A New Identification Scheme based on the Bilinear Diffie-Hellman Problem*

Myungsun Kim and Kwangjo Kim

International Research center for Information Security (IRIS)
Information and Communications Univ. (ICU)
58-4 Hwaamdong, Yuseong-gu, Daejon, 305-732, Korea.
{ms.kim, kkj}@icu.ac.kr

Abstract. We construct an interactive identification scheme based on the bilinear Diffie-Hellman problem and analyze its security. This scheme is practical in terms of key size, communication complexity, and availability of identity-variance provided that an algorithm of computing the Weil-pairing is feasible. We prove that this scheme is secure against active attacks as well as passive attacks if the bilinear Diffie-Hellman problem is intractable. Our proof is based on the fact that the computational Diffie-Hellman problem is hard in the additive group of points of an elliptic curve over a finite field, on the other hand, the decisional Diffie-Hellman problem is easy in the multiplicative group of the finite field mapped by a bilinear map. Finally, this scheme is compared with other identification schemes.

Keywords. Gap-problems, Identification scheme, Bilinear Diffie-Hellman problem, Weil-pairing

1 Introduction

It is well known that an *identification scheme* is a very important and useful cryptographic tool. The identification scheme is an interactive protocol where a prover, $\mathcal{P}$, tries to convince a verifier, $\mathcal{V}$, of his identity. Only $\mathcal{P}$ knows the secret value corresponding to his public one, and the secret value allows to convince $\mathcal{V}$ of his identity. If we replace "identity" by "authenticity" of messages, identification schemes are nearly equivalent to *signature schemes*. As mentioned by Fiat and Shamir [7] and Shoup [21], the distinction between identification and signature schemes is very subtle. Therefore, two types of schemes can be used interchangeably [7, 10, 17, 15].

Since Okamoto and Pointcheval [16] initiated the concept of the Gap-problems and proposed that a Gap Diffie-Hellman problem offers a signature scheme, several cryptographic schemes based on such variants of Diffie-Hellman assumption have been studied. Using the bilinear Diffie-Hellman problem as an instance of

* This work was supported by both Ministry of Information and Communication and Korea Information Security Agency, Korea, under project 2002-130

the Gap Diffie-Hellman problem, Boneh and Franklin [2] and Boneh *et al.* [3] suggested an efficient ID-based encryption scheme and a short signature scheme, respectively. These imply that the bilinear Diffie-Hellman problem provides identification schemes. To the best of our knowledge, there is no identification scheme based on the bilinear Diffie-Hellman problem published in the open literature.

In this paper, we construct a new identification scheme based on the Bilinear Diffie-Hellman problem, which is a typical instance of the Gap Diffie-Hellman problem and prove that this scheme is secure against passive and even active attacks if the bilinear Diffie-Hellman problem is intractable, which are the main contribution of this paper.

The rest of this paper is organized as follows: After describing several identification schemes in this section, we formally state our definition of security as well as basic tools used in our scheme in Section 2. Our basic identification scheme is presented based on the Bilinear Diffie-Hellman problem in Section 3. In Section 4 we give a proof of security for our scheme. In Section 5 we present a generalized model of our basic identification scheme. In what follows, we compare with other schemes in the light of performance, and finally, we end with concluding remarks.

1.1 Previous Works

Types of attacks.
In general, an identification scheme is said to be broken if an adversary succeeds in an impersonation attempt (making the verifier accept with non-negligible probability). We can classify the type of attacks according to the interaction allowed to the adversary before an impersonation attempt [21].

The weakest form of attack is a *passive attack*, where the adversary is not allowed to interact with the system at all before attempting an impersonation; the only available information to the adversary is the public key of the prover. Other attacks of intermediate level such as *eavesdropping attack* or *honest-verifier attack* are essentially equivalent to the passive attack.

The strongest form of attack is an *active attack*, in which the adversary is allowed to interact with $\mathcal{P}$ several times, posing as $\mathcal{V}$. We may consider active attacks as adaptive chosen ciphertext attacks. We should note that active attacks are quite feasible in practice.

Fiat-Shamir (FS) scheme. Fiat and Shamir [7] proposed the identification scheme based on the factorization problem. A key generation algorithm constructs a modulus n by multiplying two distinct random primes, chooses randomly an element $a \in \mathbb{Z}_n^*$, and sets $b = a^2$. The public key is $\langle b, n \rangle$, and the secret key is a.

The protocol repeats the following steps t times:

1. $\mathcal{P}$ chooses $r \in \mathbb{Z}_n^*$ at random, computes $x = r^2$, and sends x to $\mathcal{V}$.
2. $\mathcal{V}$ chooses $\epsilon \in \{0, 1\}$ at random, and sends ϵ to $\mathcal{P}$.

3. $\mathcal{P}$ computes $y = r \cdot a^{\epsilon}$ and sends y to $\mathcal{V}$; $\mathcal{V}$ accepts if $y^2 = x \cdot b^{\epsilon}$, and rejects otherwise.

The FS scheme is secure against active attacks if factorization is hard.

Feige-Fiat-Shamir (FFS) scheme. This scheme is also based on the factorization problem. A key generation algorithm chooses a modulus n as in the FS scheme. A secret key consists of a list $a_1, \ldots, a_l \in \mathbb{Z}_n^*$ chosen randomly, where l is a given constant, and the corresponding public key consists of $b_1, \ldots, b_l \in \mathbb{Z}_n^*$, where $b_i = a_i^2$ for $1 \leq i \leq l$.

The protocol executes the followings t times in parallel:

1. $\mathcal{P}$ chooses $r \in \mathbb{Z}_n^*$ at random, computes $x = r^2$, and sends x to $\mathcal{V}$.
2. $\mathcal{V}$ randomly chooses $\epsilon_1, \ldots, \epsilon_l \in \{0, 1\}$, sends $\epsilon_1, \ldots, \epsilon_l$ to $\mathcal{P}$.
3. $\mathcal{P}$ computes $y = r \prod_{j=1}^{l} a_j^{\epsilon_j}$ and sends y to $\mathcal{V}$; $\mathcal{V}$ accepts if $x = y^2 \prod_{j=1}^{l} b_j^{\epsilon_j}$, and rejects otherwise.

This scheme is also secure against active attacks if factorization is hard [6].

Other schemes. The Guillou-Quisquater (GQ) scheme is based on the RSA-inversion problem. Guillou [10] shows that this scheme is secure against passive attacks provided that factorization is hard. Ohta and Okamoto (OO) [17] present a modification of the FS scheme on the basis of the difficulty of extracting the L-th roots, and they prove that their scheme is as secure as the FS scheme. Okamoto [15] proposes three identification schemes. The first one is based on the discrete logarithm problem (DLP), the second on the RSA-problem, and the last one on the factorization problem. All of these schemes are proved to be secure against active attacks. Schnorr [20] also proposes identification schemes that are based on the factorization problem or DLP.

2 Definitions

A general approach of proving that an identification scheme is secure is to show that it exhibits a zero-knowledge proof of knowledge. However, the results of Goldreich and Krawczyk [8], together with the argument of Shoup [21] say that any efficient black box simulator for a three round, public coin system can be turned into a prover that succeeds with non-negligible probability.

In this paper, we make use of a computational reduction from solving a well-established problem to break the cryptosystem rather than zero-knowledge proof techniques. That is to say, the proving method is to use an adversary that breaks the cryptosystem to solve the computational Diffie-Hellman problem.

2.1 Notions of Security

We formally define a secure identification scheme, using the same notations as in [19] and [21].

If $A(\cdot)$ is a probabilistic algorithm, then for any input x, the notation A_x refers to the probability space that assigns to the string σ the probability space that A, on input x, outputs σ.

If S is a probability space, then $[S]$ denotes the set of elements in this space that occur with non-zero probability, and $\Pr_S[e]$ denotes the probability that S associates with the element e. If S is any probability space, then $x \leftarrow S$ denotes the algorithm which assigns to x an element randomly selected according to S.

The notation $\Pr[p(x_1, x_2 \ldots) | x_1 \leftarrow S_1; x_2 \leftarrow S_2; \ldots]$ denotes the probability that the predicate $p(x_1, x_2, \ldots)$ will be true after the ordered execution of the algorithms $x_1 \leftarrow S_1, x_2 \leftarrow S_2, \ldots$.

In addition, we use the same conventions in [6]:

1. $\bar{\mathcal{P}}$ represents an honest prover who follows its designated protocol, $\tilde{\mathcal{P}}$ does a polynomial-time cheater, and $\mathcal{P}$ acts as $\bar{\mathcal{P}}$ or $\tilde{\mathcal{P}}$.
2. $\bar{\mathcal{V}}$ represents a valid verifier who follows the designated protocol, $\tilde{\mathcal{P}}$ does an arbitrary polynomial-time algorithm which may try to extract additional information from $\mathcal{P}$, and $\mathcal{V}$ acts as $\bar{\mathcal{V}}$ or $\tilde{\mathcal{V}}$.
3. $(\mathcal{P}, \mathcal{V})$ represents the execution of the two party protocol where $\mathcal{P}$ is the prover and $\mathcal{V}$ is the verifier.

In general, an identification scheme $(\mathcal{G}, \mathcal{P}, \mathcal{V})$ consists of a probabilistic polynomial-time algorithm $\mathcal{G}$, and two probabilistic polynomial-time interactive algorithms $\mathcal{P}$ and $\mathcal{V}$ with the following properties [6, 21]:

1. The algorithm $\mathcal{G}$ is a *key generation algorithm*. It takes a string of the form 1^k as input, and outputs a pair of string (I, S). k is called a security parameter, I is called a *public key*, and S is called a *secret key*.
2. As input, $\mathcal{P}$ receives the pair (I, S) and $\mathcal{V}$ does I. After an interactive execution of $\mathcal{P}$ and $\mathcal{V}$, $\mathcal{V}$ outputs either 1 (indicating `"accept"`) or 0 (indicating `"reject"`). For given I and S, the output of $\mathcal{V}$ at the end of this interaction is a probability space which is denoted by $\langle \mathcal{P}(I, S), \mathcal{V}(I) \rangle$.
3. A valid prover should always be able to succeed in convincing the verifier. Formally speaking, for all k and for all $(I, S) \in [\mathcal{G}(1^k)]$, $\langle \mathcal{P}(I, S), \mathcal{V}(I) \rangle = 1$ with probability 1.

An *adversary* $(\tilde{\mathcal{P}}, \tilde{\mathcal{V}})$ is a pair of probabilistic polynomial-time interactive algorithms. For given key pair (I, S), we denote by $\langle \bar{\mathcal{P}}(I, S), \tilde{\mathcal{V}}(I) \rangle$ the string h output by $\tilde{\mathcal{V}}$ after interacting with $\bar{\mathcal{P}}$ several times. For given I and S, yet again $\langle \bar{\mathcal{P}}(I, S), \tilde{\mathcal{V}}(I) \rangle$ is a probability space. The string h (called a `"help string"`) is used as input to $\tilde{\mathcal{P}}$ who attempts to convince $\bar{\mathcal{V}}$. We denote by $\langle \tilde{\mathcal{P}}(h), \bar{\mathcal{V}}(I) \rangle$ the output of $\bar{\mathcal{V}}$ after interacting with $\tilde{\mathcal{P}}(h)$.

We adopt the definition of security against active attacks with respect to such adversaries from [21] as follows:

Definition 1. *An identification scheme* $(\mathcal{G}, \mathcal{P}, \mathcal{V})$ *is* secure against active attacks *if for all adversaries* $(\tilde{\mathcal{P}}, \tilde{\mathcal{V}})$*, for all constants* $c > 0$*, and for all sufficiently*

large k,

$$\Pr\left[\sigma = 1 \;\middle|\; \begin{array}{l}(I,S) \leftarrow \mathcal{G}(1^k);\\ h \leftarrow \langle \bar{\mathcal{P}}(I,S), \tilde{\mathcal{V}}(I)\rangle;\\ \sigma \leftarrow \langle \tilde{\mathcal{P}}(h), \bar{\mathcal{V}}(I)\rangle\end{array}\right] < k^{-c}.$$

2.2 Bilinear Diffie-Hellman Problem

The computational assumptions when constructing cryptographic schemes can mainly be classified into two types. One is the intractability of an inverting problem such as inverting the RSA function, and computing the Diffie-Hellman (DH) problem. The other is the intractability of a decision problem such as the decisional Diffie-Hellman problem.

In addition to these problems, Okamoto and Pointcheval [16] define a new class of problems, called the Gap-problem. Let $f : \{0,1\}^* \times \{0,1\}^* \rightarrow \{0,1\}$ and $R : \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^* \rightarrow \{0,1\}$ be any relation.

- The *inverting problem* of f is, given x, to compute any y such as $f(x,y) = 1$ if it exists, or to answer Fail.
- The *R-decision problem* of f is, given (x,y), to decide whether $R(f,x,y) = 1$ or not. Here y may be the null string, $\perp$.

Usually, it is accepted that there exists the gap of difficulty between two problems. Using this property, the Gap-problem can be defined as follows:

Definition 2. *The* R-gap problem *of f is to solve the inverting problem of f with the help of the oracle of the R-decision problem of f.*

Okamoto and Pointcheval [16] claim that the DH problems are the typical instance of the Gap problem. Since the inverting problem can be viewed as the computational problem, the computational Diffie-Hellman (C-DH) problem corresponds to the inverting one, and the decisional Diffie-Hellman (D-DH) problem does to the R-decision one. Here, we describe the Gap Diffie-Hellman (G-DH) problem. Let $\mathbb{G}$ be any group of prime order m.

- The C-DH problem: given a triple of $\mathbb{G}$ elements (g, g^a, g^b), find the element $C = g^{ab}$.
- The D-DH problem: given a quadruple of $\mathbb{G}$ elements (g, g^a, g^b, g^c), decide whether $c = ab \pmod q$ or not.
- The G-DH problem: given a triple of $\mathbb{G}$ elements (g, g^a, g^b), find the element $C = g^{ab}$ with the help of a D-DH oracle (which answers whether a given quadruple is a DH quadruple or not).

The Tate-pairing is given as a specific example that satisfies the property of the G-DH problem [16].

We focus on the bilinear Diffie-Hellman (B-DH) problem that is a variant of the C-DH problem, which is the underlying problem of our new design. Now we describe the B-DH problem and the security defined over this problem.

Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two cyclic groups of prime order m and let P be a generator of $\mathbb{G}_1$. Let $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ be a bilinear map which will be discussed in Section 2.3.

Definition 3. *The* B-DH problem *in* $(\mathbb{G}_1, \mathbb{G}_2, \hat{e})$ *is the following: given* (P, aP, bP, cP) *for some* $a, b, c \in \mathbb{Z}_m^*$*, compute* $v \in \mathbb{G}_2$ *such that* $v = \hat{e}(P, P)^{abc}$*.*

In practice, we make use of the Weil-pairing as the bilinear map. The security over groups in which the B-DH problem is defined is as follows.

Definition 4. *Let* $\mathbb{G}$ *be a cyclic group of a prime order with an arbitrary generator. For any polynomial-time probabilistic algorithm* $\mathcal{A}$*:*

- $\mathbb{G}$ *is said to be a* τ-breakable D-DH group *if the* D-DH *problem can be computed on* $\mathbb{G}$ *by* $\mathcal{A}$ *whose running time is bounded by* τ*.*
- $\mathcal{A}$ *is said to* (t, ϵ)-break C-DH *problem in* $\mathbb{G}$ *if the* C-DH *problem can be solved by* $\mathcal{A}$ *whose running time is bounded by* t*, the success probability* $\mathsf{Succ}^{\mathbb{G}}(\mathcal{A}) \geq \epsilon$*.*
- $\mathbb{G}$ *is said to be a* (τ, t, ϵ)-B-DH group *if it is a* τ*-breakable* D-DH *group and no algorithm* (t, ϵ)*-breaks* C-DH *on it.*

2.3 Weil-pairing

We can make use of any bilinear map on an elliptic curve to construct a group $\mathbb{G}$ in which the C-DH problem is intractable, but the D-DH problem is tractable [11, 2, 3]. In particular, we make use of the Weil-pairing among bilinear maps.

Let E be a elliptic curve over a base field K and let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two cyclic groups of order m for some large prime m. The *Weil pairing* [22, 12, 4, 2, 3] is defined by a bilinear map e,

$$e : \mathbb{G}_1 \times \mathbb{G}_1 \longrightarrow \mathbb{G}_2,$$

where $\mathbb{G}_1$ corresponds to the additive group of points of E/K, and $\mathbb{G}_2$ corresponds to the multiplicative group of an extension field $\overline{K}$ of K.

Let $P, Q \in \mathbb{G}_1$. The Weil pairing e has the following properties:

1. *Identity*: For all $P \in \mathbb{G}_1,\ e(P, P) = 1$.
2. *Alternation*: For all $P, Q \in \mathbb{G}_1,\ e(P, Q) = e(Q, P)^{-1}$.
3. *Bilinearity*: For all $P, Q, R \in \mathbb{G}_1,\ e(P+Q, R) = e(P, R) \cdot e(Q, R)$ and $e(P, Q+R) = e(P, Q) \cdot e(P, R)$.
4. *Non-degeneracy*: If $e(P, Q) = 1$ for all $Q \in \mathbb{G}_1$, then $P = \mathcal{O}$, where $\mathcal{O}$ is a point at infinity.

In addition to these properties, we have an efficient algorithm to compute $e(P, Q)$ for all $P, Q \in \mathbb{G}_1$ by [14]. In practice, in our basic scheme, we employ the *modified* Weil-pairing $\hat{e}(P, Q) = e(P, \phi(Q))$, where ϕ is an automorphism on the group of points of E [2, 3]. For more details, we can refer to [4], [2], and [12].

As noted in [2], the existence of the bilinear map e implies (1) DLP in $\mathbb{G}_1$ can be reduced to DLP in $\mathbb{G}_2$, (2) C-DH problem in $\mathbb{G}_1$ is still hard even though D-DH in $\mathbb{G}_1$ is easy [11].

3 Basic Identification Scheme

For a security parameter k, a pair of secret and public parameters is generated as follows:

Key generation.
On input k, the key generation algorithm $\mathcal{G}$ works as follows:

1. Generate two cyclic groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of order m for some large prime m and a bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$.
2. Generate an arbitrary generator $P \in \mathbb{G}_1$.
3. Choose randomly $a, b, c \in \mathbb{Z}_m^*$ and compute $v = \hat{e}(P, P)^{abc}$.
4. The public parameter is $\mathsf{Pub} = \langle \mathbb{G}_1, \mathbb{G}_2, P, aP, bP, cP, \hat{e}, v \rangle$, and the secret parameter is $\mathsf{Sec} = \langle a, b, c \rangle$. And then publish them.

Protocol actions between $\mathcal{P}$ and $\mathcal{V}$.
As is the case for other identification schemes, this scheme consists of several rounds. The protocol executes just once the following:

1. $\mathcal{P}$ chooses $r_1, r_2, r_3 \in \mathbb{Z}_m^*$ at random, computes $x = \hat{e}(P, P)^{r_1 r_2 r_3}$, $Q = r_1 r_2 r_3 P$, and sends $\langle x, Q \rangle$ to $\mathcal{V}$.
2. $\mathcal{V}$ picks $\omega \in \mathbb{Z}_m^*$ at random, and sends $R = \omega P$ to $\mathcal{P}$.
3. On receiving R, $\mathcal{P}$ sets $S = r_1 r_2 r_3 R$, computes $Y \in \mathbb{G}_1$ such that

$$Y = abcP + (a + b + c)S,$$

and sends it to $\mathcal{V}$; $\mathcal{V}$ accepts $\mathcal{P}$'s proof of identity if both $x = \hat{e}(P, Q)$ and $\hat{e}(Y, P) = v \cdot \hat{e}(aP + bP + cP, Q)^{\omega}$, and rejects otherwise.

4 Proof of Security

Our proof of security is based on the intractability of the B-DH problem. First, we formally describe this assumption as follows, called as it the bilinear Diffie-Hellman Intractability Assumption (B-DHIA):

Definition 5. *Let Z be a probability space consisting uniform distribution over all integers in $\mathbb{Z}_m^*$. Let G_1 be a probability space consisting the uniform distribution over all elements of the form $nP \neq \mathcal{O} \in \mathbb{G}_1$, where $n \in_{\mathcal{U}} Z$ and let G_2 be a probability space consisting of uniform distribution over all elements in $\mathbb{G}_2$.* B-DHIA *is defined as the following: Given $C = \hat{e}(P, P)^{abc} \in \mathbb{G}_2$, for all polynomial-time probabilistic algorithm $\mathcal{A}$, for all constant $c > 0$, and for all sufficiently large k,*

$$\Pr_{G_2}\left[C = C' \,\middle|\, \begin{array}{l} x \leftarrow Z, xP \in G_1; \\ y \leftarrow Z, yP \in G_1; \\ z \leftarrow Z, zP \in G_1; \\ C' \leftarrow \mathcal{A}(\hat{e}, xP, yP, zP) \end{array} \right] < k^{-c}.$$

Now we are ready to prove:

Theorem 1. *Under B-DHIA, the basic identification scheme on* (τ, t, ϵ)-B-DH *groups is secure against active attacks.*

Proof. As mentioned before, the basic way of proving this theorem is just to show that any adversary $\mathcal{I}$ who succeeds in impersonating with non-negligible probability can be reduced into a polynomial-time probabilistic algorithm $\mathcal{A}$ that (τ, t, ϵ)-breaks C-DH problem with non-negligible probability. This will be proved in Lemma 2.

First to construct such an adversary $\mathcal{I} = (\tilde{\mathcal{P}}, \tilde{\mathcal{V}})$, we consider the adversary with the following polynomials [21]:

- $T_{\mathcal{V}}(k)$: a time bound required for $\tilde{\mathcal{V}}$ to run the protocol once with $\bar{\mathcal{P}}$ including $\bar{\mathcal{P}}$'s computing time.
- $N_{\mathcal{V}}(k)$: an iteration bound for $\tilde{\mathcal{V}}$ to run the protocol with $\bar{\mathcal{P}}$.
- $T_{\mathsf{off}}(k)$: an off-line time bound for $\tilde{\mathcal{V}}$ to spend other than running the protocol with $\bar{\mathcal{P}}$.
- $T_{\mathcal{P}}(k)$: a time bound for $\tilde{\mathcal{P}}$ to run the protocol with $\bar{\mathcal{V}}$.

Then for a given public parameter Pub and `"help string"` h, let

$$\Pr[(\tilde{\mathcal{P}}(h), \bar{\mathcal{V}}(\mathsf{Pub}) = 1] = \varepsilon(h, \mathsf{Pub}),$$

where the probability is taken over the coin tosses of $\tilde{\mathcal{P}}$ and $\bar{\mathcal{V}}$. Since we assume that the adversary succeeds in breaking the protocol, there must exist polynomial $\mathit{\Pi}_1(k)$ and $\mathit{\Pi}_2(k)$ such that, for sufficiently large k,

$$\Pr\left[\varepsilon(h, \mathsf{Pub}) \geq \tfrac{1}{\mathit{\Pi}_2(k)} \,\middle|\, \begin{array}{l}(\mathsf{Sec}, \mathsf{Pub}) \leftarrow \mathcal{G}(1^k); \\ h \leftarrow (\bar{\mathcal{P}}(\mathsf{Sec}, \mathsf{Pub}), \tilde{\mathcal{V}}(\mathsf{Pub}))\end{array}\right] \geq \frac{1}{\mathit{\Pi}_1(k)}.$$

Lemma 1. *Let $\hat{e}$ be the modified Weil-pairing as defined in Section 2.3. The sample space is the set of all triples $\mathcal{S} = \{(P, Q) | P, Q \in E(K)\}$, where E is an elliptic curve over K, and the distribution on the sample points is uniform, i.e., $P, Q \in_{\mathcal{U}} \mathcal{S}$. Let a, b, and c be indeterminates and consider the polynomial*

$$e_{a,b,c}(P, Q) = \hat{e}(P, Q)^{abc}.$$

For all $a, b, c \in \mathbb{Z}_m^*$, define random variable

$$X_i(a, b, c) = e_{a_i, b_i, c_i}(P, Q).$$

Then $\langle X_0, \ldots, X_{\ell(m)-1} \rangle$, where $\ell(m)$ is the order of the extension field $\overline{K}$ of K, are uniformly distributed in $\overline{K}$ and pairwise independent.

Proof. For any pair i, j in positive integers, $i \neq j$, and for any pair of points $P, Q \in E(K)$, there is a unique solution $a, b, c \in \mathbb{Z}_m^*$ to the pair of equations:

$$e_{a_i, b_i, c_i}(P, Q) = \alpha,$$
$$e_{a_j, b_j, c_j}(P, Q) = \beta.$$

Thus, $\Pr\left[(X_i(P, Q) = \alpha) \wedge (X_j(P, Q) = \beta)\right] = \Pr[X_i(P, Q) = \alpha] \cdot \Pr[X_j(P, Q) = \beta] = 1/\ell(m)^2$. ∎

Lemma 2. *Assume that there exists an adversary $\mathcal{I}$ as above. Then there exists a polynomial-time probabilistic algorithm $\mathcal{A}$ that (t, ϵ)-breaks* C-DH *problem, whose running time τ is defined by*

$$O((N_{\mathcal{V}}(k)T_{\mathcal{V}}(k) + T_{\mathcal{P}}(k))\Pi_2(k) + T_{\mathsf{off}}(k))$$

and for a valid C-DH *value C, the success probability ϵ is bounded by*

$$\Pr_{G_2}\left[\, C = C' \;\middle|\; \begin{array}{l} x \leftarrow Z, xP \in G_1; \\ y \leftarrow Z, yP \in G_1; \\ z \leftarrow Z, zP \in G_1; \\ C' \leftarrow \mathcal{A}(\hat{e}, xP, yP, zP) \end{array} \right] \geq \frac{\Pi_1(k)^{-1}}{16}.$$

Proof. First let E denote an elliptic curve over a field K, with $E[m]$ its group of m-torsion points. From the definition of the Weil pairing, we know that if $p = 0$ or p does not divides m then $E[m] \cong (\mathbb{Z}/m\mathbb{Z}) \times (\mathbb{Z}/m\mathbb{Z})$, where p is the characteristic of the field. Let Φ be a natural map in the modified Weil pairing. Note that, for random $P \in E(K)$, revealing $\hat{e}(P, P)$ gives no information on $\Phi(P)$; *i.e.* the distribution of $\hat{e}(P, P)$ and $\Phi(P)$ are independent from Lemma 1.

Throughout this paper, the underlying probability space consists of the random choice of input $x, y, z \in \mathbb{Z}_m^*$ and $P \in_R E(K)$ including the coin tosses of the algorithm.

As a proving method, rather than constructing the algorithm $\mathcal{A}$ *in toto*, we will increasingly construct $\mathcal{A}$ in series of "phases". The algorithm runs in five phases. In the first phase, we generate a public parameter $\mathsf{Pub} = \langle P, aP, bP \rangle$ with the corresponding secret parameter $\mathsf{Sec} = \langle a, b \rangle$.

In this phase we simulate the view that the adversary $\mathcal{I}$ would have if it interacted with a proving holding a "real" witness. In the second phase we make the adversary try to convince a honest verifier. In the third phase we use the approximate witness to solve the C-DH problem, $\hat{e}(P, P)^{ab}$. In the fourth phase, we rerun the adversary $\mathcal{I}$ with the public parameter $\mathsf{Pub} = \langle P, aP, bP, cP \rangle$ with additional value cP and its corresponding secret parameter $\mathsf{Sec} = \langle a, b, c \rangle$. In practice, this phase simply executes the above three phases repeatedly. In the last phase, the final algorithm $\mathcal{A}$ is constructed, which solves the C-DH problem, $\hat{e}(P, P)^{abc}$.

Phase 1. This phase takes as input P, aP, bP, runs in the expected time

$$O(N_{\mathcal{V}}(k)T_{\mathcal{V}}(k)\Pi_2(k) + T_{\mathsf{off}}(k)),$$

and outputs $(\tilde{a}, {\gamma_i}^f, v, h)$, where $v = \hat{e}(P, P)^{\tilde{a}{\gamma_i}^f}$, and h is a `"help string"`. In addition, we know that

i. $\Pr[\varepsilon(h, \mathsf{Pub}) \geq \Pi_2(k)^{-1}] \geq \Pi_1(k)^{-1}$,
ii. The distribution of $\Phi(\tilde{c})$ is uniform and independent of that of (h, Pub).

This stage runs as follows: We choose $\tilde{a}, {\gamma_i}^f \in \mathbb{Z}_m^*$, at random and compute $v = \hat{e}(P, P)^{\tilde{a}{\gamma_i}^f}$ and $\hat{X}_i \equiv \tilde{a}{\gamma_i}^f \pmod{m}$, where $f \not\equiv (m-1) \pmod{m}$. With

the help of D-DH oracle, we can easily verify that $(P, \tilde{a}P, {\gamma_i}^f P, abP)$ is a valid DH value. We then simulate the interaction $(\bar{\mathcal{P}}(\cdot, \mathsf{Pub}), \tilde{\mathcal{V}}(\mathsf{Pub}))$.

To simulate the interaction, we employ a zero-knowledge simulation technique [9, 21]. We then modify the identification protocol as the following:

I. $\bar{\mathcal{P}}$ chooses $\omega_0', r_1, r_2 \in \mathbb{Z}_m^*$ at random, computes $x = \hat{e}(P,P)^{\omega_0' r_1 r_2}$, $Q = \omega_0' r_1 r_2 P$, and sends $\langle x, Q \rangle$ to $\tilde{\mathcal{V}}$.
II. $\tilde{\mathcal{V}}$ chooses $\omega \in \mathbb{Z}_m^*$ at random, sets $R = \omega P$, and sends R to $\bar{\mathcal{P}}$.
III. On receiving R, $\bar{\mathcal{P}}$ checks $\hat{e}(R,P) = \hat{e}\big(\frac{\tilde{a}+{\gamma_i}^f-\omega_1}{(\tilde{a}+{\gamma_i}^f)\omega_0} P, P\big)$. If $\omega_0' \neq \omega_0$, we go back to step I. Otherwise, $\bar{\mathcal{P}}$ sets $S = r_1 r_2 P$, computes $Y = \tilde{a}{\gamma_i}^f P + (\tilde{a}+{\gamma_i}^f - \omega_1)S$, and sends it to $\tilde{\mathcal{V}}$.

When the adversary completes the protocol, we outputs the `"help string"` h that $\tilde{\mathcal{V}}$ outputs, along with $\hat{X}_i$.

In this step, the distribution of C is uniformly distributed in $\mathbb{G}_2$, and its distribution is independent of every variable other than in the adversary's view up to that point, and is also independent of the hidden variable ω'. Therefore, up to this point, this simulation is perfectly correct, and furthermore, the probability that $\omega_0 = \omega_0'$ is $1/|\mathbb{Z}_m^*|$. If $\omega_0 = \omega_0'$, then

$$\begin{aligned} v \cdot \hat{e}(\tilde{a}P + \tilde{b}P, Q)^\omega &= v \cdot \hat{e}(\tilde{a}P + {\gamma_i}^f P, \omega_0' r_1 r_2 P)^\omega \\ &= \hat{e}(P,P)^{\tilde{a}{\gamma_i}^f} \cdot \hat{e}(P,P)^{(\tilde{a}+{\gamma_i}^f)\omega_0' r_1 r_2 \omega} \\ &= \hat{e}(P,P)^{\tilde{a}{\gamma_i}^f + (\tilde{a}+{\gamma_i}^f)\omega_0' r_1 r_2 \omega}, \end{aligned}$$

and

$$\begin{aligned} \hat{e}(Y,P) &= \hat{e}(\tilde{a}{\gamma_i}^f P + (\tilde{a} + {\gamma_i}^f - \omega_1) r_1 r_2 P, P) \\ &= \hat{e}(P,P)^{\tilde{a}{\gamma_i}^f + (\tilde{a}+{\gamma_i}^f - \omega_1) r_1 r_2}. \end{aligned}$$

Since $\omega_0 = \omega_0'$ and

$$\begin{aligned} \tilde{a}{\gamma_i}^f + (\tilde{a}+{\gamma_i}^f)\omega_0' r_1 r_2 \omega &\equiv \tilde{a}{\gamma_i}^f + (\tilde{a}+{\gamma_i}^f)\omega_0' r_1 r_2 \frac{\tilde{a}+{\gamma_i}^f - \omega_1}{(\tilde{a}+{\gamma_i}^f)\omega_0} \\ &\equiv \tilde{a}{\gamma_i}^f + (\tilde{a}+{\gamma_i}^f - \omega_1) r_1 r_2, \end{aligned}$$

we have $\hat{e}(Y,P) = v \cdot \hat{e}(\tilde{a}P + \tilde{b}P, Q)^\omega$.

Moreover, C reveals no information of $\Phi(Q_1), \Phi(Q_2)$, and $\Phi(\mathsf{Sec})$, and the distribution of $\Phi(Y)$ is uniform and independent of $\Phi(\mathsf{Sec})$. From the above result, the expected value of the total number of iteration rounds is $(|\mathbb{Z}_m^*| \cdot N_{\mathcal{V}}(k))$. This completes *Phase 1.*

Phase 2. This phase takes as input h, Pub, and output from *Phase 1*, and runs in time $O(T_{\mathcal{P}}(k) \Pi_2(k))$. It outputs Fail or Success according to success outputs u such that $u \equiv \tilde{a}{\gamma_i}^f \equiv ab \pmod{m}$, since $\hat{e}(P,P)^u = \hat{e}(P,P)^{\tilde{a}{\gamma_i}^f} = \hat{e}(P,P)^{ab}$. The probability of success, given that $\varepsilon(h, \mathsf{Pub}) \geq \Pi_2(k)^{-1}$, is at least $1/2$.

For the sake of convenience, let $\varepsilon = \varepsilon(h, \mathsf{Pub})$, and assume $\varepsilon \geq \Pi_2(k)^{-1}$.

This stage runs as follows: First run $(\tilde{\mathcal{P}}(h), \bar{\mathcal{V}}(\mathsf{Pub}))$ up to $\lceil \Pi_2(k) \rceil$ times, or until $\bar{\mathcal{V}}$ accepts. If $\bar{\mathcal{V}}$ accepts, let

$$\begin{aligned}\hat{e}(Y, P) &= \hat{e}(\tilde{a}\tilde{b}P + (\tilde{a} + {\gamma_i}^f - \omega_1)S \\ &= \hat{e}(\omega P, P)^{\tilde{a}\tilde{b} + (\tilde{a} + {\gamma_i}^f - \omega_1) r_1 r_2} \\ &= v \cdot \hat{e}(\tilde{a}P + {\gamma_i}^f P, Q)^{\omega}\end{aligned}$$

be the accepting conversation. Fixing the coin tosses of $\tilde{\mathcal{P}}$, run the interaction again up to $\lceil 4\Pi_2(k) \rceil$, or until $\bar{\mathcal{V}}$ accepts again with a challenge $\omega'' \not\equiv \omega \pmod{m}$. In this case, let $\hat{X}_j \equiv \tilde{a}{\gamma_j}^f \pmod{m}$. If $\bar{\mathcal{V}}$ accepts this challenge, then we have another accepting conversation

$$\begin{aligned}\hat{e}(Y', P) &= \hat{e}(\tilde{a}{\gamma_j}^f P + (\tilde{a} + {\gamma_j}^f - \omega_1')S \\ &= \hat{e}(\omega P, P)^{\tilde{a}{\gamma_j}^f + (\tilde{a} + {\gamma_j}^f - \omega_1') r_1 r_2} \\ &= v \cdot \hat{e}(\tilde{a}P + {\gamma_j}^f P, Q)^{\omega''}\end{aligned}$$

where $u \equiv a{\gamma_i}^f \pmod{m}$, $u \equiv a{\gamma_j}^f \pmod{m}$, and $\omega a{\gamma_i}^f \equiv \omega'' a{\gamma_j}^f \pmod{m}$. Therefore, we can easily calculate $f = \log_{\frac{\gamma_j}{\gamma_i}} \omega - \log_{\frac{\gamma_j}{\gamma_i}} \omega''$.

To show that there is another solution with non-negligible probability, we make use of the same method as employed in [6, 17, 21]. Let M be a Boolean matrix of which rows are indexed by the coin tosses ω' of $\tilde{\mathcal{P}}$ and of which columns are indexed by the challenge ω of $\bar{\mathcal{V}}$. Let $M(\omega', \omega) = 1$ if and only if the pair of (ω', ω) makes $\bar{\mathcal{V}}$ be convinced by $\tilde{\mathcal{P}}$.

Just the same as in [6, 17, 21], we call a row ω' in M "heavy" if the fraction of 1's in this row is at least $3\varepsilon/4$. Then the fraction of 1's in M that lies in heavy rows is at least $1/4$. The reason comes from the following equations: let r be the number of rows in M and c be the number of columns in M, and $\bar{r}$ be the number of non-heavy rows, then the total number of 1's in M is $rc\varepsilon$. Then the total number of 1's that lies in non-heavy rows is $\bar{r}c\frac{3\varepsilon}{4} \leq \left(\frac{3}{4}\right) rc\varepsilon$. Therefore, the fraction of 1's in heavy rows is induced by

$$\begin{aligned}rc\varepsilon - \bar{r}c\frac{3\varepsilon}{4} &\geq rc\varepsilon - rc\frac{3\varepsilon}{4} \\ &= \frac{1}{4}(rc\varepsilon).\end{aligned}$$

Now consider an accepting conversations by (ω', ω) such that $M(\omega', \omega) = 1$. Since we have another accepting conversation by (ω'', ω) satisfying that $M(\omega'', \omega) = 1$. Then the fraction of ω'' which satisfies

$$M(\omega'', \omega) = 1 \qquad \omega'' \not\equiv \omega \pmod{m}$$

is at least

$$\begin{aligned}\left| \frac{3\varepsilon}{4} - \frac{1}{|\mathbb{Z}_m^*| - 2} \right| &\geq \left| \frac{3(\Pi_2(k)^{-1})}{4} - \frac{1}{\Pi_2(k)} \right| \\ &= \frac{1}{4}\frac{1}{\Pi_2(k)} = \frac{\Pi_2(k)^{-1}}{4}.\end{aligned}$$

To complete the construction of this phase, we use the simple fact that if ε is a small real number, then $(1-\varepsilon) \leq e^{-\varepsilon}$ [23]. Let ε be a success probability. When an experiment is repeated at least t times, the probability that all of experiments fail is at most $(1-\varepsilon)^t \leq e^{-t\varepsilon}$. Thus, if $t \geq 1/\varepsilon$, the probability that at least one experiment succeeds is at least $1-e^{-1}$. Therefore, for two accepting conversations, the probability that the above procedure succeeds is at least

$$(1-e^{-1}) \cdot \frac{1}{4} \cdot (1-e^{-1}) = \frac{\left(1-e^{-1}\right)^2}{4}.$$

Thus, by a simple calculation, we can obtain the fact that one of fourteen experiments must succeed, thus the probability that one of seven experiments succeeds is at least $1/2$.

Phase 3. This phase takes as input, the output $\hat{X}_i$ from *Phase 1*, and the value u from *Phase 2*. Its running time is $O(\Pi_2(k) \cdot \log(\Pi_2)^2)$. When *Phase 2* succeeds, the probability that it solves the C-DH problem is $1/2$.

Recall that $\omega \equiv \frac{\tilde{a}+{\gamma_i}^f-\omega_1}{(\tilde{a}+{\gamma_i}^f)\omega_0} \pmod m$, if $\omega' = \omega_0$ then

$$\tilde{a}{\gamma_i}^f \equiv \hat{X}_i \pmod m, \tag{1}$$

$$f \not\equiv (m-1) \pmod m \quad \text{and} \quad f = \log_{\frac{\gamma_j}{\gamma_i}} \omega - \log_{\frac{\gamma_j}{\gamma_i}} \omega'', \tag{2}$$

$$u \equiv \tilde{a}{\gamma_i}^f \pmod m \quad \text{or} \quad u \equiv \tilde{a}{\gamma_j}^f \pmod m, \tag{3}$$

and

$$u \equiv \tilde{a}\tilde{b} \equiv ab \pmod m.$$

Now consider only the case where *Phase 2* succeeds at least with the probability $1/2$. First from Eq. (1), we have $\hat{e}(aP, \gamma_i P) = \hat{e}(P,P)^{a\gamma_i}$, and from Eqs. (2) and (3), we have

$$\begin{aligned}
\hat{e}(P,P)^u &= \hat{e}(P,P)^{\tilde{a}{\gamma_i}^f} \\
&= \hat{e}(\tilde{a}P, {\gamma_i}^f P) \\
&= \hat{e}(\tilde{a}P, \tilde{b}P) \\
&= \hat{e}(P,P)^{\tilde{a}\tilde{b}} = \hat{e}(P,P)^{ab}.
\end{aligned}$$

Then with the probability $1/2$, we can solve the C-DH problem from the following equations: This completes *Phase 3*.

It follows that, for sufficiently large k, the overall success probability of the algorithm $\mathcal{A}$ is at least

$$\varepsilon(h, \mathsf{Pub}) \times \frac{1}{2} \times \frac{1}{2} = \Pi_1(k)^{-1} \times \frac{1}{2} \times \frac{1}{2} = \frac{\Pi_1(k)^{-1}}{4}.$$

Phase 4. This phase repeatedly executes *Phase 1* to *Phase 3* to solve the C-DH problem, $\hat{e}(P,P)^{xc}$, where $x \equiv ab \pmod{m}$. If phases from 1 to 3 succeed, it is straightforward that this phase must succeed with the above probability.

Phase 5. If *Phase 4* succeeds with given probability, it is equivalent to solving the C-DH problem

$$\hat{e}(P,P)^{xc} = \hat{e}(P,P)^{abc}$$

with probability

$$\Pr_{G_2}[C = C'] = \frac{\Pi_1(k)^{-1}}{16}.$$

This completes the proof of Lemma 2. ■

Therefore, we can conclude that the basic scheme satisfies the requirement of Definition 1. This completes the proof of Theorem 1. ■

5 Generalized Scheme

We now describe a generalized model of the basic identification scheme. The generalized identification scheme extends the basic scheme in Section 3 using k random numbers. The key generation algorithm $\mathcal{G}$ is similar to that of the basic scheme except generating k random numbers.

Key generation.
On input k, the key generation algorithm $\mathcal{G}$ works as follows:

1. Generates two cyclic groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of order m for some large prime m and a bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$.
2. Generates an arbitrary generator $P \in \mathbb{G}_1$.
3. Chooses randomly $a_1, \ldots, a_{3k} \in \mathbb{Z}_m^*$ and computes $v_1 = \hat{e}(P,P)^{a_1 a_2 a_3}, \cdots, v_k = \hat{e}(P,P)^{a_{3k-2} a_{3k-1} a_{3k}}$.
4. The public parameter is $\mathsf{Pub} = \langle \mathbb{G}_1, \mathbb{G}_2, P, a_1 P, \ldots, a_{3k} P, \hat{e}, v_1, \cdots, v_k \rangle$, and the secret parameter is $\mathsf{Sec} = \langle a_1, \ldots, a_{3k} \rangle$. And then publishes them.

Protocol actions between $\mathcal{P}$ and $\mathcal{V}$.
The generalized scheme is similar to the basic scheme, however, each round is performed in parallel as follows:

1. $\mathcal{P}$ chooses $r_1, r_2, r_3 \in \mathbb{Z}_m^*$ at random, computes $x = \hat{e}(P,P)^{r_1 r_2 r_3}$, $Q_1 = r_1 r_2 r_3 P$, and sends $\langle x, Q \rangle$ to B.
2. $\mathcal{V}$ picks $\omega_1, \ldots, \omega_k \in \mathbb{Z}_m^*$ at random, and sends $R_1 = \omega_1 P, \ldots, R_k = \omega_k P$ to $\mathcal{P}$.
3. On receiving k random values, $\mathcal{P}$ sets

$$S_1 = r_1 r_2 r_3 R_1, S_2 = r_1 r_2 r_3 R_2, \ldots, S_k = r_1 r_2 r_3 R_k,$$

computes Y such that

$$Y = \sum_{i=1}^{k} a_{3i-2}a_{3i-1}a_{3i}P + \sum_{i=1}^{k}(a_{3i-2} + a_{3i-1} + a_{3i})S_i$$

and sends it to $\mathcal{V}$; $\mathcal{V}$ accepts if both $x = \hat{e}(P, Q)$ and $\hat{e}(Y, P) = \prod_{i=1}^{k} v_i \cdot \hat{e}(a_{3i-2}P + a_{3i-1}P + a_{3i}P, Q)^{\omega_i}$, and rejects otherwise.

Theorem 2. *Under B-DHIA, the generalized identification scheme in this section on* (τ', t', ϵ')-B-DH *groups is secure against active attacks.*

Proof(sketch) At first we assume that there exists an (t', ϵ')-breakable adversary $\mathcal{A}$ who can break this identification scheme. Then from the proof of Theorem 1, we can prove Theorem 2.

6 Comparison

In this section, we compare our basic scheme with the prior schemes in terms of not only the computation overhead in the light of key size, communication overhead, processing complexity but also their security.

We assume that an elliptic curve E over a base field K is chosen in the same manner as [2]. That is, let E be the elliptic curve defined by the equation $y^2 = x^3 + 1$ over $\mathbb{F}_p$, where p is a prime satisfying $p \equiv 2 \pmod 3$ and $p = 6q - 1$ for some prime $q > 3$. Note that for the sake of the convenience m is replaced by q. As pointed out in [2], from the practical point of view, we can assume that p and q is a 512-bit prime and a 140-bit prime respectively, since the MOV reduction [13] then leads to a DLP in a finite field of size approximately 2^{1024}.

In addition, we assume that system parameters p and q for our basic scheme, Schnorr, and Okamoto are 512-bit and 140-bit respectively, and the modulus n for FFS, GQ scheme is 512-bit. We assume that the standard binary method is employed for the modular exponentiation as well as for the point multiplication in polynomial basis form. We also assume that the parameters for FFS are $l = 20$ and $t = 1$. Here, we only consider Okamoto scheme as an *Identification scheme 1* proposed in [15]. Note that for the purpose of comparison with arithmetic operations of each scheme, we denote M the cost of modular multiplication over a given finite field and A the cost of point addition over a given elliptic curve. Table 1 shows the comparison of identification schemes. If the Weierstraß equation over the affine coordinates in fields of characteristic two is given by $y^2 + xy = x^3 + a_2x^2 + a_6$, then our scheme has $a_2 = 0$. Furthermore, since a generator P of the group $\mathbb{G}_1$ is initially known all parties, we can enable the point multiplication in elliptic curves to be more faster. In fact, the point multiplication consists of point doublings and point additions. The binary method requires $(\ell - 1)$ point doublings and $(W - 1)$ point additions, where ℓ is the bit length and W the Hamming weight of the binary expansion, in general, $W = \ell/2$. Therefore, if the point doublings are pre-computed, the point multiplication requires $\frac{\ell}{2}$A-point addition in average and ℓA-point addition in the

Table 1. Comparison of identification schemes

	Our scheme	Schnorr	Okamoto	FFS	GQ
Security proof	Yes	Yes	Yes	Yes	Yes
Secure against active attacks	Yes	No	Yes	Yes	No
Underlying problem	B-DH	DLP	DLP	RSA	RSA
ID-based variant	Possible	Possible	Possible	Possible	Possible
Public key size (bits)	512	512	512	10,240	1,024
Private key size (bits)	420	140	280	10,240	512
Communication overhead (bits)	932	672	812	1,044	1,044
Preprocessing (Prover) (# of field multiplications or point additions)	140A	210M	245M	1M	30M
On-line processing (Prover) (# of field multiplications over a given finite field)	2M	Almost 0M	Almost 0M	10M	31M
On-line processing (Verifier) (# of field multiplications over a given finite field)	141M	210M	248M	11M	35M

worst case [4]. The pre-computation is possible because P is initially given. In these cases, we can estimate that A costs less than or equal to two times M, i.e., $\mathsf{A} \leq 2\mathsf{M}$.

From Table 1, we can state the properties of our scheme as follows: (1) Our scheme is more efficient than Schnorr and Okamoto with respect to preprocessing of prover and on-line processing overhead of both parties (prover and verifier). (2) However, our scheme requires memory for secret key about two times that of Schnorr and Okamoto. Moreover, its communication overhead increases around four times more than those two schemes.

7 Concluding Remarks

In this paper we present a practical construction of a new identification scheme based on the B-DH problem using the Weil pairing. Then we prove that our identification scheme is secure against active attacks. Our proposal can be extended to a signature scheme using the Weil pairing. Also similar to IBE (Identity-Based Encryption) scheme proposed by Boneh *et al.*, our scheme can be associated with the public identity such as e-mail. It remains as an open problem to implement an algorithm to efficiently compute the Weil pairing as suggested in [24].

Acknowledgements

The authors are grateful to Fangguo Zhang for his useful comments to improve the 1st version of this paper.

References

1. M. Bellare and P. Rogaway, "Random Oracles are Practical: A Paradigm for Designing Efficient Protocols", *ACM Conference on Computer and Communications Security*, pp. 62–73, 1993.
2. D. Boneh and M. Franklin, "ID-based encryption from the Weil-pairing", *Advances in Cryptology – Crypto '2001*, LNCS 2139, Springer-Verlag, pp. 213–229, 2001.
3. D. Boneh, H. Shacham, and B. Lynn, "Short signatures from the Weil-pairing", *Advances in Cryptology – Asiacrypt '2001*, LNCS 2248, Springer-Verlag, pp. 514–532, 2001.
4. I. Blake, G. Seroussi and N. Smart, "Elliptic curves in cryptography", Cambridge University Prress, LNS 265, 1999.
5. J.-S. Coron, "On the security of full domain hash", *Advances in Cryptology – Crypto '2000*, LNCS 1880, Springer-Verlag, pp. 229–235, 2000.
6. U. Feige, A. Fiat, and A. Shamir, "Zero-knowledge proofs of identity", *J. Cryptology*, 1: 77–94, 1988.
7. A. Fiat and A. Shamir, "How to prove yourself: pratical solutions to identification and signature problems", *Advances in Cryptology – Crypto '86*, LNCS 263, Springer-Verlag, pp. 186-194, 1987.
8. O. Goldreich and H. Krawczyk, "On the composition of zero-knowledge proof systems", In *Proceedings of the 17th ICALP*, LNCS 443, Springer-Verlag, pp. 268–282, 1990.
9. S. Goldwasser, S. Micali, and C. Rackoff, "The knowledge complexity of interactive proof systems", *SIAM J. Comput.*, 18: 186–208, 1989.
10. L. Guillou and J. Quisquater, "A practical zero-knowledge protocol fitted to security microprocessors minimizing both transmission and memory", *Advances in Cryptology – Eurocrypt '88*, LNCS 330, Springer-Verlag, pp. 123–128, 1989.
11. A. Joux and K. Nguyen, "Seperating decision Diffie-Hellman from Diffie-Hellman in cryptographic groups", available from `eprint.iacr.org`.
12. A. J. Menezes, "Elliptic curve public key cryptosystems", Kluwer Academic Publishers, 1993.
13. A. J. Menezes, T. Okamoto, and S. A. Vanstone, "Reducing elliptic curve logarithms to logarithms in a finite field", *IEEE Trans. Inform. Theory*, 39(1993), pp. 1639–1646.
14. V. Miller, "Short programs for functions on curves", unpublished manuscript, 1986.
15. T. Okamoto, "Provably secure and practical identification schemes and corresponding signature schemes", *Advances in Cryptology – Crypto '92*, LNCS 740, Springer-Verlag, pp. 31–53, 1993.
16. T. Okamoto and D. Pointcheval, "The gap-problem: a new class of problems for the security of cryptographic schemes", *PKC 2001*, LNCS 1992, Springer-Verlag, pp. 104–118, 2001.
17. K. Ohta and T. Okamoto, "A modification of the Fiat-Shamir scheme", *Advances in Cryptology – Crypto '88*, LNCS 403, Springer-Verlag, pp. 232–243, 1990.
18. C. Popescu, "An identification scheme based on the elliptic curve discrete logarithm problem", *IEEE High Performance Computing in the Asia-Pacific Region*, Volume: 2, pp. 624–625, 2000.
19. A.D. Santis, S. Micali, and G. Persiano, "Non-interactive zero-knowledge proof systems", *Advances in Cryptology – Crypto '87*, LNCS 293, pp. 52–72, 1988.
20. C. Schnorr, "Security of 2^t-root identification and signatures", *Advances in Cryptology – Crypto '96*, LNCS 1109, Springer-Verlag, pp. 143–156, 1996.

21. V. Shoup, "On the security of a practical identification scheme", *J. Cryptology* 12: 247–260, 1999.
22. J. H. Silverman, "The arithmetic of elliptic curves", Springer-Verlag, GTM 106, 1986.
23. D.R. Stinson, "Cryptography: Theory and Practice", CRC Press, Boca Raton, Florida, pp. 236, 1995.
24. T. Yamanaka, R. Sakai, and M. Kasahara, "Fast computation of pairings over elliptic curves", *Proc. of SCIS 2002*, pp. 709–714, Jan. 29 – Feb. 1, 2002, Shirahama, Japan.