Transitive Signatures Based on Non-adaptive Standard Signatures

Zhou Sujing

Nanyang Technological University, Singapore,
zhousujing@pmail.ntu.edu.sg

Abstract. Transitive signature, motivated by signing vertices and edges of a dynamically growing, transitively closed graph, was first proposed by Micali and Rivest. The general designing paradigm proposed there involved a underlying standard signature scheme, which is required to be existentially unforgeable against adaptive chosen message attacks. We show that the requirement for the underlying signature is not necessarily so strong, instead non-adaptive security is enough to guarantee the transitive signature scheme secure in the strongest sense, i.e, transitively unforgeable under adaptive chosen message attack (defined by Bellare and Neven). We give a general proof of such transitive signature schemes, and also propose a specific transitive signature scheme based on factoring and strong-RSA. Hence the choice of standard signatures that can be employed by transitive signature schemes is enlarged. The efficiency of transitive signature schemes may be improved since efficiency and security are trade-off parameters for standard signature schemes.

Keywords: Signatures, Transitive signatures.

1 Introduction

The concept of *transitive signature* was first proposed by Micali and Rivest in [1]; it is used to sign vertices and edges of a dynamically growing, transitively closed graph. Transitively closed graph has the property of including any edge if there is a path between its two vertices.

A transitive signature has the following properties: (1) given the signatures of two adjacent edges (i, j) and (j, k), *anyone* can easily derive the signature of edge (i, k); (2) it should be hard for an adversary to forge valid signature of an edge outside the *transitive closure* of the graph, and of a new vertex outside the graph, even after the adversary has adaptively queried about signatures of a number of vertices and edges of its choice.

Both [1] and [2] adopt the following paradigm to construct transitive signatures: (1) Signer publishes representation value $L(i)$ of a node with index i by a trapdoor one-way function applied to the secret value $l(i)$ of node i, i.e., $L(i) = f(l(i))$. (2) Signer generates the signature σ_i of $(i, L(i))$ using a standard signature scheme, $(i, L(i), \sigma_i)$ is called certificate of the node i. Signer generates and publishes the certificate of another node j in the same way. (3) Then Signer generates the signature δ_{ij} of the edge (i,j) either by division $(\delta_{ij} = l(i)l(j)^{-1})$, or addition $(\delta_{ij} = l(i) - l(j))$, so that transitive feature remains $(\delta_{ik} = \delta_{ij}\delta_{jk}^{-1}$ or $\delta_{ik} = \delta_{ij} + \delta_{jk})$. The resulting transitive signature of an edge includes two node certificates and a signature of the edge $((i, L(i), \sigma_i), (j, L(j), \sigma_j), \delta_{ij})$.

The security of transitive signatures constructed by the paradigm above depends on the security of the underlying standard signature and the trapdoor one-way function. In [1], the trapdoor one-way function used is $f(x,y) = g^x h^y$, so the security is based on discrete logarithm problem. In [2], the trapdoor one-way functions adopted are modulo square and RSA function, so the security is based on factoring problem and RSA assumption respectively. In all the schemes, the underlying standard signatures are required existentially unforgeable against *adaptive* chosen message attack, which is the strongest security notion in [3].

Our contribution: We examined the security proof of such transitive signature in detail rather than viewing the standard signature as a black box, and found that the security requirement of the underlying standard signature is too strong in the transitive signature scenario. It is enough to have a standard signature that is existentially unforgeable against non-adaptive chosen message attack for the transitive signature to be secure against adaptive chosen message attack. Hence the choice of standard signature is enlarged since there are more standard signatures that are provably secure against non-adaptive chosen message attack than that against adaptive attack. As an example, we adopted such a standard signature scheme which is more efficient than most of the strongly provably secure signatures, such as Cramer-Shoup's scheme[4], Fischlin's scheme [5]; the resulting transitive signature is provably secure against adaptive chosen message attack based on factoring and strong RSA assumption in standard model.

Organization of this paper: We give some definitions needed in section 2. In section 3 we describe our general transitive signature scheme and its proof. Then we present a specific transitive signature in detail in section 4. Section 5 is the conclusion.

2 Notations and Definitions

In this section, we give some notations and definitions that will be used in the paper.

2.1 Notations

In the paper, we let $a \leftarrow A$ denote a is selected from the set A, $a \overset{R}{\leftarrow} A$ denote a is selected uniform randomly from set A.

Let $a \leftarrow b$ (b is an element) denote a is assigned the value of b.

Let $a \leftarrow \mathcal{A}$ denote a is assigned the value of the output of algorithm $\mathcal{A}$.

$|a|$ denotes the binary bit length of a, $|H(\cdot)|$ denotes the binary bit length of the output value of function H.

2.2 Definitions

Definition 1. ***Transitive Closure of a Graph****[1]. Transitive closure of a graph $G = (V, E)$ is the graph $G^* = (V, E^*)$, such that an edge $(i, j) \in E^*$ if and only if there is a path from i to j. If $G = G^*$, then we call the graph G is transitively closed. Here we only consider undirected graph.*

Definition 2. ***Transitive Signature Scheme****[2]. A transitive signature scheme $TS = (TKG, TSgign, TVf, Comp)$ is determined by the following four polynomial time algorithms:*

1. *TKG (key generation): a probabilistic algorithm, with security parameter as input, returns a pair of public key and secret key (tpk, tsk).*
2. *TSign (signing): a probabilistic algorithm, takes input tsk, and nodes i, j, returns the signature δ_{ij} of edge (i, j).*
3. *TVf (verification): a deterministic algorithm, given tpk, nodes i, j, and a candidate signature δ, verify if δ is a valid signature of (i, j), returns 1 if so, otherwise returns 0.*
4. *Comp (composition): a deterministic algorithm, given tpk, and nodes i, j, k, and corresponding signatures δ_{ij},δ_{jk}, returns a value δ_{ik} which should be valid signature of edge (i, k), or returns failure if $(i, k) \notin E^*$.*

Definition 3. ***Security of Transitive Signature Scheme****[2]. Let (V, E^*) denotes a transitive closure of graph $G = (V, E)$. A transitive signature scheme $TS = (TKG, TSign, TVf, Comp)$ is transitively unforgeable under adaptive chosen message attack if for any adversary A with running time polynomial in k, the following value $Adv_{TS}^{tu-acma}$ is negligible:*

$$Adv_{TS,A}^{tu-acma} = \mathrm{P}[Exp_{TS,A}^{tu-acma}(k) = 1], \qquad (1)$$

where $tu - acma$ denotes transitively unforgeable under adaptive chosen message attack. $Exp_{TS,F}^{tu-acma}(k)$ is an experiment defined as:

```
Experiment: $Exp_{TS,F}^{tu-acma}(k)$
$(tpk, tsk) \leftarrow TKG(1^k)$
$(i', j', \sigma') \leftarrow A^{TSign(tsk,\cdot,\cdot)}(tpk)$
if $TVf(i', j', \sigma') \neq 1$ then
  return 0
else if $(i', j' \in V \wedge (i', j') \notin E^*)$ or $(i' \notin V \vee j' \notin V)$ then
  return 1
end if
```

($A^{TSign(tsk,\cdot,\cdot)}$ denotes the output of A after A queried about Oracle $TSign(tsk, \cdot, \cdot)$.)

In the following, we give some definitions needed in security proof of the standard signature employed by our proposed specific transitive signature scheme.

Definition 4. ***RSA Assumption****[4].Given a randomly generated RSA modulus n, a random $z \in \mathbb{Z}_n^*$, and r that satisfy certain requirement, it is hard to find $y \in \mathbb{Z}_n^*$ such that $y^r = z \bmod n$.*

Definition 5. ***Strong RSA Assumption***[4]. *Given a randomly generated RSA modulus n, a random $z \in \mathbb{Z}_n^*$, it is hard to find $r > 1$ and $y \in \mathbb{Z}_n^*$ such that $y^r = z \bmod n$.*

Lemma 1. *Given $x, y \in \mathbb{Z}_n^*$, and $a, b \in \mathbb{Z}$, such that $x^a = y^b \bmod n$, $\gcd(a,b) = 1$, $\widetilde{x} \in \mathbb{Z}_n^*$ can be efficiently calculated such that $\widetilde{x}^a = y \bmod n$.*

This lemma is very useful in proving the security of some signature schemes, as the one in this paper.

3 General Scheme

The general transitive signature scheme we describe below also employs standard signature scheme to generate certificates of the nodes, with the difference that the standard signature here is only provable secure against chosen message attack instead of adaptive attack as required in [2] and [1]. There are three types of transitive signatures proposed so far, they are based on discrete logarithm [1], factoring and RSA one-more-inversion [2] respectively. Here we construct our general scheme based on factoring, but it can also work with the other two trapdoor one-way functions.

3.1 Proposed General Scheme

We construct a transitive signature scheme TS as follows. We choose a standard signature that is provable existentially unforgeable against chosen message attack, denoted as SDS, with signing algorithm SDS_{sign}, verification algorithm SDS_{vrf}. The four algorithms $TKG, TSign, TVf, Comp$ are defined as follows:

- TKG: given input 1^k, generates (pk, sk) of SDS, output $tpk = (n, pk)$ and $tsk = (p, q, sk)$, where n is a RSA modulus with p, q as its prime factors.
- TSign: TSign maintains state (V, l, C), suppose $Node$ is the set of integers indexing all the the nodes in graph G, then $V \subset Node$ represents queried nodes; l consists of elements $l(i) \in \mathbb{Z}_n^*, i \in Node$, which are kept secret; C is set of certificate of queried nodes, with the element $C_i = (i, L(i), \sigma_i)$, where $i \in V$, $L(i)$ is the public representation of node i, σ_i is the signature of the node i. At first, the initial state of (V, l, C) is empty. When invoked on inputs (tsk, tpk, i, j), TSign does:

```
if $j < i$ then
    swap $i, j$
else if $i \notin V$ then
    $V \leftarrow V \cup i$; $l(i) \overset{R}{\leftarrow} \mathbb{Z}_n^*$; calculates $L(i) = l(i)^2 \bmod n$; calculates $\sigma_i = SDS_{sign}(sk, i, L(i))$
else if $j \notin V$ then
    $V \leftarrow V \cup j$; $l(j) \overset{R}{\leftarrow} \mathbb{Z}_n^*$; calculate $x_j = l(j)^2 \bmod n$; calculates $\sigma_j = SDS_{sign}(sk, j, L(j))$
```

```
  end if
    set $\delta_{ij} = l(i)l(j)^{-1} \bmod n$, $C_i = (i, L(i), \sigma_i)$, $C_j = (j, L(j), \sigma_j)$.
```

The resulting transitive signature of edge (i, j) is (C_i, C_j, δ_{ij}).

- TVf: given input (C_i, C_j, δ_{ij}), parses C_i as $(i, L(i), \sigma_i)$, C_j as $(j, L(j), \sigma_j)$, then check if C_i and C_j are valid according to the verification algorithm of SDS (SDS_{vrf}), and if $\delta_{ij}^2 = L(i)L(j)^{-1} \bmod n$.
- Comp: given valid transitive signatures (C_i, C_j, δ_{ij}) and (C_j, C_k, δ_{jk}), suppose $i < j < k$, if not so, we can swap the sequence of i, j, k. Calculates $\delta_{jk} = \delta_{ij}\delta_{jk}$ which can be verified by $\delta_{ik}^2 = L(i)L(j)^{-1} \bmod n$.

3.2 Security Proof

We recall the security proof of standard signature schemes. For a standard signature scheme, an adversary is considered successful if it can figure out a valid signature of a new message, after a number of queries about signatures of any messages of its choice. In security proof of a standard signature scheme, we assume such an adversary exists, then we construct a simulator of the signer; when the adversary queries about the signature of a message, the simulator responses a simulated signature that is valid in the view of the adversary; then after a number of queries, the adversary calculate a forgery signature of a new message that has never been queried; if the simulator can utilize the forgery to solve some hard problem that is considered unsolvable in polynomial time, the signature scheme is provably secure.

For transitive signature schemes however, it is easy to find such adversaries, because anyone can compose the signature of an edge in the transitive closure of the graph. So here an adversary is considered successful only if it can figure out a valid signature of an edge *outside* the transitive closure of the graph, or a valid signature of a new node outside the graph, after it queries about signatures of polynomial number of edges at its will.

Theorem 1. *The above transitive signature TS is transitively unforgeable against adaptive chosen message attack if the underlying standard signature SDS is existentially unforgeable against non-adaptive chosen message attack and factoring is hard.*

Proof. Let S_{ts} denote the simulator of the above transitive signature TS, S_{sds} denote the simulator of the employed standard signature SDS. Suppose a forger of the transitive signature F_{ts} is able to output a forgery signature with non-negligible probability. We construct an adversary A to solve the factoring problem or break the underlying standard signature. First A invoke S_{ts} as follows: generates $l(i)$ randomly from $\mathbb{Z}_n^*$, calculates $L(i) = f(l(i))$, $i \in [1, Q]$ (Q is the query number), invoke simulator S_{sds}, queries S_{sds} about signatures of $L(i), i \in [1, Q]$, after gets responses from S_{sds}, sets $\sigma_i = S_{sds}(L(i))$, then we get $Node$, the set of $(l(i), L(i), \sigma_i), i \in [1, Q]$; invoke F_{ts}:

```
reply $F_{ts}$'s query of $(i, j)$ as:
if $i = j$ then
```

```
    return failure
  else if $i \notin V$ then
    $V \leftarrow V \cup i$
    select an element randomly from $Node$, denote as $Node_i$, $Node = Node - \{Node_i\}$
  else if $j \notin V$ then
    $V \leftarrow V \cup j$
    select an element randomly from $Node$, denote as $Node_j$, $Node = Node - \{Node_j\}$
  end if
  if $i < j$ then
    set $\delta_{ij} = l(i)l(j)^{-1} \bmod n$
  else if $i > j$ then
    set $\delta_{ji} = l(j)l(i)^{-1} \bmod n$
  end if
  return to $F_{ts}$ the value of $((i, L(i), \sigma_i), (j, L(j), \sigma_j), \delta_{ij})$ as the queried signature of edge $(i, j)$.
```

Suppose F_{ts} outputs an edge signature $((i', L(i'), \sigma_i{}'), (j', L(j'), \sigma_j{}'), \delta')$ from the above query results it has got. The adversary A can do the following:

```
  if $i', j' \in V \wedge (i', j') \notin E^*$ then
    $S_{ts}$ can factor $n$ by knowing $\delta'$ and $\delta_{i'j'} = l(i')l(j')^{-1} \bmod n$
  else if $i' \notin V$ or $j' \notin V$ (let $j' \notin V$ wlog) then
    $(L(j'), \sigma_{j'})$ is a forgery message-signature pair of $SDS$, forward it to $S_{sds}$, $S_{sds}$ use it to solve the underlying problem
  end if
```

By the similar argument as in the proof of [2], we obtain

$$Adv_{TS}^{tu-acma} = 2Adv_{MG}^{fac} + Adv_{SDS}^{uf-cma}, \tag{2}$$

where Adv_{SDS}^{uf-cma} means the probability of forging a signature under chosen message attacks, Adv_{MG}^{fac} means the probability of breaking the RSA modulus generated by the modulus generator MG. □

Thus we prove that the above transitive signature is secure against adaptive chosen message attack though the underlying standard signature is only secure against non-adaptive chosen message attack. The reason is that the adaptive queries from the adversary are not directly fed to the standard signature simulator S_{sds}, but intercepted by the transitive signature simulator S_{ts} first, the resulting queries of standard signatures are generated by S_{ts}, which can be done adaptively or non-adaptively.

Since the security requirement of the underlying signature is relaxed, there are more signature schemes to choose from for a transitive signature, i.e., signature schemes secure against non-adaptive chosen message attack. Generally these signature schemes are more efficient than the ones provable secure in the strongest sense. In the following section, we provide such an efficient signature scheme that provable secure under non-adaptive chosen message attack in standard model.

4 A Specific Transitive Signature Scheme

4.1 A Standard Signature Scheme

We first introduce a standard signature that is provable existentially unforgeable against non-adaptive chosen message attack in standard model, and more efficient than most known provable secure signature schemes against adaptive chosen message attack, such as Cramer-Shoup's scheme [4] and Fischlin's scheme [5]. This standard signature will be used to construct a specific transitive signature conforming to the general scheme described above.

Scheme 1 *A standard signature scheme*

- *Key Generation: choose a RSA modulus $n = pq$, where $p = 2p' + 1, q = q' + 1$, p', q' are primes; choose random $x, h \in QR_n$, QR_n denotes the quadratic residue modulo n; choose a collision resistant hash function $H : \{0,1\}^* \rightarrow \{0,1\}^k$; the public key is (n, h, H), secret key is (p, q, p', q').*
- *Signature Generation: to sign message m, choose a random $(k+1)$ bit prime e, calculate y from*
$$y^e = xh^{H(m)} \bmod n$$
The signature of m is (e, y).
- *Verification: to verify a putative message signature pair (m, e, y), first check if e is an odd $(k + 1)$ bits number, then check if $y^e = xh^{H(m)} \bmod n$.*

This signature scheme can be proved secure against chosen message attack in a way similar to Cramer-Shoup's scheme[4], but its security against adaptive attack remains unknown so far.

4.2 Security Proof under Non-adaptive Chosen Message Attack

Initialization: Given a modulus n without knowing its factors, and a random $z \in \mathbb{Z}_n^*$, we want to find a pair (r, y) satisfying $y^r = z \bmod n$. In non-adaptive chosen message attack, the queries of chosen messages are done at one time. Simulator $\mathcal{S}$ randomly chooses Q primes that are $(k + 1)$ bits long: $e_1, e_2, ..., e_Q$. Suppose $\mathcal{F}$ output forgery message-signature pair (m, e, y). Then there are two kinds of possible forgeries:

1. $e \neq e_k$, for all $k \in [1, Q]$
2. $e = e_t, \exists t \in [1, Q]$

$\mathcal{S}$ just guesses which kind of forgery will come out, then construct public keys accordingly.
Case 1: $e \neq e_k$, for all $k \in [1, Q]$, we denote this event as E_1.

Set $h = z^{2\prod_k e_k}$, $x = h^a$, $a \in_R \{1, ..., n^2\}$. On queries of signatures for $(m_1, m_2, ..., m_Q)$ from $\mathcal{F}$, $\mathcal{S}$ does as follows:

$$y_i = z^{2\prod_{k\neq i} e_k(a+H(m))} \tag{3}$$

Return $(e_i, y_i), i = 1, ..., Q$ to $\mathcal{F}$.
$\mathcal{F}$ then outputs forgery message-signature pair (m, e, y). So

$$\begin{aligned} y^e &= xh^{H(m)} \bmod n \\ &= h^{a+H(m)} \bmod n \\ &= z^{2\prod_k e_k(a+H(m))} \bmod n \end{aligned} \tag{4}$$

It is easy to prove that $\gcd(e, 2\prod_k e_k(a + H(m))) = 1$ with high probability, so we can solve the strong RSA problem according to Lemma 1; otherwise, $\mathcal{S}$ aborts.

Case 2: $e = e_t, \exists t \in [1, Q]$, we denote this event as E_2.
In this case, $\mathcal{S}$ guesses t, and sets

$$\begin{aligned} h &= z^{2\prod_{k\neq t} e_k} \\ y_t &= w^{2\prod_{k\neq t} e_k}, w \in_R \mathbb{Z}_n^*, \\ x &= {y_t}^{e_t} h^{-H(m_t)} \bmod n \end{aligned} \tag{5}$$

On queries of signatures for $(m_1, m_2, ..., m_Q)$ from $\mathcal{F}$, $\mathcal{S}$ does as follows:

```
if $i = t$ then
  $\sigma(m_i) = (e_t, y_t)$
else if $i \neq t$ then
  $y_i = w^{2\prod_{k\neq i} e_k} \cdot z^{2\prod_{k\neq t,i} e_k(H(m_i)-H(m_t))}$
  $\sigma(m_i) = (e_i, y_i)$
end if
```

Then we get the following two equations:

$$\begin{aligned} {y_t}^{e_t} &= xh^{H(m_t)} \bmod n \\ y^e &= xh^{H(m)} \bmod n \end{aligned} \tag{6}$$

$\mathcal{S}$ aborts if its guess is not right, else i.e., $e_t = e$, we get

$$\begin{aligned} (y_t y^{-1})^e &= h^{H(m_t)-H(m)} \bmod n \\ &= z^{2\prod_{k\neq t} e_k(H(m_t)-H(m))} \bmod n \end{aligned} \tag{7}$$

Because e_t is co-prime to $2\prod_{k\neq t} e_k$ and the length $|e_t| > |H(\cdot)|$, so

$$\gcd(e_t, 2\prod_{k\neq t} e_k(H(m_t) - H(m))) = 1,$$

then we can solve the strong RSA problem according to Lemma 1. □

But it is not sure whether this scheme has security proof against adaptive chosen message attack so far. This scheme has simpler form than Cramer-Shoup's scheme [4] and Fischlin's scheme [5], which means more efficiency in calculation. In the following table, we give a simple compare of the efficiency between the standard signatures mentioned. We compare the efficiency in terms of computations number, such as the computations of exponent (denoted as exp.), root extraction (denoted as root extr.), and the resulting signature size.

	Sign	Verify	Signature Size
Cramer-Shoup's[4]	3 exp.,1 root extr.	4 exp.	$k + 2\log n$
Fischlin's[5]	2 exp.,1 root extr.	3 exp.	$2k + \log n$
This Scheme	1 exp.,1 root extr.	2 exp.	$k + \log n$

By the table above, we do not mean to show that we have contributed a better signature scheme, because they provide different security levels. The scheme described has better efficiency because we only require it provably secure against non-adaptive chosen message attack. What we show is that there really exists such a standard signature that can be used in constructing provably secure transitive signature schemes, by providing more efficiency.

In the following section we describe the specific transitive signature scheme combined with the standard signature introduced here.

4.3 Transitive Signature Scheme

The transitive signature scheme is defined as follows:

- TKG: a probabilistic algorithm, given input 1^k,1^l output $tpk = (n, x, h, H)$ and $tsk = (p, q)$, where n is a RSA modulus with p, q as its prime factors, $x, h \overset{R}{\leftarrow} QR_n$, $H : \{0,1\}^* \rightarrow \{0,1\}^k$ is a collision resistant hash function.
- TSign: TSign maintains state (V, l, C), where definition of V, l and C are the same as in the proposed general scheme. Elements of l are kept secret. When invoked on inputs (tsk, tpk, i, j), that is when asked to produce a signature on edge (i, j), TSign does:

```
**if** $j < i$ **then**
  swap $i, j$
**else if** $i \notin V$ **then**
  $V \leftarrow V \cup i$; $l(i) \overset{R}{\leftarrow} \mathbb{Z}_n^*$; calculate $x_i = l(i)^2$; select a $(k+1)$ bits prime
  $e_i$ randomly, calculate $y_i \in Z_n$ from ${y_i}^{e_i} = xh^{H(x_i)} \bmod n$;
**else if** $j \notin V$ **then**
  $V \leftarrow V \cup j$; $l(j) \overset{R}{\leftarrow} \mathbb{Z}_n^*$; calculate $x_j = l(j)^2$; select a $(k+1)$ bits prime
  $e_j$ randomly, calculate $y_j \in Z_n$ from ${y_j}^{e_j} = xh^{H(x_j)} \bmod n$;
**end if**
set $\delta_{ij} = l(i)l(j)^{-1} \bmod n$, $C_i = (i, e_i, y_i, x_i)$, $C_j = (j, e_j, y_j, x_j)$.
```

The resulting transitive signature of edge (i, j) is (δ_{ij}, C_i, C_j).

- TVf: given input (δ_{ij}, C_i, C_j), parses C_i as (i, e_i, y_i, x_i), C_j as (j, e_j, y_j, x_j), then does the following:

```
**if** $j < i$ **then**
```

```
  swap $i, j$
end if
if $|e_i| = |e_j| = k+1$ and ${y_i}^{e_i} = xh^{H(x_i)}$mod and ${y_j}^{e_j} = xh^{H(x_j)}$ mod $n$
and $\delta_{ij}^2 = x_i {x_j}^{-1}$ mod $n$ then
  returns 1
else
  returns 0
end if
```

The verifier accepts the signature if return value of TVf is 1, rejects it if returns 0.

- Comp: given valid transitive signatures (δ_{ij}, C_i, C_j) and (δ_{jk}, C_j, C_k), suppose $i < j < k$, if not so, we can swap the sequence of i, j, k. Calculates $\delta_{jk} = \delta_{ij}\delta_{jk}$. The correctness of Comp can be verified by $\delta_{ik}^2 = x_i {x_k}^{-1}$ mod n.

The security proof can be done similarly to the proof of general scheme.

Compared with other transitive signature schemes that utilize a standard signature to generate the node certificate, this scheme is more efficient, as mentioned in section 4.2.

There are also transitive signature schemes that avoid node certificate from standard signature. As the schemes proposed in [2]: FBTS-2, RSATS-2. The nodes are presented by the hash values of node index, which can be verified publicly, hence avoid node certification, i.e., avoid employing another standard signature scheme. They are even more efficient than our scheme. But they are provable secure only in random oracle model so far, while the security proof of our proposed scheme is done in standard model.

5 Conclusions

In this paper, we have pointed out that the underlying standard signature is not necessary to be secure against adaptive chosen message attack for the transitive signature scheme to be secure against adaptive attack, therefor the choice of candidate standard signature is larger. In particular, we have constructed a transitive signature scheme that is secure against adaptive attack, by utilizing a standard signature scheme that is efficient and provably secure against non-adaptive chosen message attack while its security against adaptive attack is unknown.

References

1. S. Micali and R. L. Rivest, "Transitive signaure schemes," in *Topics in Cryptology - CT-RSA'02*, LNCS 2271, pp. 236–243, Springer, 2002.
2. M. Bellare and G. Neven, "Transitive signaures based on factoring and RSA," in *Advances in Cryptology - ASIACRYPT'02*, LNCS 2501, pp. 391–414, Springer, 2002. Full version on http://www-cse.ucsd.edu/users/mihir.

3. S. Goldwasser, S. Micali, and R. L. Rivest, “A digital signature scheme secure against adaptive chosen-message attacks,” *SIAM J. Computing*, vol. 17, pp. 281–308, Apr. 1988.
4. R. Cramer and V. Shoup, “Signature schemes based on the strong RSA assumption,” in *Proc. of the 6th ACM Conf. on Computer and Communications Security(CCS '99)*, pp. 46–52, ACM Press, 1999.
5. M. Fischlin, “The Cramer-Shoup Strong-RSA signature scheme revisited,” in *Public Key Cryptography -PKC' 03*, LNCS 2567, pp. 116–129, Springer, 2003.