Efficient "on the fly" signature schemes based on integer factoring

Takeshi Okamoto, Mitsuru Tada, and Atsuko Miyaji

School of Information Science,
Japan Advanced Institute of Science and Technology (JAIST),
Asahidai 1-1, Tatsunokuchi, Nomi, Ishikawa, 923-1292, Japan
{kenchan, mt, miyaji}@jaist.ac.jp

Abstract. In 1999, Poupard and Stern proposed *on the fly signature scheme* (PS-scheme), which aims at minimizing the on-line computational work for a signer. In this paper, we propose more efficient on the fly signature schemes by improving the PS-scheme. In PS-scheme, the size of secret-key is fixed by modulus n, so that this feature leads to some drawbacks in terms of both the computational work and the communication load. The main idea of our schemes is to reduce the size of secret-key in PS-scheme by using a public element g which has a specific structure. Consequently, our schemes are improved with respect to the computational work (which means the computational cost for "pre-computation", "(on-line) signature generation" and "verification") and the data size such as a secret-key and a signature.

1 Introduction

As well-known, a *signature scheme* is an important tool for secure communication in an open network. Furthermore, a public-key infrastructure (PKI) actually requires compact signature schemes. *Compactness* on both computational work and data size, gives users' convenience, and is acceptable for various application to capacity limited devices such as a smart card.

Focus on the computational work in *a generic digital signature scheme*[1]. In such a signature scheme, there are two kinds of computation to generate a signature, that is, it consists of *pre-computation* and *(actual) signature generation*. To estimate the efficiency of a signature scheme, we should separately consider the computational cost for pre-computation and that for signature generation. The information generated at the pre-computation does not depend upon the message to be signed. Therefore the pre-computation can be executed in off-line, i.e. before a message to be signed is given. This means that such a computational cost does not influence the processing time after a message is given.

On the other hand, the computational cost in the signature generation step, does directly influence the processing time after being given a message. With

[1] As well as in [PS00], in this paper, *a generic (digital) signature scheme* means a signature scheme which can be derived from a three-pass identification scheme by using an appropriate hash function.

respect to a fast signature generation, Naccache et al. [NMVR94] proposed the efficient technique: a trusted authority computes the information in off-line, and treats those as *coupons*. In coupon based signature, the reduction of computation work in on line is the very target for fast signature. Consequently, we can say that it is a worthwhile work to make the computational cost small in signature scheme.

In 1992, Girault [Gir92] modified Schnorr's signature scheme [Sch91] in which an RSA-modulus[2] is used instead of a prime modulus. This modification leads to no modulo reduction in the signature generation. Therefore, in Girault's scheme, faster processing of the signature generation is possible than in Schnorr's one. In 1998, Poupard and Stern [PS98] investigated and gave provable security for Girault's scheme, and named that scheme GPS-scheme. In this paper, we call a generic signature scheme in which modulo reduction is not necessary at the (on-line) signature generation step, *on the fly signature scheme*.

In 1999, Poupard and Stern [PS99] proposed a generic signature scheme (PS-scheme), whose security relies on the difficulty of integer factoring. In this scheme, the size of the public-key is smaller than that in GPS-scheme. Consequently, compared with GPS-scheme, the computational cost and the data size can be decreased, and PS-scheme is seemed more secure under the *one-key attack* scenario [PS99]. However, PS-scheme has some drawbacks. For instance, the size of secret key is only dependent on modulus n, and considerably large (about $|n|/2$). This drawback leads to inefficient results in both communication work and data size. Moreover, computational cost in the verification is very high.

In this paper, we improve PS-scheme and propose new "on the fly" signature schemes (Scheme I and II) which is based on integer factoring. In our schemes, a public-key g has a specific structure. Consequently, in comparison with PS-scheme, the size of secret-key is small ($\ll |n|/2$). In the following, our schemes realize a compactness of signature. Especially, the computation work in verification are much reduced by the changing n in $x = g^{y-ne} \bmod n$ (PS-scheme) into z in $x = g^{y-ze} \bmod n$ (our schemes).

As for Scheme I, a public-key n is RSA modulus, which is the same as that in PS-scheme. The performance in Scheme I is much superior to that in PS-scheme and the security is as secure as integer factoring problem for modulus n (in the random oracle model). To satisfy the security, Scheme I uses *asymmetric basis* g in $\mathbb{Z}_n^*$ which is a variant of [Po00],

As for Scheme II, a public-key n consists of three or more primes instead of RSA modulus in Scheme I (or PS-scheme). In [Sil99], we can see several trials to get faster computation for RSA cryptosystem [RSA78] by the technique of increasing the numbers of the factors of the modulus. Scheme II can make use of the very technique. The security is as secure as specially defined mathematical problem *finding order problem* (in the random oracle model), which is derived from integer factoring .

[2] In this paper, we call a modulus to be a product of two distinct primes *an RSA-modulus*.

Concrete to say, compared with PS-scheme, the size of a secret-key in Scheme I (resp. Scheme II) and a signature can be reduced by at least 69% and 47% (resp. 63% and 43%), respectively. Furthermore, Scheme I (resp. Scheme II) has an advantage that the computational cost can also be smaller. Compared with PS-scheme, the computational cost in Scheme I (resp. Scheme II) for pre-computation, signature generation and verification can be reduced by at least 38%, 69%, and 64% (resp. 54%, 63%, and 61%), respectively.

This paper is organized as follows. In Section 2, we will review PS-scheme and will discuss it. In Section 3, we will introduce our proposed signature scheme (Scheme I), will describe some features of ours, and will give provable security for ours. In Section 4, we will introduce an optimized scheme (Scheme II) and discuss in the same way as Section 3. In Section 5, we will discuss the security consideration with respect to (1)the size of n and (2)the number of prime factors with n in our schemes. In Section 6, we will evaluate the performance of our schemes by comparing with those of several existing schemes. The conclusion will be given in Section 7.

2 Previous Scheme

In this section, we review the signature scheme (PS-scheme) in [PS99]. This scheme is a generic signature scheme which is derived from the identification scheme. We first introduce some notation. The symbol $\varphi(\cdot)$ denotes Euler totient function, that is, $\varphi(n)$ is the number of the natural numbers less than n and coprime to n. The symbol $\lambda(\cdot)$ denotes so-called Carmichael function, that is, $\lambda(n)$ is the greatest number among the possible orders of elements in $\mathbb{Z}_n^*$. The order of an element $g \in \mathbb{Z}_n^*$ is represented as $\mathrm{Ord}_n(g)$.

2.1 Protocols

In PS-scheme, the following parameters exist: k and κ are *the security parameter* and *the information leak parameter*, respectively. The security parameter k is $|n|/2$, and the information leak parameter κ is assumed so that 2^κ-time computation is intractable. The parameters A and B satisfy $A < n$ and $|A| = \kappa + k + |B|$. Also B is assumed that B-time computation is intractable. We use an appropriate hash function $\mathcal{H} : \{0,1\}^* \to \{0,1\}^{|B|}$

Key generation step: The signer picks up two same-size primes p and q, and computes $n = pq$. After that, she picks up $g \in \mathbb{Z}_n^*$ satisfying $\mathrm{Ord}_n(g) \in \{\lambda(n), \lambda(n)/2\}$ and computes $s = n - \varphi(n)$ $(= p + q - 1)$. The secret-key is defined by s. The corresponding public-key is (n, g).

Signature generation step: Imagine that the signer generates a signature for a message $m \in \{0,1\}^*$. The signer picks up a random number $r \in \mathbb{Z}_A$ to compute $x = g^r \mod n$, $e = \mathcal{H}(x, m)$ and $y = r + se$. Note that y is the very value of $r + se$ on $\mathbb{Z}$. The signature for a message m is (e, y).

Verification step: Given the public-key of the signer (n, g), a message m and a signature (e, y), the verifier accepts the signature, if both $y < A$ and $e = \mathcal{H}(g^{y-ne} \bmod n, m)$ hold, and rejects it, otherwise.

2.2 Features and Drawbacks

A secret-key in PS-scheme is $s = n - \varphi(n)$ which depends only upon (a part of) the public-key n. The two parameters n and s are congruent under the modulo $\varphi(n)$, and the size of s is about a half of that of n.

Moreover, the computation of y is executed on $\mathbb{Z}$, and the information on a secret-key is protected by computing $r + se$ with condition $r \gg se$. Therefore, we can see that the size of r also depend upon that of se.

In the verification step, the size of y has to be explicitly verified whether the condition $y < A$ holds or not. This kind of verification cannot be seen in the existing signature schemes [ElG85,NIST91,RSA78,Sch91], hence we can say that such a verification indeed characterizes PS-scheme.

Unfortunately, PS-scheme has the following drawbacks.

High computational cost for verifier: In the verification step, $y \ll ne$ holds actually. And the order of $g \in \mathbb{Z}_n^*$ is not open. Therefore, the computational cost for a verifier is considerably large as $|ne|$ increases. The verifier must compute full exponentiation ($|y-ne|$ bits) calculus such as $x = g^{y-ne} \bmod n$.

Inefficiency by the increase of a secret-key size: If the size of a secret-key s increase for the security reason, then this scheme shall get inefficient in view of (1) the computational cost for pre-computation, signature generation and verification, and (2) data size such as the size of signature.

Restriction for the structure of a public-key n: When we set up a public-key n to be the product of three or more primes, the size of a secret-key shall accordingly increase. For example, in case n is the product of three primes, that is, p, q and r, the secret-key s $(= n - \varphi(n))$ turns out to be $n - (p-1)(q-1)(r-1)$ $(= pq + qr + rp - (p+q+r) + 1)$, whose size is about 3/2 times of that in case n is the product of two primes.

3 Proposed Scheme

In this section, we introduce our signature scheme (Scheme I). The main idea of Scheme I is to reduce the size of secret-key by using element g which has a specific structure. Furthermore, we wish to construct that Scheme I has the same security of PS-scheme, so that the following basis is existed in Scheme I.

Definition 1 (Asymmetric basis) Let n be an RSA modulus such that $n = pq$. Then we say that g is an asymmetric basis in $\mathbb{Z}_n^*$ if the multiplicity of 2 in $\mathrm{Ord}_p(g)$ is not equal to the multiplicity of 2 in $\mathrm{Ord}_q(g)$. ■

We can say that this definition is more relaxed in comparison with that of [Po00].

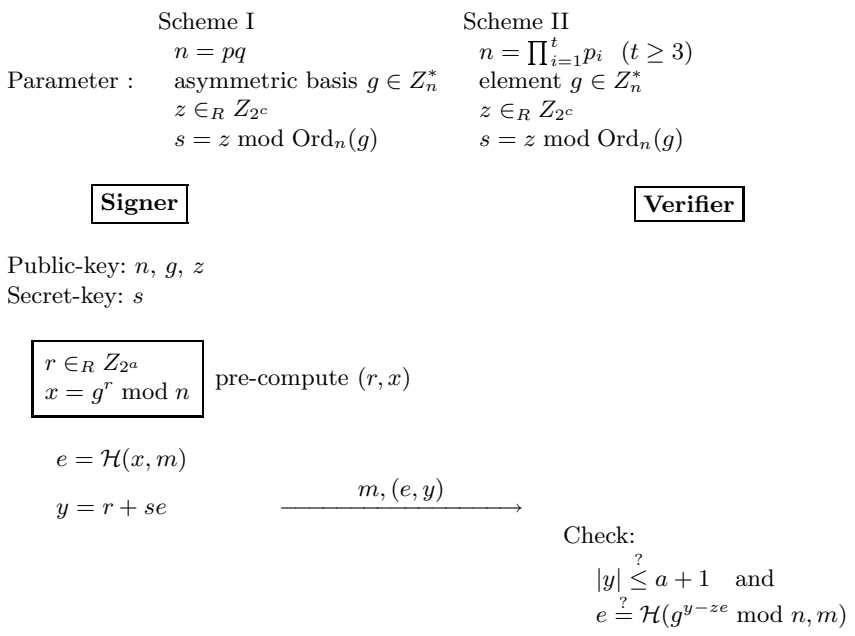

Fig. 1. Proposed signature schemes

3.1 Protocols

Scheme I has the parameters k, κ, a, b and c, where k is the security parameter, that is, the length of the secret-key, and κ is the information leak parameter, that is, 2^κ-time computation shall be intractable. Those parameters are assumed to satisfying $a \geq b + k + \kappa$ and $c \geq k + 2\kappa$. The detailed conditions on the parameters are mentioned in Section 3.2. We use an appropriate hash function $\mathcal{H} : \{0,1\}^* \to \{0,1\}^b$.

Key generation step: The signer picks up two same-size primes p and q, and computes $n = pq$. After that, she chooses an element $g \in \mathbb{Z}_n^*$ which is an asymmetric basis in Z_n^*. She picks up a random number $z \in \mathbb{Z}_{2^c}$ and computes $s = z \bmod q$, where $\mathrm{Ord}_n(g) = q$. The secret-key is s and the corresponding public-key is (n, g).

Signature generation step: Imagine that a signer having a public-key (n, g, z) and the corresponding secret-key s, generates a signature for a message $m \in \{0,1\}^*$. Then she picks up a random number $r \in \mathbb{Z}_{2^a}$ to compute $x = g^r \bmod n$ and $e = \mathcal{H}(x, m)$. She also computes $y = r + se$, where y is the very value of $r + se$ on $\mathbb{Z}$. The signature for a message m is (e, y).

Verification step: Given the public-key of the signer (n, g, z), a message m and a signature (e, y), the verifier accepts the signature, if both $|y| \leq a + 1$ and $e = \mathcal{H}(g^{y-ze} \bmod n, m)$ hold, and rejects it, otherwise.

3.2 Parameter Generation

We describe remarks on the parameters for the security of Scheme I. In case of signature $y = r + se$, with $|r| = a$, $|s| = k$ and $|e| = b$, the values of a, b, k, κ shall satisfy $a \geq b + k + \kappa$ for its security.

If an adversary could figure out $r \in \mathbb{Z}_q^*$ from x $(= g^r \bmod n)$ generated by the actual signer, then she could break the signature scheme. We can see the algorithms to extract r, such as *Pollard lambda method* in [Po78] and *the baby-step giant-step method* in [Knu98]. One may say that the former is better than the latter since it has same computational complexity (exponential-time: $O(\sqrt{q})$) but does not need memory. The size of q shall be set up not so that r can be figured out with such an algorithm.

The information leak parameter κ should be set up so that 2^κ-time computation should be intractable.

If $y > ze$ were allowed, then an adversary could impersonate the signer to easily compute y, along with the actual protocol, such that $x = g^{y-ze} \bmod n$ holds. To keep off such an attack, the condition of $c \geq k + 2\kappa$ shall be required from $c + b \geq a + \kappa \geq b + k + 2\kappa$. Furthermore, if $q > 2^c$ were satisfied, then $s = z$ would hold, that is, the secret-key would be disclosed. Hence also $q \leq 2^{c-\kappa}$ shall be required, and it is always held since $q < 2^k \leq 2^{c-2\kappa} \leq 2^{c-\kappa}$.

Next, we describe how to find p, q and an asymmetric basis g in $\mathbb{Z}_n^*$.

- Pick up two primes $p = 2p'p'' + 1$ and $q = 2q'q'' + 1$ such that p' and q' are also primes, and p'' and q'' are odd numbers.
- Choose $\alpha_p \in \mathbb{Z}_p^*$ satisfying $g_p = \alpha_p^{(p-1)/p'} \neq 1 \bmod p$. In the same way, choose $\alpha_q \in \mathbb{Z}_q^*$ satisfying $\alpha_q \neq q - 1 \bmod q$, $\alpha_q^{(q-1)/2} \neq 1 \bmod q$ and $g_q = \alpha_q^{(q-1)/2q'} \neq 1 \bmod q$.
- Compute $n = pq$ and $g = q(q^{-1} \bmod p)g_p + p(p^{-1} \bmod q)g_q \bmod n$.

In the last step, g is computed by using the technique of Chinese Reminder Theorem (CRT). Note that $\mathrm{Ord}_p(g) = p'$ and $\mathrm{Ord}_q(g) = 2q'$. Therefore $\mathrm{Ord}_n(g) = \mathrm{lcm}(p', 2q') = 2p'q'$.

Finally, we discuss secure hash algorithm which we should adopt. If $\mathcal{H}$ were an *ideal* hash function, then the proposed signature scheme would be *secure* as described in Section 3.3. Since such a random function does not exist in the real world, in implementation, we are recommended SHA-1 by [NIST95] which is designed so that the algorithm can be a collision intractable hash function [Dam88].

3.3 Security Analysis

In this paper, we say that a signature scheme is *secure*, if no polynomial-time adversary A can existentially forge a signature under *the adaptive chosen mes-*

sage attack. In this section, we show that Scheme I is secure, by using *the forking lemma* in [PS00], and showing protocol in signature generation step (see Section 3.1) can be simulated by a polynomial-time machine in *the random oracle model* [BR93]. To discuss the provable security, we regard the signature for message m as (x, e, y).

As a strategy, we show that if there exists a polynomial-time adversary which can existentially forge a signature under the strongest attack, that is, an adaptive chosen-message attack, then we can construct a polynomial-time machine which can compute the integer factoring.

We say that a positive function $f(k) : \mathbb{N} \to \mathbb{R}$ is said to be *negligible*, if for any c, there exists a k_c such that $f(k) \leq k^{-c}$ for any $k \geq k_c$. Otherwise f is said to be *non-negligible*.

Lemma 2 Let n be an RSA modulus and g be an asymmetric basis in $\mathbb{Z}_n^*$. Assume that we find $L > 0$ such that $g^L = 1 \bmod n$. Then we can construct a Turing machine M which on input n, g and L outputs a factor of n in time $O(|L||n|^2)$

Proof. (Sketch) This lemma is basically due to [Po00]. Hereafter, we describe how to construct M.

At first, M extract the odd part b of L, such that $L = 2^a b$. Since g is an asymmetric basis in $\mathbb{Z}_n^*$, it holds $g^{2b} = 1 \bmod p$ and $g^{2b} = 1 \bmod q$, and also holds $g^b = 1 \bmod p$ and $g^b = -1 \bmod q$. Then we have the following results: $p \mid g^b - 1$ and $n \nmid g^b - 1$. Consequently, M can find a factor of n by computing $\gcd(g^b - 1 \bmod n, n)$.

Note that modular exponentiation algorithm (resp. extended Euclidean algorithm) has a running time of $O(|L||n|^2)$ (resp. $O(|n|^2)$). Hence M can execute the above steps in time $O(|L||n|^2)$. ∎

Theorem 3 Let Q (resp. R) be the number of queries which a polynomial-time adversary $\mathcal{A}$ can ask to the random oracle (resp. the actual signer). Assume that $2^b q/2^a$ and $1/2^b$ are negligible. Also assume that, by executing adaptive chosen-message attack, $\mathcal{A}$ can forge a signature with non-negligible probability $\varepsilon \geq 10(R+1)(R+q)/2^b$, and with the average running time T. Then we can construct a polynomial-time machine M which can factor n with non-negligible probability in expected time $O(QT/\varepsilon + |n|^{O(1)})$.

Proof. (Sketch) We firstly show that the signatures in the proposed scheme can be statistically simulated by a polynomial-time machine. This machine is simulated according to the protocol like in [PS00].

We denote, by $p(\alpha, \beta, \gamma)$ and $p'(\alpha, \beta, \gamma)$, the probabilities that (α, β, γ) is output by the signature algorithm and the simulator, respectively. We set $\phi = (2^b - 1)(2^k - 1)$, and let $\mathcal{R} : \{0,1\}^* \to \{0,1\}^b$ be an ideal hash function (random oracle) for a given message $m \in \{0,1\}^*$. For an integer A and a positive constant Δ, $\mathcal{N}(\mathcal{R}, A, \Delta)$ is defined to be the number of pairs $(e, y) \in [0, 2^b) \times [A, A + \Delta)$

such that $\mathcal{R}(g^{y-ze}, m) = e$. Then we have the following:

$$p(\alpha,\beta,\gamma) = \frac{\chi\begin{pmatrix} g^{\gamma-z\beta} \bmod n = \alpha, \\ \mathcal{R}(\alpha, m) = \beta, \\ \gamma - s\beta \in [0, 2^a) \end{pmatrix}}{2^a} \quad \text{and} \quad p'(\alpha,\beta,\gamma) = \frac{\chi\begin{pmatrix} g^{\gamma-z\beta} \bmod n = \alpha, \\ \mathcal{R}(\alpha, m) = \beta, \\ \gamma \in [\phi, 2^a) \end{pmatrix}}{\mathcal{N}(\mathcal{R}, \phi, 2^a - \phi)},$$

where for a predicate $\wp$, $\chi(\wp)$ is the characteristic function of $\wp$, that is, $\chi(\wp) = 1$, if $\wp$ is true, and $\chi(\wp) = 0$, otherwise.

Therefore, the summation $\Sigma = \sum_{\alpha,\beta,\gamma} |p(\alpha,\beta,\gamma) - p'(\alpha,\beta,\gamma)|$, has a upper bound of $8q(2^b - 1)/2^a$, because $\Sigma = 2(1 - \mathcal{N}(\mathcal{R}, \phi, 2^a - \phi)/2^a)$ holds similarly with [PS98], because $2^a - \Phi \leq \mathcal{N}(\mathcal{R}, \phi, 2^a - \phi)$ holds, and because $\phi = (2^b - 1)(2^k - 1) \leq (2^b - 1)2q$ follows from $2^{k-1} \leq q < 2^k$. If $q/2^a$ is negligible, then so is $8q(2^b - 1)/2^a$, and consequently, the output by real signer and that by the simulator are statistically indistinguishable.

Next, by using the technique in [PS00], we can get a multiple of $\mathrm{Ord}_n(g)$ such that $g^L = 1 \bmod n$. Here g is an asymmetric basis in $\mathbb{Z}_n^*$, therefore by the result of Lemma 2 we can get a factor of n. ∎

4 Optimized Scheme

In this section, we give an optimized scheme (Scheme II) whish is superior to Scheme I in terms of computational work for a signer. The main feature in Scheme II is that the modulus n consists of three or more primes instead of using an RSA modulus in Scheme I. So a signer can make good use of the technique of CRT more efficiently. For example, in Scheme II with n having three prime factors, the computational cost for pre-computation x $(= g^r \bmod n)$ can be reduced to about 4/9 times of that in the Scheme I (or PS-scheme) with RSA modulus n. A preprint version of Scheme II can be seen in [OTM01]. In this paper, we consider further concrete security in Scheme II.

4.1 Protocols

Key generation step: The signer determines the number of factors, that is, $t \geq 3$, picks up same-size t primes p_i $(1 \leq i \leq t)$ and computes $n = \prod_{i=1}^{t} p_i$. After that, she chooses divisor q of $\lambda(n)$ and finds an order-q element $g \in \mathbb{Z}_n^*$. Also she picks up a random number $z \in \mathbb{Z}_{2^c}$ and compute $s = z \bmod q$. The secret-key is s and the corresponding public-key is (n, g).

The other steps are executed in the same way as Scheme I (see Section 3.1).

4.2 Description

The conditions of parameters such as k, κ, a, b and c are the same as those in Scheme I (see Section 3.2). Furthermore, primes p_i $(1 \leq i \leq t)$ and $g \in \mathbb{Z}_n^*$ will be generated under the line of work described in Section 3.2.

In [PS98,PS99] we can see the two types of attack: *one key attack*, an adversary try to forge valid signatures for fixed public key, and *possible key attack*, an adversary try to forge valid signatures for possible public keys, where possible public key means any public key satisfying the condition of the parameter. The security consideration under the one key attack scenario seems to be more strict analysis of security than that under the possible key attack scenario.

We have seen that the security in Scheme I is based on integer factoring. On the other hand, it is not unknown, under the one key attack scenario, whether Scheme II is as secure as the problem or not. To estimate more concrete security, we define the following problem.

Definition 4 (Finding order problem) This problem is as follows. Given $n \in \mathbb{N}_{>1}$ and $g \in \mathbb{Z}_n^*$, find L, where L is a multiple of $\mathrm{Ord}_n(g)$ and $|L|$ is bounded by a polynomial in $|n|$. ■

In Scheme II, if we assume the intractability of finding order problem, same result like Theorem 3 is obtained. Then the result (i.e. theorem) is proved, without loss of generality, using in the proof of the Theorem 3.

5 Integer Factoring Problem

In this section, we consider the secure size of n, and also discuss secure number of the prime factors for n in our schemes.

Of course, if the modulus n were factored, then the proposed signature schemes would be broken. In [LLMP90], we can see *the number field sieve method* for factorization, which is the most efficient algorithm ever proposed, and whose running time depends upon the size of n. On the other hand, in [Len87], we can see *the elliptic curve method*, which is also one of efficient algorithms for factorization, and whose running time depends upon the size of factors of n. Therefore, the faster one is determined according to the size of the input and upon the number of the factors of n.

As for Scheme II, referring to [Sil99] for computational cost of algorithms, in case that $|n| = 1024$ and that n has three prime factors, the number field sieve method is faster, whereas in case n has four prime factors, the other is faster. Hence supposing that $|n|$ is 1024 and t is 3 in the proposed scheme, and that $|n|$ is 1024 in PS-scheme, we can say that the number field sieve method is the faster (and fastest) algorithm to factor n in the respective schemes, and that the respective computational cost for factoring n can be almost the same.

6 Performance

In this section, we evaluate the efficiency of our schemes by comparing existing schemes. The parameters in the proposed Scheme I (resp. Scheme II) are set up to be $|n| = 1024$, $k = 160$ by taking $\kappa = 80$, $b = 80$ and $a = c = 320$ (resp. $|n| = 1024$, $t = 3$, $k = 192$ by taking $\kappa = 80$, $b = 80$ and $a = c = 352$).

Scheme	UMP	CPC ($\times M$)	CSG	CVF ($\times M$)	SPK (bits)	SSK (bits)	SSig (bits)
Scheme I $\lvert n\rvert = 1024$, $a = 320$, $\kappa = 80$	Integer factoring	240	80×160	600	2048	160	400
Scheme II $\lvert n\rvert = 1024$, $a = 352$ $t = 3$, $\kappa = 80$	Finding order	176	80×192	648	2048	192	432
PS-scheme [PS99] $\lvert n\rvert = 1024, \lvert A\rvert = 672$	Integer factoring	384	80×512	1656	1024	513	752
GPS-scheme [PS98] $\lvert n\rvert = 1024$	Discrete log. modulo n	384	80×1024	1796	3072	1024	1264

Table 1. Performance of signature schemes

Table 1 gives the performance of various signature schemes including ours. Here, a primitive arithmetic of binary methods [Knu81] is used. For all schemes in the table, we set up the parameter under the line of the one key attack scenario. Hence the size of secret-key in GPS-scheme is 1024 bits. For more discussion on it, we refer to [Po00].

UMP means the underlying mathematical problem that the signature scheme relies on for its security. The terms CPC, CSG and CVF mean the computational cost for pre-computation, signature generation and verification, respectively. The terms SPK, SSK and SSig means the size of a public-key, a secret-key and a signature, respectively.

In CPC, the signer uses the technique of CRT if it is possible. In SPK with our schemes, the size of public-key is optimized: we regard actual public-key as (n, g), and z is computed by $z = \mathcal{H}'(n, g)$, where $\mathcal{H}'$ is a hash function $\mathcal{H}' : \{0,1\}^* \to \{0,1\}^c$.

For respective computational cost, the unit M represents the computational cost for one multiplication under a 1024-bit modulus, $\alpha \times \beta$ represents the computational cost for multiplication of an α-bit number and a β-bit number on $\mathbb{Z}$.

Since PS-scheme is intended to be used with a modulus product of two *strong* primes, $g = 2$ is a correct basis and do not have to be included in the public key. Consequently, we set SPC $= 1024$ for PS-scheme. Therefore, one may say that PS-scheme is more efficient than our schemes in terms of size of public key.

We can say that the proposed signature scheme is quite efficient one in view of both the computational cost and the data size. Concrete to say, Scheme I (resp. Scheme II) enables the computational cost to be reduced by 38% (resp. 54%) for pre-computation, by 69% (resp. 63%) for signature generation, and by 64% (resp. 61%) for verification, comparing with PS-scheme. For the data size,

the secret-key size in ours is 69% (resp. 63%) of that in PS-scheme, and the signature size is 47% (resp. 43%) of that in PS-scheme.

By Table 1, we can say that the proposed signature scheme is efficient, and requires a relatively weak computational assumption for its security.

7 Conclusion

In this paper, we have proposed efficient signature schemes, which are derived from a three-pass identification scheme, and which are constructed by improving PS-scheme in terms of a compactness of signature. As well as PS-scheme (or GPS-scheme), the proposed schemes are so-called "on the fly" signature schemes, that is, it does not require modulo reduction in the signature generation step. We have shown that our schemes are existentially unforgeable against any polynomial-time adversaries that can execute adaptive chosen message attack in the random oracle model. Furthermore, the underlying computational problem in ours is the integer factoring problem in Scheme I and mathematically well defined problem (i.e. finding order problem) in Scheme II, respectively. We also have shown that ours are more efficient than PS-scheme in view of the computational cost and also in view of the size of a secret-key and a signature.

References

[BR93] M. Bellare and P. Rogaway: "*Random oracles are practical: a paradigm for designing efficient protocols*", Proceedings of the 1st ACM Conference on Computer and Communications Security (CCS), 1993.

[Dam88] I. Damgård: "*Collision free hash functions and public key signature schemes*", Advances in cryptology - Eurocrypt'87, Lecture Notes in Computer Science 304, Springer-Verlag, pp.203-216, 1988.

[ElG85] T. ElGamal: "*A public-key cryptosystem and a signature scheme based on discrete logarithms*", IEEE transactions of information theory, vol.IT-31, no.4, pp.469-472, 1985.

[FFS88] U. Feige, A. Fiat and A. Shamir: "*Zero-knowledge proofs of identity*", Journal of cryptology, vol.1, pp.77-95, 1988.

[FMO92] A. Fujisaki, S. Miyaguchi and T. Okamoto: "*ESIGN: an efficient digital signature implementation for smart cards*", Advances in cryptology - Eurocrypt'91, Lecture Notes in Computer Science 547, Springer-Verlag, pp.446-457, 1992.

[Gir91] M. Girault: "*An identity-based identification scheme based on discrete logarithms modulo a composite number*", Advances in cryptology - Eurocrypt'90, Lecture Notes in Computer Science 473, Springer-Verlag, pp.481-486, 1991.

[Gir92] M. Girault: "*Self-certified public keys*", Advances in cryptology - Eurocrypt'91, Lecture Notes in Computer Science 547, Springer-Verlag, pp.490-497, 1992.

[Knu81] D. E. Knuth: "*Seminumerical Algorithms*", The art of computer programming, vol.2, Second edition, Addison-Wesley, 1981.

[Knu98] D. E. Knuth: "*Sorting and Searching*", The art of computer programming, vol.3, Second edition, Addison-Wesley, 1998.

[LLMP90] A. K. Lenstra, H. W. Lenstra Jr., M. S. Manasse and J. M. Pollard: "*The number field sieve*", Proceedings of ACM Annual Symposium on Theory of Computing, pp.564-572, 1990.

[Len87] H. W. Lenstra Jr.: "*Factoring integers with elliptic curves*", Annuals of Mathematics, vol.126, pp.649-673, 1987.

[NMVR94] D. Naccache, D. M'raihi, S. Vaudenay and D. Raphaeli : "*Can DSA be improved ?*", Advances in cryptology - Eurocrypt'94, Lecture Notes in Computer Science 950, 1995.

[NIST91] National Institute of Standards and Technology (NIST): "*Digital signature standards (DSS)*", Federal Information Processing Standards, 1991.

[NIST95] National Institute of Standards and Technology (NIST): "*Secure hash standards (SHS)*", Federal Information Processing Standards, 1995.

[OTM01] T. Okamoto, M. Tada and A. Miyaji: "*Proposal of Efficient Signature Schemes Based on Factoring*" (in Japanese), Transactions of Information Processing Society of Japan, vol. 42, no. 8, pp.2123-2133, 2001.

[Po00] D. Pointcheval: "*The Composite Discrete Logarithm and Secure Authentication*", Advances in cryptology - PKC'00, Lecture Notes in Computer Science 1751, 2000.

[Po78] J. Pollard: "*Monte Carlo methods for index computation mod p*", Mathematics of Computation, vol 32, pp.918-924, 1978.

[PS96] D. Pointcheval and J. Stern: "*Security proofs for signature schemes*", Advances in cryptology - Eurocrypt'96, Lecture Notes in Computer Science 1070, 1996.

[PS00] D. Pointcheval and J. Stern: "*Security arguments for digital signatures and blind signatures*", Journal of cryptology, vol.13, no.3, Springer-Verlag, pp.361-396, 2000.

[PS98] G. Poupard and J. Stern: "*Security analysis of a practical 'on the fly' authentication and signature generation*", Advances in cryptology - Eurocrypt'98, Lecture Notes in Computer Science 1403, Springer-Verlag, pp.422-436, 1998.

[PS99] G. Poupard and J. Stern: "*On the fly signatures based on factoring*", Proceedings of the 6th ACM Conference on Computer and Communications Security (CCS), pp.48-57, 1999.

[Riv92] R. L. Rivest: "*The MD5 message-digest algorithm*", Internet Request for Comments, RFC 1321, 1992.

[RSA78] R. L. Rivest, A. Shamir and L. M. Adleman: "*A method for obtaining digital signatures and public-key cryptosystems*", Communications of the ACM, vol.21, no.2, pp.120-126, 1978.

[Sch91] C. P. Schnorr: "*Efficient signature generation by smart cards*", Journal of cryptology, vol.4, Springer-Verlag, pp.161-174, 1991.

[Sil99] R. D. Silverman: "*A cost-based security analysis of symmetric and asymmetric key length*", RSA Laboratories, CryptoBytes, Bulletins, no.13, 1999. Available from: `http://www.rsasecurity.com/rsalabs/bulletins/bulletin13.html`.

[SZ00] R. Steinfeid and Y. Zheng: "*A Signencryption Scheme Based on Integer Factorization*", Advances in cryptology - ISW'00, Lecture Notes in Computer Science 1975, 2000.