A New Transitive Signature Scheme based on RSA-based Security Assumptions

Dang Nguyen Duc, Han Kyusuk, Zeen Kim and Kwangjo Kim

International Research center for Information Security (IRIS)
Information and Communications University (ICU)
119 Munjiro, Yuseong-gu
Daejon, 305-732, Korea
{nguyenduc, hankyusuk, zeenkim, kkj}@icu.ac.kr

Abstract. A transitive signature scheme allows a signer to publish a graph in an authenticated and cost-saving manner. The resulting authenticated graph is indeed the transitive closure of the graph constructed by edges which are explicitly signed by the signer. A property of the transitive signature scheme enables such scenario is called composability which means that by knowing signatures on two edges of a triangle, one can infer to a valid signature on the other edge of the triangle without knowledge of the signer's secret key thereby saving the signer from signing one signature. Several transitive signature schemes have been proposed so far [1–3]. Their security assumptions are based on the intractability of computing discrete logarithm, inverting RSA function, factoring and solving Diffie-Hellman problem. In this paper, we will present another transitive signature scheme based the Guillou-Quisquater (GQ for short) signature scheme. The security of our proposed can be proven under the assumption that solving the strong RSA problem is hard in case of non-adaptive chosen-message attack. In case of adaptive chosen-message attack, similar to Bellare and Neven's work [2, 3], we can show that breaking our scheme is as hard as solving the one-more-RSA inversion problem. [1]

Key words: Transitive signature scheme, provable security, strong RSA assumption, one-more-RSA-inversion assumption, chosen-message attack.

1 Introduction

1.1 The basic concept

Graph, consisting of vertices and edges, is a very common data structure to represent relations between objects. For example, a graph can be used to represent a computer network, some organization structure, *etc.* In many scenarios, one needs to publish a graph representing some structure in an authenticated (and efficient) manner. In 2002, Micali and Rivest proposed such a solution for signing a graph called transitive signature [1]. The name "*transitive*" comes from the fact that, at any time, the actual authenticated graph is the transitive closure of the graph whose edges are signed explicitly by the signer. Therefore, to publish a graph in an authenticated manner, the signer just needs to sign a sub-graph of the original graph as long as this sub-graph preserves the connectivity of the graph. It is because given a same vertex set, two connected graphs have the same transitive closure. Considering the fact that a graph in practice is often complicated and transitively closed, this

[1] Some parts of this work were presented at the Symposium for Cryptography and Information Security (SCIS) 20005 in Kobe, Japan.

is much more efficient way to sign a graph. One special property of a transitive signature scheme which enables such behavior is that it allows composition of signatures. More specifically, if we denote an edge on a graph as $\{i, j\}$ where i and j are vertex indexes, then, given two signatures on edge $\{i, j\}$ and edge $\{j, k\}$, without the secret key of the signer, one can produce a valid signature on edge $\{i,\ k\}$. Like any standard signature scheme, a transitive signature scheme must be unforgeable under the strongest type of attack, namely chosen-message attack. However, in case of transitive signature schemes, composability can be seen as a type of forgery because it does not need the signer's secret key to function. Therefore, for a transitive signature scheme, composition of signatures is required to be the only possible type of forgery. If a transitive signature scheme satisfies such security requirement, we say that it is transitively unforgeable. Another requirement for a transitive signature scheme mentioned by Micali and Rivest [1] is for privacy purpose. This requirement states that signatures obtained via composition procedure should be indistinguishable from signatures explicitly signed by the signer. It is true that in practice, if one finds that a given signature is not produced by the original signer, he might not accept it even though that signature is a valid one. Bellare and Neven argued that this is not necessary a security requirement but a "*correctness*" requirement of the composibility feature [2, 3].

1.2 Potential applications

Beside the motivated application mentioned in [1] in which a transitive signature scheme is used to sign the relationship between an officer and his immediate supervisor, we describe an application of a transitive signature scheme in managing trusts in a distributed system. Let's suppose that we have a single administrative domain with n nodes and every node trusts each other. There is a super node in charge of authenticating trustworthiness between nodes in the domain. If we use a standard signature scheme for the super node to authenticate trusts between nodes, then it has to produce signatures for every pair of nodes (more specifically, $\frac{n(n-1)}{2}$ signatures) which is considerably expensive. If we consider the domain as a graph where vertices are all nodes in the domain and edges represents trust between two nodes, then, this graph is clearly transitively closed (even complete) because every node trusts each other. From this observation, we can use a transitive signature scheme for the super node to sign only $n - 1$ signatures corresponding $n - 1$ edges forming a sub-graph that preserves the connectivity of the original graph. This is a very significant cost saving for the super node. We can also see that transitive signature schemes capture the transitivity nature of trust relationship, *i.e.,* if A trusts B and B trusts C, then it is reasonable that A also trusts C.

1.3 Our contribution

It is a common practice in cryptography that one should find alternative solutions for the same problem to seek performance gain, additional properties and probably new insights. For realizing the transitive signature concept, four security assumptions have been used. They include the intractability assumptions of RSA inversion, computing discrete logarithm, factoring and solving Diffie-Hellman problem [1–3]. In this paper, we present a new transitive signature scheme based on a Guillou-Quisquater (GQ for short) signature scheme [8]. Our proposed scheme is proven

to be secure against non-adaptive chosen-message attack under the strong RSA assumption [11, 12]. Also, similar to [2, 3], we can also prove the security of our scheme in case of adaptive chosen-message attacks assuming that the one-more-RSA inversion problem [13] is hard. Even though our proposed scheme does not provide any performance or security gain, it shows a further (although weak) evidence that one-way trapdoor permutation is not enough to construct a secure transitive signature scheme [10].

2 Background and Definitions

2.1 Some Terminologies in Graph Theory

Transitive signature schemes target signing a graph. Therefore, we briefly recall some related terminologies in graph theory as follows:

- A graph G consists of two sets, a vertex set V and a set edges $E = \{\{i, j\} : i, j \in V\}$. G is called an undirected graph if the edge $\{i, j\}$ is identical to the edge $\{j, i\}$. For the sake of simplicity, we assume that vertex index, i, is a positive integer (*i.e.,* $V \subset N$). Wlog, we also assume that an undirected edge from vertex i to vertex j, denoted as $\{i, j\}$, implies $i < j$.
- A graph is said to be connected if there is a path between any pair of vertices. An arbitrary graph $G = (V, E)$ can be divided into connected sub-graphs $G' = (V', E')$ where $V' \subset V$ and $E' \subset E$.
- A graph is said to be transitively closed if there is a path between two vertices, then there is an edge between them.
- The transitive closure of a graph $G = (V, E)$ is a graph $G' = (V, E')$ such that if there is a path from i to j in G, then $\{i, j\} \in E'$. It is easy to see that two connected graphs with the same vertex set result in the same transitive closure.

All graphs in this paper are undirected. It is still open to construct a transitive signature scheme for directed graphs [10].

2.2 Formalization of Transitive Signature Scheme

We follow the formalization given by Bellare and Neven in [2] which is the first and sound one after the introduction of the transitive signature concept by Micali and Rivest [1]. First of all, we give a formal definition of a transitive signature scheme according to [2].

Definition 1. *A transitive signature scheme $\mathcal{TS}$ consists of four polynomial-time algorithms described as follows:*

- `TKG` *is a randomized key generation algorithm which the security parameter k as its input and produces a key pair (tpk, tsk) including the public key tpk and the corresponding secret key tsk.*
- `TESign` *is an edge signing algorithm which takes the secret key tsk and two vertices i and j as its input and outputs a signature on edge $\{i, j\}$, σ_{ij}.* `TESign` *can be stateful.*

– `TEVf` *is a deterministic edge signature verification algorithm. Given the public key tpk, two vertices i and j and a candidate signature on edge* $\{i, j\}$, σ, `TEVf` *outputs 'accept' if* σ *is a valid signature on edge* $\{i, j\}$ *relative to tpk. Otherwise, it outputs 'reject'.*
– `TComp` *is also a deterministic algorithm.* `TComp` *takes the public key tpk, three vertices* i, j *and* k*, and two signatures* σ_1 *and* σ_2 *on edges* $\{i, j\}$ *and* $\{j, k\}$*, respectively, as its input and outputs either a valid signature on edge* $\{i, k\}$ *or a symbol of failure,* $\perp$.

The first three components of a transitive signature scheme are very similar to those of a standard signature scheme. However, regarding the `TComp` algorithm, there are several subtle matters we should consider. They are:

– `TComp` should work properly even though the input signatures are not signatures obtained via the edge signing algorithm, `TESign` (instead, it can be any valid one obtained via `TComp` algorithm itself).
– `TComp` should guarantee that signatures obtained via composition and `TESign` are indistinguishable.

Taking the above considerations into account, Bellare and Neven formally defined a notion called "*correctness*" of the composition algorithm [2, 3]. The definition is achieved via an experiment in which an adversary $\mathcal{A}$ (not necessary computationally bounded) fails to fool the composition algorithm. We refer interested readers to [2, 3] for more details.

As being mentioned in the definition, the edge signing algorithm `TESign` can be stateful. It is because `TESign` needs to remember the state of a graph whose edges have been signed. In particular, it is common in previous transitive signature schemes [1–3] that each vertex is associated with two labels, a secret one and a public one, throughout the lifetime of the system. Also, the public label is required to be certified. This can be done by employing a standard signature scheme to sign the public label. The stateful nature of `TESign` can be avoided by enabling the signer to recompute vertex labels whenever required [2, 3].

We now shall define what we mean by saying that a given transitive signature scheme is secure. As usual, we shall consider the strongest kind of adversary called chosen message-attack adversary, say $\mathcal{F}$. Similar to [2, 3], the security of a scheme is defined via an experiment in which $\mathcal{F}$ with its signing oracle, $\texttt{TESign}(tsk, ., .,)$, attempts to forge a valid signature. We denote the experiment given the security parameter $k \in N$ as $\mathbf{Exp}^{\texttt{tu}-\texttt{cma}}_{\mathcal{TS},\texttt{F}}(k)$. In the experiment, after executing the key generation procedure to generate the key pair (tpk, tsk), the signing oracle $\texttt{TESign}(tsk, ., .,)$ is made available to $\mathcal{F}$. $\mathcal{F}$ makes queries to the signing oracle (in an adaptive or non-adaptive manner) with two distinct vertices i and j per query. Let E be the set of all pairs $\{i, j\}$ such that $\mathcal{F}$ made oracle query i, j and let V be the set of all vertices appeared in E (the cardinality of V should also be upper bounded by some polynomial). Eventually, $\mathcal{F}$ will produce two vertices i', j' and a forged signature σ'. The experiment will return 1 if $\texttt{TEVf}(tpk, i', j', \sigma')$ returns 'accept' and edge $\{i', j'\}$ is not in the transitive closure of the graph $G = (V, E)$. Otherwise, the experiment returns 0. We are now ready to define security of a transitive signature scheme as follows:

Definition 2. *A transitive signature scheme is said to be secure or transitively unforgeable under chosen-message attack if given any polynomial-time (polynomial in*

security parameter) adversary $\mathcal{F}$, the below quantity (called advantage of $\mathcal{F}$) is negligible in the security parameter [2]*:*

$$\mathbf{Adv}^{\mathtt{tu-cma}}_{\mathcal{TS},\mathtt{F}}(k) = \Pr[\mathbf{Exp}^{\mathtt{tu-cma}}_{\mathcal{TS},\mathtt{F}}(k) = 1]$$

As mentioned earlier, a standard digital signature scheme (denoted as $\mathcal{SDS}$=(SKG,SSign,SVerify) where SKG is a key generation algorithm, SSign is a signing algorithm and SVerify is verification algorithm) is required to produce vertex certificates (*i.e.,* signatures on vertex public labels). The security of $\mathcal{SDS}$ certainly contributes to the security the transitive signature scheme which makes use of $\mathcal{SDS}$. Like in [6], we define the security strength of $\mathcal{SDS}$ as the successful probability or advantage (as a function of security parameter k) of a chosen-message attack adversary $\mathcal{B}$, $\mathbf{Adv}^{\mathtt{uf-cma}}_{\mathcal{SDS},\mathtt{B}}(k)$. We say that $\mathcal{SDS}$ is secure (unforgeable) under chosen-message attack if $\mathbf{Adv}^{\mathtt{uf-cma}}_{\mathcal{SDS},\mathtt{B}}(k)$ is negligible for every polynomial-time adversary $\mathcal{B}$.

2.3 The Strong RSA Assumption

A variant of the standard RSA assumption (*i.e.,* RSA function is one-way) were introduced in [11, 12] called the strong RSA assumption. Intuitively speaking, the strong RSA assumption states that given a RSA modulus N and a value $\alpha \in Z^*_N$, it is infeasible to find $\beta \in Z^*_N$ and an integer number r such that $\beta^r = \alpha \bmod N$. In this paper, we are interested in a class of the strong RSA assumption where N is the product of two safe primes [3]. Suppose that N is k-bit long, we define the strong RSA assumption by saying that the successful probability of any polynomial-time strong RSA problem solver $\mathcal{A}$, $\mathbf{Adv}^{\mathtt{s-rsa}}_{\mathtt{A}}(k, l)$, is negligible. We state a relevant lemma which we will use in our security proof as follows:

Lemma 1. *Let G be a finite group. Suppose that e_1 and e_2 are two integers such that gcd(e_1, e_2) = g and gcd($g, |G|$) = 1. Given a and $b \in G$ such that $a^{e_1} = b^{e_2}$, one can compute c such that $c^{\frac{e_2}{g}} = a$ in $O(\log \frac{e_1+e_2}{g})$ group operations.*

Proof. A proof of this lemma is given in [7].

2.4 The One-More-RSA Inversion Assumption

The one-more-RSA inversion problem was introduced in [13]. The problem setting is given as follows: an adversary $\mathcal{A}$ is equipped with two oracles, CHALL(.) and INV(.) where CHALL(.) returns a random element in Z^*_N (N is product of two primes) and INV(.) inverts RSA function with respect to the RSA modulus N and e of RSA public key (*i.e.*, return $x^{e^{-1}} \bmod N$ on input x). $\mathcal{A}$'s job is to compute RSA inversion of all k challenges returned by CHALL(.) by asking strictly less k times the RSA inversion oracle, INV(.). The one-more-RSA inversion assumption states that the chance for $\mathcal{A}$ to succeed is negligible.

[2] A function $f(k)$ is said to be negligible if it is upper bounded by the inverse of any positive polynomial $1/p(k)$ for sufficiently large k.

[3] A prime number $p = 2q + 1$ is safe if q is also prime. q is also known as Sophie Germain prime.

3 The New Scheme

We present our proposed transitive signature scheme as an extension of the ordinary GQ signature scheme [8]. We name our scheme as $\mathcal{SRSA}$–$\mathcal{TS}$. Like previous schemes, our scheme makes use of a standard digital signature scheme $\mathcal{SDS} = (\mathtt{SKG}, \mathtt{SSign}, \mathtt{SVerify})$. We now describe four components of $\mathcal{SRSA}$–$\mathcal{TS}$ as follows:

Key generation. The key generation algorithm `TKG`, given key parameters k and l, does the following:

1. Run `SKG` to generate a key pair (spk, ssk) for $\mathcal{SDS}$
2. Generate two $k/2$ bit safe primes p, q and compute $N \leftarrow pq$
3. Randomly choose s from Z_N^* and an $(l+1)$-bit odd integer e and compute $v \leftarrow 1/s^e \bmod N$.
4. Discard p, q and output $tpk = (N, e, v, spk)$ and $ssk = (N, e, s, ssk)$

Edge signature generation. The edge signing algorithm `TESign`, given the secret key tsk and two vertices i, j $(i < j)$, outputs a signature on edge $\{i, j\}$. `TESign` maintains its state which includes a vertex index set V, a vertex label table Δ and a vertex certificate table Σ (we refer $\Delta(i)$, $\Sigma(i)$ as the containers for labels and certificate of vertex i). It does the following:

1. **For** each t of the set $\{i, j\}$ **do**
2. **If** $t \notin V$ **then**
3. $V \leftarrow V \cup \{t\}$
4. Randomly choose a secret label $\ell(t)$ from Z_N^*
5. Compute the pubic label $L(t) \leftarrow \ell(t)^e \bmod N$
6. Generate vertex certificate $\Sigma(t) \leftarrow \mathtt{SSign}(ssk, t||L(t))$
7. Randomly choose another l-bit secret label x_t
8. $\Delta(t) \leftarrow (\ell(t), x_t, L(t))$
9. Compute $z_i \leftarrow \ell(i)s^{x_i} \bmod N$
10. Compute $z_j \leftarrow \ell(j)s^{x_j} \bmod N$
11. Compute $z \leftarrow z_i/z_j \bmod N$ and $x \leftarrow x_i - x_j$
12. Let $C_i \leftarrow (L(i), \Sigma(i))$ and $C_j \leftarrow (L(j), \Sigma(j))$
13. Output $\sigma_{ij} \leftarrow (C_i, C_j, z, x)$

Edge signature verification. The edge signature verification algorithm `TEVf`, given the public key spk, two vertices i, j $(i < j)$ and a candidate signature on edge $\{i, j\}$, σ, outputs either 'accept' or 'reject'. It does the following:

1. Parse σ as (C_i, C_j, z, x)
2. Parse C_i as $(L_i, \Sigma(i))$ and C_j as $(L_j, \Sigma(j))$
3. **If** $\mathtt{SVerify}(spk, i||L_i, \Sigma(i)) =$ 'reject' $\vee$ $\mathtt{SVerify}(spk, j||L_j, \Sigma(j)) =$ 'reject'
4. **then Return** 'reject'
5. **If** not $(|x| < 2^l)$ **then Return** 'reject'
6. **If** $z^e v^x \neq L_i/L_j \bmod N$ **then Return** 'reject' **Else Return** 'accept'

We can easily show that the edge signature verification algorithm always returns 'accept' if σ is a valid signature because $z^e v^x = \left(\ell(i)s^{x_i}\ell(j)^{-1}s^{-x_j}\right)^e (1/s^e)^{x_i - x_j} = \ell(i)^e \ell(j)^{-e} = L(i)/L(j) \bmod N$.

Signature Composition. The signature composition algorithm `TComp` takes three vertices i, j, k $(i < j < k)$ and two signatures σ_1, σ_2 as its input and does the following:

1. **If** $\texttt{TEVf}(tpk, i, j, \sigma_1) =$ 'reject' $\vee$ $\texttt{TEVf}(tpk, j, k, \sigma_2) =$ 'reject'
2. **then Return** $\perp$
3. Parse σ_1 as (C_i, C_j, z, x) and σ_2 as (C_j, C_k, z', x')
4. Output $\sigma_{ik} \leftarrow (C_i, C_k, zz' \bmod N, x + x')$

It is intuitive to see that the correctness of $\texttt{TComp}$ is satisfied since $zz' = (z_i/zj)(z_j/z_k) = z_i/z_k \bmod N$ and $x + x' = x_i - x_j + x_j - x_k = x_i - x_k$. The two values zz' and $x + x'$ are the same values that the real signer would produce himself for a valid signature on edge $\{i, k\}$. For more rigorous correctness proof of $\texttt{TComp}$, please see Bellare and Neven's papers [2, 3].

Implementing key-evolving protocol. Since our proposed scheme and the Itkis-Reyzin forward-secure signature scheme [7] are both based on the GQ signature scheme (*e.g.,* secret key does not contain information about factors of N). It is possible to adapt their key-evolving protocol to our scheme so that our scheme can provide forward secrecy.

4 Security Analysis

In this part, we sometimes use the two terminologies signature and edge interchangeably as an edge is assumed to exist if and only if it is authenticated by a signer's signature. We state the following two theorems regarding the security of our proposed transitive signature scheme.

Theorem 1. *If the strong RSA assumption holds and $\mathcal{SDS}$ is unforgeable under chosen-message attack, then the $\mathcal{SRSA}$–$\mathcal{TS}$ scheme is transitively unforgeable under non-adaptive chosen-message attack.*

Proof. Suppose that we are given a polynomial-time adversary $\mathcal{F}$ to attack $\mathcal{SRSA}$–$\mathcal{TS}$. It is desirable to show that, for all security parameter k, l, the following inequality holds

$$\mathbf{Adv}^{\texttt{tu-cma}}_{\mathcal{SRSA}-\mathcal{TS},\mathrm{F}}(k, l) \leq c_1 \mathbf{Adv}^{\texttt{s-rsa}}_{\mathrm{A}}(k, l) + c_2 \mathbf{Adv}^{\texttt{uf-cma}}_{\mathcal{SDS},\mathrm{B}}(k)$$

where c_1 and c_2 are two constants or upper bounded by some polynomial (in security parameter). This inequality says that if the right hand side is negligible, so does the left hand side which proves the theorem. In this proof, we consider only the case that $\mathcal{F}$ is non-adaptive meaning that $\mathcal{F}$, given the public key tpk, prepares in advance its queries to the signing oracle. Suppose that $\mathcal{F}$'s queries forms a graph $G = (V, E)$. After querying the signing oracle for signatures on edges of the graph G, $\mathcal{F}$ outputs a forged signature on edge $\{i', j'\}$ such that $\{i', j'\}$ is not on the transitive closure of $\mathcal{F}$. Let $\mathbf{E}$ be the event both i' and j' are in V. We have:

$$\begin{aligned}\mathbf{Adv}^{\texttt{tu-cma}}_{\mathcal{SRSA}-\mathcal{TS},\mathrm{F}}(k, l) &= \Pr[\mathbf{Exp}^{\texttt{tu-cma}}_{\mathcal{SRSA}-\mathcal{TS},\mathrm{F}}(k, l) = 1] \\ &= \Pr[\mathbf{Exp}^{\texttt{tu-cma}}_{\mathcal{SRSA}-\mathcal{TS},\mathrm{F}}(k, l) = 1 \wedge \mathbf{E}] \\ &\quad + \Pr[\mathbf{Exp}^{\texttt{tu-cma}}_{\mathcal{SRSA}-\mathcal{TS},\mathrm{F}}(k, l) = 1 \wedge \overline{\mathbf{E}}]\end{aligned}$$

In case of the event $\overline{\mathbf{E}}$ (either i' or j' is not in V), $\mathcal{F}$ needs at least one forged vertex certificate as vertex certificate is included into edge signature. Therefore, in this case, we can construct an adversary $\mathcal{B}$ attacking $\mathcal{SDS}$ which is used to produce vertex certificates. This leads to:

$$\Pr[\mathbf{Exp}^{\texttt{tu-cma}}_{\mathcal{SRSA}-\mathcal{TS},\mathrm{F}}(k, l) = 1 \wedge \overline{\mathbf{E}}] \leq \mathbf{Adv}^{\texttt{uf-cma}}_{\mathcal{SDS},\mathrm{B}}(k). \quad (1)$$

We now construct an adversary $\mathcal{A}$ attacking the strong RSA assumption in order to evaluate $\Pr[\mathbf{Exp}^{\mathtt{tu-cma}}_{\mathcal{SRSA}-\mathcal{TS},\mathrm{F}}(k,l)=1 \wedge \mathbf{E}]$. Let's recall $\mathcal{A}$'s job: given N and $\alpha \in Z_N^*$, find $\beta \in Z_N^*$ and an integer r such that $\beta^r = \alpha \bmod N$. We now describe $\mathcal{A}$ in detail. $\mathcal{A}$ first needs to generate tpk for $\mathcal{F}$. $\mathcal{A}$ does so by assigning $v = \alpha$ and generates e, spk and ssk as the real signer. $\mathcal{A}$ then run $\mathcal{F}$ with the input $tpk = (N, v, e, spk)$. When receiving $\mathcal{F}$'s queries which forms the graph G, $\mathcal{A}$ answers the queries as follows:

- $\mathcal{A}$ firstly divides G into a set of disjoint sub-graphs $G' = (V', E')$ $(V' \subset V, E' \subset E)$ such that each G' is connected and signs each G' separately.
- $\mathcal{A}$ does not need to sign exactly all edges of G', it can sign any other set of edges E'' as long as (V', E'') also forms a connected graph. It is because the transitive closures of G' and G'' are the same, therefore, by signing the set of edges E'', $\mathcal{A}$ can infer signatures on edges belonging to E' using signature composition.

We now show that $\mathcal{A}$ can produce signature on edges of the graph G' without knowing s of the secret key as follows: $\mathcal{A}$ first chooses a reference vertex, say vertex i, from V'. To sign the edge $\{i, j\}$ for all other vertices $j \in V'$, $\mathcal{A}$ randomly chooses $z \in Z_N^*$ and an integer x such that $|x| < 2^l$. $\mathcal{A}$ also chooses a secret label $\ell(i)$ for vertex i at random from Z_N^*. It then computes public labels of two vertices

$$L(i) = \ell(i)^e \bmod N \text{ and } L(j) = L(i)/z^e v^x \bmod N$$

and uses $\mathcal{SDS}$ to produce vertex certificates:

$$\Sigma(i) = \mathtt{SSign}(ssk, i||L(i)) \text{ and } \Sigma(j) = \mathtt{SSign}(ssk, j||L(j))$$

Finally, $\mathcal{A}$ returns a valid signature on edge $\{i, j\}$ as $((L(i), \Sigma(i)), (L(j), \Sigma(j)), z, x)$ to $\mathcal{F}$. This signature is valid because $z^e v^x = L(i)/L(j) \bmod N$.

We have just shown that $\mathcal{A}$ can always answer $\mathcal{F}$'s queries as long as $\mathcal{F}$ is non-adaptive. Suppose that after the querying phase, $\mathcal{F}$ outputs a forged signature on edge $\{i', j'\}$ as $\sigma' = ((L(i'), \Sigma(i')), (L(j'), \Sigma(j')), z', x')$ such that the edge $\{i', j'\}$ is not on the transitive closure of the graph G formed by all $\mathcal{F}$'s queries. In case of the event $\mathbf{E}$ (certificates of i' and j' are reused), i' and j' are in V. Since $\{i', j'\}$ is not the transitive closure of G, then, i' and j' must be on two different disjoint connected sub-graphs of G. As we have shown earlier, for each disjoint connected sub-graph of G, $\mathcal{A}$ needs to generate one secret label of a vertex in that sub-graph. Therefore, with probability at least $\frac{1}{m}$ where m is the total number of vertices involved in querying phase of $\mathcal{F}$, $\mathcal{A}$ knows $\ell(i')$ and with probability $\frac{1}{m} \cdot \frac{1}{m} = \frac{1}{m^2}$, $\mathcal{A}$ knows both $\ell(i')$ and $\ell(j')$. If σ' is a valid signature on edge $\{i', j'\}$, then, the following equality holds:

$$\begin{aligned} z'^e v^{x'} &= L(i')L(j')^-1 = \ell(i')^e \ell(j')^{-e} \bmod N \\ \Rightarrow \quad v^{x'} &= \left(\frac{\ell(i')}{\ell(j')z'}\right)^e \bmod N \end{aligned}$$

Because $|x'| < 2^l$ is enforced by the edge signature verification procedure and e is $(l+1)$-bit long, then $e > |x'|$ and $\gcd(e, x') = g$ is less than e. Let $r = e/g$, then $r > 1$. Note that it is likely that $\gcd(e, \phi(N)) = 1$ since N is product of two safe primes. To see that, suppose $N = pq = (2p'+1)(2q'+1)$ and $\phi(N) = 4p'q'$ where p, q, p', q' are all prime. Because $\mathcal{A}$ picks e as a $(l+1)$-bit odd integer, it is likely that $\gcd(e, 4p'q') = 1$. As a result, $\gcd(g, \phi(N))$ is also 1. As we know, $\phi(N)$ is the order

of the multiplicative group Z_N^*, following the **Lemma 1**, $\mathcal{A}$ can efficiently compute r-th root of v which is its target α. So, with probability 1/4 and in case of the event $\mathbf{E}$, if $\mathcal{F}$ succeeds in forging a valid signature, $\mathcal{A}$ can solve the strong RSA problem. This implies:

$$\frac{1}{m^2} \cdot \Pr[\mathbf{Exp}^{\mathtt{tu-cma}}_{\mathcal{SRSA-TS},\mathrm{F}}(k,l) = 1 \wedge \mathbf{E}] \leq \mathbf{Adv}^{\mathtt{s-rsa}}_{\mathrm{A}}(k,l) \tag{2}$$

Combine (1) and (2), we achieved the desirable inequality:

$$\mathbf{Adv}^{\mathtt{tu-cma}}_{\mathcal{SRSA-TS},\mathrm{F}}(k,l) \leq m^2 \mathbf{Adv}^{\mathtt{s-rsa}}_{\mathrm{A}}(k,l) + \mathbf{Adv}^{\mathtt{uf-cma}}_{\mathcal{SDS},\mathrm{B}}(k).$$

Since m should be upper bounded by some polynomial, so does m^2. Therefore, we achieve our proof.

□

The above strategy to construct $\mathcal{A}$ does not work in case $\mathcal{F}$ is adaptive because $\mathcal{A}$ does not know all $\mathcal{F}$'s queries before answering them. If $\mathcal{A}$ attempts to use the same strategy for adaptive $\mathcal{F}$, the chance that $\mathcal{A}$ can answer each $\mathcal{F}$'s query is 1/2. Therefore, if $\mathcal{F}$ asks q_{sig} queries, the successful probability of $\mathcal{A}$ will be proportional to $(1/2)^{q_{sig}}$ which is obviously infeasible.

After failing to prove the security of our scheme in adaptive adversary case under the strong RSA assumption, we found that the technique employed by Bellare and Neven [2, 3] (which uses the one-more-RSA-inversion assumption) also worked for our scheme. We present here a proof of security of our scheme in case of adaptive chosen-message adversary using their idea but in a little more intuitive manner [4]

Theorem 2. *If the one-more-RSA inversion assumption holds and $\mathcal{SDS}$ is unforgeable under chosen-message attack, then the $\mathcal{SRSA-TS}$ scheme is transitively unforgeable under adaptive chosen-message attack.*

Proof. Similar to the proof of **Theorem 1**, we also consider two types of the forger $\mathcal{F}$. We will describe only the use of the second type of the forger $\mathcal{F}$ (reusing vertex certificates) to violate the one-more-RSA inversion assumption. As in [2, 3], the main idea of constructing an adversary $\mathcal{A}$ to attack the one-more-RSA inversion assumption is to assign all challenges returned by `CHALL`(.) to vertex public labels. By doing so, $\mathcal{A}$ can answer all signature queries of the adaptive forger $\mathcal{F}$ as follows: whenever $\mathcal{F}$ ask for a signature on edge $\{i, j\}$, $\mathcal{A}$ do the following:

- $\mathcal{A}$ first checks whether a signature on edge $\{i, j\}$ can be obtained via composition (of signatures previously asked by $\mathcal{F}$.
- If $\mathcal{A}$ cannot answer $\mathcal{F}$'s query using signature composition (*i.e.,* $\{i, j\}$ are not on the transitive closure the graph formed by signatures previously asked by $\mathcal{F}$), $\mathcal{A}$ proceeds as follows:

[4] In fact, Hohenberger has already generalized the proof by Bellare and Neven by showing that any one-way group isomorphism implies a secure transitve signature scheme under an autologous assumption of the one-more-RSA-inversion assumption [10]. However, since Hohenberger used a different model for a transitive signature scheme (*e.g.,* a signing algorithm to produce vertex certificate is designed explicitly rather than using any standard digital signature scheme). So, for the self-containment and clarity purposes, we still brief the security proof under the one-more-RSA-inversion assumption here.

1. If vertex i has not been created, $\mathcal{A}$ lets $L(i) = \texttt{CHALL}(.)$. $\mathcal{A}$ then computes vertex certificate for i, C_i, as the real signer.
2. If vertex j has not been created, $\mathcal{A}$ lets $L(j) = \texttt{CHALL}(.)$. $\mathcal{A}$ then computes vertex certificate for j, C_j, as the real signer.
3. $\mathcal{A}$ computes

$$z = \texttt{INV}\left(\frac{L(j)L(j)^{-1}}{v^x}\right)$$

where x is randomly chosen as long as it satisfies the `TVerify`'s second check. $\mathcal{A}$ returns a valid signature on edge $\{i, j\}$ as (C_i, C_j, z, x). This signature is valid because the following equality always holds:

$$z^e v^x = L(i)L(j)^{-1} \bmod N$$

As we can see, to answer every signature query from $\mathcal{F}$, $\mathcal{A}$ need to ask the RSA inversion oracle $\texttt{INV}(.)$ at most once. We can easily show that if $\mathcal{F}$ asks for signatures forming a connected graph G of m vertices, $\mathcal{A}$ needs to call $\texttt{INV}(.)$ exactly $m-1$ times (since the minimal connected graph of m vertices consists of $m-1$ edges). Since G has m vertices which means $\mathcal{A}$ has to return RSA inversion of m challenges from $\texttt{CHALL}(.)$, $\mathcal{A}$ can do so by asking $\texttt{INV}(.)$ to invert the public label of any vertex in G, say $L(j)$: $\ell(j) = \texttt{INV}(L(j))$. And then, for each other challenge, say $L(i)$ (wlog assume that $i < j$), $\mathcal{A}$ can compute its RSA inversion as $\ell(i) = z/(\ell(j)^{-1}s^x) \bmod N$ where z, x are parts of a signature on edge $\{i, j\}$ (either asked explicitly by $\mathcal{F}$ or obtained via composition). To conclude, if $\mathcal{F}$ asks $\mathcal{A}$ to sign a connected graph G with m vertices, in order to return RSA inversions of m challenges, $\mathcal{A}$ needs m calls to $\texttt{INV}(.)$.
In the general case, the graph G that $\mathcal{F}$ asks $\mathcal{A}$ to sign can be divided into some connected sub-graphs. Suppose that $\mathcal{F}$ outputs a forged signature on edge $\{i', j'\}$ which is not on the transitive closure of G. Using the similar argument we made in the proof of **Theorem 1**, i' and j' are on two different connected sub-graphs of G. This implies that the forged signature *connect* two sub-graphs of G. We know that for each connected graph of m vertices, $\mathcal{A}$ needs to call $\texttt{INV}(.)$ m times to answer challenges from $\texttt{CHALL}(.)$. But now, thank to $\mathcal{F}$, two disjoint connected sub-graphs are connected together for free, therefore, $\mathcal{A}$ saves one call to $\texttt{INV}(.)$ which proves the theorem.

□

Note that, in the proof of **Theorem 2**, we do not require x to be ℓ-bit long (or strictly less than e). Therefore, our proposed scheme can be less restricted, yet enjoys stronger security comparing to the case of security under the strong RSA assumption.

5 Conclusion and Future Works

We have presented a new transitive signature scheme and proved its security. Our scheme can provide forward security by employing a readily available key-evolving protocol of [7]. In addition, Our inability to prove the security of our scheme in case of adaptive chosen-message attacks assuming the strong RSA assumption and the

fact that the less restricted version of our scheme easily enjoys better security proof are evidences (although weak) that one-way trapdoor permutation is not enough to construct a secure transitive signature scheme [10]. Our future work is to show that such claim is true.

Acknowledgement

The first author sincerely thanks anonymous reviewers of ACNS 2005 for their insightful comments. This work is supported by a grant No.R12-2003-004-01004-0 from Ministry of Commerce, Industry and Energy.

References

1. Silvio Micali and Ronald L. Rivest, "Transitive Signature Schemes", *In the Proceedings of the Cryptographer's Track at the RSA Conference 2002*, Bart Preneel (Ed.), Springer-Verlag, LNCS 2271, pp. 236-243, 2002.
2. Mihir Bellare and Gregory Neven, "Transitive Signatures based on Factoring and RSA", *In the Proceedings of ASIACRYPT'02*, Y. Zheng (Ed.), Springer-Verlag, LNCS 2501, pp. 397-414, 2002.
3. Mihir Bellare and Gregory Neven, "Transitive Signatures: New Schemes and Proofs", Available at http://eprint.iacr.org/2004/215/.
4. Mihir Bellare and Phillip Rogaway, "Random Oracles are Practical: A Paradigm for Designing Efficient Protocols", *In the Proceedings of the First Annual Conference on Computer and Communications Security*, ACM Press, pp. 62-73, 1993.
5. Robert Johnson, David Molnar, Dawn Song and David Wagner, "Homomorphic Signature Schemes", *In the Proceedings of the Cryptographer's Track at the RSA Conference 2002*, Bart Preneel (Ed.), Springer-Verlag, LNCS 2271, pp. 244-262, 2002.
6. Shafi Goldwasser, Silvio Micali and Ronald L. Rivest, "A Digital Signature Scheme Secure against Adaptive Chosen-Message Attack", *SIAM Journal on Computing*, 17(2), pp. 281-308, April, 1988.
7. Gene Itkis and Leonid Reyzin, "Forward-Secure Signatures with Optimal Signing and Verifying", *In the Proceedings of CRYPTO'01*, J. Killian (Ed.), Springer-Verlag, LNCS 2139, pp. 332-354, 2001.
8. Louis C. Guillou and Jean J. Quisquater, "A Paradoxical Identity-Based Signature Scheme Resulting from Zero-Knowledge", *In the Proceedings of CRYPTO'88*, Shafi Goldwasser (Ed.), Springer-Verlag, LNCS 403, pp. 21-25, 1990.
9. David Pointcheval and Jacques Stern, "Security Proofs for Signature Schemes", *In the Proceedings of EUROCRYPT'96*, Ueli Maurer (Ed.), Springer-Verlag, LNCS 1070, pp. 387-398, 1996.
10. Susan Hohenberger, "The Cryptographic Impact of Groups with Infeasible Inversion", Master Thesis, Available at http://theory.lcs.mit.edu/ cis/cis-theses.html, May 2003.
11. Niko Baric and Birgit Pfitzmann, "Collision-free Accumulators and Fail-stop Signature Schemes without Trees", *In the Proceedings of EUROCRYPT 97*, Springer-Verlag, LNCS 1233, pp. 480–494, 1997.
12. Eiichiro Fujisaki and Tatsuaki Okamoto, "Statistical Zero-Knowledge Protocols to Prove Modular Polynomial Relations", *In the Proceedings of CRYPTO'97*, B. Kaliski (Ed.), Springer-Verlag, LNCS 1294, pp. 16–30, 1997.
13. Mihir Bellare, Chanathip Namprempre, David Pointcheval and Michael Semanko, "The One-More-RSA-Inversion Problems and the Security of Chaum's Blind Signature Scheme", *Journal of Cryptology*, 16(3), pp. 185–215, 2003.