A Proxy Transitive Signature Scheme

Fei Zhu[1], Xiaoling Tao[2], Chao Lin[1], Wei Wu[1]

[1] Fujian Provincial Key Laboratory of Network Security and Cryptology, School of Mathematics and Computer Science, Fujian Normal University, China

[2] Guangxi Cooperative Innovation Center of cloud computing and Big Data, Guangxi Colleges and Universities Key Laboratory of cloud computing and complex systems, Guilin University of Electronic Technology, China

feizcscoding@163.com, txl@guet.edu.cn, lcfjnu@foxmail.com, weiwu81@gmail.com

Abstract

Motivated by the research of privacy protection in the graph-based big data system, we propose a new signature scheme. The properties of the constructed signature scheme are twofold. On one hand, it is consistent with the requests of the proxy signatures. Our method allows a signer to delegate his/her signing right to another user in such a way that the latter can validly sign messages on behalf of the former, and other parties (who are not designated as the proxy signer) cannot create a valid proxy signature. On the other hand, it satisfies the properties of transitive signatures. One of the fundamental but vital properties is composability, which means that given two signatures σ_{ij} and σ_{jk} on adjacent edges (i, j) and (j,k), anyone without knowing the signer's secret key can easily derive the digital signature σ_{ik} of the edge (i,k). This property allows the signer to authenticate a graph in a cost-saving manner. Actually, our proposed design meets all the features needed for proxy signatures and transitive signatures, and it is provably secure in the random oracle model.

Keywords: Proxy signatures, Transitive signatures, Privacy

1 Introduction

Our world today is further connected than ever before, and the connections produce an increasingly large amount of data from our activities, movements, preferences, and relationships. The terminology "Big data" is used to capture such massive-data-conglomeration phenomenon. In spite of the economic and social benefits that big data offers, the concerns on privacy should not be neglected. According to the research "The ITRC Breach Report" published by the Identity Theft Resource Center (ITRC) in December 6th, 2016, the number of total data breaches have reached up to 957 and more than 35 million individual records have been exposed in the year of 2016 [1]. To address these concerns, it should be ensured that data are collected and transmitted securely, and they are accessed only by authorized parties. Namely, security should be a key element when designing the big data system.

In certain scenarios, such as administrative domains, certificate chains in PKI and military chains of command, one needs to publish a graph (which is used to represent the structures) in an authenticated manner. This study focus on the problem of the authentication in the graph-based big data system. As raised by Micali and Rivest [2], a specific solution for authenticating a graph is called as the transitive signature, which permits a signer to authenticate a dynamically growing and naturally transitive graph edge by edge. According to the graph's characteristics, i.e., directed or undirected, the scheme can be divided into directed transitive signature scheme and undirected transitive signature scheme. Actually, a transitive signature scheme differs from the standard digital signature in that the former supports the properties of transitive and unforgeability. That is:

(1) Given two signatures σ_{ij} and σ_{jk} on adjacent edges (i, j) and (j,k) respectively, anyone with the signer's public key can easily derive a signature σ_{ik} for the composed edge (i,k), where i, j and k are vertices in a graph and σ_{ik} is the equivalent of the signature that the legitimate signer will produce;

(2) It should be difficult for an adaptive chosen-message adversary to forge a valid signature for an edge not in the transitive closure of edges, even after the adversary has adaptively queried signatures of a great many vertices and edges of his choice.

It is demonstrated that the overall signing complexity and the communication overhead of the transitive signature scheme can be significantly reduced, which is important in the big data paradigm (please refer to [3-4] for more details).

Let's consider another scenario. In this scenario, the original signer has no time (or for some other reasons)

*Corresponding Author: Wei Wu; E-mail: weiwu81@gmail.com
DOI: 10.3966/160792642018081904029

to authenticate the nodes in the transitive graph. As an option, the signer can utilize an authentication scheme in a "proxy" environment to solve the above problem securely. Proxy signature was introduced by Mambo et al. for the scenario when a signer intends to share his/her authenticating capability with another signer [5]. Once the delegation is fulfilled, the proxy signer is able to authenticate the graph as the representative of the original signer. Some properties of proxy signatures can be abstract as follows [6]:

(1) Unforgeability. Only the designated proxy signer can validly create a proxy signature for the original signer;

(2) Verifiability. After verification, the verifier is assured of the original signer's agreement on the authen- ticated message;

(3) Undeniability. As soon as the proxy signer validly generates the proxy signature, he cannot disaffirm his signature creation against anyone;

(4) Distinguishability. It is easy to distinguish the proxy signature from the original signature;

(5) Proxy signer's deviation. The proxy signature generated by the proxy signer must be valid.

1.1 Motivation

As introduced in [2, 6], transitive signatures for an undirected graph can be used to authenticate a computer administrative domain efficiently. Located in the same administrative domain, computers can exchange data communications securely between each other. As mentioned previously, we are interested in such a scenario that an original signer (named Alice) needs to authenticate a graph-based computer administrative domain to show the relationships between every two members, where vertices represent computers and an undirected edge (i, j) means that i and j are in the same administrative domain. Due to lack of time (or for some other reasons), Alice wishes to delegate her signing right to the other user (named Bob). The above requirements can be met by our Proxy Transitive Signature scheme, which will be introduced in Section 4.

1.2 Our Contributions

Considering the privacy concerns in the graph-based big data system, in this paper, we combine the concepts of proxy signatures with transitive signatures, and propose a new signature scheme called Proxy Transitive Signature (PTS). Specifically, in our PTS scheme, the original transitive signer Alice is capable of using her secret key to generate a witness signature (i.e., a warrant that specifies the authorization information). Then, she sends the signature to the proxy signer Bob directly. Upon receiving the witness signature, Bob verifies it and produces a proxy public/secret key pair using his secret key. Completing these operations, Bob obtains the same signing capability as that of Alice.

We put forward the model of PTS and provide a concrete construction. Additionally, we prove the security of our design in the random oracle model. The security analysis shows that our PTS scheme can efficiently address privacy concerns in the aforementioned scenario associated with dissemination of transitive signatures.

1.3 Organization

The remainder of this paper is organized as follows. Section 2 reviews the related works. Section 3 introduces some preliminaries required in this study. Section 4 provides formal definitions of PTS and the security models. In Section 5, we present our proposed PTS scheme and give its security/performance analysis. Finally, Section 6 concludes this study.

2 Related Works

In this section, we review two types of signature schemes, that is, proxy signature schemes and transitive signature schemes.

2.1 Proxy Signatures

In 1996, since the notion of proxy signature was firstly introduced by Mambo et al. [5], many proxy signature schemes have been proposed. Specifically, proxy signature schemes are designed for the scenarios when one person intends to share his/her signing capability with another. According to the types of delegation, proxy signatures could be break down into three categories: full delegation, partial delegation and delegation by warrant [5, 7]. In full delegation, Alice's private key is given to Bob through a secure channel so that Bob has the same signing capability as Alice. In practice, however, the full delegation schemes are impractical and insecure. And one can find that it is difficult to prevent the signature from being misused. In a partial delegation scheme, Alice uses her secret key to generate a proxy secret key and sends it to Bob. Receiving the proxy secret key, Bob can generate proxy signatures. However, the proxy signatures generated by Bob are different from these original signatures generated by Alice. Additionally, this method can be applied in the following two scenarios, proxy-protected proxy signature and proxy-unprotected proxy signature. In a proxy-protected scheme, only the proxy signer Bob can generate proxy signatures, while in a proxy-unprotected scheme both Bob and the original signer Alice can generate proxy signatures because both of them have the knowledge of the proxy secret key. In many applications, proxy-protected schemes are required to avoid the potential disputes between the original signer and the proxy signer. Moreover, according to the various demands, partial delegation schemes can be combined with other types

of signatures to obtain new types of proxy signatures, e.g., a nonrepudiable threshold proxy signature scheme introduced by Sun [8], a proxy multi-signature scheme proposed by Yi et al. [9], proxy blind signature schemes proposed by Tan et al. [10-11], controllable proxy signature scheme proposed by Liu and Liu [12]. Later, there are numerous other properties of proxy signature schemes have also been proposed, such as proxy signature schemes with forward secure [13-14], design- ated-receiver [15], shared verification [16], anonymous proxy [17] and so on. In such partial delegation schemes, however, there is no limit on the scope of the messages to be signed. This limitation is eliminated in delegation by warrant schemes. In such a scheme, the warrant is added to specify what kind of messages are delegated, and the warrant may contain the identities of Alice and Bob, the delegation period, etc. [18-19].

In fact, the boundary between partial delegation schemes and delegation by warrant schemes is not rigid [6]. While the latter could demarcate the scope of messages to be signed, the former has characteristics of faster processing speed. Combining the merits of both, Kim et al. introduced the concept of partial delegation by warrant, and proposed a threshold proxy signature [20]. Lee et al. [21] discussed whether a secure channel for transmitting the signed warrant is necessary in existing schemes, and pointed out that [5, 22-23] are all insecure if the secure channel is not provided. To remove the requirement of a secure channel and overcome some other weaknesses, they revised these schemes (please refer to [21] for details).

As a matter of fact, new security considerations and constructions have been proposed. As a result, the security of some existing schemes can be violated. Motivated by these observations, other proxy signature schemes, such as [24-27], are proposed based on new constructions.

2.2 Transitive Signatures

In 2002, the idea of "transitive signature" was introduced by Micali and Rivest in [2]. The transitive signature scheme DLTS is designed based on the hardness of discrete logarithm problem. Additionally, it is proved that the DLTS is transitively unforgeable under adaptive chosen-message attacks. Likewise, they presented another scheme RSATS-1 which is designed based on the RSA assumption. Additionally, it is proved that the scheme is secure against the non-adaptive chosen-message attacker. Motivated by the pioneering work of Micali and Rivest, several new schemes were developed on the basis of the difficulty of certain assumptions, such as, one-more RSA - inversion assumption, factoring assumption, one- more discrete logarithm assumption and one-more gap Diffie-Hellman assumption., i.e., FactTS-1 , DLTS-1M and GapTS-1 were proven to be transitively unforgeable under adaptive chosen-message attacks in the standard model [4, 28]. Meanwhile, Bellare and Neven eliminated node certificates for some of the above schemes by specifying the public label of a node i as the output of a hash function applied to i and constructed RSATS-2 , FactTS-2 and GapTS-2 [4]. But they merely proved their security in the random oracle model. Then, by dint of the random oracle model, Wang et~al. firstly proposed a transitive signature scheme on the basis of braid groups [29]. Gong et al. constructed a transitive signature scheme LFSR-TS from Linear Feedback Sequence Register [30].

The above transitive signature schemes are only applicable to undirected graphs, no directed transitive signature schemes have emerged since being envisioned by Micali and Rivest [2]. Actually, theoretical analysis shows that the general directed transitive signature schemes may be exceedingly hard to construct [31]. It is due to the reason that its edge signatures form a special Abelian trapdoor group, whose existence is still unkown. Upon this point, the research of the directed transitive signature schemes was confined to a case that the directed graph was a directed tree. Whereafter, in 2007, Yi provided a scheme RSADTS based on the hardness of RSA - inversion problem in the standard model [32]. In 2008, Neven [33] presented a conceptually simple and generic construction of a transitive signature scheme for directed trees from any standard signature scheme [3, 28]. Specifically, the construction was designed without using any RSA -related assumptions, and it is more efficient than Yi's scheme. The drawbacks of [32] and [33] include that the size of composed signature increases linearly with the number of recursive applications of composition, and the creating history of composed edge is not hidden properly. To address these issues, Jia Xu raised a directed transitive signature scheme DTTS in 2009 [34]. It has a constant signature size and achieves privacy preserving property. Additionally, it is provably secure in the standard model. Later, also aimed at directed trees, Camacho and Hevia exploited a late-model collision-resistant hash function and came up with a novel practical transitive signature scheme, which is the most effective scheme so far [35].

3 Preliminaries

3.1 Notations

Let $\mathbb{Z}_q$ denote the definite field of the prime order q. Let $\mathbb{N}$ denote the set of positive integers and $\|$ the concatenation operator on strings. A function $f(\cdot):\mathbb{N} \to \mathbb{R}$ is negligible if for every positive integer c, there exists an integer k_0 such that for all $k > k_0$,

$|f(k)|<1/k^c$. We remark that $\mathcal{PPT}$ is the abbreviation of probabilistic polynomial time. The notation $x \xleftarrow{\$} \mathcal{X}$ characterized as x is sampled from set $\mathcal{X}$ uniformly and randomly. We denote by $H_w(\cdot):\{0,1\}^* \times \mathbb{G} \to \mathcal{Z}_q^*$ the hash function while $H(\cdot):\mathbb{N} \to \mathcal{Z}_N^*$ is a collision-resistant hash function.

3.2 Graphs

This paper considers $G=(V,E)$ is an undirected graph with vertexes set V and edges set E. We implicitly assume that $V=\{1,2,...,n\}$ and (i,j) represent an edge in G, and $\tilde{G}=(V,\tilde{E})$ is the transitive closure of G iff there is a path from i to j in G. Its transitive reduction graph $G^*=(V,E^*)$ is defined to have the minimum subset of edges with the same transitive closure as G.

3.3 Bilinear Mapping

Suppose that $\mathbb{G}$ and $\mathbb{G}_T$ are two multiplicative cyclic groups of prime order q, let g be a generator of $\mathbb{G}$. We define an admissible bilinear pairings $\hat{e}:\mathbb{G}\times\mathbb{G}\to\mathbb{G}_T$ satisfies the following three properties:

- Non-degenerate: $\hat{e}(g,g)\neq 1$.
- Bilinear: for all $m,n\in\mathbb{Z}_q^*$, $\hat{e}(g^m,g^n)=\hat{e}(g,g)^{mn}$.
- Computable: for all $m,n\in\mathbb{Z}_q^*$, $\hat{e}(g^m,g^n)$ can be efficiently computed.

3.4 Complexity Problems and Security Assumptions

- Discrete Logarithm problem (DL): Given g, $g^m\in\mathbb{G}$, find $m\in\mathbb{G}$, where $m\xleftarrow{\$}\mathbb{Z}_q^*$.
- Computational Diffie-Hellman problem (CDH): Given $g,g^m,g^n\in\mathbb{G}$, compute $g^{mn}\in\mathbb{G}$, where $m,n\xleftarrow{\$}\mathbb{Z}_q^*$.
- Bilinear Diffie-Hellman problem (BDH): Given $g,g^l,g^m,g^n\in\mathbb{G}$, compute $\hat{e}(g,g)^{lmn}\in\mathbb{G}_T$, where $l,m,n\xleftarrow{\$}\mathbb{Z}_q^*$.

Since a multitude of hard mathematic problems are extensively used in analyzing and proving the security of cryptographic schemes, we will formally define two security assumptions about the above problems as follows.

Definition 1. (*DL* assumption). Fix a q-order cyclic group $\mathbb{G}$ and a generator $g\in\mathbb{G}$, given an element $h\in\mathbb{G}$ uniformly. The $\mathcal{PPT}$ adversary $\mathcal{F}$ is given and output $m\in\mathbb{G}$ such that $h=g^m$.

The adversary $\mathcal{F}$ wins if it outputs the valid $m\in\mathbb{G}$ eventually.

Definition 2. (one-more *BDH* assumption). Let $l,m \xleftarrow{\$} \mathbb{Z}_q$ and let $y_l=g^l, y_m=g^m$. The adversary $\mathcal{F}$ is given $(\hat{e},\mathbb{G},\mathbb{G}_T,g,q,y_l,y_m)$ and has access to a Challenge oracle $\mathcal{O}_{\mathcal{CH}}(\cdot)$ and a *CDH* oracle $\mathcal{O}_{\mathcal{CDH}}(\cdot)$.

- $\mathcal{O}_{\mathcal{CH}}(\cdot)$: when invoked, it returns a random challenge point from $\mathbb{G}$.
- $\mathcal{O}_{\mathcal{CDH}}(\cdot)$: given a random point $h_i\xleftarrow{\$}\mathbb{G}$ as input, outputs $(h_i)^l\in\mathbb{G}$.

The adversary $\mathcal{F}$ wins if it succeeds in inverting *BDH* solutions $\hat{e}(g,h_i)^{lm}\in\mathbb{G}_T$ with all s points $h_i\in\mathbb{G}$ output by the Challenge oracle, but using strictly less than s queries to the *CDH* oracle.

4 Proxy Transitive Signature (PTS)

This section provides the formal definition of Proxy Transitive Signature (PTS).

4.1 Syntax of PTS

Let Alice (abbreviated as A) represents the original signer and Bob (abbreviated as B) is the proxy signer. A Proxy Transitive Signature scheme PTS consists of the following algorithms: PTS = (CPG,CKG,WS,WV, PKG,PTSign,PTVf,PComp).

- CPG: Common Parameter Generation algorithm. It takes the security parameter 1^k as input and returns a common parameter *cp* publicly. That is: $cp\leftarrow$ CPG(1^k).
- CKG: Key Generation algorithm. It takes the common parameter *cp* as input and generates key pair (x_i,y_i) where $i\in\{A,B\}$. That is: $(x_i,y_i)\leftarrow$ CKG(*cp*).
- WS: Witness Signing algorithm is utilized by A. It takes $ID_A,ID_B,m_w,r_A,x_A,y_B$ as input and returns a witness signature s_A. That is $s_A\leftarrow$ WS(cp,ID_A, ID_B,m_w,r_A,x_A,y_B), m_w is the warrant that specifies what kinds of messages are delegated, ID_A and ID_B are the identities of A and B, $r_A\in\mathbb{G}$ is a secret label which is randomly chosen by A.
- WV: Witness Verifying algorithm is utilized by B to check the validity of the witness signature. It takes as input $y_A,R_A,y_B,ID_A,ID_B,m_w$ and outputs $f\in\{0,1\}$. That is $\{0,1\}\leftarrow$ WV(s_A,ID_A,ID_B,m_w,R_A, y_A,y_B). Here R_A is a public label which is related to r_A.
- PKG: Proxy Key Generation algorithm. B uses this algorithm, and takes s_A,x_B as input, the algorithm gene- rates key pair (x_p,y_p). That is $(x_p,y_p)\leftarrow$ PKG (s_A,x_B).

- PTSign: Proxy Transitive Signing algorithm. B takes proxy private key x_p, nodes $i, j \in \mathbb{N}$ as input, the algorithm generates a proxy signature σ_{ij} of edge (i, j). That is $\sigma_{ij} \leftarrow \text{PTSign}(x_p, i, j)$.
- PTVf: Proxy Signature Verification algorithm. Anyone can utilize it to check the validity of the proxy signature. It takes y_A, R_A, y_B, nodes $i, j \in \mathbb{N}$ and a candidate proxy signature σ_{ij} as input, returns $f \in \{0,1\}$. That is $\{0,1\} \leftarrow \text{PTVf}(cp, ID_A, ID_B, m_w, y_A, R_A, y_B, \sigma_{ij}, i, j)$.
- PComp: Proxy Transitive Signature Composition algorithm. The algorithm takes y_p, nodes $i, j, k \in \mathbb{N}$ and corresponding proxy transitive signatures σ_{ij}, σ_{jk} as input, outputs either a composed signature σ_{ik} of edge (i,k) or $\perp$ to be indicative of failure. That is: $\{\sigma_{ik}, \perp\} \leftarrow \text{PComp}(y_p, \sigma_{ij}, \sigma_{jk}, i, j, k)$.
- Consistency of PTS: Concerning these algorithms, we require three extra consistency requirements in PTS.
- WV Consistency of WS: The witness signature generated by the WS algorithm should be valid by the WV algorithm: $Pr[\text{WV}(s_A, ID_A, ID_B, m_w, R_A, y_A, y_B, \text{WS}(cp, ID_A, ID_B, m_w, r_A, x_A, y_B)) = 1] = 1$.
- PTVf Consistency of PTSign: The proxy transiti- ve signature generated by the PTSign algorithm should be valid by the PTVf algorithm: $Pr[\text{PTVf}(cp, ID_A, ID_B, m_w, y_A, R_A, y_B, \text{PTSign}(x_p, i, j)) = 1] = 1$.
- PTVf Consistency of PComp: Given two legitim- ate signatures (those obtained from the signer or through composition of legitimate signatures [4]) σ_{ij} of edge (i, j) and σ_{jk} of edge (j,k), the proxy transitive signature generated by the PComp algorithm should be valid by the PTVf algorithm: $Pr[\text{PTVf}(cp, ID_A, ID_B, m_w, y_A, R_A, y_B, \text{PComp}(y_p, \sigma_{ij}, \sigma_{jk}, i, j, k)) = 1] = 1$.

4.2 Security requirements of PTS

This section defines security requirements, such as unforgeability and privacy, which should be met by the PTS scheme.

4.2.1 Unforgeability

There are two types of unforgeability in our proposed proxy transitive signature scheme PTS [36-38].

Initially, we remark it as proxy signature unforgeability (WV -Unforgeability). There exits three types of adversaries as follows:

- Type I : The adversary can obtain the public keys of Alice and Bob.
- Type II : The adversary possesses the secret/public key pair of Alice and Bob's public key.
- Type III : The adversary possesses the secret/public key pair of Bob and Alice's public key.

As one can see that our proposed scheme PTS is unforgeable against Type I adversary if it is unforgeable against Type II or Type III adversary.

Secondly, any adaptive chosen-message adversary cannot forge a valid transitive signature. And we simply regard it as transitive signature unforgeability (PTVf - Unforgeability).

In this case, three attack models related to WV - Unforgeability and PTVf -Unforgeability will be given out.

Type II adversary: Note this type of adversary $\mathcal{F}_{II}$ as a malicious proxy signer. He aims to forge a witness signature of a valid warrant m_w that given by the original signer. We use the following game between a $\mathcal{PPT}$ adversary $\mathcal{F}_{II}$ and a challenger $\mathcal{C}$ to describe this type of malicious attacks.

- Setup: The challenger $\mathcal{C}$ runs CPG algorithm to obtain the common parameters cp, CKG algorithm to generate not only Alice's secret/public key pair (x_A, y_A) but Bob's secret/public key pair (x_B, y_B). Afterwards $\mathcal{C}$ sends $(cp, y_A, x_B, y_B, ID_A, ID_B, m_w)$ to the adversary $\mathcal{F}_{II}$.
- Hash queries: $\mathcal{F}_{II}$ sets the i^{th} user adaptively in this system as the proxy signer and sends the public key y_i and identity ID_i (which is related to the i^{th} user) to $\mathcal{C}$. $\mathcal{F}_{II}$ is able to get the hash value on any proxy signer with the identity ID_i he chooses. $\mathcal{C}$ calls the hash oracle and returns the hash value as response.
- WS queries: $\mathcal{F}_{II}$ asks the witness signature with the proxy signer's identity ID_i. In response, $\mathcal{C}$ returns a witness signature.

Finally, $\mathcal{F}_{II}$ submits a forgery witness signature s_A' with the target proxy signer's identity ID_B. $\mathcal{F}_{II}$ wins if:

(1) s_A' is a valid witness signature on ID_B.

(2) ID_B has not been requested during the WS queries.

The advantage of an adaptive chosen-message adversary $\mathcal{F}_{II}$ is defined as $Adv^{uf-cma}_{\mathcal{F}_{II},\text{PTS}}$.

Type III adversary: Note this type of adversary $\mathcal{F}_{III}$ as a malicious original signer.

He aims to forge a proxy secret key of a valid warrant m_w that given by the original signer. We utilize the following game between a $\mathcal{PPT}$ adversary $\mathcal{F}_{III}$ and a challenger $\mathcal{C}$ to describe this type.

- Setup 1: The challenger $\mathcal{C}$ runs CPG algorithm to obtain the common parameters cp, CKG algorithm to generate not only Alice's secret/public key pair (x_A, y_A) but Bob's secret/public key pair (x_B, y_B). Afterwards $\mathcal{C}$ sends $(cp, x_A, y_A, y_B, ID_A, ID_B, m_w)$ to the adversary $\mathcal{F}_{III}$.
- Setup 2: The challenger $\mathcal{C}$ uses WS algorithm to get a public label $R_A \in \mathbb{G}$ relevant to a secret label r_A and a witness signature s_A. $\mathcal{C}$ then sends s_A to $\mathcal{F}_{III}$.

At last, $\mathcal{F}_{III}$ submits a forgery $x_p^{'}$ as the proxy secret key. $\mathcal{F}$ wins if $x_p^{'}$ holds the equation $g^{x'_p} = g^{s_A} y_B^{y_A}$.

The advantage of an adaptive chosen-message adversary $\mathcal{F}_{III}$ is defined as $Adv^{uf-cma}_{\mathcal{F}_{III},\mathrm{PTS}}$.

Definition 3. A *PTS* scheme is unforgeable against chosen-message attacks if it is secure against both *Type II* and *Type III* adversaries.

In addition to the above attack models, we also consider a $\mathcal{PPT}$ adversary $\mathcal{F}$ to define the PTVf - Unforgeability.

- Setup: The challenger $\mathcal{C}$ runs CPG algorithm to obtain the common parameters cp, CKG algorithm to generate not only Alice's secret/public key pair (x_A, y_A) but Bob's secret/public key pair (x_B, y_B). $\mathcal{C}$ also uses WS algorithm to get a public label $R_A \in \mathbb{G}$ relevant to a secret label r_A and a witness signature s_A, PKG algorithm to generate proxy secret/public key pair (x_p, y_p). Afterwards $\mathcal{C}$ sends $(cp, y_A, y_B, R_A, ID_A, ID_B, m_w, y_p)$ to the adversary $\mathcal{F}$.
- PTSign queries: $\mathcal{F}$ chooses nodes $i, j \in \mathbb{N}$ and asks the proxy transitive signature σ_{ij} of edge (i, j). $\mathcal{C}$ runs PTSign algorithm and outputs a signature σ_{ij} as response.

At last, $\mathcal{F}$ submits a forgery $\sigma_{i'j'}$ for edge (i', j'). Let $G' = (V', E')$ be the graph formed by all $\mathcal{F}$'s queries. Among the rest, V' is the set of all i' and j' while E' be the set of (i', j'). $\mathcal{F}$ wins if:

(1) $\mathrm{PTVf}(cp, ID_A, ID_B, m_w, y_A, R_A, y_B, i', j', \sigma_{i'j'}) = 1$.

(2) $(i', j') \notin \tilde{G}'$. Here $\tilde{G}'$ is the transitive closure of G'.

(3) (i', j') has never been queried during the game.

The advantage of an adaptive chosen-message adversary $\mathcal{F}$ is defined as $Adv^{uf-cma}_{\mathcal{F},\mathrm{PTS}}$.

Definition 4. We say that a *PTS* scheme is unforgeable under chosen-message attacks if $Adv^{uf-cma}_{\mathcal{F},\mathrm{PTS}}$ is negligible for any $\mathcal{PPT}$ adversary $\mathcal{F}$.

4.2.2 Privacy

For a PTS scheme, the basic privacy requirement is the privacy of transitive signatures. That is, for same edge, even an adaptive chosen-message attacker cannot distinguish a valid signature from the original signer or the composition algorithm. This means that the composition algorithm can work properly even if the given signatures were obtained by composition algorithm.

Definition 5. We say that a *PTS* scheme meets the requirement of indistinguishability if the composition algorithm is invoked on legitimate signatures.

5 Our Construction

In this section, we combine Sunitha et al.'s [27] work with Bellare and Neven's [4] GapTS-2 scheme to obtain a Proxy Transitive Signature (PTS) scheme. Later, we prove that our scheme is unforgeable under adaptive chosen-message attacks in a model where the hash function is a random oracle.

Furthermore, in accordance with the syntax of PTS , we let Alice denote the original signer and Bob is the proxy signer.

5.1 Concrete Scheme

- CPG: Let $\mathbb{G}$ is a q-order multiplicative subgroup of $\mathbb{Z}_q^*$ generated by g. Note a bilinear mapping $\hat{e}: \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ named $\hat{e}$. Then the common parameters are $cp = (\hat{e}, g)$.
- CKG: Taking the common parameter cp as input, the algorithm outputs Alice's secret/public key pair (x_A, y_A) and Bob's secret/public key pair (x_B, y_B), where $y_A = g^{x_A}$ and $y_B = g^{x_B}$.
- WS: Alice chooses a secret label $r_A \in \mathbb{Z}_q^*$, computes $R_A = g^{r_A}$, $h = H_w(ID_A \parallel ID_B \parallel m_w \parallel R_A)$ and $s_A = r_A y_B + x_A h$. Here ID_A and ID_B are the identities of Alice and Bob, m_w is the warrant. Alice then sends triple (s_A, R_A, m_w) to Bob.
- WV: Bob verifies whether $h = H_w(ID_A \parallel ID_B \parallel m_w \parallel R_A)$ and $g^{s_A} = R_A^{y_B} y_A^h$. As a result, the algorithm returns 1 for both the equations hold and 0 for not.
- PKG: Bob computes $x_p = s_A + x_B y_A$ and $y_p = g^{x_p}$ as the proxy key pair.
- PTSign: Given nodes $i, j \in \mathbb{N}$, Bob computes $\sigma_{ij} = (h_i h_j^{-1})^{x_p} \in \mathbb{G}$, where $h_i = H(i)$ and $h_j = H(j)$. (w.l.o.g., $i < j$, one can swap them if not the case).
- PTVf: Anyone can verify whether $\hat{e}(g, \sigma_{ij}) =$

$\hat{e}(R_A^{y_B} y_A^h y_B^{y_A}, h_i h_j^{-1})$, the receiver accepts it as a valid proxy transitive signature iff the equation holds and rejects it for not.

- PComp: Given nodes $i, j, k \in \mathbb{N}$ and corresponding proxy transitive signatures σ_{ij}, σ_{jk}, if PTVf $(cp, ID_A, ID_B, m_w, y_A, R_A, y_B, \sigma_{ij}, i, j) = 0$ or PTVf $(cp, ID_A, ID_B, m_w, y_A, R_A, y_B, \sigma_{jk}, j, k) = 0$ then returns $\perp$ as an indication of failure. Otherwise, returns $\sigma_{ik} = \sigma_{ij} \cdot \sigma_{jk}$ as the composed signature for the edge (i,k). (w.l.o.g., $i < j < k$, one can swap them if not the case).

5.2 Correctness

Consistency of PTS : We show three consistency requirements in PTS .

- WV Consistency of WS: If $h = H_w(ID_A \parallel ID_B \parallel m_w \parallel R_A)$ and $s_A = \mathrm{WS}(cp, ID_A, ID_B, m_w, r_A, x_A, y_B) = r_A y_B + x_A h$, then $g^{s_A} = g^{r_A y_B + x_A h} = g^{r_A y_B} g^{x_A h} = R_A^{y_B} y_A^h$.

  Therefore $Pr[\mathrm{WV}(s_A, ID_A, ID_B, m_w, R_A, y_A, y_B)=1]=1$.
- PTVf Consistency of PTSign: If $\sigma_{ij} = \mathrm{PTSign}(x_p, i, j) = (h_i h_j^{-1})^{x_p}$, where $h_i = H(i)$ and $h_j = H(j)$, then $\hat{e}(g, \sigma_{ij}) = \hat{e}(g, (h_i h_j^{-1})^{x_p}) = \hat{e}(g, h_i h_j^{-1})^{x_p} = \hat{e}(g^{x_p}, h_i h_j^{-1}) = \hat{e}(y_s, h_i h_j^{-1})$.

  Therefore $Pr[\mathrm{PTVf}(cp, ID_A, ID_B, m_w, y_A, R_A, y_B, \sigma_{ij}) = 1] = 1$.
- PTVf Consistency of PComp: $\sigma_{ik} = \sigma_{ij} \cdot \sigma_{jk} = (h_i h_j^{-1})^{x_p} \cdot (h_j h_k^{-1})^{x_p} = (h_i h_k^{-1})^{x_p}$, where $h_i = H(i)$, $h_j = H(j)$ and $h_k = H(k)$, then $\hat{e}(g, \sigma_{ik}) = \hat{e}(g, (h_i h_k^{-1})^{x_p}) = \hat{e}(g, h_i h_k^{-1})^{x_p} = \hat{e}(g^{x_p}, h_i h_k^{-1}) = \hat{e}(R_A^{y_B} y_A^h y_B^{y_A}, h_i h_k^{-1})$.

  Therefore $\Pr[\mathrm{PTVf}(cp, ID_A, ID_B, m_w, y_A, R_A, y_B, \sigma_{ik}, i, k)) = 1] = 1$.

5.3 Security Results

We present the security results of our PTS scheme with the following three theorems and provide formal security proofs in this section.

Theorem 1. (Unforgeability of *WV*). If *DL* problem is hard, then our proposed *PTS* scheme is unforgeable against an adaptive chosen-message $\mathcal{PPT}$ adversary $\mathcal{F}_{II}$ or $\mathcal{F}_{III}$ in the random oracle model.

Proof. Firstly, we consider the $\mathcal{F}_{II}$ adversary described above. Given a $\mathcal{PPT}$ adversary $\mathcal{F}_{II}$ for PTS with $Adv^{uf-cma}_{\mathcal{F}_{II},\mathrm{PTS}}$, We show a $\mathcal{PPT}$ *DL* adversary $\mathcal{F}_{\mathcal{D}}$, such that for every $k \in \mathbb{N}$, $Adv^{uf-cma}_{\mathcal{F}_{II},\mathrm{PTS}}(k) \le Adv^{DL}_{\mathcal{F}_{\mathcal{D}}}(k)$.

The assumption, namely the *DL* problem is hard and $\mathcal{F}_{\mathcal{D}}$ solves it in polynomial-time is equally hard, implies that the advantage function $Adv^{DL}_{\mathcal{F}_{\mathcal{D}}}(k)$ is negligible. Then we get that $Adv^{uf-cma}_{\mathcal{F}_{II},\mathrm{PTS}}$, which completes the proof. We now describe $\mathcal{F}_{II}$ in detail.

Given $(g, y_0 = g^{x_0} \in \mathbb{G})$ for some unknown $x_0 \in \mathcal{Z}_q^*$ as an instance of *DL* problem, $\mathcal{F}_{\mathcal{D}}$ wins if it computes the value x_0. $\mathcal{F}_{\mathcal{D}}$ sets Alice's public key $y_A = y_0 = g^{x_0}$, $\mathcal{F}_{II}$ can ask a hash oracle $\mathcal{O}_{\mathcal{HW}}(\cdot)$ and the WS algorithm adaptively and $\mathcal{F}_{\mathcal{D}}$ will act both of them in our proof.

Besides, $\mathcal{F}_{\mathcal{D}}$ maintains a list H_w to state all the hash queries. Now, we will show how $\mathcal{F}_{\mathcal{D}}$ answers $\mathcal{F}_{II}$'s queries in the following.

- Setup: $\mathcal{F}_{\mathcal{D}}$ will be given the common parameters *cp*, Alice's public key y_A, Bob's secret/public key pair (x_B, y_B), Alice's and Bob's identities ID_A and ID_B and the warrant m_w. Afterwards $\mathcal{F}_{\mathcal{D}}$ sends $(cp, y_A, x_B, y_B, ID_A, ID_B, m_w)$ to the adversary $\mathcal{F}_{II}$.
- Hash queries: $\mathcal{F}_{II}$ sets the i^{th} user adaptively in this system as the proxy signer and sends the public key y_i and identity ID_i related to the i^{th} user to $\mathcal{F}_{\mathcal{D}}$. $\mathcal{F}_{II}$ can request the hash value on any proxy signer with the identity ID_i he chooses. For each query (ID_i, R_i), $\mathcal{F}_{\mathcal{D}}$ checks the H_w list, if the list exits an item (ID_i, R_i, h_i), $\mathcal{F}_{\mathcal{D}}$ sets $h = H_w(ID_A \parallel ID_B \parallel m_w \parallel R_A)$ and sends h_i to $\mathcal{F}_{II}$, otherwise, $\mathcal{F}_{\mathcal{D}}$ chooses $h_i \in \mathcal{Z}_q^*$ randomly such that there is no item $(\cdot, \cdot, h_i)$ in the list, $\mathcal{F}_{\mathcal{D}}$ then adds (ID_i, R_i, h_i) to the list and returns h_i as response.
- WS queries: $\mathcal{F}_{II}$ asks the witness signature with the proxy signer's identity ID_i. For each query, $\mathcal{F}_{\mathcal{D}}$ firstly chooses $S_i, v_i \in \mathcal{Z}_q^*$ randomly and computes $R_i = [g^{S_i} (y_0^{v_i})^{-1}]^{y_i^{-1}}$. $\mathcal{F}_{\mathcal{D}}$ then sets $H_w(ID_A \parallel ID_i \parallel m_w \parallel R_i) = v_i$, stores (ID_i, R_i, v_i) to the H_w list and returns (S_i, R_i, m_w) as response.

We can see that the simulation is perfect in the random oracle model [39-40]. After all the simulations, $\mathcal{F}_{II}$ forges (s_A', R_A), which is a valid WS signature on the identity ID_B with the restriction that ID_B has never been queried before. Then $\mathcal{F}_{\mathcal{D}}$ can produce two valid WS signature $s_A' = r_A y_B + x_A h'$, $h' = H'_w(ID_A \parallel ID_B \parallel m_w \parallel R_A)$, and $s_A = r_A y_B + x_A h$, $h = H_w(ID_A \parallel ID_B \parallel m_w \parallel R_A)$, on the same identity ID_B such that $h' \neq h$. $\mathcal{F}_{\mathcal{D}}$ outputs $x_0 = x_A = (s_A - s_A')(h - h')^{-1}$.

$\mathcal{F}_{\mathcal{D}}$ definitely solve the *DL* problem and he will not

abort during the Hash , WS queries. Accordingly we have $Adv^{uf-cma}_{\mathcal{F}_{II},\mathrm{PTS}}(k) \le Adv^{DL}_{\mathcal{F}_{\mathcal{D}}}(k)$.

We then consider the $\mathcal{F}_{III}$ adversary.

Let $\mathcal{F}_{III}$ be a Type *III* adversary who can forge a valid signature of proposed scheme PTS with $Adv^{uf-cma}_{\mathcal{F}_{III},\mathrm{PTS}}(k)$, then there exits an adversary $\mathcal{F}_{\mathcal{D}}$ who can utilize $\mathcal{F}_{III}$ to solve *DL* problem, such that for every $k \in \mathbb{N}$, $Adv^{uf-cma}_{\mathcal{F}_{III},\mathrm{PTS}}(k) \le Adv^{DL}_{\mathcal{F}_{\mathcal{D}}}(k)$.

This assumption means that the advantage function $Adv^{DL}_{\mathcal{F}_{\mathcal{D}}}(k)$ is negligible when the *DL* problem is hard. The equation then says that $Adv^{uf-cma}_{\mathcal{F}_{III},\mathrm{PTS}}(k)$ is also negligible, which completes the proof. We now describe $\mathcal{F}_{III}$ in detail.

Again, given $(g, y_0 = g^{x_0} \in \mathbb{G})$ for some unknown $x_0 \in \mathcal{Z}_q^*$ as an instance of *DL* problem, $\mathcal{F}_{\mathcal{D}}$ wins if it computes the value x_0. $\mathcal{F}_{\mathcal{D}}$ sets Bob's public key $y_0 = y_B = g^{x_B}$. We then describe how it operates.

- Setup 1: $\mathcal{F}_{\mathcal{D}}$ will be given the common parameters cp, Alice's secret/public key pair (x_A, y_A), Bob's public key y_B, Alice's and Bob's identity ID_A and ID_B, the warrant m_w. Afterwards $\mathcal{F}_{\mathcal{D}}$ sends $(cp, x_A, y_A, y_B, ID_A, ID_B, m_w)$ to the adversary $\mathcal{F}_{III}$.
- Setup 2: $\mathcal{F}_{III}$ chooses a secret label $r_A \in \mathbb{Z}_q^*$ adaptively, computes $R_A = g^{r_A}$, $h = H_w(ID_A \| ID_B \| m_w \| R_A)$ and $s_A = r_A y_B + x_A h$. $\mathcal{F}_{III}$ then sends triple (s_A, R_A, m_w) to $\mathcal{F}_{\mathcal{D}}$.

Finally, $\mathcal{F}_{III}$ terminates the game and outputs a valid forgery $x_p^{'}$ which holds the equation $g^{x_p^{'}} = g^{S_A} y_B^{y_A}$. $\mathcal{F}_{\mathcal{D}}$ uses $x_p^{'}$ to compute $(x_p^{'} - S_A) y_A^{-1}$ as x_0 and outputs x_0 as a solution to its challenge.

Now, $\mathcal{F}_{\mathcal{D}}$ definitely solve the *DL* problem and we have $Adv^{uf-cma}_{\mathcal{F}_{III},\mathrm{PTS}}(k) \le Adv^{DL}_{\mathcal{F}_{\mathcal{D}}}(k)$.

Theorem 2. (Unforgeability of *PTVf*). If one-more *BDH* problem is hard, then our proposed *PTS* scheme is unforgeable against an adaptive chosen-message $\mathcal{PPT}$ adversary $\mathcal{F}$ in the random oracle model.

Proof. Given a $\mathcal{PPT}$ adversary $\mathcal{F}$ for PTS with $Adv^{uf-cma}_{\mathcal{F},\mathrm{PTS}}(k)$, We construct a $\mathcal{PPT}$ one-more *BDH* adversary $\mathcal{F}_{\mathcal{B}}$, such that for every $k \in \mathbb{N}$, $Adv^{uf-cma}_{\mathcal{F},\mathrm{PTS}}(k) \le Adv^{om-BDH}_{\mathcal{F}_{\mathcal{B}}}(k)$.

The assumption, namely the one-more *BDH* problem is hard, implies that the advantage function $Adv^{om-BDH}_{\mathcal{F}_{\mathcal{B}}}(k)$ is negligible. The equation then says that $Adv^{uf-cma}_{\mathcal{F},\mathrm{PTS}}(k)$ is also negligible, which completes the proof.

Given $(\hat{e}, \mathbb{G}, \mathbb{G}_T, g, q, y_A, y_B)$, the one-more *BDH* adv- ersary $\mathcal{F}_{\mathcal{B}}$ also has access to a challenge oracle $\mathcal{O}_{\mathcal{CH}}(\cdot)$ and a *CDH* oracle $\mathcal{O}_{\mathcal{CDH}}(\cdot)$. $\mathcal{F}_{\mathcal{B}}$ wins if it computes all *n* points output by $\mathcal{O}_{\mathcal{CH}}(\cdot)$, using strictly less than *n* queries to $\mathcal{O}_{\mathcal{CDH}}(\cdot)$. Besides, $\mathcal{F}_{\mathcal{B}}$ maintains state information (V, Δ), where V is the set of all queried nodes, and $\Delta : V \times V \to \mathbb{G}$ is a function storing known edge signatures. We then describe how it operates.

- Setup: As per the definitions above, $\mathcal{F}_{\mathcal{B}}$ gets the common parameters cp, Alice's public key y_A, Bob's public key y_B and a warrant m_w. $\mathcal{F}_{\mathcal{B}}$ then chooses a secret label $r_A \in \mathbb{Z}_q^*$, computes $R_A = g^{r_A}$ and the proxy public key $y_p = R_A^{y_B} y_A^h y_B^{y_A}$ where $h = H_w(ID_A \| ID_B \| m_w \| R_A)$. Finally, $\mathcal{F}_{\mathcal{B}}$ sends $(cp, y_A, y_B, R_A, ID_A, ID_B, m_w, y_p)$ to $\mathcal{F}$.
- Hash queries: $\mathcal{F}_{\mathcal{B}}$ maintains a table $H(\cdot)$. For each query is $H(i)$ made by $\mathcal{F}$, $\mathcal{F}_{\mathcal{B}}$ proceeds as follows:

  (1) If $i \notin V$ then $V \leftarrow V \cup \{i\}$; $H(i) \xleftarrow{\$} \mathcal{O}_{\mathcal{CH}}(\cdot)$; $\Delta(i,i) \leftarrow 1$

  (2) Return $H(i)$ to $\mathcal{F}$.
- PTSign queries: When answering $\mathcal{F}$'s signature queries, $\mathcal{F}_{\mathcal{B}}$ uses $\mathcal{O}_{\mathcal{CDH}}(\cdot)$ to compute the edge signature, calling the oracle only when the requested signature cannot be computed by composing previously signed edges. For each signature on edge (i, j), $\mathcal{F}_{\mathcal{B}}$ proceeds as follows:

  (3) If $i > j$ then swap (i, j)

  (4) If $i \notin V$ then; $V \leftarrow V \cup \{i\}$; $H(i) \xleftarrow{\$} \mathcal{O}_{\mathcal{CH}}(\cdot)$; $\Delta(i,i) \leftarrow 1$

  (5) If $j \notin V$ then $V \leftarrow V \cup \{j\}$; $H(j) \xleftarrow{\$} \mathcal{O}_{\mathcal{CH}}(\cdot)$; $\Delta(j,j) \leftarrow 1$

  (6) If $\Delta(i, j)$ is not defined then

  (7) $\Delta(i, j) \leftarrow \mathcal{O}_{\mathcal{CDH}}(H(i)H(j)^{-1})$

  (8) $\Delta(j, i) \leftarrow \Delta(i, j)^{-1}$

  (9) For all $v \in V \setminus \{i, j\}$ do

  (10) If $\Delta(v, i)$ is defined then

  (11) $\Delta(v, j) \leftarrow \Delta(v, i) \cdot \Delta(i, j)$

  (12) $\Delta(j, v) \leftarrow \Delta(v, j)^{-1}$

  (13) If $\Delta(v, j)$ is defined then

  (14) $\Delta(v, i) \leftarrow \Delta(v, j) \cdot \Delta(j, i)$

  (15) $\Delta(i, v) \leftarrow \Delta(v, i)^{-1}$

  (16) $\sigma_{ij} \leftarrow \Delta(i, j)$

  (17) σ_{ij} to $\mathcal{F}$.

Finally, $\mathcal{F}$ forges a signature $\sigma_{i'j'}$ on edge (i', j'). Assume that $\mathcal{F}$ queries the hash oracle on i' and j' (and hence that $i', j' \in V$.), otherwise $\mathcal{F}_{\mathcal{B}}$ can query the hash oracle itself after $\mathcal{F}$ outputs his forgery. (In this analysis, we assume without loss of generality that $i' < j'$. If this is not the case, one can swap i' and j'.)

Let $G = (V, E)$ be the graph formed by $\mathcal{F}$'s PTSign queries, and let $\tilde{G} = (V, \tilde{E})$ be its transitive closure. If $\sigma_{i'j'}$ is a valid forgery, meaning that:

(1) $\mathrm{PTVf}(cp, ID_A, ID_B, m_w, y_A, R_A, y_B, i', j', \sigma_{i'j'}) = 1$.

(2) $(i', j') \notin \tilde{G}$.

(3) (i', j') has not been queried during the game.

$\mathcal{F}_{\mathcal{B}}$ then solves BDH solutions for all challenges that it received from the challenge oracle as follows. $\mathcal{F}_{\mathcal{B}}$ firstly divides $\tilde{G}$ into r disjoint subgraphs $V_t \subset V$ for $t = 1, 2, \ldots, r$. Let $V_{t'}$ denotes the subgraph containing node i'.

For every $t = 1, 2, \ldots, r$, $t \neq t'$, $\mathcal{F}_{\mathcal{B}}$ computes the BDH solutions of all nodes in $V_{t'}$ as follows:

(18) $\sigma_{i'} \leftarrow \sigma_{i'j'} \cdot \sigma_{j'}$

(19) For all $v \in V_{t'} \setminus \{i'\}$ do

(20) $\sigma_v \leftarrow \mathcal{O}_{\mathcal{CDH}}(H(v))$,

while the BDH solutions of all nodes in V_t are computed as

(21) For all $v \in V_t$ do

(22) $\sigma_v \leftarrow \mathcal{O}_{\mathcal{CDH}}(H(v))$.

For each $i \in V$, the hash $H(i)$ was obtained as a result of a query to $\mathcal{O}_{\mathcal{CH}}(\cdot)$, so $\mathcal{F}_{\mathcal{B}}$ can spontaneously output BDH solutions for all $i \in V$.

Now we need to check that $\mathcal{F}_{\mathcal{B}}$ actually won the game. Thus, we have to count the number of CDH queries. The component $V_{t'}$ needed $|V_{t'}| - 1$ queries for it did not need the query to get $\mathcal{O}_{\mathcal{CDH}}(H(v))$. Simultaneously, for each component V_t, $t \neq t'$, $\mathcal{F}_{\mathcal{B}}$ needed V_t CDH queries. In conclusion, $\mathcal{F}$ computed $|V|$ BDH solutions using $(|V_{t'}| - 1) + \sum_{t \neq t'} |V_t| = |V| - 1$ CDH queries, and hence wins the game.

It remains to analyze the probability of $\mathcal{F}_{\mathcal{B}}$ not aborting. $\mathcal{F}_{\mathcal{B}}$ will not abort during the Hash , PTSign queries. Accordingly we have $Adv^{uf-cma}_{\mathcal{F},\mathrm{PTS}}(k) \leq Adv^{om-BDH}_{\mathcal{F}_{\mathcal{B}}}(k)$. This completes the proof of the theorem.

Theorem 3. (Indistinguishability of *PTS*). The proposed *PTS* scheme is indistinguishable with the restriction that the composition algorithm is invoked on legitimate signatures.

Proof. Suppose the composition algorithm is invoked on legitimate signatures, we then show that our proposed PTS is perfectly indistinguishable in privacy protection.

Suppose i', j', k' are distinct nodes and $h_{i'} = H(i')$, $h_{j'} = H(j'), h_{k'} = H(k')$. Also suppose $\sigma_{i'j'} = (h_{i'} h_{j'}^{-1})^{x_p}$, $\sigma_{j'k'} = (h_{j'} h_{k'}^{-1})^{x_p}$ are legitimate signature of edge (i', j') and (j', k') (w.l.o.g., $i' < j' < k'$.), the defined PTSign algor- ithm returns $\sigma_{i'k'}$.

For the equation $\sigma_{i'k'} = \sigma_{i'j'} \cdot \sigma_{j'k'} = (h_{i'} h_{j'}^{-1})^{x_p} \cdot (h_{j'} h_{k'}^{-1})^{x_p} = (h_{i'} h_{k'}^{-1})^{x_p}$, we would say that the signature $\sigma_{i'k'}$ generated by the algorithm PTSign is the same as the signature produced by the algorithm PComp . Thus, our scheme meets the privacy requirement.

5.4 Performance Analysis

In order to facilitate the analysis of our PTS scheme, we first define the following notations to denote the computational costs in algorithms. That is: "C_e" for computing an exponentiation operation; "C_h" for computing one hash operation; "C_{mt}" for mapping one point operation; "C_i" for computing one inversion operation; "C_{mu}" for computing one modular multiplication; "C_p" for computing one pairing operation; "C_a" for computing one modular addition.

In the algorithm WS, we need one exponentiation operation to compute $R_A = g^{r_A}$, one hash operation to compute $h = H_w(ID_A \parallel ID_B \parallel m_w \parallel R_A)$. Meanwhile, we need two modular multiplication and one modular addition to get $s_A = r_A y_B + x_A h$. Hence, we list the result of the proposed scheme in Table 1.

Table 1. Computational costs

CKG	WS	WV
$2C_e$	$1C_e+1C_h+2C_{mu}+1C_a$	$1C_h+3C_e+1C_{mu}$
PKG	PTSign	PTvf
$1C_{mu}+1C_a+1C_e$	$2C_{mt}+1C_i+1C_e+1C_{mu}$	$2C_p+2C_{mt}+1C_i+3C_e+3C_{mu}$

Furthermore, for our proposed PTS scheme, we also evaluate its performance by analyzing the time cost of its sub algorithms. We use pairing-based library (version 0.5.12, http://crypto.stanford.edu/pbc/) to conduct our simulation experiments. Specifi- cally, in our experiments, the composition algorithm (PComp) is omitted for the reason that it is just one multiply computation. Table 2 provides the details of the employed platform and Figure 1 shows the time costs in our simulation.

Table 2. Simulation Platform

OS	Ubuntu 10.10
CPU	Intel(R) Core(TM) i7
Memory	16G RAM
Hard disk	256G
Programming language	C

In Figure 1, CKG, WS, WV, PKG, PTSign, PTVf refer to the algorithms in our proposed scheme PTS. Numbers are used to count the time costs when these algorithms were ran (We tested 20 sets of data.). Take "PTvf" as an example, when running the Proxy Signature Verification algorithm, the max time cost is 0.043394 *s*, the min time cost is 0.031336 *s*, and the average time cost is 0.040097 *s*. It can be found in the figure that this algorithm is the most time-consuming algorithm, but its time cost is still extremely small. Generally speaking, abundant experimental tests and analyses on real datasets show that our proposed scheme PTS can achieve good performance.

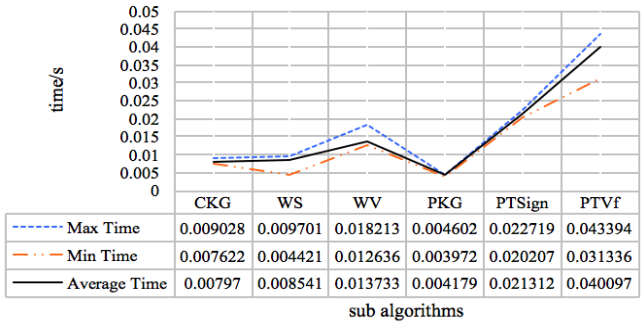


Figure 1. Time costs

6 Conclusions

In this paper, we introduced a Proxy Transitive Signature (PTS) scheme, which can be successfully applied to address the privacy issue in the graph-based big data system. The proposed scheme is provably secure in the random oracle based on the *DL* and one-more *BDH* assumptions. Nevertheless, as shown in the Performance Analysis, the Proxy Signature Verification algorithm is the most time-consuming algorithm mainly for the reason that it needs two pairing operations. As a future research work, we will try to design new schemes in order to provide more efficient solutions. Furthermore, we aim to design a secure PTS scheme which can be proved secure in the standard model.

Acknowledgements

This work is supported by National Natural Science Foundation of China (Grant No. 61402110), Distinguished Young Scholars Fund of Fujian (Grant No. 2016J06013) and Guangxi Colleges and Universities Key Lab-oratory of Cloud Computing and Complex Systems (Grant No. YF16103).

References

[1] The Identity Theft Resource Center, Data Breach Reports. http://www.idtheftcenter.org/2016databreaches.html/

[2] S. Micali, R. L. Rivest, Transitive Signature Schemes, *Proceedings of the Cryptographer's Track at the RSA Conference 2002*, San Jose, CA, 2002, pp. 236-243.

[3] S. Goldwasser, S. Micali, R. L. Rivest, A Digital Signature Scheme Secure against Adaptive Chosen-message Attacks, *SIAM Journal on Computing*, Vol. 17, No. 2, pp. 281-308, April, 1988.

[4] M. Bellare, G. Neven, Transitive Signatures: New Schemes and Proofs, *IEEE Transactions on Information Theory*, Vol. 51, No. 6, pp. 2133-2151, June, 2005.

[5] M. Mambo, K. Usuda, E. Okamoto, Proxy Signatures: Delegation of the Power to Sign Messages, *IEICE Transactions on Fundamentals of Electronics, Communications & Computer Sciences*, Vol. E79-A, No. 9, pp. 1338-1354, September, 1996.

[6] J. Li, Z. Cao, J. Li, Y. Zhang, Present Situation and Progress of Proxy Signature, *Journal of China Institute of Communications*, Vol. 24, No. 10, pp. 114-124, October, 2003.

[7] M. Mambo, K. Usuda, E. Okamoto, Proxy Signatures for Delegating Signing Operation, *Proceedings of the 3rd ACM Conference on Computer and Communications Security*, New Delhi, India, 1996, pp. 48-57.

[8] H. Sun, An Efficient Nonrepudiable Threshold Proxy Signature Scheme with Known Signers, *Computer Communications*, Vol. 22, No. 8, pp. 717- 722, May, 1999.

[9] L. Yi, G. Bai, G. Xiao, Proxy Multi-signature: A New Type of Proxy Signature Scheme, *Acta Electronica Sinica*, Vol. 29, No. 4, pp. 569-570, April, 2001.

[10] Z. Tan, Z. Liu, C. Tang, Digital Proxy Blind Signature Schemes based on DLP and ECDLP, *Mathematics-Mechanization Research Preprints*, No. 21, pp. 212-217, December, 2002.

[11] Z. Tan, Z. Liu, C. Tang, A Proxy Blind Signature Scheme based on DLP, *Journal of Software*, Vol. 14, No. 11, pp. 1931-1935, November, 2003.

[12] Y. Liu, J. Liu, One Controllable Proxy Signature Scheme, *Journal of Sichuan University (Natural Science Edition)*, Vol. 41, No. 5, pp. 982-985, October, 2004.

[13] J. He, X. Li, L. Li, C. Tang, A New Forward-secure Proxy Signature Scheme, *Proceedings of the 2010 International Forum on Information Technology and Applications*, Vol. 02, Kunming, China, 2010, pp. 30-33.

[14] S. Cui, F. Wen, Improvement of a Forward-secure Proxy Signature Scheme, *Proceedings of the 2nd International Conference on Computer Engineering and Technology- Vol. 1*, Chengdu, China, 2010, pp. 441-444.

[15] J.-Z. Dai, X.-H. Yang, J.-X. Dong, Designated-receiver Proxy Signature Scheme for Electronic Commerce, *Proceedings of the IEEE International Conference on Systems, Man & Cybernetics*, Washington, DC, 2003, pp. 384-389.

[16] S. Tzeng, C. Yang, M. Hwang, A Nonrepudiable Threshold Multi-proxy Multi-signature Scheme with Shared Verification, *Future Generation Computer Systems*, Vol. 20, No. 5, pp. 887-893, June, 2004.

[17] Y. Yu, C. Xu, X. Huang, Y. Mu, An Efficient Anonymous Proxy Signature Scheme with Provable Security, *Computer Standards & Interfaces*, Vol. 31, No. 2, pp. 348-353, February, 2009.

[18] V. Varadharajan, P. Allen, S. Black, An Analysis of the Proxy Problem in Distributed Systems, *Proceedings of the IEEE Symposium on Research in Security and Privacy*, Oakland, CA, 1991, pp. 255-275.

[19] B. C. Neuman, Proxy-based Authorization and Accounting for Distributed Systems, *Proceedings of the 13th International Conference on Distributed Computing Systems*, Pittsburgh, PA, 1993, pp. 283-291.

[20] S. Kim, S. Park, D. Won, Proxy Signatures, Revisited, *Proceedings of the First International Conference on Information and Communications Security*, Beijing, China, 1997, pp. 223-232.

[21] J.-Y. Lee, J. H. Cheon, S. Kim, An Analysis of Proxy Signatures: Is A Secure Channel Necessary?, *Proceedings of the Cryptographers' Track at the RSA Conference 2003*, San Francisco, CA, 2003, pp. 68-79.

[22] B. Lee, H. Kim, K. Kim, Strong Proxy Signature and its Applications, *Proceedings of the Symposium on Cryptography and Information Security*, Oiso, Japan, 2001, pp. 603-608.

[23] B. Lee, H. Kim, K. Kim, Secure Mobile Agent using Strong Non-designated Proxy Signature, *Proceedings of the 6th Australasian Conference on Information Security and Privacy*, Sydney, Australia, 2001, pp. 474-486.

[24] T. Okamoto, A. Inomata, E. Okamoto, A Proposal of Short Proxy Signature Using Pairing, *Proceedings of the International Symposium on Information Technology: Coding and Computing*, Las Vegas, NV, 2005, pp. 631-635.

[25] X. Huang, Y. Mu, W. Susilo, F. Zhang, X. Chen, A Short Proxy Signature Scheme: Efficient Authentication in the Ubiquitous World, *Proceedings of the Embedded and Ubiquitous Computing - EUC 2005 Workshops*, Nagasaki, Japan, 2005, pp. 480-489.

[26] A. Boldyreva, A. Palacio, B. Warinschi, Secure Proxy Signature Schemes for Delegation of Signing Rights, *Journal of Cryptology*, Vol. 25, No. 1, pp. 57-115, January, 2012.

[27] N. R. Sunitha, B. B. Amberker, P. Koulgi, Controlled Delegation in E-cheques using Proxy Signatures, *Proceedings of the 11th IEEE International Enterprise Distributed Object Computing Conference*, Annapolis, MD, 2007, pp. 414-419.

[28] M. Bellare, G. Neven, Transitive Signatures based on Factoring and RSA, *Proceedings of the 8th International Conference on the Theory and Application of Cryptology and Information Security*, Queenstown, New Zealand, 2002, pp. 397-414.

[29] L. Wang, Z. Cao, S. Zheng, X. Huang, Y. Yang, Transitive Signatures from Braid Groups, *Proceedings of the 8th International Conference on Cryptology in India*, Chennai, India, 2007, pp. 183-196.

[30] Z. Gong, Z. Huang, W. Qiu, K. Chen, Transitive Signature Scheme from LFSR, *Journal of Information Science and Engineering*, Vol. 26, No. 1, pp. 131-143, January, 2010.

[31] S. Hohenberger, *The Cryptographic Impact of Groups with Infeasible Inversion*, Master's Thesis, Massachusettes Institute of Technology, Cambridge, MA, 2003.

[32] X. Yi, Directed Transitive Signature Scheme, *Proceedings of the Cryptographers' Track at the RSA Conference 2007*, San Francisco, CA, 2007, pp. 129-144.

[33] G. Neven, A Simple Transitive Signature Scheme for Directed Trees, *Theoretical Computer Science*, Vol. 396, No. 1, pp. 277-282, May, 2008.

[34] J. Xu, E.-C. Chang, J. Zhou, On Directed Transitive Signature, *IACR Cryptology ePrint Archive*, Vol. 2009, p. 209, 2009.

[35] P. Camacho, A. Hevia, Short Transitive Signatures for Directed Trees, *in: Proceedings of the Cryptographers' Track at the RSA Conference 2012*, San Francisco, CA, 2012, pp. 35-50.

[36] S. Hou, X. Huang, J. K. Liu, J. Li, L. Xu, Universal Designated Verifier Transitive Signatures for Graph-based Big Data, *Information Sciences*, Vol. 318, pp. 144-156, October, 2015.

[37] X. Chen, J. Li, X. Huang, J. Ma, W. Lou, New Publicly Verifiable Databases with Efficient Updates, *IEEE Transactions on Dependable and Secure Computing*, Vol. 12, No. 5, pp. 546-556, October, 2015.

[38] Y. Ren, J. Shen, J. Wang, J. Han, S. Lee, Mutual Verifiable Provable Data Auditing in Public Cloud Storage, *Journal of Internet Technology*, Vol. 16, No. 2, pp. 317-323, March, 2015.

[39] M. Bellare, P. Rogaway, Random Oracles are Practical: A Paradigm for Designing Efficient Protocols, *Proceedings of the 1st ACM Conference on Computer and Communications Security*, Fairfax, VA, 1993, pp. 62-73.

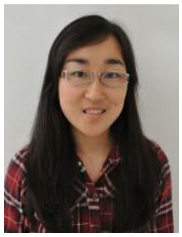
[40] W. Wu, Y. Mu, W. Susilo, X. Huang, L. Xu, An Efficient Construction of Convertible Undeniable Proxy Signatures, *Journal of Internet Technology*, Vol. 15, No. 3, pp. 471-484, May, 2014.

Biographies

Fei Zhu received his B.S. degree in 2015 and is currently working towards the M.S. degree at Fujian Normal University, Fujian, China. His current research interests include cryptography and information security.

Xiaoling Tao received the M.S. Degree in 2008 and is currently studying for her Ph.D degree in information and communication engineering at Guilin University of Electronic Technology, China. She is currently at this university as a professor. Her current research interests include network security, cloud computing and computational intelligence.

Chao Lin received his B.S. degree in 2013 and is currently working towards the M.S. degree in School of Mathematics and Computer Science, Fujian Normal University, China. His research interests include network and information security.



Wei Wu received her Ph.D. degree from the School of Computer Science and Software Engineering, University of Wollongong, Australia, in 2011. She is currently an Associate Professor at the School of Mathematics and Computer Science, Fujian Normal University, China. Her research interests include cryptography and information security.