An extended abstract of this paper appears in Indocrypt 2007, K. Srinathan, C. Pandu Rangan, M. Yung (Eds.), volume 4859 of LNCS, pp. 197-209, Sringer-Verlag, 2007.

Proxy Re-Signature Schemes without Random Oracles*

Jun Shao
chn.junshao@gmail.com

Zhenfu Cao†
zfcao@cs.sjtu.edu.cn

Licheng Wang
wanglc.cn@gmail.com

Xiaohui Liang
liangxh127@gmail.com

Department of Computer Science and Engineering
Shanghai Jiao Tong University

Abstract

To construct a suitable and secure proxy re-signature scheme is not an easy job, up to now, there exist only three schemes, one is proposed by Blaze *et al.* [6] at EUROCRYPT 1998, and the others are proposed by Ateniese and Hohenberger [2] at ACM CCS 2005. However, none of these schemes is proved in the standard model (i.e., do not rely on the random oracle heuristic). In this paper, based on Waters' approach [19], we first propose a multi-use bidirectional proxy re-signature scheme, denoted as S_{mb}, which is existentially unforgeable in the standard model. And then, we extend S_{mb} to be a multi-use bidirectional ID-based proxy re-signature scheme, denoted by S_{id-mb}, which is also existentially unforgeable in the standard model. Both of these two proposed schemes are computationally efficient, and their security bases on the Computational Diffie-Hellman (CDH) assumption.

Keywords: *proxy re-signature, standard model, ID-based, bilinear maps, existential unforgeability.*

1 Introduction

Proxy re-signature schemes, introduced by Blaze, Bleumer, and Strauss [6], and formalized later by Ateniese and Hohenberger [2], allow a semi-trusted proxy to transform a delegatee's signature into a delegator's signature on the same message by using some additional information. The proxy, however, cannot generate arbitrary signatures on behalf of either the delegatee or the delegator. Generally speaking, a proxy re-signature scheme has eight desirable properties [2], though none of existing schemes satisfies all properties, see Table 1.

1. **Unidirectional:** In an *unidirectional* scheme, a re-signature key allows the proxy to transform A's signature to B's but not vice versa. In a *bidirectional* scheme, on the other hand, the re-signature key allows the proxy to transform A's signature to B's as well as B's signature to A's.
2. **Multi-use:** In a *multi-use* scheme, a transformed signature can be re-transformed again by the proxy. In a *single-use* scheme, the proxy can transform only the signatures that have not been transformed.
3. **Private Proxy:** The re-signature key can be kept secret by the proxy in a *private proxy* scheme, but can be recomputed by observing the proxy passively in a *public proxy* scheme.
4. **Transparent:** In a *transparent* scheme, a signature on the same message signed by the delegator is computationally indistinguishable from a signature transformed by a proxy.
5. **Key-Optimal:** In a *key-optimal* scheme, a user is required to protect and store only a small constant amount of secrets no matter how many signature delegations the user gives or accepts.
6. **Non-interactive:** The delegatee is not required to participate in a delegation process.

*Supported by National Natural Science Foundation of China, No.60673079 and No.60572155, Research Fund for the Doctoral Program of Higher Education, No.20060248008.

†Corresponding Author.

7. Non-transitive: A re-signing right cannot be re-delegated by the proxy alone.

8. Temporary: A re-signing right is temporary.

Table 1: The properties that the existing proxy re-signature schemes and ours satisfy.

Property	BBS [6]	S_{bi} [2]	S_{uni} [2]	S_{mb}	S_{id-mb}
1.	No	No	Yes	No	No
2.	Yes	Yes	No	Yes	Yes
3.	No	Yes	No	Yes	Yes
4.	Yes	Yes	Yes	Yes	Yes
5.	Yes	Yes	Yes	Yes	Yes
6.	No	No	Yes	No	No
7.	No	No	Yes	No	No
8.	No	No	Yes	No	No

Due to the transformation function, proxy re-signature schemes are very useful and can be applied in many applications, including simplifying key management [6], providing a proof for a path that has been taken, managing group signatures, simplifying certificate management [2], constructing a Digital Rights Management (DRM) interoperable system [18]. However, as mentioned in [2], "Finding suitable and secure proxy re-signature schemes required a substantial effort. Natural extensions of several standard signatures were susceptible to the sort of problems." To our best knowledge, there are only three proxy re-signature schemes, the first one is a *bidirectional*, *multi-use*, and *public proxy* scheme, proposed by Blaze, Bleumer and Strauss at Eurocrypt 1998 [6], and the left two are both proposed by Ateniese and Hohenberger at ACM CCS 2005 [2]. One of them is a *multi-use bidirectional* scheme, and the other is a *single-use unidirectional* scheme.

However, there exist two disadvantages in the above three schemes.

- All of these three schemes are only proven secure in the random oracle model, i.e., the proof of security relies on the random oracle heuristic. However, it has been shown that some schemes are proven secure in the random oracle model, but are trivially insecure under any instantiation of the oracle [9, 5]. Up to now, there are many signatures proven secure in the standard model, such as [10, 12, 3, 4, 19, 20]. It is natural to ask whether we can construct a new proxy re-signature scheme which can be proved in the standard model.
- The public keys in these three schemes are arbitrary strings unrelated to their owner's identity. A certificate issued by an authority is needed to bind the public key to its owner's identity before the public key is used by others. This creates complexity of certificate management, though proxy re-signature schemes can be used to simplify certificate management. A natural solution to this disadvantage is to apply ID-based cryptography [17]. In ID-based cryptography, a user's unique ID such as an email address is also the user's public key. The corresponding private key is computed from the public key by a Private Key Generator (PKG) who has the knowledge of a master secret. As a result, complexity of certificate management can be eliminated. We can use the method in [11] to convert any proxy re-signature into an ID-based proxy re-signature. However, as mentioned in [15], this method expands the size of signature, and increases the complexity of verification. We hope that we get an ID-based proxy re-signature by a direct construction.

In this paper, we attempt to propose a new proxy re-signature scheme which recovers the above two disadvantages.

1.1 Our Contribution

In this paper, based on Waters' approach [19], we first propose the first proxy re-signature scheme which is existentially unforgeable in the standard model, we denote it as S_{mb}. S_{mb} satisfies bidirectional, multi-use, private proxy, transparent properties. And then we proposed the first ID-based proxy re-signature which is existentially unforgeable in the standard model, we denote it as S_{id-mb}. S_{id-mb} also satisfies bidirectional, multi-use, private proxy, transparent properties. Actually, S_{id-mb} can be considered as an ID-based extension of S_{mb}. As the schemes in [19], both of our proposed schemes are constructed in bilinear groups, and proven secure under the Computational Diffie-Hellman (CDH) assumption. The only drawback of our proposed schemes is the relatively large size of its public parameters inheriting from Waters' approach [19]. However, we can use the techniques of Naccache [14] and Sarkar and Chatterjee [16] to reduce the size of the public parameters.

1.2 Paper Organization

The remaining paper is organized as follows. In Section 2, we review the definitions of (ID-based) proxy re-signatures and their security. And then we present S_{mb}, S_{id-mb} and their security proofs in Section 3. Finally, We conclude the paper in Section 4.

2 Definitions

The security notions in this section are all for existential unforgeablility under an adaptive chosen message (and identity) attack. That is, a valid forgery should be a valid signature on a new message, which is not signed by the singer before. These security models can be easily extended to cover strong unforgeability [1], where a valid forgery should be a valid signature which is not computed by the signer. However, our concrete schemes do not enjoy security in this stronger sense, since an adversary can easily modify existing signatures into new signatures on same message.

2.1 Bidirectional Proxy Re-Signature

In this subsection, we briefly review the definitions about bidirectional proxy re-signatures. The security notion in this subsection is for existential unforgeability under an adaptive chosen message attack, which is weaker than that in [2]. We refer the reader to [2] for details.

Definition 1 *A bidirectional proxy re-signature scheme is a tuple of (possibly probabilistic) polynomial time algorithms* ($\texttt{KeyGen}$, $\texttt{ReKey}$, $\texttt{Sign}$, $\texttt{ReSign}$, $\texttt{Verify}$)*, where:*

- ($\texttt{KeyGen}$, $\texttt{Sign}$, $\texttt{Verify}$) are the same as those in the standard digital signatures[1].
- On input (sk_A, sk_B), the re-signature key generation algorithm, $\texttt{ReKey}$, outputs a key $rk_{A\leftrightarrow B}$ for the proxy, where sk_A and sk_B are the secret key of A and B, respectively.
- On input $rk_{A\leftrightarrow B}$, a public key pk_A, a message m, and a signature σ, the re-signature function, $\texttt{ReSign}$, outputs a new signature σ' on message m corresponding to pk_B, if $\texttt{Verify}(pk_A, m, \sigma) = 1$ and $\perp$ otherwise.

Correctness. For any message m in the message space and any key pairs $(pk, sk), (pk', sk') \leftarrow \texttt{KeyGen}(1^k)$, let $\sigma = \texttt{Sign}(sk, m)$ and $rk \leftarrow \texttt{ReKey}(sk, sk')$. Then the following two conditions must hold:

$$\texttt{Verify}(pk, m, \sigma) = 1 \ \text{ and } \ \texttt{Verify}(pk', m, \texttt{ReSign}(rk, pk, m, \sigma)) = 1.$$

Unlike the security notion in [2], we define security for bidirectional proxy re-signature schemes by the following game between a challenger and an adversary: (Note that we adopt the method in [8] to define the security notion of bidirectional proxy re-encryption schemes: static corruption, i.e., in this security notion, the adversary has to determine the corrupted parties before the computation starts, and it does not allow adaptive corruption of proxies between corrupted and uncorrupted parties.)

[1] For the definition of standard digital signatures, we refer the reader to [13].

Queries. The adversary adaptively makes a number of different queries to the challenger. Each query can be one of the following.

- Uncorrupted Key Generation $\mathcal{O}_{UKeyGen}$: Obtain a new key pair as $(pk, sk) \leftarrow \texttt{KeyGen}(1^k)$. The adversary is given pk.
- Corrupted Key Generation $\mathcal{O}_{CKeyGen}$: Obtain a new key pair as $(pk, sk) \leftarrow \texttt{KeyGen}(1^k)$. The adversary is given pk and sk.
- Re-Signature key Generation $\mathcal{O}_{ReKey}$: On input (pk, pk') by the adversary, where pk, pk' were generated before by `KeyGen`, return the re-signature key $rk_{pk \leftrightarrow pk'} = \texttt{ReKey}(sk, sk')$, where sk, sk' are the secret keys that correspond to pk, pk'. Like the security notion in [8], here, we also require that both pk and pk' are corrupted, or both are uncorrupted.
- Re-signature $\mathcal{O}_{ReSign}$: On input (pk, pk', m, σ), where pk, pk' were generated before by `KeyGen`. The adversary is given the re-signed signature $\sigma' = \texttt{ReSign}(\texttt{ReKey}(sk, sk'), pk, m, \sigma)$, where sk, sk' are the secret keys that correspond to pk, pk'.
- Signature $\mathcal{O}_{Sign}$: On input a public key pk, a message m, where pk was generated before by `KeyGen`. The adversary is given the corresponding signature $\sigma = \texttt{Sign}(sk, m)$, where sk is the secret key that correspond to pk.

Forgery. The adversary outputs a message m^*, a public key pk^*, and a string σ^*. The adversary succeeds if the following hold true:

1. $\texttt{Verify}(pk^*, m^*, \sigma^*) = 1$.
2. pk^* is not from $\mathcal{O}_{CKeyGen}$.
3. (pk^*, m^*) is not a query to $\mathcal{O}_{Sign}$.
4. $(\Diamond, pk^*, m^*, \blacklozenge)$ is not a query to $\mathcal{O}_{ReSign}$, where $\Diamond$ denotes any public key, and $\blacklozenge$ denotes any signature.

The advantage of an adversary $\mathcal{A}$ in the above game is defined to be $\text{Adv}_{\mathcal{A}} = \Pr[\mathcal{A} \text{ succeeds}]$, where the probability is taken over all coin tosses made by the challenger and the adversary.

2.2 Bidirectional ID-based Proxy Re-Signature

Definition 2 (Bidirectional ID-based Proxy Re-Signature) *A Bidirectional ID-based proxy re-signature scheme $\mathcal{S}$ consists of the following six random algorithms:* `Setup`*,* `Extract`*,* `ReKey`*,* `Sign`*,* `ReSign`*, and* `Verify` *where:*

- (`Setup`, `Extract`, `Sign`, `Verify`) are the same as those in a standard ID-based signature[2].
- On input (d_A, d_B), the re-signature key generation algorithm, `ReKey`, outputs a key $rk_{A \leftrightarrow B}$ for the proxy, where d_A (d_B) is A's (B's) secret key.
- On input $rk_{A \leftrightarrow B}$, an identity ID_A, a message m, and a signature σ, the re-signature algorithm, `ReSign`, outputs a new signature σ' on message m corresponding to ID_B, if $\texttt{Verify}(ID_A, m, \sigma) = 1$ and $\perp$ otherwise.

Correctness: This is the same as that in standard proxy re-signature schemes. The following property must be satisfied for the correctness of a proxy re-signature: For any message m in the message space and any two key pairs (ID_A, d_A), and (ID_B, d_B), let $\sigma_A = \texttt{Sign}(d_A, m)$ and $rk_{A \leftrightarrow B} \leftarrow \texttt{Rekey}(d_A, d_B)$, the following two equations must hold:

$$\texttt{Verify}(ID_A, m, \sigma_A) = 1, \text{ and } \texttt{Verify}(ID_B, m, \texttt{ReSign}(rk_{A \leftrightarrow B}, ID_A, m, \sigma_A)) = 1.$$

We also define the security notion of bidirectional ID-based proxy re-signature with static corruption by a game between a challenger and an adversary.

[2] For the definition of ID-based signatures, we refer the reader to [17].

Setup. The challenger runs `Setup` and obtains both the public parameters *params* and the master secret mk. The adversary is given *params* but the master secret *mk* is kept by the challenger.

Queries. The adversary adaptively makes a number of different queries to the challenger. Each query can be one of the following.

- Extract oracle for corrupted parties $O_{Extract}$: On input an identity ID by the adversary, the challenger responds by running $\texttt{Extract}(mk, ID)$. and sends the resulting private key d_{ID} to the adversary.
- Re-Signature key Generation $\mathcal{O}_{ReKey}$: On input (ID_A, ID_B) by the adversary, the challenger returns the re-signature key $rk_{A\leftrightarrow B} = \texttt{ReKey}(\texttt{Extract}(mk, ID_A), \texttt{Extract}(mk, ID_B))$. Here, we also require that both ID_A and ID_B are corrupted, or both are uncorrupted.
- Re-signature $\mathcal{O}_{ReSign}$: On input (ID_A, ID_B, m, σ), the adversary is given the re-signed signature

$$\sigma' = \texttt{ReSign}(\texttt{ReKey}(\texttt{Extract}(mk, ID_A), \texttt{Extract}(mk, ID_B)), ID_A, m, \sigma).$$

- Signature $\mathcal{O}_{Sign}$: On input an identity ID, a message m. The adversary is given the corresponding signature $\sigma = \texttt{Sign}(\texttt{Extract}(mk, ID), m)$.

Forgery. The adversary outputs a message m^*, an identity ID^*, and a string σ^*. The adversary succeeds if the following hold:

1. $\texttt{Verify}(pk^*, m^*, \sigma^*) = 1$.
2. ID^* is uncorrupted.
3. (ID^*, m^*) is not a query to $\mathcal{O}_{Sign}$.
4. $(\Diamond, ID^*, m^*, \blacklozenge)$ is not a query to $\mathcal{O}_{ReSign}$, where $\Diamond$ denotes any identity, and $\blacklozenge$ denotes any signature.

The advantage of an adversary $\mathcal{A}$ in the above game is defined to be $\text{Adv}_{\mathcal{A}} = \Pr[\mathcal{A} \text{ succeeds}]$, where the probability is taken over all coin tosses made by the challenger and the adversary.

2.3 Bilinear maps

In this subsection, we briefly review definitions about bilinear maps and bilinear map groups, which follow that in [7].

1. $\mathbb{G}_1$ and $\mathbb{G}_2$ are two (multiplicative) cyclic groups of prime order p;
2. g is a generator of $\mathbb{G}_1$;
3. e is a bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$.

Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two groups as above. An *admissible bilinear map* is a map $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ with the following properties:

1. *Identity*: For all $P \in \mathbb{G}_1$, $e(P, P) = 1$;
2. *Alternation*: For all $P, Q \in \mathbb{G}_1$, $e(P, Q) = e(Q, P)^{-1}$;
3. *Bilinearity*: For all $P, Q, R \in \mathbb{G}_1$, $e(P \cdot Q, R) = e(P, R) \cdot e(Q, R)$ and $e(P, Q \cdot R) = e(P, Q) \cdot e(P, R)$.
4. *Non-degeneracy*: If $e(P, Q) = 1$ for all $Q \in G_1$, then $P = \mathcal{O}$, where $\mathcal{O}$ is a point at infinity.

We say that $\mathbb{G}_1$ is a bilinear group if the group action in $\mathbb{G}_1$ can be computed efficiently and there exists a group $\mathbb{G}_2$ and an efficiently computable bilinear map as above.

2.4 The Computational Diffie-Hellman Assumption (CDH)

Computational Diffie-Hellman Problem. Let $\mathbb{G}$ be a group of prime order p and let g be a generator of $\mathbb{G}$. The CDH problem is as follows: Given $\langle g, g^a, g^b \rangle$ for some $a, b \in \mathbb{Z}_p^*$ compute g^{ab}. An algorithm $\mathcal{A}$ has advantage ε in solving CDH in $\mathbb{G}$ if

$$\Pr[A(g, g^a, g^b) = g^{ab}] \geq \varepsilon$$

where the probability is over the random choice of a, b in $\mathbb{Z}_p^*$, the random choice of $g \in \mathbb{G}^*$, and the random bits of $\mathcal{A}$.

Definition 3 *We say that the (ε, t)-CDH assumption holds in $\mathbb{G}$ if no t-time algorithm has advantage at least ε in solving the CDH problem in $\mathbb{G}$.*

3 Bidirectional Proxy Re-signature Schemes

3.1 S_{mb}: $\mathcal{M}$ulti-Use $\mathcal{B}$idirectional Scheme

We now present a new multi-use bidirectional proxy re-signature scheme, denoted as S_{mb}, using the signature scheme due to Waters [19]. This scheme requires a bilinear map, as discussed in Section 2. We assume that the messages can be represented as bit strings of length n_m, which is unrelated to p. We can achieve this by a collision-resistant hash function $H : \{0,1\}^* \rightarrow \{0,1\}^{n_m}$.

`KeyGen`: On input the security parameter 1^k, it chooses two groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of prime order $p = \Theta(2^k)$, such that an admissible pairing $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ can be constructed and chooses a generator g of $\mathbb{G}_1$. Furthermore, it selects a random a from $\mathbb{Z}_p$, and $n_m + 2$ random number $(g_2, u', u_1, \cdots, u_{n_m})$ from $\mathbb{G}_1$, and output the key pair $pk = g_1 = g^a$ and $sk = a$, the public parameters $(\mathbb{G}_1, \mathbb{G}_2, e, g_2, u', u_1, \cdots, u_{n_m})$.

`ReKey`: On input two secret keys $sk_A = a$, $sk_B = b$, output the re-signature key $rk_{A \rightarrow B} = b/a \bmod p$.

(Note that we make use of the same method and assumptions in [2] to get the re-signature key, we refer the reader to [2][Section 3.3] for details.)

`Sign`: On input a secret key $sk = a$ and a n_m-bit message m, output $\sigma = (\mathfrak{A}, \mathfrak{B}) = (g_2^a \cdot w^r, g^r)$, where r is chosen randomly from $\mathbb{Z}_p$, and $w = u' \cdot \prod_{i \in \mathcal{U}} u_i$, $\mathcal{U} \subset \{1, \ldots, n_m\}$ is the set of indicies i such that $m[i] = 1$, and $m[i]$ is the i-th bit of m.

`ReSign`: On input a re-signature key $rk_{A \rightarrow B}$, a public key pk_A, a signature σ_A, and a n_m-bit message m, check that $\texttt{Verify}(pk_A, m, \sigma_A) = 1$. If σ_A does not verify, output $\perp$; otherwise, output $\sigma_B = \sigma_A^{rk_{A \rightarrow B}} = (g_2^b \cdot w^{rb/a}, g^{rb/a}) = (g_2^b w^{r'}, g^{r'})$, where $r' = rb/a \bmod p$.

`Verify`: On input a public key pk, a n_m-bit message m, and a purported signature $\sigma = (\mathfrak{A}, \mathfrak{B})$, output 1, if $e(pk, g_2) e(\mathfrak{B}, w) = e(\mathfrak{A}, g)$ and 0 otherwise.

Theorem 1 (Security of S_{mb}) *In the standard model, bidirectional proxy re-signature scheme S_{mb} is correct and existentially unforgeable under the Computational Diffie-Hellman (CDH) assumption in $\mathbb{G}_1$; that is, for random $g \in \mathbb{G}_1$, and $x, y \in \mathbb{Z}_p^*$, give (g, g^x, g^y), it is hard to compute g^{xy}.*

Proof. The correctness property is easily observable. We show security following the approaches in [19, 15], especially the one in [15].

If there exists an adversary $\mathcal{A}$ that can break the above proxy re-signature scheme with non-negligible probability ε in time t after making at most q_S sign queries, q_{RS} resign queries, q_K (un)corrupted key queries, and q_{RK} rekey queries, then there also exists an adversary $\mathcal{B}$ that can solve the CDH problem in $\mathbb{G}_1$ with probability $\frac{\varepsilon}{4(q_S + q_{RS})(n_m + 1)}$ in time $t + O((q_S + q_{RS}) n_m \rho + (q_S + q_{RS} + q_K) \tau)$, where ρ and τ are the time for a multiplication and an exponentiation in $\mathbb{G}_1$, respectively.

On input (g, g^a, g^b), the CDH adversary $\mathcal{B}$ simulates a bidirectional proxy re-signature security game for $\mathcal{A}$ as follows:

To prepare the simulation, $\mathcal{B}$ first sets $l_m = 2(q_S + q_{RS})$, and randomly chooses a number k_m, such that $0 \leq k_m \leq n_m$, and $l_m(n_m + 1) < p$. $\mathcal{B}$ then chooses $n_m + 1$ random numbers x', $x_i(i = 1, \dots, n_m)$ from $\mathbb{Z}_{l_m}$. Lastly, $\mathcal{B}$ chooses $n_m + 1$ random numbers y', $y_i(i = 1, \dots, n_m)$ from $\mathbb{Z}_p$.

To make expression simpler, we use the following notations:

$$F(m) = x' + \sum_{i \in \mathcal{U}} x_i - l_m k_m \quad \text{and} \quad J(m) = y' + \sum_{i \in \mathcal{U}} y_i.$$

Now, $\mathcal{B}$ sets the public parameters:

$$g_2 = g^b, \; u' = g_2^{x' - l_m k_m} g^{y'}, \; u_i = g_2^{x_i} g^{y_i} (1 \leq i \leq n_m).$$

Note that for any message m, there exists the following equation:

$$w = u' \prod_{i \in \mathcal{U}} u_i = g_2^{F(m)} g^{J(m)}.$$

Queries: $\mathcal{B}$ builds the following oracles:

$O_{UKeyGen}$: $\mathcal{B}$ chooses a random $x_i \in Z_p^*$, and outputs $pk_i = (g^a)^{x_i}$.

$O_{CKeyGen}$: $\mathcal{B}$ chooses a random $x_i \in Z_p^*$, and outputs $(pk_i, sk_i) = (g^{x_i}, x_i)$.

$\mathcal{O}_{Sign}$: On input (pk_i, m), if pk_i is corrupted, $\mathcal{B}$ returns the signature $\sigma = (g_2^{x_j} w^r, g^r)$, where $w = u' \prod_{i \in \mathcal{U}} u_i$. Otherwise, $\mathcal{B}$ performs as follows.

- If $F(m) \not\equiv 0 \bmod p$, $\mathcal{B}$ picks a random $r \in \mathbb{Z}_p$ and computes the signature as,

$$\sigma = (g_1^{-J(m)/F(m)} (u' \prod_{i \in \mathcal{U}} u_i)^r, g_1^{-1/F(m)} g^r).$$

For $\tilde{r} = r - a/F(m)$, we have that

$$\begin{aligned} & g_1^{-J(m)/F(m)} (u' \textstyle\prod_{i \in \mathcal{U}} u_i)^r \\ = \; & g_1^{-J(m)/F(m)} (g^{J(m)} g_2^{F(m)})^r \\ = \; & g_2^a (g_2^{F(m)} g^{J(m)})^{-a/F(m)} (g^{J(m)} g_2^{F(m)})^r \\ = \; & g_2^a (g_2^{F(m)} g^{J(m)})^{r - a/F(m)} \\ = \; & g^{ab} (u' \textstyle\prod_{i=1}^n u_i^{\mathfrak{m}_i})^{\tilde{r}}, \end{aligned}$$

and

$$\begin{aligned} g_1^{-1/F(m)} g^r &= g^{r - a/F(m)} \\ &= g^{\tilde{r}}, \end{aligned}$$

which shows that σ has the correct signature as in the actual scheme.

- If $F(m) \equiv 0(\mathrm{mod} p)$, $\mathcal{B}$ is unable to compute the signature σ and must abort the simulation.

$\mathcal{O}_{ReKey}$: On input (pk_i, pk_j), if pk_i and pk_j are both corrupted or both uncorrupted, B returns $rk_{i \to j} = (x_j/x_i) \bmod p$; else, this input is illegal.

$\mathcal{O}_{ReSign}$: On input (pk_i, pk_j, m, σ). If $\mathtt{Verify}(pk_i, m, \sigma) \neq 1$, $\mathcal{B}$ outputs $\perp$. Otherwise, $\mathcal{B}$ does:

- If pk_i and pk_j are both corrupted or both uncorrupted, output $\mathtt{ReSign}(\mathcal{O}_{ReKey}(pk_i, pk_j), pk_i, m, \sigma)$.
- else, output $\mathcal{O}_{Sign}(pk_j, m)$.

Forgery: If $\mathcal{B}$ does not abort as a consequence of one of the queries above, $\mathcal{A}$ will, with probability at least ε, return a message m^* and a valid forgery $\sigma^* = (\mathfrak{A}^*, \mathfrak{B}^*)$ on m^*. If $F(m^*) \not\equiv 0 \bmod p$, $\mathcal{B}$ aborts. Otherwise, the forgery must be of the form, for some $r^* \in \mathbb{Z}_p$,

$$\begin{aligned}\sigma^* &= (g^{ab}(u' \textstyle\prod_{i \in \mathcal{U}} u_i)^{r^*}, g^{r^*})\\ &= (g^{ab}(g_2^{F(m^*)} g^{J(m^*)})^{r^*}, g^{r^*})\\ &= (g^{ab+J(m^*)r^*}, g^{r^*})\\ &= (\mathfrak{A}^*, \mathfrak{B}^*).\end{aligned}$$

To solve the CDH instance, $\mathcal{B}$ outputs $(\mathfrak{A}^*) \cdot (\mathfrak{B}^*)^{-J(m^*)} = g^{ab}$.

To conclude, we bound the probability that $\mathcal{B}$ completes the simulation without aborting. For the simulation to complete without aborting, we require that all sign and resign queries on a message m have $F(m) \not\equiv 0 \bmod p$, and that $F(m^*) \equiv 0 \bmod p$.

Let $m_1, \ldots, m_{q_Q}$ be the messages appearing in sign queries or resign queries not involving the message m^*. Clearly, $q_Q \leq q_S + q_{RS}$. We define the events E_i, E_i', and E^* as:

$$E_i : F(m_i) \not\equiv 0 \bmod p,\ E_i' : F(m_i) \not\equiv 0 \bmod l_m,\ E^* : F(m^*) \equiv 0 \bmod p.$$

The probability of $\mathcal{B}$ not aborting is $\Pr[\neg abort] \geq \Pr[\bigwedge_{i=1}^{q_Q} E_i \wedge E^* \wedge E]$. It is easy to see that the events $(\bigwedge_{i=1}^{q_Q} E_i)$, E^*, and E are independent, and $\Pr[E] = 1/q_K$.

From $l_m(n_m + 1) < p$ and x' and $x_i (i = 1, \ldots, n_m)$ are all from $\mathbb{Z}_{l_m}$, we have $0 \leq l_m k_m < p$ and $0 \leq x' + \prod_{i \in \mathcal{U}} x_i < p$. Then it is easy to see that $F(m) \equiv 0 \bmod p$ implies that $F(m) \equiv 0 \bmod l_m$. We can get that $F(m) \not\equiv 0 \bmod l_m$ implies that $F(m) \not\equiv 0 \bmod p$. Hence, we have: $\Pr[E_i] \geq \Pr[E_i']$,

$$\begin{aligned}&\Pr[E^*]\\ =\ &\Pr[F(m^*) \equiv 0 \bmod p \wedge F(m^*) \equiv 0 \bmod l_m]\\ =\ &\Pr[F(m^*) \equiv 0 \bmod l_m]\\ &\Pr[F(m^*) \equiv 0 \bmod p | F(m^*) \equiv 0 \bmod l_m]\\ =\ &\tfrac{1}{l_m} \tfrac{1}{n_m+1}\end{aligned}$$

and

$$\begin{aligned}\Pr[\textstyle\bigwedge_{i=1}^{q_Q} E_i] &\geq \Pr[\textstyle\bigwedge_{i=1}^{q_Q} E_i']\\ &= 1 - \Pr[\textstyle\bigvee_{i=1}^{q_Q} \neg E_i']\\ &\geq 1 - \textstyle\sum_{i=1}^{q_Q} \Pr[\neg E_i']\\ &= 1 - \tfrac{q_Q}{l_m}\\ &\geq 1 - \tfrac{q_S + q_{RS}}{l_m}.\end{aligned}$$

and $l_m = 2(q_S + q_{RS})$ as in the simulation.

Hence, we get that

$$\begin{aligned}&\Pr[\neg abort]\\ \geq\ &\Pr[\textstyle\bigwedge_{i=1}^{q_Q} E_i]\Pr[E^*]\\ \geq\ &\tfrac{1}{l_m(n_m+1)} \cdot (1 - \tfrac{q_S+q_{RS}}{l_m})\\ \geq\ &\tfrac{1}{2(q_S+q_{RS})(n_m+1)} \cdot \tfrac{1}{2}\\ =\ &\tfrac{1}{4(q_S+q_{RS})(n_m+1)}\end{aligned}$$

Since there are $O(n_m)$ and $O(n_m)$ multiplications in sign queries and resign queries, respectively, and $O(1)$, $O(1)$, and $O(1)$ exponentiations in sign queries, resign queries and (un)corrupted key queries, respectively, hence the time complexity of $\mathcal{B}$ is $t + O((q_S + q_{RS})n_m\rho + (q_S + q_{RS} + q_K)\tau)$.

Thus, the theorem follows. ■

Discussion of Scheme S_{mb}: This scheme is transparent, since the signature from `Sign` algorithm is the same of that from `ReSign` algorithm. This fact also implies that this scheme is multi-use. Furthermore, it is easy to see that $rk_{A \to B} = 1/rk_{B \to A}$, which shows the scheme is bidirectional. Last, since each user just stores one signing key, the scheme is also key optimal.

3.2 S_{id-mb}: $\mathcal{ID}$-based $\mathcal{M}$ulti-Use $\mathcal{B}$idirectional Scheme

In this subsection, we will extend S_{mb} to an ID-based multi-use bidirectional scheme, denoted as S_{id-mb}. The scheme is consisted of six algorithms. In the following we assume that all identities and messages are n_{id}-bit and n_m-bit strings, respectively. We can achieve this by applying two collision-resistant hash functions, $H_{id} : \{0,1\}^* \to \{0,1\}^{n_{id}}$, and $H_m : \{0,1\}^* \to \{0,1\}^{n_m}$.

`Setup`: On input the security parameter 1^k, it chooses groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of prime order $p = \Theta(2^k)$, such that an admissible pairing $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ can be constructed and pick a generator g of $\mathbb{G}_1$. Furthermore, choose a random number α from $\mathbb{Z}_p$, compute $g_1 = g^\alpha$, and then choose u', u_i $(i = 1, \cdots, n_{id})$, v', and v_i $(i = 1, \cdots, n_m)$ from $\mathbb{G}_1$.

The public parameters are $(\mathbb{G}_1, \mathbb{G}_2, e, g, g_1, g_2, u', u_i(i = 1, \cdots, n_{id}), v', v_i(i = 1, \cdots, n_m))$ and the master secret key is α.

`Extract`: On input an n_{id}-bit identity ID, output the corresponding private key d_{id},

$$d_{id} = (d_{id}^{(1)}, d_{id}^{(2)}) = (g_2^\alpha (u' \prod_{i \in \mathcal{U}} u_i)^{r_{id}}, g^{r_{id}}),$$

where r_{id} is a random number from $\mathbb{Z}_p$, $\mathcal{U} \subset \{1, \cdots, n_{id}\}$ is the set of indices i such that $u[i] = 1$, and $u[i]$ is the i-th bit of ID.

`Rekey`: On input two private keys $d_A = (d_A^{(1)}, d_A^{(2)})$ and $d_B = (d_B^{(1)}, d_B^{(2)})$, output the re-signature key

$$rk_{A \to B} = \frac{d_B}{d_A} = (\frac{d_B^{(1)}}{d_A^{(1)}}, \frac{d_B^{(2)}}{d_A^{(2)}}).$$

(Note that we make use of the same method and assumptions in [2] to get the re-signature key.)

`Sign`: On input a private key $d_{id} = (d_{id}^{(1)}, d_{id}^{(2)})$ and a n_m-bit message m, output

$$\sigma = (\mathfrak{A}, \mathfrak{B}, \mathfrak{C}) = (d_{id}^{(1)} (v' \prod_{i \in \mathcal{V}} v_i)^{r_m}, d_{id}^{(2)}, g^{r_m}),$$

where r_m is a random number from $\mathbb{Z}_p$, $\mathcal{V} \subset \{1, \cdots, n_m\}$ is the set of indicies i such that $m[i] = 1$, and $m[i]$ is the i-th bit of m.

`ReSign`: On input a re-signature key $rk_{A \to B} = (\frac{d_B^{(1)}}{d_A^{(1)}}, \frac{d_B^{(2)}}{d_A^{(2)}})$, an n_{id}-bit identity ID_A, a signature σ_A, and an n_m-bit message, check that $\texttt{Verify}(ID_A, m, \sigma_A) = 1$. If $\sigma_A = (\mathfrak{A}_A, \mathfrak{B}_A, \mathfrak{C}_A)$ does not verify, output $\perp$; otherwise, output

$$\begin{aligned} \sigma_B &= (\mathfrak{A}_A \cdot \frac{d_B^{(1)}}{d_A^{(1)}} \cdot (v' \textstyle\prod_{i \in \mathcal{V}} v_i)^{\Delta r}, \mathfrak{B}_A \frac{d_B^{(2)}}{d_A^{(2)}}, \mathfrak{C}_A \cdot g^{\Delta r}) \\ &= (d_B^{(1)} (v' \textstyle\prod_{i \in \mathcal{V}} v_i)^{r_m + \Delta r}, d_B^{(2)}, g^{r_m + \Delta r}), \end{aligned}$$

where Δr is a random number from $\mathbb{Z}_p$.

Verify: On input an n_{id}-bit identity ID, an n_m-bit message m, and a purported signature $\sigma = (\mathfrak{A}, \mathfrak{B}, \mathfrak{C})$, output 1, if $e(\mathfrak{A}, g) = e(g_2, g_1) e(u' \prod_{i \in \mathcal{U}} u_i, \mathfrak{B}) e(v' \prod_{i \in \mathcal{V}} v_i, \mathfrak{C})$ and 0 otherwise.

Theorem 2 (Security of S_{id-mb}) *In the standard model, ID-based bidirectional proxy re-signature scheme S_{mb} is correct and existentially unforgeable under the Computational Diffie-Hellman (CDH) assumption in $\mathbb{G}_1$; that is, for random $g \in \mathbb{G}_1$, and $x, y \in \mathbb{Z}_p^*$, give (g, g^x, g^y), it is hard to compute g^{xy}.*

Proof. Firstly, we use the following equations to show S_{id-mb}'s correctness.

$$\begin{array}{rcl} e(\mathfrak{A}, g) & = & e(d^{(1)(v' \prod_{i \in \mathcal{V}} v_i)^{r_m}}, g) \\ & = & e(g_2^{\alpha}(u' \prod_{i \in \mathcal{U}} u_i)^{r_{id}} (v' \prod_{i \in \mathcal{V}} v_i)^{r_m}, g) \\ & = & e(g_2^{\alpha}, g) e((u' \prod_{i \in \mathcal{U}} u_i)^{r_{id}}, g) e((v' \prod_{i \in \mathcal{V}} v_i)^{r_m}, g) \\ & = & e(g_2, g_1) e(u' \prod_{i \in \mathcal{U}} u_i, \mathfrak{B}) e(v' \prod_{i \in \mathcal{V}} v_i, \mathfrak{C}) \end{array}$$

And then we show security as in Theorem 1, the approach is also based on that of [19, 15].

We show if there exists any adversary $\mathcal{A}$ that can break the external security of the above proxy re-signature scheme with non-negligible probability ε in time t after making at most q_E extract queries, q_S sign queries, and q_{RS} resign queries, there must exist an adversary $\mathcal{B}$ that solves the CDH problem in $\mathbb{G}_1$ with probability $\frac{1}{16(q_E+q_S+q_{RS}+2q_{RK})(q_S+q_{RS})(n_{id+1})(n_m+1)}$ in time $t + O(((q_E + q_{RK})n_{id} + (q_S + q_{RS})(n_{id} + n_m))\rho + (q_E + q_S + q_{RS} + q_{RK})\tau)$, where ρ and τ are the time for a multiplication and an exponentiation in $\mathbb{G}_1$, respectively.

On input (g, g^a, g^b), the CDH adversary $\mathcal{B}$ simulates a proxy re-signature security game for $\mathcal{A}$ as follows:

To prepare the simulation, $\mathcal{B}$ first sets $l_{id} = 2(q_E + q_S + q_{RS} + 2q_{RK})$ and $l_m = 2(q_S + q_{RS})$, and randomly chooses two numbers k_{id} and k_m, such that $0 \leq k_{id} \leq n_{id}$, $l_{id}(n_{id} + 1) < p$, $0 \leq k_m \leq n_m$, and $l_m(n_m + 1) < p$. $\mathcal{B}$ then chooses $n_{id} + 1$ random numbers x', $x_i(i = 1, \cdots, n_{id})$ from $\mathbb{Z}_{l_{id}}$, and $n_m + 1$ random numbers z', $z_i(i = 1, \cdots, n_m)$ from $\mathbb{Z}_m$. Lastly, $\mathcal{B}$ chooses $n_{id} + n_m + 2$ random numbers y', $y_i(i = 1, \cdots, n_{id})$, w', $w_i(i = 1, \cdots, n_m)$ from $\mathbb{Z}_p$.

To make expression simpler, we use the following notations:

$$F(ID) = x' + \sum_{i \in \mathcal{U}} x_i - l_{id}k_{id} \quad \text{and} \quad J(ID) = y' + \sum_{i \in \mathcal{U}} y_i,$$

$$K(m) = z' + \sum_{i \in \mathcal{V}} z_i - l_m k_m \quad \text{and} \quad L(m) = w' + \sum_{i \in \mathcal{V}} w_i.$$

Now, $\mathcal{B}$ sets the public parameters:

$$g_1 = g^a, \ g_2 = g^b,$$
$$u' = g_2^{x' - l_{id}k_{id}} g^{y'}, \ u_i = g_2^{x_i} g^{y_i} (1 \leq i \leq n_{id})$$
$$v' = g_2^{z' - l_m k_m} g^{w'}, \ v_i = g_2^{z_i} g^{w_i} (1 \leq i \leq n_m).$$

Note that for any identity ID and message m, there exists the following equations:

$$u' \prod_{i \in \mathcal{U}} u_i = g_2^{F(ID)} g^{J(ID)} \quad \text{and} \quad v' \prod_{i \in \mathcal{V}} v_i = g_2^{F(m)} g^{J(m)}.$$

Queries: $\mathcal{B}$ builds the following oracles:

$\mathcal{O}_{Extract}$: On input ID, if ID is uncorrupted, then this input is illegal; else, $\mathcal{B}$ computes $F(ID)$. If $F(ID) \equiv 0 \bmod p$, $\mathcal{B}$ aborts; otherwise, $\mathcal{B}$ computes the corresponding private key:

$$\begin{array}{rcl} d_{ID} & = & (d_{ID}^{(1)}, d_{ID}^{(2)}) \\ & = & (g_1^{\frac{-J(ID)}{F(ID)}} (u' \prod_{i \in \mathcal{U}} u_i)^{r_{id}}, g_1^{\frac{-1}{F(ID)}} g^{r_{id}}), \end{array}$$

where r_{id} is a random number from $\mathbb{Z}_p$.

Writing $\tilde{r}_{id} = r_{id} - a/F(ID)$, we have

$$\begin{array}{rcl} d_{id}^{(1)} & = & g_1^{-J(ID)/F(ID)}(u' \prod_{i \in \mathcal{U}} u_i)^{r_{id}} \\ & = & g_1^{-J(ID)/F(ID)}(g_2^{F(ID)} g^{J(ID)})^{r_id} \\ & = & g_2^a (g_2^{F(ID)} g^{J(ID)})^{-a/F(ID)} (g_2^{F(ID)} g^{J(ID)})^{r_{id}} \\ & = & g_2^a (g_2^{F(ID)} g^{J(ID)})^{r_{id} - a/F(ID)} \\ & = & g_2^a (u' \prod_{i \in \mathcal{U}} u_i)^{\tilde{r}_{id}}, \end{array}$$

and

$$d_{ID}^{(2)} = g_1^{-/F(ID)} g^{r_{id}} = g^{r_{id} - a/F(ID)} = g^{\tilde{r}_{id}}.$$

Hence, from the adversary's point of view, all private keys computed by $\mathcal{B}$ will be indistinguishable from the keys generated by the real PKG.

$\mathcal{O}_{Sign}$**:** On input (ID, m), $\mathcal{B}$ first computes $F(ID)$.

- If $F(ID) \not\equiv 0 \bmod p$, $\mathcal{B}$ can just compute the private key corresponding to identity ID as in an extract query, and then use the `Sign` algorithm to create a signature on m.
- If $F(ID) \equiv 0 \bmod p$, $\mathcal{B}$ computes $K(m)$, if $K(m) \equiv 0 \bmod p$ $\mathcal{B}$ aborts; otherwise, $\mathcal{B}$ creates a signature on m $\sigma = (\mathfrak{A}, \mathfrak{B}, \mathfrak{C})$.

$$\begin{array}{rcl} \mathfrak{A} & = & (u' \prod_{i \in \mathcal{U}} u_i)^{r_{id}} g_1^{\frac{-L(m)}{K(m)}} (v' \prod_{i \in \mathcal{V}} v_i)^{r_m} \\ & = & g_2^a (u' \prod_{i \in \mathcal{U}} u_i)^{r_{id}} (v' \prod_{i \in \mathcal{V}} v_i)^{\tilde{r}_m}, \\ & & \\ \mathfrak{B} & = & g^{r_{id}}, \\ & & \\ \mathfrak{C} & = & g_1^{\frac{-1}{K(m)}} g^{r_m} \\ & = & g^{\tilde{r}_m}, \end{array}$$

where r_{id} and r_m are random numbers from $\mathbb{Z}_p$, and $\tilde{r}_m = r_m - a/K(m)$. The last equation shows that the signatures computed by $\mathcal{B}$ are indistinguishable to that generated by the real user, from $\mathcal{A}$'s point of view.

$\mathcal{O}_{ReKey}$**:** On input (ID_i, ID_j), if one of ID_i and ID_j is corrupted, and the other is uncorrupted, then the input is illegal; else, $\mathcal{B}$ does: if $F(ID_i) \equiv 0 \bmod p$ or $F(ID_i) \equiv 0 \bmod p$, abort; else, return $rk_{i \to j} = \texttt{ReKey}(\mathcal{O}_{Extract}(ID_i), \mathcal{O}_{Extract}(ID_j))$ (via calling oracle $\mathcal{O}_{Extract}$).

$\mathcal{O}_{ReSign}$**:** On input (ID_i, ID_j, m, σ). If $\texttt{Verify}(ID_i, m, \sigma) \neq 1$, $\mathcal{B}$ outputs $\perp$. Otherwise, $\mathcal{B}$ does:

- If ID_i and ID_j are both corrupted or uncorrupted, output $\texttt{ReSign}(\mathcal{O}_{ReKey}(ID_i, ID_j), ID_i, m, \sigma)$.
- else, output $\mathcal{O}_{Sign}(ID_j, m)$.

Forgery: If $\mathcal{B}$ does not abort as a consequence of any queries above, $\mathcal{A}$ will, with probability at least ε, return a message m^*, an identity ID^*, and a valid forgery $\sigma^* = (A^*, B^*)$ of ID^* on m^*. If $F(ID^*) \not\equiv 0 \bmod p$ or $K(m^*) \not\equiv 0 \bmod p$, $\mathcal{B}$ aborts. Otherwise, the forgery must be of the form, for some $r_{id}^*, r_m^* \in \mathbb{Z}_p$,

$$\begin{array}{rcl} \sigma^* & = & (g^{ab}(u' \prod_{i \in \mathcal{U}} u_i)^{r_{id}^*} (v' \prod_{i \in \mathcal{V}} v_i)^{r_m^*}, g^{r_{id}^*}, g^{r_m^*}) \\ & = & (g^{ab}(g_2^{F(ID^*)} g^{J(ID^*)})^{r_{id}^*} (g_2^{K(m^*)} g^{L(m^*)})^{r_m^*}, g^{r_{id}^*}, g^{r_m^*}) \\ & = & (g^{ab + J(ID^*) r_{id}^* + K(m^*) r_m^*}, g^{r_{id}^*}, g^{r_m^*}) \\ & = & (\mathfrak{A}^*, \mathfrak{B}^*, \mathfrak{C}^*). \end{array}$$

To solve the CDH instance, $\mathcal{B}$ outputs $(\mathfrak{A}^*) \cdot (\mathfrak{B}^*)^{-J(ID^*)} \cdot (\mathfrak{C}^*)^{-L(m^*)} = g^{ab}$.

To conclude, we bound the probability that $\mathcal{B}$ completes the simulation without aborting. For the simulation to complete without aborting, we require that all extract queries on an identity ID have $F(ID) \not\equiv 0 \bmod p$, that all sign and resign queries on a message (ID, m) have $F(ID) \not\equiv 0 \bmod p$ or $K(m) \not\equiv 0 \bmod p$, that all rekey queries on identity pair (ID_i, ID_j) have $F(ID_i) \not\equiv 0 \bmod p$ and $F(ID_j) \not\equiv 0 \bmod p$, and that $F(ID^*) \equiv 0 \bmod p$ and $K(m^*) \equiv 0 \bmod p$.

Let $ID_1, \cdots, ID_{q_{ID}}$ be the identities appearing in extract queries, sign queries or resign queries not involving the identity ID^*, and $m_1, \cdots, m_{q_M}$ be the messages appearing in sign queries or resign queries involving the identity ID^*. Clearly, $q_{ID} \leq q_E + q_S + q_{RS} + 2q_{RK}$ and $q_M \leq q_S + q_{RS}$. We define the events E_i^F, E'^F_i, E_F^*, E_i^K, E'^K_i, and E_K^* as:

$$E_i^F : F(ID_i) \not\equiv 0 \bmod p,\ E'^F_i : F(ID_i) \not\equiv 0 \bmod l_{id},\ E_F^* : F(ID^*) \equiv 0 \bmod p,$$

$$E_i^K : K(m_i) \not\equiv 0 \bmod p,\ E'^K_i : F(m_i) \not\equiv 0 \bmod l_m,\ E_K^* : K(m^*) \equiv 0 \bmod p.$$

The probability of $\mathcal{B}$ not aborting is

$$\Pr[\neg abort] \geq \Pr[\bigwedge_{i=1}^{q_{ID}} E_i^F \wedge E_F^* \wedge \bigwedge_{i=1}^{q_M} E_i^K \wedge E_K^*].$$

It is easy to see that the events $(\bigwedge_{i=1}^{q_{ID}} E_i^F)$, E_F^*, $(\bigwedge_{i=1}^{q_M} E_i^K)$, and E_K^* are independent.

From $l_{id}(n_{id}+1) < p$ and x' and $x_i (i = 1, \cdots, n_{id})$ are all from $\mathbb{Z}_{l_{id}}$, we have $0 \leq l_{id}k_{id} < p$ and $0 \leq x' + \prod_{i \in \mathcal{U}} x_i < p$. Then it is easy to see that $F(ID) \equiv 0 \bmod p$ implies that $F(ID) \equiv 0 \bmod l_{id}$. We can get that $F(ID) \not\equiv 0 \bmod l_{id}$ implies that $F(m) \not\equiv 0 \bmod p$. Hence, we have: $\Pr[E_i^F] \geq \Pr[E'^F_i]$,

$$\begin{array}{rcl} \Pr[E_F^*] & = & \Pr[F(ID^*) \equiv 0 \bmod p \wedge F(ID^*) \equiv 0 \bmod l_{id}] \\ & = & \Pr[F(ID^*) \equiv 0 \bmod l_{id}] \\ & & \Pr[F(ID^*) \equiv 0 \bmod p | F(ID^*) \equiv 0 \bmod l_{id}] \\ & = & \frac{1}{l_{id}} \frac{1}{n_{id}+1} \\ & = & \frac{1}{2(q_E+q_S+q_{RS}+2q_{RK})} \frac{1}{n_{id}+1} (\text{since } l_{id} = 2(q_E + q_S + q_{RS} + 2q_{RK})) \end{array}$$

and

$$\begin{array}{rcl} \Pr[\bigwedge_{i=1}^{q_{ID}} E_i^F] & \geq & \Pr[\bigwedge_{i=1}^{q_{ID}} E'^F_i] \\ & = & 1 - \Pr[\bigvee_{i=1}^{q_{ID}} \neg E'^F_i] \\ & \geq & 1 - \sum_{i=1}^{q_{ID}} \Pr[\neg E'^F_i] \\ & = & 1 - \frac{q_{ID}}{l_{id}} \\ & \geq & 1 - \frac{q_E+q_S+q_{RS}+2q_{RK}}{l_{id}} \\ & = & 1/2 \ \ (\text{ since } l_{id} = 2(q_E + q_S + q_{RS} + 2q_{RK})). \end{array}$$

Similarly, we get that

$$\Pr[E_K^*] \geq \frac{1}{2(q_S + q_{RS})} \frac{1}{n_m + 1} \quad \text{and} \quad \Pr[\bigwedge_{i=1}^{q_M} E_i^K] \geq 1/2.$$

Hence, we get that

$$\begin{array}{rcl} \Pr[\neg abort] & \geq & \Pr[\bigwedge_{i=1}^{q_Q} E_i^F \wedge E_F^* \wedge \bigwedge_{i=1}^{q_M} E_i^K \wedge E_K^*] \\ & \geq & \frac{1}{16(q_E+q_S+q_{RS}+2q_{RK})(q_S+q_{RS})(n_{id+1})(n_m+1)}. \end{array}$$

Since there are $O(n_{id})$, $O(n_{id})$, $O(n_{id}+n_m)$ and $O(n_{id}+n_m)$ multiplications in extract queries, rekey queries, sign queries and resign queries, respectively, and $O(1)$, $O(1)$, $O(1)$, and $O(1)$ exponentiations in extract queries, rekey queries, sign queries and resign queries, respectively, hence the time complexity of $\mathcal{B}$ is $t + O(((q_E + q_{RK})n_{id} + (q_S + q_{RS})(n_{id} + n_m))\rho + (q_E + q_S + q_{RS} + q_{RK})\tau)$.

Thus, the theorem follows. ■

Discussion of Scheme S_{id-mb}: As S_{mb}, S_{id-mb} is bidirectional, multi-use, transparent, and key optimal.

4 Conclusions

We have presented the first two proxy re-signature schemes which are proven secure in the standard model. Especially, the second one is an ID-based proxy re-signature scheme. Both of them are computational efficient, only two exponentiations in $\mathbb{G}_1$ in `Sign` and `ReSign` algorithms. However, their public parameters' size is relatively large. We can make a tradeoff between the public parameters' size and the security reduction by using the techniques of Naccache [14] and Sarkar and Chatterjee [16] to reduce its size. Note that, our proposals are only proven secure with static corruption not the adaptive corruption, we left it as the future work.

References

[1] J.H. An, Y. Dodis, and T. Rabin. On the Security of Joint Signature and Encrption. In: *EUROCRPYT 2002*, LNCS 2332, pp. 83-107, 2002. 2

[2] G. Ateniese and S. Hohenberger. Proxy Re-Signatures: New Definitions, Algorithms, and Applications. In: *ACM CCS 2005*, pp. 310-319, 2005. (document), 1, 1, 2.1, 2.1, 3.1, 3.2

[3] D. Boneh and X. Boyen. Short signatures without random oracles. In: *EUROCRYPT 2004*, LNCS 3027, pp. 56-73, 2004. 1

[4] D. Boneh and X. Boyen. Secure Identity Based Encryption Without Random Oracles. In: *CRYPTO 2004*, LNCS 3027, pp. 443-459, 2004. 1

[5] M. Bellare, A. Boldyreva, and A. Palacio. An uninstantiable random-oracle-model scheme for a hybrid-encryption problem. In: *EUROCRYPT 2004*, LNCS 3027, pp. 171-188, 2004. 1

[6] M. Blaze, G. Bleumer, and M. Strauss. "Divertible protocols and atomic proxy cryptography", In: *EUROCRYPT 1998*, LNCS 1403, pp. 127-144, 1998. (document), 1, 1

[7] D. Boneh and M. Franklin. Identity-based encryption from the Weil pairing. SIAM Journal of Computing. vol. 32, no. 3, 2003, pp. 586-615. 2.3

[8] R. Canetti and S. Hohenberger. Chosen-ciphertext secure proxy re-encryption. 2007. Cryptology ePrint Archieve: Report 2007/171. 2.1

[9] R. Canetti, O. Goldreich, and S. Halevi. The random oracle methodology, revisited. In: *STOC 1998*, pp. 209-218, 1998. 1

[10] R. Cramer and V. Shoup. Signature schemes based on the strong RSA assumption. *ACM TISSEC*, vol. 3, no. 3, 2000, pp. 161-185. 1

[11] D. Galindo, J. Herranz and E. Kiltz, "On the Generic Construction of Identity-Based Signatures with Additional Properties", In *ASIACRYPT 2006*, LNCS 4284, pp. 178-193, 2006. 1

[12] R. Gennaro, S. Halevi, and T. Rabin. Secure hash-and-sign signatures without the random oracle. In: *EUROCRYPT 1999*, LNCS 1592, pp. 123-139, 1999. 1

[13] S. Goldwasser, S. Micali, and R.L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal of Computing*, vol. 17, no. 2, 1988, pp. 281-308. 1

[14] D. Naccache. Secure and Practical Identity-based encryption. Cryptology ePrint Archive, Report 2005/369, 2005. http://eprint.iacr.org/. 1.1, 4

[15] K.G. Paterson and J.C.N. Schuldt. Efficient Identity-based Signatures Secure in the Standard Model. In: *ACISP 2006*, LNCS 4058, pp. 207-222, 2006. 1, 3.1, 3.2

[16] P. Sarkar and S. Chatterjee. Trading time for space: Towards an efficient IBE scheme with short(er) public parameters in the standard model. In: *ICISC 2005*, LNCS 3935, pp. 424-440, 2006. 1.1, 4

[17] A. Shamir. Identity-based cryptosystems and signature schemes", In: *Crypto 1984*, LNCS 196, Springer-Verlag, pp. 47-53, 1984. 1, 2

[18] G. Taban, A.A. Cárdenas and V.D. Gligor. Towards a Secure and Interoperable DRM Architecture. In: *ACM DRM 2006*, pp. 69-78, 2006. 1

[19] B. Waters. Efficient Identity-based Encryption Without Random Oracles. In: *EUROCRYPT 2005*, LNCS 3494, pp. 114-127, 2005. (document), 1, 1.1, 3.1, 3.1, 3.2

[20] F. Zhang, X. Chen, W. Susilo, and Y. Mu. A New Short Signature Scheme without Random Oracles from Bilinear Pairings. In: *VIETCRYPT 2006*, LNCS 4341, pp. 67-80, 2006. 1