Verifiable Signature Sharing Scheme Based on Strong RSA Assumption

KEWEI LV[1]
YANHUA YANG[2]

[1]State Key Laboratory of Information Security, Graduate University of Chinese Academy of Sciences
P. O. Box 4588, Beijing 100049, P. R. China
[2]Mathematic department of Capital Normal University, Beijing 100037,P. R. China
[1]conwaylu@tom.com

Abstract. In 2000, Cramer and Shoup presented a signature scheme which is secure against adaptive chosen-message attacks under the strong RSA assumption in the standard model. Then, in 2003, under the strong RSA assumption only, Fischlin produced a signature of roughly half the length. A verifiable signature sharing scheme (VΣS) introduced by Franklin and Reiter in 1995 enables the recipient of a signature to share it among n proxies so that a subset of them can reconstruct it later. In this paper, we first give a modification of Fischlin's scheme under the strong RSA assumption and then get a new verifiable signature sharing scheme based on it in the standard model. At last, we prove that our new VΣS can tolerate the malicious behavior of up to half of the proxies corrupted by a static adversary except for a negligible probability. Our scheme is efficient and the length of signature in our scheme is similar to Fischlin's and roughly half of Cramer-Shoup signature scheme.

Keywords: Strong RSA, Hash function, Digital signature, Discrete-log, Verifiable Signature Sharing.

(Received May 14, 2009 / Accepted August 18, 2009)

1 Introduction

A signature scheme provides a way for each user to sign messages so that the signatures can later be verified by anyone else. More specifically, each user can create a matched pair of private and public keys so that only he can create a signature for a message using his private key, but anyone can verify the signature for the message using the signer's public key. The verifier can convince himself that the message contents have not been altered since the message was signed. Also, the signer can not later repudiate having signed the message, since no one but the signer possesses his private key.

In [4], Cramer and Shoup have presented a signature scheme. For a 1024-bits RSA modulus and a 160-bit message, a signature has about 2200 bits. The scheme is secure against adaptive chosen-message attacks under the strong RSA assumption and does not rely on random oracle model. By security we mean security against an adaptive chosen message attack, as defined by Goldwasser et al in [9]. Cramer and Shoup added discrete-log assumption to the variation of their scheme and then produced signatures of roughly half the length (about 1350 bits). In [5], Fischlin revisited their scheme and achieved the same signature size under the strong RSA assumption only, even with a slightly improved performance than in the original strong-RSA only case or the discrete-log and strong-RSA case. But Fischlin's scheme includes X-OR operation, so that the scheme is difficult to be applied to the Verifiable Signature Sharing scheme (VΣS).

Verifiable Signature Sharing scheme (VΣS), which was a protocol first introduced by Franklin and Reiter in [6], enables the recipient of a digital signature, who is not necessarily the original signer, to share such signature among n proxies so that a subset of them can later reconstruct it. A VΣS is divided into a sharing

phase and a recover phase. At the end of the sharing phase each proxy can verify that a valid signature for the given document can be reconstructed. At the end of the recover phase such signature is reconstructed no matter what a malicious subset of proxies may do. As we known, the VΣS for RSA, Rabin, ElGamal, Schnorr, and DSS are all discussed. But the solutions for Cramer-Shoup seem more elusive. The reason of it may be the complex operations of Cramer-Shoup.

In this paper, we first show a modified version of Fischlin's scheme under the strong RSA assumption only. Our solutions change a random l-bit string α and l-bit hash value into a random $(l-1)$-bit string α and $(l-1)$-bit hash value and rearrange a representation $(-\alpha, -(\alpha \bigoplus H(m)), y)$ by a representation $(-\alpha, -(\alpha + H(m)), y)$. The length of signature in our scheme is similar to Fischlin's and roughly half of the original Cramer-Shoup signature scheme. The efficiency of our scheme corresponds to that of Fischlin. Then we base on it to get a new secure and efficient VΣS.

Organization: We first briefly overview the digital signature and the verifiable signature sharing in section 2 and 3 respectively, and describe previously proposed solutions to Cramer-Shoup signature scheme and our scheme with a sketch proof of security. Then basing on the scheme, we present a new VΣS and show the proof of its security. In section 4, we give the conclusion.

2 Signature Scheme

In this section, we give some definitions and assumptions to our protocol, then describe our protocol and sketch proof of its security.

2.1 The RSA and strong RSA Assumptions

Now we review the RSA and strong RSA assumptions in somewhat more detail.

RSA Problem: Given a randomly generated RSA modulus n, a exponent r, and a random $z \in Z_n^*$,find $y \in Z_n^*$ such that $y^r = z$. The RSA assumption is this problem is hard to solve.

Strong RSA Problem: Given an RSA modulus n and a random $z \in Z_n^*$, find $r > 1$ and $y \in Z_n^*$, such that $y^r = z$.The strong RSA assumption is this problem ia hard to solve.

Note that these assumptions are different, in that for RSA assumption the exponent r is chosen independently of z; whereas for the strong RSA assumption r may be chose in a way that depends on z. Barić and Pfitzmann in [1] introduced the strong RSA assumption. It has subsequently been used in the analysis of several cryptographic schemes in [7] and [8]. This is a potentially stronger assumption than the RSA assumption, but at the present, the only known method for breaking either assumption is to solve the integer factorization problem.

2.2 Definition of Digital Signature scheme

A user's signature on a message m is a string which depends on m, on pubic and secret key specific to the user and possibly on randomly chosen data, in such a way that anyone can check the validity of the signature by using public data only. Obviously, we would like to prevent the forgery of user's signature without knowledge of his secret key. In this section, we show the definition of digital signature and of the possible attacks against them.

Definition 1 *A digital signature scheme within the public key framework, is defined as a triple of algorithms $(\mathcal{G}, \Sigma, \mathcal{V})$such that*

1. *Key generation algorithm $\mathcal{G}$ is a probabilistic, polynomial-time algorithm which on input a security parameter 1^k, produces pairs(P, S), where P is called a public key and S a secret key. We use the notation $(P, S) \in \mathcal{G}(1^k)$ indicates that the pair (P, S) is produced by the algorithm $\mathcal{G}$.*

2. *Signing algorithm Σ is a probabilistic, polynomial-time algorithm which is given a security parameter 1^k, a secret key S in range $\mathcal{G}(1^k)$ and a message $m \in \{0,1\}^k$ and produces as output strings s which we call the signature of m. We use notation $s \in \Sigma(1^k, S, m)$ if the signing algorithm is probabilistic and $s = \Sigma(1^k, S, m)$. Moreover, when the context is clear, we will write $s \in \Sigma(S, m)$ to mean that s is the signature of message m.*

3. *Verification algorithm $\mathcal{V}$ is a polynomial-time algorithm which is given a signature s, a message m and a public key P and tests whether s is a valid signature of m with respect to P. In general, the verification algorithm need not be probabilistic.*

2.3 Attacks Against Digital Signatures

A theoretical treatment of digital signatures security was initiated by Goldwasser, Micali, and Yao in [8] and followed in [2],[9],[10] and [11]. we now distinguish three basic kinds of attacks, listed below in the order of increasing severity.

1. Key-Only Attack: In this attack, the adversary knows only the pubic key of the signer and therefore only has the capability of checking the validity of signatures of messages given to him.

2. Known-Signature Attacks: The adversary knows the pubic key of the signer and has been message/ signature pairs chosen and produced by the legal signer. In reality, this is the minimum an adversary can do.

3. Chosen Message Attack: The adversary is allowed to ask the signer to sign a number of messages of the adversary's choice. The choice of these messages may depend on previously obtained signatures. This attacks contains three subclasses: the generic chosen-message attacks, the oriented chosen-message attack and the adaptively chosen-message attack.

It is easy to know that the last one is the most serious attack. We now classify the expected results of an attack:

- Disclosing the secret key of the signer: The adversary can compute the signer's secret key. It is the most serious attack, which is called total break.
- Constructing an efficient algorithm: The adversary succeeds in forging the signature of some message of his choice. This is called universal forgery.
- Providing a new message signature pair: The adversary is able to forge the signature of one massage not necessarily of his choice. This is called existential forgery.

Clearly, different levels of security may be required of different applications. Sometimes, it may suffice to show that an adversary who is capable of a known signature attack can not succeed on selective forgery, while for other applications it may be required that an adversary capable of a chosen signature attack can not succeed even on existential forgery with non-negligible probability.

Here we call a signature scheme to be secure if an existential forgery is computationally impossible even under an adaptively chosen-message attack.

2.4 Our revised protocol

In this section, we recall the original Cramer-Shoup scheme and the Fischlin's scheme. Furthermore, we revised Fischlin's scheme and prove its security.

These schemes are parameterized by two security parameters l and l', where $l + 1 < l'$.They all make use of a collision-resistant hash function H, whose output can be interpreted as a positive integer less than 2^l. A reasonable choice for H might be SHA-1 in [12]. For a positive integer N,we let $\mathrm{Q}R_n$ denote the subgroup of Z_n^* of squares.

2.4.1 Cramer-Shoup Signature scheme

The original Cramer-Shoup scheme is presented as follows in Figure 1.

1. Key Generation: The signer generate a composite number N, where $N = pq$ which is product of two safe primes, such that $p = 2p' + 1, q = 2q' + 1$ for primes p, q, p', q'. Pick two quadratic residues $h, x \in QR_N$ and a random $(l + 1)$-bit prime e', so that pubic key is (N, h, x, e') and private key is (p, q).

2. Signing: To sign a message m, compute the l-bit hash value $H(m)$ with a collision-intractable hash function $H(\cdot)$. Pick a random $(l + 1)$-bit prime $e \neq e'$, y and a random $y' \in QR_N$, then compute x' such that $(y')^{e'} = x'h^{H(m)} \ mod \ N$ and $y^e = xh^{H(x')} \ mod \ N$. The signature is (e, y, y').

3. Verification: To verify a putative signature (e, y, y') on a message m. First check that e is an odd $(l + 1)$-bit number different from e', then compute $x' = (y')^e h^{-H(m)}$ and verify that $x = y^e h^{-H(x')}$.

Figure 1. Cramer-Shoup signature scheme

Remark: In the scheme, y can be calculated using the factorization of N. It is not necessary in verification to verify that e is prime due to [4].

2.4.2 Fischlin's Signature scheme

Under the strong RSA assumption only, Fischlin revised and simplified the scheme of Cramer-Shoup as follows in Figure 2.

2.4.3 Revision of Fischlin's Signature scheme

As pointed out in [5], Fischlin assimilate the trapdoor commitment to the representation problem, where one may split the message into α and $\alpha + H(m)$. Here we use the same idea to turn the trapdoor commitment into the representation problem. But the representation in our scheme is $(-\alpha, -(\alpha + H(m)), y)$ instead of $(-\alpha, -(\alpha \oplus H(m)), y)$ in Fischlin's scheme. (To see Figure 3)

1. Key Generation: The signer generate a composite number N, where $N = pq$ which is product of two safe primes such that $p = 2p' + 1, q = 2q'+1$ for primes p, q, p', q'. Pick three quadratic residues $h_1, h_2, x \in QR_N$, so that pubic key is (N, h_1, h_2, x) and private key is (p, q).

2. Signing: To sign a message m, compute the l-bit hash value $H(m)$ with a collision-intractable hash function $H(\cdot)$. Pick a random $(l + 1)$ bit prime e, 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, where $(y)^e = xh_1^{\alpha}h_2^{\alpha \oplus H(m)} mod\ N$ The signature is (e, α, y).

3. Verification: To verify a putative signature (e, α, y) on a message m. First check that e is an odd $(l + 1)$-bit integer, that α is l bits long and that $(y)^e = xh_1^{\alpha}h_2^{\alpha \oplus H(m)} mod\ N$

Figure 2. The Fischlin′s Signature scheme

1. Key Generation: The signer generate a composite number N, where $N = pq$, which is product of two safe primes, such that $p = 2p' + 1, q = 2q'+1$ for primes p, q, p', q'. Pick three quadratic residues $h_1, h_2, x \in QR_N$, so that pubic key is (N, h_1, h_2, x) and private key is (p, q).

2. Signing: To sign a message m, compute the $(l - 1)$-bit hash value $H(m)$ with a collision-intractable hash function $H(\cdot)$. Pick a random $(l + 1)$-bit prime e, a random $(l - 1)$-bit string α and compute a representation $(-\alpha, -(\alpha + H(m)), y)$ of x with respect to h_1, h_2, e, N where $(y)^e = xh_1^{\alpha}h_2^{\alpha+H(m)} mod\ N$ The signature is (e, α, y).

3. Verification: To verify a putative signature (e, α, y) on a message m. First check that e is an odd $(l + 1)$-bit integer,that α is $(l - 1)$-bits long and that $(y)^e = xh_1^{\alpha}h_2^{\alpha+H(m)} mod\ N$

Figure 3. Revision of Fischlin's Signature Scheme

Compared to Cramer-Shoup signature scheme and Fischlin's Signature scheme, we can see that efficiency of our scheme corresponds to that of Fischlin's and more optimal than Cramer-Shoup. And the differences between ours and Fischlin's are that we use the operation $\alpha + H(m)$ instead of $\alpha \oplus H(m)$. If we use some precomputation techniques, the efficiency of them is comparative.

2.5 Security of the Revision

Now we give the security of the Revised Scheme.

Theorem 1 *The revised signature scheme is secure against adaptive chosen message attack, under the strong RSA assumption and the assumption that H is collision resistant.*

First, we review the notion of an adaptive chosen message attack. The key generation algorithm for the signature scheme is moved, generating a public key which is given to the adversary and a private key which is given to a "signing oracle". Next, the adversary queries the signing oracle a number of times, which submits a message of its choice to signing oracle each query. The signing oracle signs the given message and gives the signature to the adversary. In the course, the power of choosing the message of the adversary is free. At the end of its execution, it outputs a forged signature on a message which was not submitted to the signing oracle. Of course, the adversary is either allowed to fail or outputs a forged signature. Because the adaptive chosen message attack is the most serious attack, if a signature scheme can against this attack, we call it secure.

Before proving Theorem 1, for convenience,we give a well-known and useful lemma.

Lemma 1 *Given $X, Y \in Z_N^*$ and $a, r \in Z$, such that $X^a = Y^r$ and $gcd(a, r) = 1$. Then one can efficiently compute x, such that $x^r = X$.*

Proof. Due to $gcd(a, r) = 1$, so we use the extended Euclidean algorithm to compute integers s, t, such that $sa + tr = 1$. We set $x = Y^s X^t$, so $x^r = (Y^s X^t)^r = Y^{sr} X^{tr} = X^{as} X^{tr} = X^{as+tr} = X$. From the above description, we find x does the job. □

Here we will give sketch proof of Theorem 1 since the proof is similar to that in [5]. The specific proof of it can be found in[13]. We assume the adversary makes t signing queries and then produces a forgery. Let m_i denote the i-th query to the singer and (e_i, α_i, y) the answer. Let (e, α, y) be forgery on message m and we assume that all e_i chosen by the signer during an attack are distinct (yet, the adversary's choice e may equal some e_j and $H(m) \neq H(m_i)$ for m_i).

We only need discuss types of forgers [4] as following :

- **Type 1**: For some $1 \leq j \leq t$, $e = e_j$ and $x' \neq x'_j$.

- **Type 2**: The adversary outputs $e = e_j$ for some j.
- **Type 3**: The adversary outputs a new e, $e \neq e_j$ for all e_j.

Forgers of Type 1 disappear due to our modification. So there are two types of forgers in our scheme to be done.

Forger of Type 2 : We assume the value j in Type 2 is found, if not,we can guess it. Since $H(m_j) \neq H(m)$, we have $\alpha_j \neq \alpha$ or $\alpha_j + H(m_j) \neq \alpha + H(m)$. We can guess in advance which case will happen with probability $1/2$. Here we assume $\alpha_j \neq \alpha$. The other case is treated in the analogous way.

Given $N, z \in Z_N^*$ and odd prime r, we can output $z^{1/r}$. We will invoke the Type 2 Forger on the following pubic key and signature oracle. Set $e_j = r$ and for $i \neq j$, where $1 \leq i \leq t$, choose a random $(l+1)$-bit prime e_i. Let $h_1 = v^{2\prod_i e_i}$, $h_2 = z^{2\prod_{i \neq j} e_i}$, $x = h_1^{-\beta} w^{2\prod_i e_i}$ for random $v, w \in Z_N^*$ and a random $(l-1)$-bit string β. The public key is (N, h_1, h_2, x).

On behalf of the signer to sign the i-th message for $i \neq j$. Choose a random $(l-1)$ bit string α_i and compute $y_i = (x_1 h_1^{\alpha_i} h_2^{\alpha_i + H(m_i)})^{1/e_i}$. For the j-th message query,set $\alpha_j = -H(m_j)$ and compute $y_j = (x h_1^{\alpha_j} h_2^{(\alpha_j + H(m_j))})^{1/e_j}$.

It is easy to see that the distribution of the data in the simulation is identical to the real attack. And adversary can get

$$h_1^{-\alpha_j} h_2^{-(\alpha_j + H(m_j))} y_j^r = x = h_1^{-\alpha} h_2^{-(\alpha + H(m))} y^r \; mod \, N.$$

Since $h_1 = v^{2\prod_i e_i}$ and $h_2 = z^{2\prod_{i \neq j} e_i}$, we can set $a = 2\prod_{i \neq j} e_i((\alpha + H(m)) - (\alpha_j + H(m_j)))$ and $Y = v^{2\prod_{i \neq j} e_i(\alpha_j - \alpha)} y y_j^{-1}$ and get $z^a = Y^r$. Since $((\alpha + H(m)) - (\alpha_j + H(m_j)))$ is at most l-bit and all e_k are relatively prime. we can easily compute an r-th root of z by lemma 1.

Forger of Type 3: This case is almost identical to the one discussed in [5] except for the addition instead of XOR to get the conclusion. We omit it here.

3 Verifiable Signature Sharing Scheme

In this section, making use of the revised scheme, we will get a new secure VΣS. VΣS enables the recipient of a digital signature, who is not necessarily the original signer, to share such signature among n proxies so that a subset of them can later reconstruct it, which consists of a sharing phase and a recover phase. At the end of the sharing phase, each proxy can verify that a valid signature for the given document can be reconstructed. At the end of the recover phase, such signature is reconstructed no matter what a malicious subset of proxies may do.

Now we try to present an overview of some basic elements in our work, which is intended as a high-level introduction to some of the issues underlying the protocol design and proofs.

3.1 The Model

Communication Model. We assume that our computation model is composed of three entities: the signer, called Bob, the recipient, called Alice, and a set of n proxies $P_1, ..., P_n$ that can be modeled by probabilistic polynomial-time Turing machines. The VΣS will be run between Alice and the proxies and not involve Bob.

Alice and the proxies are connected by a complex network of private point-to-point channels and by a broadcast channel. These assumptions allow us to focus on a high-level description of the protocols. It is worth noting that these abstraction can be substituted with standard cryptographic techniques for privacy commitment and authentication. Furthermore, we assume communication channel is synchronous.

The Adversary. We assume that there exist an adversary $\mathcal{A}$, who can corrupt Alice and up to t of the n proxies in the network for any value of $t < n/2$, which may be the best achievable threshold or resilience for solution that provide both secrecy and robustness. By corrupting a player, $\mathcal{A}$ can read his memory and cause him to deviate arbitrarily from the protocol. We also assume that adversary $\mathcal{A}$ can be adequately modeled by a probabilistic polynomial time Turing machine and is static, i.e., she chooses the corrupted players at the beginning of the protocol.

NOTATION. In the rest paper, let n denote the number of proxies and $L = n!$.

3.2 Feldman's VSS over Z_N

Catalano and Gennaro [3] show Feldman's VSS over a composite (see Fig.4), which will be used it as a crucial tool to our Verifiable Signature Sharing Scheme, and prove that it is a verifiable secret sharing scheme satisfying unanimity, verifiability and privacy with fault-tolerance t for any n, t with $n > 2t$. Now we give Feldman's VSS over Z_N as following, which we shall infer to as Feldman-Z_N-VSS.

Let $N = pq$ be a composite modulus, which is product of two safe primes such that $p = 2p' + 1, q = 2q' + 1$ for primes p, q, p', q'. We denote with $\phi(N) = (p-1)(q-1) = 4p'q'$, which is the order of the multiplicative group Z_N^* and $\gcd(\phi(N), N) = 1$. Let G_0

be a random element in Z_N^*. It is easy to verify that the order of G_0 is either $p'q'$ or $2p'q'$ (see [3]). Set $G = G_0^{L^3} \ mod \ N$, then G has order $p'q'$. We denote with $DLog_G A \ mod \ N$ the unique integer a such that $A = G^a \ mod \ N$.

Theorem 2 *The Feldman-Z_N-VSS (see Fig.4) is a verifiable secret sharing scheme satisfying unanimity, verifiability and privacy.*

Proof. See [3]. □

3.3 Definition of Verifiable Signature Sharing

VΣS[6] consists of two protocols (ΣShare,ΣRecover) for Alice and n proxies. The inputs of ΣShare for all the players consists of a message and the public verification key VK of the signer. The secret input for Alice is a signature S of m under the signer's key. The output of ΣShare for each proxy P_i is a value S_i, which can assume the special value $S_i = \omega$ denoting that the proxy has rejected the sharing. The protocol ΣRecover is then run on the output of ΣShare by the proxies.

Definition 2 *We say that VΣS is a Verifiable Signature Sharing protocol with fault-tolerance t if, in the presence of an adversary $\mathcal{A}$ that can corrupt Alice and at most t proxies, it satisfies the following properties:*

- ***completeness:*** *If Alice is not corrupted then the output of ΣRecover is a signature S on m under the signer's key VK.*
- ***soundness:*** *If $S_i \neq \omega$ for good player then output of ΣRecover is a signature S on m under the signer's key VK. If a good proxy P_i output $S_i = \omega$ at the end of ΣShare then each good player P_j output $S_j = \omega$.*
- ***security:*** *No information on S can be learned by the adversary except for the commitment of it. More formally, we state this condition in terms of simulation: for every adversary $\mathcal{A}$, there exists a simulator SIM such that, on input m and VK and with black-box access to $\mathcal{A}$, it produces output strings with a distribution which is computationally indistinguishable from the set of messages sent and received by the bad players during the ΣShare protocol.*

We accept a negligible probability that these conditions are violated.

Informally, completeness means that if Alice honestly shares the signature S of m then no matter what malicious proxies do, at the end the signature will be recovered. Soundness means that if Alice is malicious, then she will be caught trying to cheat, or else she will share a valid signature. Security means that running the ΣShare phase, the adversary get no information that he could compute the signature with the message and the public key.

Sharing Phase:

Input for all players: A composite number N, an element $G = G_0^{L^3} \ mod \ N$, where $G_0 \in_R Z_N^*$.

Input for the dealer: A secret $\sigma \in [-N^2, \cdots, N^2]$.

The dealer carries out the following steps:

1. Choose $a_1, \cdots, a_t \in [-L^2N^3, \cdots, L^2N^3]$ and define $f(z) = L\sigma + a_1 z + \cdots + a_t z^t$.

2. Compute $\sigma_i = f(i) \in Z$ for $1 \leq i \leq n$ and $\alpha_i = G^{a_i} \ mod \ N$ for $1 \leq i \leq t$ and $\alpha_0 = G^{\sigma} \ mod \ N$.

3. Send the integer $\sigma_i = f(i)$ to the player P_i and broadcast $\alpha_t, \cdots, \alpha_0$.

Verification steps:

4. Player P_i verifies that

$$G^{\sigma_i} = \alpha_0^L \prod_{j=1}^{t} \alpha_j^{i^j} \ mod \ N \qquad (1)$$

If the equation is not satisfied, he requests that the dealer make $f(i)$ public. If more than t players make this request the dealer is disqualified.

5. The dealer broadcasts all shares requested in the previous step, if he fails to do so he is disqualified.

6. Player P_i carries out the verification of step 1 for all public shares. If the verification fails the dealer is disqualified.

Reconstruction Phase:

Input for all players: The element G, composite N, values $\alpha_t, \cdots, \alpha_0 \ mod \ N$.

Player P_i broadcasts σ_i. Accept those for which Equation(1) is satisfied. Take $t+1$ accepted shares and interpolate over the rational the unique polynomial $f(z)$ of degree t passing through them. Compute the secret σ as $f(0)/L$.

Figure 4. Feldman-Z_N-VSS over a composite modulus

3.4 New Verifiable Signature Sharing Scheme

Now we will give a new verifiable signature sharing scheme (i.e., VΣS protocol) making use of our revised signature scheme and then give its proof of security.

Theorem 3 *The VΣS in Figure 5 is a secure VΣS protocol for our revisited scheme with fault-tolerance t for any n, t with $n > 2t$ except for a negligible probability.*

For completeness of the proof of Theorem 3, we prove the following lemmas in detail here again, although both had been proven in [3].

Lemma 2 *Given a t-adversary who can corrupt at most t players, the view of the adversary of the secret shares generated by the protocol Feldman-Z_N-VSS of a secret σ using a polynomial $f(z)$ such that $f(0) = \sigma$ and of the sharing of a random secret r by a polynomial $r(z)$ with coefficients taken from the appropriate range are statistically indistinguishable.*

Proof. With loss of generality, we can assume that the adversary corrupts the first t player $P_1, ..., P_t$. We prove that with high probability there exists a sharing of r with a polynomial $r(z)$ which satisfies that for each player P_i $(1 \leq i \leq t)$ the share $f(i)$ received in the sharing of σ is equal to the share received in the sharing of r. Furthermore, the coefficients of $r(z)$ are taken from the appropriate range.

Define a t-degree polynomial $h(z)$ such that $h(0) = (\sigma - r)L$ and $h(1) = h(2) = ... = h(t) = 0$. That is, $h(z) = \sum_{i=0}^{t} h(i) \prod_{j \neq i j=0,\cdots,t} \frac{z-j}{i-j}$. The only non-zero value of $h(z)$ is at evaluation point 0. Thus we have that $h(z) = L(\sigma - r) \prod_{j=1}^{t} \frac{z-j}{-j}$ and the coefficient z^i is $L(\sigma - r) \sum_{B \subseteq \{1,\cdots,t\}\ |B|=i} \frac{\prod_{j \in B}(-j)}{\prod_{j=1,\cdots,t}(-j)}$. Because $L = n!$, this value is an integer. Furthermore, the coefficient can be bounded in absolute value by

$$\sum_{B \subseteq \{1,\cdots,t\} | B|=i} L(\sigma - r) \leq (\sigma - r)L \begin{pmatrix} t \\ i \end{pmatrix} \leq \frac{(\sigma-r)Lt!}{i!(t-i)!} \leq (\sigma - r)Lt! \leq L^2N^2 \quad .$$

The desired polynomial $r(z)$ is $f(z) - h(z)$, its coefficients are integers in the range $[-L^2N^3 - L^2N^2, \cdots, L^2N^3 + L^2N^2]$, thus the probability that the coefficients of $r(z)$ will not be in the right range is at most $t\frac{2L^2N^2}{2(L^2N^3+L^2N^2)} \leq \frac{t}{N}$, which is negligible. □

Lemma 3 *Assume $G = G_0^{L^3}$, where $L = n!$ and $G_0 \in_R Z_N^*$. Given values $\sigma_1, \cdots, \sigma_t$ and an additional value G^σ mod$N$, it is possible to compute values $G^{\sigma_1}, \cdots, G^{\sigma_t}$ mod$N$ such that the polynomial $f(z) = L\sigma + a_1 z + \cdots + a_t z^t$ satisfies that $f(z) = \sigma_i$ for $1 \leq i \leq t$*

Proof. Define the polynomial $f(z) = \sum_{i=0}^{t} \sigma_i \prod_{j \neq i} \frac{z-j}{i-j}$, where $\sigma_0 = L\sigma$. Clearly, $f(i) = \sigma_i$ for $1 \leq i \leq t$, thus it remains to be shown that we can compute G raised to the coefficients of $f(z)$. Rearranging terms we have that the coefficient of z^k is

$$a_k = \sum_{i=0}^{t} \frac{\sigma_i}{\prod_{j \neq i}(i-j)} \lambda_{k,i} \ \ (0 \leq k \leq t).$$

Thus

$$\begin{aligned} G^{a_k} &= G^{\sum_{i=0}^{t} \frac{\sigma_i}{\prod_{j \neq i}(i-j)} \lambda_{k,i}} = \prod_{i=0}^{t} G^{\frac{\sigma_i}{\prod_{j \neq i}(i-j)} \lambda_{k,i}} \\ &= G^{\frac{\sigma_0 \lambda_{k,0}}{(-1)^t t!}} \prod_{i=1}^{t} G^{\frac{\lambda_{k,i}\sigma_i}{\prod_{j \neq i}(i-j)}} \\ &= (\alpha_0)^{\frac{L\lambda_{k,0}}{(-1)^t t!}} \prod_{i=1}^{t} (G_0^{\lambda_{k,i}\sigma_i})^{\frac{L^3}{\prod_{j \neq i}(i-j)}}. \end{aligned}$$

Notice the all the exponents are integers now. □

ΣShare:

1. Alice broadcasts e, y to the proxies. She runs Feldman-Z_N-VSS on the secret $\sigma = \alpha$ and with basis $G = (h_1 h_2)^{L^3}$. Let α_0 be the commitment to the secret generated in the Feldman-Z_N-VSS. If Alice is honest, $\alpha_0 = (h_1 h_2)^{L^3 \sigma}$
2. The proxies run the verification phase of Feldman's VSS. They reject if either Alice is disqualified during the verification of Feldman-Z_N-VSS or $\alpha_0 \neq (\frac{y^e}{x h_2^{H(m)}})^{L^3} \ mod\ N$.

ΣRecover:

The proxies run the reconstruction phase of Feldman-Z_N-VSS to recover σ. They compute $\sigma = \alpha \ mod\ p'q'$ and output (e, y, α).

Figure 5 . New Verifiable Signature Sharing Scheme

Proof of theorem 3: Completeness is clear due to Feldman-Z_N-VSS.

Soundness is also quite clear. If the proxies accept then the shared value σ satisfies our revisited equation $y^e = x h_1^{\alpha} h_2^{\alpha + H(m)} \ mod\ N$. So the value reconstructed in ΣRecover must be a correct signature.

Security relies on simulation. Assume w.l.o.g that the adversary corrupts the first t proxies. The simulator $\mathcal{S}$ on input (h_1, h_2, x, N, m) works as follows. It simulates Feldman-Z_N-VSS with $G = (h_1 h_2)^{L^3}$ as basis and $r^* = (y^e / x h_2^{H(m)})^{L^3}$ as public commitment to the secret α. Define $f(z) = \sum_{i=0}^{t} a_i z^t$ the t-degree polynomial $a_i \in [-L^2 N^3, \cdots, L^2 N^3]$ for $1 \leq i \leq t$ such that $f(0) = DLog_G r^*$ and $f(i) = \sigma_i$, so we can compute the values $\alpha_i = G^{a_i} \bmod N$ via "interpolation in the exponent"(in particular $\alpha_0 = r^*$).

By Lemma 4, we can easily find the simulated view of the adversary is defined as the t shares $\sigma_1, \cdots, \sigma_t$ and the pubic values $\alpha_1, \cdots, \alpha_t$. By Lemma 4,we know that the simulated view of adversary and the real one are statistically indistinguishable. This completes the proof. □

4 Conclusion

We presented a revision of Fischlin's signature scheme which is based on Cramer-Shoup signature scheme. The efficient of our signature protocol corresponds to that of Fischlin, where we make use of addition instead of the XOR. Furthermore, we apply our protocol to the wider aspects and especially get an efficient, secure VΣS against static t-adversary.

Acknowledgements. This work is partially supported by HTRP "863" of China (no.2006AA01Z434) and NGRP "973" (no.2007CB311202).

References

[1] Barić, P. B., N. Collision-free accumulators and fail-stop signature schemes without trees. *Proceedings of the Conference on Advances in Cryptology ($EUROCRYPT'97$, Berlin,Germany), W.Fumy,Ed., Springer-Verlag, New York*, 1997.

[2] Bellare, M. S., M. How to sign given any trapdoor permutation. *Journal of the ACM*, 39(1):214–233, January 1992.

[3] Catalano, D. and Gennaro, R. New efficient and secure protocols for verifiable signature and other applications. *CRYPTO'98, Lecture Notes in Computer Science, Springer-Verlag*, 1998.

[4] Cramer, R. and Shoup, V. Signature schemes based on the strong rsa assumption. *ACM Transactions on Information and System Security(ACM TISSEC)*, 2000.

[5] Fischlin, M. The cramer-shoup strong-rsa signature scheme revisited. *Proceedings of the PKC 2003,Lecture Notes in Computer Science,Berlin:Springer-Verlag*, 2003.

[6] Franklin, R. M., M. Verifiable signature sharing. *Eurocrypt'95, Lecture Notes in Computer Science, Springer-Verlag*, 1995.

[7] Fujisaki, O. T., E. Statistical zero knowledge protocols to prove modualr polynomial relations. *Proceedings of the 17th Annual International Conference on Advances in Cryptology (CRYPTO'97), Springer-Verlag, New York*, August 1997.

[8] Gennaro, H. S., R. and Rabin, T. Secure hash-and-sign signatures without the random oracle. *Proceedings of the Conference on Advances in Cryptology (EUROCRYPT'99). Springer-Verlag,New York,NY*, August 1999.

[9] Goldwasser, M. S., S. and Rivest, R. L. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM J.Comput.*, 17(2):281–308, April 1988.

[10] Naor, Y. M., M. Universal one-way hash functions and their cryptographic applications. *Proc.21st ACM Symp. on Theory of Computing*, 1989.

[11] Rompel, J. One-way functions are necessary and sufficient for secure signatures. *Proc. 22nd ACM Symp.on Theory of Computing*, 1990.

[12] SHA and NIST. Secure hash standard. *National Institute of Standards and Technology,Gaithersburg,MD*.

[13] Yang, H. L. L. K., Y. and Zhu, Y. A revision of cramer-shoup strong-rsa signature scheme. *Proceeding of the 17th National Conference on Tecnology of Information Confidentiality (China)*, 2007.