Attribute-Based Threshold Key-Insulated Signature

Jianhong Chen, Wenhao Wang, Kun Yu
Faculty of Computer Engineering, Huaiyin Institute of Technology, Huaian 223003, Jiangsu, China

Abstract
Key insulation is an important technique to protect private keys. To deal with the signing key exposure problem in attribute-based signature systems, we propose an attribute-based threshold key-insulated signature (ABTKIS) scheme. It strengthens the security and flexibility of existing attribute-based key-insulated signature schemes. Our scheme is provably secure in the standard model (i.e. without random oracles).

Key words: Threshold key-insulated, Attribute-based, Key exposure, Signature.

1. INTRODUCTION
To protect private keys, Dodis et al. proposed the idea of key insulation in EUROCRYPT 2002 (Dodis,Katz,Xuand Yung, 2002). The first key-insulated signature scheme was formalized by Dodis et al. in PKC 2003 (Dodis,Katz,Xuand Yung, 2003). Attribute-Based Signature (ABS) can manifest apretension to the attributes that the underlying signer owns (Li and Kim, 2010; Li, Au, Susilo, Xie and Ren, 2010; Shahandashti and Safavi-Naini, 2009). ABS can be applied to attribute based messaging systems and anonymous authentication. The first attribute-based key-insulated signature scheme (ABKIS) was proposed by Chen et al. (Chen, Long, and Guo, 2014). After that, attribute-based parallel key-insulated signature (ABPKIS) schemes were given for some special situations (Chen, Yu, Long and Chen, 2015).

To enhance the security and flexibility of the key insulation mechanism, Weng et al. (Weng, Li, Chen and Liu, 2008) put forward the threshold key insulation mechanism in CT-RSA 2008. In this paper, we extend the threshold key-insulated mechanism to the complementary case of digital signatures and put forward an attribute-based threshold key-insulated signature (ABTKIS) scheme. In a \((k, n)\) threshold key-insulated ABTKIS system, long-term keys (called helper keys) are kept in physically-secure but computationally-limited devices called helpers. On the other hand, users store short-term signing keys (called temporary signing keys) in a powerful but insecure device where cryptographic computations happen. At least \(k\) out of \(n\) helpers are needed to update the user’s temporary signing keys. The ABTKIS cryptosystem refreshes the temporary private keys at discrete time periods via interaction between the user and the helper, and the public key (attribute set) remains unchanged throughout the lifetime of the system. Our proposed ABTKIS scheme is key-insulated, and even if \(k-1\) helper keys and some of his temporary signing keys are compromised, an adversary is still unable to derive this user’s temporary signing keys in other time periods. It is strongly key-insulated, and even if \(n\) helper keys and none of his temporary signing keys are exposed, it is still impossible for an adversary to obtain all of this user’s temporary signing keys.

2. PRELIMINARIES
Throughout this paper, we let \(Z_p\) denote the set \(\{0,1,2,\ldots,p-1\}\) and \(Z'\) denote \(Z_p\setminus\{0\}\). For a finite set \(S\), \(x \leftarrow S\) means choosing an element \(x\) from \(S\) with a uniform distribution.

2.1. Bilinear Pairings
Our ABTKIS scheme uses a bilinear map (pairing), \(\hat{e}: G_1 \times G_1 \rightarrow G_2\), where \(G_1\) is a multiplicative group with prime order \(p\) and \(G_2\) is also a multiplicative group with prime order \(p\). The pairings satisfies the following conditions:

- **Bilinear**: For all \(g_1, g_2 \in G_1\) and for all \(a, b \in Z_p^*\), we have \(\hat{e}(g_1^a, g_2^b) = \hat{e}(g_1, g_2)^{ab}\).
- **Non-degenerate**: There exists \(g_1, g_2 \in G_1\) such that \(\hat{e}(g_1, g_2) \neq 1\).
- **Computable**: There is an efficient algorithm to compute \(\hat{e}(g_1, g_2)\) for all \(g_1, g_2 \in G_1\).

2.2. Computational Diffie-Hellman Assumptions

**Definition 1.** The CDH problem in \(G_1\) is, given \(g, g^a, g^b \in G_1\), to compute \(g^{ab}\) (for unknown randomly chosen \(a, b \in Z_p^*\)).
Definition 2. We say that the \((t, \varepsilon)-\text{CDH}\) assumption holds in a group \(G_1\) if no algorithm running in time at most \(t\) can solve the CDH problem in \(G_1\) with probability at least \(\varepsilon\).

3. DEFINITION OF ABTKIS

- **Setup**: Given a security parameter \(k\), the length \(l\) of some universe \(U\), of size \(|U|\) and a threshold value \(d\), the PKG runs this algorithm to output a master key \(msk\) and public parameters \(cp\).
- **KeyGen**: Given the user’s identity \(\omega \in U\), as a set representing a user’s attributes, public parameters \(cp\) and the master key \(msk\), the PKG runs this algorithm to output an initial private key \(TK_{\omega,0}\) and \(n\) helper keys \(HK_{\omega,i}(1 \leq i \leq n)\) that corresponds to \(\omega\).
- **HelperUpt**: Given the \(T\)-th helper key \(HK_{\omega,T}\) for identity \(\omega\) and the period index \(t\), the \(i\)-th helper runs this algorithm and outputs the \(T\)-th update key share \(UI_{\omega,T,i}\) for identity \(\omega\) and period \(t\).
- **UserUpt**: Given an identity \(\omega\), user \(\omega\)’s temporary signing key \(TK_{\omega,t'}\) for period \(t'\), and an update key share set \(\{UI_{\omega,T,i}\}_{i=1}^{T}\) of for identity \(\omega\) and period \(t\), where \(\omega \in \{1, \ldots, n\}\) and \(|\omega| \geq d\). User \(\omega\) runs this algorithm and outputs his temporary signing key \(TK_{\omega,t'}\) for period \(t\).
- **Sign**: Given public parameters \(cp\), a period index \(t\), a message \(m\) and a temporary private key \(TK_{\omega,t}\), this algorithm outputs a signature \((t, \sigma)\) with regard to the period index \(t\) and the attribute set \(\omega' \in \omega\) and message \(m\).
- **Verify**: Given public parameters \(cp\), a message \(m\), an attribute set \(\omega'\) and a signature \((t, \sigma)\) with regard to period \(t\), an attribute set \(\omega'\) and message \(m\), this algorithm outputs 1 if \((t, \sigma)\) is a valid signature and 0 otherwise.

4. SECURITY NOTIONS FOR ABTKIS

For convenience, we give the definition of a restricted identity as below: a restricted identity \(\omega^\beta\) satisfies \(\omega \subseteq \omega^\beta\), where \(\omega\) is the challenge identity.

4.1. Key-insulated Security

The key-insulated security notion captures the intuition that, if an adversary does not compromise the helper key for a given identity (i.e., an attribute set), then exposure of any of the private keys does not enable an adversary to forge a valid signature for the non-exposed time periods.

Formally, for an ABTKIS scheme, its key-insulated security can be defined via the following game of existential unforgeability against a chosen identity and an adaptive chosen message attack under key-exposure (UF-ID&KE-CMA) between an adversary \(A\) and a challenger \(X\):  
- **Init.** The adversary declares the identity \(\alpha\), where \(|\alpha| < d\) and \(d\) is the threshold and the period index \(t^*\) that he wishes to be challenged upon.
- **Setup.** The challenger \(X\) runs the algorithm \(Setup\) and tells the adversary \(A\) the public parameters.
- **Query Phase.** The adversary \(A\) adaptively issues a set of queries as below:
  - **Key Generation queries** \(\alpha\): \(X\) first runs the algorithm \(KeyGen\) to obtain the initial private key \(TK_{\omega,0}\) and the helper key \(HK_{\alpha,i}\) that corresponds to the identity \(\alpha\). It then sends these results to the adversary.
  - **Helper Key queries** \(\alpha, T\): \(X\) runs algorithm \(KeyGen\) to generate \(HK_{\omega,T}\) and sends it to the adversary.
  - **Temporary Private Key queries** \(\alpha, t\): \(X\) runs the algorithm \(UserUpt\) to obtain the temporary private key for the identity \(\omega\) and period index \(t\). It then sends the result to \(A\).
  - **Signing queries** \(\alpha, t, m\): \(A\) runs the algorithm \(Sign\(\((cp, t, m, TK_{\omega,t})\)\) to generate a signature \((t, \sigma)\). Then, \(A\) returns \((t, \sigma)\) to \(A\).
- **Output.** Finally, \(A\) outputs an identity \(\alpha\), a period index \(t^*\) and a corresponding signature \((t^*, \sigma')\).

In the above game, it is also mandated that the following conditions are simultaneously satisfied:  
1. **Verify** \(cp, (t^*, \sigma'), m, \omega\) = 1;  
2. \(A\) is disallowed to issue key generation queries for any restricted identity;  
3. \(A\) is disallowed to issue temporary private key queries for any restricted identity and the challenged time period \(t^*\);  
4. \(A\) is disallowed to issue signing queries for any restricted identity, the challenged time period \(t^*\) and message \(m\).

4.2. Strongly Key-insulated Security

The strong key-insulated security for ABTKIS systems says that, even if one of a user’s helper key and some of
his temporary signing keys are exposed, it is still impossible for an adversary to obtain all of this user’s temporary signing keys.

Formally, for an ABTKIS scheme, its strongly key-insulated security can be defined via the following strongly-UF-ID&KE-CMA game between an adversary A and a challenger X:

- **Init.** The same as a UF-ID&KE-CMA game.
- **Setup.** The same as a UF-ID&KE-CMA game.
- **Query Phase.** The adversary A adaptively issues a set of queries such as those given below:
  - Key Generation queries($\alpha_{i}$): The same as a UF-ID&KE-CMA game.
  - Helper Key queries($\alpha_{i}T$): X runs algorithm KeyGen to generate $HK_{\alpha_{i}}$ and sends it to the adversary.
  - Signing queries($\alpha_{i}, t, m$): The same as a UF-ID&KE-CMA game.
- **Output.** Finally, X outputs an identity $\alpha$, a period index $i^{'},$ and a corresponding signature $(i^{'}, \sigma^{'})$.

In the above game, it is also mandated that the following conditions are simultaneously satisfied:

1. $\text{Verify}(cp,$ $(i^{'}, \sigma^{'})$, $m^{'}, \alpha) = 1$;
2. A is disallowed to issue key generation queries for any restricted identity;
3. A is disallowed to issue signing queries for any restricted identity, the challenged time period $t$ and message $m$.

### 4.3. Anonymity

The An ABTKIS scheme satisfies the anonymity requirement if no adversary A can win the following ANONY-ABTKIS game between $\mathcal{A}$ and a challenger X with a non-negligible advantage:

- X runs the algorithm Setup to generate a master key $msk$ and public parameters $cp$ and sends them to A.
- A can use the master key $msk$ to generate temporary private keys and signatures.
- A will next submit a challenge period index $t$, a message $m^{'}$ two identities $(\alpha_{1}, \alpha_{2})$ and a challenge identity $\alpha$, where $\alpha \subseteq (\alpha_{1} \cap \alpha_{2})$ and $|\alpha| \leq d$.
- Assume that $\mathcal{A}$ has issued temporary private key queries $(\alpha_{1}, t^{'})$ and $(\alpha_{2}, t^{'})$. Let $TK_{\alpha_{1}, t^{'}}$ and $TK_{\alpha_{2}, t^{'}}$ be temporary private keys for $(\alpha_{1}, t^{'})$ and $(\alpha_{2}, t^{'})$, respectively.
- X flips a random coin, $b$, computes a signature $(i^{'}, \sigma^{'}) =$ Sign($cp,$ $i^{'}, m^{'}, TK_{\alpha_{b}, t^{'}}$) and sends it to A.
- Finally, $\mathcal{A}$ outputs a guess $b'$ of $b$ by judging whether $(i^{'}, \sigma^{'})$ is generated from $TK_{\alpha_{1}, t^{'}}$ or $TK_{\alpha_{2}, t^{'}}$.

### 5. OURPROPOSED ABTKIS

#### 5.1. Description of our scheme

We define the Lagrange coefficient

$$\Lambda_{s}(x)= \prod_{j \in \mathbb{Z}_{n}} \frac{x-s_{j}}{j-s_{j}}$$

for $i \in \mathbb{Z}_{n}$ and a set $S$ of elements in $\mathbb{Z}_{n}$. The proposed ABTKIS scheme consists of the following algorithms:

- **Setup($\kappa, l, d$):** Given a security parameter $\kappa$, the length $l$ of some universe $U$, of size $|U|$, and a threshold value $d$, $\text{PKG}$ works as follows:
  1. Define the universe $U$. For simplicity, we can take the first $l$ elements of $Z_{n}$ to be the universe, specifically, the integers $l, \ldots, l$ (mod $p$); Choose the total number $n$ of each user’s helpers and a threshold value $d$.
  2. Let $G_{1}$ and $G_{2}$ be two groups with prime order $p$ of size $\kappa$, let $g$ be a generator of $G_{1}$, let $\hat{e}: G_{1} \times G_{1} \rightarrow G_{2}$ denote the bilinear map.
  3. Let $H_{1}: \{0, 1\}^{l} \rightarrow \{0, 1\}^{n_{w}}, H_{2}: \{0, 1\}^{l} \rightarrow \{0, 1\}^{n_{w}}$ be two collision-resistant hash functions with $n_{w}, n_{m} \in \mathbb{Z}_{n}$.
  4. Pick $y \leftarrow \mathbb{Z}_{p}$ and $g_{1} \leftarrow G_{1}$, set $g^{y}$.
  5. Choose a default set of ($d$-1) attributes from $Z_{n}$, $\Omega = \{l+1, l+2, \ldots, l+d-1\}$.
  6. Pick a random $(l+d-1)$-length vector $\hat{H} = (h_{i})$ whose elements are randomly chosen from $G_{1}$.
  7. Pick $w \leftarrow G_{1}$ and a random $n_{w}$-length vector $\hat{W} = (w_{i})$ whose elements are randomly chosen from $G_{1}$.
  8. Pick $m \leftarrow G_{1}$ and a random $n_{w}$-length vector $\hat{M} = (m_{i})$ whose elements are randomly chosen from $G_{1}$.

Output the public parameters and the master secret key $ascp = (G_{1}, G_{2}, \hat{e}, g, g_{1}, g_{2}, w, m, \hat{H}, \hat{W}, \hat{M}, \Omega, m_{i})$, $msk = y$. For convenience, we define two functions $L_{1}$ and $L_{2}$ such that $L_{1}(S)=w \prod_{i \in S} w_{i}, L_{2}(S)=m \prod_{i \in S} m_{i}$. In addition, for a given time period $t$ and a given message $m$, we hereafter use $W_{t}$ and $M_{m}$ to denote the following sets:

$$W_{t} = \{i \mid a[i] = 1, a = H_{t}(t) \subseteq \{1, \ldots, n_{w}\}\}, M_{m} = \{j \mid b[j] = 1, b = H_{t}(m) \subseteq \{1, \ldots, n_{w}\}\}.$$
• KeyGen(msk, cp, ω): To generate the helper key and the initial private key for an identity ω∈Y, the PKG works as follows:

1. A d-1 degree polynomial q is randomly chosen such that q(0) = 1;
2. Generate a new attribute set $\hat{\omega} = \omega \cup \Omega$;
3. For each $i \in \hat{\omega}$, pick $\beta_i \leftarrow Z_p$ and set the initial signing key for identity ω as

$$TK_{i\omega} = \{(d_{i\omega}, \ast, \ast, \ast)\}_{i\omega} = \{(g_2^{d_{i\omega}}, \ast, \ast, \ast)\}_{i\omega}$$

For each $i \in \hat{\omega}$ and for each $\overline{t} \in \{1, \ldots, \overline{t}-1\}$, pick $c_{i\overline{t}}, r_{i\overline{t}} \leftarrow Z_p$, and sets the $\overline{t}$-th helper key as

$$HK_{i\overline{t}} = \{(HK_{i\overline{t}}, HK_{i\overline{t}}')\}_{i\overline{t}} = \{(g_2^\beta (g_i h_i)^{c_{i\overline{t}}}, g_2^{r_{i\overline{t}}})\}_{i\overline{t}}$$

Let $S' = \{0, 1, \ldots, \overline{t} - 1\}$. For each $i \in \hat{\omega}$, pick $r_i \leftarrow Z_p$. For each remaining index $\overline{t} \in \{\overline{t} \ldots, ||\},$ set the $\overline{t}$-th helper key $HK_{\overline{t}}$ to be

$$HK_{\overline{t}} = \{(HK_{\overline{t}}, HK_{\overline{t}}')\}_{i\overline{t}} = \{(g_2^{q(i) - \beta} (g_i h_i)^{c_{\overline{t}}})^{\overline{t}}, (\prod_{j=1}^{\overline{t}-1} HK_{ij})^{\overline{t}}, (g_2^r)^{\overline{t}} (\prod_{j=1}^{\overline{t}-1} HK_{ij})^{\overline{t}}\}_{i\overline{t}}$$

Here we claim that the helper keys in Eq. (1) have the same form as those in Eq. (1). To see this, for each $i \in \hat{\omega}$, let $f_{i1} (x)$ denote the $(\overline{t} - 1)$-degree polynomial such that $f_{i1} (0) = q(i) - \beta$, and $f_{i1} (\overline{t}) = c_{i\overline{t}}$ for each $\overline{t} \in \{1, \ldots, \overline{t} - 1\}$. Also let $f_{i2} (\overline{t})$ denote the $(\overline{t}-1)$-degree polynomial such that $f_{i2} (0) = r_i$, and $f_{i2} (\overline{t}) = r_{i\overline{t}}$ for each $\overline{t} \in \{1, \ldots, \overline{t} - 1\}$. Besides, for each $\overline{t} \in \{\overline{t}, \ldots, ||\}$, we set $f_{i1} (\overline{t})$ and $f_{i2} (\overline{t})$ denote $c_{i\overline{t}}$ and $r_{i\overline{t}}$ respectively.

Then, for each $\overline{t} \in \{\overline{t}, \ldots, ||\}$, we have

$$= (g_2^{q(i) - \beta} (g_i h_i)^{c_{\overline{t}}})^{\overline{t}} x(j) (\prod_{j=1}^{\overline{t}-1} HK_{ij})^{\overline{t}} x(j)$$

$$= (g_2^{q(i) - \beta} (g_i h_i)^{c_{\overline{t}}})^{\overline{t}} x(j) (\prod_{j=1}^{\overline{t}-1} g_2^{r_{\overline{t}}} (g_i h_i)^{c_{\overline{t}}})^{\overline{t}} x(j)$$

$$= g_2^{q(i) - \beta} (g_i h_i)^{c_{\overline{t}}} (\prod_{j=1}^{\overline{t}-1} g_2^{r_{\overline{t}}} (g_i h_i)^{c_{\overline{t}}}) x(j)$$

$$= g_2^{q(i) - \beta} (\sum_{j=1}^{\overline{t}-1} g_2^{r_{\overline{t}}} (g_i h_i)^{c_{\overline{t}}}) x(j)$$

$$= g_2^{q(i) - \beta} (\sum_{j=1}^{\overline{t}-1} g_2^{r_{\overline{t}}} (g_i h_i)^{c_{\overline{t}}}) x(j)$$

Similarly, for each $\overline{t} \in \{\overline{t}, \ldots, ||\}$, we have $HK_{\overline{t}} = g_2^{r_{\overline{t}}}$. 

* HelperUpt(cp, t, ω, HKω) : Given the $\overline{t}$-th helper key $HK_{\overline{t}}$ for identity ω and the period index t, for
each $i \in \omega$, this algorithm picks $\vec{s} \in Z_\rho$ and outputs user $\omega$'s $T$-th update key share for period $t$ as

$$U_{\omega,T} = \{(U_{1,T}, U_{2,T}, U_{3,T})\}_{i=0}^{\rho}$$

$$= \{(HK_{\omega,T}, V_{\omega,T}) \}_{i=0}^{\rho}$$

$$= \{(g^{\omega_{1}}(g_{i}^{\omega}), y_{i}^{\omega}, L_{i}(\rho)^{\omega}, g^{\omega}, g^{\omega'})\}_{i=0}^{\rho}$$

- **UserUpt($cp$, $t$, $\omega$, $\{U_{\omega,T}\}_{T=0}^{\rho}$, $TK_{\omega}$):** Given an identity $\omega \in Y$, a temporary private key for the identity $\omega$ and period $t$, and a set $\{U_{\omega,T}\}_{T=0}^{\rho}$ of key-update information shares for period $t$, where $\hat{S} = \{1, \ldots, \pi\}$ and $|\hat{S}| > T$ (for convenience, we assume $|\hat{S}| = T$), this algorithm works as follows:

1. Parse $TK_{\omega}$ as $\{(d_{1,T}, d_{2,T}, d_{3,T}, d_{4,T})\}_{i=0}^{\pi}$
2. Parse $U_{\omega,T}$ as $\{(U_{1,T}, U_{2,T}, U_{3,T})\}_{i=0}^{\pi}$
3. Set the temporary private key for the identity $\omega$ and period $t$

$$TK_{\omega} = \{(d_{1,T}, d_{2,T}, d_{3,T}, d_{4,T})\}_{i=0}^{\pi}$$

4. Delete $TK_{\omega}$ and $U_{\omega,T}$
5. Return $TK_{\omega}$

Note that if let $t = \sum_{j=0}^{T} t_j \hat{a}_{j}(j)$ and $\bar{X} = \sum_{j=0}^{T} t_j \hat{a}_{j}(j)$, then $TK_{\omega}$ is always set to be

$$TK_{\omega} = \{(g_{2}^{\omega}, g_{2}^{\hat{o}_{t}(\hat{a})}(g_{1}^{\omega}), y_{i}^{\omega}, L_{1}(\rho)^{\omega}, g^{\omega}, g^{\omega'})\}_{i=0}^{\pi}$$

This can be seen from the following:

$$= \{(g_{2}^{\hat{o}_{t}(\hat{a})}(g_{1}^{\omega}), y_{i}^{\omega}, L_{1}(\rho)^{\omega}, g^{\omega}, g^{\omega'})\}_{i=0}^{\pi}$$

$$= \{(g_{2}^{\hat{o}_{t}(\hat{a})}(g_{1}^{\omega}), y_{i}^{\omega}, L_{1}(\rho)^{\omega}, g^{\omega}, g^{\omega'})\}_{i=0}^{\pi}$$

$$= \{(g_{2}^{\hat{o}_{t}(\hat{a})}(g_{1}^{\omega}), y_{i}^{\omega}, L_{1}(\rho)^{\omega}, g^{\omega}, g^{\omega'})\}_{i=0}^{\pi}$$

$$= \{(g_{2}^{\hat{o}_{t}(\hat{a})}(g_{1}^{\omega}), y_{i}^{\omega}, L_{1}(\rho)^{\omega}, g^{\omega}, g^{\omega'})\}_{i=0}^{\pi}$$

- **Sign($cp$, $t$, $m$, $TK_{\omega}$):** Suppose that the signer has a temporary private key for the attribute set $\omega$ and time period $t$. In period $t$, the signer can produce the signature on $m$ with the attribute set $\omega = \{i_1, i_2, \ldots, i_k\} \subseteq \omega$, where
1≤k≤d, as follows:

1. Parse TK_{si} as TK_{si} = \{(d_{1,p}, d_{2,p}, d_{3,si}, d_{4,si})\}_{i=1}^{n};

2. Select a d- k default attribute subset \( \mathcal{I} = \{ i_{k+1}, i_{k+2}, \ldots, i_{d}\}\subseteq \Omega \). Then, pick \( r_i, r'_i, \ldots, r'_i, s_1, s_2, \ldots, s_d, \)
\( c_{1,i}, c_{2,i}, \ldots, c_{d,i}, Z_i \), choose a degree polynomial \( q(x) \) such that \( q(0) = 0; \)

3. For each \( v \in \{1, \ldots, d\} \), compute \( \sigma_{v1} = d_{1,v}, d_{2,v}, g_{2,v}^{(i)}(g, h) \), \( L_i(W_i)^{r_i}, L_2(M_m)^{s_i}, \sigma_{v2} = d_{2,v}, g^{r_i}, \sigma_{v3} = g^{r_i}, \sigma_{v4} = d_{1,v}, g^{r_i}; \)

4. Output the signature \( (t, \sigma) = (t, \{\sigma_{v1}, \sigma_{v2}, \sigma_{v3}, \sigma_{v4}\}_{i=1}^{n}) \). Note that if we let \( \bar{r}_i = r_i + r'_i, \bar{r}'_i = r'_i + e_i \), the signature \( (t, \sigma) \) is always set to be \((t, \{g_2^{r_i}(g, h)^{\bar{r}_i}, L_i(W_i)^{\bar{r}_i}, L_2(M_m)^{r_i}, g^{\bar{r}_i}, g^{\bar{r}'_i}\})_{i=1}^{n}\)

This result can be seen from the following:

\[ \sigma_{v1}, \sigma_{v2}, \sigma_{v3}, \sigma_{v4} \]

\[ = (d_{1,v}, d_{2,v}, g_{2,v}^{(i)}(g, h)^{\bar{r}_i}, L_i(W_i)^{\bar{r}_i}, L_2(M_m)^{r_i}, g^{\bar{r}_i}, g^{\bar{r}'_i}, d_{1,v}, g^{r_i}) \]

\[ = (g_2^{r_i}(g, h)^{\bar{r}_i}, L_i(W_i)^{\bar{r}_i}, L_2(M_m)^{r_i}, g^{\bar{r}_i}, g^{\bar{r}'_i}, g^{r_i}) \]

\[ = (g_2^{r_i}(g, h)^{\bar{r}_i}, L_i(W_i)^{\bar{r}_i}, L_2(M_m)^{r_i}, g^{\bar{r}_i}, g^{\bar{r}'_i}, g^{r_i}) \]

* Verify (cp, m, \omega, (t, \sigma)): Let \( S = \{ i_1, \ldots, i_d \} \). Given the signature \( (t, \sigma) \) for period \( t \), attribute set \( \omega = \{i_1, \ldots, i_k\} \) and a message \( m \) with the default attribute set \( \Omega = \{i_{k+1}, i_{k+2}, \ldots, i_d\} \), the verifier accepts \( (t, \sigma) \) if the following equality holds:

\[ \prod_{i=1}^{n} (e(g_{2,i}^{r_i}(g, h)^{\bar{r}_i}, L_i(W_i)^{\bar{r}_i}, L_2(M_m)^{r_i}, g^{\bar{r}_i}, g^{\bar{r}'_i}, g^{r_i}))^{\lambda_i, (0)} = Z \]

The consistency of this scheme can be explained as follows:

\[ \prod_{i=1}^{n} (e(g_{2,i}^{r_i}(g, h)^{\bar{r}_i}, L_i(W_i)^{\bar{r}_i}, L_2(M_m)^{r_i}, g^{\bar{r}_i}, g^{\bar{r}'_i}, g^{r_i}))^{\lambda_i, (0)} = e(g, g_j)^{\lambda_i, (0)} = e(g, g_j)^{\lambda_i, (0)} = e(g, g_j)^{\lambda_i, (0)} = e(g, g_j)^{\lambda_i, (0)} = Z \]

5.2. Security

**Theorem 1.** Proposed ABTISK is key-insulated in the Selective-ID model, assuming that the CDH assumption holds in group \( G_1 \) and the hash function \( H \) is collision-resistant. Concretely, if there exists a \((T, \varepsilon)-\)
UF-ID&KE-CMA adversary \( A \) against our scheme, asking at most \( q_0(q_p, q_m, \text{ respectively}) \) queries to the oracle of key generation queries (temporary private key queries, signature queries, respectively), then there exists an efficient algorithm \( \hat{A} \) that can solve the \((T, \varepsilon)-\)CDH assumption in group \( G_1 \) with \( T \leq T^* \) \( O((q_0 + q_p + q_m)\mu_{t_x} + (n_0(q_0 + q_p) + (n_m + n_m)(q_0 + q_m)) \mu_{t_y}), \varepsilon \geq \frac{\varepsilon}{64(n_0 + n_0 + 1)(q_0 + q_0)^{\frac{d_{k-1}}{d_{k-1}}} \mu_{t_x} \mu_{t_y}} \); where \( t_x \) and \( t_y \) denote the running time of an exponentiation and a multiplication in group \( G_{i_1} \), respectively, where \( d \) is the threshold value, and \( k \) is the length of the challenge identity.
Proof. The proof is similar with that of ref [1] except the help key queries.

Help key queries: \( \square \) maintains a list \( HK_{\text{list}} \) which is initially empty. On receiving a helper key query \( \langle \omega, i \rangle \), algorithm B first checks whether \( HK_{\text{list}} \) has contained a tuple for this input. If yes, the predefined value is returned to A. Otherwise, for each \( i \in \omega \) and for each \( T \in \{1, \ldots, T-1\} \), \( \square \) picks \( r_{T} \leftarrow Z_{p} \), and sets the \( T \)-th helper key as

\[
HK_{\omega, r} = \{(HK_{\omega, r}, HK_{\omega, r})\}_{i \in \omega} = \{(g_{1}^{r}, (g_{2} h_{i})^{r}, g_{3}^{r})\}_{i \in \omega}
\]

Next, it adds tuple \( \langle \omega, T, HK_{\omega, r} \rangle \) into list \( HK_{\text{list}} \), and returns \( HK_{\omega, r} \) to A. Note that in adversary A.'s view, the above helper key is valid, since A can only corrupt up to \( T \)-helper keys for the challenge identity \( \omega \) and helper key queries for the non-challenge identities are implied by the extraction queries.

Theorem 2. The proposed ABKTIS scheme is key-insulated in the Selective-ID model, assuming that the CDH assumption holds in group \( G_{1} \) and the hash function \( H \) is collision-resistant. Concretely, if there exists a \( (T, \varepsilon) \)-strongly-UF-IDKE-\text{CMA} adversary \( \mathcal{A} \) against our scheme, asking at most \( q_{t} (q_{b}, q_{g}, \text{respectively}) \) queries to the oracle of key generation queries (helper key queries, signature queries respectively), then there exists an efficient algorithm \( \mathcal{B} \) that can solve the \( (T, \varepsilon) \)-CDH assumption in group \( G_{1} \) with \( T \leq T^{*} \).

\[ \mathcal{B} = \frac{2 \varepsilon}{16 (n + 1) t q_{t} q_{b} (d + 1) d} \text{ for } t, n, d, k \text{ denote the same quantities as in Theorem 1.} \]

Proof. The proof is the same as that of theorem 1 except that: temporary private key queries are no longer provided to A.

Theorem 3. The proposed ABKTIS scheme satisfies anonymity.

Proof. First, the challenger \( \mathcal{C} \) runs the algorithm Setup to obtain the public parameters \( cp \) and the master secret key \( msk = y \). \( \mathcal{C} \) also gives \( cp \) and \( msk = y \) to the adversary A. After these interactions, the adversary outputs two identities \( \omega \) and \( \omega' \), where \( \omega' = \omega \cap \omega' \). Note that the temporary private key for each user should include the \( (d-1) \)-element default attributes set \( T \). Let \( \omega_{Q} = \omega \cup \omega' \). Assume that X or A has generated the temporary private key \( TK_{\omega, i} = \{(d_{1}, d_{2}, d_{3}, d_{4})\}_{i \in \omega} \) for \( \omega \) and \( TK_{\omega', i} = \{(d_{1}, d_{2}, d_{3}, d_{4})\}_{i \in \omega'} \) for \( \omega' \). For each \( i \in \omega_{Q} \), let

\[
(d_{1}^{b}, d_{2}^{b}, d_{3}^{b}, d_{4}^{b}) = (g_{1}^{b}, g_{2}^{b}, L_{i}(W_{i})^{b}, g_{3}^{b}, g_{4}^{b})
\]

where \( b \in \{0, 1\} \), \( \beta = \beta' \), \( \gamma = \gamma' \), \( \gamma e Z_{p} \) is a \( (d-1) \)-degree polynomial such that \( q_{b}(0) = y \).

Then, A outputs the period index \( i' \), message \( m' \) and a \( k \)-element subset \( \omega' = \{1, \ldots, k\} \subseteq \omega' \), where \( |\omega'| \leq d \). A asks X to generate a signature on message \( m' \) with respect to \( \omega' \) and \( i' \) from either \( TK_{\omega', i} \) or \( TK_{\omega', i} \). X picks \( b' \in \{0, 1\} \) and a \( (d-k) \)-element subset \( \Omega = \{i_{k+1}, \ldots, i_{d}\} \subseteq \Omega \). Then, X runs algorithm \( \text{Sign}(cp, i', m', TK_{\omega, i}) \) to output a signature

\[
(i', \{(d_{1}^{b}, d_{2}^{b}, d_{3}^{b}, d_{4}^{b}), g_{2}^{(b')^{Y}}(g_{2} h_{i})^{Y} L_{i}(W_{i})^{b'}, L_{2}(M_{a})^{b'}, L_{3}(M_{a})^{b'}, L_{4}(M_{a})^{b'}\})_{i \in \omega_{e} \cap \omega'},
\]

where \( \text{TK}_{\omega, i} = \{(d_{1}, d_{2}, d_{3}, d_{4})\}_{i \in \omega} \) and \( r_{i}^{*}, s_{b}, c_{i}, c_{i'} e Z_{p} \) is a \( d-1 \)-degree polynomial function with \( q' \) \( (0) = 0 \). The signature \( (i', \sigma') \) could be generated from either \( TK_{\omega', i} \) or \( TK_{\omega', i} \). If \( b = 1 \), the signature \( (i', \sigma') \) from \( \text{TK}_{\omega', i} \) is

\[
(i', \{(d_{1}^{b}, d_{2}^{b}, d_{3}^{b}, d_{4}^{b}), g_{2}^{(b')^{Y}}(g_{2} h_{i})^{Y} L_{i}(W_{i})^{b'}, L_{2}(M_{a})^{b'}, L_{3}(M_{a})^{b'}, L_{4}(M_{a})^{b'}\}).
\]
\[ d_{i,j}^l g^\cdot, g^\cdot, d_{i,j}^l g^\cdot \}_{i \in \alpha, \beta, \gamma}, \]

We prove that this signature could be generated from TK \( \alpha, \beta, \gamma \) as follows:

(1) From the construction of a temporary private key, we have

\[ \frac{d_{i,j}^l}{d_{i,j}^l} \frac{g^\cdot}{g^\cdot} = g^\cdot, \]

\[ \frac{d_{i,j}^l}{d_{i,j}^l} \frac{g^\cdot}{g^\cdot} = g^\cdot, \]

\[ \frac{d_{i,j}^l}{d_{i,j}^l} \frac{g^\cdot}{g^\cdot} = g^\cdot. \]

Then, the signature

\[ \frac{d_{i,j}^l}{d_{i,j}^l} \frac{g^\cdot}{g^\cdot} = g^\cdot, \]

(2) A \( d-1 \) degree polynomial \( \bar{q} \) is randomly chosen such that \( \bar{q} (0) = 0. \)

(3) Then, we have

\[ (d_{i,j}^l, d_{i,j}^l) g^\cdot, (g, h)^\cdot L_1(W_i)^\cdot L_2(M_i)^\cdot L_2(M_i)^\cdot, d_{i,j}^l g^\cdot, g^\cdot, d_{i,j}^l g^\cdot) \]

\[ = (d_{i,j}^l, d_{i,j}^l) g^\cdot, (g, h)^\cdot L_1(W_i)^\cdot L_2(M_i)^\cdot L_2(M_i)^\cdot, d_{i,j}^l g^\cdot, g^\cdot, d_{i,j}^l g^\cdot) \]

\[ = (d_{i,j}^l, d_{i,j}^l) g^\cdot, (g, h)^\cdot L_1(W_i)^\cdot L_2(M_i)^\cdot L_2(M_i)^\cdot, d_{i,j}^l g^\cdot, g^\cdot, d_{i,j}^l g^\cdot) \]

\[ = (d_{i,j}^l, d_{i,j}^l) g^\cdot, (g, h)^\cdot L_1(W_i)^\cdot L_2(M_i)^\cdot L_2(M_i)^\cdot, d_{i,j}^l g^\cdot, g^\cdot, d_{i,j}^l g^\cdot) \]

(4) A \( d - 1 \) degree polynomial \( \bar{q} \) is randomly chosen such that \( \bar{q} (x) = q_1(x) - q_2(x) + q_3(x) \). Then, we have \( q_3 (0) = 0, q_{(i)} = q_1(i) - q_2(i) + q_3(i) \). Let \( r'' = r'' - r'' + r'' \), \( c'' = c'' + r'' \), \( s'' = s'' + r'' \). Then, the signature \( (r'' , c'' ) \) could be rewritten as

\[ (r'' , c'' , d_{i,j}^l g^\cdot, (g, h)^\cdot L_1(W_i)^\cdot L_2(M_i)^\cdot L_2(M_i)^\cdot, d_{i,j}^l g^\cdot, g^\cdot, d_{i,j}^l g^\cdot) \]

which is a valid signature generated from TK \( \alpha, \beta, \gamma \).

Similarly, a signature \( (r'' , c'' ) \) from TK \( \alpha, \beta, \gamma \) can also be generated from TK \( \alpha, \beta, \gamma \). From the proof, it has been shown that the proposed ABTKIS scheme satisfies unconditional anonymity.

6. CONCLUSIONS

We introduce the notion of an attribute-based threshold key insulated signature (ABTKIS) and describe a construction that is based on an attribute-based signature (ABS).

Acknowledgements

This work is supported by NSFC under the Grants 60903189, 61202023, 61133014, 61073157.

REFERENCES


