ID-based Proxy Re-signature with Aggregate Property

ZHI-WEI WANG1,2 AND AI-DONG XIA1

1College of Computer
Nanjing University of Posts and Telecommunications
Nanjing, Jiangsu, 210003 China
E-mail: zhwwang@njupt.edu.cn
2Nanjing University of Information Science and Technology
Nanjing, Jiangsu, 210044 China

Recently, Garg et al. proposed an approximate candidate of leveled multi-linear map that can be used for unrestricted aggregation. In this work, we explore building construction of ID-based proxy re-signature with aggregate property, which has many applications. Our construction utilizes the full domain hash structure from multi-linear map proposed by Hohenburger et al. In particular, Hohenburger et al. proposed an ID-based signature with unrestricted aggregation. We build on this result to offer the first bidirectional ID-based proxy re-signature that admits unrestricted aggregation. In our construction, an arbitrary-sized set of signatures or re-signatures can be aggregated into a single group element, which authenticates the whole set. Our scheme can be proved selectively secure under the l + n-MCDH assumption.

Keywords: ID-based proxy re-signature, multi-linear map, unrestricted aggregation, l + n-MCDH assumption, selectively secure

1. INTRODUCTION

Proxy re-signature is a novel cryptographic primitive, which allows a proxy transform Alice’s (delegatee) signature to Bob’s (delegator) signature on the same message by using the re-signature key. Proxy re-signature is a good solution to many problems, e.g., proving the passed path that has been taken. Proxy re-signature was introduced by Blaze et al. (BBS) [2] in 1998, and Atieniese and Hohenberger [3] formalized it in 2005. After then, some proxy re-signature schemes have been proposed [4-6]. ID-based cryptography, proposed by Shamir [7], eliminates the necessity for the public key certificates. Hu et al. [19] firstly proposed an ID-based proxy re-signature scheme under the q-SDH (Strong Diffie-Hellman) assumption. However, Menon [20] pointed out that there exists a flaw in Hu’s scheme, with respect to the definitions of delegator and delegate security defined by Atieniese et al. [3]. Then, Shao et al. [8] firstly proposed a unidirectional ID-based proxy re-signature scheme in 2011. Recently, Tian [18] proposed an Identity-based proxy re-signature scheme from lattices, which underlying lattice problems are intractable for the quantum computers. In this work, we propose a bidirectional ID-based proxy re-signature scheme with aggregate property. Aggregate property is very important for the ID-based cryptographic primitives [15-17], which can greatly be reduced the communication cost. In general, there are eight properties for proxy re-signature [8].

Bidirectional: We call that a proxy re-signature scheme is a bidirectional scheme, on the
condition that the re-signature key allows proxy to transform A’s signature to B’s, and vice versa. Otherwise, if B’s signature cannot be transformed to A’s, we call it unidirectional.

**Multi-use:** If the signature can be re-signed for multi-times, then we call that the proxy re-signature scheme is a multi-use scheme.

**Private proxy:** If the re-signature key should be kept secretly by an honest proxy, then we call that the proxy re-signature scheme is a private proxy scheme.

**Transparent:** If a user cannot know whether a proxy exists in a scheme, then the proxy re-signature scheme is a transparent scheme. In a transparent scheme, the re-signature cannot be distinguished whether it is transformed by a proxy or generated by a signer.

**Key-optimal:** If a user only needs to keep a small number of secret keys regardless of how many re-signature processes he attends, then we call that the proxy re-signature scheme is a key-optimal scheme.

**Non-interactive:** If the delegatee’s secret key is not used to compute the re-signature key, then the scheme is a non-interactive scheme.

**ID-based:** If the user’s private key is generated from user’s identity information, and the signature should be verified by the user’s identity, then the proxy re-signature scheme is an ID-based scheme.

**Aggregate property:** If the signatures in proxy re-signature scheme can be aggregated, then we call the scheme has aggregate property.

We compare our ID-based scheme with other three ID-based schemes [8, 18, 19] in terms of the satisfied properties Table 1.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Bidirectional</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Multi-use</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Private proxy</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Transparent</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Key-optimal</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Non-interactive</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>ID-based</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Aggregate property</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Applications:** ID-based proxy re-signature scheme can be deployed in many application scenarios. For example, many countries are currently in processing of adopting E-passport. Compared with traditional passport, the E-passport has a more large storage capac-
ity for digital signatures. Suppose Alice arrives in Beijing from her motherland Japan, and she shows a signature $\sigma_A$ from Japan to China border patrol for proving that she is a good citizen. The border patrol officer checks this signature and translates it into $\sigma_B$ on his ID$_p$, stating that Alice has been passed the border patrol check. Next, when Alice’s E-passport is transferred to the custom officer, the custom officer only need to verify the signature on Alice’s passport by using the border patrol’s ID$_p$. If the signature is valid, and Alice passes the customs, then the custom officer translates the signature into $\sigma_C$ on his ID$_c$, etc.

If the number of arrival passengers is very large, then the channel between the border patrol and the custom officer is very busy. Thus, if the border patrol can aggregate a group of re-signatures into a single one, and send it to the custom officer for one time, then obviously, this measure will greatly improve the work efficiency. Then, this idea inspires us to design ID-based proxy re-signature with aggregate property.

If an ID-based proxy re-signature with aggregate property can be deployed in the E-passport system, it has many benefits. First, even the custom officer is corrupted, Alice can only skip the customs check, but she cannot gone through the initial checks by Japan and border patrol, since each checkpoint only has the re-signature key instead of the secret key. Second, the transformed signatures are aggregated by the border patrol to a single one [21], which greatly improves the communication efficiency between the border patrol and the custom officer.

**Organization:** In Section 2, we propose some definitions related to our construction and proof, and define the security definition of bidirectional ID-based proxy re-signature. In Section 3, we devise a basic proxy re-signature scheme from multi-linear maps. In Section 4, we construct a bidirectional ID-based proxy re-signature scheme with aggregate property. In Section 5, we give a proof to our proposed scheme, which is secure under $l + n$-MCDH assumption. In Section 6, we provide the performance analyses of our proposed scheme. In Section 7, we conclude our paper.
2. DEFINITION

2.1 Leveled Multi-linear Maps

In this section, we give a brief description of leveled multi-linear maps. More details of the Grah, Gentry and Halevi (GGH) grade algebras analogue of multi-linear maps can be seen in [13]. For generic, we assume that there exists a group generator \( \mathcal{G} \), which takes as input the security parameter \( 1^\lambda \) as well an integer number \( k \) to denote the number of allowed pairing operations, and outputs a sequence of groups \( \mathcal{G} = (\mathcal{G}_1, \ldots, \mathcal{G}_k) \) of prime order \( p \), with generators \( g_1, \ldots, g_k \) respectively, where we let \( g = g_1 \).

The map \( e_{ij} \) in the set of bilinear maps \( \{e_{ij} : \mathcal{G}_i \times \mathcal{G}_j \rightarrow \mathcal{G}_{i+j} \mid i, j \geq 1; i + j \leq k \} \) should satisfy that:

\[
e_{ij}(g^a, g^b) = g^{ab} \quad \forall a, b \in \mathbb{Z}_p.
\]

For simplicity, we write \( e(g_a, g_b) = g_{ab} \) in the following part.

**Assumption 1 (Multi-linear Computational Diffie-Hellman: \( k \)-MCDH):** A group generator \( \mathcal{G} \) takes as input \( 1^\lambda \) and \( k \), and outputs a sequence of groups \( \mathcal{G} = (\mathcal{G}_1, \ldots, \mathcal{G}_k) \) of prime order \( p \), with generators \( g_1, \ldots, g_k \) respectively, where we let \( g = g_1 \). Then, a challenger picks random \( c_1, \ldots, c_k \in \mathbb{Z}_p \). The assumption is that any polynomial time (PPT) adversary can compute \( g_{c_1}^{c_2} \cdots g_{c_k}^{c_1} \) with non-negligible probability, given \( g^{c_1}, \ldots, g^{c_k} \).

In [1], Hohenburger *et al.* proposed a variant of \( k \)-MCDH assumption in the approximate multi-linear maps setting of GGH.

2.2 Bidirectional ID-based Proxy Re-signature

**Definition 1 (Bidirectional ID-based Proxy Re-Signature):** A bidirectional ID-based proxy re-signature scheme consists of the following six probabilistic polynomial time (PPT) algorithms: \( \text{Initialize, KeyGen, ReKeyGen, Sign, ReSign, and Verify} \).

**Initialize:** On input the security parameter \( 1^\lambda \), the algorithm outputs the master public key \( MPK \) and the master secret key \( MSK \) for the trusted third party (TTP). Note that in the following algorithms, we implicitly contain the \( MPK \).

**KeyGen:** On input the \( MSK \) and a user’s identity \( ID \), the algorithm outputs the private key \( SK_{ID} \) for \( ID \).

**ReKeyGen:** On input the delegatee’s secret key \( SK_{ID_1} \) and the delegator’s secret key \( SK_{ID_2} \), the re-signature key generation algorithm outputs a re-signature key \( K_{ID_1 \rightarrow ID_2} \), which can be used to transform the signature of \( ID_1 \) to another signature of \( ID_2 \) on the same message. On the other hand, it also can generate a re-signature key \( K_{ID_2 \rightarrow ID_1} \), which can be used to transform the signature conversely.

**Sign:** On input a secret key \( SK_{ID} \) for the identity \( ID \), a message \( M \) from the message
space, the signature generation algorithm outputs the signature \( \sigma \) on \( M \) on behalf of \( ID \).

**ReSign:** On input a signature \( \sigma \) on message \( M \) on behalf of \( ID_1 \), and the re-signature key \( K_{ID_1 \rightarrow ID_2} \), the re-signature generation algorithm outputs a signature \( \sigma' \) on message \( M \) on behalf of \( ID_2 \) if \( \sigma \) is valid; \( \bot \), otherwise.

**Verify:** On input a signature \( \sigma \) on message \( M \) on behalf of \( ID \), and identity \( ID \), the verification algorithm outputs 1, if \( \sigma \) is valid; 0, otherwise.

**Correctness:** The following property must be satisfied for the correctness of an ID-based proxy re-signature scheme: For any message \( M \) in the message space and any two key pairs \((ID_1, SK_{ID_1})\) and \((ID_2, SK_{ID_2})\), let \( K_{ID_1 \rightarrow ID_2} \leftarrow \text{ReKeyGen}(SK_{ID_1}, SK_{ID_2}) \), the following two equations must hold:

\[ \text{Verify}(\sigma, M, ID_1) = 1, \]
\[ \text{Verify}(\text{ReSign}(\sigma, K_{ID_1 \rightarrow ID_2}), M, ID_2) = 1, \]

where \( \sigma \) is a signature on \( M \) on behalf of \( ID_1 \) from Sign or ReSign.

### 2.3 Security Model

The security model of ID-based proxy re-signature protects user from two kinds of attacks. The first one is launched from the parties outside the system (**External Security**). The second one is launched from inside the system, e.g., the proxy, or another valid user (**Internal Security**). We now provide the formal definition of these security notions.

**External Security:** The security notion protects a user from adversaries outside the system. That is, this security notion may make sense to require the standard notion of exist-\footnote{The algorithm stops, and outputs “failure”}.tual unforgeability. In this security notion, the re-signature key should be kept secretly, or it is easy for an adversary to “win”. We define it by the following game between an adversary \( A \) and a challenger \( C \). The challenger \( C \) should maintain an index/identity/secret key triples \( T \).

**Setup:** \( C \) runs the **Initialize** algorithm to get the \( MPK/MSK \), and sends \( MPK \) to \( A \).

**Queries:** \( A \) can make the following queries for polynomial times.

1. **Extract Queries:** On input an identity \( I \), if \( I = I^* \), \( C \) returns an error and records \( \langle i, I^*, \bot \rangle \) in \( T \). Otherwise, \( C \) responses \( A \) with **KeyGen**(MSK, \( I \)). Finally, \( C \) records \( \langle i, I, SK_i \rangle \) in \( T \).

2. **Sign Queries:** On input a message \( M \in \{0, 1\}^l \) and an index \( i \), the challenger \( C \) checks \( T \) whether \( \langle i, I, SK_i \rangle \) exists in \( T \). If not exists, then \( C \) returns an error. Otherwise, it returns **Sign**(\( M, I, SK_i \)).

3. **ReSign Queries:** On input \( (b, B, M, \sigma) \), \( C \) checks whether \( \text{Verify}(I_b, \sigma, M) = 1 \) holds.

   If it holds, it makes **Sign Query** on \( (B, M) \), and returns the result. Otherwise, it returns an error.

**Response:** Eventually, \( A \) outputs a signature \( \sigma^* \) on \( (I^*, M^*) \).

We say \( A \) wins the above game if (1) \( \text{Verify}(I^*, M^*, \sigma^*) = 1 \), (2) \( M^* \) was not queried
for a signature or re-signature by $A$ on any index corresponding to $\mathcal{I}_A$. We define the winning probability of $A$ as $ID-Forg_A^{external}$.

**Definition 2 (Adaptive Unforgeability for External Attacks):** An ID-based proxy re-signature scheme is existential unforgeability with respect to adaptive chosen-message attacks if for all PPT external adversaries, $ID-Forg_A^{External}$ is negligible.

If there is an initialization phase before the Setup phase, where in $A$ gives the challenger a forgery identity/message pair $(\mathcal{I}; M^{*})$, and $A$ cannot query the signing key or re-signing key for $M^{*}$, then we call it **selective security**.

Furthermore, in some unidirectional ID-based proxy re-signature schemes, one might want the re-signature keys to be public that it can make all users proxies. When in this case, there are no “external adversaries” to the system.

**Internal Security:** This security notion protects a user who is fooled by a rogue proxy or delegation partners. There are three guarantees to make.

**Limited Proxy:** If the delegatee and delegator are all honest, then the proxy cannot generate signatures for the delegator unless it has been signed by one of her delegatees, and cannot create any signatures for the delegatee. The secure game of limited proxy is similar to the external security game except that $A$ can make the re-signature key queries instead of the resign queries.

**Delegatee Security:** If the delegatee is honest, then he is “safe” from a colluding delegator and proxy. That is, they cannot produce any signatures for delegatee. However, in the bidirectional scheme, this property doesn’t apply, since both parties are delegators and delegatees.

**Delegator Security:** If the delegator is honest, then he is “safe” from a colluding delegatee and proxy. That is, they cannot produce any first level signatures for delegatee. However, in the bidirectional scheme, this property doesn’t apply, since both parties are delegators and delegatees.

3. OUR BASIC CONSTRUCTION OF PROXY RE-SIGNATURE

**Initialize $(\mathcal{I}^*, l)$:** The algorithm takes input as the security parameter $1^l$ as well the bit-length $l$ messages. The algorithm first runs the group generator $G(1^l, k = l + 1)$ and outputs a sequence of groups $\mathbb{G} = (\mathbb{G}_1, \ldots, \mathbb{G}_k)$ of prime order $p$, with generators $g_1, \ldots, g_k$ respectively, where we let $g = g_1$. Secondly, it randomly selects group elements $(A_{1,0} = g_1^{a_{1,0}}, A_{1,1} = g_1^{a_{1,1}}), \ldots, (A_{l,0} = g_1^{a_{l,0}}, A_{l,1} = g_1^{a_{l,1}}) \in \mathbb{G}_1$. Then, it will define a full domain hash function $H(M) : \{0, 1\}^l \rightarrow \mathbb{G}_{l+1}$. Let $m_1, \ldots, m_l$ be the bits of message $M$. The full domain hash function $H$ is computed iteratively as

$$H_i(M) = A_{i,m_i}, \quad \text{for } i \in \{2, \ldots, l\}, \quad H(M) = e(H_{l+1}(M), A_{l,0}).$$

We define $H(M) = H_i(M) = g_1^{\prod_{i\in[0,l]} m_i}$. The public parameters $PP$ is consisted of the
ID-BASED PROXY RE-SIGNATURE WITH AGGREGATE PROPERTY

4. BIDIRECTIONAL ID-BASED PROXY RE-SIGNATURE WITH AGGREGATE PROPERTY

Initialize $\langle 1^k, l, n \rangle$: The algorithm is run by the trusted third party of ID-based system. It takes as input the security parameter $1^k$ as well the bit-length $l$ messages and bit-length $n$ of identities. The algorithm first runs the group generator $G = \langle G_1, \ldots, G_l \rangle$ of prime order $p$, with generators $g_1, \ldots, g_l$ respectively, where we let $g = g_1$. Secondly it randomly selects group elements $(A_{1,0} = g^{a_{1,0}}, A_{1,1} = g^{a_{1,1}}, \ldots, A_{l,0} = g^{a_{l,0}}, A_{l,1} = g^{a_{l,1}}) \in G^l$. It also chooses random exponents $(b_{l,0}, b_{l,1}), \ldots, (b_{0,l}, b_{1,l}) \in \mathbb{Z}_p$, and sets $B_{i,\beta} = g^\beta$ for $i \in [n]$ and $\beta \in \{0, 1\}$.

Then, it will define a full domain hash function $H(\mathcal{I}, M): \{0, 1\}^n \times \{0, 1\}^l \to G_k$. Let $m_1, \ldots, m_l$ be the bits of message $M$ and $id_{i_1}, \ldots, id_{i_n}$ as the bits of $\mathcal{I}$. The full domain hash function $H$ is computed iteratively as

$$H_1(\mathcal{I}, M) = B_{1, id_1},$$
$$H_i(\mathcal{I}, M) = e(H_{i-1}(\mathcal{I}, M), B_{i, id_i})$$

for $i \in \{2, \ldots, n\}$ and $i \in \{n+1, \ldots, n+l = k\}$.

Then, for $i \in \{2, \ldots, n\}$ and $i \in \{n+1, \ldots, n+l = k\}$, and for $i \in \{2, \ldots, n\}$ and $i \in \{n+1, \ldots, n+l = k\}$, we have

$$H_i(\mathcal{I}, M) = e(H_{i-1}(\mathcal{I}, M), A_{i, id_i}).$$
The MPK is consisted of the group sequence description plus:

\[(A_{1,0}, A_{1,1}), \ldots, (A_{l,0}, A_{l,1}), \ldots, (B_{n,0}, B_{n,1})\]

The MSK is \((b_{1,0}, b_{1,1}), \ldots, (b_{n,0}, b_{n,1})\).

**KeyGen**\((MSK, \mathcal{I} \in \{0, 1\}^n)\): This algorithm takes as input the MSK and the identity \(\mathcal{I}\), outputs \(SK_{\mathcal{I}} = (g_{i=1}^{\prod g_{i=1}^{\prod}} g_{i=1}^{\prod} \). \)

**ReKeyGen**\((SK_{\mathcal{I}}, SK_{\mathcal{J}})\): This algorithm takes as input the delegate \(\mathcal{I}\)'s secret key \(SK_{\mathcal{I}} = (g_{i=1}^{\prod g_{i=1}^{\prod}} g_{i=1}^{\prod} \) and the delegator \(\mathcal{J}\)'s secret key \(SK_{\mathcal{J}} = (g_{i=1}^{\prod g_{i=1}^{\prod}} g_{i=1}^{\prod} \), output the re-signature key as \(K_{\mathcal{I}, SK_{\mathcal{J}}} = e(g_{i=1}^{\prod g_{i=1}^{\prod}} g_{i=1}^{\prod} = g_{i=1}^{\prod} \).

**Sign**\((M \in \{0, 1\}^l, SK_{\mathcal{I}}, M \in \{0, 1\}^n)\): The **Sign** algorithm sets \(D_0 = SK_{\mathcal{I}}\) and for \(i = 1\) to \(l\), it computes \(D_i = e(D_{i-1}, A_{i,0}) \in G_{n,1}^{\prod} \). The output signature is \(\sigma = D_l = g_{i=1}^{\prod} g_{i=1}^{\prod} \).

**ReSign**\((\sigma, K_{\mathcal{I} \rightarrow \mathcal{J}}, \mathcal{J}, \sigma')\): This algorithm takes as input the delegate \(b\)'s signature \(\sigma_b\) and the re-signature key \(K_{\mathcal{I} \rightarrow \mathcal{J}}\), and outputs the transformed signature of delegator \(B\) as \(\sigma_b = e(g_{i=1}^{\prod g_{i=1}^{\prod}} g_{i=1}^{\prod} = g_{i=1}^{\prod} \).

**Verify**\((\mathcal{I}, M, \sigma)\): The algorithm accepts if and only if \(e(\sigma, g) = H(\mathcal{I}, M)\).

**Aggregate Property**: Our scheme has the aggregate property that if \(\tilde{\sigma} \) and \(\sigma'\) serves as two aggregate signatures for the (single elements) multi-sets \(S = (\tilde{\mathcal{I}}, \tilde{M})\) and \(S' = (\mathcal{J}', M')\), then the aggregation algorithm simply computes the aggregate signature \(\sigma = e(\tilde{\sigma}, g_{i=1}^{\prod})\) on the multi-set \(S = \tilde{S} \cup S'\). The form of signature in our scheme is the same as Hohenburger’s ID-based aggregate signature [1]. So, the aggregation is unrestricted and can be done by any third party.

**5. SECURITY PROOF**

**Theorem 1**: Our ID-based proxy re-signature scheme for message length \(l\) and identity length \(n\) is selective secure under \(l + n\)-MCDH assumption (External and Internal Security).

**Proof**: We show the security in two parts.

**External Security**: We show that if there exists a PPT adversary \(A\) can break the selective security of our ID-based proxy re-signature scheme with probability \(\epsilon\) for message length \(l\), identity length \(n\) and security parameter \(\lambda\), then there exists a PPT challenger \(C\)
break the \( l + n \)-MCDH assumption for security parameter \( \lambda \) with the same probability. The challenger \( \mathcal{C} \) takes as input a MCDH instance \( g, g^\lambda, \ldots, g^\lambda \) together with group descriptions where \( k = l + n \). Let \( m_i \) denote the \( i \)th bit of message \( M \), and \( id_i \) denote the \( i \)th bit of \( T \). The challenger \( \mathcal{C} \) should maintain an index/identity/secret key triples \( T \), and interacts with \( A \) in the game as follows:

**Init:** Let \( T \in \{0, 1\}^n \) and \( M' \in \{0, 1\}^l' \) be the forgery identity/message pair provided by \( A \).

**Setup:** \( \mathcal{C} \) chooses randomly \( x_1, \ldots, x_l, y_1, \ldots, y_n \in \mathbb{Z}_p \). For \( i = 1 \) to \( n \), let \( B_{i \cdot \lambda} = g^{x_i} \) and \( B_{i \cdot \lambda} = g^{y_i} \), and for \( i = 1 \) to \( n \), let \( A_{i \cdot \lambda} = g^{x_i} \) and \( B_{i \cdot \lambda} = g^{y_i} \).

**Queries:** If the queried identity and message are different from the challenged identity and message in at least one bit, then \( \mathcal{C} \) will be able to respond the extracted secret keys and signatures for \( A \).

1. **Extract Queries:** On input an identity \( T \), if \( T = T' \), \( \mathcal{C} \) returns an error and records \( \langle i, T', _\perp \rangle \) in \( T \). Otherwise, \( \mathcal{C} \) computes \( s_i = g_{i,0}^{y_i} \), \( s_i = g_{i,0}^{y_i} \), and returns \( \mathcal{S} \), \( \mathcal{S} \), \( \mathcal{S} \), and \( \mathcal{T} \). Finally, \( \mathcal{C} \) records \( \langle i, T, \mathcal{S} \rangle \) in \( T \).

2. **Sign Queries:** On input a message \( M = \{0, 1\}^{l'} \) and an index \( i \), the challenger \( \mathcal{C} \) checks \( T \) whether \( \langle i, T, \mathcal{S} \rangle \) exists in \( T \). If not exists, then \( \mathcal{C} \) returns an error. Otherwise, if \( T_i = T' \), the challenger signs \( M \) in the usual way. If \( T_i = T' \), then \( M' \neq M \) should hold. Let \( \beta \) be the first bit such that \( M_\beta \neq M'_\beta \). Then, \( \mathcal{C} \) computes \( \sigma = g_{i,0}^{y_i} \), and \( \sigma' = g_{i,0}^{y_i} \). Following that, \( \mathcal{C} \) computes \( \rho = g_{i,0}^{y_i} \). Finally, \( \mathcal{C} \) returns \( \sigma = e(\rho, \sigma') = g_{i,0}^{y_i} \).

3. **Resign Queries:** On input \( (b, B, M, \sigma) \), \( \mathcal{C} \) checks whether \( \mathcal{B} \) is a MCDH problem, since \( e(\sigma', g) = H(T', M') = g_{i,0}^{y_i} \). So, \( \mathcal{C} \) gives \( \sigma' = g_{i,0}^{y_i} \) as the solution to the MCDH problem.

**Internal Security:** Since our scheme is a bidirectional, internal security refers only to Limited Proxy security that is a guarantee that the proxy cannot sign on behalf of other honest users by using its re-signature key. We show that if there exists a PPT proxy \( \mathcal{A} \) can forge with probability \( \epsilon \) for message length \( l \), identity length \( n \) and security parameter \( \lambda \), then there exists a PPT challenger \( \mathcal{C} \) can break the \( l+n \)-MCDH assumption for security parameter \( \lambda \) with the same probability \( \epsilon \). The challenger \( \mathcal{C} \) takes as input a MCDH instance \( g, g^\lambda, \ldots, g^\lambda \) together with group descriptions where \( k = l + n \). Let \( m_i \) denote the \( i \)th bit of message \( M \), and \( id_i \) denote the \( i \)th bit of \( T \). The challenger \( \mathcal{C} \) should maintain an index/identity/secret key triples \( T \), and interacts with \( \mathcal{A} \) in the game as follows:
Init: Let $T \in \{0, 1\}^n$ and $M^* \in \{0, 1\}^i$ be the forgery identity/message pair provided by $A$.

Setup: $C$ chooses randomly $x_1, \ldots, x_t, y_1, \ldots, y_n \in \mathbb{Z}_p$. For $i = 1$ to $n$, let $B_{i, \sigma'} = g^{x_i}$ and $B_{i, \sigma} = g^{y_i}$, and for $i = 1$ to $n$, let $A_{i, \sigma} = g^{x_i}$ and $A_{i, \sigma'} = g^{y_i}$.

Queries: If the queried identity and message are different from the challenged identity and message in at least one bit, then $C$ will be able to respond the extracted secret keys and signatures for $A$.

1. Extract Queries: On input an identity $T$, if $T = T'$, $C$ returns an error and records $(i, T', \bot)$ in $T$. Otherwise, $C$ responses $A$ as follows. Let $\beta$ be the first bit such that $id_i \neq id_i'$. Then $C$ computes $s_1 = g_{x_1}^{\prod_{a \neq \sigma} b^{y_a}}$, $s_2 = g_{x_1}^{\prod_{a \neq \sigma'} b^{y_a}}$, and returns $SK_z = (s_1^y, s_2^y) = g_{x_1}^{\prod_{a \neq \sigma} b^{y_a}}$, $g_{x_1}^{\prod_{a \neq \sigma'} b^{y_a}}$. Finally, $C$ records $(i, T, SK_z)$ in $T$.

2. Sign Queries: On input a message $M^* \in \{0, 1\}^i$ and an index $i$, the challenger $C$ checks $T$ whether $(i, T, SK_z)$ exists in $T$. If not exists, then $C$ returns an error. Otherwise, if $T_i = T_i'$, the challenger signs $M$ in the usual way. If $T_i = T_i'$, then $M_i \neq M_i'$ should hold. Let $\beta$ be the first bit such that $M_i \neq M_i'$. Then, $C$ computes $\sigma^* = g_{x_1}^{\prod_{a \neq \sigma} b^{y_a}}$, and $\sigma^* = g_{x_1}^{\prod_{a \neq \sigma'} b^{y_a}}$. Following that, $C$ computes $\rho = g_{x_1}^{\prod_{a \neq \sigma} b^{y_a}}$. Finally, $C$ returns $\sigma = e(\rho, \sigma^*) = g_{x_1}^{\prod_{a \neq \sigma} b^{y_a}}$

3. ReKey Queries: On input $(T, \sigma)$, if $\sigma = \sigma'$ or $\sigma = \sigma'$, then $C$ outputs an error. Otherwise, it makes the Extract Queries on $(T, \sigma)$, and gets $SK_z = (SK_{B1}, SK_{B2})$ and $SK_{z} = (SK_{A1}, SK_{A2})$. Then, it responses $e(SK_{B1}, SK_{B2})$ as the re-signature key $K_{z} = \sigma^z$.

Response: Eventually, $A$ outputs a signature $\sigma^*$ on $(T', M^*)$. Then, $C$ will extract from this as a solution to the MCDH problem, since $e(\sigma^*, g) = H(T', M^*) = g_{x_1}^{\prod_{a \neq \sigma} b^{y_a}}g_{x_1}^{\prod_{a \neq \sigma'} b^{y_a}} = g_{x_1}^{\prod_{a \neq \sigma} b^{y_a}}$. So, $C$ gives $\sigma^* = g_{x_1}^{\prod_{a \neq \sigma} b^{y_a}}$ as the solution to the MCDH problem.

6. PERFORMANCE ANALYSES

In this section, we will show the performance of our ID-based scheme with respect to the required computational complexity and the communication cost in each phases. Computational complexity is mainly measured by the required pairing operations and the exponentiation operations, since these two operations are the most “expensive” operations compared with other operations. We test the time of pairing and exponentiation in group by using the Stanford Pairing-based Crypto library [14]. We choose the type A elliptic curve with the order $r$ of group is 160bits long, and the base field order $q$ is 1024 bits long. We compile our test code on the hardware platform: a 2.5GHz Intel Core i5 CPU with 4GB 1600MHz DDR3 RAM running OS X 10.9.3. The time of pairing requires about 4.1ms, while the time of exponentiation needs about 3.6ms. However, the time of scalar multiplication needs only about 0.037ms.

Let $T_p$ denote the time of one pairing operation, and $T_e$ denote the time of one exponentiation operation. We assume that elements in $p$-order group can be encoded as bit strings of length $\log p$. Table 2 shows the analyses of computational complexity and communication cost of our ID-based scheme in each phases.
Note: We only consider the pairings and exponentiations in computational complexity, and omit other operations. Let $n$, $l$ denote the bit-lengths of identity and message respectively.

Our ID-based scheme has the aggregate property, while Shao et al.’s scheme [8] has not. We assume that there exists $v$ signatures to be verified. In our aggregate scheme, the communication cost of signature in our aggregate scheme only needs $\log p$ bits, while Shao et al.’s scheme [8] requires $2v\log p$ bits and Hu et al.’s scheme [19] requires $4v\log p$ bits.

Table 2. Performance analyses of our ID-based scheme.

<table>
<thead>
<tr>
<th>Phase</th>
<th>Computational complexity</th>
<th>Communication cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initialize</td>
<td>$2(n+l)T_e$</td>
<td>$2(n+l)\log p$</td>
</tr>
<tr>
<td>KeyGen</td>
<td>$2T_e$</td>
<td>$2\log p$</td>
</tr>
<tr>
<td>ReKeyGen</td>
<td>$T_p$</td>
<td>$\log p$</td>
</tr>
<tr>
<td>Sign</td>
<td>$lT_p$</td>
<td>$\log p$</td>
</tr>
<tr>
<td>ReSign</td>
<td>None</td>
<td>$\log p$</td>
</tr>
<tr>
<td>Verify</td>
<td>$(n+l)T_p$</td>
<td>None</td>
</tr>
</tbody>
</table>

7. CONCLUSIONS

In this paper, we propose a bidirectional ID-based proxy re-signature scheme with aggregate property, which can be proved selective secure under $l + n$-MCDH assumption. If this scheme can be deployed in practical, it has many benefits. First, the outside attacker cannot forge a signature even for a previously signed message. Second, a rogue proxy or a delegation partner also cannot forge a signature for a user. Third, the aggregate property makes the communication cost to be greatly reduced.

ACKNOWLEDGMENT

This research is partially supported by the National Natural Science Foundation of China under Grant No. 61373006, 61202353, 61272422 and the PAPD fund.

REFERENCES


21. Z. Wang, G. Sun, and D. Chen, “A new definition of homomorphic signature for

**Zhiwei Wang** received his Ph.D. degree in Cryptography from the Beijing University of Posts and Telecommunications, Beijing in 2009. Currently, he is an Associate Professor in the Department of Information Security at Nanjing University of Posts and Telecommunications. 2014.3-2015.3 He served as a Research Associate in the Department of Computer Sciences at the University of Hongkong. His research interests include digital signatures, provable security, cryptographic protocols, and network and cloud security. Dr. Wang has published over 30 papers at prestigious journals and conferences. He has been served as a TPC member for MobiPST2011-2014, ACSA-Summer 2012, and MIST 2012, and a session chair for IEEE TrustCom 2013, ACSA 2011.

**Aidong Xia** is a master student of Nanjing University of Posts and Telecommunications. His research direction is cryptography and information security.