Knapsack Diffie-Hellman: A New Family of Diffie-Hellman

Song Han[1], Elizabeth Chang[1], and Tharam Dillon[2]

[1] School of Information Systems, Curtin Business School
Curtin University of Technology,
GPO Box U1987, WA 6845, Australia
[2] Faculty of Information Technology, University of Technology Sydney,
PO Box 123 Broadway NSW 2007 Australia

Abstract. Diffie-Hellman problems have been widely involved in the design of various cryptographic protocols. Its general family is based on the discrete logarithm over a finite field. Since 2000, its another family which is based on elliptic curve discrete logarithm as well as bilinear pairing, e.g. Weil or Tate pairing, has been attracted significant studies. Thereafter, various cryptographic protocols have been proposed using Diffie-Hellman problem associated with bilinear pairings. This paper we will present a new family of Diffie-Hellman problem by utilizing subset sum problem. It is named as Knapsack Diffie-Hellman Problems with bilinear pairings. We will propose a number of definitions on the family and then analyze their relationships.
Keywords: Diffie-Hellman Problem, Knapsack Problem, Bilinear Pairings, Subset Sum, Cryptographic Protocol.

1 Introduction

Diffie-Hellman problems have been widely involved in the design of various cryptographic protocols. The original Diffie-Hellman key agreement provided the first practical solution to the key distribution problem, allowing two parties, never having met in advance or shared keying material, to establish a shared secret by exchanging messages over an open channel.

The basic approach for Diffie-Hellman key agreement is that if Alice and Bob wish to create a common secret key, they agree on a group $\langle g \rangle$, and then Alice chooses a random integer a; , while Bob chooses a random integer b . Alice then computes g^a and sends it to Bob over a public channel, while Bob computes g^b and sends it to Alice. Now Alice and Bob can both compute $g^{ab} = (g^a)^b = (g^b)^a$ while an eavesdropper who happens to have overheard the exchange, and thus knows g, g^a, and g^b, will hopefully not be able to compute the secret g^{ab}. If the discrete log problem for the group $\langle g \rangle$ is easy, an eavesdropper can compute either a or b, and can find out what g^{ab} is. It is an important open question whether determining g^{ab} knowing just g, g^a, and g^b is as hard as the discrete log problem in general. However, a fast discrete log algorithm would definitely destroy the utility of the widely used Diffie- Hellman protocol [19, 20, 7].

Diffie-Hellman key agreement on its own does not provide authentication of either party, and is therefore vulnerable to a man-in-the-middle attack. Menezes and Blake-Wilson provided an authenticated Diffie-Hellman key agreement protocol in [21].The current secure standard version of Diffie-Hellman key agreement is 'PKCS ♯3 Diffie-Hellman Key Agreement Standard' [27].

Since the Diffie-Hellman key-exchange was proposed, a number of new cryptographic assumptions were studied. Maurer and Wolf studied the relations between the Diffie-Hellman, Decision Diffie-Hellman, and Discrete Logarithms [20]. Shparlinski et al studied the statistical properties of Diffie-Hellman distributions in [26]. Cheon and Lee studied the relations among the Diffie-Hellman problem, the discrete logarithm problem and the bilinear Diffie-Hellman problem [8]. Koh et al designed a new Diffie-Hellman: Braid Diffie-Hellman [17]. It is still interesting to design new family of Diffie-Hellman, that may be used to design new cryptographic protocols.

The Subset Sum Probelm (Kanpack problem) [1, 14, 3] is a problem involving selecting objects with pre-designed weights and profits in such a way that a specified capacity is not exceeded and a specified target profit is attained. It has two different versions: the Decisional version and the Computational (i.e. Search) version. The Decisional version of knapsack problem is NP-complete, and the Computational version is NP-hard [28].

The first Knapsack cryptosystem was proposed by Merkle and Hellman [18]. The Merkle-Hellman kanpsack encryption scheme is important for historical reasons, as it was the first concrete realization of a public-key encryption scheme. Many variations have been proposed but most, including the original, have been later found to be insecure [2, 3, 25, 22–24]. The main contribution of the Merkle-Hellman Knapsack cryptosystem is that it demonstrates how an NP-complete problem can be used for public-key cryptography.

The Chor-Rivest knapsack cryptogsystem is the only knapsack cryptosystem that does not use some form of modular multiplication to disguise an easy knapsack [9]. The attack in [29] only provided special attacks on the system with the recommended parameters. There is no feasible attack known for breaking their cryptosystem, if the parameters of the system are carefully chosen. In particular, the density of the knapsack set $(c_0, c_1, ..., c_{p-1})$ is $p/\lg(maxc_i)$, which is large enough to thwart the low-density attacks on the general subset sum problem [1].

The bilinear pairings, especially modified Weil/Tate pairings have been a useful constructing block for cryptographic protocols since Joux designed a tripartite Diffie-Hellman protocol [16]. Some new Diffie-Hellman problems and a number of cryptographic protocols have been proposed based on pairings [4, 6, 15, 16]. Dutta et al surveyed a great number of cryptographic protocols which were based on pairings [10]. An open problem is whether we can design a new family of Diffie-Hellman that is based on Diffie-Hellman problem with pairings and Knapsack problem. In this paper, we will address this issue.

The motivation of our paper is based on the three points: (1) The Diffie-Hellman related assumptions have played an important role in designing various cryptographic protocols. Apart from the existing Diffie-Hellman assumptions, is

it possible to propose new Diffie-Hellman assumption that will be built upon to design new cryptographic protocols? (2) Knapsack cryptosystems had ever received a great deal of attention in the community of cryptography and computational complexity in 1970s' and 1980s'. Most of the existing Knapsack cryptosystems were broken. One exceptional system is the Chor-Rivest Knapsack cryptosystem, even though some special attacks were proposed. An interesting question is: Has the Knapsack problem already been falling from designing optimistic cryptographic protocols? Is the Knapsack assumption yet/still an useful constructing assumption for public key cryptography? [13]. (3) Pairings over elliptic curves have already been combined with the Diffie-Hellman and thus created a number of Diffie-Hellman assumptions, that have been used to design cryptographic protocols. Is it possible to design a new family of Diffie-Hellman assumptions, that will provide various or optimistic choice for designing cryptographic protocols?

The organization of the rest of the paper is as follows: In the next section, some computational preliminaries are presented. Section 3 provides the definition of Knapsack Diffie-Hellman. Section 4 analyses the relations of the problems within the Knapsack Diffie-Hellman. The last section concludes this paper.

2 Preliminaries

In this section we will review the Diffie-Hellman problems over elliptic curve defined in prime-order groups G_1 and G_2. G_1 is an additive group. G_2 is a multiplicative group. Assume the general discrete logarithm problem is hard in G_1 as well as in G_2. We also review the general subset sum problem, which will be compared with the proposed Knapsack Diffile-Hellman problem in Section 3.

2.1 Pairings over Elliptic Curves

Let G_1 and G_2 be two groups of prime order q. G_1 is an additive group. G_2 is a multiplicative group. Assume the general discrete logarithm problem is hard in G_1 as well as in G_2. A bilinear function $e : G_1 \times G_1 \rightarrow G_2$ is called a bilinear pairing which satisfies the following properties:

(1) Bilinearity: $e(aP, bQ) = e(P, Q)^{ab}$ for every pair $P, Q \in G_1$ and for any $a, b \in Z_p^*$.

(2) Non-degeneracy: there exists at least one point $P \in G_1$ such that $e(P, P) \neq 1$. In other words, if P is a generator of G_1, then $e(P, P)$ is a generator of G_2.

(3) Computability: there are efficient algorithms to compute the bilinear pairing e for every pair $P, Q \in G_1$.

2.2 Elliptic Curve Discrete Logarithms

Definition (Elliptic Curve Discrete Logarithm Problem) *Given P as a generator of G_1, and given xP, where x is an unknown random element of Z_q^*, the Elliptic Curve Discrete Logarithm (ECDL) problem is to find x.*

ECDLP Assumption ECDLP Assumption implies there is no probabilistic polynomial time algorithm to solve the Elliptic Curve Discrete Logarithm problem with non-negligible advantage.

2.3 Diffie-Hellman Problems over Elliptic Curve

Let G_1 and G_2 be two groups of prime order q over elliptic curve . G_1 is an additive group. G_2 is a multiplicative group. Assume the general discrete logarithm problem is hard in G_1 as well as in G_2.

Definition The Computational Diffie-Hellman (CDH) problem is defined as:
Problem Instance: given P, aP, bP, where $a, b \in Z_q^*$.
Output: abP.

CDH Assumption For $a, b \in Z_q^*$, $P \in G_1$, given $\langle P, aP, bP \rangle$, to compute abP is hard.

Definition The Decisional Diffie-Hellman (DDH) problem is defined as:
Problem Instance: given P, aP, bP, cP, where $a, b \in Z_q^*$.
Output: yes, if $c = ab \bmod q$; no, otherwise.

DDH Assumption For $a, b \in Z_q^*$, $P \in G_1$, given $\langle P, aP, bP, cP \rangle$, to determine whether $c = ab \bmod q$ is hard. However, MOV reduction [11, 5] implies that DDH problem is easy in G_1. Fortunately, the DDH Assumption still holds in the multiplicative groups (of large finite fields [20]) in such a way that there is no large prime factor in the order of the groups.

2.4 Subset Sum Problem

Subset sum problem known as Knapsack problem in cryptography [14, 18] is stated as follows:

Definition The Subset Sum (SS) problem is the following:
Problem Instance: given $I = (b_1, b_2, ..., b_n, T)$, where b_i and T are positive integers. The $b_i's$ are called sizes (i.e. weight in Knapsack) and T is called the target sum (i.e. the capacity of Knapsack).
Question: Is there a binary string $x = (x_1, x_2, .., x_n)$ such that

$$\sum_{i=1}^{n} x_i b_i = T \tag{1}$$

Fact: The general SS problem is NP-complete. Its computational version is NP-hard. [14, 28]

Definition The Superincreasing sequence: an integer sequence $(b_1, b_2, ..., b_n)$ is called a superincreasing sequence if

$$b_i > \sum_{j=1}^{i-1} b_j \ , (2 \leq i \leq n) \tag{2}$$

Fact: The computational version of Subset Sum problem with a superincreasing sequence can be solved in time $O(n)$, and a solution x (if it exists) must be unique.

3 Knapsack Diffie-Hellman Problems

The Knapsack problem, as it is usually defined, is a problem involving selecting objects with given weights and profits in such a way that a specified capacity is not exceeded and a specified target profit is attained. This can be mathematically translated into the Subset Sum problem as above.

The well-known Merkle-Hellman Knapsack cryptosystem [18] is important for historical reasons, as it was the first concrete realization of a public-key encryption scheme. Many variations have subsequently been proposed but most, including the original, have been broken [2, 18, 1, 3, 22–25] excluding the knapsack scheme designed by Chor and Rivest [9]. Therefore, it is still interesting to propose the new version of Knapsack problem and the corresponding cryptographic protocols, since it has the conceptual elegance and the desirable computational complexity.

3.1 Definitions of Knapsack Diffie-Hellman

Let G_1 and G_2 be two groups of prime order q. G_1 is an additive group. G_2 is a multiplicative group. Assume the general discrete logarithm problem is hard in G_1 as well as in G_2. G_3 is a set of binary strings, $G_3 = \{0,1\}^n$, where n is a large positive integer. The bilinear pairing e is defined as above in Section 2. The followings are the proposed Knapsack Diffie-Hellman problems:

Definition 1. The Computational Knapsack Diffie-Hellman (CKDH) problem in G_1 is defined as follows:

Given: $P \in G_1$ (P is a generator), $Q \in G_1$, and $(n+1)$-tuple $\langle P_1, P_2, ..., P_n, H\rangle$, and $P_i = b_i P$ $(1 \le i \le n)$, and $H = sQ$; where $b_i \in_R Z^+$ (Z^+ denotes the set of all positive integers), $s \in_R Z_q^*$, and $(b_1, b_2, ..., b_n)$ and s are all unknown elements.

Output: A binary string $x = (x_1, x_2, ..., x_n) \in G_3$, which (provided that such x exists) satisfies

$$e(\sum_{i=1}^{n} x_i P_i, Q) = e(H, P) \tag{3}$$

Definition If there exists $x = (x_1, x_2, ..., x_n) \in G_3$ such that $e(\sum_{i=1}^{n} x_i P_i, Q) = e(H, P)$, then we call $\langle P, Q, P_1, P_2, ..., P_n, H\rangle$ a Knapsack Diffie-Hellman tuple.

Assumption 1. (**CKDH Assumption**) For every probabilistic polynomial time, 0/1 valued algorithm $\mathbf{A}$, the $Succ^{CKDH}_{\mathbf{A},G_3,G_1}$ is negligible. Here

$$Succ^{CKDH}_{A,G_3,G_1} = Pr[\mathbf{A}(P, Q, P_1, P_2, ..., P_n, H) = 1 : P_i = b_i P, H = sQ, \tag{4}$$

$$b_i \in_R Z^+ (1 \leq i \leq n), s \in_R Z_q^*]$$

The success probability of any probabilistic polynomial time, 0/1 valued algorithm $\mathbf{A}$ in solving the CKDH problem in G_1 and G_3, is defined to be $Succ^{CKDH}_{A,G_3,G_1}$ as above.

Remark 1. CKDH Assumption is reasonable, since its computational complexity is comparable with the computational complexity of the Computational Diffie-Hellman problem. (See the following Section 3.2 for details.)

Definition 2. The Decisional Posterior Knapsack Diffie-Hellman (DPKDH) problem is defined as follows:

Given: $P \in G_1$ (P is a generator), $Q \in G_1$, and (n+1)-tuple $\langle P_1, P_2, ..., P_n, H \rangle$, and $P_i = b_i P$ $(1 \leq i \leq n)$, and $H = sQ$; where $b_i \in_R Z^+$ (Z^+ denotes the set of all positive integers), $s \in_R Z_q^*$, and $(b_1, b_2, ..., b_n)$ and s are all unknown elements.

Output: *yes*, if there exists an $x = (x_1, x_2, ..., x_n) \in G_3$ such that

$$\langle P, Q, P_1, P_2, ..., P_n, H \rangle \tag{5}$$

is a Knapsack Diffie-Hellman tuple; Otherwise, output *no*.

Assumption 2. DPKDH Assumption For every probabilistic polynomial time, 0/1 valued algorithm $\mathbf{A}$, the advantage $Adv^{DPKDH}_{\mathbf{A},G_1,G_3}$ of $\mathbf{A}$ is negligible.

The advantage of any probabilistic polynomial time, 0/1 valued algorithm $\mathbf{A}$ in solving DPKDH problem in G_1 and G_3 is defined as follows:

$$Adv^{DPKDH}_{\mathbf{A},G_1,G_3} = |Pr[\mathbf{A}(P, Q, P_1, P_2, ..., P_n, \sum_{i=1}^{n} x_i b_i Q, (x_1, x_2, ..., x_n)) = 1] - \tag{6}$$

$$Pr[\mathbf{A}(P, Q, P_1, P_2, ..., P_n, H, (x_1, x_2, ..., x_n)) = 1] :$$
$$P_i = b_i P, H = sQ, b_i \in_R Z^+ (1 \leq i \leq n), s \in_R Z_q^*,$$
$$x = (x_1, x_2, ..., x_n) \in_R G_3|$$

Remark 2. DPKDH Assumption is reasonable, since the DPKDH problem is computationally equivalent to the CKDH problem, while the CKDH problem's computational complexity is comparable with the computational complexity of the Computational Diffie-Hellman problem.

Definition 3. The Decisional Prior Knapsack Diffie-Hellman (D$\bar{\text{P}}$KDH) problem is defined as follows:

Given: $P \in G_1$ (P is a generator), $Q \in G_1$, and (n+1)-tuple $\langle P_1, P_2, ..., P_n, H \rangle$, and $P_i = b_i P$ $(1 \leq i \leq n)$, and $H = sQ$, and a binary string $x = (x_1, x_2, ..., x_n) \in G_3$; where $b_i \in_R Z^+$ (Z^+ denotes the set of all positive integers), $s \in_R Z_q^*$, and $(b_1, b_2, ..., b_n)$ and s are all unknown elements.

Output: *yes*, if

$$\sum_{i=1}^{n} x_i b_i = s(\text{mod } q) \tag{7}$$

Otherwise, output *no*.

Assumption 3. DP̄KDH Assumption For every probabilistic polynomial time, 0/1 valued algorithm $\mathbf{A}$, the advantage $Adv^{D\bar{P}KDH}_{\mathbf{A},G_1,G_3}$ of $\mathbf{A}$ is negligible. Here

$$Adv^{D\bar{P}KDH}_{\mathbf{A},G_1,G_3} = |Pr[\mathbf{A}(P, Q, P_1, P_2, ..., P_n, \sum_{i=1}^{n} x_i b_i Q, (x_1, x_2, ..., x_n)) = 1] - \tag{8}$$

$$Pr[\mathbf{A}(P, Q, P_1, P_2, ..., P_n, H, (x_1, x_2, ..., x_n)) = 1] :$$

$$P_i = b_i P, H = sQ, b_i \in_R Z^+ (1 \leq i \leq n), s \in_R Z_q^* |$$

The advantage of any probabilistic polynomial time, 0/1 valued algorithm $\mathbf{A}$ in solving the DP̄KDH problem in G_1 is defined to be $Adv^{D\bar{P}KDH}_{\mathbf{A},G_1,G_3}$ as above.

Remark 3. DP̄KDH Assumption does not hold in G_1, since the bilinear pairing e defined over $G_1 \times G_1 \rightarrow G_2$ presents a useful tool to solve it in polynomial time. We will discuss the relationship between the DP̄KDH Assumption and the DPKDH Assumption.

Definition 4. The Computational Superincreasing Knapsack Diffie-Hellman (CSKDH) problem is defined as follows:

Given: $P \in G_1$ (P is a generator), $Q \in G_1$, and $(n+1)$-tuple $\langle P_1, P_2, ..., P_n, H \rangle$, and $P_i = b_i P$ $(1 \leq i \leq n)$, and $H = sQ$; where $b_i \in_R Z^+$ (Z^+ denotes the set of all positive integers), $s \in_R Z_q^*$, and $(b_1, b_2, ..., b_n)$ satisfies the following condition:

$$b_i > \sum_{j=1}^{i-1} b_j \ , (2 \leq i \leq n) \tag{9}$$

In addition, $(b_1, b_2, ..., b_n)$ and s are all unknown elements.

Output: A binary string $x = (x_1, x_2, ..., x_n) \in G_3$, which (provided that such x exists) satisfies

$$e(\sum_{i=1}^{n} x_i P_i, Q) = e(H, P) \tag{10}$$

Assumption 4. CSKDH Assumption For every probabilistic polynomial time, 0/1 valued algorithm $\mathbf{A}$, the $Succ^{CSKDH}_{\mathbf{A},G_3,G_1}$ is negligible.

The success probability of any probabilistic polynomial time, 0/1 valued algorithm $\mathbf{A}$ in solving the CSKDH problem in G_1 and G_3, is defined to be

$Succ^{CSKDH}_{\mathbf{A},G_3,G_1}$ as follows.

$$Succ^{CKDH}_{A,G_3,G_1} = Pr[\mathbf{A}(P,Q,P_1,P_2,...,P_n,H) = 1 : P_i = b_iP, H = sQ, \quad (11)$$
$$b_i \in_R Z^+(1 \le i \le n), s \in_R Z_q^*, b_i > \sum_{j=1}^{i-1} b_j \ (2 \le i \le n)]$$

Remark 4. CSKDH Assumption is reasonable, since the CSKDH problem is at least computationally as hard as the CKDH problem, while the CKDH problem's computational complexity is comparable with the computational complexity of the Computational Diffie-Hellman problem.

Definition 5. The Decisional Posterior Superincreasing Knapsack Diffie-Hellman (DPSKDH) problem is defined as follows:

Given: $P \in G_1$ (P is a generator), $Q \in G_1$, and $(n+1)$-tuple $\langle P_1, P_2, ..., P_n, H\rangle$, and $P_i = b_iP$ $(1 \le i \le n)$, and $H = sQ$; where $b_i \in_R Z^+$ (Z^+ denotes the set of all positive integers), $s \in_R Z_q^*$, and the binary string $(b_1, b_2, ..., b_n)$ satisfies

$$b_i > \sum_{j=1}^{i-1} b_j \ , (2 \le i \le n) \quad (12)$$

Also, $(b_1, b_2, ..., b_n)$ and s are all unknown elements.

Output: *yes*, if there exists an $x = (x_1, x_2, ..., x_n) \in G_3$ such that

$$\langle P, Q, P_1, P_2, ..., P_n, H\rangle \quad (13)$$

is a Knapsack Diffie-Hellman tuple; Otherwise, output *no*.

Assumption 5. DPSKDH Assumption For every probabilistic polynomial time, 0/1 valued algorithm $\mathbf{A}$, the advantage $Adv^{DPSKDH}_{\mathbf{A},G_1,G_3}$ of $\mathbf{A}$ is negligible.

The advantage of any probabilistic polynomial time, 0/1 valued algorithm algorithm $\mathbf{A}$ in solving DPSKDH problem in G_1 and G_3 is defined as

$$Adv^{DPSKDH}_{\mathbf{A},G_1,G_3} = |Pr[\mathbf{A}(P,Q,P_1,P_2,...,P_n,\sum_{i=1}^{n} x_ib_iQ,(x_1,x_2,...,x_n)) = 1] - \quad (14)$$
$$Pr[\mathbf{A}(P,Q,P_1,P_2,...,P_n,H,(x_1,x_2,...,x_n)) = 1] :$$
$$P_i = b_iP, H = sQ, b_i \in_R Z^+(1 \le i \le n), s \in_R Z_q^*,$$
$$x = (x_1,x_2,...,x_n) \in_R G_3, b_i > \sum_{j=1}^{i-1} b_j \ (2 \le i \le n)|$$

Remark 5. DPSKDH Assumption is reasonable, since the DPSKDH problem is computationally equivalent to the DPKDH problem, while the DPKDH problem's computational complexity is comparable with the computational complexity of the CKDH problem.

Definition 6. The Decisional Prior Superincreasing Knapsack Diffie-Hellman (D$\bar{\text{P}}$SKDH) problem is defined as follows:

Given: $P \in G_1$ (P is a generator), $Q \in G_1$, and $(n+1)$-tuple $\langle P_1, P_2, ..., P_n, H\rangle$, and $P_i = b_i P$ $(1 \leq i \leq n)$, and $H = sQ$, and a binary string $x = (x_1, x_2, ..., x_n) \in G_3$; where $b_i \in_R Z^+$ (Z^+ denotes the set of all positive integers), $s \in_R Z_q^*$, and the binary string $(b_1, b_2, ..., b_n)$ satisfies

$$b_i > \sum_{j=1}^{i-1} b_j \ , (2 \leq i \leq n) \tag{15}$$

In addition, $(b_1, b_2, ..., b_n)$ and s are all unknown elements.

Output: *yes*, if

$$\sum_{i=1}^{n} x_i b_i = s(\text{mod } q) \tag{16}$$

Otherwise, output *no*.

Assumption 6. D$\bar{\text{P}}$SKDH Assumption For every probabilistic polynomial time, 0/1 valued algorithm $\mathbf{A}$, the advantage $Adv^{D\bar{P}SKDH}_{\mathbf{A},G_1,G_3}$ of $\mathbf{A}$ is negligible. Here, $Adv^{D\bar{P}SKDH}_{\mathbf{A},G_1,G_3}$ is defined as follows:

$$\begin{aligned} Adv^{D\bar{P}SKDH}_{\mathbf{A},G_1,G_3} = |Pr[\mathbf{A}(P, Q, P_1, P_2, ..., P_n, \sum_{i=1}^{n} x_i b_i Q, (x_1, x_2, ..., x_n)) = 1] - \\ Pr[\mathbf{A}(P, Q, P_1, P_2, ..., P_n, H, (x_1, x_2, ..., x_n)) = 1] : \\ P_i = b_i P, H = sQ, b_i \in_R Z^+ (1 \leq i \leq n), s \in_R Z_q^*, \\ b_i > \sum_{j=1}^{i-1} b_j \ (2 \leq i \leq n)| \end{aligned} \tag{17}$$

It is the advantage of any probabilistic polynomial time, 0/1 valued algorithm algorithm $\mathbf{A}$ in solving D$\bar{\text{P}}$SKDH problem in G_1 and G_3 as above.

Remark 6. D$\bar{\text{P}}$SKDH Assumption does not hold in G_1, since the bilinear pairing e defined over $G_1 \times G_1$ presents a useful tool to solve it in polynomial time. The relationship between the D$\bar{\text{P}}$SKDH Assumption and the DPSKDH Assumption will be discussed in the following.

4 Relations of Knapsack Diffie-Hellman Assumptions

We will study the relations between the subset sum problem and the proposed Knapsack Diffie-Hellman problems. We also study the relations between the various Diffie-Hellman problems including CDH, DDH, CKDH, DPKDH, DP̄KDH, CSKDH, DPSKDH, and DPSKDH in G_1. We will utilize the bilinear pairing $e : G_1 \times G_1 \rightarrow G_2$ within the study.

Theorem 1. Under the condition that the discrete logarithm problem is hard in G_1 and G_2, the CKDH problem is computationally equivalent to the computational version of subset sum (SS) problem given in Section 2.

Proof. As the setting of Definition 1 in Section 3, we can draw the following equivalent relation:

$$e(\sum_{i=1}^{n} x_i P_i, Q) = e(H, P) \tag{18}$$

$$\Leftrightarrow$$

$$e(P, Q)^{\sum_{i=1}^{n} x_i b_i} = e(sQ, P) \tag{19}$$

$$\Leftrightarrow$$

$$e(P, Q)^{\sum_{i=1}^{n} x_i b_i} = e(Q, P)^s \tag{20}$$

$$\Leftrightarrow$$

$$e(P, Q)^{\sum_{i=1}^{n} x_i b_i} = e(P, Q)^s \tag{21}$$

Therefore, $\sum_{i=1}^{n} x_i b_i = s$. This is the general subset sum problem.

Theorem 2. The DPKDH problem is computationally equivalent to the CKDH problem.

Proof. If there is a polynomial time algorithm Ω which can solve the CKDH problem, then there is a polynomial time algorithm Ω' which can solve the DPKDH problem. In fact, given an instance of DPKDH problem $\langle P, Q, P_1, P_2, ..., P_n, sQ\rangle$, Ω' works as follows:

(1) Take this instance as the input of the oracle Ω, then Ω will return a solution $x = (x_1, ..., x_n)$ (provided that such x exists) in polynomial time.

(2) Compute $A = \sum_{i=1}^{n} x_i P_i \in G_1$ in polynomial time.

3. Check whether $e(\sum_{i=1}^{n} x_i P_i, Q) = e(sQ, P)$. If it holds, then output '*yes*'; otherwise, *no*.

(4) Given any instance of the DPKDH problem in G_1, Ω' can success as above the steps (1)-(3) in polynomial time.

If there is a polynomial time algorithm Ω which can solve the DPKDH problem, then there is a polynomial time algorithm Ω' which can solve the CKDH problem. In fact, given an instance of CKDH problem $\langle P, Q, P_1, P_2, ..., P_n, sQ\rangle$. Ω' works as follows:

(1) Take this instance as the input of the oracle Ω, then Ω will check whether there exists an $x = (x_1, ..., x_n)$ (provided that such x exists) such that $e(\sum_{i=1}^{n} x_i P_i, Q) = e(sQ, P)$ in polynomial time.
(2) If such x exists, then output "yes"; otherwise, "no".
(3) Output that x in step (1). Therefore, x is a solution to the instance $\langle P, Q, P_1, P_2, ..., P_n, sQ\rangle$ of CKDH.

Theorem 3. If there is a polynomial time algorithm Ω which can solve the CKDH problem, then there is an algorithm Ω' which can solve the D$\bar{\text{P}}$KDH problem in polynomial time.

Proof. Given an instance of the D$\bar{\text{P}}$KDH problem: $\langle P, Q, P_1, P_2, ..., P_n, sQ, x = (x_1, ..., x_n)\rangle$, Ω' works as follows:
(1) Take $\langle P, Q, P_1, P_2, ..., P_n, sQ\rangle$ as the input of the oracle Ω.
(2) Compute $A = \sum_{i=1}^{n} x_i P_i$ and $B = e(A, Q)$.
(3) Compute $C = e(sQ, P)$ and check whether $C = B$. Output "yes" if $C = B$; otherwise, "no".

The following figure demonstrates the relations among the proposed Knapsack Diffie-Hellman assumptions as well as the Subset Sum problem, the Computational Diffie-Hellman assumption, and the Decisional Diffie-Hellman assumption.

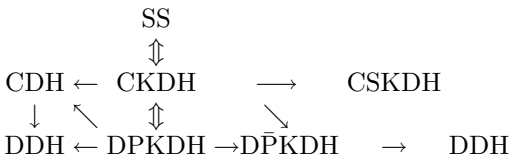


Fig 1. Hierarchical Relations between Diffie-Hellman Problems and Subset Sum Problem. $A \Leftrightarrow B$ means that A and B are computational equivalent. $A \rightarrow B$ means that Problem A is at least as hard as Problem B. In other words, If there is a polynomial time algorithm or oracle Ω to solve A, then B can be solved by a polynomial time algorithm Ω'.

Corollary 4. Under the condition that $x = (x_1, ..., x_n) \in Z_q^{(n)}$ instead of $x = (x_1, ..., x_n) \in \{0, 1\}^n$ in the Definition 1 (See Section 3), if there is a polynomial time algorithm Ω which can solve the CKDH problem, then there exists an algorithm Ω' which can solve the CDH problem in polynomial time.

Corollary 5. If there is a polynomial time algorithm Ω which can solve the DPKDH problem, then there is a polynomial time algorithm Ω' which can solve the D$\bar{\text{P}}$KDH problem.

Corollary 6. If there is a polynomial time algorithm Ω which can solve the DPKDH problem, then there exists an algorithm Ω' which can solve the CDH

and DDH problem in polynomial time.

Corollary 7. If there is a polynomial time algorithm Ω which can solve the DPKDH problem, then there is a polynomial time algorithm Ω' which can solve the DDH problem.

Corollary 8. If there is a polynomial time algorithm Ω which can solve the CKDH problem, then there is a polynomial time algorithm Ω' which can solve the CSKDH problem.

5 Conclusions

In this paper, we have presented a new family of Diffie-Hellman problems by utilizing Knapsack/subset-sum problem. It is named as Knapsack Diffie-Hellman Problem. A number of definitions on the family and the related computational complexity assumptions are studied. We also analyzed their relations between various Diffie-Hellman problems over elliptic curves.

References

[1] L. M. Adleman, *On breaking generalized Knapsack public key cryptosystems,* Proceedings of the 15th ACM Sysposisum on Theory of Computing, pp. 402-412, 1983.

[2] G. Brassard, *A Note on the Complexity of Cryptography,* IEEE Transactions on Information Theory, Vol. IT-25, No. 5, pp. 232-233, 1979.

[3] E.F. Brickell, J.C. Lagarias and A. M. Odlyzko, *Evaluation of the Adleman attack on multiply iterated Knapsack cryptosystems,* Advances in Cryptology - CRYPTO '83 Proceedings, pp. 39-42, Plenum Press, 1984.

[4] A. Boldyreva, *Efficient threshold signature, multisignature and blind signature schemes based on the Gap-Diffie-Hellman-group signature scheme.* PKC 2003, LNCS 2567, Springer-Verlag, 31-46, 2003.

[5] G. Frey, M. Müller and H. Rück, *The Tate pairing and the Discrete Logarithm applied to elliptic curve cryptosystems,* IEEE Transactions on Information Theory 45(5), 1717-1719, 1999.

[6] D.Boneh and M.Franklin, *Identity-based encryption from the Weil pairing,* Proceedings of CRYPTO 2001, Springer-verlag, LNCS 2139, 213-229, 2001.

[7] W. Diffie and M. E. Hellman, *New Directions in Cryptography,* IEEE Transactions on Information Theory, Vol. IT-22, No. 6, pp. 644-654. November 1976.

[8] J. H. Cheon and D. H. Lee, *Diffie-Hellman problems and bilinear maps,* Cryptology ePrint Archive.

[9] B. Chor and R. Rivest, *A Knapsack Type Public Key Cryptosystem based on arithmetic in finite fields,* IEEE Transactions on Information Theory, Vol. 34, 901-909, (1988).

[10] R. Dutta, R. Barua and Palash Sarkar, *Pairing-based cryptographic protocols : a survey,* Cryptology ePrint Archive, Report 2004/064, 2004.

[11] A. Menezes, T. Okamoto and S. Vanstone, *Reducing elliptic curve logarithms to in a finite field,* IEEE Transactions on Information Theory 39(5), 1639-1646, 1993.

[12] S. D. Galbraith, K. Harrison and D. Soldera, *Implementing the Tate pairing,* ANTS-V, Springer-Verlag, LNCS 2369, 324-337, 2002.
[13] S. Han and E. Chang, *Knapsack Diffie-Hellman and subset sum problem,* CBS IS Research Report RR-02-05, Curtin University of Technology, February 2005.
[14] M. E. Hellman, *The Mathematics of Public-Key Cryptography,* Scientific American, Vol. 241, No. 8, pp. 146-157, August 1979.
[15] F.Hess, *Efficient identity based signature schemes based on pairings,* K. Nyberg and H. Heys(Eds.), Selected Areas in Cryptography, SAC 2002, Springer-Verlag, 310-324, 2003.
[16] A.Joux, *A one-round protocol for tripartite Diffie-Hellman,* Algorithm Number Theory Symposium - ANTS-IV, Springer-Verlag, LNCS 1838, 385-394, 2000.
[17] K. Koh, S. Lee, J. Cheon, J. Han, J. Kang and C. Park, *A Public Key Cryptosystem based on Braid Group,* Proc. of Crypto2000, LNCS, Vol. 1880, 166-183, Springer-Verlag, 2000.
[18] R. C. Merkle and M. E. Hellman, *Hiding information and signatures in trapdoor Knapsacks,* IEEE Transactions on Information Theory, Vol. 24, No. 5, pp. 525-530, 1978.
[19] U. Maurer and S. Wolf, *Diffie-Hellman Oracles,* Advances in Cryptology - CRYPTO '96 Proceedings, pp.268-282, Springer-Verlag, 1996.
[20] U. Maurer and S. Wolf, *Diffie-Hellman, Decision Diffie-Hellman, and Discrete Logarithms,* Proceedings of ISIT '98, IEEE Information Theory Society, pp. 327, 1998.
[21] A. Menezes and S. Blake-Wilson, *Authenticated Diffie-Hellman key agreement protocols,* Selected Areas in Cryptography - SAC '98 Proceeding, pp. 339-361, Springer-Verlag, 1999.
[22] A. M. Odlyzko, *Cryptanalytic Attacks on the Multiplicative Knapsack Cryptosystem and on Shamir's Signature Scheme,* IEEE Transactions on Information Theory, Vol. IT-30, pp. 594-601, 1984.
[23] A. M. Odlyzko, *The Rise and Fall of Knapsack Cryptosystems,* Cryptology and Computational Number Theory, Am. Math. Soc., Proc. Symp. Appl. Math., Vol. 42, pp. 75-88, 1990.
[24] A. Shamir, *A Fast Signature Scheme,* MIT/LCS/TM-107, MIT Laboratory for Computer Science, 1978.
[25] A. Shamir, *A Polynomial-Time Algorithm for Breaking the Basic Merkle-Hellman Cryptosystem,* IEEE Transactions on Information Theory, Vol. IT-30, pp. 699-704, 1984.
[26] I. Shparlinski, R. Canetti, J. B. Friedlander, S. Konyagin, M. Larsen and D. Lieman, *On the Statistical Properties of Diffie-Hellman Distributions,* Israel J. Math., Vol.120, pp. 23-46, 2000.
[27] PKCS ♯3: *Diffie-Hellman Key Agreement Standard,* RSA Laboratories - Current draft version of PKCS ♯3 is 1.4. http://www.rsasecurity.com/rsalabs/node.asp?id=2126
[28] D. R. Stison, *Cryptography: theory and practice,* CRC Press, February 2002
[29] S. Vaudenay, *Cryptanalysis of the Chor-Rivest Cryptosystem,* Advances in Cryptology - CRYPTO '98 Proceedings, pp. 243-256, Springer-Verlag, 1998.