2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications

Anonymous Proxy Signature with Restricted Traceability

Jiannan Wei
Centre for Computer and Information Security Research
School of Computer Science and Software Engineering,
University of Wollongong,
Wollongong, NSW 2522, Australia
Email: jw903@uowmail.edu.au

Guomin Yang
Centre for Computer and Information Security Research
School of Computer Science and Software Engineering,
University of Wollongong,
Wollongong, NSW 2522, Australia
Email: gyang@uow.edu.au

Yi Mu
Centre for Computer and Information Security Research
School of Computer Science and Software Engineering,
University of Wollongong,
Wollongong, NSW 2522, Australia
Email: ymu@uow.edu.au

***Abstract*—Signer anonymity is an important security requirement for many digital signature schemes due to the need of user privacy in many applications. In this paper, we study signer anonymity for proxy signature which allows a signer to delegate his/her signing right to another user (or proxy). Since the proxy signer is the actual singer in a proxy signature scheme, we are interested in protecting the proxy signer's identity in this paper. However, one potential problem in an anonymous proxy signature scheme is that the proxy signer may abuse the delegated signing right due to the anonymity property of a proxy signature. In this paper, we propose a novel anonymous proxy signature scheme with restricted traceability, which allows the original singer to trace dishonest proxy signers. However, if the proxy signer is honest, then his/her identity is well protected against any user (including the original signer). Our scheme will be useful in many applications such as anonymous authentication protocols and anonymous voting schemes.**

I. Introduction

Proxy signature, introduced by Mambo, Usuda and Okamoto in [1], is a special type of digital signature that allows a signer to delegate his/her signing right to another user, namely a proxy signer. The proxy signer can then sign messages on behalf of the original signer when the latter is unavailable. Proxy signature is widely used in many practical applications, such as electronic commerce [2], e-cash [3], mobile communications [4], and global distribution networks [5].

Based on the delegation type, Mambo et al. [1] classified proxy signature into three categories: *full delegation*, *partial delegation*, and *delegation by warrant*. In the last type, the added warrant can specify the identities of the proxy signers, the type of message that can be signed by the proxy signers, as well as the delegation period. In this paper, we only pay attention to warrant-based proxy signature, which is the most popular setting among all the three types of proxy signature schemes.

Signer anonymity is an important security requirement of many digital signature schemes such as Group signatures [6], Ring signatures [7], and Anonymous signature. In a ring signature scheme [7], any ring member can sign messages on behalf of the whole ring without revealing his/her real identity. Moreover, no one can determine whether two signatures have been signed by the same ring member or not. A group signature [6] is similar to a ring signature except that the former has a group manager who is able to reveal the identity of the signer for any valid group signature.

In order to protect the privacy of the proxy signers, several anonymous proxy signature schemes have been proposed [8][9][10][11][12][13][14]. The idea behind these schemes is to combine ring signature with proxy signature. However, there is a potential problem in these schemes: since the proxy signer's identity is protected, he/she may abuse the signing right without being caught.

One potential solution to solve the problem is to use a group signature instead of a ring signature in constructing an anonymous proxy signature. However, this would allow the group manager to trace any signature including those generated by honest proxy signers. Another possible solution to solve the problem is to combine a proxy signature scheme with a *traceable* ring signature scheme introduced by Fujisaki and Suzuki in [15]. Traceable ring signature is a tag-based signature with the restriction that a signer can sign messages only once per tag. If a ring member signs two different messages under the same tag twice, the his/her identity can be *publicly* traced. On the other hand, if the signer is honest (i.e., only signs once per tag), then the anonymity property can still be preserved. Traceable ring signature seems to be a good candidate for us. However, the public traceability supported by the Fujisaki-Suzuki traceable ring signature scheme may not be desirable in some applications. Take as an example, a company may only want to identify dishonest employees *internally* (i.e., outsiders cannot trace dishonest proxy signers) in order to protect the the image and reputation of the company.

In this paper, we solve the problem by proposing a new anonymous proxy signature scheme with *restricted traceability*. Our proposed scheme allows the original signer to trace of identities of dishonest proxy signers, but at the same time can keep an honest proxy signer's identity anonymous even against the original signer. Below we summarize the security features supported by our proposed scheme:

- **Unforgeability.** Only legitimate proxy signers can generate valid proxy signatures. Even the original signer cannot forge a proxy signature.

978-1-4799-6513-7/14 $31.00 © 2014 IEEE
DOI 10.1109/TrustCom.2014.73

- **Anonymity.** As long as the proxy signer behaves honestly (i.e., signs once per tag), his/her anonymity is ensured, that means even the original signer cannot determine which proxy signer has generated the signature or link two different signatures generated by the same proxy signer.
- **Restricted Traceability.** If the proxy signer is dishonest, then two proxy signatures generated by that proxy signer can be linked or traced by the original signer or other proxy signers who have also been delegated the same signing right. However, no outsider is able to link signatures generated by the dishonest proxy signer.
- **Exculpability.** An honest signer cannot be accused of being dishonest, i.e., an honest user cannot be framed by other users in the system.

Paper organization. In the next section, we provide the formal definition for anonymous proxy signature with restricted traceability. Then we present our concrete scheme in Section III, which is followed by the security models in Section IV and formal security proofs in Section V. We conclude the paper in Section VI.

II. Definition

An anonymous proxy signature with restricted traceability consists of the following algorithms:

Parameter generation: This is a probabilistic polynomial-time (PPT) algorithm that on input a security parameter κ outputs the system parameters $Param$.

Key Generation ($\mathcal{KG}$): Given the system parameters $Param$, it output a user public and private key pair (Y, x).

Delegation sign ($\mathcal{DS}$): Given a warrant m_ω and an original signer's private key x_0, it outputs a delegation signing key σ_0 for m_ω.

Delegation verification ($\mathcal{DV}$): Given an original signer's public key Y_0, a warrant m_ω, and a delegation signing key σ_0, it outputs "accept" if σ_0 is valid with regards to Y_0 and m_ω; otherwise, it outputs "reject".

Proxy Sign ($\mathcal{PS}$): On input a message $m \in \{0,1\}^*$, the original signer's public key Y_0, a tag $L = (issue, Y_N)$ where $Y_N = \{Y_1, \ldots, Y_n\}$ are the public keys of n proxy signers, a proxy signer's secret key $x_i (1 \leq i \leq n)$, and a delegation signing key σ_0 for a warrant m_ω, it outputs a proxy signature σ.

Verification($\mathcal{PV}$): On input a message m, the original signer's public key Y_0, a proxy signature σ, a tag $L = (issue, Y_N)$ as defined above, and a warrant m_ω, it outputs "accept" if the signature is valid; otherwise, it outputs "reject".

Trace ($\mathcal{TR}$): On input two message-signature pairs (m, σ) and (m', σ') with respect to the same tag L and warrant m_ω, and a valid delegation signing key σ_0 for m_ω, it outputs either "indep", "linked", or $Y_i \in Y_N$.

III. Our Anonymous Proxy Signature with Restricted Traceability

The idea behind our construction is to borrow the traceability technique in [15]. However, as mentioned earlier, the traceable ring signature in [15] only supports public traceability. So a challenge problem is to develop new techniques that could disallow outsiders to perform the trace operation. Our idea to solve the problem is to randomize each proxy signature so that only the original signer or another proxy signer who has also been delegated the same signing right has the secret to de-randomize the proxy signature. Below are the details of our proposed scheme.

Parameter generation. Taking as input the security parameter κ, this algorithm outputs (G, q, P) where G is a cyclic group of prime order q and P is a generator of G. Let $H_0 : \{0,1\} * G \to \mathbb{Z}_q$, $H : \{0,1\}^* \to G$, $H' : \{0,1\}^* \to G$ and $H'' : \{0,1\}^* \to \mathbb{Z}_q$ denote independent cryptographic hash functions. The system parameters are $Param = (G, q, P, H_0, H, H', H'')$.

Key generation. User i randomly selects $x_i \in \mathbb{Z}_q$ and computes $Y_i = x_i P$. The public key of user i is Y_i and the corresponding secret key is x_i.

Delegation sign. The original signer first generates a warrant m_ω. There is an explicit description of the delegation relation such as the identities of the original signer u_0 and the proxy signers $\mathcal{U} = \{u_1, u_2, \ldots, u_n\}$, the expiration time of the delegation, etc. Then the original signer randomly chooses $\alpha \in \mathbb{Z}_q$ and computes $W_o = \alpha P$. After that, the proxy signer picks another random number $r \in \mathbb{Z}_q$ and computes $R = rP$, $s = r + x_0 H_0(m_\omega, R, W_o) \ mod \ q$. Finally, the proxy signer sends (m_ω, α, R, s) to all the proxy signers in $\mathcal{U}$ via a secure channel.

Delegation verification. Upon receiving (m_ω, α, R, s), the proxy signer u_i checks if $sP = R + H_0(m_\omega, R, W_o = \alpha P) Y_0$. If it does not hold, the delegation is rejected. Otherwise, the proxy signer u_i computes his/her proxy signing secret key $psk_i = s + x_i H_0(m_\omega, R, W_o) = r + H_0(m_\omega, R, W_o)(x_0 + x_i) \ mod \ q$. For simplicity, let $pk_i = psk_i P = R + H_0(m_\omega, R, W_o)(Y_0 + Y_i)$ denote the corresponding proxy signing public key.

Proxy sign. To sign a message $m \in \{0,1\}^*$ with respect to a tag $L = (issue, Y_N)$ where Y_N are public keys of some proxy signers described in the the warrant m_ω, the real proxy signer u_i proceeds as follows:

1) Randomly choose $\beta \in \mathbb{Z}_q$, compute $F = H(L)$, $W_p = (W_{p1}, W_{p2}) = (\alpha\beta P, \beta F)$ and $\sigma_i = \alpha\beta F + psk_i F = (\alpha\beta + psk_i) F$.
2) Set $A_0 = H'(L, m)$ and $A_1 = \frac{1}{i}(\sigma_i - A_0)$.
3) For all $j \neq i$, compute $\sigma_j = A_0 + j A_1 \in G$. Note that every $(j, \log_F(\sigma_j))$ is on the line defined by $(0, \log_F(A_0))$ and $(i, psk_i + \alpha\beta)$.
4) Generate (c_N, z_N) based on a (non-interactive) zero-knowledge proof of knowledge for the language

$$\mathcal{L} = \{(L, F, \sigma_N) \mid \exists i \in N \ s.t. \log_P(pk'_i) = \log_F(\sigma_i)\}$$

where $\sigma_N = (\sigma_1, \sigma_2, \ldots, \sigma_n)$ and $pk'_i = W_{p1} + pk_i$ as follows:

   a) Pick random $\omega_i \leftarrow \mathbb{Z}_q$ and set $a_i = \omega_i P$, $b_i = \omega_i F \in G$.

b) Pick random $z_j, c_j \leftarrow \mathbb{Z}_q$, and set $a_j = z_j P + c_j pk'_j$, $b_j = z_j F + c_j \sigma_j \in G$ for every $j \neq i$.
c) Set $c = H''(L, m, A_0, A_1, a_N, b_N)$ where $a_N = (a_1, \ldots, a_n)$ and $b_N = (b_1, \ldots, b_n)$.
d) Set $c_i = c - \Sigma_{j \neq i} c_j \ mod \ q$ and $z_i = \omega_i - c_i(\alpha\beta + psk_i) \ mod \ q$.
e) Return (c_N, z_N), where $c_N = (c_1, \ldots, c_n)$ and $z_N = (z_1, \ldots, z_n)$, as a proof for $\mathcal{L}$.

5) Perform another (non-interactive) zero-knowledge proof of knowledge for

$$\mathcal{L}' = \{(F, W_{p2}, W_o, W_{p1}) \mid \log_{W_o} W_{p1} = \log_F W_{p2}\}$$

as follows
a) Pick random $\omega \leftarrow \mathbb{Z}_p$ and set $\tilde{a} = \omega W_o$, $\tilde{b} = \omega F \in G$.
b) Set $\tilde{c} = H''(L, m, A_0, A_1, \tilde{a}, \tilde{b})$.
c) Set $\tilde{z} = \omega - \tilde{c}\beta$.
d) Return $(\tilde{c}, \tilde{z})$ as a proof for $\mathcal{L}'$.

6) Return $\sigma = (A_1, R, W_o, W_p, c_N, z_N, \tilde{c}, \tilde{z})$ as the signature on (L, m).

Verification To verify a proxy signature $\sigma = (A_1, R, W_o, W_p, c_N, z_N, \tilde{c}, \tilde{z})$ on message m and tag L, check the following:

1) Parse L as $(issue, Y_N)$, and compute $pk'_i = W_{p1} + pk_i = W_{p1} + R + H_0(m_\omega, R, W_o)(Y_0 + Y_i)$ for all $i \in N$.
2) Set $F = H(L)$ and $A_0 = H'(L, m)$, and compute $\sigma_i = A_0 + iA_1 \in G$ for all $i \in N$.
3) Compute $a_i = z_i P + c_i pk'_i$, $b_i = z_i F + c_i \sigma_i$, for all $i \in N$.
4) Check that $H''(L, m, A_0, A_1, a_N, b_N) = \Sigma_{i \in N} c_i \ mod \ q$, where $a_N = (a_1, \ldots, a_n)$ and $b_N = (b_1, \ldots, b_n)$.
5) Compute $\tilde{a} = \tilde{z} W_o + \tilde{c} W_{p1}$, $\tilde{b} = \tilde{z} F + \tilde{c} W_{p2}$.
6) Check if $H''(L, m, A_0, A_1, \tilde{a}, \tilde{b}) = \tilde{c}$.
7) If all the above checks are successful, outputs accept; otherwise, outputs reject.

Correcness. the correctness of our scheme can be verified as follows

$$\begin{aligned}
& z_i P + c_i pk'_i \\
=& \Big[\omega_i - c_i\big[\alpha\beta + (r + H_0(m_\omega, R, W_o)(x_0 + x_i))\big]\Big] P + c_i pk'_i \\
=& \omega_i P - c_i\big[\alpha\beta + r + H_0(m_\omega, R, W_o)(x_0 + x_i)\big] P + \\
& c_i[\alpha\beta P + R + H_0(m_\omega, R, W_o)(Y_0 + Y_i)] \\
=& \omega_i P \\
=& a_i
\end{aligned}$$

$$\begin{aligned}
& z_i F + c_i \sigma_i \\
=& \Big[\omega_i - c_i\big[\alpha\beta + (r + H_0(m_\omega, R, W_o)(x_0 + x_i))\big]\Big] F \\
& + c_i(\alpha\beta F + psk_i F) \\
=& \omega_i F - c_i\big(\alpha\beta + (r + H_0(m_\omega, R, W_o)(x_0 + x_i))\big) F + c_i \alpha\beta F \\
& + c_i F(r + H_0(m_\omega, R, W_o)(x_0 + x_i)) \\
=& \omega_i F \\
=& b_i
\end{aligned}$$

$$\begin{aligned}
& \tilde{z} W_o + \tilde{c} W_{p1} \\
& = (\omega - \tilde{c}\beta)\alpha P + \tilde{c}\alpha\beta P \\
& = \omega\alpha P \\
& = \tilde{a}
\end{aligned}$$

$$\begin{aligned}
& \tilde{z} F + \tilde{c} W_{p2} \\
& = (\omega - \tilde{c}\beta) F + \tilde{c}\beta F \\
& = \omega F \\
& = \tilde{b}
\end{aligned}$$

Trace To check the relation between (m, σ) and (m', σ') under the same warrant m_ω and the same tag L where $\sigma = (A_1, R, W_o, W_p, c_N, z_N, \tilde{c}, \tilde{z})$ and $\sigma' = (A'_1, R, W_o, W'_p, c'_N, z'_N, \tilde{c}', \tilde{z}')$, check the following:

1) Parse L as $(issue, Y_N)$. Set $F = H(L)$ and $A_0 = H'(L, m)$. Compute $\sigma_i = A_0 + iA_1 \in G$ for all $i \in N$. Since $W_p = (\alpha\beta P, \beta F)$, with the secret α, the original signer or any proxy signer specified in the warrant m_ω can compute $\widehat{\sigma_i} = \sigma_i - \alpha\beta F = psk_i F \in G$ for all $i \in N$. Do the same operation for σ' to get $\widehat{\sigma'_i}$ for all $i \in N$.
2) For all $i \in N$, if $\widehat{\sigma_i} = \widehat{\sigma'_i}$, store pk_i in **TList**, where **TList** is initially empty.
3) Output pk if pk is the only entry in **TList**; "linked" if **TList** $= Y_N$; "indep" otherwise.

IV. SECURITY MODEL

In this section, we present the formal security models for anonymous proxy signature with restricted traceability.

A. Unforgeability

Type I Adversary (or an outsider). This type of adversary only has the public keys of the original signer and the proxy signers. His aim is to forge a valid proxy signature.

Type II Adversary is a proxy signer. This type of adversary has all the public keys and the private key of a proxy signer. His aim is to forge a delegation signing key for a warrant m_ω. Note that once he can forge the delegation signing key of a warrant m_ω, he could forge any proxy signature.

Type III Adversary is the original signer. This type of adversary has all the public keys and the private key of the original signer. His aim is to forge a valid proxy signature.

It is obvious that if an anonymous proxy signature scheme is unforgeable against Type II and Type III adversaries, it is also unforgeable against Type I adversary. So we will only focus on the adversarial models with regards to Type II and Type III adversaries in the rest of this paper.

1) Unforgeability against type II adversary: $\mathcal{A}_{II}$ aims to forge a valid delegation signing key for a warrant m^*_ω. The model is defined via the following game.

Setup: The challenger $\mathcal{C}$ runs the key generation algorithm to obtain the secret key and public key pairs (x_0, Y_0), (x_1, Y_1), $\ldots$, (x_n, Y_n) representing the keys of the original signer and n proxy signers, respectively. $\mathcal{C}$ then sends $(Y_0, Y_1, \ldots, Y_n, x_1, \ldots, x_n)$ to the adversary $\mathcal{A}_{II}$.

Delegation signing queries: $\mathcal{A}_{II}$ can request a delegation signing key on any warrant m_ω he chooses. In response, $\mathcal{C}$ returns a signature σ_0 for m_ω.

Output: Finally, $\mathcal{A}_{II}$ outputs a target warrant m_ω^* and σ^* and we say $\mathcal{A}_{II}$ wins the game if

- σ^* is a valid delegation signing key on m_ω^*; and
- m_ω^* has never appeared in the delegation signing queries.

We define the advantage of the adversary as

$$\mathbf{Adv}_{\mathcal{A}_{II}}^{UF}(k) = \Pr[\ \mathcal{A}_{II} \text{ wins the game}].$$

2) Unforgeability against Type III adversary: $\mathcal{A}_{III}$ is an original signer who aims to forge an anonymous proxy signature. It is defined via the following security game.

Setup: The challenger $\mathcal{C}$ runs the key generation algorithm to obtain the secret key and public key pairs (x_0, Y_0), (x_1, Y_1), $\ldots$, (x_n, Y_n) representing the keys of the original signer and n proxy signers, respectively. $\mathcal{C}$ then sends $(Y_0, Y_1, \ldots, Y_n, x_0)$ to the adversary $\mathcal{A}_{III}$.

Proxy signing queries: $\mathcal{A}_{III}$ can access the proxy signing oracles: $\mathbf{Sig}_{psk_i}$ for any $1 \le i \le n$ by providing a valid delegation signing key σ_0 for any warrant m_ω, and a tag L which includes Y_i. $\mathcal{C}$ then generates the proxy signature using psk_i and returns it to $\mathcal{A}_{III}$.

Output: Finally, $\mathcal{A}_{III}$ outputs a warrant m_ω, a tag $L = (issue, Y_N)$, and a message-signature pair (m^*, σ^*) and we say $\mathcal{A}_{III}$ wins the game if

- σ^* is a valid proxy signature; and
- (m_ω, L, m^*) has never appeared in the proxy signing queries.

We define the advantage of the adversary as

$$\mathbf{Adv}_{\mathcal{A}_{III}}^{UF}(k) = \Pr[\ \mathcal{A}_{III} \text{ wins the game}].$$

B. Restricted Traceability

We separate the restricted traceability into two models. First, we define Tag-linkability against the proxy signer, and then the Untraceability against outsiders.

1) Tag-linkability: Tag-linkability is defined by following the security model given in [15]. The adversary $\mathcal{A}$ which is a probabilistic polynomial-time algorithm take as input the system parameters, and outputs the original signer's public key Y_0, a warrant m_ω and a delegation signing key σ_0, a tag $L = (issue, Y_N)$ where $Y_N = (Y_1, \ldots, Y_n)$, and $n+1$ message-signature pairs $\{(m^{(1)}, \sigma^{(1)}), \ldots, (m^{(n+1)}, \sigma^{(n+1)})\}$. We define the adversary's advantage as

$$\mathbf{Adv}_{\mathcal{A}}^{TL}(k) = \Pr[\mathbf{Expt}^{(\mathcal{A})}(k) = 1]$$

where $\mathbf{Expt}^{\mathcal{A}}(k)$ is defined as follows:

1) $(Y_0, m_\omega, \sigma_0, L, \{(m^{(1)}, \sigma^{(1)}), \ldots, (m^{(n+1)}, \sigma^{(n+1)})\}) \leftarrow \mathcal{A}(1^k)$;
2) Return 1 iff
   - $\mathcal{DV}(Y_0, m_\omega, \sigma_0) = 1$, and
   - $\mathcal{PV}(Y_0, m_\omega, L, m^{(i)}, \sigma^{(i)}) = 1$, for all $i \in \{1, \ldots, n+1\}$, and
   - $\mathcal{TR}(Y_0, m_\omega, \sigma_0, L, m^{(i)}, \sigma^{(i)}, m^{(j)}, \sigma^{(j)}) =$ "indep" for all $i, j \in \{1, \ldots, n+1\}$ where $i \ne j$.

2) Untraceability against outsider: it is defined via the following game.

Setup: The challenger $\mathcal{C}$ runs the key generation algorithm to obtain the secret key and public key pairs (x_0, Y_0), (x_1, Y_1), $\ldots$, (x_n, Y_n) of the original signer and n proxy signers, respectively. $\mathcal{C}$ then sends $(Y_0, Y_1, \ldots, Y_n)$ to the adversary $\mathcal{A}$.

Key selection: The adversary $\mathcal{A}$ outputs a warrant m_ω, a tag $L = (issue, Y_N)$, and (Y_i, Y_j) where $i, j \in N$ as the two target proxy signer's public keys. The challenger then randomly selects $b \in \{i, j\}$.

Proxy signing query: $\mathcal{A}$ may access three signing oracles: $\mathbf{Sig}_{psk_b}$, $\mathbf{Sig}_{psk_i}$ and $\mathbf{Sig}_{psk_j}$ for the warrant m_ω and the tag L where

- $\mathbf{Sig}_{psk_b}$ is the signing oracle with respect to proxy signer b (notice that $b \in \{i, j\}$) who has a valid proxy signing key psk_b;
- $\mathbf{Sig}_{psk_i}$ (resp. $\mathbf{Sig}_{psk_j}$) is the signing oracle with respect to proxy signer i (resp. proxy signer j) who has a valid proxy signing key psk_i (resp. psk_j).

Output: Finally, $\mathcal{A}$ outputs a bit b'. $\mathcal{A}$ wins the game if $b' = b$. Define

$$\mathbf{Adv}_{\mathcal{A}}^{UT}(k) = \Pr[b' = b] - \frac{1}{2}.$$

C. Anonymity against original signer

We define the anonymity against the original signer via the following game.

Setup: The challenger $\mathcal{C}$ runs the key generation algorithm to obtain the secret key and public key pairs (x_0, Y_0), (x_1, Y_1), $\ldots$, (x_n, Y_n) representing the keys of the original signer and n proxy signers, respectively. $\mathcal{C}$ then sends $(Y_0, Y_1, \ldots, Y_n, x_0)$ to the adversary $\mathcal{A}$.

Key selection: The adversary $\mathcal{A}$ outputs (Y_i, Y_j) as the two target proxy signer's public keys. Let $b \in \{i, j\}$ be a random bit chosen by the challenger.

Proxy signing query: $\mathcal{A}$ may access three signing oracles: $\mathbf{Sig}_{psk_b}$, $\mathbf{Sig}_{psk_i}$ and $\mathbf{Sig}_{psk_j}$ by providing a valid delegation signing key σ_0 for any warrant m_ω, and a tag L which includes both Y_i, Y_j.

- $\mathbf{Sig}_{psk_b}$ is the signing oracle with respect to proxy signer b (notice that $b \in \{i, j\}$) who has a valid proxy signing key psk_b derived based x_b and σ_0;
- $\mathbf{Sig}_{psk_i}$ (resp. $\mathbf{Sig}_{psk_j}$) is the signing oracle with respect to proxy signer i (resp. proxy signer j) who

has a valid proxy signing key psk_i (resp. psk_j) derived based on x_i (resp. x_j) and σ_0.

The following conditions must hold for all the signing queries made by $\mathcal{A}$:

- If (L, m) and (L, m') are two queries of $\mathcal{A}$ to the challenge signing oracle $\mathbf{Sig}_{psk_b}$, then $m = m'$.
- If (L, m) is a query of $\mathcal{A}$ to $\mathbf{Sig}_{psk_b}$ and $(\hat{L}, \hat{m})$ is a query of $\mathcal{A}$ to $\mathbf{Sig}_{psk_i}$ or $\mathbf{Sig}_{psk_j}$, then $L \neq \hat{L}$.

Output: Finally, $\mathcal{A}$ outputs a bit b'. $\mathcal{A}$ wins the game if $b' = b$. Define the advantage of $\mathcal{A}$ as

$$\mathbf{Adv}_{\mathcal{A}}^{AN}(k) = \Pr\left[b = b'\right] - \frac{1}{2}.$$

D. Exculpability

The exculpability of an anonymous proxy signature scheme with restricted traceability is defined via the following game.

Setup: The challenger $\mathcal{C}$ runs the key generation algorithm to obtain the secret key and public key pairs (x_0, Y_0), (x_1, Y_1), $\ldots$, (x_n, Y_n) representing the keys of the original signer and n proxy signers, respectively. $\mathcal{C}$ then sends $(Y_0, Y_1, \ldots, Y_n)$ to the adversary $\mathcal{A}$.

Key selection: The adversary $\mathcal{A}$ outputs (Y_i) as the target proxy signer's public key. The challenger then gives $(x_0, x_1, \cdots, x_{i-1}, x_{i+1}, \cdots, x_n)$ to $\mathcal{A}$.

Proxy signing query: $\mathcal{A}$ may access the signing oracles $\mathbf{Sig}_{psk_i}$ by providing a valid delegation signing key σ_0 for any warrant m_ω, and a tag L which includes Y_i. $\mathbf{Sig}_{psk_i}$ is the signing oracle with respect to the target proxy signer i who has a valid proxy signing key psk_i derived based on x_i and σ_0.

Output: Finally, $\mathcal{A}$ outputs a warrant m_ω, a valid delegation signing key σ_0, a tag $L = (issue, Y_N)$, and two message-signature pairs (m, σ) and (m', σ') such that

- $Y_i \in Y_N$, and
- $\mathcal{PV}(Y_0, m_\omega, L, m, \sigma) = 1$, and
- $\mathcal{PV}(Y_0, m_\omega, L, m', \sigma') = 1$, and
- at least one of $(m_\omega, \sigma_0, L, m, \sigma)$ or $(m_\omega, \sigma_0, L, m', \sigma')$ is not linked to any $(m_\omega, \sigma_0, L, m'', \sigma'')$ that has appeared in the proxy signing queries.

Define the advantage of the adversary $\mathcal{A}$ as

$$\mathbf{Adv}_{\mathcal{A}}^{EX}(k) = \Pr\left[\mathcal{TR}(Y_0, m_\omega, \sigma_0, L, m, \sigma, m', \sigma') = Y_i\right].$$

V. Security Analysis

In this section, we analyse the security of our proposed anonymous proxy signature with restricted traceability. The security proofs for Tag-linkability, Anonymity, Exculpability, and Unforgeability against the Type III adversary can be obtained straightforwardly by following the corresponding proofs in [15] where the proxy signers in our scheme have the same role as the ring members in [15]. Below we focus on the security proofs for two new security properties, namely *Unforgeability against the Type II adversary* and *Untraceability against outsiders*, that are important for our anonymous proxy signature with restricted traceability.

A. Assumptions

Definition 1 (Discrete Log (DL) Problem): Let G denote a cyclic group of order q and P a generator of G. Given $(P, xP) \in G^2$ for a randomly selected $x \in \mathbb{Z}_q$, compute x. The advantage of an algorithm $\mathcal{A}$ to solve the DLP is defined as

$$Adv_{\mathcal{A}}^{DLP}(k) = \Pr[\mathcal{A}(P, xP) = x]$$

Definition 2 (Decisional Diffie-Hellman (DDH) Problem): Let (G, q, P) be defined as in the the DL problem. Given $(P, xP, yP, zP) \in G^4$, decide whether $z = xy$. The advantage of an algorithm $\mathcal{A}$ to solve the DDH is defined as

$$Adv_{\mathcal{A}}^{DDH}(k)$$
$$= \Pr[\mathcal{A}(P, xP, yP, xyP) = 1] - \Pr[\mathcal{A}(P, xP, yP, rP) = 1]$$

where x, y, r are randomly chosen in $\mathbb{Z}_q$.

B. Unforgeability against The Type II Adversary

Theorem 1: If there exists a type II adversary $\mathcal{A}_{\mathcal{II}}$ which can break the proposed anonymous proxy signature scheme, then we can construct another adversary $\mathcal{B}$ who can use $\mathcal{A}_{\mathcal{II}}$ to solve the Discrete Log problem.

Proof. Given $(P, Y^* = x^*P)$ for some unknown $x^* \in \mathbb{Z}_q$ as an instance of DL problem, we will show how $\mathcal{B}$ can use $\mathcal{A}_{\mathcal{II}}$ to get the value x^*. $\mathcal{B}$ sets the original signer's public key $Y_0 = Y^* = x^*P$, and generates the keys for the proxy signers honestly. After that, $\mathcal{B}$ sends Y_0 and all the public/private key pairs of the proxy signers to adversary $\mathcal{A}_{\mathcal{II}}$.

$\mathcal{B}$ maintains a H_0 list to record all of hash queries/answers as follows:

H_0 **hash queries:** $\mathcal{A}_{\mathcal{II}}$ send the query (m_ω, R, W_o), $\mathcal{B}$ will check the H_0 list.

1) If (m_ω, R, W_o) has already been queried to H_0 oracle, and there is a record of $((m_\omega, R, W_o), h_i)$ in the H_0 list, $\mathcal{B}$ simply returns h_i to $\mathcal{A}_{\mathcal{II}}$.
2) Otherwise, $\mathcal{B}$ choose a random number $h_i \in \mathbb{Z}_q$, adds $((m_\omega, R, W_o), h_i)$ to the H_0 list, and returns h_i to $\mathcal{A}_{\mathcal{II}}$.

Delegation signing queries: For each query $m_{\omega i}$ chosen by $\mathcal{A}_{\mathcal{II}}$, $\mathcal{B}$ performs the following steps:

1) Randomly choose $c, s, \alpha \in \mathbb{Z}_q^*$ and compute $R = sP - cY^*$.
2) Set $W_o = \alpha P$, $H_0(m_\omega, R, W_o) = c$ and store $((m_\omega, R, W_o), c)$ into the H_0 list.
3) Return $\sigma_0 = (\alpha, R, s)$ as the delegation signing key for m_ω.

Finally, $\mathcal{A}_{\mathcal{II}}$ outputs $\sigma_0 = (\alpha^*, R^*, s^*)$ which is a valid delegation signing key for the warrant m_ω^*. Notice that m_ω^* should not have been queried before.

Based on the forking lemma, by rewinding $\mathcal{A}_{\mathcal{II}}$, $\mathcal{B}$ can obtain $s_1^* = r^* + x_0 c_1^* \ mod\ q)$ and $s_2^* = r^* + x_0 c_2^* \ mod\ q$ where c_1^* and c_2^* are the two hash outputs of $H_0(m_\omega^*, R^*, W_o^*)$. Therefore, $\mathcal{B}$ can output

$$x_0^* = \frac{s_1^* - s_2^*}{c_1^* - c_2^*} \ mod\ q$$

as the value of x^* and solve the DL problem. $\square$

C. Untraceability against Outsiders

Theorem 2: If there exists an outsider adversary $\mathcal{D}$ which can break the untraceability of the proposed anonymous proxy signature scheme, we can construct another adversary $\mathcal{B}$ who can solve the DDH problem.

Proof. If there exists an adversary $\mathcal{D}$ who can correctly guess b with an non-negligible advantage ϵ, we can construct another algorithm $\mathcal{B}$ that can solve the DDH problem. Let (P, aP, bP, zP) be a given instance of the DDH problem. We construct $\mathcal{B}$ as follows:

Setup: $\mathcal{B}$ generates the user public and private keys $(Y_0, x_0), (Y_1, x_1), \ldots, (Y_n, x_n)$ for the original signer and the proxy signers by running the key generation algorithm. $\mathcal{B}$ then gives all the public keys $Y_0, Y_1, \ldots, Y_n$ to the adversary $\mathcal{D}$.

Key selection: $\mathcal{D}$ outputs a warrant m_ω, a tag $L = (issue, Y_N)$, and (Y_i, Y_j) where $i, j \in N$ as the two target proxy signer's public keys. $\mathcal{B}$ then sets $W_o = aP$ and $F = H(L) = bP$, randomly selects $r \in \mathbb{Z}_q$ and computes $R = rP$ and $s = r + x_0 H_0(m_\omega, R, W_o)$. $\mathcal{B}$ also randomly selects $b \in \{i, j\}$, and answers $\mathcal{D}$'s queries as follows.

Hash queries: All the hash queries made by $\mathcal{D}$ are answered as in the previous proof where $\mathcal{B}$ maintains a hash table for each hash oracle.

Proxy signing queries: When $\mathcal{D}$ makes a proxy signing query to $\mathbf{Sig}_{psk_i}$ on message m, $\mathcal{B}$ randomly selects $\beta \in \mathbb{Z}_q$, and computes $W_p = (\beta W_o, \beta F)$ and $\sigma_i = \beta zP + psk_i F$. $\mathcal{B}$ generates A_0, A_1 and $\sigma_j (j \neq i)$ by following the proxy signing algorithm. $\mathcal{B}$ also simulates the NIZK proof for language $\mathcal{L}$ using the following simulator.

NIZK Simulator:

1) For all $i \in N$, uniformly pick up at random $z_i, c_i \in \mathbb{Z}_q$, and compute $a_i = z_i P + c_i pk_i'$, $b_i = z_i F + c_i \sigma_i \in G$.
2) Set $H''(L, m, A_0, A_1, a_N, b_N)$ as $c := \Sigma_{i \in N} c_i$ where $a_N = (a_1, a_2, \ldots, a_n)$ and $b_N = (b_1, b_2, \ldots, b_n)$.
3) Output (c_N, z_N), where $c_N = (c_1, \ldots, c_n)$ and $z_N = (z_1, \ldots, z_n)$.

$\mathcal{B}$ also simulates the NIZK proof $(\tilde{c}, \tilde{z})$ for language $\mathcal{L}'$ honestly using the knowledge of β. Finally, $\mathcal{B}$ returns $\sigma = (A_1, R, W_o, W_p, c_N, z_N, \tilde{c}, \tilde{z})$ to $\mathcal{D}$.

$\mathcal{B}$ also uses the same method to simulate signing query to $\mathbf{Sig}_{psk_j}$. Notice that since $b \in \{i, j\}$, the signing queries to $\mathbf{Sig}_{psk_b}$ are simulated using either $\mathbf{Sig}_{psk_i}$ or $\mathbf{Sig}_{psk_j}$.

Output: Finally, $\mathcal{D}$ output b'. If $b = b'$, $\mathcal{B}$ output 1; otherwise, $\mathcal{B}$ outputs 0.

Analysis: if (P, aP, bP, zP) is a DDH tuple, then the simulation is identical to the original game, and hence the adversary $\mathcal{D}$ has probability $\frac{1}{2} + \epsilon$ to guess b correctly. On the other hand, if (P, aP, bP, zP) is not a DDH tuple, i.e., z is a random element of $\mathbb{Z}_q$, then the simulation does not reveal any information of b, and hence $\mathcal{D}$ has probability $\frac{1}{2}$ to guess b correctly. Therefore, the advantage of $\mathcal{B}$ to solve the DDH problem is at least ϵ. $\square$

VI. Conclusion

In this paper, we proposed an efficient anonymous proxy signature scheme with restricted traceability. Different from the existing anonymous proxy signature schemes, our new scheme allows the original signer to trace dishonest proxy signers. However, if the proxy signer is honest, then his/her identity is well protected against any user (including the original signer). Another interesting feature of our scheme is that outsiders will not be able to trace any proxy signatures even if the proxy signer is dishonest. This feature is important if a company only wants to trace dishonest users internally. We also provided formal security models for different adversaries and proved the security of our scheme under some standard assumptions.

References

[1] Mambo M, Usuda K, Okamoto E. Proxy signatures: delegation of the power to sign messages. IEICE transactions on fundamentals of electronics, communications and computer sciences. 1996;79(9):1338–1354.

[2] Dai JZ, Yang XH, Dong JX. Designated-receiver proxy signature scheme for electronic commerce. In: Systems, Man and Cybernetics, 2003. IEEE International Conference on. vol. 1. IEEE; 2003. p. 384–389.

[3] Hu X, Huang S. A novel proxy key generation protocol and its application. Computer Standards & interfaces. 2007;29(2):191–195.

[4] Wang G, Bao F, Zhou J, Deng RH. Security analysis of some proxy signatures. In: Information Security and Cryptology-ICISC 2003. Springer; 2004. p. 305–319.

[5] Bakker A, Van Steen M, Tanenbaum AS. A law-abiding peer-to-peer network for free-software distribution. In: Network Computing and Applications, 2001. NCA 2001. IEEE International Symposium on. IEEE; 2001. p. 60–67.

[6] Chaum D, Van Heyst E. Group signatures. In: Advances in Cryptology-EUROCRYPT'91. Springer; 1991. p. 257–265.

[7] Rivest RL, Shamir A, Tauman Y. How to leak a secret. In: Advances in Cryptology-ASIACRYPT 2001. Springer; 2001. p. 552–565.

[8] Zhang F, Safavi-Naini R, Lin CY. New Proxy Signature, Proxy Blind Signature and Proxy Ring Signature Schemes from Bilinear Pairing. IACR Cryptology ePrint Archive. 2003;2003:104.

[9] Zhang F, Safavi-Naini R, Lin CY. Some new proxy signature schemes from pairings. In: Progress on Cryptography. Springer; 2004. p. 59–66.

[10] Cheng W, Lang W, Yang Z, Liu G, Tan Y. An identity-based proxy ring signature scheme from bilinear pairings. In: Computers and Communications, 2004. Proceedings. ISCC 2004. Ninth International Symposium on. vol. 1. IEEE; 2004. p. 424–429.

[11] Li J, Chen X, Yuen TH, Wang Y. Proxy ring signature: formal definitions, efficient construction and new variant. In: Computational Intelligence and Security, 2006 International Conference on. vol. 2. IEEE; 2006. p. 1259–1264.

[12] Yu Y, Xu C, Huang X, Mu Y. An efficient anonymous proxy signature scheme with provable security. Computer Standards & Interfaces. 2009;31(2):348–353.

[13] Xiong H, Qin Z, Li F. A Certificateless Proxy Ring Signature Scheme with Provable Security. IJ Network Security. 2011;12(2):92–106.

[14] Toluee R, Asaar MR, Salmasizadeh M. An anonymous proxy signature scheme without random oracles. IACR Cryptology ePrint Archive. 2012;2012:313.

[15] Fujisaki E, Suzuki K. Traceable ring signature. IEICE transactions on fundamentals of electronics, communications and computer sciences. 2008;91(1):83–93.