Diffie-Hellman Problems and Bilinear Maps

Jung Hee Cheon[1] and Dong Hoon Lee[2]

[1] Information and Communications University (ICU), Taejon, Korea
jhcheon@icu.ac.kr,
[2] National Security Research Institute (NSRI), Taejon, Korea
dlee@etri.re.kr

Abstract. We investigate relations among the discrete logarithm (DL) problem, the Diffie-Hellman (DH) problem and the bilinear Diffie-Hellman (BDH) problem when we have an efficient computable non-degenerate bilinear map $e : G \times G \to H$. Under a certain assumption on the order of G, we show that the DH problem on H implies the DH problem on G, and both of them are equivalent to the BDH problem when e is *weak-invertible*. Moreover, we show that given the bilinear map e an injective homomorphism $f : H \to G$ enables us to solve the DH problem on G efficiently, which implies the non-existence a *self-bilinear* map $e : G \times G \to G$ when the DH problem on G is hard. Finally we introduce a sequence of bilinear maps and its applications.

1 Introduction

The Weil pairing on an elliptic curve have been used to solve cryptographic problems such as the discrete logarithm (DL) problem, the (computational) Diffie-Hellman (DH) problem, the decisional Diffie-Hellman (DDH) problem [13]. After Joux proposed tripartite Diffie-Hellman protocol using the Weil paring, however, the Weil (or Tate) pairing is being used as a building block of interesting cryptographic protocols including ID-based schemes, a short signature scheme, and self-blindable credentials [9, 4, 7, 6, 18, 16].

The *bilinear* property of the pairings plays an important role on pairing-based protocols. Given two groups G and H, a map $e : G \times G \to H$ is said to be bilinear if $e(g_1^{x_1}, g_2^{x_2}) = e(g_1, g_2)^{x_1 x_2}$ for all $x_i \in \mathbb{Z}$ and $g_i \in G$. Given a quadruple (g, g^x, g^y, g^z) the bilinear Diffie-Hellman (BDH) problem asks to find $e(g, g)^{xyz}$. Though security of most paring-based protocols relies on the BDH problem, the hardness of the BDH problem or its relations with other well-known problems are not studied well.

In this paper, we investigate relations between the BDH problem and other well-known problems and how the properties of bilinear maps influence on their strength. More precisely, we showed that the DH problem on G and H implies the BDH problem. The DH problem on H implies the DH problem on G if the order of G satisfies a certain condition which is believed to be true for almost all primes [11, 12].

To show the inverse direction of the above implications, we need to have an *invertible* bilinear map. Since a bilinear map is a two-variable function, invertibility can be defined several ways. A bilinear map is said to be *weak-invertible* if one can efficiently compute an inverse image (g_1, g_2) of h such that $e(g_1, g_2) = h$ for any $h \in H$. Under the weak-invertible assumption of a bilinear map, we show that the DH problem on H, and so the DH problem on G on a certain assumption, are equivalent to the BDH problem.

We can consider a stronger notion that there exists $g \in G$ such that one can efficiently compute an inverse image g_1 of h such that $e(g_1, g) = h$ for any $h \in H$. In this case the DH problem on G is efficiently solved, and so is the BDH problem. More generally, we show that given a bilinear map $e : G \times G \to H$, if there is an injective homomorphism $f : H \to G$, then the DH problem on G is efficiently solved. As a corollary, we show that an efficiently computable non-degenerate bilinear map $e_s : G \times G \to G$ does not exist on a group G on which the DH problem is hard. A similar result was introduced by Verheul on the XTR groups [17].

It is natural to consider n-multilinear maps as an extension of a bilinear map. An n-multilinear map is a map from n-tuple of a cyclic group to another cyclic group that is linear on each variable. This map can be used to design many interesting cryptographic protocols including a non-interactive multiparty Diffie-Hellman, a broadcast encryption, and a unique signature [5]. One possible approach to find multilinear maps is to have a sequence of groups and bilinear maps between each consecutive groups. However our result implies that there should not exist efficiently computable isomorphism between any of two groups in this chain. The applications of the family includes a forward-secure Diffie-Hellman as well as all applications of multilinear maps.

The rest of the paper is organized as follows: In Section 2, we introduce bilinear maps and several DH and BDH related problems. In Section 3, we recall known relations between the DH related problems and propose a useful lemma for granularity. In Section 4, we investigate the relations as invertible properties of a bilinear map vary. In Section 5 we introduce a sequence of bilinear maps and its applications. We conclude in Section 6.

2 Problems related to the DH and the BDH

Let G and H be cyclic groups of prime order p. We use the multiplicative group notations. We define bilinear maps and several problems related to the DH problem and the BDH problem.

2.1 Problems related to the DH

Definition 1. *For each $g \in G$ the DL_g problem is defined as follows: Given (g, g^x), compute $x \in \mathbb{Z}_p$. The DL_G problem asks to solve the DL_g problem for arbitrary $g \in G$. We call by the DL_g oracle a probabilistic algorithm to give a solution of the DL_g problem. By notation, we set $DL_g(g, g^x) = x$.*

Definition 2. *For each $g \in G$ the DH_g problem is defined as follows: Given (g, g^x, g^y), compute g^{xy}. The DH_G problem asks to solve the DH_g problem for arbitrary $g \in G$. We* *call by the DH_g oracle a probabilistic algorithm to give a solution of the DH_g problem. By notation, we set $DH_g(g, g^x, g^y) = g^{xy}$.*

Definition 3. *For each $g \in G$ the DDH_g problem is defined as follows: Given (g, g^x, g^y, g^z), decide if $xy = z \bmod p$. The DDH_G problem asks to solve the DDH_g problem for arbitrary $g \in G$. We call by the DDH_g oracle a probabilistic algorithm to give a decision of the DDH_g problem. By notation, we set $DDH_g(g, g^x, g^y, g^z) = 1$ if $xy = z \bmod p$ and 0 otherwise.*

Notation. Suppose A and B are problems. Throughout this paper, '$A \longrightarrow B$' means that A is a strong problem than B, that is if there is a polynomially-bounded algorithm $\mathcal{A}_A$ solving the problem A then we can build another polynomially-bounded algorithm $\mathcal{A}_B$ with polynomially-bounded access to $\mathcal{A}_A$ which solves the problem B. '$A \longleftrightarrow B$' means '$A \longrightarrow B$' and '$B \longrightarrow A$'. It is trivial that $DL_g \longrightarrow DH_g$ and $DH_g \longrightarrow DDH_g$.

2.2 Problems related to the BDH

A map $e : G \times G \to H$ is said to be *bilinear* provided that $e(g_1^{x_1}, g_2^{x_2}) = e(g_1, g_2)^{x_1 x_2}$ for all $x_i \in \mathbb{Z}/p\mathbb{Z}$ and $g_i \in G$. We denote $\mathbb{Z}/p\mathbb{Z}$ by $\mathbb{Z}_p$. The Weil pairing for an elliptic curve is a good example of a bilinear map from an elliptic curve to a finite field. In this paper, we assume that the bilinear map e has the following properties for practical purposes:

1. Non-degenerate: There exists a $g \in G$ such that $e(g, g) \neq 1_H$.
2. Efficient computable: There is an efficient algorithm to compute $e(g_1, g_2)$ for any $g_1, g_2 \in G$.

In fact, the original Weil pairing does not satisfy the non-degeneracy, but a modified Weil pairing defined over supersingular curve has the above properties. A modified Weil paring is described in [4].

Definition 4. *For each $g \in G$ the BDL_g problem is defined as follows: Given (g, g^x, g^y), compute t such that $e(g^x, g^y) = e(g, g)^t$. The BDL_G problem asks to solve the BDL_g problem for arbitrary $g \in G$. We call by the BDL_g oracle a probabilistic algorithm to give a solution of the BDL_g problem. By notation, we set $BDL_g(g, g^x, g^y) = xy$.*

Definition 5. *For each $g \in G$ the BDH_g problem is defined as follows: Given (g, g^x, g^y, g^z), compute $e(g, g)^{xyz}$. The BDH_G problem asks to solve the BDH_g problem for arbitrary $g \in G$. We call by the BDH_g oracle a probabilistic algorithm to give a solution of the BDH_g problem. By notation, we set $BDH_g(g, g^x, g^y, g^z) = e(g, g)^{xyz}$.*

Definition 6. *For each $g \in G$ the $DBDH_g$ problem is defined as follows: Given (g, g^x, g^y, g^z, h^w) where $h = e(g, g)$, decide if $xyz = w \bmod p$. The $DBDH_G$ problem asks to solve the $DBDH_g$ problem for arbitrary $g \in G$. We call by the $DBDH_g$ oracle a probabilistic algorithm to give a solution of the $DBDH_g$ problem. By notation, we set $DBDH_g(g, g^x, g^y, g^z, h^w) = 1$ if $xyz = w \bmod p$ and 0 otherwise.*

Remark. It is easy to show that the DL_g problem is equivalent to the BDL_g problem. Clearly, $BDL_g \longrightarrow BDH_g$ and $BDH_g \longrightarrow DBDH_g$ hold.

3 Relations between DH-related Problems

We defined three DH-related problems. Roughly speaking, the DL problem is equivalent to the DH problem, but the DDH problem is different from either one on most cyclic groups. We briefly summarize known results on these relations [11, 12]. Moreover we will present a useful lemma on relations between problems according to granularity.

DL versus DH. It is trivial that the DH problem is solved using the DL oracle. For the converse, Maurer and Wolf showed the DL problem is reduced to the DH problem for some special class of groups [11, 12]. Let p be an order of a group G. According to their result, if one can construct an elliptic curve over $\mathbb{F}_p$ such that the elliptic curve variant DL problem on the curve is feasible (such curves always exist e.g. anomalous curves and smooth curves), then the DL problem can be solved by $O(\log^3 p)$ calls DH oracle.

Unfortunately it is not clear how to construct an elliptic curve over $\mathbb{F}_p$ which has a given order. There are a few classes of curves, whose order is known, such as supersingular curves and curves with complex multiplications. For convenience, we define the following condition on a prime p.

Condition $(*)$**.** There exists an elliptic curve over $\mathbb{F}_p$ on which the DL problem is solvable.

Consequently, if G has an order p satisfying the condition $(*)$, then the DL problem is equivalent to the DH problem.

DH versus DDH. Many cryptographic protocols rely on the hardness of the DDH problem for their security. However the DDH problem is not difficult for some groups, especially for supersingular elliptic curves [3]. Moreover Joux and Nguyen constructed elliptic curve groups where the DDH problem is easy while the DH problem is equivalent to the DL problem in [10].

Actually, if there is an efficient computable bilinear map $e : G \times G \to H$, then the DDH problem on G is easy. For given (g, g^x, g^y, g^z), we can decide whether $z = xy \mod p$ by checking the equality between $e(g, g^z)$ and $e(g^x, g^y)$. For example, the DDH problem on supersingular curves is easy since the modified Weil pairing for supersingular curves is an efficient computable bilinear map.

Granularity of DL and DH.

Lemma 1. *Let G be a cyclic group of prime order p and $g \in G$. We have $DL_G \longleftrightarrow DL_g$ and $DH_G \longleftrightarrow DH_g$. More precisely, for any $g_1 \in G$ we have*

1. *If we have a DL_g oracle with success probability ϵ, we can build a DL_{g_1} oracle with success probability ϵ^2 by two calls of the DL_g oracle.*
2. *If we have a DH_g oracle with success probability ϵ, we can build a DH_{g_1} oracle with success probability $\epsilon^{O(\log p)}$ by $O(\log p)$ calls of the DH_g oracle.*

Proof. Since the DL_G (resp. DH_G) problem implies the DL_g (resp. DH_g) problem, the lemma follows from the two assertions.

1. Suppose that we are given a pair (g_1, g_1^x) for any generator g_1 of G. We can compute $t = DL_g(g, g_1)$ and $y = DL_g(g, g_1^x)$ by two calls of the DL_g oracle. Then $x = yt^{-1} \bmod p$.
2. Suppose that we are given a triple (g_1, g_1^x, g_1^y) for any generator g_1 of G. Let $g_1 = g^t$ for some $t \in \mathbb{Z}_p$. Each of $g^{t^{2i}} = DH_g(g, g^{t^i}, g^{t^i})$ and $g^{t^{i+1}} = DH_g(g, g^{t^i}, g^t)$ can be computed by one call of the DH_g oracle. Hence $g^{t^{-1}} = g^{t^{p-2}}$ requires $O(\log p)$ calls of DH_g oracle. Since $g^{t^2xy} = DH_g(g, g^{tx}, g^{ty})$ and $g_1^{xy} = DH_g(g, g^{t^{-1}}, g^{t^2xy})$, g_1^{xy} can also be computed by $O(\log p)$ calls.

Consequently, we can consider the DL problem or the DH problem without fixing a generator. However, we do not know whether the DDH_g oracle can be used to build a DDH_{g_1} oracle for another $g_1 \in G$. Let $g_1 = g^t$ for some $t \neq 1$. The $DDHP_g(g, g_1^x, g_1^y, g_1^z)$ decides only whether $tx \cdot ty \equiv tz \mod p$, hence this information does not helpful to decide whether $xy \equiv z \mod p$ without knowing the t value.

4 Relations between BDH-related Problems with a Bilinear Map

We investigate relations of BDH-related problems when we have a bilinear map. Moreover we consider the situation the bilinear map is strong or weak-invertible. The notions will be defined later.

4.1 With a Bilinear Map

Suppose we have an efficiently computable non-degenerate bilinear map $e : G \times G \to H$.

Theorem 1. *Let G and H be cyclic groups of prime order p. We have $DL_h \longrightarrow DL_g$ and $DH_h \dashrightarrow DH_g$.*[3] *More precisely, for any $g_1 \in G$ we have*

1. *If we have a DL_h oracle with success probability ϵ, we can build a DL_g oracle with success probability ϵ by one call of the DL_h oracle.*
2. *Assume that p satisfies the condition $(*)$. If we have a DH_h oracle with success probability ϵ, we can build a DH_g oracle with success probability $\epsilon^{O(\log^3 p)}$ by $O(\log^3 p)$ calls of the DH_h oracle.*

[3] Throughout this paper, $A \dashrightarrow B$ means that the problem A implies the problem B if p satisfies the condition $(*)$.

Proof. 1. Since a given pair $(g, g^x) \in G \times G$ is reduced to a pair $(h, h^x) \in H \times H$ via the bilinear map, it is trivial that the first part of the theorem follows.

2. Since p satisfies the condition $(*)$, we may assume that we are able to solve the discrete logarithm on H with help of DH_h oracle. Suppose that we are given a triple (g, g^x, g^y). Let $h = e(g, g)$, $h_1 = e(g, g^x)$ and $h_2 = e(g, g^y)$. Actually $h_1 = h^x$ and $h_2 = h^y$. By the assumption, we can compute the discrete logarithm of h_1 and h_2 as described in [11, 12] by $O(\log^3 p)$ calls of the DH_h oracle. Therefore we have g^{xy}.

We don't know whether the converse of Theorem 1 holds in general. However we will show in Theorem 5 that the converse is true if an inverse image of the bilinear map is efficiently computable for any element of H.

BDH versus DH.

Theorem 2. *Let G and H be cyclic groups of prime order p. Let $h = e(g, g)$ for $g \in G$. We have $DH_g \longrightarrow BDH_g$ and $DH_h \dashrightarrow BDH_g$. More precisely, we have*

1. *If we have a DH_g oracle with success probability ϵ, we can build a BDH_g oracle with success probability ϵ by one call of the DH_g oracle.*
2. *If we have a DH_h oracle with success probability ϵ, we can build a BDH_g oracle with success probability ϵ^2 by two calls of the DH_h oracle.*

Proof. 1. Assume that we are given (g, g^x, g^y, g^z) for $g \in G$. We can compute $g^{xy} = DH_g(g, g^x, g^y)$ by one call of the DH_g oracle. Hence we get $e(g, g)^{xyz} = e(g^{xy}, g^z)$.

2. Assume that we are given (g, g^x, g^y, g^z) for $g \in G$. Let $h_1 = e(g, g^x)$, $h_2 = e(g, g^y)$ and $h_3 = e(g, g^z)$. We can compute $h^{xy} = DH_h(h, h_1, h_2)$ by one call of the DH_h oracle. Hence we can get $e(g, g)^{xyz} = h^{xyz} = DH_h(h, h^{xy}, h_3)$ by one more call of DH_h oracle.

DBDH versus DDH. As mentioned before, DDH problem on G has polynomial time complexity since there is a bilinear map. However we don't know whether the DDH problem on H is easy or not in general.

Theorem 3. *Let G and H be cyclic groups of prime order p. Let $h = e(g, g)$ for $g \in G$. We have $DDH_h \longrightarrow DBDH_g$. More precisely, if we have a DDH_h oracle with success probability ϵ, we can build a $DBDH_g$ oracle with success probability ϵ by one call of the DDH_h oracle.*

Proof. Assume that we are given (g, g^x, g^y, g^z, h^w) where $h = e(g, g)$. Let $h_1 = e(g^x, g^y)$ and $h_2 = e(g, g^z)$. We can decide whether $xyz = w \bmod p$ by one call of DDH_h oracle, i.e. $DDH_h(h, h_1, h_2, h^w)$.

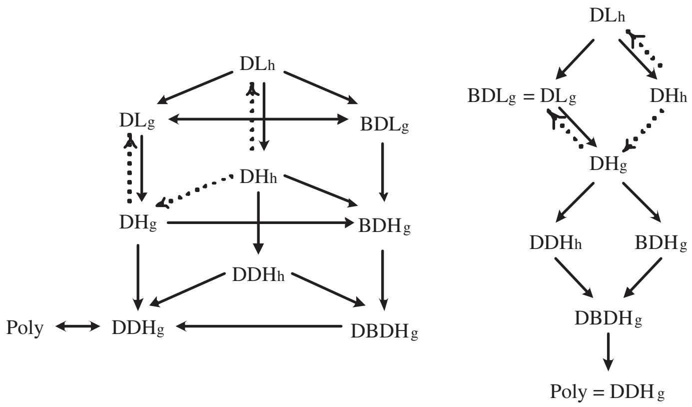

Fig. 1. Problems hierarchy

Problems Hierarchy. We can obtain the following diagram (figure 1) from the above results.

4.2 With a Weak-invertible Bilinear Map

Definition 7. *A bilinear map $e : G \times G \to H$ is said to be weak-invertible provided that an inverse image (g_1, g_2) of h, that is $e(g_1, g_2) = h$, is efficiently computable for any $h \in H$.*

Theorem 4. *Suppose that $e : G \times G \to H$ is weak-invertible and $h = e(g, g)$ for $g \in G$. We have $BDH_g \longrightarrow DH_h$. More precisely, if we have a BDH_g oracle with success probability ϵ, we can build a DH_h oracle with success probability ϵ^2 by two calls of the BDH_g oracle. Consequently $BDH_g \longleftrightarrow DH_h$.*

Proof. Since e is weak-invertible, we can compute (g_0, g_1, g_2, g_3) such that

$$e(g_0, g_1) = h^x, \text{ and } \quad e(g_2, g_3) = h^y.$$

Let $g_0 = g^t$ and $g_i = g_0^{a_i}$ for a positive integer t and a_i's. We can compute $(h^{xy})^{t^{-1}}$ by one call of BDH_g oracle.

$$BDH_g(g, g_1, g_2, g_3) = e(g, g)^{t^3 a_1 a_2 a_3} = (h^{xy})^{t^{-1}}.$$

And let (g_4, g_5) be an inverse image of $(h^{xy})^{t^{-1}}$, i.e. $e(g_4, g_5) = (h^{xy})^{t^{-1}}$. Finally we can compute h^{xy} by one more call of BDH_g oracle:

$$BDH_g(g, g_0, g_4, g_5) = e(g, g)^{t^4 a_1 a_2 a_3} = h^{xy}.$$

Theorem 5. *Suppose that $e : G \times G \to H$ is weak-invertible and $h = e(g,g)$ for $g \in G$. We have $DL_g \longrightarrow DL_h$ and $DH_g \longrightarrow DH_h$. More precisely,*

1. *If we have a DL_g oracle with success probability ϵ, we can build a DL_h oracle with success probability ϵ^4 by four calls of the DL_g oracle. Consequently $DL_g \longleftrightarrow DL_h$.*
2. *if we have a DH_g oracle with success probability ϵ, we can build a DH_h oracle with success probability ϵ^3 by three calls of the DH_g oracle.*

Proof. 1. For given (h, h^x), we compute g_1, g_2, g_3 and g_4 such that $h = e(g_1, g_2)$ and $h^x = e(g_3, g_4)$ since e is weak-invertible. Let $g_i = g^{a_i}$ for some a_i's. We can compute $a_i = DL_g(g, g^i)$ by four calls of DL_g oracle. Hence $x = (a_3a_4)(a_1a_2)^{-1}$.
2. For given (h, h^x, h^y), we compute g_i, $(i = 1, 2, \cdots, 4)$ such that $h^x = e(g_1, g_2)$ and $h^y = e(g_3, g_4)$ since e is weak-invertible. Then we can compute g^x and g^y by two calls of DH_g oracle.
$$DH_g(g, g_1, g_2) = g^x \text{ and } DH_g(g, g_3, g_4) = g^y.$$
From the triple (g, g^x, g^y), we can compute g^{xy} by one more call of DH_g oracle. Therefore we obtain $h^{xy} = e(g, g^{xy})$.

We can obtain the following diagram (figure 2) from the above results.

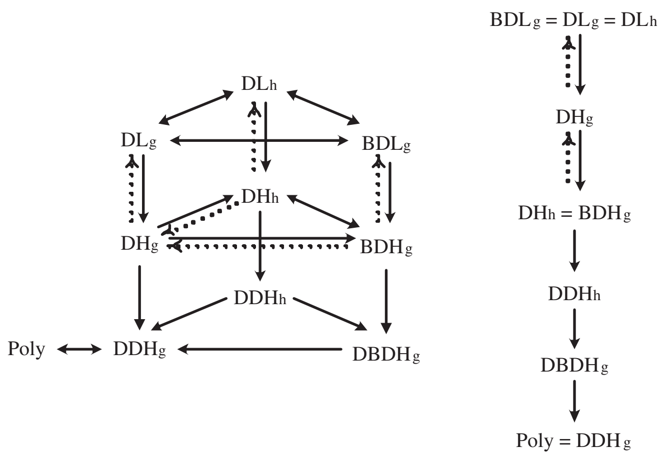

Fig. 2. Problems hierarchy if e is weak-invertible

4.3 With a Strong-invertible Bilinear Map

Definition 8. *A bilinear map $e : G \times G \to H$ is said to be strong-invertible provided that there is an element $g \in G$ such that an inverse g' of h with respect to g, that is $e(g, g') = h$, is efficiently computable for any $h \in H$.*

If e is strong-invertible with respect to $g \in G$ we can easily solve the DH_g problem as follows:

Theorem 6. *If $e : G \times G \to H$ is strong-invertible with respect to $g \in G$, then the DH_g problem is solved using one evaluation of e and one inverse operation of e.*

Proof. For given (g, g^x, g^y), we first compute $e(g^x, g^y) = h_1$ and the inverse image g_1 of h_1 such that $e(g, g_1) = h_1$ by the invertibility of e. Since $h_1 = e(g, g)^{xy} = h^{xy}$, we have $g_1 = g^{xy}$.

If we combine the above result with Lemma 1, we can see that a strong-invertible bilinear map e implies the DH_G problem. Under the condition $(*)$, the DL_G problem is also solved efficiently. More precisely, we have

Corollary 1. *If we have a strong-invertible bilinear map $e : G \times G \to H$, we can solve the DH_G problem. Under the condition $(*)$, we can solve DL_G problem by $O(\log^3 p)$ evaluation of e and inverse operation of e respectively.*

The above result can be extended to more general situation. Suppose we have an efficient computable injective homomorphism $f : H \to G$. By composition of e and f we can construct a self-bilinear map $e_s : G \times G \to G$. That is,

$$e_s(g_1, g_2) = f(e(g_1, g_2)).$$

e_s is clearly an efficient computable non-degenerate bilinear map since f is injective homomorphism.

Lemma 2. *Consider two functions on positive integers:*

$$f(i) = 2i + 1, \qquad g(i) = i + 1.$$

Any k-bit positive integer can be generated from 1 by at most $2(k-1)$ evaluation of f and g.

Proof. Use mathematical induction on k. When $k = 1$, it is trivial. When $k = 2$, it is true since $10_2 = g(1)$ and $11_2 = f(1)$ where the subscript 2 denotes a binary representation. Assume the claim holds for any positive integer of at most $k-1$ bit. Any k-bit integer i has the form $i = 2j$ or $i = 2j + 1$ for a $(k-1)$-bit integer j. Then $2j = g(f(j-1))$ and $2j + 1 = f(j)$. Since both of $j - 1$ and j are at most $(k-1)$-bit integers, they can be computed by at most $2(k-2)$ evaluation of f and g. Thus i can be computed by at most $2(k-1)$ evaluation of f and g which proves the claim for k. By mathematical induction, the claim holds for any positive integer.

Theorem 7. *Let G be a cyclic group of prime order p. If we have an efficiently computable non-degenerate bilinear map $e : G \times G \to G$, we can solve the DH_G problem on G by $O(\log p)$ evaluation of e.*

Proof. Let $g \in G$ and $e(g, g) = g^t$ for a positive integer t. Using Lemma 2, given a triple (g, g^x, g^y), one can compute $g^{t^{-2}} = g^{t^{p-3}}$ by at most $O(\log p)$ times e-computations since $e(g^{t^i}, g^{t^i}) = g^{t^{2i+1}}$ and $e(g^{t^i}, g) = g^{t^{i+1}}$. Therefore we can solve the DH_G as follows.

$$e(e(g^x, g^y), g^{t^{-2}}) = e(g^{txy}, g^{t^{-2}}) = g^{xy}.$$

From the above theorem, we can derive the following corollary.

Corollary 2. *Assume we have a non-degenerate bilinear map $e : G \times G \to H$ for two cyclic group of prime order p. Then there is no injective homomorphism from H to G if the DH_G problem is hard.*

Verheul showed in [17] that if there is an injective homomorphism from the XTR subgroup to the associated supersingular curve, then the homomorphism can be utilized to make an oracle which computes the DH problem over XTR group. The proof technique is similar, but the above corollary gives the same result in more general situation.

5 A Sequence of Bilinear Maps

A bilinear map can be used to tripartite key agreement protocol [9]. More generally an $(n-1)$-multilinear map can be used to construct a non-interactive n-party key agreement protocol. This map also can be used to construct a broadcast encryption with very short broadcasts and private keys and a unique signature scheme [5].

One possible approach to construct a multilinear map is to find a sequence of cyclic groups and bilinear maps between each of consecutive two groups. Suppose that for each positive integer n we have a cyclic group G_n of prime order p with a generator g_n and an efficiently computable, non-degenerate bilinear map $e_n : G_n \times G_n \to G_{n+1}$. For each n-tuple G^n of G, define $f_2 = e_1 : G_1^2 \to G_2$ and

$$f_n : G_1^n \to G_n; f_n(x_1, x_2, \cdots, x_n) = e_{n-1}(f_{n-1}(x_1, x_2, \cdots, x_{n-1}), f_{n-1}(x_n, g, \cdots, g)).$$

Then $f_n : G_1^n \to G_n$ is n-multilinear map for each positive n.

Another application of a family of bilinear maps includes a forward secure Diffie-Hellman key agreement scheme. The notion of the forward secrecy was first introduced by Anderson in 1997 to preserve the security even after the secret key has been exposed [1]. While several forward-secure signature schemes were proposed [2, 8], no forward-secure encryption scheme was announced. Note that a forward-secure encryption scheme is easily followed from a forward-secure Diffie-Hellman key agreement scheme using ElGamal encryption technique or even using symmetric key encryption scheme.

Choose an integer N whose factorization is hard and keep the factorization of N to be secret. Assume we have a family of cyclic groups G_n of order N and efficiently computable, non-degenerate bilinear maps $e_n : G_n \times G_n \to G_{n+1}$ for each positive integer n. Define g_n to be a generator of G_n and $g_{n+1} = e_n(g_n, g_n) \in G_{n+1}$ for each positive integer n.

1. **Setup** Take N and G_n, e_n, g_n for each positive integer n satisfying the above properties.
2. **Initial Keys**. A user randomly takes his initial private key $sk_1 = a$ in $\mathbb{Z}/n$. The initial public key is $pk_1 = g_1^a$.
3. **Private Key on the time** n. $sk_n \equiv a^{2^{n-1}} \mod N$.
4. **Public Key on the time** n. $pk_n = g_n^{a^{2^{n-1}}} \in G_n$
5. **Key Generation on the time** $n+1$. sk_{n+1} is computed by $sk_n^2 \equiv (a^{2^{n-1}})^2 \equiv a^{2^n} \mod N$. pk_{n+1} is computed by $pk_{n+1} = e_{n+1}(pk_n, pk_n) \in G_{n+1}$

Observe that the private key evolving procedure is easy, but the reverse procedure is equivalent to factoring N. The key evolving procedure in this scheme is very efficient since the private key evolving requires one squaring and the public key evolving requires one evaluation of e_n.

If we have an efficiently computable injective homomorphism f from G_j to G_i for $i < j$, by composing it with the bilinear maps e_n for $i \leq n < j$ we can construct a self-bilinear map $e : G_i \times G_i \to G_i$. For example, when $i = 1$ and $j = 4$ we can define $e : G_1 \times G_1 \to G_1$ by

$$e(x, y) = f(e_3(g_3, e_2(g_2, e_1(x, y)))) \quad \text{for some } g_i \in G_i.$$

Since a self-linear map can be used to solve efficiently the DH problem on G_i, our result implies that there should not exist efficiently computable isomorphism between any of two groups in this chain in order to use the family for cryptographic use.

It is not known yet whether such sequence of bilinear map influence on the security of the Diffie-Hellman problem on the base group G_1. But obviously the DDH problem on G_n can be easily solved using a bilinear map $e_n : G_n \times G_n \to G_{n+1}$.

6 Conclusion

We investigated relations between problems related to the DH problem when we have a bilinear map $e : G \times G \to H$. We showed the BDH problem is equivalent to the DH problem on H when the bilinear map is weak-invertible. We do not know if the weak invertibility condition can be weakened. It is interesting to study how the weak-invertible property of a bilinear map influences on the security of the DH problem or the BDH problem.

We also proposed a use of a sequence of bilinear maps. This sequence has more applications than multilinear maps, but it still looks difficult to find such sequence as much as multilinear maps. We pointed out that there should not exist an efficiently computable isomorphism between any of two groups in the family. Hence the first step to construct such sequence is to find a family of groups with the same order such that there does not exist an efficiently computable isomorphism between any two of them.

References

1. R. Anderson, Invited Lecture, *Fourth Annual Conference on Computer and Communications Security*, ACM, 1997.
2. M. Bellare and S. Miner, A Forward-secure Digital Signature Scheme, *Advances in Cryptology - CRYPTO 99*, LNCS 1666, Springer-Verlag, pp.431–448, 1999.
3. D. Boneh, The Decision Diffie-Hellman Problem, *Third Algorithmic Number Theory Symposium*, LNCS 1423, Springer-Verlag, pp.48–63, 1998.
4. D. Boneh and M. Franklin, Identity Based Encryption from the Weil Pairing, *Advances in Cryptology - Crypto 2001*, LNCS 2139, Springer-Verlag, pp.213-229, 2001.
5. D. Boneh and A. Silverberg, Applications of Multilinear Forms to Cryptography, preprint, 2002. Available from `http://eprint.iacr.org`.
6. D. Boneh, B. Lynn and H. Shacham, Short Signatures from the Weil Pairing, *Advances in Cryptology - Asiacrypt 2001*, LNCS 2248, Springer-Verlag, pp.514–532, 2001.
7. J. Cha and J. Cheon, An Identity-based Signature from Gap Diffie-Hellman Groups, preprint, 2002. Available from `http://eprint.iacr.org`.
8. G. Itkis and L. Reyzin, Forward-secure Signatures with Optimal Signing and Verifying, *Advances in Cryptology - CRYPYO 2001*, LNCS 2139, Springer-Verlag, pp.332–354, 2001.
9. A. Joux, A One Round Protocol for Tripartite Diffie-Hellman, *Fourth Algorithmic Number Theory Symposium*, LNCS 1838, Springer-Verlag, pp.385–394, 2000.
10. A. Joux and K. Nguyen, Separating Decision Diffie-Hellman from Diffie-Hellman in Cryptographic Groups, preprint 2001. Available from `http://eprint.iacr.org`.
11. U. Maurer, Towards the Equivalence of Breaking the Diffie-Hellman Protocol and Computing Discrete Logarithms, *Advances in Cryptlogy - CRYPTO 94*, LNCS 839, Springer-Verlag, pp.271–281, 1994.
12. U. Maurer and S. Wolf, Diffie-Hellman Oracles, *Advances in Cryptology - CRYPTO 96*, LNCS 1109, Springer-Verlag, pp.268–282, 1996.
13. A. Menezes, T. Okamoto, and S. Vanstone, Reducing Elliptic Curve Logarithms to Logarithms in a Finite Field, *IEEE Trans. Inform. Theory*, Vol. 39, pp.1639–1646, 1993.
14. K. Paterson, ID-based Signatures from Pairings on Elliptic Curves, preprint, 2002. Available from `http://eprint.iacr.org`.
15. A.-R. Sadeghi and M. Steiner, Assumptions related to Discrete Logarithms: Why Subtleties Make a Real Difference, *Advances in Cryptology - EUROCRYPT 2001*, LNCS 2045, Springer-Verlag, pp.244–261, 2001.
16. N. Smart, Identity-based Authenticated Key Agreement Protocol based on Weil Pairing, *IEE Electronic Letters*, vol.38, pp.630–632, June 2002.
17. E. Verheul, Evidence that XTR is More Secure than Supersingular Elliptic Curve Cryptosystems, *Advances in Cryptology - EUROCRYPT 2001*, LNCS 2045, Springer-Verlag, pp.195–210, 2001.
18. E. Verheul, Self-blindable Credetial Certificates from the Weil Pairing, *Advances in Cryptology - ASIACRYPT 2001*, LNCS 2248, Springer-Verlag, pp.533–551, 2001.