Cryptanalysis and Improvement of Identity-based Proxy Multi-signature scheme

Jayaprakash Kar

Information Security Research Group
Department of Information Systems
Faculty of Computing & Information Technology
King Abdulaziz University,
Kingdom of Saudi Arabia, Jeddah-21589
jgopabandhu@kau.edu.sa

Abstract. Cao-Cao's recently proposed an identity-based proxy signature scheme and claim that the scheme is provably secure in random oracle model. In this paper we have reviewed the scheme and proven that the scheme is vulnerable to chosen message attack under the defined security model. To prevent this attack, we propose an improved version of the scheme. A Proxy multi-signature scheme allows an authorized proxy signer to sign on a message on behalf of a group of original signers.

Keywords:Chosen-message attack, Bilinear map, Digital signature, Public Key Infrastructures

1 Introduction

The concept of proxy signature scheme was first initiated by Mambo et al [2] in 1996. The Proxy signature scheme allows an original signer to delegate his signing capability to an authorized signer called the proxy signer to sign on a message on behalf of him. There are many versions of proxy signature like proxy multi-signature, which allows two or more original signers delegate their signing capability to a single proxy signer. This concept was introduced by Yi *et* al. in 2000 [3]. Another variant is multi-proxy signature where the original signer delegates his signing capability to two or more proxy signers. Also many new type of proxy signature, such as threshold proxy signature [12], proxy ring signature [16], proxy blind signature [14], one-time proxy signature [13], proxy blind multi-signature [15] have been constructed by joining proxy signature with the other special signature .

The concept of multi-proxy signature scheme was first introduced by Hwang and shi [8] in 2000. In 2004 Hwang *et* al. proposed multi proxy multi-signature scheme by combining both proxy multi-signature and multi proxy signature scheme [10]. It is a group-based proxy signature scheme. In this scheme, a group of original signers delegates the signing capability to a group of users named as a proxy signer group can sign messages on behalf of the group of original signer. However, in this type of scheme, both the original and proxy signers have a proper synchronization to create a certificate called proxy certificate.

The proxy signature scheme is very useful in many applications. For example, an organization has many departments such as production, HR, finance, accounts, etc. A document has to sign jointly by the department manager or the managers have to authorize to one trusted signer on behalf of them to sign. This is a very useful application of proxy multi-signature scheme to reduce the computational overhead of the company. Also proxy signature is applied in distributed shared systems [9], mobile agent environment [7], grid computing, global distribution networks, etc. Proxy multi-signature resolve the difficulty of signing multiple documents individually.

1.1 Security goals

A secure proxy signature scheme should have the following security goals [11]:

- `Distinguishability`: Anyone can distinguish proxy signature from a typical signature.

- `Verifiability`: There should be proper synchronization between the original signer and verifier. The verifier should accept the agreement of the original signer on the signed message.
  `Strong unforgeability`: No one except the proxy signer can generate a valid proxy signature on behalf of the original signer.
  `Identifiability`: Everyone can find out identity information such as proxy signer's identity, period of delegation etc from the proxy signature.
  `Non-deniability`: After the generation of proxy signature on behalf of the original signer, proxy signer should not deny that, he has not signed with the message.
  `Prevention of misuse`: The proxy secret key cannot be used by the proxy signer to generate a valid signature for other purpose. That is, he cannot sign the message, that have not been delegated by the original signer.

2 Organization of the paper

Section-3 describes some preliminaries on mathematical assumptions, Section-4 and 5 outlines the framework of proxy multi-signature scheme and security model respectively. In section-6 and 7, we present the review of Cao et al's scheme and its analysis. Section-8 provides about the preventing of the attack. Finally, we briefs the conclusion in section-9.

3 Mathematical Assumptions

3.1 Bilinear Pairings

Let $\mathbb{G}_1$ and $\mathbb{G}_2$ are two cyclic groups of prime order q with respect to operation addition and multiplication respectively. The bilinear map $\hat{e}$ is known as non-degenerated and computable if and only if it satisfies the following properties:

$$\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$$

holds following

- **Bilinearity**: Let $a, b \in \mathbb{Z}_q^*$ and $P, Q \in \mathbb{G}_1$
  1. $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$ for all $a, b \in \mathbb{Z}_q^*$
  2. $\hat{e}(P + Q, R) = \hat{e}(P, R)\hat{e}(Q, R)$, for $P, Q, R \in \mathbb{G}_1$.
- **Non-degenerate**: There exists $P \in \mathbb{G}_1$ such that $\hat{e}(P, P) \neq 1_{\mathbb{G}_2}$
- **Computability**: There exist an efficient algorithm to compute $\hat{e}(P, Q)$ or all $P, Q \in \mathbb{G}_1$.

This type of bilinear map is called as admissible.

3.2 Complexity assumption

Definition 1. *Computational Diffie-Hellman(CDH) Problem is defined for a given group $\mathbb{G}_1$ of prime order q with generator P and element $aP, bP \in \mathbb{G}_1$, where $a, b \in \mathbb{Z}_q^*$ are chosen randomly, the problem is to compute abP in $\mathbb{G}_1$.*

Definition 2. *The assumption (t, ϵ)-CDH holds in $\mathbb{G}_1$, if there does not exist any algorithm that can solve the CDH problem at most in time t with probability at least ϵ.*

4 Framework of Proxy multi-Signature Scheme

There are three type of entities involve on a proxy multi-signature scheme, namely a group of original signer $O = \{\mathcal{O}_1, \dots \mathcal{O}_n\}$ with identities $ID_1 \dots ID_n$, a proxy signer $\mathcal{P}$ with identity ID_p designated by all original signers and a verifier like a Clark or administrative assistant of an organization. A proxy multi-signature scheme comprises following seven algorithms:

- `Setup`: PKG runs this algorithm on the security parameters $1^\mu (\mu \in \mathbb{N})$ as input and returns the public parameters `params` and master secret key s. PKG makes `params` public and keeps secret s.

- `Extract:` This algorithm takes the user's identity ID, master secret key s and system parameter `params` and returns the user's private key d_{ID}. PKG runs this algorithm, generates the private keys of all users participating in this protocol and sends to the respective users through a secure channel.
- `Sign:` The algorithm takes the system parameters `params`, signer's identity ID, message m and signer's private key d_{ID} and generates signature σ on message m. The algorithm runs in a probabilistic polynomial time.
- `Veri:` This is a deterministic algorithm that verifies the validity of the signature. It takes the system public parameter `params`, signer's identity ID, signature σ and message m as input and returns 1 if the verification equation holds. Then accept the signature. Otherwise, it returns 0 if the equation does not hold and reject the signature.
- `PMGen:` This algorithm is run by both the proxy signer $\mathcal{P}$ and all original signers $\mathcal{O}_1, \ldots \mathcal{O}_n$ participate in the protocol. It takes the input of identities $ID_p, ID_1 \ldots ID_n$ of all members, original signer's private keys $d_1 \ldots d_n$ and warrant of delegation w contains a period of warrant, type of information delegated etc. Also it takes the proxy signer's private key d_p as input and returns the proxy signer's signing key sk_p which is used to generate proxy-multi signature on behalf of original signers.
- `PMSign:` This is a randomized algorithm, takes the proxy signing key sk_p, the message m to be signed and the warrant w as input and generate proxy multi-signature σ on behalf of all original signers $\mathcal{O}_1, \ldots \mathcal{O}_n$.
- `PMVeri:` This is a deterministic algorithm, takes all the member's identities $ID_p, ID_1 \ldots ID_n$, proxy multi-signature σ, message m and warrant w as input. Returns 1 if it passes through the verification equation and accept the signature, otherwise if the output is 0, reject it mean the equation does not holds.

5 Security Model

The security model is the game played between the adversary and the challenger. Assume that the challenger is a single honest user says 1. The adversary interact with the challenger provides his identity 1 and obtains all user's private keys participate in the protocols. It is to be assumed that the channel between the proxy signer and the original signer is not secure. The model allows the adversary to access the following three oracles as:

1. Signing
2. Delegation
3. Proxy multi-signature

The adversary $\mathcal{A}$ appeals to the user 1 to act as the role of proxy signer or one of the original signer. $\mathcal{A}$'s aim is to obtain one of the following forgeries:

- User 1 generates a typical signature for a message m with restriction, it was not queried earlier to the signing oracle.
- User 1 generates Proxy multi-signature for a message m on behalf of the original signers with the condition, neither 1 is designated by the original signers nor m was queried to the proxy multi-signature oracle.
- Proxy multi-signature for a message m by any user except the user 1 on behalf of the original signers $\mathcal{O}_2 \ldots \mathcal{O}_{n+1}$, such that any users have not been delegated by the original signers before and one of $\mathcal{O}_2 \ldots \mathcal{O}_{n+1}$ is the user 1.

Let the adversary is denoted by $\mathcal{A}$ and a challenger by $\mathcal{C}$. It is carried out in the following queries:

- `Setup`: $\mathcal{C}$ runs the setup algorithm on input the security parameters and outputs the public system parameter `params` and master secret key s. $\mathcal{C}$ keeps secret s and sends `params` to the adversary $\mathcal{A}$.
- `Extract query`: $\mathcal{A}$ can submit the query for the private key of the user's identity $i \neq 1$. $\mathcal{C}$ answers the query by running the Extract query and sends the private keys d_i to $\mathcal{A}$.
- `Queries for Signing`: $\mathcal{A}$ can ask a polynomial number of bounded Signing query in an adaptive manner on message m of his choice with the private key d_1 to the corresponding user 1. Returns the standard signature σ by 1. Then the message m is included to the list L_{q_s}.

- `Queries for delegation`: Consider the following two cases:
  - In this oracle, the user 1 plays as the role of proxy signer and other members $\mathcal{O}_2, \ldots \mathcal{O}_{n+1}$ plays as originally signers. $\mathcal{C}$ runs the algorithm `PMGen` for the message m on warrant w as the input of the user 1 chosen by $\mathcal{A}$ and returns the proxy signing key sk_p eventually. Then include (sk_p, w) to the list L_{warro}. We assume that $\mathcal{A}$ cannot access the list $L_{q_{warro}}$.
  - User 1 plays as one of $\mathcal{O}_2, \ldots \mathcal{O}_{n+1}$'s role and $\mathcal{A}$ plays the role of $\mathcal{P}$. Let without loss of generality, assume $\mathcal{O}_{n+1}$ is the proxy signer and $\mathcal{O}_1 \ldots \mathcal{O}_n$ are the original signers. $\mathcal{C}$ answers by executing the algorithm `PMGen` on warrant w as the input which is chosen by $\mathcal{A}$ and returns the proxy signing key sk_p eventually. Then include (sk_p, w) to the list $L_{q_{warro}}$.
- `Queries for Proxy multi-signature`: The adversary $\mathcal{A}$ can ask a polynomial number of bounded signing queries on (m, w) in an adaptive manner where there exists sk_p such that $(sk_p, w) \in L_{q_{warro}}$ and m satisfies w. Returns a proxy multi-signature σ on message m eventually and include to the list $L_{q_{Pms}}$.

If one of the following event take place, then $\mathcal{A}$ wins the game.

1. E_1: $\mathcal{A}$ forges a valid signature σ on the message m for user 1 where the verification equation hold and the query for m was not submitted to the signing oracle *i*.e $m \notin L_{q_s}$.
2. E_2: $\mathcal{A}$ generate a valid forge signature σ on the message m which satisfies the warrant w *i*.e $(m, w) \notin L_{q_{Pms}}$. Here the user 1 plays the role of proxy signer and other members $\mathcal{O}_2, \ldots \mathcal{O}_{n+1}$ are act as original signers.
3. E_3: $\mathcal{A}$ forges a valid proxy multi-signature σ on the message m with warrant $w \notin L_{q_{warro}}$. Where $\mathcal{O}_{n+1}$ is the proxy signer and $\mathcal{O}_1 \ldots \mathcal{O}_n$ are the original signers.

The probability of success is defined by the advantage of the adversary $\mathcal{A}$. Formally it can be written as:

$$Adv_{IDPMS}^{UF}(\mathcal{A}) = Pr[Succ]$$

Let the advantage for the adversary is ϵ. The success probability of $\mathcal{A}$ wins the above game is

$$Succ_{\mathcal{A}}^{UF}(\mu) \leq \tfrac{1}{2} + \epsilon$$

6 Review of Cao-Cao's Scheme

- `Setup:` Let $\mathbb{G}_1$ is a cyclic additive group of prime order q and P be the generator. G_2 be a cyclic multiplicative group with same prime order q. $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ is a bilinear map. PKG picks $s \in \mathbb{Z}_q^*$ randomly as master secret key and computes $P_{pub} = sP$. $H_i, i = 1, 2, 3, 4$ are cryptographic hash functions, $H_i : \{0,1\}^* \rightarrow \mathbb{G}_1, i = 1, 2, 3$ and $H_4 : \{0,1\}^* \rightarrow \mathbb{Z}_q^*$.
- `Extract:` Given the identity of the user ID, computes $Q_{ID} = H_1(ID) \in \mathbb{G}_1$ and the corresponding user's private key $d_{ID} = sQ_{ID} \in \mathbb{G}_1$.
- `Sign:` To generate a signature on message $m \in \{0,1\}^*$, the signer use his own private key d_{ID} and perform the following steps:
  1. Picks $r \in \mathbb{Z}_q^*$ randomly and computes $U = rP$ and $H = H_2(m \| U) \in \mathbb{G}_1$.
  2. Computes $V = d_{ID} + r \cdot H \in \mathbb{G}_1$

  Signature on message m is $\sigma = (U, V)$.
- `Veri:` For verification of the signature σ for the user's identity ID, the verifier performs the following steps:
  1. Computes $Q_{ID} = H_1(ID)$ and $H^{'} = H_2(m \| U)$.
  2. If the equation $\hat{e}(P, V) = \hat{e}(P_{pub}, Q_D)\hat{e}(U, H_2(m \| U)$ holds, the he accepts the signature, otherwise rejects.
- `PMGen:` To generate proxy multi-signature, the original signer performs the following:
  1. `Generation of delegation`: In order to delegate the power of signing to the proxy signer $\mathcal{P}$, the original signers $\mathcal{O}$ $\{\mathcal{O}_1 \ldots \mathcal{O}_n$ do the following to construct the signed warrant w which contains all the details of proxy includes identity information of the original and proxy signer, period of delegation and the type of information delegate. To sign on this delegation, original signers does the following to generate signed warrant w.

    - $\mathcal{O}_i$ picks $r_i \in \mathbb{Z}_q^*$ randomly and compute $U_i = r_iP$, for all $i = 1 \dots n$ and broadcast to other $n-1$ signers.
    - $\mathcal{O}_i$ computes $U = \sum_{i=1}^{n} U_i$, $H = H_2(w \| U)$, $V_i = d_i + r_iH$ for all $i = 1 \dots n$.
    - $\mathcal{O}_i$ sends (w, U_i, V_i) to the proxy signer $\mathcal{P}$.
  2. `Verification of delegation`: The proxy signer $\mathcal{P}$ verifies the validity of delegation after he received all (w_i, U_i, V_i) for all $i = \dots n$. He performed the following steps:
    - Computes $U^{'} = \sum_{i=1}^{n} U_i$ and $H^{'} = H_2(w \| U^{'})$.
    - Checks $\hat{e}(P, V_i) = \hat{e}(P_{pub}, Q_i)\hat{e}(U_i, H^{'})$, where $Q_i = H(ID_i)$, for all $i = 1 \dots n$.
  3. `Generation of Proxy secret key`: If all the delegations (w, U_i, V_i) for all $i = 1 \dots n$ are correct, $\mathcal{P}$ accepts the delegation and computes the proxy key as $sk_p = \sum_{i=1}^{n} V_i + H_4(ID_p \| w \| U)d_{ID_p}$.
- `PMSign:` $\mathcal{P}$ signs on message m on behalf of all the original signers $\mathcal{O}_1, \mathcal{O}_2 \dots \mathcal{O}_n$ using the secret proxy key sk_p. He performs the following steps:
  - Picks $r_p \in \mathbb{Z}_q^*$ randomly and computes $U_p = r_pP, H_p = H_3(ID_p \| w \| \| m \| U_p)$
  - Computes $V_p = sk_p + r_pH_p$.

  Proxy signature is $\sigma_p = (w, U_p, V_p, U)$ for message m.
- `PMVeri:` In order to verify the proxy signature σ_p under warrant w for the message, the verifier does the following computations:
  - Examines whether m complies to w or not. If it does comply, abort the simulation, else continue.
  - Verifies whether the original signers $\mathcal{O}_1, \mathcal{O}_2 \dots \mathcal{O}_n$ have authorized to the proxy signer $\mathcal{P}$ on the validated warrant w for the message m or not. If not abort the simulation, otherwise continue.
  - Finally, computes $Q_{ID_p} = H_1(ID_p)$ and verifies the following equation

$$\hat{e}(P, V_p) = \hat{e}(P_{pub}, \sum_{i=1}^{n} Q_i + H_4(ID_p \| w \| U \| Q_{ID_p})\hat{e}(U, H_2(w \| U))\hat{e}(U_p, H_3(ID_p \| w \| U_p)) \tag{1}$$

If holds, then accepts the proxy signature, otherwise rejects.

7 Vulnerability of Cao-Cao's Scheme

Under the defined security model, here we have proven that Cao-Cao's identity-based proxy multi-signature scheme is vulnerable to chosen message attack where $\mathcal{A}$ can forge a valid proxy multi-signature scheme.

Let $\mathcal{A}$ chooses a warrant $w^{'}$ where $w^{'} \notin Warro$ for the proxy signer or challenger with identity ID_1 and can forge a valid proxy multi-signature $(w^{'}, m^{'}, \sigma^{'})$ on behalf of original signers $\mathcal{O}_2 \dots \mathcal{O}_{n+1}$. Hence the event E_2 occurs and the adversary $\mathcal{A}$ wins the game.

To forge a valid proxy multi-signature $\mathcal{A}$ has to perform the following steps.

1. $\mathcal{A}$ constructs a warrant $w^{'}$ that the proxy signer $\mathcal{P}$ have identity ID_1 is designated by the original signer $\mathcal{O} = \{\mathcal{O}_2 \dots \mathcal{O}_{n+1}\}$ have identities $ID_2 \dots ID_{n+1}$. He adds the type of information delegated, period of delegations etc.
2. $\mathcal{A}$ submits signature queries on $(ID_i, w^{'})$ for $i = \{2 \dots n+1\}$. Then $\mathcal{A}$ returns the answers $(U_i^{'}, V_i^{'})$ satisfies $\hat{e}(P, V_i^{'}) = \hat{e}(P_{pub}, Q_{ID_i})\hat{e}(U_i^{'}, H_2(w^{'} \| U_i^{'}))$, for $i = \{2 \dots n+1\}$. Includes $w^{'}$ in the list L_s.
3. $\mathcal{A}$ submits extraction query for the proxy signer with identity ID_1 and returns proxy key as

   $$sk_{P_1}^{'} = \textstyle\sum_{i=\ 2}^{n+1} V_i^{'} + H_4(ID_1^{'} \| w^{'} \| U^{'})d_1, \text{ where } d_1 = sH_1(ID_1), U^{'} = \sum_{i=2}^{n+2} U_i^{'}$$
4. $\mathcal{A}$ can constructs a valid proxy multi-signature $\sigma^{'} = (w^{'}, U_1^{'}, V_1^{'}, U^{'})$. Where $U_1^{'} = r_1^{'}P, U_i^{'} = r_i^{'}P, U^{'} = \sum_{i=2}^{n+1} U_i^{'}, H_1^1 = H_3(ID_1 \| w^{'} \| m^{'} \| U_1^{'}), H^{'} = H_2(w^{'} \| U^{'}), V_1^{'} = sk_{P_1}^{'} + r_1^{'}H^{'}$.

<u>`Proof of Correctness`</u>

$$\hat{e}(P, V_1^{'}) = \hat{e}(P_{pub}, \textstyle\sum_{i=2}^{n+2} Q_i^{'} + H_4(ID_1^{'} \| w^{'} \| U^{'}), Q_{ID_1})e(U^{'}, H^{'})\hat{e}(U_1^{'}, H_1^{'})$$
$$\hat{e}(P, V_1^{'}) = \hat{e}(P, sk_{P_1}^{'}\hat{e}(P, r_1H_3(ID_1^{'} \| w^{'} \| m^{'} \| U_1^{'}))$$
$$= \hat{e}(P, \textstyle\sum_{i=2}^{n+2} V_i^{'})\hat{e}(P, H_4(ID_1^{'} \| w^{'} \| U^{'})d_1)\hat{e}(U_1^{'}, H_3(ID_1^{'} \| w^{'} \| m^{'} \| U_1^{'}))$$
$$= e(P_{pub}, \textstyle\sum_{i=2}^{n+2} Q_i^{'})\hat{e}(U^{'}, H^{'})\hat{e}(P_{pub}, H_4(ID_1^{'} \| w^{'} \| U^{'})Q_{ID_1})\hat{e}(U_1^{'}, H_1^{'})$$
$$= \hat{e}(P_{pub}, \textstyle\sum_{i=2}^{n+2} Q_i^{'} + H_4(ID_1^{'} \| w^{'} U^{'})Q_{ID_1})\hat{e}(U^{'}, H^{'})\hat{e}(U_1^{'}, H_1^{'})$$

8 Countermeasure of the Attack

In this section, we proposed the improved version of the scheme that eliminate the vulnerability by preventing the attack.

We follows the similar procedure [] to prevent the attack. Let we append 11 to the message in the form of binary string that shows an ordinary message, 00 to represent a proxy designated message and 01 to represent proxy multi-signature message in the proposed improved version of the scheme. The scheme is as follows

- `Setup:` Consider an additive group $\mathbb{G}_1$ which is a cyclic group. The Order of the group is q a prime and P be the generator. G_2 be a cyclic multiplicative group with the same order. Let $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ be a bilinear map. PKG picks $s \in \mathbb{Z}_q^*$ randomly as master secret key and computes $P_{pub} = sP$. $H_i, i = 1, 2, 3, 4$ are cryptographic hash functions, $H_i : \{0, 1\}^* \to \mathbb{G}_1, i = 1, 2, 3$ and $H_4 : \{0, 1\}^* \to \mathbb{Z}_q^*$.
- `Extract:` Given the identity of the user ID, computes $Q_{ID} = H_1(ID) \in \mathbb{G}_1$ and the corresponding user's private key $d_{ID} = sQ_{ID} \in \mathbb{G}_1$.
- `Sign:` To generate a signature on message $m \in \{0, 1\}^*$, the signer use his own private key d_{ID} and perform the following steps:
  1. Picks $r \in \mathbb{Z}_q^*$ randomly and computes $U = rP$ and $H = H_2(m\|U\|11) \in \mathbb{G}_1$.
  2. Computes $V = d_{ID} + r \cdot H \in \mathbb{G}_1$

  Signature on message m is $\sigma = (U, V)$.
- `Veri:` For verification of the signature σ for the user's identity ID, the verifier performs the following steps:
  1. Computes $Q_{ID} = H_1(ID)$ and $H^{'} = H_2(m\|U\|11)$.
  2. If the equation $\hat{e}(P, V) = \hat{e}(P_{pub}, Q_D)\hat{e}(U, H_2(m\|U\|11)$ holds, then he accepts the signature, otherwise rejects.
- `PMGen:` To generate proxy multi-signature, the original signer performs the following:
  1. `Generation of delegation`: In order to delegate the capability of signing to the proxy signer $\mathcal{P}$, the original signers $\mathcal{O}$ $\{\mathcal{O}_1 \ldots \mathcal{O}_n$ do the following to construct the signed warrant w which contains all the details of proxy includes information about original and proxy signer's identity,timing of delegation and the type of information delegate. To sign on this delegation, original signers does the following to generate signed warrant w.
     - $\mathcal{O}_i$ picks $r_i \in \mathbb{Z}_q^*$ randomly and compute $U_i = r_iP$, for all $i = 1 \ldots n$ and broadcast to other $n - 1$ signers.
     - $\mathcal{O}_i$ computes $U = \sum_{i=1}^{n} U_i$, $H = H_2(w\|U\|00)$, $V_i = d_i + r_iH$ for all $i = 1 \ldots n$.
     - $\mathcal{O}_i$ sends (w, U_i, V_i) to the proxy signer $\mathcal{P}$.
  2. `Verification of delegation`: The proxy signer $\mathcal{P}$ verifies the validity if delegation after he received all (w_i, U_i, V_i) for all $i = \ldots n$. He performed the following steps:
     - Computes $U^{'} = \sum_{i=1}^{n} U_i$ and $H^{'} = H_2(w\|U^{'}\|00)$.
     - Checks $\hat{e}(P, V_i) = \hat{e}(P_{pub}, Q_i)\hat{e}(U_i, H^{'})$, where $Q_i = H(ID_i)$, for all $i = 1 \ldots n$.
  3. `Generation of Proxy secret key`: If all the delegations (w, U_i, V_i) for all $i = 1 \ldots n$ are correct, $\mathcal{P}$ accepts the delegation and computes the proxy key as $sk_p = \sum_{i=1}^{n} V_i + H_4(ID_p\|w\|U)d_{ID_p}$.
- `PMSign:` $\mathcal{P}$ signs on message m on behalf of all the original signers $\mathcal{O}_1, \mathcal{O}_2 \ldots \mathcal{O}_n$ using the secret proxy key sk_p. He performed the following steps:
  - Picks $r_p \in \mathbb{Z}_q^*$ randomly and computes $U_p = r_pP, H_p = H_3(ID_p\|w\|\|m\|U_p\|01)$
  - Computes $V_p = sk_p + r_pH_p$.

  Proxy signature is $\sigma_p = (w, U_p, V_p, U)$ for message m.
- `PMVeri:` In order to verify the proxy signature σ_p under warrant w for the message, the verifier does the following computations:
  - Examines the message m complies to warrant w or not. If it does comply, abort, otherwise continue.
  - Verifies whether the original signers $\mathcal{O}_1, \mathcal{O}_2 \ldots \mathcal{O}_n$ have authorized to the proxy signer $\mathcal{P}$ on the validated warrant w for the message m or not. If not abort the simulation, otherwise continue.
  - Finally, computes $Q_{ID_p} = H_1(ID_p)$ and verifies the following equation

$$\hat{e}(P, V_p) = \hat{e}(P_{pub}, \sum_{i=1}^{n} Q_i + H_4(ID_p\|w\|U\|Q_{ID_p})\hat{e}(U, H_2(w\|U\|00))\hat{e}(U_p, H_3(ID_p\|w\|U_p\|01)) \tag{2}$$

If holds, then accepts the proxy signature, otherwise rejects.

9 Conclusions

In this paper, we have reviewed the Cao-Cao's proxy multi-signature scheme and point out that the scheme is not secure against chosen message attack under their defined security model *i*.e event E_2 occurs where the attacker can forge a valid proxy multi-signature. Further, we proposed an improved version of the scheme which can prevent this attack. For future work, the proposed scheme can be extended to multi-proxy multi-signature scheme in the random oracle model.

10 Acknowledgments

The author would like to thank to Deanship of Scientific Research(DSR), King Abdulaziz University, P.O.Box-80221, Jeddah-21589, Kingdom of Saudi Arabia for their technical and financial support. This work is supported by DSR under Distinct Study Research, grant No. `611-34-D1435`.

References

1. A. Shamir "Identity-based cryptosystems and signature schemes", *in Advances in Cryptology, vol. 196 of Lecture Notes in Computer Science*, pp. 47–53, Springer, Berlin, Germany, 1984.
2. M. Mambo, K. Usuda, E. Okamoto " Proxy signatures: delegation of the power to sign message, *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* E79-A (9), pp.1338–1353, 1996.
3. L. Yi, G. Bai, G. Xiao " Proxy multi-signature scheme: A new type of proxy signature scheme",*Electronics Letters* Vol-36(6), pp. 527–528, 2000.
4. "A security architecture for computational grids", *Fifth ACM Conference on Computers and Communications Security*, pp. 83–92, 1998.
5. B. Lee, H. Kim, K. Kim "Strong proxy signature and its applications", *Symposium on Cryptography and Information Security (SCIS02001)*, pp.603–608, 2001.
6. A. Bakker, M. Steen, A.S. Tanenbaum " A law-abiding peer-to-peer network for free-software distribution", *IEEE International Symposium on Network Computing and Applications*, pp.60–67,2001.
7. H. U.Park, I.Y. Lee " A digital nominative proxy signature scheme for mobile communication", *Information and Communications Security (ICICS 2001)*, LNCS 2229, 451-455, 2001.
8. S. Hwang, C. Shi "A simple multi-proxy signature scheme", *In: Proceedings of the 10th national conference on information security, Hualien, Taiwan, ROC*, pp.134–138, 200.
9. J. Leiwo, C. Hanle, P. Homburg, A.S. Tanenbaum "Disallowing unauthorized state changes of distributed shared objects", *Information Security for Global Information Infrastructures (SEC'00)"*, pp.381–390, 2000.
10. S.-J. Hwang, C.-C. Chen " New multi-proxy multi-signature schemes", *Applied Mathematics and Computation* Vol.147, pp. 57-67, 2004.
11. B. Lee, H. Kim, K. Won "Secure mobile agent using strong non-designated proxy signature",*In Proceeding of ACISP2001* , LNCS, vol. 2119, Springer-Verlag, pp. 474-486, 2001.
12. K. Zhang " Threshold proxy signature schemes", *In Proceedings of the first international workshop on information security*, pp. 282-90, 1997.
13. H. Kim, J. Baek, B. Lee, K. Kim "Secret computation with secrets for mobile agent using one-time proxy signature", *In Cryptography and information security* 2001.
14. A.K. Awasthi, S.Lal " Proxy blind signature scheme", *Trans Cryptol 2005*, vol-2(1), pp.5-11, 2005.
15. J. Kar "Proxy Blind Multi-signature Scheme using ECC for hand-held devices", *IACR Archive e-print 2011-043*, 2011.
16. J. Li, TH. Yuen, XF Chen, Y. Wang "Proxy ring signature: formal definitions, efficient construction and new variant", *In Proceedings of international conference of computational intelligence and security (CIS'06)*, vol. 2, pp. 1259-64, 2006.