# Multi-use Unidirectional Forward-Secure Proxy Re-signature Scheme

N.R.Sunitha
Department of Computer Science & Engg.
Siddaganga Institute of Technology,
Tumkur, Karnataka, India.

B.B.Amberker
Department of Computer Science & Engg.
National Institute of Technology,
Warangal, Andhra Pradesh, India.

## Abstract

In e-banking, on many occasions, there is need to translate one person's signature to another person's signature with mutual consent. The proxy re-signature scheme proposed by Blaze, Bleumer, and Strauss (BBS) in 1998 addresses this problem. Here, a semi-trusted proxy acts as a translator between Alice and Bob to translate a signature from Alice into a signature from Bob on the same message. The proxy, however, does not learn any signing key and cannot sign arbitrary messages on behalf of either Alice or Bob. Blaze et al.s construction is bidirectional (i.e. the proxy information allows translating signatures in either direction) and multi-use (i.e. the translation of signatures can be performed in sequence and multiple times by distinct proxies). In 2005 Ateniese and Hohenberger identified the limitations of the scheme and proposed two constructions based on bilinear maps. They left as open challenges the design of multi-use unidirectional systems. Benoit Libert and Damien Vergnaud have given one solution based on bilinear groups.

We propose another solution for multi-use unidirectional proxy re-signature scheme using the property of forward-security. Our forward-secure proxy re-signature scheme which is based on the hardness of factoring translates one person's signature to another person's signature and additionally facilitates the signers as well as the proxy to guarantee the security of messages signed in the past even if their secret key is exposed today (property of forward-security). With a minor change in resigning key, we can make the scheme to behave as a multi-use bidirectional scheme. The scheme also satisfies the following properties: private proxy, transparent, unlinkable, key optimal, interactive(as banking applications need), non-transitive and temporary. Our scheme is proven to be forward secure based on the hardness of factoring.

**Keywords :** e-banking, Proxy re-signature, Proxy Signature, Forward-Security, Proxy revocation, Private proxy.

## 1 Introduction

In Eurocrypt 98, Blaze, Bleumer, and Strauss (BBS)[8] proposed proxy re-signatures, in which a semi-trusted proxy acts as a translator between Alice and Bob. To translate, the proxy converts a signature from Alice into a signature from Bob on the same message. The proxy, however, does not learn any signing key and cannot sign arbitrary messages on behalf of either Alice or Bob. Since the BBS proposal, the proxy re-signature primitive has been largely ignored, until Ateniese and Hohenberger [3] showed that it is a very useful tool for sharing web certificates, forming weak group signatures, and authenticating a network path.

Ateniese and Hohenberger [3] re-opened the discussion of proxy re-signature by providing four separate results: (1) motivation for the need of improved schemes, by pointing out that the original BBS scheme [8], while satisfying their security notion, is unsuitable for most practical applications, including the ones proposed in the original paper, (2) formal definitions and a security model, (3) provably secure proxy re-signature constructions from bilinear maps, and (4) new applications. Nonetheless, they left open the problem of designing a multi-use unidirectional scheme where the proxy is able to translate in only one direction and signatures can be re-translated several times. Benoit Libert and Damien Vergnaud [6] have presented the first constructions of multi-use unidirectional proxy re-signature wherein the proxy can only translate signatures in one direction and messages can be re-signed a polynomial number of times.

Further, Ateniese and Hohenberger, while formalising the primitive, pinned down the following useful properties that can be expected from proxy re-signature schemes.

1. Unidirectional: re-signature keys can only be used for delegation in one direction.
2. Multi-use: a message can be re-signed a polynomial number of times.
3. Private Proxy: re-signature keys can be kept secret by an honest proxy.
4. Transparent: a user may not even know that a proxy exists.
5. Unlinkable: a re-signature cannot be linked to the one from which it was generated.
6. Key optimal: a user is only required to store a constant amount of secret data.
7. Non-interactive: the delegatee does not act in the delegation process.
8. Non-transitive: the proxy cannot re-delegate signing rights.
9. Temporary : revoke the rights given to proxy.

The construction given by Blaze et al. is bidirectional and multi-use. However, Ateniese and Hohenberger [3] pinpointed a flaw in the latter scheme: given a signature/re-signature pair, anyone can deduce the re-signature key that has been used in the delegation (i.e. the private proxy property is not satisfied). Another issue in [8] is that the proxy and the delegatee can collude to expose the delegators secret. To overcome these limitations, Ateniese and Hohenberger proposed two constructions

978-1-4244-4793-0/09/$25.00 

based on bilinear maps. The first one is a multi-use, bidirectional protocol built on Boneh-Lynn-Shacham (BLS) signatures [7]. Their second scheme is unidirectional (the design of such a scheme was an open problem raised in [8]) but single-use. It involves two different signature algorithms: first-level signatures can be translated by the proxy whilst second-level signatures cannot. A slightly less efficient variant was also suggested to ensure the privacy of re-signature keys kept at the proxy. The security of all schemes was analyzed in the random oracle model [5].

Digital signatures are vulnerable to leakage of secret key. If the secret key is compromised, any message can be forged. To prevent future forgery of signatures, both public key and secret key must be changed. Notice, that this will not protect previously signed messages: such messages will have to be re-signed with new pair of public key and secret key, but this is not feasible. Also changing the keys frequently is not a practical solution. To address the above problem, the notion of forward security for digital signatures was first proposed by Anderson in [1], and carefully formalised by Bellare and Miner in [4] (see also[2, 10, 9, 11]). The basic idea is to extend a standard digital signature scheme with a key updation algorithm so that the secret key can be changed frequently while the public key stays the same.

We propose a new construction for multi-use (i.e. the translation of signatures can be performed in sequence and multiple times by distinct proxies) unidirectional (i.e. the proxy information allows translating signatures in only one direction) proxy re-signature scheme using the property of forward-security. Our forward-secure proxy re-signature scheme, based on the hardness of factoring, translates one persons signature to another persons signature and additionally facilitates the signers as well as the proxy to guarantee the security of messages signed in the past even if their secret key is exposed today (property of forward-security). With a minor change in resigning key, we can make the scheme to behave as a multi-use bidirectional scheme. The scheme also satisfies the properties viz. private proxy, transparent, unlinkable, key optimal, interactive, non-transitive and temporary.

The organisation of our paper is as follows: In Section 2, we explain two of our schemes i.e Forward-Secure Bi-directional Multi-use Proxy Re-Signature Scheme and Forward-Secure Uni-directional Multi-use Proxy Re-Signature Scheme. In Section 3, we discuss the application of proxy re-signatures in banking environment. Lastly in Section 4, we conclude.

# 2 Forward-Secure Proxy Re-signature Scheme

As digital signatures, proxy re-signatures are also vulnerable to leakage of re-signing key. If the re-signing key is compromised, any one can become a proxy. To prevent future forgery of re-signatures, both the delegator as well as the delegatee must change their public key and secret key pair and a new re-signing key computed. But this will not protect previously signed messages: such messages will have to be re-signed with new pair of public key and secret key which is not feasible. To address this problem, we use the concept of forward security for proxy re-signatures.

To translate Alice's signature to Bob's signature, the secret and public keys are generated as indicated in the Key Generation algorithm. We know that a forward secure signature scheme has its operation divided into time periods, each of which uses a different secret key to sign a message. The new secret keys are generated as described in the Key Evolution algorithm. Alice signs in any time period $j$ using the Signature Generation algorithm. This signature is required to be converted to Bob's signature. Using the protocol indicated in the Re-Signature Key Generation the proxy generates the re-signature key $rk_{A\rightarrow B}$ and executes the Re-sign algorithm to translate Alice's signature to Bob's signature. This scheme works for a period of $T$ time periods, i.e. the proxy has the power to resign only for $T$ time periods and after the expiry of $T$ time periods the proxy is automatically revoked.

## 2.1 Multi-use Bi-directional Proxy Re-Signature Scheme

We propose a new construction for multi-use bidirectional proxy re-signature scheme using the property of forward-security. Our forward-secure proxy re-signature scheme, based on the hardness of factoring, translates one persons signature to another persons signature and additionally facilitates the signers as well as the proxy to guarantee the security of messages signed in the past even if their secret key is exposed today (property of forward-security). With a minor change in re-signing key, we can make the scheme to behave as a multi-use bidirectional scheme. The scheme also satisfies the properties viz. private proxy, transparent, unlinkable, key optimal, interactive, non-transitive and temporary. Following are the algorithms for the Forward-Secure Multi-use Bi-directional Proxy Re-Signature Scheme.

1. **Key generation:** Both Alice and Bob generate the keys by running the following algorithm which takes as input the security parameter $k$, the number $l$ of points in the keys and the number $T$ of time periods over which the scheme is to operate. The notations are same as in Bellare-Miner Forward-secure signature scheme discussed in Chapter 3. $p, q$ are random distinct $k/2$ bit primes each congruent to 3 mod 4. $N \leftarrow p.q$. Alice and Bob agree upon common $N$.
   *Alice's keys:* The base secret key $SKA_0 = (SA_{1,0}, \ldots, SA_{l,0}, N, 0)$ (where $SA_{i,0} \stackrel{R}{\leftarrow} Z_N^*$ and $N$ is a Blum-Williams integer). For verifying signatures the verifier is given the public key $PKA$, calculated as the value obtained on updating the base secret key $T+1$ times: $PKA = (UA_1, \ldots, UA_l, N, T)$ where $UA_i = SA_{i,0}^{2^{T+1}} \bmod N_A, i = 1, \ldots, l$.
   *Bob's keys:* The base secret key $SKB_0 = (SB_{1,0}, \ldots, SB_{l,0}, N, 0)$ (where $SB_{i,0} \stackrel{R}{\leftarrow} Z_N^*$ and $N$ is a Blum-Williams integer). For verifying signatures the verifier is given the public key: $PKB = (UB_1, \ldots, UB_l, N, T)$ where $UB_i = SB_{i,0}^{2^{T+1}} \bmod N_B, i = 1, \ldots, l$.
2. **Key evolution:** During time period $j$ the signer signs using key $SK_j$. This key is generated at the start of period $j$ by applying a key update algorithm to the key $SK_{j-1}$. The update algorithm squares the $l$ points of the secret key at the previous stage to get the secret key at the next stage.
   *Key evolution for Alice:* The secret key $SKA_j = (SA_{1,j}, \ldots, SA_{l,j}, N_A, j)$ of the time period $j$ is obtained from the secret key

$SKA_{j-1} = (SA_{1,j-1}, \ldots, SA_{l,j-1}, N_A, j-1)$ of the previous time period via the update rule: $SA_{i,j} = SA_{i,j-1}^2 \bmod N_A, i = 1, \ldots, l; j = 1, \ldots, T$.
*Key evolution for Bob:* The secret key $SKB_j = (SB_{1,j}, \ldots, SB_{l,j}, N_B, j)$ of the time period $j$ is obtained from the secret key $SKB_{j-1} = (SB_{1,j-1}, \ldots, SB_{l,j-1}, N_B, j-1)$ of the previous time period via the update rule: $SB_{i,j} = SB_{i,j-1}^2 \bmod N_B, i = 1, \ldots, l; j = 1, \ldots, T$.

3. **Re-Signature Key Generation (ReKey):** On input two secret keys $SKA_j = (SA_{1,j}, \ldots, SA_{l,j}, N_A, j)$ and $SKB_j = (SB_{1,j}, \ldots, SB_{l,j}, N_b, j)$, the re-signature key, $rk_{A\to B,j} = (rk_{1,j}, \ldots, rk_{l,j})$ is computed as

$$rk_{i,j} = SB_{i,j}/SA_{i,j} \bmod N$$

where $i = 1, \ldots, l; j = 1, \ldots, T$. The scheme can be used as a bidirectional multi-use proxy re-signature scheme.

Observe that the key $rk_{A\to B}$ can be securely generated as follows:

(a) Proxy sends a random $r \in Z_N^*$ to Alice.

(b) Alice sends $(r/SA_{1,j}, \ldots, r/SA_{l,j})$ to Bob.

(c) Bob sends $(r(SB_{1,j}/SA_{1,j}), \ldots, r(SB_{l,j}/SA_{l,j}))$ to the proxy.

(d) Proxy recovers $(SB_{1,j}/SA_{1,j}, \ldots, SB_{l,j}/SA_{l,j})$.

*Key evolution for Proxy:* The re-signature key $rk_{A\to B,j} = (rk_{1,j}, \ldots, rk_{l,j})$ of the time period $j$ is obtained from the re-signature key $rk_{A\to B,j-1} = (rk_{1,j-1}, \ldots, rk_{l,j-1})$ of the previous time period via the update rule: $rk_{i,j} = rk_{i,j-1}^2 \bmod N, i = 1, \ldots, l; j = 1, \ldots, T$.

4. **Signature Generation:** It has as input the secret key $SKA$ of the current period, the message $M$ to be signed, and the value $j$ of the period itself to return a signature $\langle j, (Y, Z)\rangle$ where $Y, Z$ in $Z_N^*$ are calculated as follows:

$$Y = R^{2^{(T+1-j)}} \bmod N, \quad \text{where} \quad R \overset{R}{\leftarrow} Z_N^* \tag{1}$$

$$Z = R\prod_{i=1}^{l} SA_{i,j}^{c_i} \bmod N, \quad \text{where} \quad c_1, \ldots, c_l = H(j, Y, M) \tag{2}$$

being the $l$ output bits of a public hash function.

5. **Re-Sign (ReSign):** We verify the signature before we re-sign. On input a re-signature key $rk_{A\to B,j}$, a public key $PKA$, a signature $\langle j, (Y, Z)\rangle$, and a message $M$, we check that $\text{Verify}(PKA, m, \langle j, (Y, Z)\rangle) = 1$. If the signature,$\langle j, (Y, Z)\rangle$, does not verify, re-signing is not done and an error message is displayed.

If the signature is verified, we set

$$Z' = Z\prod_{i=1}^{l} rk_{i,j}^{c_i} \bmod N,$$

where $c_1, \ldots, c_l = H(j, Y, M)$ and output the signature $\langle j, (Y, Z')\rangle$. Observe that

$$\begin{aligned} Z' &= Z\prod_{i=1}^{l} rk_{i,j}^{c_i} \bmod N \\ &= R\prod_{i=1}^{l} SA_{i,j}^{c_i}(\prod_{i=1}^{l} SB_{i,j})^{c_i}/(\prod_{i=1}^{l} SA_{i,j})^{c_i} \bmod N \\ &= R\prod_{i=1}^{l} SB_{i,j}^{c_i}, \end{aligned}$$

which shows that the signature $\langle j, (Y, Z')\rangle$ is Bob's signature. Thus, Re-Sign has translated Alice's signature into Bob's signature.

Though, just as in BBS scheme, our scheme also computes the resigning key as the ratio of secret keys of Alice and Bob, but the resigning key cannot be computed using the signature/re-signature pair. BBS proxy re-signature scheme is briefly described in Appendix A.9. Let the re-signature key be

$$rk_{A\to B,j} = (rk_{1,j}, \ldots, rk_{l,j})$$

where $rk_{i,j} = SB_{i,j}/SA_{i,j} \bmod N$. Let $\langle j, (Y, Z)\rangle$ and $\langle j, (Y, Z')\rangle$ be the signature and re-signature pair respectively, where $Z = R\prod_{i=1}^{l} SA_{i,j}^{c_i} \bmod N$ and $Z' = R\prod_{i=1}^{l} SB_{i,j}^{c_i} \bmod N$. The ratio of re-signature to signature is

$$\begin{aligned} Z'/Z &= (R\prod_{i=1}^{l} SB_{i,j}^{c_i} \bmod N)/(R\prod_{i=1}^{l} SA_{i,j}^{c_i} \bmod N) \\ &= (\prod_{i=1}^{l} SB_{i,j}^{c_i}/\prod_{i=1}^{l} SA_{i,j}^{c_i}) \bmod N \\ &= (\prod_{i=1}^{l} (SB_{i,j}/SA_{i,j})^{c_i} \bmod N. \end{aligned}$$

Observe that the ratio of re-signature to signature does not yield the resigning key. Using the re-signature key the proxy can turn Alice's signatures into Bob's and Bob's to Alice's by just inverting the ratio of signatures. Thus the scheme here is bidirectional.

The signature generated by Signature generation algorithm is provided as one of the inputs to the Re-Sign algorithm. When we observe the equations of Signature generation and Re-Sign algorithms, we can say that both are generating signatures of the form $\langle j, (Y, Z)\rangle$ (Bellare-Miner signatures). Thus, signatures generated by either the Sign or ReSign algorithms can be taken as input to Re-Sign. This property when applied repeatedly can be used to translate Bob's signature to Carol's signature using the Re-sign key $rk_{B\to C,j}$, Carol's signature to Dick's signature using the Re-sign key $rk_{C\to D,j}$ and so on. Therefore we claim that a message can be re-signed several times which is the property of multi-use scheme. This Bidirectional Multi-use scheme is a Transitive scheme as shown below:
To translate Alice's signature to Bob's signature we use,

$$rk_{A\to B,j} = (rk_{1,j}^{AB}, \ldots, rk_{l,j}^{AB})$$

where $rk_{i,j}^{AB} = SB_{i,j}/SA_{i,j} \bmod N$.
To translate Bob's signature to Carols's signature we use,

$$rk_{B\to C,j} = (rk_{1,j}^{BC}, \ldots, rk_{l,j}^{BC})$$

where $rk_{i,j}^{BC} = SC_{i,j}/SB_{i,j} \bmod N$.

To translate Alice's signature to Carols's signature we are required to have,

$$rk_{A\to C,j} = (rk_{1,j}^{AC}, \ldots, rk_{l,j}^{AC})$$

where $rk_{i,j}^{AC} = SC_{i,j}/SA_{i,j} \bmod N$. Note that $rk_{i,j}^{AC} = SC_{i,j}/SA_{i,j} = (SC_{i,j}/SB_{i,j}).(SB_{i,j}/SA_{i,j}) = rk_{1,j}^{BC}.rk_{1,j}^{AC} \bmod N$

6. **Signature Verification:** A claimed signature $\langle j, (Y, Z)\rangle$ for the message $M$ in time period $j$ is accepted if

$$Z^{2^{(T+1-j)}} = Y \prod_{i=1}^{l} UA_i^{c_i} \bmod N \tag{3}$$

where $c_1, \ldots, c_l = H(j, Y, M)$, and rejected otherwise. Notice that since

$$\begin{aligned} Z^{2^{(T+1-j)}} &= (R(\prod_{i=1}^{l} SA_{i,j}^{c_i}))^{2^{(T+1-j)}} \bmod N \\ &= Y.(\prod_{i=1}^{l} SA_{i,0}^{2^{(T+1)}c_i}) \bmod N \\ &= Y.\prod_{i=1}^{l} UA_i^{c_i} \bmod N. \end{aligned}$$

a signature by an honest signer with the secret key will be accepted.

## 2.2 Multi-use Unidirectional Proxy Re-signature Scheme

To address the first open problem of Ateniese and Hohenberger, we propose a new construction for multi-use (i.e. the translation of signatures can be performed in sequence and multiple times by distinct proxies) unidirectional (i.e. the proxy information allows translating signatures in only one direction) proxy re-signature scheme using the property of forward-security. Our forward-secure proxy re-signature scheme, based on the hardness of factoring, translates one persons signature to another persons signature and additionally facilitates the signers as well as the proxy to guarantee the security of messages signed in the past even if their secret key is exposed today (property of forward-security). The scheme also satisfies the properties viz. private proxy, transparent, unlinkable, key optimal, interactive, non-transitive and temporary. With a minor change in resigning key, we can make the scheme to behave as a multi-use bidirectional scheme.

The key generation, key evolution and signature generation algorithms are same as the ones used in Forward-Secure Multi-use Uni-directional Proxy Re-Signature Scheme discussed in Section 6.4.1. The other algorithms are given below:

1. **Re-Signature Key Generation (ReKey):** On input two secret keys $SKA_j = (SA_{1,j}, \ldots, SA_{l,j}, N_A, j)$ and $SKB_{j+1} = (SB_{1,j+1}, \ldots, SB_{l,j+1}, N_B, j+1)$, the re-signature key, $rk_{A\to B,j} = (rk_{1,j}, \ldots, rk_{l,j})$ is computed as $rk_{i,j} = SB_{i,j+1}/SA_{i,j} \bmod N$ where $i = 1, \ldots, l; j = 1, \ldots, T-1$.

   Observe that the key $rk_{A\to B}$ can be securely generated as follows:

   (a) The proxy sends a random $r \in Z_N$ to Alice.

   (b) Alice sends $(r/SA_{1,j}, \ldots, r/SA_{l,j})$ to Bob.

   (c) Bob sends $(r(SB_{1,j+1}/SA_{1,j}), \ldots, r(SB_{l,j+1}/SA_{l,j})$ to the proxy.

   (d) The proxy recovers $(SB_{1,j+1}/SA_{1,j}, \ldots, SB_{l,j+1}/SA_{l,j})$.

2. **Re-Sign (ReSign):** On input a re-signature key $rk_{A\to B,j}$, a public key $PKA$, a signature $\langle j, (Y, Z)\rangle$, and a message $M$, we check if $\text{Verify}(PKA, m, \langle j, (Y, Z)\rangle) = 1$. If so, we set

$$Z' = Z \prod_{i=1}^{l} rk_{i,j}^{c_i} \bmod N,$$

where $c_1, \ldots, c_l = H(j, Y, M)$ and output the signature $\langle j+1, (Y, Z')\rangle$, otherwise we output an error message. Observe that,

$$\begin{aligned} Z' &= Z.\prod_{i=1}^{l} rk_{i,j}^{c_i} \bmod N \\ &= R\prod_{i=1}^{l} SA_{i,j}^{c_i}.(\prod_{i=1}^{l} SB_{i,j+1})^{c_i}/(\prod_{i=1}^{l} SA_{i,j})^{c_i} \bmod N \\ &= R\prod_{i=1}^{l} SB_{i,j+1}^{c_i}, \end{aligned}$$

which shows that the signature $\langle j+1, (Y', Z')\rangle$ is Bob's signature. Thus, Re-Sign has translated Alice's signature into Bob's signature.

Even here, just as in BBS scheme, our scheme computes the resigning key as the ratio of secret keys of Alice and Bob, but the resigning key cannot be computed using the signature/re-signature pair as shown below:

Let the re-signature key be $rk_{A\to B,j} = (rk_{1,j}, \ldots, rk_{l,j})$ where $rk_{i,j} = SB_{i,j+1}/SA_{i,j} \bmod N$.

Let $\langle j, (Y, Z)\rangle$ and $\langle j, (Y, Z')\rangle$ be the signature/re-signature pair where, $Z = R\prod_{i=1}^{l} SA_{i,j}^{c_i} \bmod N_A$ and $Z' = R\prod_{i=1}^{l} SB_{i,j+1}^{c_i} \bmod N$. Then

$$\begin{aligned} Z'/Z &= (R\prod_{i=1}^{l} SB_{i,j+1}^{c_i} \bmod N)/(R\prod_{i=1}^{l} SA_{i,j}^{c_i} \bmod N) \\ &= (\prod_{i=1}^{l} SB_{i,j+1}^{c_i}/\prod_{i=1}^{l} SA_{i,j}^{c_i}) \bmod N \\ &= (\prod_{i=1}^{l} (SB_{i,j+1}/SA_{i,j})^{c_i} \bmod N. \end{aligned}$$

Observe that the ratio of re-signature to signature does not yield the resigning key.

In the protocol indicated in the Re-Signature Key Generation, Alice uses her secret key of $j^{th}$ time period while Bob uses his secret key of $(j+1)^{th}$ time period in the computation. Thus Alice's signature in the $j^{th}$ time period is converted into Bob's signature in the $(j+1)^{th}$ time period. Also, Bob's signature gets verified in the $(j+1)^{th}$ time period but not in $j^{th}$ time period. By choosing Bob's

$(j+1)^{th}$ time period secret key and Alice's $j^{th}$ time period secret key we are able to give the Unidirectional property (re-signature keys can only be used for delegation in one direction) to our scheme. This is explained below.

The re-signature key used to translate Alice's signature to Bob's signature is $rk_{A\to B,j} = (rk_{1,j}, \ldots, rk_{l,j})$ where $rk_{i,j} = SB_{i,j+1}/SA_{i,j} \bmod N$. And, the re-signature key required to translate Bob's signature to Alice's signature is $rk_{B\to A,j} = (rk_{1,j}, \ldots, rk_{l,j})$ where $rk_{i,j} = SA_{i,j+1}/SB_{i,j} \bmod N$. We observe that $rk_{B\to A,j}$ cannot be obtained from $rk_{A\to B,j}$ as the proxy has access to ratio of $SB_{i,j+1}/SA_{i,j}$ but not to individual secret key components $SA_{i,j+1}$ and $SB_{i,j}$.

The signature generated by Signature generation algorithm is provided as one of the inputs to the Re-Sign algorithm. When we observe the equations of Signature generation and Re-Sign algorithms, we can say that both are generating signatures of the form $\langle j, (Y, Z)\rangle$ (Bellare-Miner signatures). Thus, signatures generated by either the Sign or ReSign algorithms can be taken as input to Re-Sign. This property when applied repeatedly can be used to translate Bob's signature to Carol's signature using the Re-sign key $rk_{B\to C,j}$, Carol's signature to Dick's signature using the Re-sign key $rk_{C\to D,j}$ and so on. Therefore we claim that a message can be re-signed several times which is the property of multi-use scheme.

This Unidirectional Multi-use scheme is also Non-Transitive.

To translate Alice's signature to Bob's signature we use, $rk_{A\to B,j} = (rk_{1,j}^{AB}, \ldots, rk_{l,j}^{AB})$
where $rk_{i,j}^{AB} = SB_{i,j+1}/SA_{i,j} \bmod N$.

To translate Bob's signature to Carols's signature we use, $rk_{B\to C,j} = (rk_{1,j}^{BC}, \ldots, rk_{l,j}^{BC})$
where $rk_{i,j}^{BC} = SC_{i,j+1}/SB_{i,j} \bmod N$.

To translate Alice's signature to Carols's signature we are required to have, $rk_{A\to C,j} = (rk_{1,j}^{AC}, \ldots, rk_{l,j}^{AC})$
where $rk_{i,j}^{AC} = SC_{i,j+1}/SA_{i,j} \bmod N$. From the above, $rk_{i,j}^{AC}$ cannot be obtained from $rk_{i,j}^{AB}$ and $rk_{i,j}^{BC}$.

3. **Signature Verification:** As for verification, a claimed signature $\langle j, (Y, Z)\rangle$ for the message $M$ in time period $j$ is accepted if

$$Z^{2^{(T+1-j)}} = Y \prod_{i=1}^{l} UA_i^{c_i} \bmod N \tag{4}$$

where $c_1, \ldots, c_l = H(j, Y, M)$, and rejected otherwise. Notice that since

$$\begin{aligned} Z^{2^{(T+1-j)}} &= (R(\prod_{i=1}^{l} SA_{i,j}^{c_i})^{2^{(T+1-j)}} \bmod N \\ &= Y.(\prod_{i=1}^{l} SA_{i,0}^{2^{(T+1)}c_i}) \bmod N \\ &= Y.\prod_{i=1}^{l} UA_i^{c_i} \bmod N. \end{aligned}$$

a signature by an honest signer with the secret key will be accepted.

# 3 Applications in e-banking

As most banking applications require the consent of the signer, we have opted for an interactive method of computing the re-signing key that is, proxy, delegator and delegatee are involved in the computation of the re-signing key.

1. Loan Sanctioning process: In this process a number of bank officials are involved at various stages, right from verifying the records to sanctioning the loan. At every stage the concerned official is required to sign the loan application indicating that the documents given in support of the loan is in accordance with the bank guidelines. As each official signs independently, there is possibility that the officials sign for different data. Also, all the signatures are verified at the end before sanctioning the loan. The problem here is we need to maintain signatures and the public keys of all the officials until the sanction of the loan.

   We address this problem using proxy re-signatures. Let us assume that there are four officials, $A, B, C$ and $D$. The resigning keys, $rk_{A\to B}$, $rk_{B\to C}$ and $rk_{C\to D}$ between the officials are computed. Initially Official $A$ verifies the loan documents pertaining to his section, signs the loan application as $s_A$ and passes the loan application and the documents to Official $B$. The signature $s_A$ is verified by Official $B$. He next verifies the loan documents pertaining to his section and applies the Re-Sign algorithm which converts official $A$'s signature to his own signature, $s_B$. By doing this, along with Official $A$ Official $B$ has become responsible for verifying the documents as the signature of Official $B$ is not created independently but by using the signature of Official $A$. In this way, at every stage of loan processing, one official's signature is replaced by another official's signature. In the end, only Official $D$'s signature will be on the loan application where Official $D$ can be assumed as the Manager of the bank. By using re-signatures following are the advantages:

   - At every stage of loan processing only one verification with one public key is sufficient.
   - Only one signature needs to be stored at any stage.
   - The original message cannot be changed.
   - On re-signing, the corresponding official becomes responsible for the completion of the process at that stage.
   - At the send, only one signature verification is required instead of four verifications.

2. Frequently changing public keys: A customer of a bank may frequently change his public key due to policy of the organisation or for the sake of security or due to leakage of his secret key. Let $(PK_O, SK_O)$ be the old public key - secret key pair and $(PK_N, SK_N)$ be the new public key - secret key pair of a customer. Sometimes there may be need to verify some old documents which were signed using the old secret key. To handle this situation banks can store the resigning key $rk_{O\to N}$ (this key can be computed whenever the customer opts for change of public key secret key pair) which helps them to translate an old signature signed using $SK_O$ to a new signature which can be verified

using the new public key $PK_N$. This enables to verify old signatures and also all signatures (old or new) using the new public key.

3. Accounts to be operated by a nominee: On many occasions a customer $A$ may be disabled (for a short or long duration) to operate his account. This forces the bank to give power to the nominee $B$ to operate the account. The resigning key $rk_{B\rightarrow A}$ is required to be computed by the bank when the account holder declares his nominee. On any transaction done by the nominee $B$, bank translates the signature to the original account holder's (here $A$) signature using proxy re-signatures. This translation is not possible without the bank's intervention. By using proxy re-signatures, the bank need not store the public key of the nominee to verify his signature. This facility given to nominee can be revoked at any instant.

4. Transferrable e-cheques: The concept of transferrable e-cheques [12, 13] is already introduced in Section 4.3.2 of Chapter 4. Here, whenever a cheque is transferred from one person to another person, a partial multi-signature is generated which can be verified using the product of public keys of all the previous signers. As the cheque gets transferred to different persons the computation cost of the multi-signature increases and also we need to have the product of public keys of all the signers.

   We now propose an alternative method for transferring e-cheques with a proxy re-signature having transitive property. Let us assume that there are four persons $A, B, C$ and $D$ and $A$ issues a cheque to $B$. If $B$ wants to re-issue the same e-cheque to $C$, $B$ must act as a proxy and compute the re-signing key $rk_{A\rightarrow B}$ by communicating with $A$. Using this key, $B$ can translate $A$'s signature to that of his own. When $C$ re-issues the e-cheque to $D$, in the same way as $B$, $C$ act as a proxy and computes the re-signing key $rk_{B\rightarrow C}$ and translates $B$'s signatures to that of his own. Before $D$ deposits the e-cheque in his bank, he translates $C's$ to that of his own. The bank verifies the signature of $D$, which implies the verification of $A's$ signature by virtue of transitivity of proxy signatures. If $A$ also has an account in the same bank, the bank deducts the cheque amount from $A's$ account and credits the same to $D's$ account. If $A$ has an account in a different bank, the bank sends the e-cheque details to that bank, which on verifying $A$'s e-cheque details like account number and cheque number deducts the cheque amount from $A$'s account and sends a message to credit the cheque amount to $D$'s account. Thus, whenever a person wants to re-issue an e-cheque to another person, he can translate the signature of the issuer of the e-cheque existing on the e-cheque to his own signature. Of course, there is additional cost involved in computing the resigning key. But any person who receives the e-cheque needs to verify the signature only with the public key of the person who issued the e-cheque to him.

# 4 Conclusion

We have proposed a solution for one of the open challenges for the design of multi-use unidirectional proxy re-signature systems. We have come up with a forward-secure proxy re-signature scheme which translates one person's signature to another person's signature and additionally facilitates the signers as well as the proxy to guarantee the security of messages signed in the past even if their secret key is exposed today. Our scheme is a multi-use unidirectional scheme where the proxy is able to translate in only one direction and signatures can be re-translated several times. With a minor change in resigning key, we can make the scheme to behave as a multi-use bidirectional scheme. In view of the banking applications we have attempted to satisfy the following properties in our re-signature scheme: private proxy, transparent, unlinkable, key optimal, interactive(as banking applications need), non-transitive and temporary.

# References


[1] Anderson, R.: Invited Lecture, Fourth Annual Conference on Computer and Communications Security, ACM, (1997).

[2] Abdalla,M., Reyzin,L. *A New Forward-Secure Digital Signature Scheme.* In: ASIACRYPT 2000, LNCS 1976, pp. 116-129. Springer-Verlag, (2000),116-129.

[3] G. Ateniese, S. Hohenberger. *Proxy re-signatures: new definitions, algorithms, and applications.* In ACM CCS'05, pages 310319, ACM Press, 2005

[4] Bellare, M., Miner, S. *A Forward-Secure Digital Signature Scheme.* In: Wiener, M. (eds.): Advances in Cryptology-Crypto 99 proceedings, Lecture notes in Computer Science, Vol. 1666. Springer-Verlag, (1999).

[5] M. Bellare, P. Rogaway. *Random oracles are practical: A paradigm for designing efficient protocols.* In ACM CCS'93, pages 6273, ACM Press, 1993.

[6] Benoit Libert and Damien Vergnaud. *Multi-Use Unidirectional Proxy Re-Signatures* arXiv:0802.1113v1 [cs.CR] 8 February 2008.

[7] D. Boneh, B. Lynn, H. Shacham. *Short signatures from the Weil pairing.* In Asiacrypt'01, volume 2248 of LNCS, pages 514532. Springer, 2002.

[8] Blaze, Bleumer, and Strauss. *Divertible protocols and atomic proxy cryptography.* In Advances in Cryptology EUROCRYPT'98, volume 1403 of LNCS, Springer-Verlag, 241-256.

[9] Itkis, G., Reyzin, L. *Forward-secure signatures with optimal signing and verifying.* In: CRYPTO'01, LNCS 2139, Springer-Verlag, (2001), 332-354.

[10] Krawczyk, H. *Simple forward-secure signatures from any signature scheme.* In: Proc. of the 7th ACM Conference on Computer and Communications Security (CCS 2000), ACM, (2000), 108-115.

[11] Kozlov, A, Reyzin, L.: *Forward-Secure Signatures with Fast Key Update.* In: Security in Communication Networks (SCN 2002), LNCS 2576, Springer-Verlag, (2002), (241-256).

[12] K. Ohta and T. Okamoto. *Multisignature schemes secure against active insider attacks.* IEICE Trans. Fundamentals, E82-A/1:2131, 1999.

[13] N.R.Sunitha, B.B.Amberker, Prashant Koulgi, *Transferable e-cheques using Forward-Secure Multi-signature Scheme,* In: The World Congress on Engineering and Computer Science 2007, 24-26 October, 2007, San Francisco, USA.