SCIS 2002 The 2002 Symposium on Cryptography and Information Security
Shirahama,Japan, Jan.29-Feb.1,2002
The Institute of Electronics,
Information and Communication Engineers

A New Identification Scheme based on the Gap Diffie-Hellman Problem

Myungsun Kim *
ms.kim@icu.ac.kr

Kwangjo Kim *
kkj@icu.ac.kr

Abstract— We introduce a new identification scheme based on the Gap Diffie-Hellman problem. Our identification scheme makes use of 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. We prove that this scheme is secure against active attacks if the Gap Diffie-Hellman problem is intractable. Finally, we analyze efficiency of the scheme comparing with other identification schemes.

Keywords: Gap-problems, Identification scheme, Weil-pairing

1 Introduction

It is the well-known fact 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 information 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 [6] and Shoup [19], the distinction between identification and signature schemes is subtle. Therefore, two types of schemes can be used interchangeably [6]. We can find several clear evidences in [13, 9, 15].

Since Okamoto and Pointcheval [14] 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 (DH) assumption has been studied. Using the Weil-pairing, Boneh and Franklin [2] and Boneh *et al.* [3] suggested an efficient ID-based encryption scheme and short signature scheme, respectively. To the best of our knowledge, there is no an identification scheme based on a Gap-problem published in the open literature.

In this paper, we propose a new identification scheme based on the Gap Diffie-Hellman (G-DH) problem. Joux and Nguyen [10] suggested that there exist groups in which the decisional Diffie-Hellman (D-DH) problem is easy, although the computational Diffie-Hellman (C-DH) problem is hard in a group. The DH problem on such a group is called the G-DH problem. Our scheme makes use of such groups. We prove that our scheme is secure against active attacks if the G-DH problem is hard.

The rest of the paper is organized as follows. Several identification schemes are discussed in the following subsection. In Section 2 we formally state our definition of security as well as basic tools used in our scheme. Our identification scheme is presented in Section 3 based on the G-DH problem. In Section 4 we give a proof of security for our scheme. In Section 5 we present a generalized model of our identification. In Section 6 we end with concluding remarks.

1.1 Previous Works

Types of attacks.
What an identification scheme is broken means that 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 [19].

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 has is the public key of the prover. Other attacks of intermediate level such as *eavesdropping attack* or *honest-verifier attack* are essentially equivalent to a 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-cipher text attacks. we should note that active attacks are quite feasible in practice.

Fiat-Shamir (FS) scheme. Fiat and Shamir [6] proposed the identification scheme based on the factoring problem. A key generation algorithm constructs a modulus n by multiplying two distinct random primes; chooses randomly an element $a \in \mathbf{Z}_n^*$, and sets $b = a^2$. The public key is $\langle b, n \rangle$, and the secret key is a.

The protocol repeats the followings t times:

1. $\mathcal{P}$ chooses $r \in \mathbf{Z}_n^*$ 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 factoring is hard.

* International Research center for Information Security (IRIS), Information and Communications Univ. (ICU), 58-4 Hwaam-dong, Yuseong-gu, Taejon, 305-732, Korea.

Feige-Fiat-Shamir (FFS) scheme. This scheme is also based on the factoring problem. A key generation algorithm chooses a modulus n as in the FS scheme. A secret key consists of a list $a_1, \ldots, a_l \in \mathbf{Z}_n^*$ randomly chosen, where l is a given constant, and the corresponding public key consists of $b_1, \ldots, b_l \in Z_n^*$, 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 $r \in \mathbf{Z}_n^*$ 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 factoring is hard [5].

Other schemes. The Guillou-Quisquater (GQ) scheme is based on the RSA-inversion problem. Guillou [9] shows that this scheme is secure against passive attacks provided that factoring is hard. Ohta and Okamoto [15] presents 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 [13] proposes three identification schemes. The first one is based on the discrete logarithm (DLP) problem, the second is based on the RSA-problem, the last is based on the factoring problem. All of the schemes are proved to be secure against active attacks. Schnorr [18] also proposes identification schemes that are based on the factoring problem or the DLP problem.

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 [7], together with the argument of Shoup [19] 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 breaking the cryptosystem rather than zero-knowledge proof techniques. That is to say, the intuition of the proof is that we use an adversary that breaks the cryptosystem to solve the G-DH problem.

2.1 Notions of Security

We formally define a secure identification scheme, following the same notations used in [17] and [19].

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 adopt the following conventions [5]:

1. $\bar{\mathcal{P}}$ represents a 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 [5, 19]:

1. The algorithm $\mathcal{G}$ is a *key generation algorithm.* It takes as input a string of the form 1^k, 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. $\mathcal{P}$ receives as input the pair (I, S) and $\mathcal{V}$ receives as input I. After an interactive execution of $\mathcal{P}$ and $\mathcal{V}$, $\mathcal{V}$ outputs either a 1 (indicating `"accept"`) or a 0 (indicating `"reject"`). For a given I and S, the output of $\mathcal{V}$ at the end of this interaction is a probability space and 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 a 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 a 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 $\tilde{\mathcal{P}}$ which 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 $\tilde{\mathcal{P}}(h)$.

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 The Gap Diffie-Hellman Problem

The computational assumptions on which cryptographic schemes are based can largely be divided into two types. One is the intractability of an inverting problem such as inverting the RSA function, and computing the DH problem. The other is the intractability of a decision problem such as the D-DH problem.

In addition to such computational problems, Okamoto and Pointcheval [14] define a new class of problems, called the Gap-problems as follows. 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.

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 [14] claimed that the DH problems are the typical instance of the Gap problems. As the inverting problem can be viewed as the computational problem, the C-DH problem corresponds to the inverting one, and the D-DH problem does to the R-decision one. Let $\mathbf{G}$ be any group of prime order q.

- The C-DH problem: given a triple of $\mathbf{G}$ elements (g, g^a, g^b), find the element $C = g^{ab}$.
- The D-DH problem: given a quadruple of $\mathbf{G}$ elements (g, g^a, g^b, g^c), decide whether $c = ab \pmod q$ or not.
- The G-DH problem: given a triple of $\mathbf{G}$ 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 of the G-DH problem in [14], and the Weil-pairing [20, 11] appears as that of the G-DH problem in [2, 3].

Now we formally define groups in which the Weil pairing works using notions defined above. Assume any group action can be computed in a unit time.

Definition 3 *Let* $\mathbf{G}$ *be a cyclic group of a prime order with an arbitrary generator. For any polynomial-time probabilistic algorithm* $\mathcal{A}$*:*

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

2.3 The Weil-pairing

We can make use of any bilinear map on an elliptic curve to construct a group $\mathbf{G}$ in which the C-DH problem is intractable, but the D-DH problem is tractable [10, 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 $\mathbf{G}_1$ and $\mathbf{G}_2$ be two cyclic groups of order q for some large prime p. The *Weil pairing* [20] is defined by a bilinear map e,

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

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

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

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

In addition to these properties, we have an efficient algorithm to compute $e(P, Q)\ \forall\ P, Q \in \mathbf{G}_1$ by [12].

As is noted in [2], the existence of the bilinear map e implies (1) DLP in $\mathbf{G}_1$ can be reduced to DLP in $\mathbf{G}_2$, (2) C-DH problem in $\mathbf{G}_1$ is still hard even though D-DH in $\mathbf{G}_1$ is easy [10].

3 Our Identification Scheme: Type I

For security parameter k, a pair of secret and public parameters is generated as follows. In practice, in our scheme (say Identification scheme Type I), we adopt 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].

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

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

Protocol actions between $\mathcal{P}$ and $\mathcal{V}$.
As is the case for other identification schemes, Identification scheme Type I includes several rounds, each of these is performed as follows:

1. $\mathcal{P}$ chooses $r_1, r_2, r_3 \in \mathbf{Z}_m^*$ at random, computes $x = \hat{e}(P, P)^{r_1 r_2 r_3}$, $Q_1 = r_1 P$, $Q_2 = r_2 P$, and $Q_3 = r_3 P$, and sends $\langle x, Q_1, Q_2, Q_3 \rangle$ to $\mathcal{V}$.
2. $\mathcal{V}$ picks $\omega \in \mathbf{Z}_m^*$ at random, and sends ω to $\mathcal{P}$.
3. $\mathcal{P}$ computes $y = \hat{e}(\omega P, P)^{abc} \cdot \hat{e}(P, P)^{r_1 r_2 r_3}$ and sends to $\mathcal{V}$; $\mathcal{V}$ accepts if $y = v^{\omega} \cdot x$, and rejects otherwise.

4 Proof of Security

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

Definition 4 *Let* $Z_{\mathcal{K}}$ *be a probability space consisting uniform distribution over all integers in* $\mathbf{Z}_m^*$*. Let* $G_{\mathcal{K}}$ *be a probability space consisting the uniform distribution over all elements of the form* $nP \neq \mathcal{O} \in \mathbf{G_1}$*, where* $n \in Z_{\mathcal{K}}$*.* G-DHIA *is defined as the following: Given* $C = \hat{e}(P, P)^{abc} \in \mathbf{G}_2$*, for all polynomial-time probabilistic algorithm* $\mathcal{A}$*, for all constant* $c > 0$*, and for all sufficiently large* k*,*

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

Now we will prove:

Theorem 1 *Under G-DHIA, Identification scheme* Type I *on* (τ, t, ϵ)-G-DH *groups is secure against active attacks.*

As mentioned before, to prove Theorem 1, it is good enough 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 is proved in Lemma 1.

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

- $T_{\mathcal{V}}(k)$: 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 $\Pi_1(k)$ and Π_2 such that, for sufficiently large k,

$$\Pr\left[\ \varepsilon(h, \mathsf{Pub}) \geq \tfrac{1}{\Pi_2(k)} \ \middle|\ \begin{array}{l}(\mathsf{Sec}, \mathsf{Pub}) \leftarrow \mathcal{G}(1^k);\\ h \leftarrow (\bar{\mathcal{P}}(\mathsf{Sec}, \mathsf{Pub}), \tilde{\mathcal{V}}(\mathsf{Pub}))\end{array}\ \right] \geq \tfrac{1}{\Pi_1(k)}.$$

Lemma 1 *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

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

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 $E[m] \cong (\mathbf{Z}/m\mathbf{Z}) \times (\mathbf{Z}/m\mathbf{Z})$, 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.

Throughout this paper, the underlying probability space consists of the random choice of input $x, y, z \in \mathbf{Z}_m^*$, and $P \in_R E(K)$ and 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 a 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 $\hat{X}_i$, where $\hat{X}_i \equiv a{\gamma_i}^f \bmod m$, $f \not\equiv (m-1) \bmod m$, and $\gamma_i \in \mathbf{Z}_m^*$ is picked randomly by $\mathcal{A}$, and h is a "help string". In addition, we know

i. $\Pr[\varepsilon(h, \mathsf{Pub}) \geq \Pi_2(k)^{-1}] \geq \Pi_1(k)^{-1}$,

ii. the distribution of $\Phi(\hat{X}_i)$ is uniform and independent of that of (h, Pub).

This stage runs as follows: We choose $\gamma_i \in \mathbf{Z}_m^*, 1 \leq i \leq |\mathbf{Z}_m^*| - 1$, at random and compute $\hat{X}_i \equiv a{\gamma_i}^f \bmod m$. With the help of D-DH oracle, we can easily verify that $(aP, {\gamma_i}^f P)$ 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 [8, 19]. We then modify the identification protocol as the following:

I. $\bar{\mathcal{P}}$ chooses $\omega', r_1, r_2 \in \mathbf{Z}_m^*$ at random, computes $x = \hat{e}(P, P)^{\omega'} \cdot \hat{e}(P, P)^{r_1 r_2}$, $Q_1 = r_1 P$, $Q_2 = r_2 P$, and sends $\langle x, Q_1, Q_2\rangle$ to $\tilde{\mathcal{V}}$.

II. $\tilde{\mathcal{V}}$ chooses $\omega \in \mathbf{Z}_m^*$ at random, and sends ω to $\bar{\mathcal{P}}$.

III. $\bar{\mathcal{P}}$ sets $\omega \equiv (\hat{X}_i\omega_1 + \omega_0) \bmod m$.
If $\omega' \neq \omega_0$, we go back to step I. Otherwise, $\bar{\mathcal{P}}$ computes $y = \hat{e}(aP, {\gamma_i}^f P)^{\hat{X}_i\omega_1} \cdot \hat{e}(P, P)^{r_1 r_2}$ and sends 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 $\mathbf{G}_2$, 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 perfect, and furthermore, the probability that $\omega_0 = \omega'$ is $1/|\mathbf{Z}_m^*|$. If $\omega_0 = \omega'$, then

$$\begin{aligned} C' &= \hat{e}(aP, {\gamma_i}^f P)^{\hat{X}_i\omega_1}\hat{e}(P, P)^{\omega'}\hat{e}(P, P)^{r_1 r_2}\\ &= \hat{e}(aP, {\gamma_i}^f P)^{\hat{X}_i\omega_1 + \omega'}\hat{e}(P, P)^{r_1 r_2}\\ &= \hat{e}(aP, {\gamma_i}^f P)^{a{\gamma_i}^f\omega_1 + \omega_0}\hat{e}(P, P)^{r_1 r_2}\\ &= \hat{e}(P, P)^{a{\gamma_i}^f\omega}\hat{e}(P, P)^{r_1 r_2}\\ &= \hat{e}(aP, {\gamma_i}^f P)^{\omega}\hat{e}(P, P)^{r_1 r_2}\\ &= v^{\omega} \cdot x = C. \end{aligned}$$

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 $(|\mathbf{Z}_m^*| \cdot N_{\mathcal{V}}(k))$. 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 Z such that $Z \equiv a{\gamma_i}^f \equiv ab \bmod m$, since $\hat{e}(P, P)^Z = \hat{e}(P, P)^{a{\gamma_i}^f} = \hat{e}(P, P)^{ab}$, where $f \not\equiv (m-1) \bmod m$. 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

$$\begin{aligned} y &= \hat{e}(\omega P, P)^Z \hat{e}(P, P)^{r_1 r_2} \\ &= \hat{e}(\omega P, P)^{a\gamma_i{}^f} \hat{e}(P, P)^{r_1 r_2} \\ &= \hat{e}(\omega P, P)^{ab} \hat{e}(P, P)^{r_1 r_2} \\ &= v^{\omega} \cdot x \end{aligned}$$

be the accepting conversation. Fixing the coin tosses of $\tilde{\mathcal{P}}$, run the interaction again up to $\lceil 3\Pi_2(k) \rceil$, or until $\bar{\mathcal{V}}$ accepts again with a challenge $\omega'' \not\equiv \omega \bmod m$. In this case, let $\hat{X}_j \equiv a\gamma_j{}^f \bmod m$. If $\bar{\mathcal{V}}$ accepts this challenge, then we have another accepting conversation

$$\begin{aligned} y' &= \hat{e}(\omega'' P, P)^Z \hat{e}(P, P)^{r_1 r_2} \\ &= \hat{e}(\omega'' P, P)^{a\gamma_j{}^f} \hat{e}(P, P)^{r_1 r_2} \\ &= \hat{e}(\omega'' P, P)^{ab} \hat{e}(P, P)^{r_1 r_2} \\ &= v^{\omega''} \cdot x, \end{aligned}$$

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

We analyze this phase using a variant of a truncated execution tree as employed in [5, 15, 19]. 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 $\tilde{\mathcal{P}}$.

As used in [5, 15, 19], 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 lie 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

$$M(\omega'', \omega) = 1 \qquad \omega'' \not\equiv \omega \bmod m$$

is at least

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

To complete the construction of this phase, we use the simple fact that if θ is a small real number, then $(1-\theta) \leq \exp^{-\theta}$ [21]. Let θ be a success probability. When an experiment is repeated at least t times, the probability that all of experiments fail is at most $(1-\theta)^t \leq \exp^{-t\theta}$. Thus, if $t \geq 1/\theta$, the probability that at least one experiment succeeds is at least $1 - \exp^{-1}$. Therefore, for two accepting conversations, the probability that the above procedure succeeds is at least

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

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 Z from *Phase 2*. When *Phase 2* succeeded, the probability that it solves the C-DH problem is $1/2$.

Recall that $\omega \equiv (\hat{X}_i\omega_1 + \omega_0) \bmod m$, if $\omega' = \omega_0$ then

$$a\gamma_i \equiv \hat{X}_i \bmod m, \tag{1}$$

and

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

and

$$Z \equiv a\gamma_i{}^f \bmod m \quad \text{or} \quad Z \equiv a\gamma_j{}^f \bmod m, \tag{3}$$

and

$$Z \equiv ab \bmod m.$$

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 Eq. (2) and Eq. (3), we have

$$\begin{aligned} \hat{e}(P, P)^Z &= \hat{e}(P, P)^{a\gamma_i{}^f} \\ &= \hat{e}(aP, \gamma_i{}^f P) \\ &= \hat{e}(aP, bP) \\ &= \hat{e}(P, P)^{ab}. \end{aligned}$$

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 $x \equiv ab \bmod m$. If phases from 1 to 3 succeed, 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

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

This completes the proof of Lemma 1. □

Therefore, we can conclude that Identification scheme Type I satisfies the requirement of Definition 1.

5 Generalized Identification Scheme: Type II

We now describe a generalized model of identification scheme Type I, say Identification scheme Type II. Identification scheme Type II extends Identification scheme Type I using k random numbers. The key generation algorithm $\mathcal{G}$ is similar to that of Identification scheme Type I except generating k random numbers.

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

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

Protocol actions between $\mathcal{P}$ and $\mathcal{V}$.
Identification scheme Type II is similar to Identification scheme Type I, however, each round is performed in parallel as follows:

1. $\mathcal{P}$ chooses $r_1, r_2, r_3 \in \mathbf{Z}_m^*$ at random, computes $x = \hat{e}(P,P)^{r_1 r_2 r_3}$, $Q_1 = r_1 P$, $Q_2 = r_2 P$, and $Q_3 = r_3 P$, and sends $\langle x, Q_1, Q_2, Q_3 \rangle$ to B.
2. $\mathcal{V}$ picks $\omega_1, \ldots, \omega_k \in \mathbf{Z}_m^*$ at random, and sends $\omega_1, \ldots, \omega_k$ to $\mathcal{P}$.
3. $\mathcal{P}$ computes

$$y = \left(\prod_{i=1}^{k} \hat{e}(\omega_i P, P)^{a_{3k-2} a_{3k-1} a_{3k}} \right) \cdot \hat{e}(P,P)^{r_1 r_2 r_3}$$

and sends to $\mathcal{V}$; $\mathcal{V}$ accepts if $y = \left(\prod_{i=1}^{k} v_i^{\omega_i} \right) \cdot x$, and rejects otherwise.

Theorem 2 *Under G-DHIA, Identification scheme* Type II *on* (τ, t, ϵ)-G-DH *groups is secure against active attacks.*

This theorem follows immediately from Theorem 1.

6 Concluding Remarks

In this paper we present a practical construction of a new identification scheme based on the G-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.

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] J.-S. Coron, "On the security of full domain hash", *Advances in Cryptology – Crypto '2000*, LNCS 1880, Springer-Verlag, pp. 229–235, 2000.

[5] U. Feige, A. Fiat, and A. Shamir, "Zero-knowledge proofs of identity", *J. Cryptology*, 1: 77–94, 1988.

[6] 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.

[7] 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.

[8] S. Goldwasser, S. Micali, and C. Rackoff, "The knowledge complexity of interactive proof systems", *SIAM J. Comput.*, 18: 186–208, 1989.

[9] 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.

[10] A. Joux and K. Nguyen, "Seperating decision Diffie-Hellman from Diffie-Hellman in cryptographic groups", available from `eprint.iacr.org`.

[11] A. J. Menezes, "Elliptic curve public key cryptosystems", Kluwer Academic Publishers, 1993.

[12] V. Miller, "Short programs for functions on curves", unpublished manuscript, 1986.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] C. Schnorr, "Security of 2^t-root identification and signatures", *Advances in Cryptology – Crypto '96*, LNCS 1109, Springer-Verlag, pp. 143–156, 1996.

[19] V. Shoup, "On the security of a practical identification scheme", *J. Cryptology* 12: 247–260, 1999.

[20] J. H. Silverman, "The arithmetic of elliptic curves", Springer-Verlag, GTM 106, 1986.

[21] D.R. Stinson, "Cryptography: Theory and Practice", CRC Press, Boca Raton, Florida, pp. 236, 1995.