Provably Secure and Practical Identification Schemes and Corresponding Signature Schemes

Tatsuaki Okamoto

NTT Laboratories
Nippon Telegraph and Telephone Corporation
1-2356, Take, Yokosuka-shi, Kanagawa-ken, 238-03 Japan
Email: okamoto@sucaba.ntt.jp

Abstract. This paper presents a three-move interactive identification scheme and proves it to be as secure as the discrete logarithm problem. This provably secure scheme is almost as efficient as the Schnorr identification scheme, while the Schnorr scheme is not provably secure. This paper also presents another practical identification scheme which is proven to be as secure as the factoring problem and is almost as efficient as the Guillou-Quisquater identification scheme: the Guillou-Quisquater scheme is not provably secure. We also propose practical digital signature schemes based on these identification schemes. The signature schemes are almost as efficient as the Schnorr and Guillou-Quisquater signature schemes, while the security assumptions of our signature schemes are weaker than those of the Schnorr and Guillou-Quisquater.signature schemes. This paper also gives a theoretically generalized result: a three-move identification scheme can be constructed which is as secure as the random-self-reducible problem. Moreover, this paper proposes a variant which is proven to be as secure as the difficulty of solving both the discrete logarithm problem and the specific factoring problem simultaneously. Some other variants such as an identity-based variant and an elliptic curve variant are also proposed.

1 Introduction

Public-key based identification schemes and digital signature schemes are very useful and fundamental tools in many applications such as electronic fund transfer and online systems for preventing data access by invalid users and proving the authenticity of messages.

Identification schemes are typical applications of zero-knowledge interactive proofs [GMRa], and several practical zero-knowledge identification schemes have been proposed [Bet, FiS, FFS, OhO1]. However, the zero-knowledge identification schemes have the following shortcomings in practice, where we simply call "black-box simulation zero-knowledge" "zero-knowledge", since we do not know of any effective measure to prove zero-knowledgeness except the black-box simulation technique, although "auxiliary-input zero-knowledge" is more general than "black-box simulation zero-knowledge":

E.F. Brickell (Ed.): Advances in Cryptology - CRYPTO '92, LNCS 740, pp. 31-53, 1993.

- A zero-knowledge identification scheme requires more than three interactions (three-moves [1]) from Goldreich et.al.'s result [GK] unless the language for the proof is trivial. A zero-knowledge protocol is less practical than the corresponding (three-move) parallel version since interaction over a network often requires more time than taken by the calculation in these identification schemes. Although four-move and five-move zero-knowledge proofs have been proposed [BMO1, FeS2], these protocols impose fairly big additional communication and computation overheads compared to the three-move parallel versions (especially Type 2 below).
  Note: Here, the "(three-move) parallel version" denotes two types of protocols. One (Type 1) is just the parallel execution of a zero-knowledge protocol (e.g., the three-move version of the Fiat-Shamir scheme with $k = 1$ and $t = Poly(|n|)$ [FiS]). The other (Type 2) is a protocol which can be converted to zero-knowledge by executing the protocol repeatedly many times and setting the security parameter of one repetition to be constant (e.g., the three-move and higher-degree version of the Fiat-Shamir scheme [GQ, OhO1]). The communication complexity of the Type 1 protocol is the same as that of the original zero-knowledge protocol. Usually, the communication complexity of the Type 2 protocol is much less than that of the corresponding zero-knowledge protocol (or Type 1).
- No zero-knowledge identification can be converted into a signature scheme using Fiat-Shamir's technique [FiS], which is a truly practical way of converting an identification scheme into a signature scheme with a one-way hash function. This is because: if the identification protocol is zero-knowledge, the signature converted from this identification protocol through Fiat-Shamir's technique can be forged by using the same algorithm as the simulation for proving the zero-knowledgeness of the identification protocol. Therefore, for example, the above-mentioned four-move and five-move zero-knowledge proofs [BMO1, FeS2] cannot be used to construct a signature scheme.

In contrast, the three-move identification schemes [Bet, BM1, FiS, FFS, GQ, OhO1, Sch], which are the parallel version (Type 2) of zero-knowledge proofs, have the following merits in practice.

- The communication and computation overheads are smaller than those of the zero-knowledge identification schemes.
- The three-move identification schemes can be converted into practical signature schemes by using Fiat-Shamir's technique.

How then can we prove the security of the three-move identification schemes? As mentioned above, the zero-knowledge notion seems to be ineffective for this purpose. Feige, Fiat and Shamir [FFS] have developed an effective measure called "no-useful information transfer" to prove the security of their three-move identification scheme. Ohta and Okamoto [OhO1] have proposed a variant called

[1] A scheme is called "one-move" if prover A only sends one message to verifier B, and is called "two-move" if B sends to A and then A sends to B. "j-move" is defined in the obvious way.

"no transferable information with (sharp threshold) security level," which characterizes the security level theoretically. Therefore, only "no-useful information transfer" [FFS] and its variant [OhOl] have been known to be effective to prove the security of three-move identification schemes.

Only three three-move identification schemes [FFS, OhOl, BMl] have been proven to be secure assuming reasonable primitive problems, in the sense of [FFS, OhOl]. The Feige-Fiat-Shamir identification scheme [FFS], based on square root mod n, has been proven to be as secure as the factoring problem. The Ohta-Okamoto scheme [OhOl], which is the higher (the L-th) degree modification of the Feige-Fiat-Shamir scheme, has been proven to be as secure (with sharp threshold security level $1/K$) as factoring, where $v^{1/L}$ mod n has at least K solutions (e.g., $\gcd(L, p-1) = K$; see [OhOl] for more detail conditions). The Brickell-McCurley scheme [BMl], which is a modification of the Schnorr scheme [Sch], has been proven to be secure assuming that it is intractable to find a factor, q, of $p-1$, given additional information g whose order is q in Z_p^*, although the security of their scheme also depends on the discrete logarithm.

Therefore, there is no existing alternative that is "provably secure" and "three-move" practical identification if factoring intractability fails in the future, since the security of all these provably secure schemes depends on the factoring assumption. In addition, although their schemes are efficient, they have some shortcomings in practice: the transmitted information size and memory size cannot be small simultaneously [FFS], and a priori fixed value v (e.g., v is the identity of a user) cannot be used as a public key [OhOl], (or the identity based scheme [Sha] cannot be constructed on this scheme). In addition, the security assumption of [BMl] is fairly stronger than the ordinary factoring problem (or the level of the provable security is lower than those of [FFS, OhOl]).

In contrast, other previously proposed practical three-move identification schemes, the Schnorr [Sch] and Guillou-Quisquater [GQ] schemes, have some merits compared to [FFS, OhOl, BMl]: The security of the Schnorr scheme depends on the discrete logarithm, which is a promising alternative if factoring becomes tractable, since we have several different types of discrete logarithms such as elliptic curve logarithms which seem to be more intractable than factoring. Moreover, the transmitted information size and memory size with these schemes can be small simultaneously, while it is impossible in [FFS]. The Schnorr scheme is more efficient than [BMl]. In addition, in the Guillou-Quisquater scheme, a priori fixed value v can be used as the public key. Unfortunately, the Schnorr and Guillou-Quisquater schemes are not provably secure. The difficulty of proving the security of these schemes resides in the fact that the discrete logarithm and RSA inversion have single solutions in restricted domains, that is, $\log_g x \bmod p$ has a single solution (x is in the restricted domain, $\{0, 1, \ldots, \mathrm{ord}(g) - 1\}$), and $x^{1/e} \bmod n$ has also a single solution ($\gcd(e, \phi(n)) = 1$, ϕ is the Euler function).

In this paper, we propose three-move identification schemes that are proven to be as secure as the discrete logarithm or RSA inversion. We also propose a variant which is proven to be as secure as the factoring problem. Our new schemes inherit almost all the merits of the Schnorr and Guillou-Quisquater schemes even though they are provably secure. That is, these schemes are almost as efficient as the Schnorr and Guillou-Quisquater identification schemes from all practical viewpoints such as communication overhead, interaction number, required memory size, and processing speed. In addition, the new schemes duplicate the other advantage of the Guillou-Quisquater scheme: the identity based schemes can be constructed on these schemes.

This paper also develops new practical digital signature schemes from the proposed provably secure three-move identification schemes. The signature schemes are almost as efficient as the Schnorr and Guillou-Quisquater signature schemes. while the security assumptions of our schemes are weaker than those of the Schnorr and Guillou-Quisquater signature schemes. That is, the security (existentially unforgeable against adaptive chosen message attacks [GMRi]) of our new signature schemes only depends on just one reasonable assumption about the one-way hash function (or the existence of a "correlation-free one-way hash function") as well as the primitive assumption (e.g., the intractability assumption of the discrete logarithm).

We also extend these specific and practical results to a more general and theoretical result. We show that any random-self-reducible problem [TW] can lead to a provably secure and three-move identification scheme.

We also construct some variants of our new identification and signature schemes. One is a variant of our identification scheme based on the discrete logarithm using the idea of the Brickell-McCurley scheme [BM1]. This variant is proven to be as secure as the difficulty of solving both the discrete logarithm and the specific factoring problem (or the finding order problem) simultaneously, while, as mentioned above, the Brickell-McCurley scheme is proven to be secure assuming the intractability of the finding order problem, although the security of their scheme also depends on the discrete logarithm. Some other variants of our scheme, identity-based and certification-based versions, and an elliptic curve version, are also proposed. The elliptic curve variant has the significant property that it is proven to be secure assuming the intractability of the (non-supersingular) elliptic curve logarithms against which only exponential-time attacks are known so far.

2 Definition of Secure Identification

2.1 Identification

Definition 1. An *identification* scheme consists of two stages:

1. Initialization: In this stage, each user (e.g., A) generates a secret key (e.g., SK_A) and a public key (e.g., PK_A) by using probabilistic polynomial-time generation algorithm G on input of the key size. A link between each user and its public key is established. Note that in some schemes a part of the public key can be commonly shared among all users as a system parameter.
2. Operation: In this stage any user (e.g., A) can demonstrate its identity to a verifier by performing some identification protocol related to its public key

(e.g., PK_A), where the input for the verifier is the public key (e.g., PK_A). At the conclusion of this stage, the verifier either outputs "accept" or "reject".

2.2 Security of Identification schemes

We define a *secure* identification scheme based on the definition (the "no useful information transfer") given by Feige et. al. [FFS].

Definition 2. A prover A (resp. verifier B) is a "good" prover denoted by $\overline{A}$ (resp. "good" verifier denoted by $\overline{B}$), if it does not deviate from the protocols dictated by the scheme. Let $\widetilde{A}$ be a fraudulent prover who does not complete the Initialization stage of Definition 1 as A and may deviate from the protocols (so another person/machine can simulate $\widetilde{A}$). $\widetilde{B}$ is not a good B. $\widetilde{A}$ and $\widetilde{B}$ are assumed to be polynomial time bounded machines, which may be nonuniform.

An identification scheme (A, B) is *secure* if

1. $(\overline{A}, \overline{B})$ succeeds with overwhelming probability.
2. There is no coalition of $\widetilde{A}, \widetilde{B}$ with the property that, after a polynomial number of executions of $(\overline{A}, \widetilde{B})$ and relaying a transcript of the communication to $\widetilde{A}$, it is possible to execute $(\widetilde{A}, \overline{B})$ with nonnegligible probability of success. The probability is taken over the distribution of the public key and the secret key as well as the coin tosses of $\overline{A}$, $\widetilde{B}$, $\widetilde{A}$, and $\overline{B}$, up to the time of the attempted impersonation.

Remark: When an identification scheme is "witness hiding" [FeS1] and an interactive proof of "knowledge" [FFS], this scheme is secure in the sense of Definition 2. This is roughly because if there exists $(\widetilde{A}, \widetilde{B})$ with nonnegligible probability of success, we can construct a knowledge extractor (from the "knowldge soundness"), which leads to contradiction with "witness hiding". Thus there are two ways to prove the security of Definition 2: One is to prove it directly as in [FFS, OhO1], and the other way is to prove that a scheme is "witness hiding" and an interactive proof of "knowledge". Some schemes such as [OhO1] seem to be proven only in the former way, since the knowledge soundness is sometimes hard to prove (e.g., [OhO1]). In this paper, we will prove our schemes in the former way, since it is compatible with the way to prove it by a variant of Definition 2, [OhO1], to be described below, although we can prove them in the latter way.

In the Appendix A, we introduce a variant of the "no useful information transfer" given by Ohta and Okamoto [OhO1], called "no transferable information with (sharp threshold) security level". This notion does not guarantee the security guaranteed by [FFS] i.e., the success probability of cheating by any adversary $(\widetilde{A}, \widetilde{B})$ is negligible in an asymptotic sense. However, the notion sheds light on another aspect of the security of identification schemes, the *security level* in a non-asymptotic sense. In practice, the security parameter is fixed in a system (e.g., the values of k and t of the Fiat-Shamir scheme [FiS]). Then we can assume a fixed security level for the system. The definition [OhO1] guarantees that such a fixed security level has theoretical significance [2]. Note that

[2] An asymptotic extension of the security level is recently studied in [CD]

this notion is defined essentially in an asymptotic manner although the security level is characterized in a non-asymptotic manner. The provable security of an identification scheme can be guaranteed by both these notions.

3 Proposed Three-Move Identification Schemes

3.1 Identification Scheme as Secure as the Discrete Logarithm

In this subsection, we propose a new scheme which is almost as efficient as the Schnorr identification scheme [Sch], and prove that it is as secure as the discrete logarithm problem.

A user generates a public key (p, q, g_1, g_2, t, v) and a secret key (s_1, s_2) and publishes the public key. Here, if g_2 is calculated by $g_2 = g_1^{\alpha} \bmod p$, α can be discarded after publishing g_2.

- primes p and q such that $q | p - 1$. (e.g., $q \geq 2^{140}$, and $p \geq 2^{512}$.)
- g_1, g_2 of order q in the group Z_p^*, and an integer $t = O(|p|)$. (e.g., $t \geq 20$.)
- random numbers s_1, s_2 in Z_q, and $v = g_1^{-s_1} g_2^{-s_2} \bmod p$.

Remark: (p, q, g_1, g_2, t) can be published by a system manager and used commonly by all system users as a system parameter. The system manager should then also publish some information to confirm to users that these parameters were selected honestly. For example, (s)he publishes some witness that no trapdoor exists in p, g_1, g_2, or that these values are generated honestly. Since the primality test for p and q is fairly easy for users, they can confirm for themselves that g_1 and g_2 are both of order q. When, as described above, the system parameter is generated and published by each user individually, (s)he does not need to publish such information.

We now describe our new identification scheme (Identification scheme 1) by which party A (the prover) can prove its identity to B (the verifier).

Protocol: Identification scheme 1

Step 1 A picks random numbers $r_1, r_2 \in Z_q$, computes

$$x = g_1^{r_1} g_2^{r_2} \bmod p,$$

and sends x to B.

Step 2 B sends a random number $e \in Z_{2^t}$ to A.

Step 3 A sends to B (y_1, y_2) such that

$$y_1 = r_1 + es_1 \bmod q, \quad \text{and} \quad y_2 = r_2 + es_2 \bmod q.$$

Step 4 B checks that

$$x = g_1^{y_1} g_2^{y_2} v^e \bmod p.$$

If it holds, B accepts, otherwise rejects.

Next, we prove the security of the above identification scheme. First, we show a definition and lemma in preparation.

Definition 3. Let RA denote $\tilde{A}$'s random tape, and RB denote $\overline{B}$'s random tape. The possible outcomes of executing $(\tilde{A}, \overline{B})$ can be summarized in a large Boolean matrix H whose rows correspond to all possible choices of RA. Its columns correspond to all possible choices e of RB, and its entries are 1 if $\overline{B}$ accepts $\tilde{A}$'s proof, and 0 if otherwise.

When the success probability of $\tilde{A}$ is ε (or the rate of 1-entries in H is ε), we call a row *heavy* if its ratio of 1's is at least $\varepsilon/2$.

Lemma 4. *If, given A's public key (p, q, g_1, g_2, t, v), the success probability, ε, of $\tilde{A}$ is greater than 2^{-t+1}, then there exists a probabilistic algorithm which runs in expected time $O(\|\tilde{A}\|/\varepsilon)$ and outputs the history of two accepted executions of $(\tilde{A}, \overline{B})$, (x, e, y_1, y_2) and (x, e', y_1', y_2'), where $e \neq e'$. Here, $\|\tilde{A}\|$ denotes the time complexity of $\tilde{A}$. The success probability ε is taken over the coin tosses of $\tilde{A}$ and $\overline{B}$.*

Sketch of Proof:

Assume that at least 1/2 of the 1's in H are not located in heavy rows. Then the fraction of non-heavy rows in H, which we denote τ, is estimated as follows: $\tau \geq \frac{2^t\varepsilon/2}{2^t\varepsilon/2-1} > 1$. This is a contradiction. Therefore, at least 1/2 of the 1's in H are located in heavy rows. Since ε is greater than 2^{-t+1} and the width of H is 2^t, a heavy row contains at least two 1's. To find two 1's in the same row, we thus adopt the following strategy:

1. Probe $O(1/\varepsilon)$ random entries in H (or pick (RA, e) randomly and check it, and repeat this until successful).
2. After the first 1 is found (or accepted (x, e, y_1, y_2) with RA is found), probe $O(1/\varepsilon)$ random entries along the same row (or probe (x, e', y_1', y_2') with the same RA).

Since at least 1/2 of the 1's in H are located in heavy rows, this strategy succeeds with constant probability in $O(1/\varepsilon)$ probes. □

Definition 5. The discrete logarithm is (nonuniformly) intractable, if any family of boolean circuits, which, given properly chosen (g_1, g_2, p, q) in the same distribution as the output of key generator G, can compute the discrete logarithm $\alpha \in Z_q$ ($g_2 = g_1^\alpha \bmod p$) with nonnegligible probability, must grow at a rate faster than any polynomial in the size of the input, $|p|$.

Remark The discrete logarithm above might be less intractable than that when the order of g_1 is greater than q (e.g., $p - 1$), although no attack has yet been reported when q is appropriately large (considering an attack, [PH]).

Theorem 6. *Identification scheme 1 is secure if and only if the discrete logarithm is intractable.*

Sketch of Proof:

(Only if:)

Suppose that the discrete logarithm is not intractable. Clearly a (nonuniform) polynomial time machine can calculate (s'_1, s'_2) satisfying $v = g_1^{-s'_1} g_2^{-s'_2} \bmod p$ with nonnegligible probability. Thus Identification scheme 1 is not secure.

(If:)

To prove the "If" part, we show that if Identification scheme 1 is not secure, then, given (g_1, g_2, p, q) with the same distribution as the output of key generator G, the discrete logarithm $\alpha \in \boldsymbol{Z}_q$ $(g_2 = g_1^{\alpha} \bmod p)$ can be computed by a polynomial time machine P with non-negligible probability.

Assume that Identification scheme 1 is not secure. Then $(\tilde{A}, \overline{B})$ can be accepted with nonnegligible probability ε after $O(|p|^c)$ executions of $(\overline{A}, \tilde{B})$. The complete history of the executions of $(\overline{A}, \tilde{B})$ and $(\tilde{A}, \overline{B})$ can be simulated by one polynomial time procedure P, which may be nonuniform, if P knows $\overline{A}$'s secret key.

To calculate the discrete logarithm $\alpha \in \boldsymbol{Z}_q$ $(g_2 = g_1^{\alpha} \bmod p)$, given (g_1, g_2, p, q), P firstly chooses s_1^*, $s_2^* \in \boldsymbol{Z}_q$ randomly, and calculates $v = g_1^{-s_1^*} g_2^{-s_2^*} \bmod p$.

Then, using (s_1^*, s_2^*) as $\overline{A}$'s secret key, P simulates $(\overline{A}, \tilde{B})$ as well as $(\tilde{A}, \overline{B})$. So, for (v, g_1, g_2, p, q), after simulating $O(|p|^c)$ executions of $(\overline{A}, \tilde{B})$, P tries to find two accepted interactions of $(\tilde{A}, \overline{B})$, (x, e, y_1, y_2) and (x, e', y'_1, y'_2) $(e \neq e')$. From Lemma 4, this is possible with overwhelming probability, since ε is nonnegligible i.e. greater than 2^{-t+1}.

P can then calculate $(s_1, s_2) = ((y_1 - y'_1)/(e - e') \bmod q, (y_2 - y'_2)/(e - e') \bmod q)$ by

$$y_1 = r_1 + es_1 \bmod q, \quad y_2 = r_2 + es_2 \bmod q,$$

$$y'_1 = r_1 + e's_1 \bmod q, \quad y'_2 = r_2 + e's_2 \bmod q.$$

There are q solutions of (s_1, s_2) which satisfy $v = g_1^{-s_1} g_2^{-s_2} \bmod p$, given (v, g_1, g_2, p, q). Even an infinitely powerful $\tilde{B}$ cannot determine from x's, y_1's, and y_2's sent by $\overline{A}$ during the execution of $(\overline{A}, \tilde{B})$ which (s_1, s_2) satisfying $v = g_1^{-s_1} g_2^{-s_2} \bmod p$ actually uses. To prove this, for two different solutions, (s_1, s_2) and (s_1^*, s_2^*) satisfying $v = g_1^{-s_1} g_2^{-s_2} \equiv g_1^{-s_1^*} g_2^{-s_2^*} \pmod{p}$, we show that even an infinitely powerful $\tilde{B}$ cannot determine which solution was used from x's, y_1's, and y_2's. When $r_1^* = r_1 + e(s_1 - s_1^*) \bmod q$ and $r_2^* = r_2 + e(s_2 - s_2^*) \bmod q$, the following three equations hold.

$$x = g_1^{r_1} g_2^{r_2} \equiv g_1^{r_1^*} g_2^{r_2^*} \pmod{p},$$

$$y_1 = r_1 + es_1 \equiv r_1^* + es_1^* \pmod{q},$$

$$y_2 = r_2 + es_2 \equiv r_2^* + es_2^* \pmod{q}.$$

In addition, the distributions of (r_1, r_2) and (r_1^*, r_2^*) are exactly equivalent even if they satisfy the above relation. Hence, although P knows (s_1^*, s_2^*), (s_1, s_2), which is calculated by P by simulating the operations of $(\widetilde{A}, \widetilde{B})$ and $(\widetilde{A}, \overline{B})$, is independent from (s_1^*, s_2^*).

Therefore, (s_1^*, s_2^*) which was randomly chosen by P at first is different with probability $(q-1)/q$ from (s_1, s_2). Thus, α can be calculated with probability $(q-1)/q$ from (s_1, s_2) and (s_1^*, s_2^*) such that $\alpha = (s_1 - s_1^*)/(s_2^* - s_2) \bmod q$. The total success probability of P is nonnegligible.

This contradicts the intractability assumption of the discrete logarithm. □

Theorem 7. *Let $t = O(1)$. Identification scheme 1 is secure with sharp threshold security level $1/2^t$ if and only if the discrete logarithm is intractable.*

The proof of Theorem 7 is similar to that of Theorem 6. It is shown in the final version.

3.2 Identification Scheme as Secure as RSA Inversion

This subsection proposes another practical identification scheme which is almost as efficient as the Guillou-Quisquater identification scheme [GQ], and proves that it is as secure as RSA inversion.

A user generates a public key (a, k, n, v) and a secret key (s_1, s_2) and publishes the public key. Here, p, q can be discarded after publishing n. Note that (a, k) can be common among users as the system parameter.

- primes p, q, $n = pq$, and prime k such that $\gcd(k, \phi(n)) = 1$ and $|k| = O(|n|)$, where $\phi(n) = \mathrm{lcm}(p-1, q-1)$. (e.g., $k \geq 2^{20}$, $n \geq 2^{512}$)
- random number $s_1 \in Z_k$, and random numbers $a, s_2 \in Z_n^*$, and $v = a^{-s_1} s_2^{-k} \bmod n$.

We now describe our new identification scheme (Identification scheme 2) by which party A (the prover) can prove its identity to B (the verifier).

Protocol: Identification scheme 2

Step 1 A picks random numbers $r_1 \in Z_k$ and $r_2 \in Z_n^*$, computes

$$x = a^{r_1} r_2^k \bmod n,$$

and sends x to B.

Step 2 B sends a random number $e \in Z_k$ to A.

Step 3 A sends to B (y_1, y_2) such that

$$y_1 = r_1 + e s_1 \bmod k, \quad y_2 = a^{\lfloor (r_1 + e s_1)/k \rfloor} r_2 s_2^e \bmod n.$$

Step 4 B checks that $x = a^{y_1} y_2^k v^e \bmod n$.

Definition 8. RSA inversion is (nonuniformly) intractable, if any family of boolean circuits, which, given properly chosen (a, k, n) in the same distribution as the output of key generator G, can compute $a^{1/k} \bmod n$ with nonnegligible probability, must grow at a rate faster than any polynomial in the size of the input, $|n|$.

Theorem 9. *Identification scheme 2 is secure if and only if RSA inversion is intractable.*

Sketch of Proof:

(Only if:)

Suppose that the RSA inversion is not intractable. Clearly a (nonuniform) polynomial time machine can calculate (s'_1, s'_2) satisfying $v = a^{-s'_1} {s'_2}^{-k} \bmod n$ with nonnegligible probability. Thus Identification scheme 2 is not secure.

(If:)

To prove the "If" part, we can prove this in a manner similar to the "if" part proof of Theorem 6. So we only sketch the different points here.

First, P chooses $s_1^* \in Z_k$, and $s_2^* \in Z_n^*$ randomly, and calculates $v = a^{-s_1^*} {s_2^*}^{-k} \bmod n$.

Then, for (a, k, n, v), P finds (x, e, y_1, y_2) and (x, e', y'_1, y'_2) $(e \neq e')$ by the technique of Lemma 4.

Next P calculates $s_1 = (y_1 - y'_1)/(e - e') \bmod k$, and $r_1 = y_1 - es_1 \bmod k$. P then calculates X, Y as follows:

$$X = \frac{y_2 / a^{\lfloor (r_1 + es_1)/k \rfloor}}{y'_2 / a^{\lfloor (r_1 + e's_1)/k \rfloor}} \bmod n \ (= s_2^{e-e'} \bmod n),$$

$$Y = 1/(va^{s_1}) \bmod n \ (= s_2^k \bmod n).$$

Since $\gcd(k, e - e') = 1$ (as k is prime), P can compute α, β satisfying $\alpha(e - e') + \beta k = 1$ by the extended Euclidean algorithm. Hence P calculates $s_2 = X^\alpha Y^\beta \bmod n$.

There are k solutions of (s_1, s_2) which satisfy $v = a^{-s_1} s_2^{-k} \bmod n$, given (v, n, a, k). Even an infinitely powerful $\widetilde{B}$ cannot determine from x's, y_1's, and y_2's which (s_1, s_2) was actually used.

P then obtains (s_1, s_2), (s_1^*, s_2^*) $(s_i \neq s_i^*)$ such that $v = a^{s_1} s_2^k \equiv a^{s_1^*} {s_2^*}^k \pmod n$, so $a^{(1/k)(s_1 - s_1^*)} \equiv s_2^*/s_2 \pmod n$. After repeating the above procedure, P obtains another (s'_1, s'_2), $(s_1'^*, s_2'^*)$ $(s'_i \neq s_i'^*)$ such that $a^{(1/k)(s'_1 - s_1'^*)} \equiv s_2'^*/s'_2 \pmod n$ with nonnegligible probability. If $\gcd(s_1 - s_1^*, s'_1 - s_1'^*) = 1$, then P can calculate $a^{1/k} \bmod n$. The probability that $\gcd(s_1 - s_1^*, s'_1 - s_1'^*) = 1$ is more than a constant, since $s_1^*, s_1'^*$ is selected randomly and s_1, s'_1 is independent from $s_1^*, s_1'^*$. Thus, the total success probability of P is nonnegligible.

This contradicts the intractability assumption of RSA inversion. □

Theorem 10. *Let $|k| = O(1)$. Identification scheme 2 is secure with sharp threshold security level $1/k$ if and only if RSA inversion is intractable.*

3.3 Identification Scheme as Secure as Factoring

In this subsection, we show a slight variant of the previous identification scheme (Identification scheme 2), which is as secure as factoring, while Identification scheme 2 is as secure as the inversion of the RSA function. The protocol of this variant (Identification scheme 3) is exactly same as Identification scheme 2. The only difference is that the value of k is selected so that $\gcd(k, \phi(n)) = 2$ and $k/2$ is prime, while $\gcd(k, \phi(n)) = 1$ and k is prime in Identification scheme 2.

Definition 11. Factoring is (nonuniformly) intractable, if any family of boolean circuits, which, given properly chosen (n) in the same distribution as the output of key generator G, can factor n with nonnegligible probability, must grow in a rate faster than any polynomial in the size of the input, $|n|$.

Theorem 12. *Identification scheme 3 is secure if and only if factoring is intractable.*

Theorem 13. *Let* $|k| = O(1)$. *Identification scheme 3 is secure with sharp threshold security level* $1/k$ *if and only if factoring is intractable.*

4 Generalization to Random-Self-Reducible Problems

This section shows that any random self-reducible problem [TW] leads to provably secure and three-move identification.

Definition 14. Let $\mathcal{N}$ be a countable infinite set. For any $N \in \mathcal{N}$, let $|N|$ denote the length of a suitable representation of N, and denote the problem size. For any $N \in \mathcal{N}$, let X_N, Y_N be finite sets, and $R_N \subseteq X_N \times Y_N$ be a relation. Let

$$domR_N = \{x \in X_N \mid (x, y) \in R_N \text{ for some } y \in Y_N\}$$

denote the *domain* of R_N,

$$R_N(x) = \{y \mid (x, y) \in R_N\}$$

the *image* of $x \in X_N$.

R is *random self-reducible (RSR)* if and only if there is a polynomial time algorithm A that, given any inputs $N \in \mathcal{N}$, $x \in domR_N$, and a source $r \in \{0,1\}^\omega$, outputs $x' = A(N, x, r) \in domR_N$ satisfying the following seven properties.

1. If r is randomly and uniformly chosen on $\{0,1\}^\omega$, then x' is uniformly distributed over $domR_N$.
2. There is a polynomial time algorithm that, given N, x, r, and any $y' \in R_N(x')$, outputs $y \in R_N(x)$.
3. There is a polynomial time algorithm that, given N, x, r, and any $y \in R_N(x)$, outputs some $y' \in R_N(x')$. If, in addition, the bits of r is random, uniform, and independent, then y' is uniformly distributed over $R_N(x')$.
4. There is an expected polynomial time algorithm that, given N, x', and y', determines whether $(x', y') \in R_N$.
5. There is an expected polynomial time algorithm that, given N, outputs random pairs $(x', y') \in R_N$ with x' uniformly distributed over $domR_N$ and y' uniformly distributed over $R_N(x')$.

6. There is an expected polynomial time algorithm that, given N, x_0, x_1, x_2, r_1, r_2 satisfying $x_i = A(N, x_0, r_i)$ $(i = 1, 2)$, outputs r^* satisfying $x_2 = A(N, x_1, r^*)$.
7. There is an expected polynomial time algorithm that, given N, x_1, x_2, y_1, y_2 satisfying $(x_i, y_i) \in R_N$ $(i = 1, 2)$, outputs r^* satisfying $x_2 = A(N, x_1, r^*)$.

Next we construct a three-move identification scheme based on random self-reducible problem R (Identification scheme 4).

A user generates a public key (N, a, t, v) and a secret key (s_i) $(i = 0$ or $1)$ and publishes the public key.

- A random bit $i \in \{0, 1\}$, $N \in \mathcal{N}$, $a \in domR_N$, and an integer $t = O(|N|)$.
- When $i = 0$, random bits $s_0 \in \{0, 1\}^\omega$, and $v = A(N, a, s_0)$.
- When $i = 1$, a random pair $(v, s_1) \in R_N$.

Protocol: Identification scheme 4

Step 1 A generates random bits $y_{j0} \in \{0, 1\}^\omega$, and $x_{j0} = A(N, a, y_{j0})$, $(j = 1, \ldots, t)$. A also generates random pairs $(x_{j1}, y_{j1}) \in R_N$, $(j = 1, \ldots, t)$. A sets $x_j = (x_{jb_j}, x_{j(1-b_j)})$ with a random bit $b_j \in \{0, 1\}$, and sends $(x_1, x_2, \ldots, x_t)$ to B.

Step 2 B sends random bits $(e_1, \ldots, e_t)$ to A.

Step 3 A sends $(z_1, z_2, \ldots, z_t)$ to B. Here, if $e_j = 0$, $z_j = (y_{j0}, y_{j1})$. If $e_j = 1$ and $i = 0$, then $z_j = r_0$ such that $x_{j0} = A(N, v, r_0)$ (r_0 can be computed from property 6). If $e_j = 1$ and $i = 1$, then $z_j = r_1$ such that $x_{j1} = A(N, v, r_1)$ (r_1 can be computed from property 7).

Step 4 B checks the validity of the messages received from A.

Definition 15. The random self-reducible problem R is (nonuniformly) intractable, if any family of boolean circuits, which, given properly chosen (N, a) in the same distribution as the output of key generator G, can compute α satisfying $(a, \alpha) \in R_N$ with nonnegligible probability, must grow at a rate faster than any polynomial in the size of the input, $|p|$.

Theorem 16. *Identification scheme 4 is secure if and only if the random self-reducible problem R is intractable.*

The basic techniques to prove this theorem are similar to those shown in Section 3. Scheme 4 is much less efficient than the schemes in Section 3, since the schemes in Section 3 are Type 2 of the parallel versions (see Section 1), while this scheme is Type 1.

Because of space limitations, we omit the proof of this theorem in this extended abstract.

5 Variants of the Proposed Identification Schemes

5.1 Identification Scheme as Secure as the Discrete Logarithm and Factoring Simultaneously

This subsection introduces a variant of Identification scheme 1 (Identification scheme 5) using the idea of the Brickell-McCurley scheme [BM1]. This variant is proven to be as secure as the difficulty of solving both the discrete logarithm and the specific factoring problem (or the finding order problem) simultaneously.

In this identification scheme, a user generates a public key (p, g_1, g_2, v) and secret key (s_1, s_2) and publishes the public key. (q, w) can be discarded after publishing the public key. (p, g_1, g_2) can be a system parameter, which is commonly used by all users.

- primes p, q and w such that $qw|p-1$ (e.g., $q \geq 2^{140}$, $p \geq 2^{512}$, and $qw \geq 2^{512}$).
- g_1 and g_2 of order q in the group Z_p^*.
- random numbers s_1, s_2 in Z_{p-1}.
- $v = g_1^{-s_1} g_2^{-s_2} \bmod p$.

We now describe our new identification scheme (Identification scheme 5).

Protocol: Identification scheme 5

Step 1 A picks random numbers $r_1, r_2 \in Z_{p-1}$, computes

$$x = g_1^{r_1} g_2^{r_2} \bmod p,$$

and sends x to B.

Step 2 B sends random numbers $e \in Z_{2^t}$ to A.

Step 3 A sends to B (y_1, y_2) such that

$$y_1 = r_1 + e x_1 \bmod p-1, \quad \text{and} \quad y_2 = r_2 + e x_2 \bmod p-1.$$

Step 4 B checks that

$$x = g_1^{y_1} g_2^{y_2} v^e \bmod p.$$

Definition 17. The finding order problem is (nonuniformly) intractable, if any family of boolean circuits, which, given properly chosen (p, g_1) in the same distribution as the output of key generator G, can compute the order of g_1 in the group Z_p^* with nonnegligible probability, must grow at a rate faster than any polynomial in the size of the input, $|p|$.

Remark This problem is more tractable than the factoring problem (Definition 11), since if there exists an polynomial time algorithm to solve the factoring problem, then the finding order problem can be solved by factoring $p-1$. So, the finding order problem can be considered a subproblem of the factoring problem.

Theorem 18. *Identification scheme 5 is secure if and only if the problem to solve both the discrete logarithm and the finding order problem simultaneously is intractable.*

5.2 Identity-Based and Certification-Based Variants

There are two methods of eliminating the public key directory from the conventional public key schemes: one is the identity-based method and the other is the certification-based method.

In the certification-based method, a trusted center (key authentication center, or certification authority) publishes its public key and gives a user A its signature S for the pair of identity Id_A and public key PK_A of A. The user A sends (Id_A, PK_A, S) to the verifier, who checks the validity of PK_A by verifying the trusted center's signature S for (Id_A, PK_A) in place of retrieving PK_A through Id_A from the public key directory.

In the identity-based method, proposed by Shamir [Sha] and independently by Okamoto [Oka], the public key is replaced by the identity related value of a user.

The difference between the certification-based method and identity-based method is as follows:

- Any public-key system can be converted into the certification-based variant by the same technique, while each public-key system needs a peculiar technique to convert to the identity-based variant.
- The trusted center of the certification-based method does not know each user's secret key, while the trusted center of the identity-based method generates and knows each user's secret key.
- The size of the public key that a user keeps and sends to the verifier in the certification-based method is longer than that in the identity-based method.

In this extended abstract, only two examples, identity-based variants of Identification schemes 1 and 2, are introduced briefly. In particular, we show a new construction technique to realize the identity-based variant of a scheme which is based on the discrete logarithm (e.g., Identification scheme 1), although the identity-based scheme based on the discrete logarithm is usually difficult to construct. Our technique is similar to Beth's idea [Bet], but, ours seems to be more natural, since we use the digital signature corresponding to the identification (Section 6), while the ElGamal scheme is used in [Bet]. (Our technique can be also applied to the Schnorr scheme: See Appendix B.)

Identity-Based Variant of Identification scheme 1 A trusted center T (or key authentication center) generates a public key (p, q, g_1, g_2, t, v_T) and its secret key (s_{T1}, s_{T2}), and publishes the public key as a system parameter. T generates T's digital signature, (e_A, y_{A1}, y_{A2}), of A's identity, Id_A, by using its secret key. So, $e_A = h((g_1^{y_{A1}} g_2^{y_{A2}} v_T^{e_A} \bmod p), Id_A)$ (see Section 6). T gives A A's secret key (s_{A1}, s_{A2}) and e_A, where $(s_{A1}, s_{A2}) = (q - y_{A1}, q - y_{A2})$. Then A generates A's public key $v_A = g_1^{-s_{A1}} g_2^{-s_{A2}} \bmod p$ from the secrete key given by T.

In this identity-based identification protocol, A first sends (Id_A, v_A, e_A) to verifier B along with x (same as x in the first step of Identification scheme 1). B checks the validity of Id_A and v_A by checking whether $e_A = h((v_A v_T^{e_A}$ mod

$p), Id_A)$ holds or not. If the check passes, the remainig protocol is the same as Identification scheme 1 (or B sends A e, A sends B (y_1, y_2), and B checks it). So, B does not need to retrieve v_A from the public-key directory. Here, the communication overhead except (Id_A, v_A) is just e_A, whose size is much smaller than those of v_A and x.

Identity-Based Variant of Identification scheme 2 A trusted center (or key authentication center) generates a public key (a, k, n) and gives user A its secret key (s_{A1}, s_{A2}), where $Id_A = a^{-s_{A1}} s_{A2}^{-k} \bmod n$. (First $s_{A1} \in Z_k$ is randomly determined, then $s_{A2} = (Id_A a^{s_{A1}})^{-1/k} \bmod n$ is calculated. Id_A can be replaced by $h(Id_A)$ with a one-way function.)

In this identity-based identification protocol, Id_A is used in place of v in Identification scheme 2. In a manner similar to the above-mentioned identity-based protocol, Id_A is sent to B along with x in the first step and the remaining part is the same as Identification scheme 2. So, B does not need to retrieve v from the public-key directory.

5.3 Elliptic Curve Version

Some techniques to construct cryptosystems based on the elliptic curve logarithm over a finite field [HMV, Kob1, Kob2, Mil, Miy] can be straightfowardly applied to our Identification scheme 1.

The elliptic curve variant of Identification scheme 1 has the significant property that three-move practical identification is proven to be secure assuming the intractability of the (non-supersingular) elliptic curve logarithms against which only exponential-time attacks have been reported so far [MOV, Kob2].

6 Signature Schemes

This section describes digital signature schemes converted from the identification schemes given in the previous sections. We also prove the security (existentially unforgeable against adaptive chosen message attacks [GMRi]) of our new signature schemes assuming one reasonable assumption about the one-way hash function (correlation-free one-way hash function) as well as a primitive assumption.

Since this conversion [FiS] is very simple, in this extended abstract, we only show one example (Signature scheme 1) based on Identification scheme 1. Other signature schemes (Signature schemes 2 to 5, and others) can be realized in the same way based on Identification schemes 2 to 5, and the variants described in subsections 5.2 and 5.3.

6.1 Signature Scheme Based on Identification Scheme 1

Signature scheme 1 is almost as efficient as the Schnorr signature scheme and DSA (see Section 7), while the security [GMRi] assumption of our scheme is weaker and more reasonable than those of the Schnorr signature scheme and DSA.

A public key (p, q, g_1, g_2, t, v) and secret key (s_1, s_2) of each user are determined in the same manner as Identification scheme 1. h is a one-way hash function.

We now describe our new signature scheme (Signature scheme 1) by which party A (the signer) generates a signature (e, y_1, y_2) of a message m, and sends (m, e, y_1, y_2) to B (the verifier).

Protocol: Signature scheme 1

Step 1 A (signer) picks random numbers $r_1, r_2 \in \boldsymbol{Z}_q$, computes $x = g_1^{r_1} g_2^{r_2}$ mod$p$. A computes $e = h(x, m) \in \boldsymbol{Z}_{2^t}$ and $(y_1,\ y_2)$ such that $y_1 = r_1 + es_1 \bmod q$, and $y_2 = r_2 + es_2 \bmod q$.

Step 2 A sends to B (e, y_1, y_2) along with message m.

Step 3 B computes $x = g_1^{y_1} g_2^{y_2} v^e \bmod p$, and checks that $e = h(x, m)$.

6.2 Security of Signature Schemes

In this subsection, we discuss the security of our signature schemes in the sense of "existentially unforgeable against adaptive chosen message attacks" defined by [GMRi]. Fiat and Shamir [FiS] have shown that the existence of an "ideal random function" as well as factoring assumption is sufficient to prove the security of the Fiat-Shamir signature scheme. However, their assumption, the existence of an ideal random function, can never be realized in the real world, and to realize the "pseudo-random function" [GGM] as a common function requires a tamper-free device.

In this paper, we clarify a reasonable assumption to prove the security of the Fiat-Shamir type signature schemes. We introduce a new class of one-way hash functions, *correlation-free one-way hash functions*, and show that the existence of a "correlation-free one-way hash function", as well as a primitive assumption, is sufficient to prove the security of our schemes. Although the existence of a correlation-free one-way hash function seems to be a stronger assumption than those of universal one-way hash function, claw-free pair of functions and collision-free hash function, we highly believe that carefully designed practical one-way hash functions such as MD5 and SHA are correlation-free one-way hash functions with any number theoretic predicate.

Definition 19. A family of *correlation-free one-way hash functions* with F is a set of hash functions, $H = \{H_n\}$ (H_n is a subset of H with security parameter n), with the following properties:

- **Poly-time indexing:** Each function in H_n has a unique n bit index, σ_n, associated with it: $H_n = \{h_{\sigma_n} \mid \sigma_n \in \{0,1\}^n, h_{\sigma_n} : \{0,1\}^{p(n)} \times \{0,1\}^{s(n)} \rightarrow \{0,1\}^{q(n)}\}$, where $p(n)$, $s(n)$, and $q(n)$ are polynomial in n. There is a probabilistic polynomial time algorithm, which, on input n, selects uniformly and randomly σ_n in $\{\sigma_n\}$.

- **Poly-time evaluation:** There exists a polynomial time algorithm that (for all $n \geq 1$), upon input of an index σ_n and an argument $(x, m) \in \{0,1\}^{p(n)} \times \{0,1\}^{s(n)}$, computes $h_{\sigma_n}(x, m)$.
- **Correlation-freeness:** Let $F = \{F_n \mid F_n = \{f_{\delta_n}\}\}$ be a poly-time indexing (δ_n) and poly-time evaluation predicate family such that $f_{\delta_n} : \{0,1\}^{p(n)} \times \{0,1\}^{q(n)} \times \{0,1\}^{r(n)} \to \{0,1\}$, where $r(n)$ is polynomial in n. Suppose that any family of boolean circuits, which, given δ_n, can compute x and (e_i, y_i) $(i = 1, \ldots, t(n))$ ($t(n)$ is polynomial in n) with nonnegligible probability such that $f_{\delta_n}(x, e_i, y_i) = 1$, must grow at a rate faster than any polynomial in n. Then, any family of boolean circuits, which, given σ_n, and δ_n, can compute (x, e, y, m) with nonnegligible probability such that $h_{\sigma_n}(x, m) = e$ and $f_{\delta_n}(x, e, y) = 1$, must grow at a rate faster than any polynomial in n.
- **One-wayness:** Any family of boolean circuits, which, given (x, m), can compute m' $(m' \neq m)$ with nonnegligible probability such that $h_{\sigma_n}(x, m') = h_{\sigma_n}(x, m)$, must grow at a rate faster than any polynomial in n.

Theorem 20. *Signature scheme 1 is existentially unforgeable against any adaptive chosen message attacks if the discrete logarithm problem is intractable and h is a correlation-free one-way hash function with $F = \{f_{(g_1, g_2, p, v)}\}$, where $f_{(g_1, g_2, p, v)}(x, e, (y_1, y_2)) = 1$ if and only if $x = g_1^{y_1} g_2^{y_2} v^e \bmod p$ holds.*

Sketch of Proof:

Assume that there exists an adaptive chosen message attacker, P, to Signature scheme 1. We also assume that the discrete logarithm problem is intractable. Then we will show a contradiction with the assumption that h is a correlation-free one-way hash function with $F = \{f_{(g_1, g_2, p, v)}\}$.

First, assume that P can find $(x, e, y_1, y_2, e', y_1', y_2')$ $(e \neq e')$ with nonnegligible probability such that $x = g_1^{y_1} g_2^{y_2} v^e \bmod p$ and $x = g_1^{y_1'} g_2^{y_2'} v^{e'} \bmod p$, after adaptive chosen message attacks. Since, given (g_1, g_2, p), P can exactly simulate the valid signer by generating his/her secret key (s_1, s_2) and following signer's valid procedure, P can calculate the discrete logarithm α $(g_2 = g_1^{\alpha} \bmod p)$ by the technique described in the proof of Theorem 6. This contradicts the intractability assumption of the discrete logarithm problem. Therefore, P can find $(x, e, y_1, y_2, e', y_1', y_2')$ $(e \neq e')$ with negligible probability.

On the other hand, from the assumption that P is an adaptive chosen message attacker, P can find (x, e, y_1, y_2, m) with nonnegligible probability such that $h(x, m) = e$ and $x = g_1^{y_1} g_2^{y_2} v^e \bmod p$. This contradicts the assumption that h is a correlation-free hash function with $F = \{f_{(g_1, g_2, p, v)}\}$.

Thus, any attacker P cannot find a valid signature message (x, e, y_1, y_2, m) with nonnegligible probability after adaptive chosen message attacks. □

6.3 Two-Move and One-Move Identification Schemes

In this subsection, we briefly introduce two-move and one-move identification schemes by using secure signature schemes above, which are almost as efficient as the proposed three-move identification schemes.

Two-move secure identification scheme can be trivially constructed using a secure (existentially unforgeable against any adaptive chosen message attacks) signature scheme as follows: First, verifier B sends a random message x to prover A, then A generates and sends A's signature of message x to B, finally B checks the validity of A's signature.

We can easily convert a two-move identification scheme into a one-move identification by changing challenge message x into time-stamp t, which both A and B share. That is, first A sends A's signature of message t to B, then B checks it.

6.4 Multi-Signature and Blind Signature

The multi-signature and blind signature schemes of our proposed signature schemes (Signature schemes 1 to 5 and the variants) can be constructed. The multi-signature schemes are constructed in a manner similar to [OhO2], and the blind signature schemes are constructed based on the idea shown in [OkO].

Blind Signature for Signature Scheme 1 Here, we present only one example of the blind signature schemes, based on Signature scheme 1. The other blind signature schemes are constructed in the same way using the idea shown in [OkO]. (The blind signature scheme based on the Schnorr scheme is shown in Appendix B.)

In the blind signature scheme, which was originally proposed by Chaum [Cha] based on the RSA scheme, a client, Bob, generates a blinded message $b(m)$ from a message m, and sends $b(m)$ to a blind signer, Alice. She generates her signature $s_A(b(m))$ of $b(m)$, and sends it to Bob. He calculates Alice's signature $s_A(m)$ of message m from $s(b(m))$. Here, Alice has no information of m, and Bob has no information of Alice's secret key.

We now describe our blind signature scheme based on Signature scheme 1. Alice's public key is (p, q, g_1, g_2, t, v) and her secret key is (s_1, s_2), which are those of Signature scheme 1.

Protocol: Blind signature based on Signature scheme 1

Step 1 Alice (blind signer) picks random numbers $r_1, r_2 \in Z_q$, computes $x = g_1^{r_1} g_2^{r_2} \bmod p$, and sends x to Bob (client).

Step 2 Bob picks random numbers $d, u_1, u_2 \in Z_q$, and computes

$$x^* = g_1^{u_1} g_2^{u_2} v^{-d} x \bmod p, \quad e^* = h(x^*, m), \quad e = e^* + d \bmod q.$$

Bob sends e to Alice. Here, m is a message to be signed.

Step 3 Alice computes (y_1, y_2) such that $y_1 = r_1 + es_1 \bmod q$, and $y_2 = r_2 + es_2 \bmod q$, and sends (y_1, y_2) to Bob.

Step 4 Bob computes $y_1^* = y_1 + u_1 \bmod q$, $y_2^* = y_2 + u_2 \bmod q$. (e^*, y_1^*, y_2^*) is Alice's signature of message m.

Note: e is distributed on Z_q, while e^* is distributed on Z_{2^t}. The difference is no problem in the blind signature scheme, since even an infinite power attacker cannot find any linkage between e and e^*.

7 Performance

This section compares the computation amount of our schemes against those of the previous practical schemes in the light of the required number of modular multiplications, and also compare the key and signature lengths.

We assume that moduli p and q for our scheme 1, Schnorr are 512 bits and 140 bits respectively, p and q for DSA are 512 bits and 160 bits, and the modulus n for our scheme 3, Guillou-Quisquater (GQ), Ohta-Okamoto (OO) and Feige-Fiat-Shamir (FFS) is 512 bits. The security parameter for the identification schemes is assumed to be 20, or e (the challenge from the verifier) is 20 bits. The security parameter for the signature schemes is assumed to be 128, or e (the output of the hash function of x and a message) is 128 bits, since the output size of many typical hash functions such as MD5 is 128 bits. We also assume that the parameters for Feige-Fiat-Shamir are $k = |e|$ and $t = 1$.

Here, we estimate the performance of unsophisticated implementations, since the purpose of this comparison is to relatively compare some schemes with the same primitive problem (e.g., our scheme 1 and Schnorr), and many sophisticated techniques (e.g., [Mon, BGMW]) can be fairly evenly applied to the schemes with the same primitive problem. We assume the standard binary method and the extended binary method (4.6.3 ex.27 in [Kun]) for the modular exponentiation.

Table 1. Comparison of Identification Schemes

	Proposed Scheme 1	*Schnorr*	*Proposed Scheme 3*	*GQ*	*OO*	*FFS*
Provably secure?	Yes	No	Yes	No	Yes	Yes
Primitive problem	Disc.log.	Disc.log.	Fact.	RSA	Fact.	Fact.
ID-based variant	Possible	Possible	Possible	Possible	Hard	Possible
System parameter size (bits)	1676	1164	532	20	20	0
Public key size (bits)	512	512	1024	1024	1024	10240
Secret key size (bits)	280	140	532	512	512	10240
Communication amount (bits)	812	672	1064	1044	1044	1044
Preprocessing (Prover) (# of 512-bit modular multiplications)	245	210	35	30	30	1
On-line processing (Prover) (# of 512-bit modular multiplications)	almost 0	almost 0	32	31	31	10
On-line Processing (Verifier) (# of 512-bit modular multiplications)	248	210	38	35	35	11

Table 2. Comparison of Signature Schemes

	Proposed Scheme 1	*Schnorr*	*DSA*	*Proposed Scheme 3*	*GQ*	*OO*	*FFS*
Assumption	Weak	Strong	Strong	Weak	Strong	Weak	Weak
Primitive problem	Disc.log.	Disc.log.	Disc.log.	Fact.	RSA	Fact.	Fact.
ID-based variant	Possible	Possible	Possible	Possible	Possible	Hard	Possible
Multi-signature	Possible	Possible	Hard	Possible	Possible	Possible	Possible
Blind signature	Possible	Possible	Hard	Possible	Possible	Possible	Possible
System parameter size (bits)	1676	1164	1164	640	128	128	0
Public key size (bits)	512	512	512	1024	1024	1024	66048
Secret key size (bits)	280	140	160	640	512	512	65536
Signature size (bits)	408	268	320	768	640	640	640
Preprocessing for signing (# of 512-bit modular multiplications)	245	210	237	224	192	192	1
Signing (# of 512-bit modular multiplications)	almost 0	almost 0	almost 0	194	193	193	65
Verifying (# of 512-bit modular multiplications)	261	242	277	240	224	224	66

Acknowledgments

The author would like to thank Kouichi Sakurai for his valuable comments and suggestions especially on the relationship between "no useful information transfer" and "witness hiding". He would also like to thank an anonymous referee for his/her useful comments on the preliminary manuscript.

References

[Bet] T.Beth, "Efficient Zero-Knowledge Identification Scheme for Smart Cards," Proceedings of Eurocrypt '88, LNCS 330, Springer-Verlag, pp.77-86 (1988).

[BGMW] E.F.Brickell, D.M.Gordon, K.S.McCurley, and D.Wilson, "Fast Exponentiation with Precomputation", to appear in the Proceedings of Eurocrypt'92.

[BM1] E.F.Brickell, and K.S.McCurley, "An Interactive Identification Scheme Based on Discrete Logarithms and Factoring," Journal of Cryptology, Vol.5, No.1, pp.29-39 (1992).

[BM2] E.F.Brickell, and K.S.McCurley, "Interactive Identification and Digital Signatures," AT&T Technical Journal, pp.73-86, November/December (1991).

[BMO1] M.Bellare, S.Micali and R.Ostrovsky, "Perfect Zero-Knowledge in Constant Rounds," Proceedings of STOC, pp.482-493 (1990).

[BMO2] M.Bellare, S.Micali and R.Ostrovsky, "The (True) Complexity of Statistical Zero-Knowledge," Proceedings of STOC, pp.494-502 (1990).

[Cha] D.Chaum, "Security without Identification: Transaction Systems to Make Big Brother Obsolete," Comm. of the ACM, 28, 10, pp.1030-1044 (1985).

[CD] L.Chen, I.Damgård, "Security Bounds for Parallel Versions of Identification Protocols," Manuscript (1992).

[FeS1] U.Feige and A.Shamir, "Witness Indistinguishable and Witness Hiding Protocols," Proceedings of STOC, pp.416-426 (1990).

[FeS2] U.Feige and A.Shamir, "Zero Knowledge Proofs of Knowledge in Two Rounds," Proceedings of Crypto'89, LNCS 435, Springer-Verlag, pp.526-544 (1990).

[FFS] U.Feige, A.Fiat and A.Shamir, "Zero Knowledge Proofs of Identity," Proceedings of STOC, pp.210-217 (1987).

[FiS] A.Fiat and A.Shamir, "How to Prove Yourself: Practical Solutions to Identification and Signature Problems", Proceedings of CRYPTO '86, LNCS 263, Springer-Verlag, pp.186–194 (1987).

[GGM] O.Goldreich, S.Goldwasser, and S.Micali, "How to Construct Random Functions," Journal of the ACM, Vol.33, No.4 (1986).

[GK] O.Goldreich and H.Krawczyk "On the Composition of Zero-Knowledge Proof Systems," Proceedings of ICALP, LNCS 443, Springer-Verlag, pp.268-282 (1990).

[GMRa] S.Goldwasser, S.Micali and C.Rackoff, "The Knowledge Complexity of Interactive Proofs," SIAM J. Comput., 18, 1, pp.186-208 (1989).

[GMRi] S.Goldwasser, S.Micali and R.Rivest, "A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks," SIAM J. Comput., 17, 2, pp.281–308 (1988).

[GQ] L.S.Guillou, and J.J.Quisquater, "A Practical Zero-Knowledge Protocol Fitted to Security Microprocessors Minimizing both Transmission and Memory," Proceedings of Eurocrypt '88, LNCS 330, Springer-Verlag, pp.123-128 (1988).

[HMV] G.Harper, A.J.Menezes, S.A.Vanstone, "Public-Key Cryptosystems with Very Small Key Length", to appear in the Proceedings of Eurocrypt'92.

[Kobl] N.Koblitz, *A Course in Number Theory and Cryptography*, Berlin: Springer-Verlag, (1987).

[Kob2] N.Koblitz, "CM-Curves with Good Cryptographic Properties," Proceedings of Crypto '91 (1992).

[Kun] D.E.Knuth, *The Art of Computer Programming*, Vol.2, 2nd Ed. Addison-Wesley (1981).

[Mil] V.Miller, "Uses of Elliptic Curves in Cryptography," Proceedings of Crypto '85, LNCS 218, Springer-Verlag, pp.417-426 (1986).

[Miy] A.Miyaji, " On Ordinary Elliptic Curve Cryptosystems," to appear in the Proceedings of Asiacrypt'91, LNCS, Springer-Verlag.

[Mon] P.L.Montgomery, "Modular Multiplication without Trial Division," Math. of Computation, Vol.44, pp.519–521 (1985).

[MOV] A.J.Menezes, T.Okamoto, S.A.Vanstone, "Reducing Elliptic Curve Logarithms to Logarithms in a Finite Field", Proceedings of STOC, pp.80-89 (1991).

[OhO1] K.Ohta, and T.Okamoto, "A Modification of the Fiat-Shamir Scheme," Proceedings of Crypto '88, LNCS 403, Springer-Verlag, pp.232-243 (1990).

[OhO2] K.Ohta, and T.Okamoto, "A Digital Multisignature Scheme Based on the Fiat-Shamir Scheme," to appear in the Proceedings of Asiacrypt'91.

[Oka] T.Okamoto, "A Single Public-Key Authentication Scheme for Multiple Users," *Systems and Computers in Japan, 18, 10*, pp.14-24 (1987), Previous version, Technical Report of IECE Japan, IN83-92 (1984).

[OkO] T.Okamoto, and K.Ohta, "Divertible Zero-Knowledge Interactive Proofs and Commutative Random Self-Reducible," Proceedings of Eurocrypt '89, LNCS 434, Springer-Verlag, pp.134–149 (1990).

[PH] S.C.Pohlig, and M.E.Hellman, "An Improved Algorithm for Computing Logarithmsover GF(p) and Its Cryptographic Significance," IEEE Trans. Inform. Theory, 24, pp.106–110 (1978)

[RSA] R.Rivest, A.Shamir and L.Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems", Communications of the ACM, Vol.21, No.2, pp.120-126 (1978).

[Sch] C.P. Schnorr, "Efficient Signature Generation by Smart Cards," Journal of Cryptology, Vol.4, No.3, pp.161-174 (1991).

[Sha] A.Shamir, "Identity-Based Cryptosystems and Signature Scheme," Proceedings of Crypto '84, LNCS 196, Springer-Verlag, pp.47–53 (1986).

[SI] K.Sakurai, and T.Itoh, "On the Discrepancy between Serial and Parallel of Zero-Knowledge Protocols," These proceedings.

[TW] M.Tompa and H.Woll, "Random Self-Reducibility and Zero Knowledge Interactive Proofs of Possession of Information," Proceedings of FOCS, pp.472-482 (1987).

Appendix A

In this appendix, we introduce a variant of "no useful information transfer" [FFS] given by Ohta and Okamoto [OhO1], called "no transferable information with (sharp threshold) security level".

Definition 21. An identification scheme (A, B) is *secure with security level* ρ if

1. $(\overline{A}, \overline{B})$ succeeds with overwhelming probability.
2. There is no coalition of $\tilde{A}, \tilde{B}$ with the property that, after a polynomial number of executions of $(\overline{A}, \tilde{B})$ and relaying a transcript of the communication to $\tilde{A}$, it is possible to execute $(\tilde{A}, \overline{B})$ with $c \cdot \rho$ probability of success, where $c = (1 + 1/|n|^d)$ and d is an arbitrary constant. The probability is taken over the distribution of the public key and the secret key as well as the coin tosses of $\overline{A}$, $\tilde{B}$, $\tilde{A}$, and $\overline{B}$, up to the time of the attempted impersonation.

Definition 22. An identification scheme (A, B) is *secure with sharp threshold security level* ρ if

1. (A, B) is secure with security level ρ.
2. There exists $\tilde{A}$ such that it is possible to execute $(\tilde{A}, \overline{B})$ with ρ probability of success.

Appendix B

In this appendix, we introduce the identity-based variant and blind signature sceheme of the Schnorr scheme.

B.1 Identity-Based Variant of the Schnorr scheme

A trusted center T (or key authentication center) generates a public key (p, q, g, t, v_T) and its secret key s_T, and publishes the public key as a system parameter. T generates T's digital signature, (e_A, y_A), of A's identity, Id_A. T gives A A's secret key s_A and e_A, where $s_A = q - y_A$. Then A generates A's public key $v_A = g^{-s_A} \bmod p$ from the secrete key given by T.

In this identity-based identification protocol, A first sends (Id_A, v_A, e_A) to verifier B along with x B checks the validity of Id_A and v_A by checking whether $e_A = h((v_A v_T^{e_A} \bmod p), Id_A)$ holds or not. If the check passes, the remainig protocol is the same as the Schnorr scheme.

B.2 Blind Signature of the Schnorr scheme

Alice's public key is (p, q, g, t, v) and her secret key is s.

Protocol: Blind signature based on the Schnorr scheme

Step 1 Alice (blind signer) picks random number $r \in Z_q$, computes $x = g^r \bmod p$, and sends x to Bob (client).

Step 2 Bob picks random numbers $d, u \in Z_q$, computes

$$x^* = g^u v^{-d} x \bmod p, \quad e^* = h(x^*, m), \quad e = e^* + d \bmod q.$$

Bob sends e to Alice. Here, m is a message to be signed.

Step 3 Alice computes y such that $y = r + es \bmod q$, and sends y to Bob.

Step 4 Bob computes $y^* = y + u \bmod q$.
(e^*, y^*) is Alice's signature of message m.