Identity-Based Threshold Key-Insulated Proxy Signature in the Standard Model

Jianhong Chen1*

1Faculty of Computer and Software Engineering, Huaiyin Institute of Technology, Huaian, Jiangsu, 223003, China
* 139596433@qq.com

Abstract. Loss of secret keys is the deadly attack in public key cryptosystems. We present an identity-based threshold key-insulated proxy signature (IBTKIPS) scheme to settle the problem of key exposure in the identity-based proxy signature system. The proposed IBTKIPS scheme is secure in the standard model. In the IBTKIPS scheme, we use at least \( k \) out of \( n \) helper keys to refresh the temporary secret keys. The IBTKIPS scheme is more flexible and securer than the identity-based proxy signature scheme.

1. Introduction

It is very important for users to protect their secret keys in public key cryptosystems. The key insulation mechanism[2,3,6,8] was a good tool for key protection. Threshold key-insulation[7] was put forward in 2008. Threshold key-insulation in which at least \( k \) out of \( n \) helpers are used to refresh the user's secret keys. In the proxy signature system[4], the original signer delegates the signing right to the proxy signer. In the identity-based key-insulated proxy signature (IBKIPS)[1,5] system, the key-insulated mechanism is used to protect the secret keys. However, the IBKIPS scheme does not work well in some scenarios since only a helper protects the temporary secret keys. In these scenarios, at least \( k \) out of \( n \) helpers can be used to enhance the security. To settle the the problem of key exposure when at least \( k \) out of \( n \) helpers are used to refresh the secret keys, we give an identity-based threshold key-insulated proxy signature (IBTKIPS) scheme. The IBTKIPS scheme is secure in the standard model.

2. Model of IBTKIPS

2.1. Definition of IBTKIPS

In this paper, \( \mathbb{Z}_p^* \), \( x \in \mathbb{R} \), bilinear pairings, CDH assumption and PRF[1] are used. The proposed IBTKIPS scheme includes the following algorithms:

I. Extraction of public parameters and master secret key: the secret key generator uses this algorithm; the security parameter is the input; a master secret key and the public parameters are the output.

II. Extraction of \( n \) helper keys and initial secret key: the secret key generator uses this algorithm; the user’s identity is the input; a master secret key and the public parameters are the output.

III. Renew the key-refresh information: the \( i \)-th helper uses this algorithm; the user’s identity is the input; the \( i \)-th refresh key share \( d_{u,t,i} \) for this identity during time perios \( t \) are the output.
IV. Renew the temporary secret keys: the user uses this algorithm; the period index, the user’s identity, the temporary secret key of i’, and an refresh key share set $\Omega$ of for period t ($|\Omega|\geq d$) of the user and period index are the input; the temporary secret key are the output.

V. Signature delegation: the original signer uses this algorithm; the period index, the temporary secret key of the original signer and the period index, and the warrant which means the delegation relation are the input; the period index and the proxy certificate which make a tuple are the output.

VI. Verify the proxy certificate: the verifier uses this algorithm; the original signer’s identity and the proxy certificate tuple are the input; “continues” or “terminates the protocol” are the output.

VII. Renew temporary proxy secret keys: this algorithm is used by the proxy signer; the proxy certificate tuple and the period index are the input; the temporary proxy secret key of the proxy signer are the output.

VIII. Generate the proxy signature: the proxy signer uses this algorithm in the time period; the message, the warrant and the temporary secret key are the input; the period index, the proxy certificate and the proxy signature which make a tuple are the output.

IX. Verify the proxy signature: the verifier uses this algorithm; the original signer’s identity and the proxy signature tuple are the input; true or false is the output.

2.2. Security notions for IBTKIPS
We omit the security notions of our proposed identity-based threshold key-insulated proxy signature since the security notions can be gotten according to the proof of the identity-based key-insulated proxy signature[1] and the threshold key-insulated encryption[3].

3. Our Proposed IBTKIPS Scheme
I. Extraction of public parameters and master secret key:

1) Group $G_1$ and group $G_2$ are both groups of prime order p of size $\kappa$; $e: G_1 \times G_1 \rightarrow G_2$ is a bilinear pairing.

2) $F$ is a pseudo-random function of a $\kappa$-bit seeds and a $\kappa$-bit argument and outputs a $\kappa$-bit string $F_3(x)$.

3) $H_\alpha: \{0,1\}^* \rightarrow \{0,1\}^\kappa$ is a collision-resistant hash function with $n_u \in \mathbb{Z}$.

4) Get $\alpha \in \mathbb{Z}_p^*$; let $g_2 \in G_2$.

5) Get $u',v',m' \in \mathbb{Z}_p$; let $U=(u)$ of length $n_u$; let $V=(v)$ of length $n_v$; let $M=(m)$ of length $n_m$; $u$, $v$, and $v_i$ are all random elements of $G_1$.

6) The public parameters are $(G_1,G_2,e,g,g_1,g_2,u',U,v',V,m',M)$.

7) The master secret is $g_2^\alpha$ are the master secret.

II. Extraction of n helper keys and initial secret key:

1) $u$ is an identity and a bitstring of length $n_u$; $u[i]$ is the i-th bit of the identity.

2) $U_{u,0} \subset \{1,2,\ldots,n_u\}$ is the set of indices k and $u_0[i]=1$.

3) $i \in \{1,\ldots,d-1\}$, $c_i, r_i \in \mathbb{Z}_p$, and sets the $i$-th helper key as $h_{u;i}, h_{u;i}$ = ($g_2^\alpha, (u' \prod_{k \neq i} u_k)^{y_i}g_2^{v_i}$)

set the initial private key, $d_{u,0}$, as

\[ (d_0^{1}, d_0^{2}, d_0^{3}, d_0^{4}) = (g_2^\alpha, \ldots, \ldots) \]

\[ h_{u;0} \text{ is the helper key of the original signer. Let the initial secret key } d_{0,0} \text{ of the original signer be } \]

\[ (d_0^{1}, d_0^{2}, d_0^{3}, d_0^{4}) \]

\[ = (g_2^\alpha, \ldots, \ldots) \]

\[ H_{p;0} \text{ is the helper key of the proxy signer. Let the initial secret key } d_{p,0} \text{ of the proxy signer be } \]
III. Renew the key-refresh information:

(1) \( u_i \) is the output of \( H_a(u,t) \)

(2) \( u_i[k] \) is the \( k \)-th bits of \( u_i \).

(3) \( U_{a,i} \subset \{ 1, \ldots, n_a \} \) is the set of indices \( k \) such that \( u_i[k] = 1 \) and the set of indices \( r \) such that \( u_i[k] = 1 \).

(4) Suppose the user with identity \( u \) has a temporary signing key \( d_{a,t'} \) for period \( t' \) and an update key share set \( \Omega \) for period \( t \), where \( |\Omega| \geq d \).

(5) Let the temporary secret key for period \( t \) be

\[
D_{u,t} = (d_{u,t}, d_{u,t'}, d_{u',t}, d_{u',t'})
\]

(6) Let the temporary secret key for period \( t' \) be

\[
d_{a,t'} = (d_{a,t'}, d_{a,t'}, d_{a',t'}, d_{a',t'})
\]

(7) Obtain \( d_{a,t} \) and \( d_{a,t'} \) for period \( t \) as

\[
d_{a,t} = (d_{a,t'}, d_{a,t'}, d_{a,t'}, d_{a,t'})
\]

IV. Renew the temporary private keys:

(1) Suppose the user with identity \( u \) has a temporary signing key \( d_{a,t'} \) for period \( t' \) and an update key share set \( \Omega \) for period \( t \), where \( |\Omega| \geq d \).

(2) \( D_{a,t} \) is an arbitrary \( d \)-element subset of \( \Omega \) and \( |S'| = d \).

(3) Parse the temporary secret key for period \( t' \) as

\[
d_{a,t'} = (d_{a,t'}, d_{a,t'}, d_{a,t'}, d_{a,t'})
\]

(4) Parse the temporary update key share as

\[
D_{a,t} = (d_{a,t'}, D_{a,t'}, d_{a,t'}).\]

(5) Let the temporary secret keys for period \( t \) be

\[
d_{a,t} = (d_{a,t'}, D_{a,t'}, d_{a,t'}).\]

If we let

\[
r = \sum_{j \in S} A_{a,j}(f) r_j \quad \text{and} \quad s = \sum_{j \in S} A_{a,j}(f) s \]

then the temporary secret key \( d_{a,t} \) is

\[
d_{a,t} = (g_{2}^d, g_{2}^{d_2}, (u' \prod_{k \in \lambda} u_k)^{r} (u' \prod_{k \in \lambda} u_k)^{r} g^{r}, g^{r}).\]

V. Signature delegation:

(1) \( w \) is a bit string of length \( n_w \) and the warrant of delegation

(2) \( w[j] \) is the \( j \)-th bit of \( w \).

(3) \( [j] \subset \{ 1, 2, \ldots, n_w \} \) is the set of indices \( j(w[j] = 1) \).

(4) \( w \) is used by the original signer to be a warrant \( w \) in time period \( t \).

(5) Obtain \( r, \sigma \in \mathbb{Z}_p \).

(6) Set the proxy certificate tuple \((t,w,\sigma)) \) and

\[
\sigma = (\sigma_{w1}, \sigma_{w2}, \sigma_{w3}, \sigma_{w4})
\]

\[
= (d_{a,t'}, d_{a,t'}, (u' \prod_{j \in [j]} v_j)^{r}, (u' \prod_{j \in [j]} v_j)^{r}, d_{a,t'}, g^{r}, d_{a,t'}, g^{r})
\]

\[
= (g_{2}^{d_2}, d_{a,t'}^{d_2}, (u' \prod_{j \in [j]} v_j)^{r}, (u' \prod_{j \in [j]} v_j)^{r}, d_{a,t'}, g^{r}, d_{a,t'}, g^{r})
\]
\( = (g_2^β (u' \prod_{i \in \mathcal{I}_d} u_i)^{g_{\alpha, \beta}} (u' \prod_{j \in \mathcal{V}_e} v_j)^{g_{\beta, \gamma}}) \). \\

VI. Verify the proxy certificate: If \( e(g, σ_1) = e(g_1, g_2) = e(σ_{i+1}, u' \prod_{i \in \mathcal{I}_d} u_i) = e(σ_{i+2}, u' \prod_{i \in \mathcal{I}_d} u_i) = e(σ_{i+3}, u' \prod_{i \in \mathcal{I}_d} u_i) = e(σ_{i+4}, u' \prod_{i \in \mathcal{I}_d} u_i) \), the proxy certificate tuple will be valid and the proxy signer will say “yes and continue”. Otherwise, the proxy signer will say “stop”.

VII. Renew temporary proxy secret keys: 
(1) \( r'' \in \mathbb{Z}_p \).
(2) Set the temporary proxy secret key as \( tskp_4 \),
(3) \( tskp_4 \) as \( tskp_1^{(1)}, tskp_2^{(2)}, tskp_3^{(3)}, tskp_4^{(4)}, tskp_5^{(5)}, tskp_6^{(6)} \).

\( = (g_2^β (u' \prod_{i \in \mathcal{I}_d} u_i)^{g_{\alpha, \beta}} (u' \prod_{j \in \mathcal{V}_e} v_j)^{g_{\beta, \gamma}}) \).

VIII. Generate the proxy signature: 
(1) The message of \( m \) is a bit string of length \( n_m \).
(2) \( m[k] \) is the \( k \)th bit of \( m \).
(3) \( M_n = \{1, \ldots, n_m \} \) is the set of indices \( k \) (\( m[k] = 1 \)).
(4) Set \( tskp \) as \( (tskp_1^{(1)}, tskp_2^{(2)}, tskp_3^{(3)}, tskp_4^{(4)}, tskp_5^{(5)}, tskp_6^{(6)} \).

IX. Verify the proxy signature: 
(1) If the required quality of the warrant is about the message, continue. Otherwise, stop.
(2) If the original signer in the warrant permits the proxy signer, continue. Otherwise, stop.
(3) Set \( σ_p \) as \( (σ_p, σ_{p2}, σ_{p3}, σ_{p4}, σ_{p5}, σ_{p6}, σ_{p7}) \).
(4) The verifier will say “Yes” and output true when \( e(σ_4, g) \).
\[ = e(g_1, g_2)^2 e\left(u' \prod_{i \in \mathcal{L}_p} u_i, \sigma_{P_4}\right) e\left(u' \prod_{i \in \mathcal{L}_q} u_i, \sigma_{P_5}\right) e\left(u' \prod_{i \in \mathcal{L}_r} u_i, \sigma_{P_3}\right) \]

\[ e\left(v' \prod_{j \in \mathcal{L}_m} v_j, \sigma_{P_6}\right) e\left(m' \prod_{k \in \mathcal{L}_n} m_k, \sigma_{P_7}\right). \]

Else, the verifier will say “No” and output false.

4. Security

We omit the proof of the proposed identity-based threshold key-insulated proxy signature since the proof can be gotten from the proof of the identity-based key-insulated proxy signature[1] and the threshold encryption[3].

5. Conclusion

We gave the first identity-based threshold key-insulated proxy signature scheme, a new identity-based proxy signature scheme. In our identity-based threshold key-insulated proxy signature scheme, the secret key is protected by the mechanism of threshold key insulation.

References

[1] Chen, J., Chen, K., Wang, Y. (2011) Identity-Based Key-Insulated Proxy Signature Without Random Oracles. Journal of Shanghai Jiaotong University (Science), 5: 557–566.

[2] Dodis, Y., Katz J., Xu, S., Yung, M. (2002) Key-Insulated Public-Key Cryptosystem. In: Knudsen, R. (Eds.), Eurocrypt 2002. Springer, Heidelberg. pp. 65-82.

[3] Dodis, Y., Katz J., Xu, S., Yung, M. (2003) Strong Key-Insulated Signature Schemes. In: Desmedt, Y. (Eds.), PKC2003. Springer, Heidelberg. pp. 130-144.

[4] Manbo, M., Usuda, K., Okamoto, E. (1996) Proxy Signature: Delegation of the power to sign message. IEICE Transactions on Fundamentals of Electronic Communications and Computer Science, E79-A (9): 1338-1354.

[5] Wan, Z., Lai X., Weng, J. (2009) Identity-Based Key-Insulated Proxy Signature. Journal of Electronic, 26(6): 853-858.

[6] Weng, J., Liu S., Chen, K., Li, X. (2006) Identity-Based Key-Insulated Signature with Secure Key-Updates. In: Lipmaa H., Yung M., Lin D. (Eds.), Inscrypt2006. Springer, Heidelberg. pp. 13-26.

[7] Weng, J., Liu S., Chen, K., Zheng, D., Qiu, W. (2008) Identity-Based Threshold Key-Insulated Encryption Without Random Oracles. In: Malkin T. (Eds.), CT-RSA2008. Springer, Heidelberg. pp. 409-423.

[8] Zhou, Y., Cao Z., Chai, Z. (2006) Identity Based Key Insulated Signature. In: Chen T., Deng R., Lai X., Zhou J. (Eds.), ISPEC2006. Springer, Heidelberg. pp. 226-234.