IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2018.DOI

A New Transitively Closed Undirected Graph Authentication Scheme for Blockchain-based Identity Management Systems

CHAO LIN[1], DEBIAO HE[2], XINYI HUANG[3], MUHAMMAD KHURRAM KHAN[4], AND KIM-KWANG RAYMOND CHOO.[5]

[1]Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University, Wuhan, China and Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin, China (e-mail: linchao91@qq.com)
[2]Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University, Wuhan, China and Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin, China (e-mail: hedebiao@163.com)
[3]School of Mathematics and Computer Science, Fujian Normal University, China (e-mail: xyhuang81@gmail.com)
[4]Center of Excellence in Information Assurance, King Saud University, Saudi Arabia (e-mail: mkhurram@ksu.edu.sa)
[5]Department of Information Systems and Cyber Security, The University of Texas at San Antonio, San Antonio, USA (e-mail: raymond.choo@fulbrightmail.org)

Corresponding author: Debiao He (e-mail: hedebiao@163.com).

The work was supported in part by the National Key Research and Development Program of China under Grant 2017YFB0802504, in part by the National Natural Science Foundation of China under Grant 61501333, Grant 61572379, Grant 61472287, and Grant 61772377, and in part by the Natural Science Foundation of Hubei Province of China under Grant 2017CFA007 and Grant 2015CFA068.

ABSTRACT Blockchain can potentially be deployed in a wide range of applications due to its capability to ensure decentralisation, transparency and immutability. In this paper, we design a cryptographic membership authentication scheme (i.e. authenticating graph data) to support blockchain-based identity management systems (BIMS). Such a system is designed to bind a digital identity object to its real-world entity. Specifically, we introduce a new Transitively Closed Undirected Graph Authentication (TCUGA) scheme, which only needs to use node signatures (e.g. certificates for identifying nodes). The trapdoor hash function used in our scheme allows the signer to efficiently update the certificates without the need to re-sign the nodes. In other words, our scheme is efficient even though the graph dynamically adds or deletes vertices and edges. Moreover, our proposal can efficiently provide a proof when the edge between two vertices does not exist; thus, solving the existing intractability issue in transitive signature (the main tool for authenticating graph data). Finally, we prove the security of our proposed TCUGA in the standard model and evaluate its performance to show its feasibility for BIMS.

INDEX TERMS Authentication, Blockchain-based identity management, Graph data, Transitively unforgeable

I. INTRODUCTION

The popularity of blockchain is evidenced in the attempts to implement blockchain in applications ranging from finance [1]–[4], data storage [5], [6], Internet of Things (IoT) [7]–[9], data provenance [10], [11], identity [12]–[14], and so on. There has also been attempts to integrate blockchain with cryptographic designs and applications [15], [16]. Blockchain-based approaches have the potential to enhance decentralisation, transparency, and immutability, moreover, different applications have different cryptographic / security requirements. For instance, a model with amount hiding property or anonymity requirement requires an appropriate non-interactive zero knowledge scheme [3], [17], and a distributed data outsourcing and verifiable computation system design requires a specified functional algorithm [18]–[20]. In this paper, we mainly focus on the authentication of membership to support the design of blockchain-based identity management systems (BIMS).

BIMS such as those presented in [12], [14] utilizes blockchain to build digital identity objects (i.e. Personal Identification Number, PIN) and validate their bindings to real-world entities. Fig. 1 shows an overview of the identity

ledger building process [14]. In such a process, when a member wishes to bind his/her real-world public attributes to a PIN, he/she chooses a provider to validate the PIN then records this into the blockchain. Here, the provider must own a completed identity, meaning that it has been validated by an (independent) authority. Then, other members will authenticate the PIN to increase its credibility. Notably, all the involved authentications (i.e. authority-provider, provider-member, member-member) are chained into the blockchain for a public verification, and any member seeking like to authenticate another entity needs to provide the certification of their membership (i.e. represented by the red line in Fig. 1).

However, membership authentication is generally not considered in existing BIMS, such as the approach of Muftic *et al.* [14]. We posit the importance of membership authentication and for this activity to be undertaken by an administrative authority. For instance, assuming that Alice and Bob are working for the same organisation, and Alice should provide their interpersonal relationship when she wishes to authenticate Bob's PIN. Otherwise, Bob can require many non-interpersonal ones to authenticate his PIN, which will result in an 'unfaithful' credibility. Generally, the administrative authority (e.g. the organisation's cyber security team) generates the certificate between two members to prove their relationship. This will, however, incur significant computational costs since every pair requires a separate signed certificate, and clearly not scalable.

In this paper, we propose achieving membership authentication in BMIS using a graph data system formed by different equivalence classes (see Section III-A as an example). In the undirected graph, vertices and edges represent company members and their relationship (e.g. administrative domains) respectively. Transitive signature is an efficient authentication tool for graph data, and in such a scheme the signer only needs to sign a targeted graph's (denoted as G) transitive reduction graph (here, transitive reduction of G has the same transitive closure as G, but has the least number of edges). After signing the transitive reduction graph, the signer sends all signatures to an administrator (which is responsible for securely storing the signatures and replying users' queries). According to the composition algorithm in transitive signatures, the administrator or others can directly compute a novel signature of the composed edge. That is, given two signatures of edges (i, j) and (j, k), one can compose the signature of edge (i, k) without the original signer's signing private key. This composition property can help the signer to minimize computational costs (i.e. the complexity reduces from $O(N^2)$ to $O(N)$ [21]).

While there are a number of transitive signature schemes for undirected graphs in the literature, such proposals generally support only dynamically adding and not edge deletion (e.g. when a user leaves an organisation). In addition, the administrator cannot efficiently provide the certificate of non-existent edges. A trivial method is for the signer to sign non-existent edges using his/her private key. This, however, is impractical for a large number of non-existent edges which is generally the case for most larger organisations. In order to design a more practical authentication for blockchain-based identities management system, we are motivated to design a scheme that can dynamically and efficiently update vertices and edges in the undirected graph.

Contributions. We leverage the inner relation between vertices' public labels to eliminate edge signatures. Modified signature schemes (i.e., standard hash functions used in standard signature schemes are replaced by trapdoor hash functions [22]) are adopted so that the signer only needs to re-compute the randomness using trapdoor hash functions for vertices updated in the corresponding equivalence class (see Section V). In our scheme, an administrator can directly reply with the certificates of the queried vertices to prove their relationship even when they are not in the same equivalence class. After updating the nodes' certificates, the signer only needs to send the new certificates to the administrator, and the administrator chains them into the blockchain to invalidate the previous ones (i.e. only the latest public labels of a node are valid due to the use of timestamp in the certificate). In order to prevent the malicious propagation of certificates by requestors, we propose that designated verifier signatures can be integrated into our scheme in future work.

Paper Organization. Related work and relevant preliminaries are presented in Sections II and III, respectively. In Section IV, we present the security model of our proposed TCUGA, prior to presenting the scheme in Section V. Also in the latter, we present the scheme's security and performance analysis. Section VI concludes this paper.

II. RELATED WORK

Transitive signature is first proposed by Micali and Rivest [23], who presented two secure concrete schemes based on discrete logarithm and RSA assumptions, respectively. The RSA-based scheme is only proven to be transitively unforgeable (see Section IV-A) against non-adaptive chosen message attacks. In a latter work, the scheme is proven against adaptive attacks based on the one-more RSA-inversion assumption [24]. Bellare and Neven [24] also used other assumptions (i.e. the factoring, one-more discrete logarithm and gap Diffie-Hellman groups) to construct secure schemes in the standard model, as well as introducing a hashing method to avoid generating node certificates (i.e. undirected and stateless transitive signature schemes). Such an approach minimizes the computation cost at the cost of security (i.e. only secure in the random oracle model, rather than the standard model). In addition, the hash adopted in their schemes is a special hash function (i.e. MapToPoint), which is difficult to build in practice.

Lin *et al.* [25] then proposed a new undirected and stateless transitive signature scheme using general cryptographic hash functions (e.g., MD6 or SHA-512). The scheme is proven secure in the random oracle model, assuming the hardness of the M2SDH problem. There are also several other transitive signature schemes, but most of which are all designed only

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2837650, IEEE Access

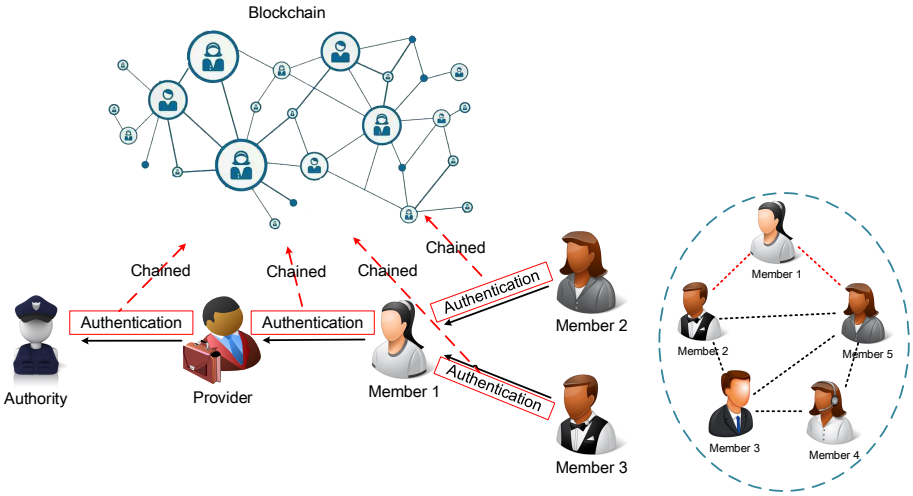

FIGURE 1. Identity ledger (with validators).

for undirected graph [26]–[29].

Transitive signature schemes for general directed graphs may be infeasible, since the construction forms a Abelian trapdoor group known as the infeasible inversion that does not have a known construction [30]. Existing directed transitive schemes were merely designed for special directed graphs (e.g., directed trees) and proven secure against adaptive chosen-message attacks [31]–[34]. Moreover, the schemes in [32], [34] were designed without edge signatures. Edges are authenticated via the inner relation (e.g., character strings or codes) between two vertices. This inspires us to apply this method into the authentication of undirected graph.

Apart from transitive signatures, Xu [33] explained how one can utilize redactable signatures [35] to authenticate general directed graphs. First, the signer signs the total directed graph using redactable signatures, prior to sending the signatures to the administrator. Once someone queries for the existence of edge (i, j), the administrator can redact the total signature and obtain the queried edge signature according to the redactable signature's property.

We note that the above authentication tools for graph data would not be effective when the graph dynamically deletes the edge's vertices. This is also observed by Ma *et al.* [36], who then designed a new authentication tool using the one-way accumulator for undirected graph. Unlike transitive signature schemes for undirected graphs, their scheme does not require edge signatures. Instead of the standard digital signatures, they utilize the one-way accumulators to eliminate the need of a signature on its edges. The administrator can directly reply with the node certificates to prove the edge even if it does not exist. Their scheme requires smaller storage yet achieves higher efficiency, as well as allowing the graph to dynamically remove vertices or edges. However, the signer needs to re-sign all nodes once the graph deletes or adds vertices (or edges) in the same equivalence class. This, as explained earlier, incurs significant computational costs. Hence, we focus on the design of a more efficient authentication scheme suitable for BIMS.

III. PRELIMINARIES

In this section, we present the preliminaries related to our proposed scheme. First, we present the relevant notations.

The notation $\mathbb{N} = \{1, 2, \dots N\}$ denotes the set of positive integers from 1 to N, and $\mathbb{Z}_q^*$ denotes the multiplicative group of integers modulo q. The notation $x \leftarrow \mathbb{S}$ indicates that x is selected randomly and uniformly from set $\mathbb{S}$. $f\colon \mathbb{N} \rightarrow \mathbb{R}$ is a negligible function if follows: $\forall m > 0$, there exists n_0 for all $n > n_0 : f(n) < 1/n^m$. A $\mathcal{PPT}$ algorithm is marked as a probabilistic algorithm with polynomial runtime.

A. GRAPHS

Membership authentication in BIMS is an authentication graph-based structure. Here, we define $G = (V, E)$ as an undirected graph, where $V \subseteq \mathbb{N}$ is a finite set of vertices and $E \subseteq V \times V$ a finite set of edges. According to the equivalent relation, we split the graph $G = (V, E)$ into several equivalence classes $D(V) = \{V_1, V_2, \dots, V_m\}$, where $m = |D(V)|$. $\widetilde{G} = (\widetilde{V}, \widetilde{E})$, where $\widetilde{V} = V$ and $(i, j) \in \widetilde{E}$, denotes G's transitive closure iff there exists a path from v_i to v_j in V. Because of the undirected graph's (i.e $G = (V, E)$) transitivity, authenticating G is equivalent to authenticating $\widetilde{G}$.

Fig. 2 shows an example of a transitive graph and its transitive closure.

B. TRAPDOOR HASH FUNCTIONS

A trapdoor hash function TH [22] comprises three algorithms, i.e. TH = (KG, HF, CSF).

- $(pk, sk) \leftarrow \mathsf{KG}(1^k)$: Given a security parameter 1^k, this key generation (KG) algorithm returns a private/public-key pair (pk, sk).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2837650, IEEE Access

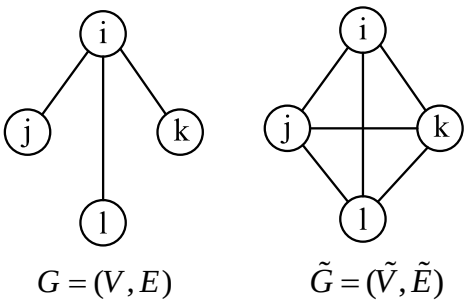

FIGURE 2. Transitive Closure $\tilde{G}$ of Graph G.

- $h \leftarrow \mathsf{HF}_{pk}(m, r)$: Given a public key pk, message $m \in \mathbb{M}$ and random $r \in \mathbb{R}$, this hash function evaluation algorithm (HF) returns a hash string $h \in \mathbb{H}$. The notations $\mathbb{M}$, $\mathbb{R}$ and $\mathbb{H}$ respectively denote the message, randomness and hash spaces.
- $r_2 \leftarrow \mathsf{CSF}((pk, sk), (m_1, r_1), m_2)$: Given a key-pair (pk, sk), a message/randomizer pair $(m_1, r_1) \in \mathbb{M} \times \mathbb{R}$ and a second message $m_2 \in \mathbb{M}$, this collision solver function (CSF) algorithm returns a second randomizer r_2 such that $\mathsf{HF}_{pk}(m_1, r_1) = \mathsf{HF}_{pk}(m_2, r_2)$.

Trapdoor hash functions $\mathrm{TH} = (\mathsf{KG}, \mathsf{HF}, \mathsf{CSF})$ satisfy the following two security properties.

1) Collision Resistance: A TH scheme is collision-resistant if $\mathsf{Succ}^{CR}_{A,\mathrm{TH}}$ in the following game is negligible (here, $\mathsf{Succ}^{CR}_{A,\mathrm{TH}}$ is the successful finding collision probability of any efficient $\mathcal{PPT}$ attacker $\mathcal{A}$). Let $(pk, sk) = \mathsf{KG}(k)$ be a key-pair, and $\mathcal{A}$ is given the security parameter k and the public key pk. $\mathcal{A}$ succeeds in a polynomial time t to output a collision (m_1, r_1) and (m_2, r_2) for HF_{pk}, satisfying $\mathsf{HF}_{pk}(m_1, r_1) = \mathsf{HF}_{pk}(m_2, r_2)$ and $m_1 \neq m_2$.
2) Trapdoor Collisions: We call a TH scheme is perfectly-trapdoor if it owns this property: $\forall (sk, pk) = \mathsf{KG}(k)$ and message pair $(m_1, m_2) \in \mathbb{M} \times \mathbb{M}$, if r_1 is uniformly and randomly chosen from $\mathbb{R}$, then $r_2 = \mathsf{CSF}((sk, pk), (m_1, r_1), m_2) \in \mathbb{R}$ has a uniform probability distribution on $\mathbb{R}$.

There are several trapdoor hash functions have been presented [22], [37], all of which can be applied in our proposed TCUGA scheme. The following design called TH_{DL} is used to analyze the performance of our scheme.

- KG: Let $\mathcal{H} : \{0,1\}^* \rightarrow \mathbb{Z}_q$ be any collision-resistant hash function, this algorithm chooses a group $\mathbb{G}$ of prime order q and $x \xleftarrow{R} \mathbb{Z}^*_q$. Then it computes $y = g^x \in \mathbb{G}$. Finally, it returns a key pair (pk, sk), where $pk = (D_G, g, y)$, $sk = (D_G, g, x)$ and D_G is a description of group $\mathbb{G}$ (including the prime group order q).
- HF: This algorithm takes as input the public key $pk = (D_G, g, y)$ and message/randomizer pair $(m, r) \in \{0,1\}^* \times \mathbb{Z}_q$, and computes the hash $h_s = F_{pk}(m, r) = g^{\mathcal{H}(m)} \cdot y^r \in \mathbb{G}$.
- CSF: This algorithm takes as input the key-pair (pk, sk) and (m_1, r_1, m_2), computes $r_2 = r_1 + (\mathcal{H}(m_1) - \mathcal{H}(m_2)) \cdot x^{-1} \pmod q$ and returns the second randomizer r_2.

The above TH_{DL} is collision-resistant assuming that $\mathcal{H}$ is collision-resistant and the discrete-log problem is hard in $\mathbb{G}$. In addition, TH_{DL} satisfies the property of perfectly-trapdoor, more details can be referred in [22].

C. MODIFIED DIGITAL SIGNATURES BASED ON TRAPDOOR FUNCTIONS

In a traditional digital signature scheme, the signer is equipped with a public/private key pair. And the usual operations of signing, denoted by Sign, and verification, denoted by Verf. We review its algorithms $\mathrm{S} = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verf})$ as follows.

- $(sk, pk) \leftarrow \mathsf{KeyGen}(1^k)$. Given the security parameter 1^k, this key generation (KeyGen) algorithm returns a private/public key pair (sk, pk).
- $\sigma \leftarrow \mathsf{Sign}(sk, m)$. Given a private key sk and a message m, this signing algorithm returns a signature σ on m.
- $\{1, 0\} \leftarrow \mathsf{Verf}(pk, m, \sigma)$. Given a public key pk and a message-signature pair (m, σ), this verifying algorithm returns either 1 or 0. If returns 1, then σ is a valid signature on m.

Consistence of S. For any message m and any key pair (sk, pk) obtained by invoking KeyGen algorithm, the equation $\mathsf{Verf}(pk, m, \mathsf{Sign}(sk, m)) = 1$ holds.

The security model of digital signature schemes, existential unforgeablity under chosen-message attacks (euf-cma) [38], is defined as a game between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$. The challenger runs KeyGen algorithm to obtain public/private key pair, and gives the public key to $\mathcal{A}$. Moreover, $\mathcal{A}$ can adaptively request the signing oracle $\mathsf{Sign}(sk, \cdot)$. Eventually, $\mathcal{A}$ outputs a forged message/signature pair (M^*, σ). Here, let $\mathsf{Adv}^{euf-cma}_{\mathcal{S},\mathcal{A}}(k)$ be the probability that $\mathcal{A}$ wins the game, meaning that $\mathsf{SVf}(pk, M^*, \sigma) = 1$ and M^* have been queried. The $\mathcal{S}$ is ufcma if $\mathsf{Adv}^{euf-cma}_{\mathcal{S},\mathcal{A}}(k)$ is negligible for any $\mathcal{PPT}$ adversary $\mathcal{A}$ [39].

In practice, we usually require an appropriate encoding of the signed information to guarantee security of S, e.g., using a cryptographic hash function (e.g., MD6 or SHA-512). In this paper, we replace the standard hash function with a trapdoor hash function $\mathrm{TH} = (\mathsf{KG}, \mathsf{HF}, \mathsf{CSF})$, where the signer holds the trapdoor information for signing the message. The Modified Signature MS scheme has some interesting properties:

1) As in a traditional digital signature scheme, the signer cannot repudiate (or deny) a signature he has generated since only does he know the private key and the trapdoor hash information.
2) The signer can pre-compute a signature σ on a message/randomness pair (m_1, r_1) in the offline phase. After he receives the target message m_2 to sign in the online phase, he can find the collision in the hash

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2837650, IEEE Access

by invoking the CSF algorithm to compute $r_2 = \mathsf{CSF}((sk, pk), (m_1, r_1), m_2) \in \mathbb{R}$ with the same signature σ. This property is really attritive when considering relief of the realtime computational burden, which makes the modified signature scheme more practical in many applications, specifically for the battery powered mobile devices or real time environment [22], [37].

IV. TRANSITIVELY CLOSED UNDIRECTED GRAPH AUTHENTICATION

A Transitively Closed Undirected Graph Authentication (TCUGA) scheme is defined by three $\mathcal{PPT}$ algorithms: TCUGA=(TKG, TASign, TAVerf).

- $(tpk, tsk) \leftarrow \mathsf{TKG}(1^k)$: Given a security parameter 1^k, the transitive key generation (TKG) algorithm returns a transitive public/private key pair (tpk, tsk).
- TASign = (TASign-Init,TASign-Node,TASign-Edge): The signing (TASign) algorithm comprises the following $\mathcal{PPT}$ algorithms.
  1) $\{Cert(v_i)\} \leftarrow \mathsf{TASign\text{-}Init}(tsk, G)$: Given the signer's private key tsk and the transitively closed graph $G = (V, E)$, this signing initialization (TASign-Init) algorithm returns the certificates $Cert(v_i)$, where $v_i \in V, i = \{1, 2, \ldots, N\}, N = |G|$.
  2) $Cert(v_k) \leftarrow \mathsf{TASign\text{-}Node}(tsk, v_k)$: Given the signer's private key tsk and node v_k, this signing node-updating (TASign-Node) algorithm returns a certificate $Cert(v_k)$ of node k.
  3) $Cert(v_k) \leftarrow \mathsf{TASign\text{-}Edge}(tsk, v_i, v_j)$: Given the signer's private key tsk and nodes v_i, v_j, this signing edge-updating algorithm (TASign-Edge) returns the certificates $Cert(v_k)$ of the updated graph $G' = (V', E')$, where $v_k \in V', k = \{1, 2, \ldots, N\}, N = |G'|$.
- $\{0, 1\} \leftarrow \mathsf{TAVerf}(tpk, v_i, v_j, Cert(v_i), Cert(v_j))$: Given the signer's public key tpk, nodes v_i, v_j and corresponding signatures $Cert(v_i), Cert(v_j)$, this verification (TAVef) algorithm returns either 1 or 0. If returns 1, then nodes v_i, v_j are in the same equivalence class.

Consistence of TCUGA Schemes. A correct TCUGA scheme should satisfy the consistency property that TASign must be accepted as valid by TAVef if and only if nodes v_i, v_j are in the same equivalence class in TCUGA. That is,

$$Pr[\mathsf{TAVerf}(tpk, v_i, v_j, \mathsf{TASign}(tsk, v_i), \mathsf{TASign}(tsk, v_j)) = 1] = 1.$$

A. TRANSITIVE UNFORGEABILITY

We introduce the security model (i.e. transitive unforgeability) of TCUGA schemes in this section. Define $\mathcal{F}$ as a $\mathcal{PPT}$ forger with adaptive chosen-message ability against the scheme TCUGA=(TKG, TASign, TAVerf). And we use the following game between $\mathcal{F}$ and a challenger $\mathcal{C}$ to describe the unforgeability.

- Setup: The challenge $\mathcal{C}$ invokes TKG to obtain a public/private key-pair (tpk, tsk), and sends tpk to the forger $\mathcal{F}$.
- TASign queries: $\mathcal{F}$ adaptively requests the signature $(Cert(v_i), Cert(v_j))$ on edge (v_i, v_j). As a response, $\mathcal{C}$ runs TASign and returns $(Cert(v_i), Cert(v_j))$ to $\mathcal{F}$.

Here, let E' represent the queried edges set by $\mathcal{F}$ to TASign oracle, and V' the nodes set involved in E'. Eventually, $\mathcal{F}$ forges a signature $(Cert(v_i*), Cert(v_j*))$ on edge (v_i*, v_j*) using the aforementioned tpk from $\mathcal{C}$. $\mathcal{F}$ wins the game if:

1) $\mathsf{TAVerf}(tpk, i^*, j^*, Cert(v_i*), Cert(v_j*)) = 1$.
2) At least one node is not in $\widetilde{G'}$ between nodes v_i* and v_j^*, where $\widetilde{G'} = (V', \widetilde{E'})$ is the transitive closure of graph $G' = (V', E')$.

Define $\mathsf{Adv}^{tu-cma}_{\mathcal{F},\mathrm{TCUGA}}$ as the advantage of $\mathcal{F}$ wins the above game. A TCUGA scheme is transitively unforgeable against adaptive chosen-message attacks if $\mathsf{Adv}^{tu-cma}_{\mathcal{F},\mathrm{TCUGA}}$ is negligible for any $\mathcal{PPT}$ forger $\mathcal{F}$.

V. A NEW TCUGA SCHEME

This section introduces the design details of our TCUGA which can be adopted in BIMSs. Our proposal is based on a modified digital signature scheme MS, among which the hash function in the Sign algorithm is replaced by a trapdoor hash function TH. This modification can efficiently authenticate the dynamically growing graph data, i.e., the signer can efficiently add, modify and delete the relationship (e.g., the same equivalence class) among the nodes in the transitively closed undirected graph.

Next, we specify our scheme as follows. The related system parameters in our scheme are $\{\mathbb{G}, \mathbb{Z}_q^*, g, q, \mathrm{TH}\}$, where TH = (KG, HF, CSF) is a trapdoor hash function used in the MS scheme, $\mathbb{G}$ is a group (of prime order q) generated by g. Notably, the signer sends the new message/signature pairs to the administrator when updating the memberships, and the administrator securely broadcasts to the previous queried requestors to invalid the old ones after updating the public label of nodes. This broadcasting operation can be realized by using blockchain to efficiently reduce communicational costs when integrating our proposal into BIMSs.

- TKG: Given a security parameter 1^k, TKG algorithm invokes the KeyGen algorithm of MS to compute $(sk, pk) = \mathsf{KeyGen}(1^k)$, and returns the public key $tpk = pk$ and corresponding private key $tsk = sk$.
- TASign: The signing algorithm TASign comprises three $\mathcal{PPT}$ algorithms: TASign = (TASign-Init,TASign-Node,TASign-Edge).
  1) TASign-Init: We describe the Signing Initialization algorithm as follows.
     - The signer splits the graph $G = (V, E)$ into several equivalence classes $D(V) = \{V_1, V_2, \ldots, V_m\}$, where $m = |D(V)|$. And initializes $T \leftarrow \emptyset$ as a table storing all chosen

public label of nodes. Fig. 3 shows an example of the Equivalence Classes Splitting that $D(V) = \{V_1, V_2, \ldots, V_m\}$ and $m = 3$, where V_1 ={i,j,k,l},V_2 ={m,n,o},V_3 ={p,q}.

- It randomly chooses $x_i \leftarrow \mathrm{Z}_q^*$ and $y_i \leftarrow \mathrm{Z}_q^*$, where $(x_i, y_i) \notin T$. Then takes (x_i, y_i) as the public label of the first node in the ith equivalence class V_i and updates $T \leftarrow T \cup (x_i, y_i)$. For $v_j \in V_i (j \neq 1)$, the algorithm randomly chooses $r_{i,j} \leftarrow \mathrm{Z}_q^*$ and computes $x_{i,j} = r_{i,j} \cdot x_i \ mod\ q$ and $y_{i,j} = r_{i,j} \cdot y_i \ mod\ q$, where $(x_{i,j}, y_{i,j})$ is the public label of the jth node in V_i. According to the above method, we can compete the initialization of public labels in $V_i (i = 1, 2, \ldots, m)$.
- It uses the signing algorithm of MS to compute the certificates $Cert(v_{i,j}) = (x_{i,j}, y_{i,j}, t_{ij}, \sigma_{i,j})$ for all the nodes in V, where t_{ij} is a current timestamp used to identify the latest message/signature pair (i.e. one will only believe the latest one when he/she received several message/signature pairs), and $\sigma_{i,j} = \mathsf{Sign}(sk, x_{i,j}, y_{i,j}, t_{ij})$.

2) TASign-Node: We describe the Signing Node-Updating algorithm as follows.

- In order to add the node v_0, the signer processes the following:
  a) If $v_0 \in V_i, i = \{1, 2, \ldots, m\}$, then it utilizes the public label (x_i, y_i) of the first node in the ith equivalence class and initializes the label as follows: it randomly chooses $r_0 \leftarrow \mathrm{Z}_p^*$ and computes $x_{v_0} = r_0 \cdot x_i \ mod\ q$ and $y_{v_0} = r_0 \cdot y_i \ mod\ q$. Finally, the signer signs the node using his/her private key to obtain $Cert(v_0) = (x_{v_0}, y_{v_0}, t_{v_0}, \sigma_{v_0})$, where t_{v_0} is also a timestamp as above mentioned and $\sigma_{v_0} = \mathsf{Sign}(sk, x_{v_0}, y_{v_0}, t_{v_0})$.
  b) If $v_0 \notin V$, then it randomly chooses $x_{m+1} \leftarrow \mathrm{Z}_q^*$ and $y_{m+1} \leftarrow \mathrm{Z}_q^*$, where (x_{m+1}, y_{m+1}) is the public label of the first node in the new $(m+1)$th equivalence class V_{m+1} and $(x_{m+1}, y_{m+1}) \notin T$. Then it updates $T \leftarrow T \cup (x_{m+1}, y_{m+1})$ and computes the certificate $Cert(v_0) = (x_{m+1}, y_{m+1}, t_{v_0}, \sigma_{v_0})$, where t_{v_0} is the timestamp and $\sigma_{v_0} = \mathsf{Sign}(sk, x_{m+1}, y_{m+1}, t_{v_0})$.
- In order to delete the node $v_0 \in V$, the signer firstly initializes $newp = \perp$ (which represents the new public label of v_0) to abolish the relationship. Then runs TH's CSF to compute $r_2 = \mathsf{CSF}(x_0||y_0||t_0, r_1, newp||t_{newp})$ such that $F_{pk}(newp||t_{newp}, r_2) = F_{pk}(x_0||y_0||t_0, r_1)$, where (x_0, y_0) is the previous public label of node v_0, t_0 and t_m are previous and recent timestamps respectively, and r_1 is randomly generated when invoking the digital signature MS scheme based on TH to compute the certificate of node v_0. Since the new public label of node v_0 will not be able to pass the TAVerf with other nodes in any equivalence class, the signer successfully deletes v_0 from the current equivalence class without re-signing the node. Additionally, if the above new public label $newp$ is changed to the public label of another equivalence class (e.g. (x_{v_0}, y_{v_0}) is a new generated public label in V_i), then the signer can compute $r_2 = \mathsf{CSF}(x_0||y_0||t_0, r_1, x_{v_0}||y_{v_0}||t_{v_0})$ as above to update v_0 into the new equivalence class V_i without the operation of re-signing .

3) TASign-Edge: We describe the Signing Edge-Updating algorithm as follows.

- The signer can invoke the above Signing Node-Updating algorithm to add the edge (v_i, v_j). The concrete process is the following:
  a) If $v_i \in V_i$ and $v_j \in V_j$, the signer can directly update the public label of node v_j from V_j into V_i.
  b) If either v_i or v_j is not in V, the signer can add the node v_i or v_j into the corresponding equivalence class.
  c) If neither v_i nor v_j is in V, the signer can create a new equivalence class and add the node v_i and v_j into the new equivalence class.
- Since the relationship among the nodes contributes to the existence of edges, the signer can delete the edge (v_i, v_j) by deleting the node v_i or v_j which is the same as the deletion operation in the above Signing Node-Updating algorithm.

- TAVerf: Given a public key tpk, nodes v_i, v_j and the corresponding candidate signatures $(Cert(v_i), Cert(v_j))$, the algorithm parses $Cert(v_i)$ as $(x_{v_i}, y_{v_i}, t_{v_i}, \sigma_{v_i})$ and $Cert(v_j)$ as $(x_{v_j}, y_{v_j}, t_{v_j}, \sigma_{v_j})$. If $\mathsf{Verf}(tpk, xv_i, yv_i, t_{v_i}, \sigma_{v_i}) = 1$, $\mathsf{Verf}(tpk, xv_j, yv_j, t_{v_j}, \sigma_{v_j}) = 1$ and $x(v_i) \cdot y(v_j) = x(v_j) \cdot y(v_i) \ mod\ q$ hold, then return 1; otherwise, return 0. Nodes v_i, v_j are in the same equivalence class if TAVerf outputs 1.

Consistence of TCUGA. Given two certificates $Cert(v_i) = (x_{v_i}, y_{v_i}, t_{v_i}, \sigma_{v_i})$ and $Cert(v_j) = (x_{v_j}, y_{v_j}, t_{v_j}, \sigma_{v_j})$, where $\sigma_{v_i} = \mathsf{Sign}(tsk, x_{v_i}, y_{v_i}, t_{v_i})$ and $\sigma_{v_j} = \mathsf{Sign}(tsk, x_{v_j}, y_{v_j}, t_{v_j})$. If nodes v_i, v_j are in the same equivalence class, then $\mathsf{TVf}(tpk, x_{v_i}, y_{v_i}, t_{v_i}, \mathsf{Sign}(tsk, x_{v_i}, y_{v_i}, t_{v_i}, \sigma_{v_i})) = 1$, $\mathsf{TVf}(tpk, x_{v_j}, y_{v_j}, t_{v_j}, \mathsf{Sign}\ (tsk, x_{v_j}, y_{v_j}, t_{v_j}, \sigma_{v_j})) = 1$ and $\exists r_0 \in \mathbb{Z}_q^*$ so that $x_{v_i} = r_0 \cdot x_{v_j} \ mod\ q$ and $y_{v_i} = r_0 \cdot y_{v_j} \ mod\ q$. Meaning that the following equality holds:

$$x_{v_i} \cdot y_{v_j} = r_0 \cdot x_{v_j} \cdot y_{v_j} \ mod\ q = x_{v_j} \cdot y_{v_i} \ mod\ q.$$

Therefore $Pr[\mathsf{TAVerf}(tpk, v_i, v_j, \mathsf{TASign}(tsk, v_i), \mathsf{TASign}(tsk, v_j)) = 1] = 1$ if and only if nodes v_i, v_j are in the same

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2837650, IEEE Access

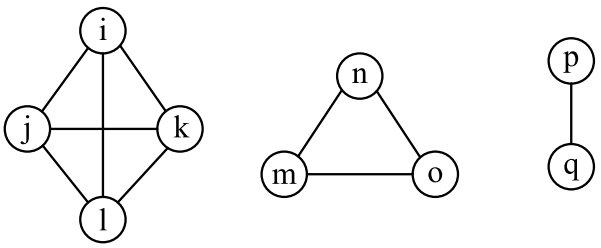

FIGURE 3. An Example of Equivalence Classes Splitting.

equivalence class.

A. TRANSITIVE UNFORGEABILITY

This section relates the transitive unforgeability of our proposal to that of the MS scheme based on trapdoor hash functions.

Theorem 1: (Unforgeability of TCUGA). If the underlying MS is secure, then our proposed TCUGA is unforgeable against an adaptive chosen message forger $\mathcal{F}$ in the standard model.

Proof. Assuming that there exists a $\mathcal{PPT}$ forger $\mathcal{F}$ can break unforgeability of TCUGA with a non-negligible probability $\mathsf{Adv}^{tu-cma}_{\mathcal{F},\mathrm{TCUGA}}(k)$. Then there is a $\mathcal{PPT}$ adversary $\mathcal{A}$ with the public key pk can utilize $\mathcal{F}$ to forge the signature of MS with an advantage $\mathsf{Adv}^{u-cma}_{\mathcal{A},\mathrm{MS}}(k)$.

Here, $\mathcal{A}$ is given pk and the ability of adaptively querying $\mathsf{Sign}(\cdot)$ oracle of MS. $\mathcal{A}$'s aim is to forge a valid message/signature pair (m, σ) without querying m itself to the $\mathsf{Sign}(\cdot)$. Firstly, $\mathcal{A}$ sets $tpk = pk$ and randomly chooses a prime q, then sends (tpk, q) to $\mathcal{F}$. If $\mathcal{F}$ queries the signature of edge (v_i, v_j), then $\mathcal{A}$ executes the process as the real TASign algorithm would, but queries $\mathsf{Sign}(\cdot)$ oracle to generate the signatures. Denote $V' = \{V'_1, V'_2, \ldots, V'_m\}, m = |V'|$ as the queried vertices set, $\triangle$ as a set storing all queried node signatures and T as a table storing all chosen public label. $\mathcal{A}$ does the following:

1) If neither v_i nor v_j is not in V', then $m = m + 1$, $V_{m+1} \leftarrow \{v_i, v_j\}$ and $V' \leftarrow V' \cup V_{m+1}$. Randomly chooses $x_0 \leftarrow \mathbb{Z}^*_q$, $y_0 \leftarrow \mathbb{Z}^*_q$ to obtain the public label (x_0, y_0) of node v_i, where $(x_0, y_0) \notin T$. Then update $T \leftarrow T \cup (x_0, y_0)$ and randomly choose $r_0 \leftarrow \mathbb{Z}^*_q$ and computes $x_1 \leftarrow r_0 \cdot x_0 \ mod \ q, y_1 \leftarrow r_0 \cdot y_0 \ mod \ q$ to obtain the public label (x_1, y_1) of node v_j. Then $\mathcal{A}$ queries $\mathsf{Sign}(\cdot)$ oracle to obtain the signature σ_0 on $x_0||y_0||t_0$ and σ_1 on $x_1||y_1||t_1$ respectively, where t_0, t_1 are the current timestamps. $\triangle \leftarrow \triangle \cup \{Cert(v_i), Cert(v_j)\}$, where $Cert(v_i) = (x_0, y_0, t_0, \sigma_0), Cert(v_j) = (x_1, y_1, t_1, \sigma_1)$.
2) If either v_i or v_j is in V', then $V' \leftarrow V' \cup v_j$ (or $V' \leftarrow V' \cup v_i$). Since v_i (or v_j) has been in one equivalence class of V', we can suppose that the public label of v_i (or v_j) is (x_0, y_0). Randomly chooses $r_0 \leftarrow \mathbb{Z}^*_q$ and computes $x_1 \leftarrow r_0 \cdot x_0 \ mod \ q, y_1 \leftarrow r_0 \cdot y_0 \ mod \ q$ to obtain the public label (x_1, y_1) of node v_j (or v_i). Then $\mathcal{A}$ queries $\mathsf{Sign}(\cdot)$ oracle to obtain the signature σ_1 on $x_1||y_1||t_1$. $\triangle \leftarrow \triangle \cup \{Cert(v_j)\}$, where $Cert(v_j) = (x_1, y_1, t_1, \sigma_1)$.
3) If both v_i and v_j are in V', then $\mathcal{A}$ searches the certificates $Cert(v_i)$ on v_i and $Cert(v_j)$ on v_j from $\triangle$ respectively.
4) $\mathcal{A}$ returns $(Cert(v_i), Cert(v_j))$ to $\mathcal{F}$, where $Cert(v_i) = (x_0, y_0, t_0, \sigma_0), Cert(v_j) = (x_1, y_1, t_1, \sigma_1)$.

Finally, $\mathcal{F}$ forges a signature $(Cert(v^*_i), Cert(v^*_j))$ on edge (v^*_i, v^*_j). Let $G' = (V', E')$ be the graph comprised of the edge and vertice queries from $\mathcal{F}$, and $\widetilde{G'} = (V', \widetilde{E'})$ the transitive closure of G'. $(Cert(v^*_i), Cert(v^*_j))$ is regarded as a valid signature if it satisfies the following:

1) $\mathsf{TAVerf}(v^*_i, Cert(v^*_i), v^*_j, Cert(v^*_j)) = 1$, that is: $\mathsf{Verf}(tpk, x(v^*_i), y(v^*_i), \sigma_{v^*_i}) = 1, \mathsf{Verf}(tpk, x(v^*_j), y(v^*_j), \sigma_{v^*_j}) = 1$ and $x(v^*_i) \cdot y(v^*_j) = x(v^*_j) \cdot y(v^*_i) \ mod \ q$.
2) At least one node is not in $\widetilde{G'}$ between nodes v_i* and v^*_j.

We suppose that v^*_i is not in $\widetilde{G'}$, then $\mathcal{A}$ outputs a solution $(x(v^*_i), y(v^*_i), \sigma_{v^*_i})$ to $\mathcal{A}$'s challenge, and $\forall k \in \mathbb{N}$,

$$\mathsf{Adv}^{u-cma}_{\mathcal{A},\mathrm{MS}}(k) \geq \mathsf{Adv}^{tu-cma}_{\mathcal{F},\mathrm{TCUGA}}(k).$$

With this, we have proved Theorem 1.

B. EVALUATION

Our contribution is mainly proposing an efficient authentication scheme for undirected graphs to support BIMSs, hence we do not consider the implementation of BIMSs. Instead, we focus on specifying the advantages of our proposal and deploying it based on the smart contract to show its feasibility.

Comparison. We compare our proposal with the existing authentication tools for undirected graphs from the view of computation cost. Here, let $\mathbb{G}$ be the group of prime order p, N a modulus used in RSA system or integer factoring theory, S_{ddh} the decision Diffie-Hellman assumption in $\mathbb{G}$ (a gap Diffie-Hellman group). And we use the following abbreviations: let "stand." represent operations involved in the standard signature scheme, "modif." operations of our

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2837650, IEEE Access

TABLE 1. Comparison between our scheme and the existing stateless schemes

Scheme	Signing Cost	Verification Cost	Composition Cost	Signature Size	Member Increase Operation	Member Decrease Operation	Proof of No-existent Edge
DLTS	2 stand.sigs 2 exp. in $\widetilde{\mathbb{G}}$	2 stand.verifs 1 exp. in $\widetilde{\mathbb{G}}$	2 adds in $\mathbb{Z}_q^*$	2 stand.sigs 2 points in $\mathbb{Z}_q$ 2 points in $\widetilde{\mathbb{G}}$	$O(1)$	NO	NO
RSATS-1	2 stand.sigs 2 RSA encs.	2 stand.verifs 1 RSA enc.	$O(\|N\|^2)$ ops.	2 stand.sigs 3 points in $\mathbb{Z}_N^*$	$O(1)$	NO	NO
FactTS-1	2 stand.sigs $O(\|N\|^2)$ ops.	2 stand.verifs $O(\|N\|^2)$ ops.	$O(\|N\|^2)$ ops.	2 stand.sigs 3 points in $\mathbb{Z}_N^*$	$O(1)$	NO	NO
DLTS-1M	2 stand.sigs 1 exp. in $\widetilde{\mathbb{G}}$	2 stand.verifs 1 exp. in $\widetilde{\mathbb{G}}$	1 add in $\mathbb{Z}_q^*$	2 stand.sigs 2 points in $\mathbb{Z}_q$ 1 point in $\widetilde{\mathbb{G}}$	$O(1)$	NO	NO
GapTS-1	2 stand.sigs 1 exp. in $\widetilde{\mathbb{G}}$	2 stand.verifs 1 S_{ddh}	$O(\|N\|^2)$ ops.	1 point in $\widetilde{\mathbb{G}}$	$O(1)$	NO	NO
RSATS-2	1 RSA Dec.	1 RSA enc.	$O(\|N\|^2)$ ops.	1 point in $\mathbb{Z}_N^*$	$O(1)$	NO	NO
FactTS-2	2 sq.roots in $\mathbb{Z}_N^*$	$O(\|N\|^2)$ ops.	$O(\|N\|^2)$ ops.	1 point in $\mathbb{Z}_N^*$	$O(1)$	NO	NO
GapTS-2	1 exp. in $\widetilde{\mathbb{G}}$	1 S_{ddh}	$O(\|N\|^2)$ ops.	1 point in $\widetilde{\mathbb{G}}$	$O(1)$	NO	NO
TGAA-BM	2 exp. in $\mathbb{Z}_N^*$ 2 inv. in $\mathbb{Z}_N^*$	2 exp. in $\mathbb{Z}_N^*$	0	4 points in $\mathbb{Z}_N^*$	$O(N)$	$O(N)$	YES
Our scheme	2 modif.sigs	2 modif.verifs	0	2 modif.sigs 2 points in $\mathbb{Z}_q^*$	$O(1)$	$O(1)$	YES

proposed modified signature scheme, "exp." a modular exponentiation, "inv." the inversion in $\mathbb{Z}_N^*$, "RSA Enc." an operation of RSA encryption, "RSA Dec." an operation of RSA decryption, "sq.root" a square root modulo N, and "ops." the number of bit operations.

As shown in Table 1, existing undirected transitive signature schemes are efficient for Member Increase Operation (i.e. adding edges and vertices), but not for Member Decrease Operation (i.e. deleting edges and vertices). Although TGAA-BM [36] can support both Member Increase Operation and Member Decrease Operation, the price is to recompute all the certificates of nodes in the equivalence class which needs updating edges or vertices. Obviously, our scheme is more efficient in this point, since the signer only need to update the corresponding nodes but all the nodes in our scheme. Moreover, the signer only need to recompute the hashing using his/her trapdoor information for the updating nodes without the need of re-signing the nodes. The other improvement is that our scheme can efficiently prove the No-existent Edge through simply providing the certificates of two queried nodes.

Performance Analysis. We also give the time cost of sub algorithms involved in our TCUGA scheme. Notably, we utilize the RSA digital signature scheme [38] with the modification of hash function replaced by the trapdoor function TH_{DL} (mentioned in Section III-B). And we use the pairing-based library (version 0.5.12)[1] and the gnu multiple precision arithmetic library (version $6.0.0a$)[2] for our simulation. Table 2 shows our testing platform information, and Table 3 gives the concrete results (where the updating algorithm executed in changing memberships only refers to executing TH_{DL} function). According to Table 1 and Table 3, one can find that our TCUGA is more efficient and practical, for which the proposed scheme takes the advantage of conveniently updating memberships (i.e., supporting efficient add or delete operations) in the undirected graph. As a result, our proposal is more promising for being integrated into BIMSs.

[1]http://crypto.stanford.edu/pbc/

[2]https://gmplib.org/

TABLE 2. Testing Platform Information

Operating System	Ubuntu 10.10
CPU	Intel Pentium Processor T4400
Memory	2.00GB RAM
Program language	C

TABLE 3. Time Cost (in s) of Sub Algorithms

Algorithm	TKG	TASign	TAVerf	Updating
Max Time	0.334995	0.062355	0.023519	0.004991
Min Time	0.302238	0.042094	0.013097	0.001443
Average Time	0.314495	0.052955	0.018375	0.002780

Implementation. In order to show the feasibility of our proposal for BIMSs, we implement our proposed scheme on Ethereum [3]. Ethereum is an open-source blockchain system that supports *Solidity* (a special javascript-like language designed for writing smart contracts). Here, the smart contract is defined as a computerized transaction protocol, which executes the terms of a contract permanently recorded in the blockchain. We can regard each contract as a database slot (with a unique address) that one can publish a transaction to invoke the functions in the smart contract (with the corresponding parameters).

We develop Smart Contract on ManageLabel 1 using *Solidity* in a private Ethereum network (without the need of transaction fees but with the same accurate results as that of a public one). The private chain is constructed in our personal computer (see Table 2 for system information). After our design smart contract is published in the private chain, we use *Web3j* (a lightweight library for Java applications) to evaluate the functions. Note that our goal of this implementation is to be compatible with the BIMS environments. However, the existing *Solidity* does not provide the APIs of complex cryptographical operations (e.g. Encryption, Signature) at the time of this research, we only realize the easy label management using smart contracts and still execute the functions of Signature in the external environment.

[3]https://www.ethereum.org/

Algorithm 1 Smart Contract on ManageLabel

Require: Function name, invoked parameters
Ensure: *Setting up functions:*
structure $nodeLabel$
% Define the structure of components in ManageLabel.
x; *% The x-label of a node.*
y; *% The y-label of a node.*
function ManageLabel()
% Constructor, automatically invokes when this smart contract is deployed.
$nodeLabel\ t[]$;
$len = 0$;
return 1;
function newClass(x, y)
% Invoked by the signer to add a new class.
$tmp.x = x$;
$tmp.y = y$;
$t.push(tmp)$;
return 1;
function genLabel(i)
% Invoked by the signer to generate a new label pair for the node.
if $i \in [0, t.length)$ **then**
r = Random();
$tmp.x = r * t[i].x$;
$tmp.y = r * t[i].y$;
return tmp;
else
return 0;
function VerifyClass($nodeLabel\ A$, $nodeLabel\ B$)
% Invoked by the nodes to verify whether two nodes are in the same equivalence class or not.
if $A.x * B.y == A.y * B.x$ **then**
return 1;
else
return 0;

VI. CONCLUSION AND FUTURE RESEARCH

Blockchain-based applications will be increasingly popular, and correspondingly there will be new requirements in such applications. In this paper, we studied the membership authentication requirement in BIMS. Specifically, we designed a novel transitively closed undirected graph authentication scheme for BIMS. In comparison to other competing authentication schemes for undirected graph, our proposal is more efficient in terms of the capability to dynamically add or delete nodes and edges. Moreover, our scheme can efficiently solve the authentication of non-existent edges, which is a known challenge in transitive signature schemes. We also provided the security analysis in the standard model and the performance evaluation.

In our scheme, however, the requestors may receive several certificates of one node, among which old certificates (that are still valid - i.e. can successfully pass the verification) may be utilized to trick other users. Therefore, one potential future extension is to integrate designated verifier signatures in our scheme to prevent the malicious propagation of nodes' relationship certificates.

REFERENCES

[1] Y. Hu, A. Manzoor, P. Ekparinya, M. Liyanage, K. Thilakarathna, G. Jourjon, A. Seneviratne, and M. E. Ylianttila, "A delay-tolerant payment scheme based on the ethereum blockchain," CoRR, vol. abs/1801.10295, 2018.

[2] I. Miers, C. Garman, M. Green, and A. D. Rubin, "Zerocoin: Anonymous distributed e-cash from bitcoin," in 2013 IEEE Symposium on Security and Privacy, SP 2013, Berkeley, CA, USA, May 19-22, 2013, pp. 397–411, IEEE Computer Society, 2013.

[3] E. Ben-Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza, "Zerocash: Decentralized anonymous payments from bitcoin," in 2014 IEEE Symposium on Security and Privacy, SP 2014, Berkeley, CA, USA, May 18-21, 2014, pp. 459–474, IEEE Computer Society, 2014.

[4] J. Sidhu, "Syscoin: A peer-to-peer electronic cash system with blockchain-based services for e-business," in 26th International Conference on Computer Communication and Networks, ICCCN 2017, Vancouver, BC, Canada, July 31 - Aug. 3, 2017, pp. 1–6, IEEE, 2017.

[5] H. Kopp, D. Mödinger, F. Hauck, F. Kargl, and C. Bösch, "Design of a privacy-preserving decentralized file storage with financial incentives," in 2017 IEEE European Symposium on Security and Privacy Workshops, EuroS&P Workshops 2017, Paris, France, April 26-28, 2017, pp. 14–22, IEEE, 2017.

[6] H. Kopp, C. Bösch, and F. Kargl, "Koppercoin - A distributed file storage with financial incentives," in Information Security Practice and Experience - 12th International Conference, ISPEC 2016, Zhangjiajie, China, November 16-18, 2016, Proceedings (F. Bao, L. Chen, R. H. Deng, and G. Wang, eds.), vol. 10060 of Lecture Notes in Computer Science, pp. 79–93, 2016.

[7] S. Huh, S. Cho, and S. Kim, "Managing iot devices using blockchain platform," in International Conference on Advanced Communication Technology, pp. 464–467, 2017.

[8] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "LSB: A lightweight scalable blockchain for iot security and privacy," CoRR, vol. abs/1712.02969, 2017.

[9] Y. Zhang, S. Kasahara, Y. Shen, X. Jiang, and J. Wan, "Smart contract-based access control for the internet of things," CoRR, vol. abs/1802.04410, 2018.

[10] X. Liang, S. Shetty, D. K. Tosh, C. A. Kamhoua, K. A. Kwiat, and L. Njilla, "Provchain: A blockchain-based data provenance architecture in cloud environment with enhanced privacy and availability," in Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, CCGRID 2017, Madrid, Spain, May 14-17, 2017, pp. 468–477, IEEE Computer Society / ACM, 2017.

[11] Q. Xia, E. B. Sifah, K. O. Asamoah, J. Gao, X. Du, and M. Guizani, "Medshare: Trust-less medical data sharing among cloud service providers via blockchain," IEEE Access, vol. 5, pp. 14757–14767, 2017.

[12] P. Dunphy and F. A. P. Petitcolas, "A first look at identity management schemes on the blockchain," CoRR, vol. abs/1801.03294, 2018.

[13] S. Raju, S. Boddepalli, S. Gampa, Q. Yan, and J. S. Deogun, "Identity management using blockchain for cognitive cellular networks," in IEEE International Conference on Communications, ICC 2017, Paris, France, May 21-25, 2017, pp. 1–6, IEEE, 2017.

[14] S. Muftic, "Blockchain identity management system based on public identities ledger," April 2017. Patent, 9635000.

[15] Q. Lin, H. Yan, Z. Huang, W. Chen, J. Shen, and Y. Tang, "An id-based linearly homomorphic signature scheme and its application in blockchain," IEEE Access.

[16] R. Mercer, "Privacy on the blockchain: Unique ring signatures," CoRR, vol. abs/1612.01188, 2016.

[17] S. Ma, Y. Deng, D. He, J. Zhang, and X. Xie, "An efficient NIZK scheme for privacy-preserving transactions over account-model blockchain," IACR Cryptology ePrint Archive, vol. 2017, p. 1239, 2017.

[18] Y. Zhang, X. Chen, J. Li, D. S. Wong, H. Li, and I. You, "Ensuring attribute privacy protection and fast decryption for outsourced data security in mobile cloud computing," Inf. Sci., vol. 379, pp. 42–61, 2017.

[19] X. Chen, J. Li, J. Weng, J. Ma, and W. Lou, "Verifiable computation over large database with incremental updates," IEEE Trans. Computers, vol. 65, no. 10, pp. 3184–3195, 2016.
[20] X. Chen, X. Huang, J. Li, J. Ma, W. Lou, and D. S. Wong, "New algorithms for secure outsourcing of large-scale systems of linear equations," IEEE Trans. Information Forensics and Security, vol. 10, no. 1, pp. 69–78, 2015.
[21] S. Hou, X. Huang, J. K. Liu, J. Li, and L. Xu, "Universal designated verifier transitive signatures for graph-based big data," Inf. Sci., vol. 318, pp. 144–156, 2015.
[22] A. Shamir and Y. Tauman, "Improved online/offline signature schemes," in Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19-23, 2001, Proceedings (J. Kilian, ed.), vol. 2139 of Lecture Notes in Computer Science, pp. 355–367, Springer, 2001.
[23] S. Micali and R. L. Rivest, "Transitive signature schemes," in Preneel [40], pp. 236–243.
[24] M. Bellare and G. Neven, "Transitive signatures: new schemes and proofs," IEEE Transactions on Information Theory, vol. 51, no. 6, pp. 2133–2151, 2005.
[25] C. Lin, F. Zhu, W. Wu, K. Liang, and K. R. Choo, "A new transitive signature scheme," in Network and System Security - 10th International Conference, NSS 2016, Taipei, Taiwan, September 28-30, 2016, Proceedings (J. Chen, V. Piuri, C. Su, and M. Yung, eds.), vol. 9955 of Lecture Notes in Computer Science, pp. 156–167, Springer, 2016.
[26] S. F. Shahandashti, M. Salmasizadeh, and J. Mohajeri, "A provably secure short transitive signature scheme from bilinear group pairs," in Security in Communication Networks, 4th International Conference, SCN 2004, Amalfi, Italy, September 8-10, 2004, Revised Selected Papers (C. Blundo and S. Cimato, eds.), vol. 3352 of Lecture Notes in Computer Science, pp. 60–76, Springer, 2004.
[27] C. Ma, P. Wu, and G. Gu, "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, Harbin, China, January 16-18, 2006, Proceedings (H. T. Shen, J. Li, M. Li, J. Ni, and W. Wang, eds.), vol. 3842 of Lecture Notes in Computer Science, pp. 897–904, Springer, 2006.
[28] L. Wang, Z. Cao, S. Zheng, X. Huang, and Y. Yang, "Transitive signatures from braid groups," in Progress in Cryptology - INDOCRYPT 2007, 8th International Conference on Cryptology in India, Chennai, India, December 9-13, 2007, Proceedings (K. Srinathan, C. P. Rangan, and M. Yung, eds.), vol. 4859 of Lecture Notes in Computer Science, pp. 183–196, Springer, 2007.
[29] Z. Gong, Z. Huang, W. Qiu, and K. Chen, "Transitive signature scheme from lfsr," Journal of Information Science and Engineering, vol. 26, no. 1, pp. 131–143, 2010.
[30] R. L. Rivest and S. R. Hohenberger, "The cryptographic impact of groups with infeasible inversion," Masterąfs Thesis Mit, 2003.
[31] X. Yi, "Directed transitive signature scheme," in Topics in Cryptology - CT-RSA 2007, The Cryptographers' Track at the RSA Conference 2007, San Francisco, CA, USA, February 5-9, 2007, Proceedings (M. Abe, ed.), vol. 4377 of Lecture Notes in Computer Science, pp. 129–144, Springer, 2007.
[32] G. Neven, "A simple transitive signature scheme for directed trees," Theor. Comput. Sci., vol. 396, no. 1-3, pp. 277–282, 2008.
[33] J. Xu, "On directed transitive signature," IACR Cryptology ePrint Archive, vol. 2009, p. 209, 2009.
[34] P. Camacho and A. Hevia, "Short transitive signatures for directed trees," IACR Cryptology ePrint Archive, vol. 2011, p. 438, 2011.
[35] R. Johnson, D. Molnar, D. X. Song, and D. Wagner, "Homomorphic signature schemes," in Preneel [40], pp. 244–262.
[36] C. Ma, P. Wu, L. Wang, and G. Zhang, "A novel method to authenticate transitively closed undirected graph," in Proceeding of the Second International Multi-Symposium of Computer and Computational Sciences (IMSCCS 2007), August 13-15, 2007, The University of Iowa, Iowa City, Iowa, USA, pp. 537–542, IEEE Computer Society, 2007.
[37] F.-Y. Yang et al., "Efficient trapdoor hash function for digital signatures," Chaoyang Journal, vol. 12, pp. 351–357, 2007.
[38] S. Goldwasser, S. Micali, and R. L. Rivest, "A digital signature scheme secure against adaptive chosen-message attacks," SIAM J. Comput., vol. 17, no. 2, pp. 281–308, 1988.
[39] J. Katz and Y. Lindell, Introduction to Modern Cryptography. Chapman and Hall/CRC Press, 2007.
[40] B. Preneel, ed., Topics in Cryptology - CT-RSA 2002, The Cryptographer's Track at the RSA Conference, 2002, San Jose, CA, USA, February 18-22, 2002, Proceedings, vol. 2271 of Lecture Notes in Computer Science, Springer, 2002.

CHAO LIN received his Bachelor and Master degrees from the School of Mathematics and Computer Science, Fujian Normal University in 2013 and 2017, respectively. Currently, he is pursuing his Ph.D. degree in School of Cyber Science and Engineering, Wuhan University. His research interests mainly include authentication of graph data and blockchain security.

DEBIAO HE received his Ph.D. degree in applied mathematics from School of Mathematics and Statistics, Wuhan University in 2009. He is currently a Professor of the School of Cyber Science and Engineering, Wuhan University. His main research interests include cryptography and information security, in particular, cryptographic protocols.

XINYI HUANG received the PhD degree from University of Wollongong, Australia. He is currently a Professor with the School of Mathematics and Computer Science, Fujian Normal University, China, and the Co-Director of Fujian Provincial Key Laboratory of Network Security and Cryptology. He is an Associate Editor of the IEEE Transactions on Dependable and Secure Computing. He serves on the Editorial Board of International Journal of Information Security (IJIS, Springer), and has served as the Program/General Chair or Program Committee Member in over 80 international conferences. His research interests include applied cryptography and network security.

MUHAMMAD KHURRAM KHAN is currently a Full Professor with the Center of Excellence in Information Assurance, King Saud University, Saudi Arabia. He is also an Adjunct Professor with Fujian University of Technology, China, and an honorary Professor with IIIRC, Shenzhen Graduate School, Harbin Institute of Technology, China. He has published over 325 research papers in the journals and conferences of international repute. He has edited seven books/proceedings published by Springer-Verlag and the IEEE. He is an inventor of 10 U.S./PCT patents. He is a fellow of the IET, U.K., BCS, U.K., and FTRA, South Korea, and a member of the IEEE Technical Committee on Security and Privacy and the IEEE Cybersecurity Community. He received the Outstanding Leadership Award from the IEEE International Conference on Networks and Systems Security 2009, Australia, the Gold Medal for the Best Invention, the Innovation Award from the 10th Malaysian Technology Expo 2011, Malaysia, and the Best Paper Award from the Journal of Network and Computer Applications (Elsevier) in 2015. He is one of the organizing chairs of over five dozen international conferences and a member of technical committees of over 10 dozen international conferences. He is the Editor-in-Chief of a well-esteemed international journal Telecommunication Systems (Springer-Verlag, 1993) with an impact factor of 1.542 (JCR 2016). He is also a full-time Editor/Associate Editor of several international journals/magazines, including the IEEE Communications Surveys and Tutorials, IEEE Communications Magazine, the Journal of Network and Computer Applications (Elsevier), the IEEE Transactions on Consumer Electronics, the IEEE Access, Security and Communication Networks, IEEE Consumer Electronics Magazine, PLOS ONE, IET Wireless Sensor Systems, Electronic Commerce Research (Springer), the Journal of Information Hiding and Multimedia Signal Processing, the International Journal of Biometrics (Inderscience).

KIM-KWANG RAYMOND CHOO (SM'15) received his Ph.D. in Information Security in 2006 from Queensland University of Technology, Australia. He currently holds the Cloud Technology Endowed Professorship at The University of Texas at San Antonio (UTSA). In 2016, he was named the Cybersecurity Educator of the Year - APAC (Cybersecurity Excellence Awards are produced in cooperation with the Information Security Community on LinkedIn), and in 2015 he and his team won the Digital Forensics Research Challenge organized by Germany's University of Erlangen-Nuremberg. He is the recipient of 2018 UTSA College of Business Col. Jean Piccione and Lt. Col. Philip Piccione Endowed Research Award for Tenured Faculty, ESORICS 2015 Best Paper Award, 2014 Highly Commended Award by the Australia New Zealand Policing Advisory Agency, Fulbright Scholarship in 2009, 2008 Australia Day Achievement Medallion, and British Computer Society's Wilkes Award in 2008. He is also a Fellow of the Australian Computer Society.