



## **New Zero-knowledge Undeniable Signatures - Forgery of Signature Equivalent to Factorisation**

Wenbo Mao  
Trusted E-Services Laboratory  
HP Laboratories Bristol  
HPL-2001-36  
February 28<sup>th</sup> , 2001\*

E-mail: [wm@hplb.hpl.hp.com](mailto:wm@hplb.hpl.hp.com)

undeniable  
signatures,  
efficient zero-  
knowledge  
protocols

We propose a new zero-knowledge undeniable signature scheme which is based on the intractability of computing high-order even powers modulo a composite. The new scheme has a number of desirable properties: (i) forgery of a signature (including existential forgery) is proven to be equivalent to factorisation, (ii) perfect zero-knowledge, (iii) efficient protocols for signature verification and non-signature denial: both measured by  $O(\log k)$  (multiplications) where  $1/k$  bounds the probability of error. For a denial protocol, this performance is unprecedented.

# New Zero-knowledge Undeniable Signatures - Forgery of Signature Equivalent to Factorisation

Wenbo Mao  
Hewlett-Packard Laboratories  
Filton Road, Stoke Gifford  
Bristol BS34 8QZ  
United Kingdom  
wm@hplb.hpl.hp.com

February 12, 2001

## Abstract

We propose a new zero-knowledge undeniable signature scheme which is based on the intractability of computing high-order even powers modulo a composite. The new scheme has a number of desirable properties: (i) forgery of a signature (including existential forgery) is proven to be equivalent to factorisation, (ii) perfect zero-knowledge, (iii) efficient protocols for signature verification and non-signature denial: both measured by  $O(\log k)$  (multiplications) where  $1/k$  bounds the probability of error. For a denial protocol, this performance is unprecedented.

**Keywords** Undeniable signatures, Efficient zero-knowledge protocols.

## 1 Introduction

Undeniable signatures pioneered by Chaum and van Antwerpen [2, 3] offer good privacy service for a signer. An undeniable signature is not self-authenticating, meaning: the verification of a signature can only be done by interacting with the signer. This property can be very useful in some electronic commerce applications such as privacy preserving auctions and contract signing.

The zero-knowledge undeniable signatures of Chaum works in a multiplicative group modulo a prime number [3]. Gennaro et al extended this notion to working in a group

modulo a composite number [4]. The latter is called the RSA-based undeniable signatures. Both share a number of attractive virtues: practical efficiency, perfect zero-knowledge and the weakest possible assumption on the prover (signer) that (s)he may be computationally unbounded.

Nevertheless, we can identify that in two aspects the previous work is inadequate.

First, missing of the exact security against signature forgery. None of the previous work could have related the difficulty of signature forgery to a standard intractability assumption.

Second, comparatively poor performance of denial protocols. Any undeniable signature scheme has to be equipped with a denial protocol which allows an alleged signer to deny an invalid signature (let us call an alleged signer with respect to an invalid signature *non-signer* and *non-signature*, respectively). Denial protocols in the previous undeniable signature schemes [3, 4] used Chaum's idea of showing inequality between the discrete logarithms of two elements, one is related to a non-signature, and the other, to a piece of information in the non-signer's key certificate. Unlike the case of showing equality of discrete logarithms (used by the signature verification protocols), Chaum's idea for demonstrating discrete logarithm inequality is by testing a non-signer's ability to perform brute-force search of a number hidden in a challenge (a real signer will not be able to do so even computationally unbounded). In order to make the brute-force search a feasible job the space to be searched is rendered to a small size and consequently, the search protocol has a no-small error probability. In order to obtain a desirably low probability for a correct denial of a non-signature, the search protocol is re-run for a desirable number of times. What makes the matter worse is that the search protocol in its basic version (e.g., as specified in [4]) is not zero-knowledge which spoils the perfect perfect zero-knowledgeness of a zero-knowledge undeniable signature scheme. However, it can be modified to being zero-knowledge by using cryptographic commitment schemes (Chaum did so in [3]). The use of a commitment scheme multiplies more rounds of interactions and adds a non-trivial computational load.

Denial protocols are regarded as the bottleneck for the previous undeniable signature schemes.

## 1.1 Our Work

We propose a new undeniable signature scheme based on the intractability of computing high-order even powers modulo a composite. Our scheme augments the virtues of undeniable signatures that we have listed earlier with two items:

- i) Exact security against forging: proven to be equivalent to factorisation, and
- ii) A new denial protocol with improved efficiency: measured by  $O(\log k)$  (multiplications) where  $1/k$  bounds the probability of error.

Like the RSA-based undeniable signature scheme of Gennaro et al [4], our scheme works in the multiplicative group modulo an RSA modulus. For a message  $M < n$  encoded as a quadratic residue, its undeniable signature is a  $2^u$ -th root of  $M$  modulo  $n$ ; namely, let  $(S, M)$  be a signature pair, they are related as follows

$$M \equiv S^{2^u} \pmod{n}. \quad (1)$$

Clearly,  $u = 1$  provides a special case of Rabin's signature scheme [5] which can be verified publicly (i.e., is self-authenticating). We shall use  $u$  sufficiently large to render an infeasibility for self-authentication. We outline below how this is done.

First, we note that without the factorisation of  $n$ , validating (1) can be done in  $u$  squarings modulo  $n$ . Secondly, because each squaring can only be performed on the result of the previous squaring, it is not known how to speedup these  $u$  squarings via parallelisation of multiple processors. Parallelisation of each squaring step cannot achieve a great deal of speedup since a squaring step only needs a trivial computational resource and so any non-trivial scale of parallelisation of a squaring step is likely to be penalised by communication delays among the processors. Thus, we believe that for  $u > 100$ , validating (1) without the signer's help is an intractable task. (The intractability and non-parallelisability have been used as the security basis for the "Time-Lock Puzzle" of Rivest et al [6].) Thirdly, we shall see that knowing the factorisation of  $n$ , creating pair  $(S, M)$  satisfying (1) can be efficiently done either from  $M$  or from  $S$ .

Thus, pairs that are related in (1) with a large  $u$  can indeed provide an undeniable signature service, provided we can devise secure and efficient zero-knowledge protocols for verifying a valid signature and for denying an invalid one. These will be the topic of this paper.

Finally, we should notice that, when necessary, it is easy for the signer to convert undeniable signature pair  $(S, M)$  into a Rabin signature pair  $(S^{2^{u-1}} \pmod{n}, M)$  and thereby turn an undeniable signature into a self-authenticating signature. In fact, it is this close relationship between our undeniable signature scheme and the Rabin signature scheme that enables us to claim the feature of forging equivalent to factorisation (to be analysed in §6).

## 1.2 Organisation

In the next section we agree on notations to be used in the paper. In Section 3 we construct a protocol for certificate establishment. In Section 4 we construct a protocol for proving/verifying a valid pair of undeniable signature. In Section 5 we construct a protocol for denying a non-signature. In Section 6 we discuss the security issues.

## 2 Notation

Throughout the paper we use the following notation.  $Z_n$  denotes the ring of integers modulo  $n$ .  $Z_n^*$  denotes the multiplicative group of integers modulo  $n$ .  $\phi(n)$  denotes Euler's phi function of  $n$ , which is the order, i.e., the number of elements, of the group  $Z_n^*$ . For an element  $a \in Z_n^*$ ,  $Order_n(a)$  denotes the multiplicative order modulo  $n$  of  $a$ , which is the least index  $i$  satisfying  $a^i \equiv 1 \pmod{n}$ ;  $\langle a \rangle$  denotes the subgroup generated by  $a$ ;  $\left(\frac{x}{n}\right)$  denotes the Jacobi symbol of  $x \pmod{n}$ . We denote by  $J_+(n)$  the subset of  $Z_n^*$  containing the elements of the positive Jacobi symbol. For integers  $a, b$ , we denote by  $gcd(a, b)$  the greatest common divisor of  $a$  and  $b$ , by  $lcm(a, b)$  the least common multiple of  $a$  and  $b$ , by  $a \parallel b$  the concatenation of the binary bits of  $a$  and  $b$ , and by  $|a|$  the binary length of  $a$ . For a real number  $r$ , we denote by  $\lfloor r \rfloor$  the floor of  $r$ , i.e.,  $r$  round down to the nearest integer. For an event  $E$ , we denote by  $Pr[E]$  the probability for  $E$  to occur.

## 3 Key Generation and Certificate Establishment

Let Alice be a user. In order to use the scheme, she should first construct her RSA modulus  $n$  with a safe-prime structure. This requires  $n = pq$ ,  $p' = (p-1)/2$ ,  $q' = (q-1)/2$  where  $p, q, p'$  and  $q'$  are all distinct primes of roughly equal size. She should prove in zero-knowledge to a key certification authority (CA) such a structure of  $n$ . This can be achieved via using, e.g., the protocol of Camenisch and Michels [1].

Alice should generate the primes  $p', q'$  such that  $Order_{2p'q'}(2)$  is sufficiently large (e.g.,  $Order_{2p'q'}(2) > 2^{150}$ ). We will need this property in §6.2.

Let  $w \in Z_n^*$  satisfy

$$gcd(w \pm 1, n) = 1, \tag{2}$$

$$\left(\frac{w}{n}\right) = -1. \tag{3}$$

It is elementary to show that  $w$  satisfying (2) and (3) has the full order  $2p'q'$ . The following lemma observes a property of  $w$ .

**Lemma 1** *Let  $n$  be an RSA modulus of a safe-prime structure and  $w \in Z_n^*$  of the full order. Then for any  $x \in Z_n^*$ , either  $x \in \langle w \rangle$  or  $-x \in \langle w \rangle$ .*

**Proof** It is easy to check  $-1 \notin \langle w \rangle$ . So  $\langle w \rangle$  and the coset  $(-1)\langle w \rangle$  both have the half the size of  $Z_n^*$ , yielding  $Z_n^* = \langle w \rangle \cup (-1)\langle w \rangle$ . Any  $x \in Z_n^*$  is either in  $\langle w \rangle$  or in  $(-1)\langle w \rangle$ . The latter case means  $-x \in \langle w \rangle$ .  $\square$

### 3.1 A Building Block Protocol

Let Alice and CA have agreed on  $n$  (based on CA's satisfaction on Alice's proof that  $n$  has a safe-prime structure). Figure 1 specifies a perfect zero-knowledge protocol for

$SQ(w, x, y, n)$

Input Common:  $n$ : an RSA modulus with a safe-prime structure;

$w \in Z_n^*$ : an element of the full-order  $2p'q' = \phi(n)/2$  (so  $w \not\equiv \pm 1 \pmod{n}$ );

$x, y \in J_+(n)$ :  $x \not\equiv \pm y \pmod{n}$ ;

Alice:  $z$ :  $x \equiv \pm w^z \pmod{n}$ ,  $y \equiv \pm w^{z^2} \pmod{n}$ ;

1. CA chooses at random  $r < n$ ,  $s < n$  and computes

$$C \stackrel{\text{def}}{=} w^r x^s \pmod{n}, \quad R \stackrel{\text{def}}{=} x^r y^s \pmod{n};$$

He only sends  $C$  to Alice;

2. Alice chooses at random  $t < n$  and sends to CA:

$$R_1 \stackrel{\text{def}}{=} C^t \pmod{n}, \quad R_2 \stackrel{\text{def}}{=} C^z w^t \pmod{n};$$

3. CA sends to Alice:  $r, s$ ;

4. Alice aborts proof if  $C \not\equiv w^r x^s \pmod{n}$ , otherwise sends to CA:  $t$ ;

5. CA accepts if  $R_1 \equiv C^t \pmod{n}$ ,  $R_2/w^t \equiv R \pmod{n}$ , or rejects otherwise.

Figure 1: A Perfect Zero-knowledge Protocol for Squaring Discrete Logarithm

Alice to prove that for  $w, x, y \in Z_n^*$  with  $n$  of a safe-prime structure,  $w$  of the full order, and  $x, y \in J_+(n)$ , they satisfy (note,  $\pm$  below means either  $+$  or  $-$ , but not both)

$$\exists z : x \equiv \pm w^z \pmod{n}, \quad y \equiv \pm w^{z^2} \pmod{n}. \quad (4)$$

Alice should of course have constructed  $w, x, y$  to satisfy (4). She sends  $w, x, y$  to CA.

CA (has verified  $n$  of a safe-prime structure) should first check (2) and (3) on  $w$  for its full-order status (which also shows  $w \not\equiv \pm 1 \pmod{n}$ ); he should also check  $x, y \in J_+(n)$ .

**Theorem 1** *Let  $w, x, y, n$  be as specified in the common input in Protocol  $SQ$ . The protocol has the following properties:*

**Completeness** *There exists  $z \in Z_n$  and  $x, y \in Z_n^*$  satisfying (4); for these values CA will always accept Alice's proof;*

**Soundness** *If (4) does not hold for the common input, then Alice, even computationally unbounded, cannot convince CA to accept her proof with probability greater than  $\frac{2p'+2q'-1}{2p'q'}$ .*

**Zero-knowledge** *CA, even dishonest, gains no information about Alice's private input.*

---

<sup>1</sup>The safe-prime structure of  $n$  implies  $p' \approx q' \approx \sqrt{n}$  and hence this probability value is approximately  $1/\sqrt{n}$ .

**Proof**

**Completeness** For any  $z \in Z_n$ , let  $x = w^z \pmod n$ ,  $y = w^{z^2} \pmod n$  (both in the plus case). It is evident from inspection of the protocol that CA will always accept Alice's proof.

**Soundness** Suppose that (4) does not hold whereas CA has accepted Alice's proof.

The first congruence of (4) holds as a result of Lemma 1. So it is the second congruence that does not hold. Let  $\xi \in Z_n^*$  satisfy

$$y \equiv \xi w^{z^2} \pmod n \quad \text{with } \text{Order}_n(\xi) > 2. \quad (5)$$

By asserting  $\text{Order}_n(\xi) > 2$  we exclude the cases for  $\xi$  being any square root of 1, which consists of either  $\pm 1$ , or the other two roots which will render  $y \notin J_+(n)$ .

We only need to consider the case  $x \equiv -w^z \pmod n$ . The other case  $x \equiv w^z \pmod n$  is completely analogous (and easier).

Since CA accepts the proof, he sees the following two congruences

$$C \equiv w^r x^s \pmod n, \quad (6)$$

$$R \equiv R_2/w^t \equiv x^r y^s \pmod n. \quad (7)$$

Examining (6), we see that  $C \equiv w^r (-x)^s \in \langle w \rangle$  if  $s$  is even, or  $-C \equiv w^r (-x)^s \in \langle w \rangle$  if  $s$  is odd. So for either cases of  $s$ , we are allowed to re-write (6) into the following linear congruence with  $r$  and  $s$  as unknowns

$$\log_w \pm C \equiv r + sz \pmod{2p'q'}.$$

For every case of  $s = 1, 2, \dots, 2p'q'$ , this linear congruence has a value for  $r$ . This means that for any fixed  $C$ , (6) has exactly  $2p'q'$  pairs of solutions. Each of these pairs will yield an  $R$  from (7). Below we argue that for any two solution pairs from (6), which we denote by  $(r, s)$  and  $(r', s')$ , if  $\gcd(s - s', 2p'q') \leq 2$  then they must yield  $R \not\equiv R' \pmod n$ . Suppose on the contrary

$$w^r x^s \equiv C \equiv w^{r'} x^{s'} \pmod n, \quad \text{i.e., } w^{r-r'} \equiv x^{s'-s} \pmod n, \quad (8)$$

it also holds

$$x^r y^s \equiv R \equiv R' \equiv x^{r'} y^{s'} \pmod n, \quad \text{i.e., } x^{r-r'} \equiv y^{s'-s} \pmod n. \quad (9)$$

Using (8) and (5) with noticing  $x \equiv -w^z$ , we can transform (9) into

$$(-1)^{[r-r'+z(s'-s)]} w^{[z^2(s'-s)]} \equiv x^{r-r'} \equiv y^{s'-s} \equiv \xi^{s'-s} w^{[z^2(s'-s)]} \pmod n,$$

which yields

$$\xi^{s'-s} \equiv (-1)^{[r-r'+z(s'-s)]} \equiv \pm 1 \pmod n, \quad \text{i.e., } \xi^{2(s'-s)} \equiv 1 \pmod n. \quad (10)$$

Recall that  $Order_n(\xi) > 2$  which implies  $Order_n(\xi)$  being a multiple of  $p'$  or  $q'$  or both. However,  $gcd(s' - s, 2p'q') \leq 2$ , i.e.,  $gcd(2(s' - s), 2p'q') = 2$ , so  $2(s' - s)$  cannot be such a multiple. Consequently (10) cannot hold and we reach a contradiction.

For any  $s \leq 2p'q'$ , it's routine to check that there are  $2p' + 2q' - 2$  cases of  $s'$  satisfying  $gcd(2(s' - s), 2p'q') > 2$ . Thus, if (4) does not hold, amongst  $2p'q'$  possible  $R$ 's matching the challenge  $C$ , there are in total  $2p' + 2q' - 1$  of them (matching  $s$  and the other  $2p' + 2q' - 2$   $s$ 's) that may collide to CA's fixing of  $R$  in Step 1. Even computationally unbounded, Alice will have at best  $\frac{2p'+2q'-1}{2p'q'}$  probability to have responded correctly.

**Zero-Knowledge** If the protocol ever proceeds to Step 5, i.e., CA causes Alice to disclose any of her knowledge, CA has already known the response  $R$  in Step 1. So no knowledge whatsoever has been disclosed to CA.  $\square$

### 3.2 Proof of Correct Construction of $w^{2^u} \pmod n$

Define

$$w(u) \stackrel{def}{=} w^{2^u} \pmod n. \quad (11)$$

With the factorisation of  $n$ , Alice can construct  $w(u)$  in  $O(\log n)$  multiplications via the following two steps:

$$v \stackrel{def}{=} 2^u \pmod{\phi(n)}, \quad (12)$$

$$w(u) \stackrel{def}{=} w^v \pmod n. \quad (13)$$

For  $u \geq 1$ , we can express  $2^u$  as

$$2^u = \begin{cases} 2^{[2 \cdot (u/2)]} = [2^{(u/2)}]^2 & \text{if } u \text{ is even} \\ 2^{[2 \cdot (u-1)/2 + 1]} = [2^{(u-1)/2}]^2 \cdot 2 & \text{if } u \text{ is odd} \end{cases}$$

Copying this expression to the exponent position of  $w^{2^u} \pmod n$ , we can express

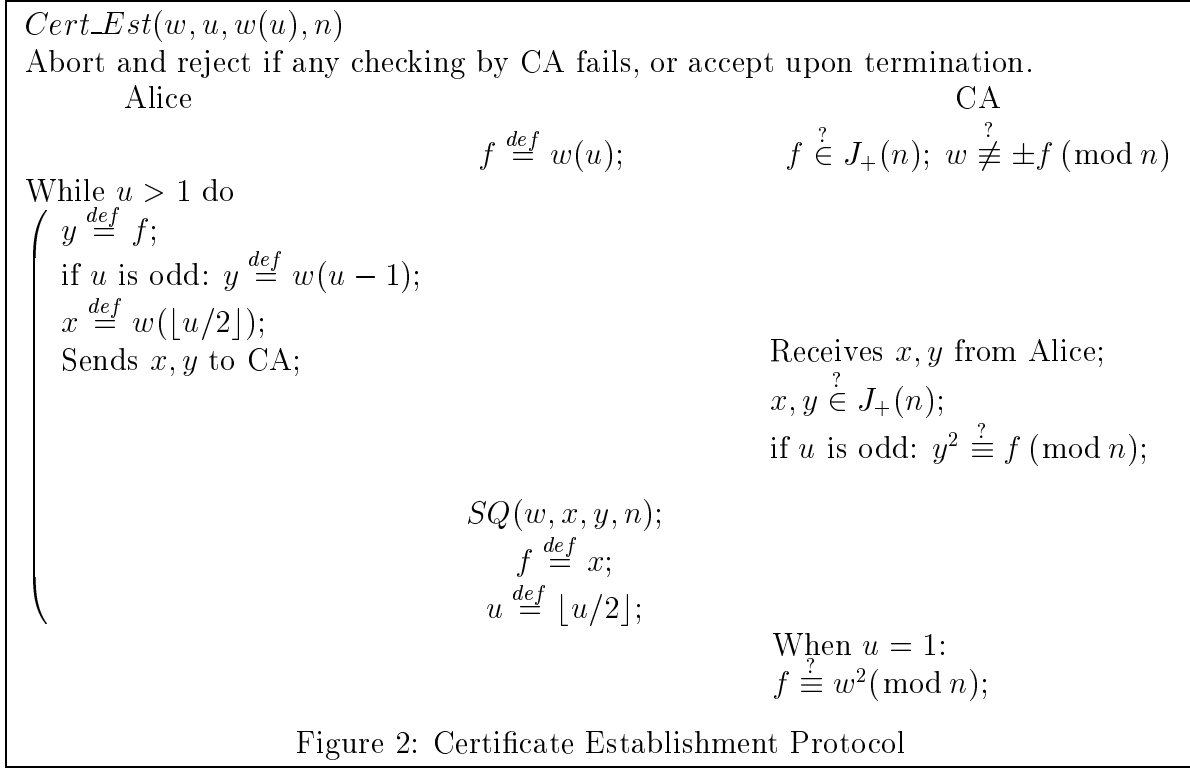
$$w^{2^u} \pmod n \equiv \begin{cases} w^{[2^{(u/2)}]^2} & \text{if } u \text{ is even} \\ (w^{[2^{(u-1)/2}]})^2 & \text{if } u \text{ is odd} \end{cases} \quad (14)$$

In (14) we see that the exponent  $2^u$  can be expressed as the square of another power of 2 with  $u$  being halved in the latter. This observation suggests that repeatedly using  $SQ$ , we can demonstrate, in  $\lceil \log_2 u \rceil$  steps, that the discrete logarithm of an element is of the form  $2^u$ . This observation translates precisely into the protocol specified in Figure 2 which will terminate within  $\log_2 u$  steps and prove the correct structure of

$$w(u+1) \equiv w^{2^{u+1}} \pmod n.$$

We shall call this protocol *Certificate Establishment Protocol* since the correctly established pair  $(w, w(u+1))$  will form a signature reference pair to be placed in Alice's key





certificate. The protocol is presented in three columns: the actions in the left column are performed by Alice, those in the right column, by CA, and those in the middle, by the both parties.

A run of  $Cert\_Est(w, u, w(u), n)$  will terminate within  $\lfloor \log_2 u \rfloor$  loops, and this is the completeness property. The perfect-zero-knowledge property follows that of  $SQ$ . We only have to show the soundness property.

**Theorem 2** *Let  $n = (2p' + 1)(2q' + 1)$  be an RSA modulus of a safe-prime structure,  $w \in Z_n^*$  be of the full order  $2p'q'$ , and  $u > 1$ . Upon acceptance termination of  $Cert\_Est(w, u, w(u), n)$ , relation  $w(u) \equiv w^{2^u} \pmod{n}$  holds with probability greater than*

$$1 - \frac{\lfloor \log_2 u \rfloor (2p' + 2q' - 1)}{2p'q'}$$

**Proof** Denote by  $SQ(w, x_1, y_1, n)$  and by  $SQ(w, x_2, y_2, n)$  any two consecutive acceptance calls of  $SQ$  in  $Cert\_Est$  (so  $y_1 = w(u)$  in the first call, and  $x_2 = w^2$  in the last call, of  $SQ$  in  $Cert\_Est$ , respectively). When  $u > 1$ , such two calls prove that there exists  $z$ :

$$x_2 \equiv \pm w^z \pmod{n}, \quad y_2 \equiv \pm w^{2^2} \pmod{n}, \quad (15)$$

and either

$$x_1 = y_2 \equiv \pm w^{z^2} \pmod{n}, \quad y_1 \equiv \pm w^{z^4} \pmod{n}, \quad (16)$$

or

$$x_1 = y_2^2 \equiv w^{2z^2} \pmod{n}, \quad y_1 \equiv \pm w^{4z^4} \pmod{n}. \quad (17)$$

Upon  $u = 1$ , CA further sees that  $x_2 = w^2$ . By induction, the exponents  $z$  (resp.  $z^2, z^4, 2z^2, 4z^4$ ) in all cases of  $\pm w^z$  (resp.  $\pm w^{z^2}, \dots$ ) in (15), (16) or (17) contain a single factor: 2, and the minus symbol disappears from (15), (16) and (17) since the even exponents imply all cases of  $x$  and  $y$  to be quadratic residues. So we can write  $w(u) = w^{2^t} \pmod{n}$  for some natural number  $t$ .

Further note that each all of  $SQ$  causes an effect of having  $2^t$  square-rooted in the integers which is equivalent to having  $t$  halved in the integers. Thus, exactly  $\lfloor \log_2 t \rfloor$  calls (and no more) of  $SQ$  can be made. But CA has counted  $\lfloor \log_2 u \rfloor$  calls of  $SQ$ , therefore  $t = u$ .

Each acceptance call of  $SQ$  has the correctness probability of  $1 - \frac{2p'+2q'-1}{2p'q'}$ . So after  $\lfloor \log_2 u \rfloor$  acceptance calls of  $SQ$ , the probability for  $Cert\_Est$  to be correct is

$$\left(1 - \frac{2p' + 2q' - 1}{2p'q'}\right)^{\lfloor \log_2 u \rfloor} > 1 - \frac{\lfloor \log_2 u \rfloor (2p' + 2q' - 1)}{2p'q'}. \quad \square$$

### 3.3 Certificate Issuance

For  $u > 2^{100}$ , upon acceptance of  $Cert\_Est(w, u, w(u), n)$ , CA shall issue a certificate for Alice. Denote by

$$Cert(Alice, w, w(u), n)$$

the certificate, which is signed by CA. Anybody, upon seeing  $Cert(Alice, w, w(u), n)$  and trusting CA, understands that the pair  $w, w(u)$  forms a reference pair of Alice's undeniable signature.

### 3.4 Performance

It is obvious that by preparing all the intermediate values in advance,  $Cert\_Est$  can be run in parallel to save the  $\lfloor \log_2 u \rfloor$  rounds of interactions. The number of bits to be exchanged is measured by  $O((\lfloor \log_2 u \rfloor)(\log_2 n))$ .

In each run of  $SQ$ , Alice (resp. CA) performs four (resp. six) exponentiation(s) mod  $n$ . So in  $Cert\_Est(w, u, w(u), n)$  Alice (resp. CA) will perform  $\lfloor 4 \log_2 u \rfloor$  (resp.  $6 \lfloor \log_2 u \rfloor$ ) exponentiations mod  $n$ . These translate to  $O(\lfloor \log_2 u \rfloor (\log_2 n)^3)$  bit operations. This performance is suitable for practical use.

## 4 Signature Creation and Verification

### 4.1 Signature Creation

Denote by  $\hat{M}$  a properly randomized message  $M$  such that  $\hat{M}$  is a quadratic residue modulo  $n$ . For example, let  $h()$  be a cryptographically secure hash function and  $r$  be a random number of the size  $|n| - |h|$ , then the following is an acceptable message randomisation scheme

$$\hat{M} \stackrel{def}{=} h(M, r) \parallel r.$$

With our modulus  $n$ , it is expected that, within every four trial of random  $r$ ,  $\hat{M}$  will be a quadratic residue modulo  $n$ . With the specification of the hash function  $h$ , it is easy to verify the relation between  $M$  and  $\hat{M}$ .

Let

$$v \stackrel{def}{=} \left( \frac{4p'q' + 4}{8} \right)^u \pmod{\phi(n)}. \quad (18)$$

Then for any  $\hat{M}$ , it is routine to check that

$$S(\hat{M}) \stackrel{def}{=} \hat{M}^v \sqrt{1} \pmod{n} \quad (19)$$

is a  $2^u$ -th root of  $\hat{M}$  (here  $\sqrt{1}$  denotes any square root of 1 modulo  $n$ ). It is clear that while  $S(\hat{M})^2$ , the  $2^{u-1}$ -th root of  $\hat{M}$ , is unique, the existence of four square roots of 1 for our modulus (a Blum integer) renders four possible  $S(\hat{M})$  for each  $\hat{M}$ . In the sequel we stipulate that Alice uses (18) and (19) to compute  $S(\hat{M})$  from  $\hat{M}$  in which the use of one of four square roots of 1 is decided by coin flipping.

For message  $M$ , the pair  $(S(\hat{M}), \hat{M})$  constitutes Alice's undeniable signature.

### 4.2 Zero-knowledge Signature Verification

Let Bob be a signature verifier and let Alice have sent to Bob her alleged signature pair  $(S(\hat{M}), \hat{M})$  and her key certificate  $Cert(Alice, w, w(u), n)$ . Figure 3 describes a perfect zero-knowledge protocol run between Alice and Bob to establish

$$\exists z \equiv 2^u \pmod{\phi(n)} : w(u) \equiv w^z \pmod{n}, \quad \hat{M} \equiv S(\hat{M})^z \pmod{n}. \quad (20)$$

Comparing (20) with (4), we note that because Bob knows that Alice's private input is even (CA has certified so), all the minus cases in (4) disappear from (20). Therefore the protocol *Sig\_Verify* in Figure 3 is essentially the same as *SQ* in Figure 1 except that here there is no need to deal with the minus cases.

**Theorem 3** *Let  $w, w(u), S(\hat{M}), \hat{M}, n$  be as specified in the common input in Protocol *Sig\_Verify*. The protocol has the following properties:*

$Sig\_Verify(w, w(u), S(\hat{M}), \hat{M}, n)$

Input Common:  $n$ : an RSA modulus with a safe-prime structure;

$w \in Z_n^*$ : an element of the full-order  $2p'q' = \phi(n)/2$  (so  $w \not\equiv \pm 1 \pmod{n}$ );

$w(u), S(\hat{M}), \hat{M} \in Z_n^*$ :  $w(u)$  is a quadratic residue;

Alice:  $z \equiv 2^u \pmod{\phi(n)}$ :  $w(u) \equiv w^z \pmod{n}$ ,  $\hat{M} \equiv S(\hat{M})^z \pmod{n}$ ;

1. Bob chooses at random  $a < n, b < n$  and computes

$$C \stackrel{def}{=} w^a S(\hat{M})^b \pmod{n}, \quad R \stackrel{def}{=} w(u)^a \hat{M}^b \pmod{n};$$

He only sends  $C$  to Alice;

2. Alice chooses at random  $t < n$  and sends to Bob:

$$R_1 \stackrel{def}{=} C^t \pmod{n}, \quad R_2 \stackrel{def}{=} C^z w^t \pmod{n};$$

3. Bob sends to Alice:  $a, b$ ;

4. Alice aborts proof if  $C \not\equiv w^a S(\hat{M})^b \pmod{n}$ , otherwise sends to CA:  $t$ ;

5. CA accepts if  $R_1 \equiv C^t \pmod{n}$ ,  $R_2/w^t \equiv R \pmod{n}$ , or rejects otherwise.

Figure 3: A Perfect Zero-knowledge Protocol for Signature Verification

**Completeness** *Bob will always accept Alice's proof;*

**Soundness** *If (20) does not hold for the common input, then Alice, even computationally unbounded, cannot convince Bob to accept her proof with probability greater than  $\frac{2p'+2q'-1}{2p'q'}$ .*

**Zero-knowledge** *Bob, even dishonest, gains no information about Alice's private input.*

Due to the similarity between  $Sig\_Verify$  and  $SQ$ , the proof is essentially the same as that for Theorem 1, with only minor difference in the soundness part. So we only need to prove the soundness.

**Proof of Soundness** Suppose that (20) does not hold whereas Bob has accepted Alice's proof.

The first congruence in (20) holds as is certified by CA. So it is the second congruence that does not hold. Let  $\xi \in Z_n^*$  satisfy

$$\hat{M} \equiv \xi S(\hat{M})^z \pmod{n} \quad \text{with} \quad \xi \not\equiv 1 \pmod{n}. \quad (21)$$

Since Bob accepts the proof, he sees the following two congruences

$$C \equiv w^a S(\hat{M})^b \pmod{n}, \quad (22)$$

$$R \equiv R_2/w^t \equiv w(u)^a \hat{M}^b \pmod{n}. \quad (23)$$

Examining (22), we see that  $C \equiv w^a(-S(\hat{M}))^b \in \langle w \rangle$  if  $b$  is even, or  $-C \equiv w^r(-S(\hat{M}))^b \in \langle w \rangle$  if  $b$  is odd (because  $w$  is of the full order, either of the two cases holds due to Lemma 1). So for either cases of  $b$ , we are allowed to re-write (22) into the following linear congruence with  $a$  and  $b$  as unknowns

$$\log_w \pm C \equiv a + bz' \pmod{2p'q'}.$$

(Note that  $z'$  has no relationship with  $z$ .) For every case of  $b = 1, 2, \dots, 2p'q'$ , this linear congruence has a value for  $a$ . This means that for any fixed  $C$ , (22) has exactly  $2p'q'$  pairs of solutions. Each of these pairs will yield an  $R$  from (23). Below we argue that for any two solution pairs from (22), which we denote by  $(a, b)$  and  $(a', b')$ , if  $\gcd(b - b', 2p'q') \leq 2$  then they must yield  $R \not\equiv R' \pmod{n}$ . Suppose on the contrary

$$w^a S(\hat{M})^b \equiv C \equiv w^{a'} S(\hat{M})^{b'} \pmod{n}, \quad \text{i.e.,} \quad w^{a-a'} \equiv S(\hat{M})^{b'-b} \pmod{n}, \quad (24)$$

it also holds

$$w(u)^a \hat{M}^b \equiv R \equiv R' \equiv w(u)^{a'} \hat{M}^{b'} \pmod{n}, \quad \text{i.e.,} \quad w(u)^{a-a'} \equiv \hat{M}^{b'-b} \pmod{n}. \quad (25)$$

Exponentiating both sides of the second congruence in (24) with  $z$  and noticing (21) and  $w^z \equiv w(u) \pmod{n}$  (blessed by CA), we have

$$w(u)^{a-a'} \equiv \left(\frac{\hat{M}}{\xi}\right)^{b'-b} \pmod{n}$$

or

$$\xi^{b'-b} w(u)^{a-a'} \equiv \hat{M}^{b'-b} \pmod{n}$$

Comparing this with the second congruence in (25), we derive

$$\xi^{b'-b} \equiv 1 \pmod{n}. \quad (26)$$

Note that  $\gcd(b' - b, 2p'q') \leq 2$ . So (26) holds for all odd cases of  $b' - b$  which is not a multiple of  $p'$  or  $q'$ . In  $Z_n^*$ , only 1 has this property. So  $\xi = 1$  and we reach a contradiction to (21).

For any  $b \leq 2p'q'$ , it's routine to check that there are  $2p' + 2q' - 2$  cases of  $b'$  satisfying  $\gcd(b' - b, 2p'q') > 2$ . Thus, if the second congruence in (20) does not hold, amongst  $2p'q'$  possible  $R$ 's matching the challenge  $C$ , there are in total  $2p' + 2q' - 1$  of them (matching  $b$  and the other  $2p' + 2q' - 2$   $b$ 's) that may collide to CA's fixing of  $R$  in Step 1. Even computationally unbounded, Alice will have at best  $\frac{2p'+2q'-1}{2p'q'}$  probability to have responded correctly.  $\square$

$Denial(w, w(u), S(\hat{M}), \hat{M}, n)$

Input Common:  $n$ : an RSA modulus with a safe-prime structure;

$w \in Z_n^*$ : an element of the full-order  $2p'q' = \phi(n)/2$  (so  $w \not\equiv \pm 1 \pmod{n}$ );

$w(u), S(\hat{M}), \hat{M} \in Z_n^*$ :  $w(u)$  is a quadratic residue;

Alice:  $z \equiv 2^u \pmod{\phi(n)}$ :  $w(u) \equiv w^z \pmod{n}$ ;

1. Alice picks at random  $e$  satisfying  $\gcd(e, \phi(n)) = 1$ ;

she computes  $T \stackrel{def}{=} (S(\hat{M})^e)^{2^u} \pmod{n}$  and sends to Judge:  $e, T$ ;

2. Alice and Judge run  $Sig\_Verify(w, w(u), S(\hat{M})^e, T, n)$ ;

3. Judge accepts the allegation if the run returns rejection;

Otherwise, Judge tests  $T \stackrel{?}{\neq} \hat{M}^e \pmod{n}$ ;

Judge dismisses (or accepts) the allegation if the inequality (equality) holds;

Figure 4: A Perfect Zero-knowledge Protocol for Non-Signature Denial

### 4.3 Performance

In signature creation, let Alice have pre-computed  $v$  and  $\sqrt{1}$  (see (18) and (19) for how to create signature by Alice); then creation of a signature takes one exponentiation modulo  $n$  and two instances of coin flipping. This amount of work is similar to signing an RSA signature.

In a run of  $Sig\_Verify$ , Alice (resp. Bob) performs four (resp. six) exponentiations. This is the same as Alice (resp. Bob) creating four (resp. six) RSA signatures. These translate to  $4 \log_2 n$  (resp.  $6 \log_2 n$ ) multiplications, or  $O(\log_2 n)^3$  bit operations.

$Sig\_Verify$  is an extremely efficient protocol.

## 5 Zero-knowledge Denial of Non-signature

We now describe our new denial protocol.

Let Alice be alleged to have created the undeniable signature pair  $(S(\hat{M}), \hat{M})$  but in fact this is a non-signature, i.e.,

$$\hat{M} \not\equiv S(\hat{M})^{2^u} \pmod{n}. \quad (27)$$

An arbitrator (Judge) wants Alice to demonstrate this inequality. Figure 4 specifies a protocol for Alice to do so.

**Theorem 4** Let  $w, w(u), S(\hat{M}), \hat{M}, n$  be as specified in the common input in Protocol Denial. The protocol has the following properties:

**Completeness** If Judge does not cause an abortion of *Sig\_Verify* (i.e., by behaving dishonestly), then Judge will always reach a decision, and in the case of  $\hat{M} \not\equiv S(\hat{M})^{2^u} \pmod{n}$ , the decision will always be dismiss;

**Soundness** If  $\hat{M} \equiv S(\hat{M})^{2^u} \pmod{n}$ , then Alice cannot persuade Judge to dismiss the allegation with probability greater than  $\frac{2^{p'}+2^{q'}-1}{2^{p'q'}}$ .

**Zero-knowledge** Judge, even dishonest, gains no information about Alice's private input.

**Proof**

**Completeness** If Judge does not cause Alice to abort the run of *Sig\_Verify*, it is trivial from the inspection of the protocol that he will always reach a decision.

The completeness property of *Sig\_Verify* means that for  $T \equiv (S(\hat{M})^e)^{2^u} \pmod{n}$ , Judge will always accept *Sig\_Verify*( $w, w(u), S(\hat{M})^e, T, n$ ). Then, since for any  $e$ ,  $\hat{M}^e \not\equiv (S(\hat{M})^{2^u})^e \pmod{n}$  implies  $\hat{M} \not\equiv S(\hat{M})^{2^u} \pmod{n}$ , Judge's decision will be dismiss.

**Soundness** Suppose  $\hat{M} \equiv S(\hat{M})^{2^u} \pmod{n}$ . Then for any  $e$  of Alice's choice, it will always hold

$$\hat{M}^e \equiv (S(\hat{M})^{2^u})^e \equiv (S(\hat{M})^e)^{2^u} \pmod{n}.$$

So in order to persuade Judge to reach a dismiss decision, Alice's only strategy is to let Judge accept *Sig\_Verify*( $w, w(u), S(\hat{M})^e, T, n$ ) for some  $T \not\equiv (S(\hat{M})^e)^{2^u} \pmod{n}$ . By the soundness property of *Sig\_Verify*, this can only be possible with the probability not exceeding  $\frac{2^{p'}+2^{q'}-1}{2^{p'q'}}$ .

**Zero-knowledge** Identical to that *Sig\_Verify*. □

## Discussions

- Because of Alice's random choice of  $e$ , the mapping from  $S(\hat{M})$  to  $S(\hat{M})^e$  is a random permutation; therefore Alice cannot be forced to create a (possibly adaptive chosen) pair of signature on any sensible message.
- This idea of zero-knowledge denial of non-signature cannot be applied to Chaum's undeniable signature scheme [3] since that scheme works in a prime field which allows public extraction of the  $e$ -th root of a field element (i.e., the random permutation using a non-secret  $e$  breaks down). Nevertheless, it is evident that our idea can be applied to the RSA-based undeniable signature scheme of Gennaro et al [4].

## 5.1 Performance

Evidently, the performance of this denial protocol is the same as that of *Sig\_Verify*.

## 6 Security Analysis

The soundness properties reasoned in Theorem 3 and Theorem 4 show the binding of a signature to a signer and the free of wrong allegation for a non-signer. In this section we argue two other security properties.

### 6.1 Unforgeability of Signatures

**Theorem 5** *Any algorithm which can create a valid signature modulo  $n$  in polynomial time can factor  $n$  in polynomial time.*

**Proof** Let  $A$  be such an algorithm (maybe probabilistic). On input  $(\hat{M}, n)$ ,  $A$  will output  $S$  in a polynomial time satisfying

$$S^{2^u} \pmod{n} \equiv \hat{M} \pmod{n}.$$

Then run  $A(\hat{M}^2, n)$ , the output satisfies

$$S'^{2^{u-1}} \pmod{n} \equiv \hat{M} \pmod{n}.$$

Clearly,  $S^4 \equiv S'^2 \pmod{n}$ . With  $n$  being a Blum integer, the probability for  $S^2 \equiv S' \pmod{n}$  is only 0.25, and when  $S^2 \not\equiv S' \pmod{n}$  we know that the probability for  $S^2/S' \pmod{n}$  to be a non-trivial square root of 1 is 0.5. Repeating this procedure a small number of times, a non-trivial square root of 1 will be found which suffices for factoring  $n$ .  $\square$

### 6.2 Indistinguishability of Signatures

One known way to decide whether  $S$  is a  $2^u$ -th root of a quadratic residue message  $\hat{M}$ , modulo  $n$ , is to go through  $u$  squarings modulo  $n$  starting from  $S$ . Recall that we have stipulated that  $u > 2^{100}$ , and  $\text{Order}_{2p'q'}(2) > 2^{150}$ . So no cycle can be met within the  $u$  squarings, namely, the no-less-than  $2^{100}$  squarings cannot be shortcut in the repeated squaring method. (Note that  $\text{Order}_n(S)$  should be at least  $p'q'$  or else we will have  $\text{gcd}(S \pm 1, n) > 1$ .) Clearly, this method is intractable (the intractability is even in the sense of resisting massive parallelisation as we have discussed in §1.1).

Considering that factorisation of  $n$  is even more intractable than performing  $u$  squarings (considering  $n > 2^{1024}$ ), we know that any method to fulfill the decision in time less than  $u$  squarings will likely to constitute a grand breakthrough.

## 7 Conclusion

We have devised a new zero-knowledge undeniable signature scheme which, while keeping all virtues of the previous undeniable signature schemes, has two important advantages over them: proven unforgeability equivalent to factorisation and a greatly



improved efficiency for zero-knowledge non-signature denial. An additional advantage is the ease of converting an undeniable signature into a Rabin signature which becomes self-authenticating.

A limitation that our scheme share with the previous RSA-based undeniable signature scheme is the need of using a modulus of a non-standard form. To devise a scheme which can use a standard RSA modulus should be a further work.

## Acknowledgments

I would like to thank Kenny Paterson and Steven Galbraith for their helpful comments on a draft of this paper.

## References

- [1] Camenisch J. and Michels, M. Proving in zero-knowledge that a number is the product of two safe primes, In *Advances in Cryptology — EUROCRYPT 99* (J. Stern ed.), Lecture Notes in Computer Science 1592, Springer-Verlag 1999, pages 106–121.
- [2] Chaum, D. and van Antwerpen, H. Undeniable signatures, *Advances in Cryptology: Proceedings of CRYPTO 89* (G. Brassard, ed.), Lecture Notes in Computer Science 435, Springer-Verlag 1990, pages 212-216.
- [3] Chaum, D. Zero-knowledge undeniable signatures, *Advances in Cryptology: Proceedings of CRYPTO 90* (I.B. Damgaard, ed.) Lecture Notes in Computer Science 473, Springer-Verlag 1991, pages 458-464.
- [4] Gennaro, R., Krawczyk, H. and Rabin, T. RSA-based undeniable signatures, *Advances in Cryptology: Proceedings of CRYPTO 97* (W. Fumy ed.), Lecture Notes in Computer Science 1294, Springer-Verlag 1997, pages 132-149. Also in *Journal of Cryptology* (2000)13:397–416.
- [5] Rabin, M.O. Digitalized signatures and public key functions as intractable as factorization, MIT Laboratory for Computer Science, January, 1979, TR 212.
- [6] Rivest, R.L., Shamir, A. and Wagner, D.A. Time-lock puzzles and timed-release crypto, Manuscript. Available at (<http://theory.lcs.mit.edu/~rivest/RivestShamirWagner-timelock.ps>).