A Transitive Signature Scheme Provably Secure Against Adaptive Chosen-message Attack

Huafei Zhu, Bao Feng, Robert H. Deng

InfoComm Security Department, Institute for InfoComm Research.
21 Heng Mui Keng Terrace, Singapore 119613.
{huafei, baofeng, deng}@i2r.a-star.edu.sg

Abstract. All node certificate based transitive signature schemes available in the literature make use of any digital signature scheme which is assumed to be provably secure against adaptive chosen-message attack, as a building block to produce node certificates in a graph. Consequently the algebraic structures to represent nodes in the graph are independent of the algebraic structure of signature scheme employed. This inconsistence of representation structures of the signature scheme, nodes and edges in the graph could increase the cost to manage those public data. For example, the transitive signature schemes presented by Micali and Rivest [5] and Bellare and Neven (the node certificate based version FBTS-1, in [1]), both heavily rely on the standard provably secure signature scheme (say Goldwasser-Micali-Rivest's signature scheme [7]). Consequently, a core problem related to transitive signature schemes is *how to construct transitive signature schemes so that the representation structures of signature schemes, nodes and edges in a graph can be implemented compactly?*

Bellare and Neven's hash-based modification, FBTS-2, achieving shorter signatures by eliminating the need for node certificates and provable under the same factoring assumption in the random oracle model, is actually the first solution to the above question. Our approach to attack the problem mentioned above, is different from Bellare and Neven's. We attack the problem by first carefully defining algebraic structure to represent vertices and edges in an undirected graph, then we construct a signature scheme so that its algebraic structure is coincident with that of vertices and edges in the graph. Finally, we present a practical realization of a transitive signature scheme that is proven transitively unforgeable under adaptive chosen message attack in the standard intractability paradigm. To the best knowledge of authors, this approach has NOT been reported in the literature.

Keywords: Discrete logarithm, signature scheme, strong RSA assumption, transitive signature scheme

1 Introduction

TRANSITIVE SIGNATURE SCHEME. The notation of transitive signature scheme, first introduced by Micali and Rivest [5], is a way to digitally sign vertices and edges of a dynamically growing transitively closed graph G, so as to guarantee the following properties:

-Given the signatures of edges (u, v) and (v, w), anyone can easily derive the digital signature of the edge (u, w);

-It is computationally hard for any adversary to forge the digital signature of any edge that is not in the transitive closure $\bar{G}$ of a graph G, even if the adversary can request the legitimate signer to digitally sign any number of G's vertices and edges of his choice in an adaptive fashion.

The transitive signature scheme presented in [5] is provably secure under adaptive chosen-message attack assuming that the discrete logarithm problem is hard in an underlying prime order group and assuming security of an underlying signature scheme to realize the concept in an undirected graph.

RELATED WORKS: Following from the pioneer works of Micali and Rivest, Johnson et al [4], have investigated related generations to model a situation where a censor can delete certain substrings of signed document without destroying the ability of the recipient to verify the integrity of the redacted document. In particular, the authors describe a scheme that allows a signature holder to construct the signature on an arbitrarily redacted sub-message of the originally signed message and also present another scheme for signing sets that is homomorphic with respect to both union and taking subsets.

Finally, Bellare and Neven [1] present novel realizations of the transitive signature primitive introduced by Micali and Rivest. The transitive scheme under the rubric of FBTS-1, is proven transitively unforgeable under adaptive chosen-message attack assuming factoring is hard. They also present a hash-based modification, FBTS-2, achieving shorter signatures by eliminating the need for node certificates, and provable under the same factoring assumption in the random oracle model.

THE PROBLEM. We realize that all node certificate based transitive signature schemes available in the literature make use of any digital signature scheme which is assumed to be provably secure against adaptive chosen-message attack, as a building block to produce node certificates in a graph. Consequently the algebraic structure to represent nodes in the graph are independent with that of the signature scheme employed. The inconsistence of representation structures of the signature scheme, nodes and edges in the graph could increase the cost to manage those public data. For example, the transitive signature schemes presented by Micali and Rivest [5] and Bellare and Neven (the node certificate based version FBTS-1, in [1]), both heavily rely on the standard provably secure signature scheme (say Goldwasser-Micali-Rivest's signature scheme [7]). Consequently a core problem related to transitive signature schemes is *how to construct transitive signature schemes so that the representation structures of signature schemes, nodes and edges in a graph can be implemented compactly?*

We emphasize the importance of the problem: one of the prospective applications of transitive signature scheme may be applied to solve secure trust delegation problem in the distributed networks [9]. In this setting, the cost of the computation and communication to manage public date of each party is most expensive and the history of direct or indirect recommendation should be updated frequently. Therefore, how to construct a security proved transitive signature scheme with minimum public date size is a interesting problem.

Bellare and Neven's hash-based modification FBTS-2 that achieves shorter signatures by eliminating the need for node certificates and it also is provable under the same factoring assumption in the random oracle model, is actually the first solution to the above question. We present alternative approach to attack the problem mentioned above, We attack the problem by first carefully defining algebraic structure to represent vertices and edges in an undirected graph, then we construct a signature scheme so that its algebraic structure is coincident with that of vertices and edges in the graph. To the best knowledge of authors, this approach has NOT been reported in the literature.

OUR CONTRIBUTIONS. In this report, we present a practical realization of the transitive signature primitive, introduced by Micali and Rivest [5]. The transitive signature scheme is proven transitively unforgeable under adaptive chosen message attack in the standard intractability model.

2 Notions and Definitions

NOTIONS A graph $G = (V, E)$ has a finite set V of vertices and a finite set $E \subseteq V \times V$ of edges. The transitive closure $G^* = (V^*, E^*)$ of a graph $G = (V, E)$ is defined to have $V^* = V$ and to have an edge (u, v) in E^* if and only if there is a path from u to v in G.

A transitive signature scheme $TS = (TKG, TSign, TVf, Comp)$ which is defined over an undirected graph, is specified by four polynomial-time algorithms and the functionality is as follows:

- The randomized key generation algorithm TKG takes input 1^k, where $k \in N$ is the security parameter, and returns a pair (tpk, tsk) consisting of public key and security key of a transitive signature scheme.
- The signing algorithm $TSign$ consists of a vertex signing algorithm $VSign$ and a edge signing algorithm $ESign$, where $VSign$ is a stateful and randomized algorithm that takes input of the security key tsk and a node i and returns a value calls certificate of node i, denoted by $Cert_i$. $ESign$ is a deterministic algorithm that takes input of the security key tsk and a two different nodes $i, j \in N$, and returns a value calls certificate of edge $\{i, j\}$ relative to tsk. $TSign$ maintains state which it updates upon each invocation.
- The deterministic verification algorithm TVf consists of two algorithms (VVf, EVf), where VVf is the deterministic vertex/node certificate verification algorithm that takes input of tpk and a certificate $Cert_i$ of vertex i, returns either 1 or 0. EVf is the deterministic algorithm that takes input of tpk and two nodes $i, j \in N$, and a certificate σ of edge $\{i, j\}$, returns either 1 or 0 (in the former case we say that σ is a valid signature of edge $\{i, j\}$ relative to tpk).

- The deterministic composition algorithm *Comp* takes input of *tpk* and nodes $i, j, k \in N$ and values σ_1, σ_2 to return either a value of σ or a symbol indicate failure.

DEFINITION OF CORRECTNESS. The definition of correctness is straight forward in a node certificate based transitive signature scheme, however it is rather a tricky matter to define the correctness in the setting where the node certificate is eliminated (please refer to [1] for more details). To achieve the goal of consistence of standard signature scheme and the representations of algebraic structures of vertices and edges in a graph, we define signing algorithm $TSign = (VSign, ESign)$ with two components so that it is easy to ensure the correctness. More details, when enquiring the $TSign$ oracle, we allow the signing oracle first checks the signature of the vertices adjacent to the edge. If there is at least one vertices has NOT been signed, then edge signing oracle runs the conventional signature scheme $VSign$ to sign the vertices at first. When the signature of both nodes in an edge are valid, it runs an edge signing oracle then.

EXPERIMENT TO ENSURE CORRECTNESS OF TRANSITIVE SIGNATURE SCHEME:

```
$(tpk, tsk) \leftarrow \mathrm{TKG}(1^k)$
$S_1 \leftarrow \emptyset$; $S_2 \leftarrow \emptyset$, Legit $\leftarrow$ true; NotOK $\leftarrow$ false
Run $Adv$ with its oracles until it halts, replying to its oracle queries as follows:
If $Adv$ makes $VSign$ query $i$ then
    If node $i$ has been signed by $VSign$, then $\sigma \leftarrow VSign(i)$
    Else
        Run $VSign$ and let $\sigma \leftarrow VSign(i)$, $S_1 = S_1 \cup \{i, VSign(i)\}$
If $Adv$ makes $ESign$ query $i, j$, then
    If $i = j$, then abort;
    Else
        If edge $(i, j)$ has been in signed by $VSign$ and $ESign$, then $\delta \leftarrow ESign(i, j)$
        Else
            Run $VSign$ to generate signatures of nodes $i, j$, then run $ESign$ and letting $\delta \leftarrow ESign(i, j)$, $S_2 = S_2 \cup \{(i, j), VSign(i, j)\}$
If A makes Comp query $(i, j, k, \delta_1, \delta_2)$, then
    If $[\{(i, j), \delta_1\} \notin S_2]$ OR $[\{(j, k), \delta_2\} \notin S_2]$ OR $[i, j, k$ are not all distinct$]$ then
        Legit $\leftarrow$ false
    Else
        Let $\tau$ be the output of the Composition oracle $Comp$, and $\delta \leftarrow ESign(i, k)$, then
            If $\tau = \delta$, then $S_2 = S_2 \cup \{(i, k), VSign(i, k)\}$
            Else NotOK $\leftarrow$ true,
When $Adv$ halts, output (Legit $\wedge$ NotOK) and halt.
```

The experiment computes a boolean predict Legit which is set to false if *Adv* ever makes an illegitimate query. It also compute a boolean predict NotOK which is set to true if a signature returned by the composition algorithm differs from the $ESgin$. The correctness of a transitive signature scheme requires that the probability Pr { Legit $\wedge$ NotOk=true } is zero. The definition of correctness in slightly different from Bellare and Neven's [1] since we distinguish $VSign$ and $ESign$ algorithms explicitly in a transitive signature scheme.

SECURITY OF TRANSITIVE SIGNATURE SCHEME: To define the security, we do the following experiment by running a key generation algorithm on input 1^k to get keys (tpk, tsk). Then we run Adv, provide this adversary with input pk and oracle access to the function $TSign$ =$(VSign, ESign)$. The oracle is assumed to maintain the state or toss coins as needed. Eventually, Adv will output $(i', j') \in N \times N$ and some τ'. Let E be the set of all edges $\{a, b\}$ such that Adv made oracle query a, b, and let V be the set of all integers a such that a is adjacent to some edge $\{i', j'\}$ is not in the transitive closure G of a graph $G = (V, E)$. The experiment returns 1 if Adv wins and 0 otherwise. The advantage of Adv in this attack defined for $k \in N$ by $Succ$=Pr$[Adv \ \ wins \ \ experiment]$.

We say that a transitive signature scheme is transitively unforgeable under adaptive chosen-message if $Succ$ is negligible for any adversary Adv whose running time is polynomial in the security parameters k.

3 A practical transitive signature scheme

SYSTEM PARAMETERS: Let p, q be two large primes such that $p - 1 = 2p'$ and $q - 1 = 2q'$, where p', q' are two $(l' + 1)$-bit strings. Let $n = pq$ and QR_n be the quadratic residue of Z_n^*. Let g, h be two generators of QR_n.

REPRESENTATION OF VERTEX: a vertex $v_i = g^{x_i} h^{y_i}$ in an undirect graph G.

REPRESENTATION OF EDGE: Signature of an edge $\{i, j\}$ is a pair: $\alpha_i = x_i - wx_j \bmod p'q'$ and $\beta_i = y_i - wy_j \bmod p'q'$ in an undirect graph G.

SIGNATURE SCHEME: We present compact implementation of transitive signature scheme that is consistent with a representation of edges in the graph. The signature scheme is defined as follows:

- Key generation algorithm: Let p, q be two large primes such that $p - 1 = 2p'$ and $q - 1 = 2q'$, where p', q' are two $(l' + 1)$-bit strings. Let $n = pq$ and QR_n be the quadratic residue of Z_n^*. Let g, h be two generators of QR_n. The public key is (n, g, h, X, H), where $X \in QR_n$ and H is a collision free hash function with output length l. The private key is (p, q).
- Signature algorithm: To sign a message m, a $(l + 1)$-bit prime e and a string $t \in \{0, 1\}^l$ are chosen at random. The equation $y^e = Xg^t h^{H(m)} \mathrm{mod} n$ is solved for y. The corresponding signature of the message m is (e, t, y).
- Verification algorithm: Given a putative triple (e, t, y), the verifier first checks that e is an odd $(l + 1)$-bit number. Second it checks the validation that

$X = y^e g^{-t} h^{-H(m)} \bmod n$. If the equation is valid, then the verifier accepts, otherwise, it rejects.

Fortunately, we are able to show that the signature scheme is immune to adaptive chosen-message attack under joint assumptions of the strong RSA problem as well as the existence of collision free hash function (see appendix for more details).

CERTIFICATE OF VERTEX: The certificate of each vertex v_i in authenticated graph is defined by $Cert_i = (e_i, y_i, t_i)$ derived from the signature equation: ${y_i}^{e_i} = X g^{t_i} h^{H(v_i)} \bmod n$.

A TRANSITIVE SIGNATURE SCHEME: We now can describe our transitive signature scheme.

-Given input 1^k, the key generation scheme algorithm a pair of signing keys (spk, ssk) for the signature scheme defined above.

-The signing algorithm $TSign = (VSign, ESign)$ maintains the state of $VSign(i), ESign(i,j)$, where the node $v_i = g^{x_i} h^{y_i}$ and signatures of the vertex is defined by $Cert_i = (e_i, y_i, t_i)$ derived from the equation ${y_i}^{e_i} = X g^{t_i} h^{H(v_i)} \bmod n$. The signature of an edge $\{i,j\}$ is $\delta_{i,j} = (\alpha_{i,j}, \beta_{i,j})$, where $\alpha_{i,j} = x_i - w x_j \bmod p'q'$ and $\beta_{i,j} = y_i - w y_j \bmod p'q'$.

-The composition algorithm $Comp$: Given nodes v_i, v_j and v_k and the signatures of edge $\{i,j\}$ and edge $\{j,k\}$, it checks the validity of certificate of each node $Cert_i$, $Cert_j$ and $Cert_k$ and it checks the validity of signature of each edge $\delta_{i,j}$ and $\delta_{j,k}$. If all are valid then it outputs $\delta_{i,k} = (\alpha_{i,k}, \beta_{i,k})$.

We remark that the representation structures of signature schemes, nodes and edges in a graph are implemented compactly in the above transitive signature scheme. We capture the compactness of the scheme by first defining algebraic structure to represent vertices and edges in an undirected graph, then we construct a signature scheme so that its algebraic structure is coincident with that of vertices and edges in the graph.

CORRECTNESS: The transitive signature scheme defined above satisfies the correctness property.

Proof: Since the composition algorithm $Comp$ checks that the certificate $Cert_j$ of v_j in the given signature of edge $\{i,j\}$ exactly matches the one in the given signature of $\{j,k\}$. This ensures that the public labels in those two certificates match, which is important in the proof of correctness. Now suppose $\{$ Legit $\wedge$ NotOK = True $\}$, i.e., Legit=True and NotOK = True. From the first statement Legit=True, it follows that all queries to the $Comp$ oracle is valid. Since the composition algorithm is a deterministic algorithm, consequently, the output of composition oracle is the same as the output of $ESign(i,k)$. Therefore the variable NotOK can never become true.

SECURITY: The transitive signature scheme is proven transitively unforgeable under adaptive chosen message attack in the standard intractability model.

Proof: Forgery of transitive signature can be in only two ways: either Type 1 Forgery: there is a forgery that recycles node certificates from previously issued signature, or Type 2 Forgery: there is a forgery that includes at least one new node. We therefore study the two cases in details below.

Type 1 Forgery: recycling node certificates from previously issued signature.

Simulator:

-on input 1^k, $\{g, h, N, p, q, H\} \leftarrow KG(1^k)$, where H is a collision free hash function defined in a proper domain.

-Defining a transitive signature oracle which is the same as that in a real transitive signature scheme.

-Defining the verification oracle which is the same as that in a real transitive signature scheme.

This completes the description of simulator. Notice that in the real transitive signature scheme, the knowledge $\log_g h$ is not a private information, therefore, the simulation defined above is the same as the real scheme from the point views of an adversary. Let E be a set of edges for which F queried a signature, and let $\bar{G} = (V, \bar{E})$ denote the transitive closure of $G = (V, E)$. For each oracle query $(VSign, ESign)$, there is no information leaked, due to the following fact:

$$\log_g(v_i) = x_i + wy_i \qquad (1)$$

$$\log_g(v_j) = x_j + wy_j \qquad (2)$$

Notice that the signature of the edge $\{i, j\}$ is $\delta_{i,j}$ =$(\alpha_{i,j}, \beta_{i,j})$, where $\alpha_{i,j} = x_i - x_j \bmod p'q'$ and $\beta_{i,j} = y_i - y_j \bmod p'q'$. Therefore $\alpha_{i,j}$ and $\beta_{i,j}$ is a linear combination of equation (1) and (2). Consequently, the distribution of variable (x_i, y_i) and (x_j, y_j) are same from the point views of the adversary. And any adversary at most with probability $1/p'q'$ to guess correctly of the secret key (x_i, y_i) (or (x_j, y_j) respectively). After the polynomial size oracle query, the adversary is able to forge a signature of edge $\{i', j'\} \notin \bar{G}$ with non-negligible advantage then it is able to forge a pair α', β' such that

$$\alpha_{i,j} + w\beta_{i,j} = \alpha' + w\beta' \mathrm{modp'q'} \qquad (3)$$

Since the simulator knows p, q, it follows that $\log_g h$ is revealed from equation (3).

Type 2 Forgery: a forgery containing at least one new node.

Simulator (given a signature scheme):

-On input 1^k, $\{g, h, N, p, q, H\} \leftarrow KG(1^k)$, where H is a collision free hash function defined in a proper domain;

-Choosing $x_i, y_i \in Z_n$ at random and defining $v_i = g^{x_i} h^{y_i}$;

-Running the given signature scheme to sign the vertex v_i.

-Defining the signature of the edge $\{i, j\}$ is $\delta_{i,j}$ =$(\alpha_{i,j}, \beta_{i,j})$, where $\alpha_{i,j} = x_i - x_j$ and $\beta_{i,j} = y_i - y_j$.

This completes the description of simulator. Notice that the simulator does not know the exact values p, q therefore we should show that the probability so that the event $\alpha_{i,j} \geq 0$ and the event $\beta_{i,j} \geq 0$ are both true with non-negligible. Notice that $\Pr\{\alpha_{i,j} > 0\} = \sum_{j=1 \cdots p'q'} p_j (\sum_{i \geq j} p_i)$ is at least 1/4, where p_i is the distribution of random variable i. Since variables $x_i, y_i, x_j, y_j \in Z_n$ are chosen at random, it follows that

$$\Pr\{\alpha_{\mathrm{i,j}} \geq 0 \wedge \beta_{\mathrm{i,j}} \geq 0\} \geq 1/16$$

By assumption, there is a forgery containing at least one new node which is not signed by the signature scheme algorithm actually with non-negligible probability. Consequently, the underlying signature scheme can be broken with non-negligible advantage, a contradiction of the assumption of security of the standard signature scheme.

4 Conclusions

We have developed a practical realization of node certificate based transitive signature primitive, introduced by Micali and Rivest [5]. The transitive signature scheme is proven transitively unforgeable under adaptive chosen message attack in the standard intractability model.

References

1. M. Bellare and G. Neven. Transitive Signatures based on Factoring and RSA. Advances in Cryptology-Asiacrypt 2002 Proceedings, Lecture Notes in Computer Science Vol. 2501, Y. Zheng ed, Springer-Verlag, 2002.
2. N. Braic and B. Pfitzmann. Collision free accumulators and fail-stop signature scheme without trees. *Eurocrypt'97*, 480-494, 1997.
3. Marc Fischlin: The Cramer-Shoup Strong-RSASignature Scheme Revisited. *Public Key Cryptography*, 2003: 116-129
4. R. Johnson, D. Molnar, Dawn X. Song, D. Wagner: Homomorphic Signature Schemes. *CT-RSA 2002*: 244-262
5. S. Micali, R.L. Rivest: Transitive Signature Schemes, CT-RSA 2002: 236-243.
6. R. Cramer and V. Shoup. Signature scheme based on the Strong RAS assumption. *6th ACM Conference on Computer and Communication Security*, Singapore, ACM Press, November 1999.
7. S. Goldwasser, S. Micali, R. Rivest: A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks. *SIAM J. Comput.* 17(2): 281-308, 1988.
8. L. Guillou, J. Quisquater. A practical zero-knowledge protocol fitted to security microprocessors minimizing both transmission and memory. *Eurocrypto'88*, 123-128, 1988.
9. H. Zhu, Bao Feng and Robert H. Deng. Computing of Trust in Distributed Networks, http://eprint.iacr.org/, 2003/056.
10. H. Zhu. New Digital Signature Scheme Attaining Immunity to Adaptive Chosen-message attack. *Chinese Journal of Electronics*, English version, Vol.10, No.4, Page 484-486, Oct, 2001.

Appendix: Security proof of Zhu's signature scheme

Zhu's signature scheme is defined as follows [10].

- Key generation algorithm: Let p, q be two large primes such that $p-1 = 2p'$ and $q-1 = 2q'$, where p', q' are two $(l'+1)$-bit strings. Let $n = pq$ and QR_n be the quadratic residue of Z_n^*. Let g, h be two generators of QR_n. The public key is (n, g, h, X, H), where $X \in QR_n$ and H is a collision free hash function with output length l. The private key is (p, q).
- Signature algorithm: To sign a message m, a $(l+1)$-bit prime e and a string $t \in \{0,1\}^l$ are chosen at random. The equation $y^e = Xg^t h^{H(m)} \bmod n$ is solved for y. The corresponding signature of the message m is (e, t, y).
- Verification algorithm: Given a putative triple (e, t, y), the verifier first checks that e is an odd $(l+1)$-bit number. Second it checks the validation that $X = y^e g^{-t} h^{-H(m)} \bmod n$. If the equation is valid, then the verifier accepts, otherwise, it rejects.

Before we provide a rigorous proof of security to Zhu's signature scheme, we remark relations between Zhu's and Fischlin's signature scheme [3].

Fischlin's signature scheme Marc Fischlin's signature scheme is defined as follows [3]:

- Key generation: Generating $n = pq$, where $p = 2p'+1$ and $q = 2q'+1$ for primes p, q, p', q'. Also pick three quadratic residue $h_1, h_2, x \in QR_n$. The public key verification key is (n, h_1, h_2, x) and the private key is (p, q).
- Signing: To sign a message m calculate the l-bit hash value $H(m)$ with a collision-intractable hash function $H(\cdot)$. Pick a random $(l+1)$-bit prime e, and a random l-bit string α and compute a representation $(-\alpha, -(\alpha \oplus H(m)), y)$ of x with respect to h_1, h_2, e, n, i.e.,

  $$y^e = x{h_1}^{\alpha}{h_2}^{\alpha \oplus H(m)} \bmod n.$$

  Computing this e-th root y from $x{h_1}^{\alpha}{h_2}^{\alpha \oplus H(m)}$ is easy given the factorization of n. The signature is (e, α, y).
- Check that e is an odd $(l+1)$-bit integer, that α is l bits long, and that $y^e = x{h_1}^{\alpha}{h_2}^{\alpha \oplus H(m)} \bmod n$.

We remark the relationships between two signature schemes below:

- It is clear that the algebraic structures of Zhu's and Fischlin's signature are same;
- If there is no collision hash function involved in the above two schemes, then it is not hard to show that the above two signature schemes are equivalent in the same security level. More precisely, if Zhu's scheme can be broken by an adversary A with non-negligible probability then there exists an adversary B^A so that Fischlin's signature scheme can be broken with the same probability. The statement is also true by means of vis-a-vis argument.

- In case of a collision free hash function involved in both schemes, suppose Zhu's signature scheme can be broken with non-negligible probability, i.e., there is an adversary A is able to forge a faking message m in Zhu's signature scheme, denoted by $\sigma(m) = (e, y, t)$ with non-negligible probability. Then there exists an adversary B^A in Fischlin's signature scheme so that it is able to produce a valid signature $\sigma(m') = (e, y, t)$ for any message in the set $S := \{m' | H(m) \oplus H(m') = t\}$, where t is a component of faking signature $\sigma(m)$ correspondent to Zhu's signature scheme. The statement is also true by means of vis-a-vis argument.

Strong RSA assumption: Strong RSA assumption was introduced by Baric and Pfitzmann very recently [2]: for any randomly chosen n, given a random element $z \in Z_n^*$, it is hard to find a pair (e, y) such that $y^e = z \mathrm{mod} n$.

Guillou-Quisquater Lemma [8] : The following lemma, suggested by Guillou and Quisquater, is useful for the proof of the main result. Suppose $w^e = z^b$ and $d = \gcd(e, b)$. Then there exists an efficient algorithm computing the (e/d)-th root of z.

Proof: Since $d = \gcd(e, b)$, by Euclidean algorithm, $d = ee' + bb'$. It yields the equation $z = (z^{e'} w^{b'})^{e/d}$.

Main result: The signature scheme is immune to adaptive chosen-message attack under joint assumptions of the strong RSA problem as well as the existence of collision free hash function.

Proof: Assume that the signature scheme is NOT secure against adaptive chosen message attack. That is, there is an adversary, who is able to forge the signature (e, t, y) of a message $m (m \neq m_i,\ 1 \leq i \leq k)$ with non-negligible probability after it has queried correspondent signature of each message $m_1, \cdots, m_k$, which is chosen adaptively by the adversary. Let $(e_1, t_1, y_1), \cdots, (e_k, t_k, y_k)$ be signatures provided by the signing oracle corresponding to a set of messages $m_1, \cdots, m_k$. We consider two types of forgeries: 1) for some $1 \leq j \leq k$, $e = e_j$; 2) for all $1 \leq j \leq t$, $e \neq e_j$. We should show that any forgery scheme of the two types will lead to a contradiction to the assumptions of the theorem. This renders any forgery impossible.

Type 1-Forger

We consider an adversary who chooses a forgery signature such that $e = e_j$ for a fixed j: $1 \leq j \leq k$, where k is the total number of the queries to the signing oracle. If the adversary succeeds in a signature forgery as type1 with non-negligible probability then given n, we are able to compute $z^{1/r}$ with non-negligible probability for a given z and r, where r is a $(l+1)$-bit prime. This contradicts to the assumed hardness of the standard RSA problem. We state the attack in details as follows: given $z \in Z_n^*$ and r, we choose a set of total $k-1$ primes with length $(l+1)$-bit $e_1, ...e_{j-1}, e_{j+1}, ..., e_k$ at random. We then create the correspondent public key (g, h) of the simulated signature scheme as follows:

$g = z^{2e_1 \dots e_{j-1} e_{j+1} \dots e_k}$, $h = v^{2e_1 \dots e_k}$ and $X = g^{-\alpha} w^{2e_1 \dots e_k}$, where $w, v \in Z_n$ and α is a l-bit string. Since QR_n is a cyclic group, we can assume that g, h are generators of QR_n with overwhelming probability. To sign the i-th message $m_i (i \neq j)$, the signing oracle selects a random string $t_i \in \{0,1\}^l$, and computes:

$${y_i}^{e_i} = ((wv)^{2e_1 \dots e_{i-1} e_{i+1} \dots e_k} z^{2(t_i - \alpha) \Pi_{s \neq i, s \neq j} e_s})^{e_i}$$

The output of the signing oracle is a signature of message m_i, denoted by $\sigma(m_i) = (e_i, y_i, t_i)$.

To sign the j-th message m_j, the signing oracle, sets $t_j \leftarrow \alpha$ and computes:

$${y_j}^{e_j} = ((wv)^{2 \Pi_{s \neq j} e_s})^{e_j}$$

The output of the signing oracle is a signature of message m_j, denoted by $\sigma(m_j) = (e_j, y_j, t_j)$.

Let $\sigma(m) = (e, y, t)$ be a valid signature forged by the adversary of message m. By assumption, we know that $y^e = X g^t h^{H(m)}$. Consequently, we have the following equation:

$$g^{t_j} h^{H(m_j)} {y_j}^{e_j} = g^t h^{H(m)} y^e$$

Equivalently

$$z^{2(\alpha - t) \Pi_{i \neq j} e_i} = (v^{2(H(m) - H(m_j)) \Pi_{i \neq j} e_i} \frac{y}{y_j})^{e_j}$$

In the case that $t \neq \alpha$, we apply Guillou-Quisquater lemma to extract the r-th root of z. We therefore arrive at the contraction of hardness of the standard RSA assumption.

In the case $t = t_j$ =α, i.e, the adversary outputs a forgery (e, y, t) such that $e = e_j$ and $t = t_j = \alpha$ with some value y, the above equation at the end is trivial, therefore we should reconsider the simulator as follows:

Again, given $z \in Z_n^*$ and r, we choose a set of total $k - 1$ primes with length $(l+1)$-bit $e_1, ... e_{j-1}, e_{j+1}, ..., e_k$ at random. We also choose $w, v \in Z_n$ at random and create the correspondent public key (g, h) of the simulated signature scheme by computing $h = z^{2e_1 \dots e_{j-1} e_{j+1} \dots e_k}$, $g = v^{e_1 \cdots e_k} z^{2e_1 \dots e_{j-1} e_{j+1} \dots e_k}$ and $X = w^{e_1 \cdots e_k} z^{2e_1 \dots e_{j-1} e_{j+1} \dots e_k (-\alpha)}$.

Since the simulator knows each e_i, therefore it is easy to compute the i-th signing query. What we need to show is how to simulate the j-th signing query. This can be done as follows:

$$y_j^{e_j} = x g^{t_j} h^{H(m_j)} = (wv)^{e_1 \cdots e_k} z^{2e_1 \dots e_{i-1} e_{i+1} \dots e_k (-\alpha + t_j + H(m_j))}$$

Now we set $-\alpha + t_j + H(m_j) = 0$, i.e, $t_j = \alpha - H(m_j)$. To show the simulation above is non-trivial, we should show $Pr\{\alpha \geq H(m_j)\}$ is an non-negligible amount. Since $H(m_j) \in \{0,1\}^l$ is random variable, we define $x_j = H(m_j)$ and $Pr(x = x_j) = p_j$, without loss of generality, we may further denote x_j by j. It is not hard to show that $Pr\{\alpha \geq x_j\} = \sum_{j=1 \cdots 2^l} p_j (\sum_{i \geq j} p_i)$. What we want to show is that the probability $Pr\{\alpha \geq x_j\}$ is an non-negligible amount. Suppose $Pr\{\alpha \geq x_j\}$ is an negligible amount, i.e., $\sum_{j=1 \cdots 2^l} x_j p_j = 2^l$, except for an negligible amount. Equivalently, H is a single valued function except for an negligible amount, this is an contradiction.

Now we suppose the adversary is able to forge a faking signature of message m, denoted by (e, y, t), such that $e_j = e(= r)$, $t_j = t$. Notice that one can not assume that $e_j = e$, $t_j = t$ and $y_j = y$, since H is a collision free hash function. Now we have two equations: $y_j^e = Xg^t h^{H(m_j)}$ and $y^e = Xg^t h^{H(m)}$. Consequently, we obtain the equation:

$$(\frac{y_j}{y})^e = h^{H(m_j)-H(m)} = z^{2e_1,\ldots e_{j-1},e_{j+1},\ldots,e_k(H(m_j)-H(m))}$$

It follows that one can extract the e-th root of z with non-negligible probability. Therefore, we arrive at the contradiction of the standard hardness of RSA assumption.

Type 2-Forger

We consider the second type of the attack: the adversary forgery is that for all $1 \leq j \leq k$, $e \neq e_j$. If the adversary succeeds in forgery with non-negligible probability, then given n, a random $z \in Z_n^*$, we are able to compute $z^{1/d}$ $(d > 1$) with non-negligible probability, which contradicts to the assumed hardness of strong RSA assumption. We state our attack in details as follows: we generate g and h with the help of z. We define $g = z^{2e_1 \cdots e_k}$ and $h = g^a$, where $a \in (1, n^2)$, is a random element. We can assume that g is a generator of QR_n with overwhelming probability. Finally, we define $X = g^b$, where $b \in (1, n^2)$. Since the simulator knows the all e_j, the signature oracle can be perfectly simulated. Let (e, t, y) be a forgery signature of message m. It yields the equation $y^e = Xg^t h^{H(m)} = z^E$, where $E = (b + t + aH(m))2e_1 ... e_k$. Since we are able to compute (e/E)-th root of z provided e is a not a divisor of E according to the lemma of Guillou and Qusiquater, it is sufficient to show that e is not a divisor of E with non-negligible probability. Due to the the fact that $\gcd(e, e_1 e_2 \cdots e_k) = 1$, it is sufficient to show that e is not a divisor of $b + t + aH(m)$ with non-negligible probability. Suppose $e|(b + t + aH(m)$, or equivalently, $b + t + aH(m) \equiv 0 \mathrm{mod} e$. Since $a \in (1, n^2)$, we can write a as $a = a'p'q' + c'$. It follows a' is a random element from the adversary's view. Hence the probability that $b + t + aH(m) \equiv 0 \mathrm{mod} e$ is about $1/e$. Thus, with non-negligible probability, e is not a divisor of $b + t + aH(m)$.

Acknowledgement We would like to thank Marc Fischlin for his comments on the signature scheme