SCIS 2005 The 2005 Symposium on
Cryptography and Information Security
Maiko Kobe, Japan, Jan.25-28, 2005
The Institute of Electronics,
Information and Communication Engineers

A New Provably Secure Transitive Signature Scheme

Dang Nguyen Duc * Zeen Kim * Kwangjo Kim *

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. Composability 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, 2, 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 GQ 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.

Keywords: Transitive signature scheme, provable security, strong RSA assumption, chosen-message attack.

1 Introduction

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, supervisor-employee relations in an organization, *etc.* In many scenarios, one needs to publish a graph representing some structure in an authenticated (and efficient) manner. Micali and Rivest proposed a specific solution for signing a graph called transitive signature [1]. A transitive signature scheme allows a signer to dynamically build an authenticated graph edge by edge. 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 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, 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 requires to be the only possible type of forgery under chosen-message attack. 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].

To realize the transitive signature concept, four security assumptions have been used so far. They include the intractability assumptions of RSA inversion, computing discrete logarithm, factoring and solving Diffie-Hellman problem [1, 2, 3]. In this paper, we present a new transitive signature scheme based on 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]. 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.

* International Research Center for Information Security (IRIS), Information and Communication University (ICU), 119 Munjiro, Yuseong-gu, Daejeon, 305-732 Republic of Korea, e-mail: {nguyenduc, zeenkim, kkj}@icu.ac.kr

2 Definition

2.1 Transitive Signature Scheme and Its Security

Similar to any standard digital signature scheme, a transitive signature scheme consist of three components, a key generation algorithm, a signature issuing algorithm and a signature verification algorithm. In addition to those components, there is another one to enable signature transitivity property which is called signature composition algorithm. Wlog, for an undirected graph $G = (V, E)$, we assume that $V \subset N$ and the notation $\{i, j\}$ represents an edge in G such that $i < j$. We now describe in detail four components of a transitive signature scheme.

- `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, 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, 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, k and two signatures σ_1, 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, $\perp$.

We now shall define what we mean by saying that a given transitive signature scheme is secure. As usual, we 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 $\mathcal{F}$ equipped with the signing oracle, $\texttt{TESign}(tsk, ., .,)$, attempts to forge a valid signature. 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 G be the graph formed by all pairs $\{i, j\}$ involved in that $\mathcal{F}$'s queries. 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. Otherwise, the experiment returns 0. We say that the given transitive signature scheme is secure if for all polynomial time $\mathcal{F}$, the above experiment returns 1 with negligible probability.

2.2 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 $\beta \in Z_N^*$ 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 [1]. 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 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, 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].

2.3 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, $\texttt{CHALL}(.)$ and $\texttt{INV}(.)$ where $\texttt{CHALL}(.)$ returns a random element in Z_N^* (N is product of two primes) and $\texttt{INV}(.)$ inverts RSA function with respect to the RSA modulus N and e of RSA public key (*i.e.,* return $x^{e^{-1}}$ mod N on input x). $\mathcal{A}$'s job is to compute RSA inversion of all k challenges returned by $\texttt{CHALL}(.)$ by asking strictly less k times the RSA inversion oracle, $\texttt{INV}(.)$. The one-more-RSA inversion assumption states that the chance for $\mathcal{A}$ to succeed is negligible.

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 $\mathcal{SDS}$ = ($\texttt{SKG}, \texttt{SSign}, \texttt{SVerify}$). 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 $ssk = (N, e, s, ssk)$

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

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:

```
For $t \in [i, j]$ do
  If $t \notin V$ then
    $V \leftarrow V \cup \{t\}$
    Randomly choose a secret label $\ell(t)$ from $Z_N^*$
    Compute the pubic label $L(t) \leftarrow \ell(t)^e \bmod N$
    Generate vertex certificate
        $\Sigma(t) \leftarrow \mathtt{SSign}(ssk, t||L(t))$
    Randomly choose another $l$-bit secret label $x_t$
    $\Delta(t) \leftarrow (\ell(t), x_t, L(t))$
Compute $z_i \leftarrow \ell(i)s^{x_i} \bmod N$
Compute $z_j \leftarrow \ell(j)s^{x_j} \bmod N$
Compute $z \leftarrow z_i/z_j \bmod N$ and $x \leftarrow x_i - x_j$
Let $C_i \leftarrow (L(i), \Sigma(i))$ and $C_j \leftarrow (L(j), \Sigma(j))$
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:

```
Parse $\sigma$ as $(C_i, C_j, z, x)$
Parse $C_i$ as $(L_i, \Sigma(i))$ and $C_j$ as $(L_j, \Sigma(i))$
If $\mathtt{SVerify}(spk, i||L_i) =$ 'reject' $\vee$ $\mathtt{SVerify}(spk, j||L_j)$ = 'reject'
  then Return 'reject'
If not $(|x| < 2^l)$ then Return 'reject'
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:

```
If $\mathtt{TEVf}(tpk, i, j, \sigma_1) =$ 'reject' $\vee$ $\mathtt{TEVf}(tpk, j, k, \sigma_2)$ = 'reject'
  then Return $\perp$
Parse $\sigma_1$ as $(C_i, C_j, z, x)$ and $\sigma_2$ as $(C_j, C_k, z', x')$
Output $\sigma_{ik} \leftarrow (C_i, C_k, zz' \bmod N, x + x')$
```

It is intuitive to see that the correctness of `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 $\{i, k\}$.

4 Security Analysis

We state the following two theorems regarding the security of our proposed scheme.

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

Proof To prove the theorem, we consider two different types of forger $\mathcal{F}$ described in the section 2.1:

- **Type I Forger**: $\mathcal{F}$ succeeds in forging a valid signature on edge $\{i', j'\}$ where at least one of i' or j' has not involved in $\mathcal{F}$'s queries to the signing oracle. In this case, $\mathcal{F}$ needs at least one forged vertex certificate since vertex certificates are included in every edge signature. Therefore, we can use this forged certificate(s) as the forged signature(s) with respect to $\mathcal{SDS}$ which violates our hypothesis that $\mathcal{SDS}$ is unforgeable.
- **Type II Forger**: $\mathcal{F}$ succeeds in forging a valid signature on edge $\{i', j'\}$ where both i' and j' appear in $\mathcal{F}$'s queries to the signing oracle. In this case, we will show that we can use $\mathcal{F}$ to solve the strong RSA problem with non-negligible probability which violates our hypothesis that the strong RSA assumption holds. To do so, we now construct an adversary $\mathcal{A}$ attacking the strong RSA assumption which uses $\mathcal{F}$ as a subrountine. 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, 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 subgraphs $G' = (V', E')$ $(V' \subset V, E' \subset E)$ 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 by randomly generating half of secret labels of vertices in V'. Wlog, we also assume that $|V'| = m$ is even (otherwise, $\mathcal{A}$ can add one more vertex to V' itself). We prove the claim by induction as follows:

- Case $m = 2$: Let $V' = \{i, j\}$, to sign the edge $\{i, j\}$, $\mathcal{A}$ 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 (or vertex j) at random from Z_N^*. It then compute public labels of two vertices

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

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

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

  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 $z^e v^x = L(i)/L(j) \bmod N$.
- Assume that the claim is true for $m = 2t$ for some positive integer t, we show that the claim is also true for $m = 2t+2$. According to the induction hypothesis, $\mathcal{A}$ can sign a sub-graph $G^* = (V^*, E^*)$ with $|V^*| = 2t$. For the remaining two vertices, say i, j ($i < j$), $\mathcal{A}$ produces a signature on edge $\{i, j\}$ similar to the case $m = 2$. To make the authenticated graph connected, $\mathcal{A}$ needs to sign one more edge from either i or j to one of vertex in V^*. Suppose that $\mathcal{A}$ decides to choose a secret label $\ell(i)$ for the vertex i when signing the edge $\{i, j\}$, $\mathtt{A}$ picks one vertex, say w, in V^* that it knows w's secret label $\ell(w)$. Then, $\mathcal{A}$ signs the edge $\{i, w\}$ (or $\{w, i\}$ if $w < i$) by choosing $x = 0$ (in other words, $x_w = x_i$) and computing the value z as the real signer: $z = \ell(i)s^{x_i}\ell(w)^{-1}s^{x_w} = \ell(i)/\ell(w)s^x = \ell(i)/\ell(w) \bmod N$. The valid signature on the edge $\{i, w\}$ is (C_i, C_w, z, x) where C_i and C_w are taken from signatures on edges $\{i, j\}$ and $\{w, w'\}$ from some vertex $w' \in V^*$. To conclude, the claim is also true for $m = 2t + 2$.

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. Because 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 half of secret labels of vertices in that sub-graph. Therefore, with probability $\frac{1}{2}$, $\mathcal{A}$ knows $\ell(i')$ and with probability $\frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}$, $\mathcal{A}$ knows both $\ell(i')$ and $\ell(j')$. If σ' is a valid signature on edge $\{i', j'\}$, then, the following equality holds:

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

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, $\mathcal{A}$ can solve the strong RSA problem.

■

Theorem 2 *If the one-more-RSA inversion assumption holds and $\mathcal{SDS}$ is unforgeable under chosen-message attack, then the $\mathcal{SRSA}$–$\mathcal{TS}$ 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 $\mathtt{CHALL}(.)$ 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:

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

2. If vertex j has not been created, $\mathcal{A}$ lets $L(j) =$ `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 `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 `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 `CHALL(.)`, $\mathcal{A}$ can do so by asking `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 `INV(.)`.

In the general case, the graph G that $\mathcal{F}$ asks $\mathcal{A}$ to sign can be divided into some connected subgraphs. 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 subgraphs of G. We know that for each connected graph of m vertices, $\mathcal{A}$ needs to call `INV(.)` m times to answer challenges from `CHALL(.)`. But now, thank to $\mathcal{F}$, two disjoint connected subgraphs are connected together for free, therefore, $\mathcal{A}$ saves one call to `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 simplified, yet enjoys stronger security comparing to the case of security proof under the strong RSA assumption.

5 Conclusion

We have presented a new transitive signature scheme and proved its security. Our scheme exhibits almost the same computational cost and signature size as in the previous schemes (see [2, 3] for a comparison table of previous schemes). We also note that, the secret key in our scheme does not contain any information about factors of the RSA modulus, therefore, it is easier to implement a key evolving protocol to provide forward security (for example, using the key evolving protocol in [7]). Our inability to prove the security of our scheme in case of adaptive chosen-message attacks assuming the strong RSA assumption intensifies the belief that one-way trapdoor permutation (thus standard digital signature scheme) is not enough to construct a secure transitive signature scheme [10]. Our future work is to show that such claim is true.

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 ASIANCRYPT'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 EURO-CRYPT 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.