Provably Secure and Efficient Identity-based Signature Scheme Based on Cubic Residues

Zhiwei Wang[1,2], Licheng Wang[1], Shihui Zheng[1], Yixian Yang[1] and Zhengming Hu[1]
(Corresponding author: Zhiwei Wang)

Key Laboratory of Network and Information Attack and Defence Technology of MOE[1]
Beijing University of Posts and Telecommunications, Beijing 100876, P.R. China
College of Computer, Nanjing University of Posts and Telecommunications[2]
(Email: zhwwang@njupt.edu.cn)
(Received Dec. 22, 2010; revised and accepted Mar. 26, 2011)

Abstract

Many identity based signature (IBS) schemes have been proposed so far. However, most of the schemes are based on bilinear pairings. Only a few IBS schemes are without pairings. Up to now, there still remains a challenge to construct a provably secure and efficient IBS scheme without pairings. In this paper, we propose an efficient IBS scheme based on cubic residues, and we prove that our scheme is secure against adaptively chosen message and ID attack.

Keywords: Cubic residues, forking lemma, identity-based signature, provably secure

1 Introduction

Identity(ID)-based cryptography (IBC) is proposed by Shamir [10] in 1984 to simplify key management procedures of traditional certificate-based PKI. In IBC, an entity's public key is derived directly from its identity, such as an e-mail address, or a social security number associated with a user. The private key is computed and issued secretly to the user by a trusted third party called private key generator (PKG). The main advantage of IBC is that it drastically reduces the needs for certificates. The first entire practical and secure identity-based encryption scheme was presented by Boneh and Franklin [1] in 2001. Since then, a rapid development of IBC has taken place. Now, IBC has become a good alternative for certificate-based public cryptosystems, especially when efficient key management is required.

But now, most of the proposed identity-based schemes are based on bilinear pairings, which are usually considered to be involved heavy computation. Very a few identity-based signature (IBS) schemes are without pairings. The first IBS scheme without pairings is proposed by Shamir [10], which is based on RSA problem. In recent years, Lee and Liao proposed an IBS scheme based on discrete logarithm problem [6]. Qiu and Chen presented an IBS scheme based on quadratic residues [9]. However, these IBS schemes did not provide formal security proof. In 2007, Chai et al. [3] proposed a new IBS scheme based on quadratic residues. They proved that their scheme is secure in the random oracle. But Chai et al.'s scheme is not a truly identity-based signature scheme (the analysis can be seen in Note 1 of Section 3).

In this paper, we propose an IBS scheme constructed from cubic residue. If we select proper parameters, the computational efficiency of constructing a cubic residue is better than constructing a quadratic residue. Thus, we choose cubic residue to construct the IBS scheme. Our scheme is very efficient, and is chosen message and ID secure in the random oracle, assuming the hardness of factoring.

The rest of the paper is organized as follows. In Section 2, we give a brief review of some concepts and related lemma. In Section 3, we depict our scheme in detail, and we discuss the scheme's efficiency in Section 4. In Section 5, we provide a formal security proof. Finally, a conclusion is drawn in Section 6.

2 Concepts and Related Lemma

In this section, we introduce some concepts and related lemma [11] about the cubic residues, which will be used in our construction.

As is well known, the multiplicative Z_p^* is a cyclic group of order $p-1$. For an integer $t \geq 1$, let $Z_{p,t}$ be the subgroup of elements in Z_p^* whose order divides t: $Z_{p,t} = \{a \in Z_p^* | a^t = 1\}$. We denote $3_p = \gcd(3, p-1)$.

We consider the cubic map $f_{p,3}$ on Z_p^*. It can easily be seen that the map $f_{p,3} : Z_p^* \to (Z_p^*)^3$ is surjective and for $x \in Z_p^*$, $f_{p,3} = 1$ if and only if $x \in Z_{p,3} \subset Z_p^*$. Moreover, we have $(Z_p^*)^3 = ((Z_p^*)^{3_p})^{3/3_p}$, and the order of $(Z_p^*)^{3_p}$ is $(p-1)/3_p$, in particular, it is prime to $3/3_p$, thus it holds that $((Z_p^*)^{3_p})^{3/3_p} = (Z_p^*)^{3_p}$. Hence, we have $\#Z_{p,3} =$

$(p-1)/\#(Z_p^*)^{3_p} = 3_p$, and from the uniqueness of the subgroup of order 3_p in Z_p^*, letting $\zeta_{p,3}$ be a primitive 3_p-th root of unity, we have $Z_{p,3} = Z_{p,3_p} =< \zeta_{p,3} >$, that is, the subgroup of cubic roots of unity is equal to that of 3_p-th roots of unity. Then due to above, we have the following.

Lemma 1. *For any $x \in Z_p^*$, we have $x^{\frac{p-1}{3_p}} \mod p \in < \zeta_{p,3} >$, and x is a cubic residue if and only if $x^{\frac{p-1}{3_p}} \mod p = 1$.*

Next, we deal with a modulus $N = pq$,where p and q are distinct odd prime numbers. If we choose $p = 2 \mod 3$ and $q = 4 \mod 9$ or $q = 7 \mod 9$, then $3_p = 1, 3_q = 3$. In this case, since any x must be a cubic residue modulus p, x is a cubic residue modulus N if and only if x is a cubic residue modulus q.

So when we construct a cubic residue x modulus q, it must be a cubic residue modulus N. In quadratic residues, when we construct a quadratic residue y modulus N, y should be a quadratic residue both modulus p and modulus q. Hence, if we choose proper p and q, it is easier to construct a cubic residue modulus N than construct a quadratic residue modulus N.

3 ID-based Signature Scheme Based on Cubic Residues

ID-based signature scheme is composed with 4 algorithms, called *Setup*, *Extract*, *Sign* and *Verify*.

Setup. A security parameter is taken as input and returns **PP** (public parameters) and **MK** (master-key) of the trusted third party (PKG). The public parameters **PP** will be publicly known, while the master-key **MK** will be known only to the PKG.

Extract. The output from Setup (**PP**, **MK**) is taken along with an arbitrary ID$\in \{0,1\}^*$ as input, and returns a private key S_{ID}. Here ID is an arbitrary string that will be used as a public key, and S_{ID} is the corresponding private sign key. The Extract phase extracts a private key from the given public key.

Sign. A message M, a private key S_{ID} and **PP** are taken as input. It returns a signature Sig.

Verify. A message M, a signature Sig, ID and **PP** are taken as input. It returns valid or invalid.

Note 1. In ID-based signature scheme, the public key can only be the signer's ID, and can not involve any other messages equivalent to the public keys in the traditional Public Key Infrastructure. Otherwise, the scheme is not a truly ID-based signature scheme. For example, the public key of Chai's scheme [3] not only involves the signer's ID, but also involves c_1, c_2. Tags c_1, c_2 are equivalent to the public keys in the traditional Public Key Infrastructure, which should be certificated. Thus, Chai's scheme is not a truly ID-based signature scheme.

In this section, we depict our ID-based signature scheme in detail.

Setup(k,l): Taking the security parameters (k,t), this algorithm will be carried out by PKG as follows:

1) Generate randomly two same length distinct prime numbers p and q, such that $p \equiv 2 \mod 3$, $q \equiv 4 \mod 9$ or $7 \mod 9$, satisfying $pq < 2^k$, then compute $N = pq$.
2) Select a non-cubic residue a modulus q.
3) Compute $\eta = [(q-1) \mod 9]/3$, and $\lambda = \eta \mod 2+1$.
4) Compute $\beta = \frac{q-1}{3}$, and $\xi = a^{\eta \cdot \beta} \mod q$.
5) Select $h_1() : \{0,1\}^* \to Z_N^*$, $h_2() : \{0,1\}^* \to \{0,1\}^t$ as two hash functions.

The master key of PKG is set to be $MK = (p, q, \beta)$, and the public parameters of PKG are $PP = (N, h_1(), h_2(), a, \eta, \lambda)$.

Extract(ID,MK,PP): Given ID, PKG computes the corresponding private key S_{ID} as follows:

1) Compute $\omega = h_1(ID)^{\lambda \cdot \beta} \mod q$.
2) Compute

$$c = \begin{cases} 0, & \omega = 1 \\ 1, & \omega = \xi \\ 2, & \omega = \xi^2 \end{cases}$$

and compute $H(ID) = a^c \cdot h_1(ID) \mod N$.

Note 2. $H(ID)$ is a cubic residue modulus N.

The proof of this note is as follows. In our scheme, since $q \equiv 4 \mod 9) or 7 \mod 9)$, $3_q = 3$. According to Lemma 1, we can compute the value of $H(ID)^{\frac{q-1}{3_q}} \equiv [a^c \cdot h_1(ID)]^{\frac{q-1}{3}} \mod q$.

In case of $q = 4 \mod 9$, $\eta = 2$, $\lambda = 1$, $\xi = a^{2\beta} \mod q$. If $c = 0$, then $\omega = h_1(ID)^{\beta} = h_1(ID)^{(\frac{q-1}{3})} = 1 \mod q$. So $H(ID)^{\frac{q-1}{3_q}} \equiv 1 \mod q$. If $c = 1$, then $\omega = h_1(ID)^{\beta} = \xi$. Since $a^{2\beta} = \xi$, $a^{\beta} = \xi^2$, $H(ID)^{\frac{q-1}{3_q}} \equiv [a \cdot h_1(ID)]^{\frac{q-1}{3}} \equiv a^{\frac{q-1}{3}} \cdot h_1(ID)^{\frac{q-1}{3}} = \xi^3 = 1 \mod q$. If $c = 2$, then $\omega = h_1(ID)^{\beta} = \xi^2$. Since $a^{2\beta} = \xi$, $H(ID)^{\frac{q-1}{3_q}} \equiv [a^2 \cdot h_1(ID)]^{\frac{q-1}{3}} \equiv \xi^3 = 1 \mod q$.

In case of $q = 7 \mod 9$, $\eta = 1$, $\lambda = 2$, $\xi = a^{\beta} \mod q$. If $c = 0$, then $\omega = h_1(ID)^{2\beta} = h_1(ID)^{2(\frac{q-1}{3})} = 1 \mod q$. So $H(ID)^{\frac{q-1}{3_q}} \equiv 1 \mod q$. If $c = 1$, then $\omega = h_1(ID)^{2\beta} = \xi$, and then $h_1(ID)^{\beta} = \xi^2$. Since $a^{\beta} = \xi$, $H(ID)^{\frac{q-1}{3_q}} \equiv [a \cdot h_1(ID)]^{\frac{q-1}{3}} \equiv a^{\frac{q-1}{3}} \cdot h_1(ID)^{\frac{q-1}{3}} = \xi^3 = 1$

mod q. If $c = 2$, then $\omega = h_1(ID)^{2\beta} = \xi^2$, and then $h_1(ID)^{\beta} = \xi$. Since $a^{\beta} = \xi$ and $a^{2\beta} = \xi^2$, $H(ID)^{\frac{q-1}{3q}} \equiv [a^2 \cdot h_1(ID)]^{\frac{q-1}{3}} \equiv \xi^3 = 1 \mod q$. Thus $H(ID)$ must be a cubic residue modulus q.

Since $p \equiv 2 \mod 3$, $3_p = 1$, $H(ID)^{\frac{p-1}{3_p}} \equiv [a^c \cdot h_1(ID)]^{p-1} mod p = 1 \mod p$. So $H(ID)$ must be a cubic residue modulus p, then $H(ID)$ also must be a cubic residue modulus N.

3). compute S_{ID} as cubic root of $H(ID)^{-1}$.

$$S_{ID} = [H(ID)]^{\frac{2^{\eta-1}(p-1)(q-1)-3}{9}} \mod N.$$

Note 3. In case of $q \equiv 4 \mod 9$, $\eta = 1$, $(p-1)(q-1) \mod 9 = 3$. In case of $q \equiv 7 \mod 9$, $\eta = 2$, $2(p-1)(q-1) \mod 9 = 3$. Thus $\frac{2^{\eta-1}(p-1)(q-1)-3}{9}$ must be an integer.

The validity of step 3 is in the fact that $H(ID)^{(p-1)(q-1)} \equiv 1 \mod N$ by Euler theorem. Consequently, $\frac{2^{\eta-1}(p-1)(q-1)-3}{9} \equiv [H(ID)^{-1}]^{\frac{1}{3}} \mod N$.

Note 4. $S_{ID}^3 \cdot H(ID) \equiv 1 \mod N$.

Note 5. Our *Extract* algorithm also provide an efficient way to compute a cubic root, given special parameters.

At last, PKG returns secretly S_{ID} to the user with ID.

Sign(M,S_{ID},PP): To sign a message M, a user do as follows:

1) Random select $r \in Z_N^*$, compute $R = r^3 \mod N$.
2) Compute $Z = r \cdot S_{ID}^{h_2(R,M)} \mod N$.

The return signature is $Sig = (Z, R)$.

Verify(PP,Sig,ID): Given a signature $Sig = (Z, R)$ on a message M, a verifier should verify the signature only by the signer's ID:

1) Compute $H_1(ID) = h_1(ID) \mod N$, $H_2(ID) = a \cdot h_1(ID) \mod N$ and $H_3(ID) = a^2 \cdot h_1(ID) \mod N$.
2) Check wether $Z^3 \cdot H_i^{h_2(R,M)}(ID) = R$ ($i \in \{1,2,3\}$) holds. If (one of $i \in \{1,2,3\}$) holds, output "valid", otherwise, output "invalid".

Note 6. Since $Z^3 \cdot H^{h_2(R,M)}(ID) = R \cdot (S(ID)^3 \cdot H(ID))^{h_2(R,M)} = R \mod N$, the validity of our scheme could be check by Step 2.

Note 7. There is only 1 modular exponentiation in the Sign algorithm, so our scheme is very efficient.

4 Security Proof

In this section, we prove that the security of our scheme is based on the hardness of factoring by using the Forking Lemma introduced by Pointcheval and Stern [7, 8].

4.1 Factoring Composite by Taking Cubic Roots

For simplicity, let $N = pq$, where p, q are large primes. Let a be a cubic residue over N, and s_1 and s_2 be its two cubic roots, satisfying $s_1 \neq s_2 mod N$. Then, N could be factored by executing **Fac**(a, s_1, s_2), which is depicted as follows (confirming $s_1^3 \equiv s_2^3 \equiv a \mod N$).

Fac(a, s_1, s_2): if $s_1 \equiv s_2 mod N$, then outputs "failure", otherwise output $GCD(s_1 - s_2, N)$ as the non-trivial divisor of N.

The function is justified by the followed facts. Since $x^3 \equiv y^3 \equiv a \mod N$,we have $(x-y)(x^2+xy+y^2) \equiv 0 \mod N$, therefore $(x-y)(x^2+xy+y^2) = kpq$ for some integer k. If $x \neq y \mod N$, and $x - y$ is not multiple of N, then $x - y$ contain non-trivial divisor of N.

Indeed, all the root pairs (s_1, s_2) satisfying $s_1 \neq s_2 \mod N$ can lead to the factorization of N.

The security notion for signature schemes is the standard notion of unforgeability under chosen-message attack (UF-CMA) [5]. An appropriate extension of it for ID-based signature schemes proposed by [2, 4], which is defined as unforgeability under chosen message and ID attack. Unforgeability under chosen message and ID attack is defined as the following game between a challenger $\mathcal{B}$ and an adversary $\mathcal{A}$.

Setup: The challenger $\mathcal{B}$ takes a security parameter k and runs the *Setup* algorithm of the IBS scheme. It gives the adversary $\mathcal{A}$ the resulting public parameters **PP**. It keeps the master-key **MK** to itself.

Queries: The adversary $\mathcal{A}$ adaptively makes a number of different queries to the challenger $\mathcal{B}$. Each query can be one of the following.

- Extract Queries: The challenger $\mathcal{B}$ responds by running *Extract* algorithm to generate the private key S_{ID} corresponding to the public key ID issued by $\mathcal{A}$. It sends S_{ID} to the adversary $\mathcal{A}$.
- Signature Queries: The adversary $\mathcal{A}$ can ask for the signature of any identity ID on any message M. The challenger $\mathcal{B}$ responds by first running *Extract* algorithm to obtain the private key S_{ID} of ID, and then running *Sign* algorithm to obtain a signature, which is forward to the adversary.

Forgery: The adversary outputs a message M^*, an identity ID^* and a string Sig^*. The adversary succeeds if the following conditions are satisfied:

1) Verify(PP,ID^*,M^*,Sig^*)=valid.

2) The adversary has not made an extract query on ID^*.

3) The adversary has not made a sign query on (ID^*, M^*).

The advantage of an adversary $\mathcal{A}$ in the above game is defined to be

$$Adv_{\mathcal{A}} = Pr[\mathcal{A} \ succeeds]$$

where the probability is taken over all coin tosses made by the challenger and the adversary. If the advantage of $\mathcal{A}$ is negligible, then the ID-based signature scheme is unforgeability under chosen message and ID attack.

4.2 Provably Secure Proof

The security notion for signature schemes is the standard notion of unforgeability under chosen-message attack (UF-CMA) [5]. An appropriate extension of it for ID-based signature schemes proposed by [2, 4], which is defined as unforgeability under chosen message and ID attack. In this section, we will prove our scheme is secure based on the hardness of factoring problem, which should use the Forking Lemma [7, 8] twice.

Theorem 1. *If the factoring problem is (t', ϵ')-hard, then our scheme is $(t, q_{h_2}, q_{sig}, \epsilon)$-secure against existential forgery on the adaptively chosen message and ID attack, which satisfying:*

$$\begin{aligned} \epsilon' &\geq \frac{6 \cdot (\epsilon - q_{sig}(q_{h_2} + q_{sig}) \cdot 2^{-k})^2}{\pi^2 (q_{h_2} + 1)} \\ &\quad - \frac{6 \cdot 2^{-l} \cdot (\epsilon - q_{sig}(q_{h_2} + q_{sig}) \cdot 2^{-k})}{\pi^2}, \\ t' &= 3t + \mathcal{O}(k^2 l + k^3), \end{aligned}$$

where l and k are security parameters.

Proof. Suppose our scheme $IBS - W$ is not secure, namely, there exists an adversary $\mathcal{A}$ who can attack $IBS - W$ with non negligible advantage, then we will show that it is possible to build an algorithm $\mathcal{B}$ to solve factoring problem with non negligible probability. That is, a contradiction against the assumption in the theorem, so $IBS - W$ is secure.

Now, we will show how to build an algorithm $\mathcal{B}$ that on input of a given instance of factoring problem $N = pq$ $(pq > 2^k, k \geq 1024 bits)$ for some unknown p and q, outputs p or q with non negligible probability.

$\mathcal{B}$ will simulate the interaction game with $\mathcal{A}$ as follows:

1. $\mathcal{B}$ chooses a non-cubic residue a modulus N, and choose a secure parameter $t \geq 160$, and sends (N, a) to $\mathcal{A}$ as public parameters. $\mathcal{B}$ also maintain three lists: a signature list and two hash query lists.

2. Then $\mathcal{B}$ responds to $\mathcal{A}$'s queries as follows:

• h_1-queries: To respond to $\mathcal{A}$'s h_1-queries, $\mathcal{B}$ maintains a list of tuples $< ID, h_1, s, c >$, where ID is the requested identity, h_1 is $\mathcal{B}$'s answer, c, s is the internal parameter as explained below.

On a query on ID, $\mathcal{B}$ will return h_1 as the answer if ID already exists on the h_1-list in the tuple $< ID, h_1, s, c >$, otherwise, $\mathcal{B}$ will choose a random number $s \in Z_N^*$ and the tag $c \in \{0, 1, 2\}$, and return $h_1 = \frac{s^3}{a^c} \mod N$ as the answer, then adds the entry $< ID, h_1, s, c >$ to the h_1-list.

• Extraction queries: Upon an extraction query on ID, $\mathcal{B}$ will return $S_{ID} = s$, as well as c, as the answer if ID already exists on the h_1-list in the tuple $< ID, h_1, s, c >$, otherwise, $\mathcal{B}$ will add a new entry contain ID the same way as handling h_1-query, then return s as the answer, and record c.

• Signature queries: Signature query on message M with ID could be almost answered at random. Since $\mathcal{B}$ controls h_1-list, $\mathcal{B}$ first compute $H_{ID} = a^c h_1(ID)$, and picks random Z and σ, if $(R' = Z^3 \cdot H_{ID}^{\sigma}, M)$ is in the h_2-list, $\mathcal{B}$ reports failure, otherwise, $\mathcal{B}$ returns (Z, R') as the signature on ID, and add (R', M, σ) to the h_2-list.

3. $\mathcal{A}$ outputs a forged signature (Z^*, R^*) on some message M^* and ID^*, which has not been queried on to the signing oracle with M^*. It should be also pointed out that ID^* should not be queried on to extraction query oracle.

In order to factor N, we need to apply the Forking Lemma twice, which means that $\mathcal{B}$ should reset $\mathcal{A}$ thrice with the same random tape. Since $\mathcal{B}$ has recorded the transcript in the first run, he can give the exact same answers to all $\mathcal{A}$'s queries before a h_2 query is asked in the next two runs. In the first run, as soon as $\mathcal{A}$ asks for this h_2 query, $\mathcal{B}$ gives σ^* as the answer. In the second run, when $\mathcal{A}$ asks for this h_2 query, $\mathcal{B}$ choose a new number $\sigma^{'}$ as the answer. In the third run, $\mathcal{B}$ once again re-selects a new number $\sigma^{''}$, which is different from σ^* and $\sigma^{'}$, as the answer for $\mathcal{A}$'s h_2 query.

After three rounds, with the same random tape, $\mathcal{A}$ gives thee forged signatures $(Z^*, R^*), (Z', R^*), (Z'', R^*)$ on the same message M^* and ID^*. Then $\mathcal{B}$ searches three h_2-lists for (R^*, M^*) to acquire $\sigma^*, \sigma', \sigma''$.

If $\sigma^*, \sigma', \sigma''$ satisfying $GCD(\sigma^* - \sigma', \sigma' - \sigma'') = 1$, $\mathcal{B}$ can get $a_1(\sigma^* - \sigma') + a_2(\sigma' - \sigma'') = 1$, for some special a_1 and a_2. Due to the same random tape, $\mathcal{B}$ can obtain

$$Z^{*3} \cdot H^{\sigma^*}(ID^*) = Z^{'3} \cdot H^{\sigma'}(ID^*) \Rightarrow (\frac{Z'}{Z^*})^3 = H^{\sigma^* - \sigma'}(ID^*)$$

and

$$Z^{'3} \cdot H^{\sigma'}(ID^*) = Z^{''3} \cdot H^{\sigma''}(ID^*) \Rightarrow (\frac{Z''}{Z'})^3 = H^{\sigma' - \sigma''}(ID^*).$$

From above, $\mathcal{B}$ can deduce

$$H(ID^*) = [(\frac{Z'}{Z^*})^{a_1} \cdot (\frac{Z''}{Z'})^{a_2}]^3 \mod N.$$

Then, $\mathcal{B}$ can get a cubic root of $H(ID^*)$ easily from the above equation. Finally, $\mathcal{B}$ search in the h_1-list to get anther cubic root of $H(ID^*)$. Then $\mathcal{B}$ can use these two different roots to factor N, which is introduced in Section 4.1.

Since $\mathcal{B}$ has to run $\mathcal{A}$ thrice, and takes some other operations to factor N, such as exponentiations and GCD operations, the time $\mathcal{B}$ used to factor N can be denoted as $t' = 3t + \mathcal{O}(k^2 l + k^3)$.

ϵ' denotes the probability of factoring N, ϵ denotes the probability of $\mathcal{A}$ forging a signature in the real attack, and ϵ^* denotes the probability of $\mathcal{A}$ forging a signature in a single run in simulation.

In the face of $\mathcal{A}$'s signature querying, $\mathcal{B}$ chooses Z at random, which may cause a collision against a value in the h_2-list. Since the h_2-list is filled according to both h_2 and signature queries, the probability of collision is $(q_{h_2} + q_{sig})/|Z_N^*|$. Thus in the simulation, the probability of $\mathcal{A}$ forging a signature in a single run is derived from:

$$\epsilon^* = \epsilon - (q_{h_2} + q_{sig})/|Z_N^*| \geq \epsilon - (q_{h_2} + q_{sig})2^{-k}.$$

ϵ_i denotes the probability of forgery based on i-th h_2-query in a single run. we can easily get:

$$\epsilon^* = \sum_{i=1}^{q_{h_2}+1} \epsilon_i.$$

Given specific string m of length n, which determines the random tape of $\mathcal{A}$, let $\epsilon_{i,m}$ denote the the probability of forgery based on i-th h_2-query in a single run.. Hence, we calculate:

$$2^n \epsilon_i = \sum_{m \in \{0,1\}^n} \epsilon_{i,m}.$$

For a specific string m, the achievement of forgery in three runs should satisfy two conditions:

1) the three answers $(\sigma^*, \sigma', \sigma'')$ of hth h_2 query in the three runs should be different.

2) $GCD(\sigma^* - \sigma', \sigma' - \sigma'') = 1$.

So the probability of achievement of forgery in three runs can be estimated as

$$\frac{6}{\pi^2} \cdot \epsilon_{i,m}(\epsilon_{i,m} - 2^{-l}).$$

Given two arbitrary number m and n, solving the probability that m and n are relatively prime can be seen in Appendix A. Let P_i be the probability that a forgery was based on the ith h_2-query in three runs. Then we can deduce:

$$\begin{aligned} P_i &= \sum_{m \in \{0,1\}^n} 2^{-n} \frac{6}{\pi^2} \cdot \epsilon_{i,m}(\epsilon_{i,m} - 2^{-l}) \\ &= 2^{-n} \frac{6}{\pi^2} \Big(\sum_{m \in \{0,1\}^n} \epsilon_{i,m}^2 - 2^{-l} \sum_{m \in \{0,1\}^n} \epsilon_{i,m} \Big) \\ &\geq \frac{6 \cdot 2^{-n} (\epsilon_i \cdot 2^n)^2}{\pi^2 \cdot (2^n)} - \frac{6 \cdot 2^{-n} \cdot 2^{-l} \cdot \epsilon_i \cdot 2^n}{\pi^2} \\ &= \frac{6}{\pi^2} \cdot \epsilon_i^2 - \frac{6 \cdot 2^{-l}}{\pi^2} \cdot \epsilon_i. \end{aligned}$$

Thus, the probability of $\mathcal{A}$ breaking $IBS - W$ scheme in real attack can be calculated as:

$$\begin{aligned} \epsilon' &= \sum_{i=1}^{q_{h_2}+1} P_i \geq \sum_{i=1}^{q_{h_2}+1} \frac{6}{\pi^2} \cdot \epsilon_i^2 - \sum_{i=1}^{q_{h_2}+1} \frac{6}{\pi^2} \cdot 2^{-l} \epsilon_i \\ &\geq \frac{6\epsilon^{*2}}{\pi^2 \cdot (q_{h_2} + 1)} - \frac{6 \cdot 2^{-l} \cdot \epsilon^*}{\pi^2} \\ &\geq \frac{6 \cdot (\epsilon - q_{sig}(q_{h_2} + q_{sig}) \cdot 2^{-k})^2}{\pi^2 (q_{h_2} + 1)} \\ &\quad - \frac{6 \cdot 2^{-l} \cdot (\epsilon - q_{sig}(q_{h_2} + q_{sig}) \cdot 2^{-k})}{\pi^2}. \end{aligned}$$

Then, we complete our security proof. □

5 Conclusion

In this paper, we first discuss the concepts and related lemma in cubic residues. Then, we construct an efficient Identity-based signature scheme from cubic residues. Finally, we prove that our scheme is secure in the random oracle, by using the Forking Lemma twice.

Acknowledgments

This study is partially supported by National Basic Research Program of China(No. 2007CB310704), National Natural Science Foundation of China (No. 61003285 and No. 60821001), National 863 (No. 2009AA01Z439), and ...

References

[1] D. Boneh, M. Franklin, "Identity-based encryption from Weil pairing," *Advance in Cryptology-CRYPTO 2001*, LNCS 2193, Springer-Verlag, pp. 213-229, 2001.

[2] C. C. Cha, J. H. Cheon, "An identity-based signature from gap diffie-hellman groups," *Proceedings of PKC'03*, LNCS 2567, Springer-Verlag, pp. 18-30, 2003.

[3] Z. C. Chai, Z. F. Cao, X. L. Dong, "Identity-based signature scheme based on quaratic residues," *Science in China Series F: Information Sciences*, vol. 50, no. 3, pp. 373-380, 2007.

[4] Y. Dodis, J. Katz, S. Xu, M. Yung, "Strong key-isulated signature schemes," *Proceedings of PKC'03*, LNCS 2567, Springer-Verlag, pp. 130-144, 2003.

[5] S. Goldwasser, S. Micali, R. Rivest, "A digital signature scheme secure against adaptive chosen-message attacks," *SIAM Journal of Computing*, vol. 17, no. 2, pp. 281-308, 1988.

[6] W. B. Lee, K. C. Liao, "Constructing identity-based cryptosystems for discrete logarithm based cryptosystems," *Journal of Network and Computer Applications*, vol. 27, pp. 191-199, 2004.

[7] D. Pointcheval, J. Stern, "Security proofs for signature schemes," *Proceedings of Eurocrypt'96*, LNCS 1070, Springer-Verlag, pp. 387-398, 1996.
[8] D. Pointcheval, J. Stern, "Security arguments for digital sigantures and blind signatures," *Journal of Cryptography*, vol. 13, pp. 361-396, 2000.
[9] W. D. Qiu, K. F. Chen, "Identity oriented based on quadratic residues," *Applied Mathematics and Computation*, vol. 168, pp. 235-242, 2005.
[10] A. Shamir, "Identity based cryptosystems and signature schemes," *Advance in Cryptology-Crypto'84*, LNCS 196, Springer-Verlag, pp. 47-53, 1984.
[11] Z. H. Sun, "On the theory of cubic residues and nonresidues," *Acta Arithmetic*, vol. 34, no. 4, pp. 291-335, 1998.

Appendix A

We show that solving the Probability that Two Arbitrary Integers m and n are Relatively Prime in this Appendix.

Let m, n be the two arbitrary integers. Let p_i denote all the prime numbers $p_1 = 2$, $p_2 = 3$, $p_3 = 5$ etc. The probability that a number has a factor p_i is $1/p_i$. Thus, the probability that m and n do not both contain p_i at the same time is estimated as $1 - 1/(p_i)^2$.

Therefore, the probability that m and n are relatively prime, or do not both contain any of the p_i at the same time is product $((1 - 1/(p_i)^2), i = 1, ..., infinite)$. It is easy to prove that the reciprocal of this product equal to $\sum 1/k^2$, $(k = 1, ..., infinite)$, which is equal to $\pi^2/6$, thus, the probability equal to $6/\pi^2$.

Zhiwei Wang is a lecturer at Nanjing University of Posts and Telecommunications. He received his Ph.D degree from Beijing University of Posts and Telecommunications. His research interests include: public key cryptography, cryptographic protocol, and information security etc.. He has published 20 scientific papers.

Licheng Wang is a lecturer at Beijing University of Posts and Telecommunications. He received her Ph.D degree from Shanghai Jiaotong University. His research interests include: braidgroup cryptography, lattice based cryptography etc.. He has published 20 scientific papers.

Shihui Zheng is a lecturer at Beijing University of Posts and Telecommunications. She received her Ph.D degree from Shandong University. Her research interests include: public key cryptography, hash function etc.. She has published 5 scientific papers.

Yixian Yang received his M.S. and Ph.D degrees from Beijing University of Posts and Telecommunications. Now, he is a professor at Beijing University of Posts and Telecommunications. His research interests lie in coding theory, cryptography etc.. He has published 300 scientific papers.

Zhengming Hu is a professor at Beijing University of Posts and Telecommunications. His research interests lie in coding theory, cryptography etc.. He has published 50 scientific papers.