Remark on Shao et al's Bidirectional Proxy Re-Signature Scheme In Indocrypt'07

Kitae Kim, Ikkwon Yie, and Seongan Lim
(Corresponding author: Kitae Kim)

Department of Mathematics, Inha University
Incheon 402-751, Republic of Korea (Email:ktkim@inha.ac.kr)
(Received Feb. 15, 2008; revised and accepted Apr. 9, 2008)

Abstract

Recently, Shao et al. proposed two bidirectional proxy re-signature schemes S_{mb} and S_{id-mb} [3]. In their paper, the authors gave security proofs to say that both of them are secure in their security model without random oracles. But, we found that the scheme S_{mb} is miss leaded and its security proof is false. In this paper, we present an attack on S_{mb} and improve it to be secure in their security model.

Keywords: Bidirectional proxy, bilinear pairing, re-signature proxy signature

1 Introduction

The primitive of proxy re-signature was introduced by Blaze, Bleumer, and Strauss [2], and formalized by Ateniese and Hohenberger [1]. In a proxy re-signature scheme, a semi-trust proxy is allowed to transform a delegatee's signatures on messages to a delegator's signatures on the same messages. But the proxy cannot generate signatures for either the delegatee or the delegator. According to [1], there are several properties related to proxy re-signature scheme. Of course, the properties which are needed may depend on specific applications.

Unidirectional: In an unidirectional scheme, a re-signature key allows the proxy to transform a delegatee's signature to a delegator's but not a delegator's to delegatee's. Schemes that do not have this property are called bidirectional.

Multi-use: In a multi-use scheme, signatures generated by either the Sign or Re-Sign Algorithms can be taken as input to ReSign. If only signatures generated by Sign can be inputs to Re-Sign then the scheme is called single-use scheme.

Private Proxy: In a private proxy scheme, the re-signature keys can be kept secret by an honest proxy. In public proxy scheme, the re-signature key can be computed by an adversary passively observing the proxy.

Transparent: In transparent scheme, the proxy is transparent. That is, the signatures generated by delegator's signature on a message m using the Sign Algorithm is computationally indistinguishable from his/her signatures on m generated by the proxy as the output of Re-Sign.

Key Optimal: In key optimal scheme, a user is required to protect and store a small constant amount of secret data regardless of how many signature delegation she gives or accepts.

Non-interactive: The delegatee is not required to participate in a delegation process.

Non-transitive: The proxy alone cannot re-delegate signing rights in non-transitive scheme.

Temporary: In temporary scheme, there is the chance that a delegator will need or want to revoke re-signing rights later on.

The proxy re-signature scheme proposed by Blaze, Bleumer, and Strauss is bidirectional, multi-use, public proxy, transparent, and key optimal [2]. The scheme was proved to be secure (in their security Definition) in the random oracle model, but is inefficient as pointed in [1]. Later, Atenesse and Hohenberger [1] proposed two proxy re-signature schemes, one of them is bidirectional, multi-use, and the other is unidirectional, single-use. Both of them are strongly unforgeable in the random oracle model. Recently, Shao et al. [3] proposed two bidirectional proxy re-signature schemes - one of them S_{mb} is non-identity based proxy re-signature scheme and the other S_{id-mb} is identity based proxy re-signature scheme. Shao et al's schemes is proved to be existentially unforgeable without random oracles. However, we find that the original construction of S_{mb} is somewhat miss leaded. In this paper, we give successful attack against the proxy re-signature Algorithm S_{mb}, and suggest a modification to repair our attack.

The rest of the paper is organized as follows. In Section 2, we review the Definitions of proxy re-signatures and their security as in [3]. In Section 3, we briefly present Shao et al's proxy re-signature scheme, analyze their scheme with a concrete attack, and then improve it to be secure in their security model.

2 Preliminaries

In this Section, we review the Definition of bidirectional proxy re-signature scheme and the security Definition of it. All the notions in this Section are following [3], and the security notion is for existential unforgeability and static corruption. We refer the reader to [3] for details.

2.1 Definition of Bidirectional Proxy Re-Signature

Definition 1. *A bidirectional proxy re-signature scheme consists of five probabilistic polynomial time Algorithms* KeyGen*,* ReKey*,* Sign*,* ReSign*,* Verify*:*

- KeyGen*,* Sign*,* Verify *form the standard digital signatures - key generation, signing, and verification Algorithms.*
- *On input (sk_A, sk_B), the re-signature key generation Algorithm,* ReKey*, outputs a key $rk_{A\leftrightarrow B}$ for the proxy. (where sk_A and sk_B are the private keys of A and B, respectively.)*
- *On input $(rk_{A\leftrightarrow B}, pk_A, m, \sigma)$, the re-signature Algorithm,* ReSign*, outputs a new signature (pk_B, σ', m) if $\mathsf{Verify}(pk_A, m, \sigma) = 1$ and $\perp$ otherwise. (where pk_A and pk_B are the public keys of A and B, respectively. And σ is a signature on message m corresponding to pk_A.)*

Correctness. The correctness property is that all signatures validly formed by either the signing or re-signing Algorithms will pass verification: For any message m in the message space and any key pairs $(pk, sk), (pk', sk') \leftarrow \mathsf{KeyGen}(1^k)$, let $\sigma = \mathsf{Sign}(sk, m)$ and $rk \leftarrow \mathsf{ReKey}(sk, sk')$. Then the following two conditions must hold:

$$\begin{aligned} \mathsf{Verify}(pk, m, \sigma) &= 1 \quad \text{and} \\ \mathsf{Verify}(pk', m, \mathsf{ReSign}(rk, pk, m, \sigma)) &= 1. \end{aligned}$$

2.2 Security Definition of Bidirectional Proxy Re-Signature

In [3], the authors define security for bidirectional proxy re-signature schemes. Their security model protects users from static corruption, i.e., in the 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. We now review Shao et al's security Definition of bidirectional proxy re-signature schemes.

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 $(pk, sk) \leftarrow \mathsf{KeyGen}(1^k)$. The adversary is given pk.
- Corrupted key generation $\mathcal{O}_{CKeyGen}$: Obtain a new key pair as $(pk, sk) \leftarrow \mathsf{KeyGen}(1^k)$. The adversary is given pk and sk.
- Re-Signature key generation $\mathcal{O}_{\mathsf{ReKey}}$: On input (pk, pk') by the adversary, where the public keys are generated before by KeyGen, return the re-signature key $rk_{pk\leftrightarrow pk'} = \mathsf{ReKey}(sk, sk')$, where sk and sk' are the secret keys that correspond to pk and pk', respectively. Here, we require that both pk and pk' are corrupted, or both are uncorrupted.
- Sign $\mathcal{O}_{Sign}$: On input pk, m, where pk was generated before by KeyGen. The adversary is given the corresponding signature $\sigma = \mathsf{Sign}(sk, m)$, where sk is the secret key that correspond to pk.
- Re-Sign $\mathcal{O}_{ReSign}$: On input (pk, pk', m, σ), where pk, pk' were generated before by KeyGen. The adversary is given the re-signed signature $\sigma' = \mathsf{ReSign}(\mathsf{ReKey}(sk, sk'), pk, m, \sigma)$, where sk, sk' are the secret keys that correspond to pk, pk'.

Forgery. The adversary outputs (m^*, pk^*, σ^*). The adversary succeeds if the following holds

1) Verify(pk^*, m^*, σ^*)=1.
2) pk^* is not from Corrupted key Generation $\mathcal{O}_{CKeyGen}$.
3) (pk^*, m^*) is not a query to Sign Query $\mathcal{O}_{Sign}$.
4) $(\lozenge, pk^*, m^*, \blacklozenge)$ is not a query to Re-Sign Query $\mathcal{O}_{ReSign}$, where $\lozenge$ and $\blacklozenge$ denote any public key and any signature, respectively.

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.

3 Overview of Shao et al's Bidirectional Proxy Re-Signature Scheme S_{mb}

Shao et al. [3] proposed two bidirectional proxy re-signature schemes, one for non-identity based systems, the other for identity based ones. The authors gave proofs that their schemes are secure in the security Definition presented in the above Section (Section 2.2). However, we found that the non-identity based scheme S_{mb} is not secure. In this Section, we describe the Shao et al's proxy re-signature scheme S_{mb}.

We assume that messages are bit strings of n_m bits, which can be achieved by a collision resistant hash function $H : \{0,1\}^* \rightarrow \{0,1\}^{n_m}$.

Setup: On input the security parameter 1^k, this Algorithm chooses two groups G_1 and G_2 of prime order $p = \Theta(2^k)$, such that an admissible pairing $e : G_1 \times G_1 \rightarrow G_2$ can be constructed, and chooses a generator g of G_1. Then it selects $n_m + 2$ random elements $(h, u', u_1, \ldots, u_{n_m}) \in G_1^{n_m+2}$, and publishes the public parameters $(G_1, G_2, e, h, u', u_1, \ldots, u_{n_m})$.

KeyGen: For each user A, this Algorithm chooses $a \in \mathbb{Z}_p$ at random, and then outputs $(pk, sk) = (g^a, a)$.

ReKey: On input $(sk_A, sk_B) = (a, b)$, this Algorithm outputs $rk_{A\rightarrow B} = b/a \pmod p$.

Sign: On input a secret key $sk = a$ and a message $m = (m[1], \ldots, m[n])$, this Algorithm

1) Computes $U(m) = u' \prod_{i\in\mathcal{U}} u_i$, where $\mathcal{U} \subset \{1, \ldots, n\}$ s.t. $m[i] = 1$;
2) Chooses a random value $r \in Z_p$;
3) Computes $\sigma_1 = h^a \cdot U(m)^r$ and $\sigma_2 = g^r$;
4) Returns $\sigma = (\sigma_1, \sigma_2)$.

ReSign: On input a re-signature key $rk_{A\rightarrow B}$, a signature σ_A of a user A corresponding to pk_A, and a message m, this Algorithm

1) Checks that $\mathsf{Verify}(pk_A, m, \sigma_A) = 1$;
2) Computes $\sigma_{B,1} = \sigma_{A,1}^{rk_{A\rightarrow B}}$;
3) Computes $\sigma_{B,2} = \sigma_{A,2}^{rk_{A\rightarrow B}}$;
4) Returns $\sigma_B = (\sigma_{B,1}, \sigma_{B,2})$. (Note that $\sigma_B = (h^b U(m)^{r'}, g^{r'})$ with $r' = rb/a \pmod p$).

Verify: On input pk, $\sigma = (\sigma_1, \sigma_2)$, and m, this Algorithm returns 1 if $e(\sigma_1, g) = e(\sigma_2, U(m)) \cdot e(pk, h)$ and 0 otherwise.

The scheme S_{mb} is constructed using the Waters' signature scheme as mentioned in [3]. And S_{mb} satisfies the bidirectional, multi-use, transparent, and key optimal properties.

4 Cryptanalysis and Repairing of Shao et al's Proxy Re-Signature Scheme

In [3], the authors gave the security proof under the assumption that every signatures signed by a signer or by a proxy are random. However, in the Shao et al's scheme S_{mb}, not all signatures are randomly distributed, in particular, from the delegatee's point of view. That is, the S_{mb} has a deterministic re-signature generation Algorithm that is not desirable for secure proxy re-signature schemes. In the following, we give a successful attack of the Shao et al's scheme and describe how to modify in order to repair our attack.

4.1 Attack by Delegatee

Suppose that an adversary corrupts a delegatee, say A, and B is a delegator of A. We do not assume that B is a corrupted user, i.e., the adversary do not know the secret information of B and the re-signature key $rk_{A\rightarrow B}$.

Now, A chooses a random r and a message m, and proceeds as follows:

1) Sign on m as $(h^a U(m)^{ra}, g^{ra})$;
2) Query 'ReSig' and gets $(h^b U(m)^{rb}, g^{rb})$;
3) Sign on the same m: $(h^a U(m)^{2ra}, g^{2ra})$;
4) Query 'ReSig' and get $(h^b U(m)^{2rb}, g^{2rb})$;
5) Compute $\frac{h^b U(m)^{rb}}{h^b U(m)^{2rb}} \cdot (h^b U(m)^{rb}) = h^b$.

Finally, A gets h^b in Step 5, and g^b from $\left(g^{rb}\right)^{1/r}$ in Step 2. Then A alone (without proxy) can freely produce signatures $(\sigma, m') = (h^b U(m')^s, (g^b)^s)$ on behalf of B for any message m'. Note that (1) B is not a corrupted party; (2) (B, m') is not a query to $\mathcal{O}_{Sign}$; and (3) $(\diamondsuit, B, m', \blacklozenge)$ is not a query to $\mathcal{O}_{Resig}$. Therefore the adversary succeeds to attack the proxy re-signature scheme.

4.2 Repair

The authors in [3] assumed that all the signatures generated by either a signer or a proxy are indistinguishable. But, in a delegatee's point of view, the signature is not random because the re-signing performed by proxy uses the same random nonce taken by the delegatee. In this Subsection, we fix this problem by allowing proxies to re-randomize in re-signature Algorithm. In more detail, we use the same Algorithms as the Shao et al's scheme except the **ReSign** Algorithm, and replace the **ReSign** to **ReSign'** described below.

ReSign': On input $(rk_{A\rightarrow B}, \sigma_A, m)$, this Algorithm

1) Checks that $\mathsf{Verify}(pk_A, m, \sigma) = 1$;
2) Selects a random value $s \in \mathbb{Z}_p$;
3) Computes $\sigma_{B,1} = \sigma_{A,1}^{rk_{A\rightarrow B}} \cdot U(m)^s = h^b U(m)^{rb/a+s}$;
4) Computes $\sigma_{B,2} = \sigma_{A,2}^{rk_{A\rightarrow B}} \cdot g^s = g^{rb/a+s}$;
5) Returns $\sigma_B = (\sigma_{B,1}, \sigma_{B,2}) = (h^b U(m)^{r'}, g^{r'})$ with $r' = rb/a + s \pmod p$.

In our improvement above, a delegatee cannot distinguish his/her signatures and the signatures re-signed by a proxy. Furthermore, all signatures produced in the improvement are randomly distributed. We conclude that our improvement is a secure bidirectional proxy re-signature scheme, and it's security proof is exactly the same as in [3].

5 Conclusion

In this paper, we have shown an attack on the Shao et al's non-identity based proxy re-signature scheme. From the attack, their scheme S_{mb} is insecure in their security model [3]. We have also suggested an improvement by allowing proxies to re-randomize the input signatures. Our improved method is a secure proxy re-signature scheme in the standard model.

References

[1] G. Ateniese, and S. Hohenberger, "Proxy re-signatures: new definitions, algorithms, and applications," *ACM CCS 2005*, pp. 310-319, ACM Press, 2005.

[2] M. Blaze, G. Bleumer, and M. Strauss, "Divertible protocols and atomic proxy cryptography," *Eurocrypt '98*, LNCS 1403, pp. 127-144, Springer-Verlag, 1998.

[3] J. Shao, Z. Cao, L. Wang, and X. Liang, "Proxy re-signature scheme without random oracles," *Indocrypt '07*, LNCS 4859, pp. 197-209, Springer-Verlag, 2007.

Kitae Kim received the B.S. degree in Mathematics from Konyang University, and the M.S. degree in Mathematics from Inha University, Korea. He is currently a Ph.D. candidate in the same university. His research interests include information theory and group oriented cryptography.

Ikkwon Yie received the B.S. and M.S. degrees in Mathematics from the Seoul National University, Seoul, Korea, and the Ph.D. degree in Mathematics from the Purdue University. He is currently a professor of Department of Mathematics in Inha University. His main research interests include Galois theory and Digital signatures.

Seongan Lim received her B.S. degree in Mathematics from the Dongguk University, Korea, in 1985. In 1987, she received her M.S. degree in Mathematics from the Seoul National University, Korea. In 1995, she received her PhD degree in Mathematics from Purdue University, USA. She is a lecturer and research staff of Department of Mathematics in Inha University, Korea. Her current research interests include cryptography, fast computer arithmetic, computer algorithms, mathematics.