet of Things (IIoT) offered interconnection and intelligence to industrial systems through sensing devices and actuators has the potential to change the world as the Internet did [3]. In practice IIoT system, for the purpose of protecting the legitimate access each device has its own unique identity and has the capability to sense, compute, authenticate and communicate [4] [5]. Many existing solutions have been introduced into security and privacy solutions in IIoT, *e.g.*, the data security, wireless power transfer, and the deployment and communication problems of devices [6] [7].

In a multi-layer distributed IIoT system, the following situations are often encountered: on the one hand the supervisor entity $\mathcal{A}$ wants to command the subordinate entity $\mathcal{B}$, $\mathcal{A}$ should provide a legal authoritative signature to show the legality of its command [4]; On the other hand, entity $\mathcal{A}$ wants to access subordinate entity $\mathcal{C}$ of entity $\mathcal{B}$ who is in the same regulatory authority as $\mathcal{A}$, $\mathcal{A}$ also should provide a legal authoritative signature to illustrate the legality of its command. In this paper, vertices and edges are used to represent company members and the authentication relationship between them, respectively. That is, given two vertices v_i and v_j do not have a signature edge, one can generates the legitimate signature edge (v_i, v_j) without the administrative involvement. This property can help the authority to mitigate the pressure of the digital signature authentication.

1.1 Related work

Since the first undirected transitive signature based on the discrete logarithm and RSA assumptions was proposed by Micali and Rivest [8], Shahandashti *et al.* [9] proposed a short transitive signature based on the bilinear group pairs. A new approach transforming state transitive signature scheme into the stateless ones without loss of security was introduced by Ma *et al.* [10], and the proposing three concrete protocols were based on hardness of Factoring and RSA problems. The security analysis showed the scheme secure against the adaptive chosen message attack in random oracle model. Subsequent to Micali and Rivest's undirected graph transitive signature work, Rivest [11] introduced a directed transitive signature scheme and showed it hard to construct based on a sophisticated mathematical group. Like Neven's [12] researches, Xu [13] also studied the directed trees, but his scheme is more consideration on the property of constant signature size and privacy preserving. Camacho *et al.* in [14] also introduced a transitive signature protocol for the undirected tree.

In the existing papers, literature [12] all have designed the transitive signature schemes for special directed graphs (*e.g.* directed tree) and the security proof under adaptive chosen-message attack. In addition to transitive signature studied, a rejectable signatures [13] were used to authenticate general directed

graphs. Followed the accumulator work by Ma *et al.* [10], Lin *et al.* [15] presented a novel transitively closed undirected graph authentication scheme to support blockchain-based identity management systems.

1.2 Contributions

In order to address the aforementioned authentication issue between IoT devices in IIOT system, we introduce a cross-domain dynamic accumulator authentication scheme (CDAA). The contributions are outlined as follows:

- In the undirected administrative domain graph, we leverage the transitivity of digital signature authentication to construct the digital authentication relation between the vertices of non-existent edge.
- In our scheme, a signer can directly compute the authentication to the verifier with the transitivity of the signature to prove its legality even when they are not in the same administrative domain.
- After updating the vertices' witness and signature, the signer only needs to publish and chain the new witness and signature, and the verifier can verify the correctness of the new witness and signature to decide whether to be signature authenticated.
- For the implementation, we give the theoretical analysis and compare CDAA with existing protocols(as shown in Table I and II). We also test the time cost on the public parameters, private key, accumulator update cost of CDAA and so on.

1.3 Organization

The rest of the paper is organized as follow. In Section II, we describes the relevant preliminaries. Section III discusses the system model. In Section IV, we provide the security models of CDAA. The CDAA scheme with its security analysis is proposed in Section V. Section VI presents the comparison of performance and simulation. Conclusion and further discussion are given in Section VII.

2 Preliminaries

In this section, we introduce some fundamental preliminaries required in this paper. $\mathbb{N} = \{1, 2, \cdots, N\}$ denotes a set of positive integers from 1 to N; Z_N^* is a multiplicative group of integers modulo N; $||$ represents the concatenation operator on strings; $f : \mathbb{N} \to \mathbb{R}$ is a negligible function; $\mathcal{PPT}$ is the abbreviation of probabilistic polynomial time and $\mathbb{S}$ shows that x is sampled randomly and uniformly from the set $\mathbb{S}$.

2.1 Graphs

In this paper, we consider $G = (V, E)$ as an undirected graph with a nonempty vertices set $V \subset \mathbb{N}$ and $E \subset V \times V$, a set of edges. Based on the definition of the partition, we split the vertices set V into several infinite sets $V = V_1 \cup V_2 \cup \cdots \cup V_u$ as Fig. 2 shows, where $u = |V|$.

2.2 Dynamic accumulator

Literature [11] generalized the definition of accumulator as follows. $X_\lambda\,(\lambda \in \mathbb{N})$ denotes the domain of values to be accumulated and F_λ is a set of pairs of functions, as well as U_f the accumulator domain. For each $(f, g) \in F_\lambda$, let $f : U_f \times X_f^{ext} \to U_f$ for some $X_f^{ext} \supseteq X_\lambda$ and $g : U_f \to U_h$ is a bijective function. For any $\lambda \in \mathbb{N}$ and $X = \{x_1, x_2, \cdots, x_q\} \subset X_\lambda$, $g\left(f\left(\ldots f\left(u, x_1\right)\ldots, x_1\right)\right)$ is called the accumulated value of the set X over u. Additionally, the following properties are satisfied:

- ***Efficient Generation***: Define a sequence of pairs functions $(f, g) \in F_\lambda$ and generate a key pair (sk_{acc}, pk_{acc}) for the accumulator.
- ***Efficient evaluation***: For every $(f, g) \in F_\lambda$, $u \in U_f$ and $X \subset X_\lambda$, there is an efficient algorithm to compute $g\left(f\left(u, X\right)\right)$.
- ***Quasi Commutativity***: For every $\lambda \in \mathbb{N}$, $(f, g) \in F_\lambda$, $u \in U_f$ and $x_1, x_2 \in X_\lambda$, it holds $f\left(f\left(u, x_1\right), x_2\right) = f\left(f\left(u, x_2\right), x_1\right)$.

2.3 Security assumption

In our scheme, several theoretic definitions also have been based, *e.g.* Bilinear pairing, Diffie-Hellman Exponent assumption (n-DHE), Strong Diffie-Hellman Problem (n-SDH), Hidden Strong Diffie-Hellman Exponent assumption (n-HSDHE) [16]and a standard digital signature scheme $SDS = \langle Gen, Sig, Ver\rangle$ [4].

3 System Model

We show an overview of our proposed architecture based on the practical application scenario as well as abstract it for a undirected tree to introduce the authentication.

3.1 Authentication hierarchical architecture

In order to achieve the trusted authentication and ensure valid verification, we establish the architecture graph with three components to enhance the illustration of authentication service and application scenarios in the IIoT. As shown in Fig. 1, the first one is the third party layer (*e.g.* cloud computation) and then there is user network layer which is followed by the application devices layer.

3.2 Transitive authentication model

In this section, we analyze detailed the relationship between various devices shown in Fig. 1. In the user network layer, we can find that Alice can view the Logistics, Order goods, and E-commerce in the office, as well as Bob can view Storage, Workshop and After-sales service in the office. If Alice's supervisor Eve wants to monitor the Workshop which belongs to another supervisor's manage

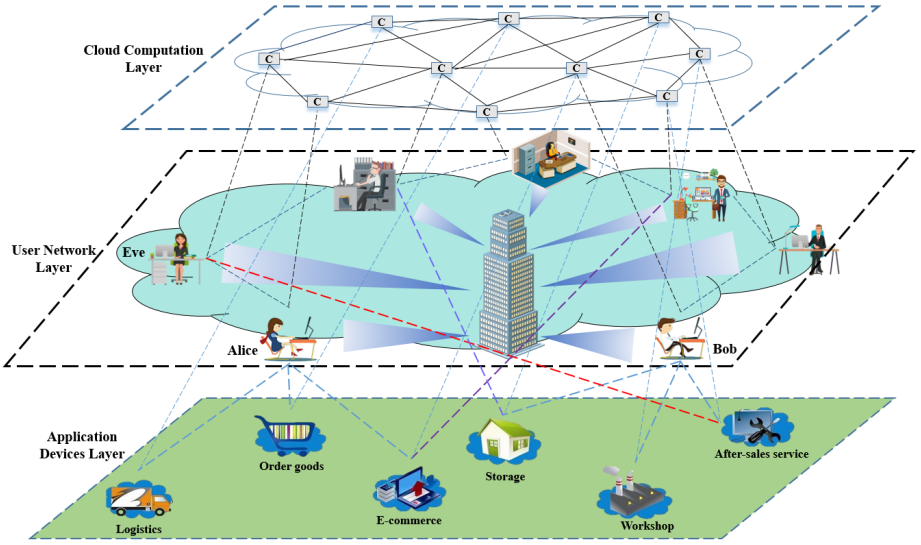


Fig. 1: System architecture

domain, she needs to go through the legal permission layer by layer, and finally gets Bob's authentication to permit it, even though she has the legitimate certification. The question is why Alice does not access the Workshop directly since she has legitimate certification.

In order to study this problem, we abstract a graph from above situation, as shown in Fig. 2. It can be clearly seen that all the nodes are valid digital signatures, node v_{1i} hence can be authenticated by node $v_{q,s}$ (*i.e.* the blue dotted line showed in Fig. 2. This authentication problem solved in [4] utilizes the transitive authentication relationship between superior and subordinate for a directed tree. However, if node v_{km} wants to access the node v_{qh}, we can see that node v_{km} and node v_{qh} belong to different administrator, that is, in the different administrator domain. In view of the above situation, this paper focuses on studying the digital authentication across the domain between v_{km} node and node v_{qh} (*i.e.* the red dotted line showed in Fig.2).

4 The security Model of CDAA

In this section, the formal definition of a cross-domain dynamic accumulator authentication scheme is proposed for the general graphs in IIoT.

4.1 Outline of CDAA

A secure cross-domain dynamic accumulator authentication scheme (CDAA) for the general graph G consists of the seven algorithms, *i.e.*: *Setup*, *EdgeLab*,

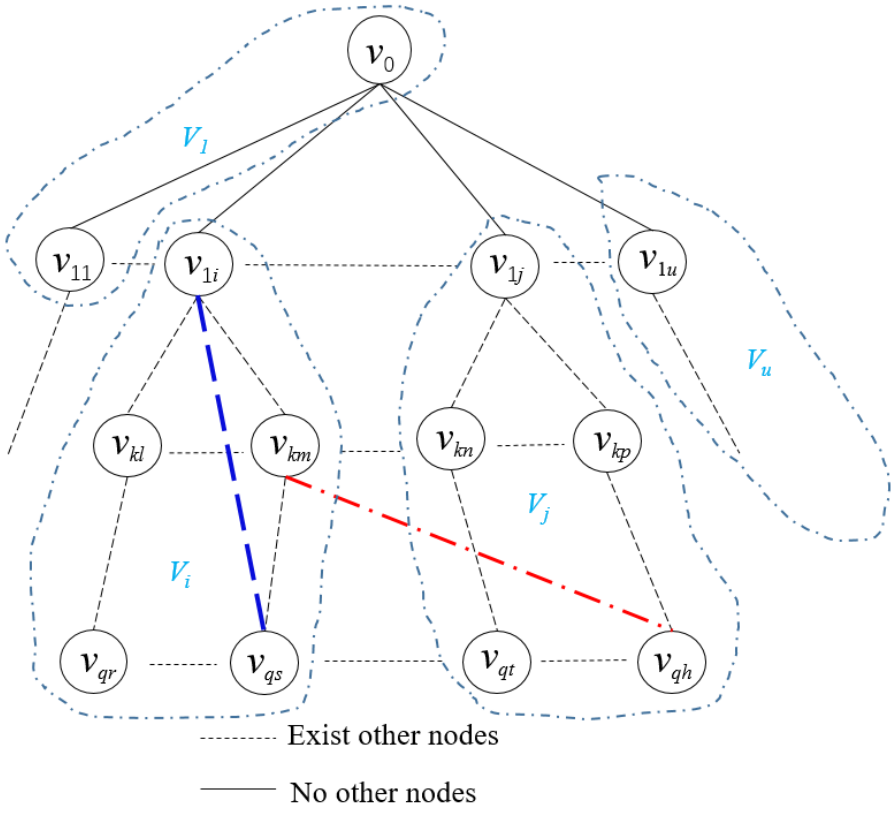


Fig. 2: System Graph Model

$EdgeVal$, $AccUpdate$, $AccWitUpdate$ and $AccVerify$. These algorithms consist of the accumulator authority, the signer, an untrusted update entity and a verifier. Among them, the authority is in charge of constructing an accumulator initial algorithm $Setup$ to generate the accumulator key pair (pk_A, sk_A), the initial accumulator status Acc_ϕ and a public state $State_\phi$. The algorithm $EdgeLab$ is used to add a new edge between node v_{ik} and v_{js}, and also to obtain a new accumulator state $Acc_{V\cup\{(v_{ik},v_{js})\}}$ and state $State_{U\cup\{(v_{ik},v_{js})\}}$, as well as a witness $wit_{(v_{ik},v_{js})}$.

Two sets of V and V_w as a bookkeeping are maintained, where V contains the values computed by $AccUpdate$ algorithm in the accumulator and V_w is the status of accumulator while a witness was $wit_{(v_{ik},v_{js})}$ created. The accumulator state $State_U$ also has a bookkeeping information set U for adding the elements in it to the accumulator, which is a superset of V and V_w. The detailed syntax of the CDAA is represented as follows:

– $Setup\left(1^k, N\right) \rightarrow \{(sk_A, pk_A), Acc_\phi, State_\phi\}$: *Initialization Setup algorithm* take as input a security parameters $1^k\,(k \in N)$, then generates public pa-

rameters set $params$, the accumulator key pair (sk_A, pk_A), a standard digital signature scheme SDS, the accumulator security key pair (spk, ssk), an empty accumulator value Acc_ϕ and a state information list $State_\phi$.
- $EdgeLab\,(v_{ik}, v_{js}) \rightarrow \{x_{i'}\}$: *Edge Label algorithm* is used to generate edge label among nodes needed to construct authentication. Then output a label $x_{i'}$ for edge (v_{ik}, v_{js}).
- $EdgeVal\,(x_{i'}, ssk) \rightarrow \{\sigma_{i'}, w\}$: *Edge Value algorithm* is used to calculate an value w, the edge representation notation and generate the new edge signature $\sigma_{i'}$.
- $AccUpdate\,(v_{ik}, v_{js}) \rightarrow \{Acc_V, State_U\}$: *Accumulator Update algorithm* is used to update $Acc_{V\cup\{(v_{ik},v_{js})\}}$ and $State_{U\cup\{(v_{ik},v_{js})\}}$, then outputs the new accumulator Acc_V and $State_U$.
- $AccWitUpdate\,(x_{i'}) \rightarrow wit'_{i'}$: *Accumulator Witness Update algorithm* is used to input edge label notation $x_{i'}$, then outputs the updated witness$wit'_{i'}$.
- $AccVerify\,(w'_{i'}, pk_A, g_{i'}, Acc_V, w) \rightarrow \{0, 1\}$: *Accumulator Verify algorithm* takes as input parameters and verifies the signature and witness.

4.2 Security of CDAA

The security requirements of the CDAA are defined through the following game between the challenger and the attacker. We call the CDAA is secure if no $\mathcal{PPT}$ adversary $\mathcal{A}$ has a non-negligible advantage in the following phase against a challenger $\mathcal{C}$.

- ***Phase 1.*** The changer $\mathcal{C}$ runs the *Setup* algorithm to generate the corresponding parameters, *.e.g.* the public parameters $params = (q, G, G_T, e, g, H, H')$, a standard digital signature scheme SDS, the accumulator key pair (sk_A, pk_A), an empty accumulator value Acc and a state information list $State$. Then $\mathcal{C}$ initializes Acc_ϕ=1 and $State_\phi$= $(\phi, g_1 = g^{\gamma^1}, \cdots, g_n = g^{\gamma^n}, g_{n+2} = g^{\gamma^{n+2}}, \cdots, g_{2n} = g^{\gamma^{2n}})$, then sends pk_A, spk to the adversary $\mathcal{A}$.

- ***Phase 2.*** In this phase, number of queries as following to $\mathcal{C}$ is given by the adversary $\mathcal{A}$: $EdgeLab$, $EdgeVal$, $AccUpdate$, $AccWitUpdate$ and $AccVerify$.
  - $EdgeLab$ and $EdgeVal$ queries. Adversary $\mathcal{A}$ randomly chooses a general graph $G = (V, E)$. He arbitrarily chooses two nodes v_{ik} and v_{js} without edge between them, where $i < j$. Then he runs $EdgeLab$ algorithm to obtain edge label $x_{i'}$ between nodes v_{ik} and v_{js}, then sends to $\mathcal{C}$.
  - $AccUpdate$ queries. $\mathcal{A}$ could change the state of accumulator adaptively, by querying $\mathcal{C}$ to add $x_{i'}$ to the set. For any $x_{i'}$, $\mathcal{A}$ queries the witness and $\mathcal{C}$ runs $WitAdd$ algorithm to respond with $wit_{i'}$ to $\mathcal{C}$. In addition, $\mathcal{C}$ runs $AccUpdate$ algorithm to obtain the newly updated state set $State$ and output the accumulator value Acc_V.
  - $AccWitUpdate$ query. $\mathcal{A}$ can query the witness many times for label notation $x_{i'}$ (i.e. corresponding with edge (v_{ik}, v_{js})). Each time $\mathcal{C}$ runs $AccWitUpdate$ algorithm to generate and update the witness $wit'_{i'}$ to $\mathcal{A}$ as a response.

– ***Phase 3.*** After all context processes have been finished, $\mathcal{A}$ submits a forgery $(x_{j'}, wit_{j'}, Acc_{V'})$ on edge $(v_{t'k'}, v_{l's'})$ chosen by himself. $\mathcal{A}$ wins the game if the following conditions holds:
  - $AccVerify\left(pk_A, x_{j'}, wit_{j'}, Acc_{V'}\right) = 1$;
  - $x_{j'}$ has never been queried during the game.

We use $Adv_A^{CDAA}(k)$ to represent the probability of an adaptive chosen-message adversary $\mathcal{A}$ that wins the above game .

Definition 1. The CDAA is unforgeable under chosen-message attacks if the following function $Adv_A^{CDAA}(k)$ is negligible for any $\mathcal{PPT}$ adversary $\mathcal{A}$.

$$\begin{aligned} Adv_A^{CDAA}(k) = \Pr[&params \leftarrow Setup\left(1^k, N\right); \\ &\Delta \leftarrow EdgeLab\left(params\right); \\ &wit_{i'} \leftarrow EdgeVal\left(params, x_{i'}\right); \\ &\left(State_U, Acc_V\right) \leftarrow AccUpdate\left(params, U, V\right); \\ &1 \leftarrow AccVerify\left(pk_A, x_{i'}, wit_{i'}\right) \wedge \left(v_{ik}, v_{js}\right) \notin G] \end{aligned} \tag{1}$$

4.3 Privacy

Let graph G be composed of nodes i, j, k and edges $(i,j), (j,i), (j,k)$. The meaning of privacy is the verifier, not to obtain any other information about the other nodes and edges, receiving a witness for (i,j). That is, the verifier could not get any other information on the graph while he verifies an edges witness.

Definition 2. The CDAA is leakage-free, even if an adaptive chosen-message attacker could know any undisclosed information of the group.

5 Our Construction

We present our CDAA scheme and the proposed construction as follows. Randomly chosen two nodes v_{ik} and v_{js} for a given group, let (v_{ik}, v_{js}) denote an edge.

5.1 Concrete scheme

Setup: Take as input a security parameters 1^k and $\ell = \lfloor k/2 \rfloor - 2$, generates an accumulator above and the setup parameters $params = (q, \mathbb{G}, \mathbb{G}_T, e, g, H, H')$ of a bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$, where $H : N \to \mathbb{Z}_N^*$ and $H' : \{0,1\}^* \to \{0,1\}^\ell$ are two public hash functions. Let g be the generation of $\mathbb{G}$. It does as follows:

– Randomly choose a value $\gamma \in Z_q$ and run a standard digital signature algorithm $Signa\left(1^k\right)$ to obtain a key pair (spk, ssk) for digital signature node.
– Output the accumulator public/private key pair (pk_A, sk_A), where $pk_A = \left(spk, z = e(g,g)^{\gamma^{n+1}}\right)$, $sk_A = (\gamma, ssk)$.

– The public algorithm creates an initial accumulator Acc_{ϕ}=1 and a public state set $State_{\phi}$= $(\phi, g_1 = g^{\gamma^1}, \cdots, g_n = g^{\gamma^n}, g_{n+2} = g^{\gamma^{n+2}}, \cdots, g_{2n} = g^{\gamma^{2n}})$.

EdgeLab: This algorithm is used to maintain a state information set Δ to store the edge labels. It does as follows:

– For a given graph G, the label of edge (v_{ik}, v_{js}) between node v_{ik} and node v_{js} is denoted by $x_{i'}$,where $x_{i'} = H'(H(i) \| H(k) \| H(j) \| H(s))$ and $i < j$.
– If $x_{i'} \notin \Delta$, then updates Δ=$\Delta \cup \{\delta_{i_k j_s}\}$.

EdgeVal: The signer runs this algorithm to calculate a value w and a signature. It does as follows:

– For any $x_{i'} \in \Delta$, compute $w = \prod_{j' \in X}^{j' \neq i'} g_{n+1-j'+i'}$.
– Run $Signa$ to calculate a signature $\sigma_{i'}$=$Signa(ssk, g_{i'} \| x_{i'})$.
– Output $wit_{i'} = (w, \sigma_{i'}, g_{i'})$.

AccUpdate: The signer runs this algorithm to check whether $(v_{ik}, v_{js}) \notin G$ (*i.e.* $x_{i'} \notin \Delta$), then does as follows:

– Update the state set $State_{U \cup \{(v_{ik}, v_{js})\}}$= $(U \cup \{(v_{ik}, v_{js})\}, g_1, \cdots, g_n, g_{n+2}, \cdots, g_{2n})$.
– If $x_{i'} \in \Delta$, calculates the accumulator value $Acc_{V \cup \{(v_{ik}, v_{js})\}} = Acc_V \cdot g_{n+1-i'}$, and outputs $Acc_V = \prod_{v' \in X} g_{n+1-v'}$; otherwise outputs $\perp$.

AccWitUpdate: The authority runs this procedure to obtain the updated witness after edge (v_{ik}, v_{js}) added into the graph G. For an edge (v_{ik}, v_{js}) with the corresponding label notation $x_{i'}$, does as follows:

– If $(v_{ik}, v_{js}) \in V$, compute $w' = w \cdot \frac{\prod_{j \in V \setminus V_w} g_{n+1-j+i}}{\prod_{j \in V_w \setminus V} g_{n+1-j+i}}$.

– Output the updated witness $wit'_{i'} = (w', \sigma_{i'}, g_{i'})$.

AccVerify: For an edge (v_{ik}, v_{js}) with the corresponding notation $x_{i'}$, this algorithm inputs $w'_{i'}$, pk_A, $g_{i'}$, Acc_V, w, and does as follows:

– If $VerA(spk, \sigma_{i'}, g_{i'} \| x_{i'}) = 0$, then outputs $\perp$.
– If $AccVerify(pk_A, x_{i'}, wit_{i'}, Acc_V) = 0$ (i.e. $\frac{e(g_{i'}, Acc_V)}{e(g,w)} \neq z$), then output reject.

5.2 Correctness

We assume that Acc_V is an accumulator for $sk_A=(params, \gamma, ssk)$, $pk_A = (params, spk, z = e(g,g)^{\gamma^{n+1}})$, and $state_U = (U, g_1 = g^{\gamma^1}, \cdots, g_n = g^{\gamma^n}, g_{n+2} = g^{\gamma^{n+2}}, \cdots, g_{2n} = g^{\gamma^{2n}})$. Before updating the witness, a correct accumulator value $Acc_V = \prod_{j' \in X} g_{n+1-j'}$ (*e.g.* $x_{j'} \in \Delta$ corresponding edge $(v_{t'k'}, v_{l's'}) \in G$) should be calculated. For each $x_{i'} \in \Delta$ (*i.e.* edge $(v_{ik}, v_{js}) \in G$), the update witness $wit'_{i'} = (w', \sigma_{i'}, g_{i'})$ is correct while the following equation holds:

$$\frac{e\left(g_{i'}, Acc_V\right)}{e\left(g, w\right)} = \frac{e(g,g)^{\sum_{j' \in X} \gamma^{n+1-j'+i'}}}{e(g,g)^{\sum_{j'}^{j' \neq i'} \gamma^{n+1-j'+i'}}} = z \tag{2}$$

5.3 Edge deletion

The proposed scheme not only supports the addition of edges, but also satisfies the deletion of edges. In this section, we slightly modify our scheme to achieve the deletion of edges. Note that, if the authentication relationships are terminated among the devices, the corresponding edges are also deleted in the graph. Assume that an edge $(v_{ik}, v_{js}) \in E$ ($x_{i'} \in \Delta$) is deleted from the graph.

Delete $(Acc_V, x_{i'}, ssk)$: On input $Acc_V, x_{i'}$ and $\sigma_{i'}$, this algorithm checks whether $x_{i'} \in \Delta$ and returns $\perp$ otherwise. It computes $\hat{A}cc = Acc_V / g_{n+1-i'}$, and runs $Signa$ to obtain $\sigma_{\hat{A}cc}$=$Signa\left(ssk, \hat{A}cc\right)$. Finally, it returns the updated accumulator $\hat{A}cc$ and its signature $\sigma_{\hat{A}cc}$. Here it should be emphasized that the signature $\sigma_{\hat{A}cc}$ is computed on $\hat{A}cc$ instead of $g_{i'} \| x_{i'}$.

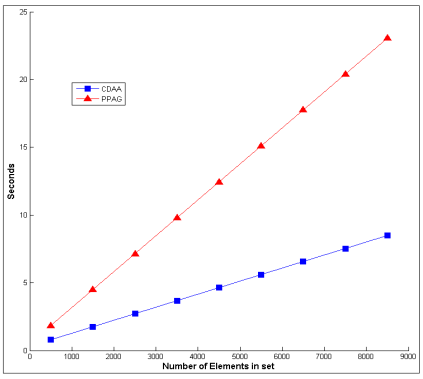
AccWidUpdate $(x_{j'}, spk, w, op)$: This algorithm inputs parameters spk, w, $x_{j'}$ and op. Here, $\hat{A}cc$ represents the already updated accumulator and Acc the accumulator before the update. Notation op will represent the $x_{j'}$ deleted or added to the accumulator $\hat{A}cc$. It returns the updated witness if $op = 0$ or if $op = 1$.

6 Performance Evaluation

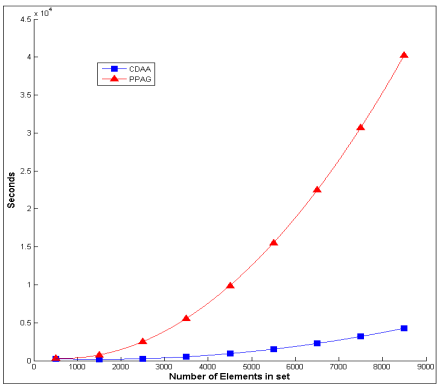
In this section, we give the whole protocol comparison and concentrate on the public parameters, private key, accumulator update cost and so on.

6.1 Computational cost

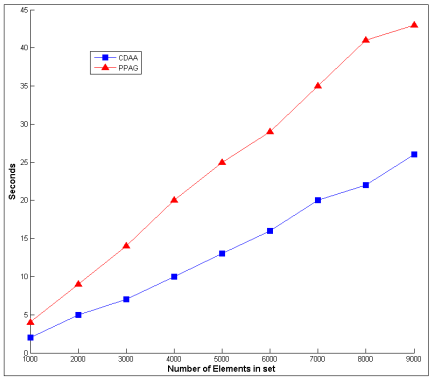
Since the accumulator add computation, accumulator up-to-date computation and accumulator witness up-to-date operation dominate the computational overhead, we only focus on these operations following estimation. Table I shows computation complexity analysis on the maximum computation cost of one member during the execution between our protocol and literature [4]. In order to easily



(a) The accumulation execution time



(b) The witness generation execution time



(c) The verification time

Fig. 3: Execution time

illustrate the analysis comparison, an abbreviated notation is given. That is: C_{h2}-the cost of computing the prime representative of one element; C_{hw}, C_H-the cost of computing one $H'(\cdot)\, and H(\cdot)$ hash operation respectively; C_e -the cost of computing one exponentiation operation; C_p-the cost of computing $e(g,g)$; C_{mult}-the cost of computing one modular multiplication; C_{Sig} and C_{Svf}-the cost of computing one signing operation and verification operation for SDS respectively. If there is n nodes and m edges (where $m = 1, \cdots, n(n-1)$)in the graph $G = (V, E)$. When adding a new edge (v_{ik}, v_{js}) to the graph G, the computation cost of all algorithms is light than the literature [4].

6.2 Experimental Test

The pairing-based cryptography (PBC) programs are C programs running on an Intel@ Core 3.4 GHz and 8G RAM desktop PC. We used SHA-1 and ECC as the

Table 1: Computation cost

$CDAA/PPAG$	Ours	Literature [4]
$EdgeLab/Elabcost$	$C_{hw} + 4C_H$	$C_{hw} + 2C_H$
$EdgeVal/Witcost$	$(2+n)\,C_e + C_{Sig}$	$C_{h2} + C_e + C_{Sig}$ $+\left(n^2-1\right)C_{mult}$
$AccUpdate/UEnalcost$	$C_e + C_{mult}$	$\left(n^2-1\right)C_{mult}$ $+C_e$
$AccWitUpdate/UWitcost$	$C_{hw} + 4C_H + C_e$ $+2C_{mult} + C_{Sig}$	$C_{hw} + 2C_H + C_{h2}$ $+C_e + C_{Sig}$
$AccVerify/Verfcost$	$C_{hw} + 4C_H + C_{Svf}$ $+C_e + C_{mult} + 2C_p$	$C_{hw} + 2C_H + C_{Svf}$ $+C_e + C_{h2}$
Total	$3C_{hw} + 12C_H + C_{Svf}$ $+(5+n)\,C_e + 2C_{Sig}$ $+2C_p + 4C_{mult}$	$8C_H + 4C_{hw} + C_{Svf}$ $+5C_e + 2C_{Sig} + 4C_{h2}$ $+\left(2n^2-2\right)C_{mult}$

hash function and Bilinear Pairing, as well as simulated our experiments on the PBC library (version 0.5.12) and the gnu multiple precision arithmetic library (version 6.0.0a). The Fig.2 (a) shows the execution time cost of comparison in each stage of the accumulator (accumulator, witness, and verification) between our CDAA scheme and PPAG [4]. We assume that there are 9000 elements and the simulation was run more than 100 times. It is worth noting that the bilinear map accumulator is much faster than the RSA accumulator by considerable amount simulations, especially the slopes differ by at least an order of magnitude the total running time of key generation, edge label algorithm, edge value algorithm and accumulator updated algorithm.

The Fig.2 (b) shows the execution time on updating the witness between PPAG [4] and CDAA, depending on the number of elements(from 500 up to 8500). Nevertheless, it is noteworthy that the chosen user nodes are all met the requirements of the PPAG scheme, *i.e.* all the nodes are subordinates of the signer.

Finally, Fig.2 (c) shows the running time of verification between the PPAG [4] and CDAA as the set number increases. As can be seen, the bilinear-map accumulator is faster than the RSA accumulator. The results clearly reflects that as the number of elements increases, the verification time of both protocols increase. Notably, when there is 9000 elements, the verification time of PPAG is almost 1.5 time than the verification of CDAA.

6.3 Comparative evaluation of related schemes

Furthermore, we show an objective comparison with other related ones to measure the security properties in TableII. Take the scheme in [4] as an example, which is another scheme to address the problem solving by this paper. But F. Zhu *et al.* [4] proposed scheme is based on the directed tree and the relationship between nodes are superiors and subordinates. It therefore could not be suited

Table 2: Computation overhead

Scheme	F. Zhu *et al.*[4]	CDAA Construction
Authentication technique	Accumulator standard digital signature	Accumulator standard digital signature
Tress	Yes	Yes
General graphs	No	Yes
Updatable	Yes	Yes
Leakage-free	Yes	Yes
Based on	Strong RSA	Bilinear Pairing
Cross domain Authentication	NO	Yes

for the general tree, in other words, it doesn't apply to the practical application. We represent a scheme which has one concrete attribute.

Based on above comparisons, we could see that our scheme has desirable properties and it is suited for the practical application. The scheme furthermore is not restricted by the relation between nodes and can be authenticated across different domains.

7 Conclusions

This paper abstracted the relationship of devices as a node in a general graph and took the advantage of a dynamic accumulator based on bilinear pairing to achieve the authentication of equipment. Notably, our authentication scheme not only realizes the authentication of vertical nodes, but also satisfies the authentication transmission of horizontal nodes (*i.e.* between the same layer). In addition, the merit of dynamic accumulator could provide the feasible solution for edge addition. The security analysis shows that our protocol satisfies secure needs on the accumulator authentication in Industrial Internet of Things. Beyond that, the scheme also satisfies the privacy-preserving authentication. In our future work, we will further design more efficient CDAA scheme to facilitate the authentication speed.

References

1. Chen, W., Ma, M., Ye, Y., et al.: IoT service based on jointcloud blockchain: The case study of smart traveling. Proc. IEEE Symp. Service Oriented Syst. Eng., pp. 216-221(2018)

2. Wang, K., Wang Y., Sun Y., et al.: Green industrial internet of things architecture: An energy-efficient perspective. IEEE Commun. Mag., vol. 54, no. 12, pp. 48-54(2016)
3. Bi, Z., Xu, L.-D., Wang, C.: Internet of things for enterprise systems of modern manufacturing. IEEE Trans. Ind. Informat., vol. 10, no. 2, pp. 1537-1546(2014)
4. Zhu, F., Wu, W., Zhang, Y., Chen, X.-F.: Privacy-Preserving Authentication for General Directed Graphs in Industrial IoT. Information Sciences(2019)
5. Lawson, T., Senthilnathan, T.: Effectiveness of the NIZKP Protocol for Authentication in IoT Environment. International Journal of Engineering & Technology, 7(2.6), pp. 231-235(2018)
6. Xiao, Y., Niyato, D., Wang, P., et al.: Dynamic energy trading for wireless powered communication networks.IEEE Commun. Mag., vol. 54, no. 11, pp. 158-164(2016)
7. Alur, R., Berger, E., Drobnis, A.-W. et al.: Systems computing challenges in the Internet of Things. 2016, [online] Available: https://arxiv.org/abs/1604.02980
8. Micali, S., Rivest, R.-L.: Transitive Signature Schemes. In: Preneel B. (eds) Topics in Cryptology - CT-RSA 2002. Lecture Notes in Computer Science, vol 2271. Springer, Berlin, Heidelberg(2002)
9. Shahandashti, S.-F., Salmasizadeh, M., Mohajeri, J.: A provably secure short transitive signature scheme from bilinear group pairs. In: Blundo, C., Cimato, S. (eds.) SCN 2004. LNCS, vol. 3352, pp. 60-76. Springer, Heidelberg(2005)
10. Ma, C., P., Wu, Gu, G.: A new method for the design of stateless transitive signature schemes. in: Advanced Web and Network Technologies, and Applications, APWeb 2006 International Workshops: XRA, IWSN, MEGA, and ICSE, Proceedings, Harbin, China, January 16-18, 2006, in: Lecture Notes in Computer Science, vol. 3842, pp. 897-904, Springer(2006)
11. Rivest, R.-L., Hohenberger, S.-R.: The cryptographic impact of groups with infeasible inversion. Master's thesis, MIT(2003)
12. Neven, G.: A simple transitive signature scheme for directed trees. Theor. Comput. Sci. vol. 396, pp. 277-282(2008)
13. Xu, J.: On directed transitive signature. IACR Cryptology ePrint Archive 2009, vol. 209. http://eprint.iacr. org/2009/209(2009)
14. Camacho, P., Hevia, A.: Short transitive signatures for directed trees. IACR Cryptol. vol. 438, ePrint Archive(2011)
15. Lin, C., He, D., Huang, X., et al.: A new transitively closed undirected graph authentication scheme for blockchain-based identity management systems. IEEE Access, vol. 6, pp. 28203-28212(2018)
16. Boneh, D., Boyen, X., Goh, E.-J.: Hierarchical identity based encryption with constant size ciphertext. In R. Cramer, editor, EUROCRYPT, vol. 3494 of Lecture Notes in Computer Science, pp. 440-456. Springer(2005)