Dengguo Feng Moti Yung
Chuankun Wu Dongdai Lin (Eds.)

INFORMATION SECURITY AND CRYPTOLOGY

SKLOIS Conference on Information Security and Cryptology 2005

(short paper proceedings)

Higher Education Press

Dengguo Feng Moti Yung
Chuankun Wu Dongdai Lin (Eds.)

INFORMATION SECURITY AND CRYPTOLOGY

SKLOIS Conference on Information Security and Cryptology 2005

(short paper proceedings)

Higher Education Press

图书在版编目(CIP)数据

信息安全与密码学 = Information Security and Cryptology/冯登国等编.—北京:高等教育出版社,2006.1

ISBN 7-04-018757-4

Ⅰ.信… Ⅱ.冯… Ⅲ.①信息系统-安全技术-国际学术会议-文集-英文②密码-理论-国际学术会议-文集-英文 Ⅳ.①TP309-53②TN918.1-53

中国版本图书馆 CIP 数据核字(2005)第 142128 号

策划编辑 赵天夫 **责任编辑** 赵天夫 **封面设计** 刘晓翔
责任印制 宋克学

出版发行	高等教育出版社	**购书热线**	010-58581118
社　　址	北京市西城区德外大街4号	**免费咨询**	800-810-0598
邮政编码	100011	**网　　址**	http://www.hep.edu.cn
总　　机	010-58581000		http://www.hep.com.cn
		网上订购	http://www.landraco.com
经　　销	蓝色畅想图书发行有限公司		http://www.landraco.com.cn
印　　刷	北京中科印刷有限公司	**畅想教育**	http://www.widedu.com
开　　本	787×1092 1/16	**版　　次**	2006年1月第1版
印　　张	21	**印　　次**	2006年1月第1次印刷
字　　数	490 000	**定　　价**	39.00元

物料号 18757-00

Preface

The first SKLOIS Conference on Information Security and Cryptography (CISC'05) was organized by the State Key Laboratory of Information Security (SKLOIS) of the Chinese Academy of Sciences. It was held in Beijing, China in December 15~17, 2005, and was sponsored by the Institute of Software, the Chinese Academy of Science, the Graduate School of the Chinese Academy of Science, and the National Science Foundations of China.

The international program committee of the conference received a total of 196 submissions (from 21 countries and regions). Each submission was reviewed by around 3 reviewers. Based on the review comments, 33 submissions were selected for presentation as regular papers which are published by Springer in the series of Lectures Notes in Computer Science, and another 32 were selected as short papers which are published in this proceedings. Note that due to the time constraint for paper review, and room limitations for the conference proceedings, many good papers have regrettably been rejected.

Many people and organizations helped in making the conference a reality. We would like to take this opportunity to thank the program committee members and the external experts for their invaluable help in producing the conference program. We thank the various sponsors and, last but not the least, we wish to thank all the authors who submitted papers to the conference, the invited speakers, the session chairs and all the conference attendees.

Finally we would like to note that the SKLOIS Conference on Information Security and Cryptology will be organized annually. We look forward to the continuous support by all the authors, reviewers, sponsors and organizers.

Program Chairs

Dengguo Feng	SKLOIS, Chinese Academy of Sciences, China
Moti Yung	RSA Labs and Columbia University, USA

Program Committee

Dan Bailey	RSA Laboratory, USA
Feng Bao	Institute for Infocomm Research, Singapore
Carlo Blundo	University of Salerno, Italy
Felix Brandt	Stanford University, USA
Ahto Buldas	Tallin Technical University, Estonia
YoungJu Choie	POSTECH, Korea
Zongduo Dai	GSCAS, Chinese Academy of Sciences, China
George Davida	UWM, USA
Ed Dawson	QUT, Australia
Cunsheng Ding	HKUST, China
Keqin Feng	Tsinghua University, China
Keith Frikken	Purdue University, USA
Jun Furukawa	NEC, Japan
Guang Gong	University of Waterloo, Canada
Jiwu Huang	Zhongshan University, China
Kwangjo Kim	ICU, Korea
Xuejia Lai	Shanghai Jiaotong University, China
Dongdai Lin	SKLOIS, Chinese Academy of Sciences, China
Mulan Liu	AMSS, Chinese Academy of Sciences, China
Wenbo Mao	Hewlett-Packard Labs, UK
Tsutomu Matsumoto	Yokohama National University, Japan
Sjouke Mauw	EUT, Netherlands
Bodo Moller	Calgary, Canada
Svetla Nikova	K.U. Leuven, Belgium
Thomas Pornin	Cryptolog, France
Michel Riguidel	ENST, France
Eiji Okamato	Tsukuba University, Japan
Duong Hieu Phan	ENS, France
Bimal Roy	Indian Statistical Institute, India
Ahmad-Reza Sadeghi	Bochum, Germany
Kouichi Sakurai	Kyushu University, Japan
Tom Shrimpton	Portland State University, USA
Willy Susilo	University of Wollongong, Australia
Vijay Varadharajan	Macquarie, Australia
Xiaoyun Wang	Shandong University, China
Chuankun Wu	SKLOIS, Chinese Academy of Sciences, China
Yixian Yang	BUPT, China
Huanguo Zhang	Wuhan University, China
Yuliang Zheng	UNCC, USA
Hong Zhu	Fudan University, China
Yuefei Zhu	Information Engineering University, China

Organizing Committee

Dongdai Lin (Co-chair)	SKLOIS, Chinese Academy of Sciences, China
Chuankun Wu (Co-chair)	SKLOIS, Chinese Academy of Sciences, China
Jiwu Jing	SKLOIS, Chinese Academy of Sciences, China
Wenling Wu	SKLOIS, Chinese Academy of Sciences, China

Table of Contents

Part I

Fundamentals of Cryptography

On the Hidden Number Problem over any Finite Fields of Large Characteristics*

Kewei Lü, Kunpeng Wang, and Bao Li

State Key Laboratory of Information Security,
Graduate University of Chinese Academy of Sciences,
P.O.Box 4588, Beijing 100049, China
conwaylu@tom.com

Abstract. In this paper, making use of the least significant bit and the most significant bits, we study Diffie-Hellman problem over any finite field of large characteristics and prove that hidden number problem with chosen multiplier is as hard as computational Diffie-Hellman problem. Furthermore, we prove the similar results of elliptic curve over any finite field and analyze bit security of tripartite Diffie-Hellman key exchange protocol.

Chinese Subject Classifications code: TP309
Keywords: least (most) significant bit, elliptic curve, Weil pairing

1 Introduction

Discrete logarithm problem (DLP) relative to a base $g \in \mathbf{Z}_p^*$ is to find x given g^x. Assuming this problem to be hard, we recall that Diffie-Hellman key exchange scheme works in the finite cyclic group $\mathcal{G} =< g > \leq \mathbf{Z}_p^*$ of order T. To establish a common key, two communicating parties, *Alice* and *Bob* execute the following protocol [12]: *Alice* chooses a random integer $x \in [1,\ T-1]$, computes and sends $X = g^x$ to *Bob*. *Bob* chooses a random integer $y \in [1,\ T-1]$, computes and sends $Y = g^y$ to *Alice*. Now both *Alice* and *Bob* can compute the common Diffie-Hellman secret$K = Y^x = X^y = g^{xy}$. Many believe that computing Diffie-Hellman function $\mathrm{DH}_g(g^x,\ g^y) = g^{xy}$ is as hard as DLP. After the secret key agreement, *Alice* and *Bob* can secure the session using encryption with a block cipher. A natural way to derive the key for the cipher would be to use a block of bits from g^{xy}. For example, if p is 1024 bit prime, one may use the 64 bit most significant bits of g^{xy}. An attacker, who may not be able to compute the whole g^{xy}, may nevertheless succeed in computing this part of the bits of g^{xy} and crack the session. Hence it is important to know if the most significant bits (MSB) of g^{xy} are secure from an adversary who knows both g^x and g^y. Boneh and Venkatesan [4] prove that computing the most significant bits of the secret key in a Diffie-Hellman key-exchange protocol from the public keys of the players is as hard as computing the secret key itself, by studying the following *hidden number problem*: Given an oracle $\mathcal{O}_\alpha(x)$ that on input x computes the k most significant bits of αg^x mod p, find α mod p.

* Supported by President's Foundation of Graduate University of CAS (yzjj2003010).

On the other hand, the computational Diffie-Hellman assumption (CDH) in group $\mathcal{G}$ states that no efficient algorithm can compute g^{xy} given g, g^x, g^y. But this does not mean that one cannot compute a few bits of g^{xy} or perhaps predict some bits of g^{xy}. In fact, to use the Diffie-Hellman protocol in an efficient system one can usually relies on stronger Decisional Diffie-Hellman assumption (DDH)[2]. Ideally, one would like to show than an algorithm for DDH in group $\mathcal{G}$ implies an algorithm for CDH in $\mathcal{G}$. As a first step, Boneh and Shparlinski [3] show that, in the group of points of an elliptic curve over a finite field, predicting the least significant bit (LSB) of the Diffie-Hellman secret, for many curves in a family of curves, is as hard as computing the entire secret. The similar results were previously known for the RSA function [1] but not for Diffie-Hellman. Most of all work is based on the field $\mathbf{Z}_p$ for a sufficient large prime p.

As applications, a number of cryptographic schemes proposed are related to or based on Diffie-Hellman function $\mathrm{DH}_g(g^x,\ g^y) = g^{xy}$. They depend on the "hidden" nature of g^{xy}. For examples, we refer to ElGamal's public key cryptosystem [5], Shamir's message passing scheme [6], Bellare-Micali non-interactive oblivious transfer [9] and Okamoto conference key sharing scheme [10], etc.

Contribution. Making use of the least significant bit and the most significant bits, we first study the Diffie-Hellman (DH) problem over a general finite field of large characteristics and prove that the hidden number problem with chosen multiplier (HNP-CM) is as hard as computational DH problem. Then we prove the same results of the elliptic curve over the general finite field and analyze the bit security of tripartite DH key exchange protocol.

2 Hidden number problem with trace

2.1 On the most significant bits

Let p be a sufficient large prime, $\lfloor s \rfloor_p$ denote the remainder of an integer s on division by p and $\lceil \log x \rceil$ be the length of x in binary. We use $x \bmod p$ to denote unique integer a in the range $[0,\ p-1]$ satisfying $x \equiv a \pmod p$. Let $\mathbf{F}_p = \mathbf{Z}_p$ be a finite field of p elements and $\mathbf{F}_{p^m}$ be the finite extension of $\mathbf{F}_p$. For an integer x, we define $\|x\|_p = \min_{a \in \mathbf{Z}} |x - ap|$ and for a given $k > 0$, denote by $\mathrm{MSB}_{k,p}(x)$ as the integer u, $0 \le u \le p-1$, such that $\|x-u\|_p \le \frac{p}{2^{k+1}}$. Roughly speaking, a value of $\mathrm{MSB}_{k,p}(x)$ gives the k most significant bits of the residue of x modulo p. We denote by $\mathrm{Tr}(z) = \sum_{i=0}^{m-1} z^{p^i}$ and $\mathrm{Nm}(z) = \prod_{i=0}^{m-1} z^{p^i}$ the trace and norm of $z \in \mathbf{F}_{p^m}$ to $\mathbf{F}_p$ respectively.

HNP-MSB The MSB hidden number problem with trace over a subgroup $\mathcal{G} \subseteq \mathbf{F}_{p^m}^*$ can be formulated as follows: *Given r elements $t_1, \cdots, t_r \in \mathcal{G}$, chosen independently and uniformly at random, the values $MSB_{k,p}(Tr(\alpha t_i))$ for $i = 1, \cdots, r$ and some $k > 0$, recover the number $\alpha \in \mathbf{F}_{p^m}^*$.*

The case of $m = 1$ and $\mathcal{G} = \mathbf{F}_p^*$ corresponds to the hidden number problem introduced in [4], and for the case $\mathcal{G} \subseteq \mathbf{F}_p^*$ see [7]. The case of $m \ge 2$ is more difficult because one of the crucial ingredients, a bound on exponential sums with elements of small subgroups of $\mathbf{F}_{p^m}$, is missing. nevertheless in some special cases results of a comparable strength have been obtained in [8]. In other cases, an alternative method from [11] can be used, leading to weaker results.

We denote by $\mathcal{N}$ the set of $z \in \mathbf{F}_{p^m}$ with norm equal to 1, thus $|\mathcal{N}| = \frac{p^m-1}{p-1}$. The following statement a partial case of Theorem 2 of [8].

Lemma 1. *Let p be a sufficiently large prime and $\mathcal{G}$ be a subgroup of $\mathcal{N}$ of order l with $l \geq p^{(m-1)/2+\rho}$ for some fixed $\rho > 0$. Then for $k = \lceil 2\sqrt{logp} \rceil$ and $r = \lceil 4(m+1)\sqrt{logp} \rceil$, there is a deterministic polynomial time algorithm $\mathcal{A}$ as follows. For any $\alpha \in \mathbf{F}^*_{p^m}$, if $t_1, \cdots, t_r$ are chosen uniformly and independently at random from $\mathcal{G}$ and if $u_i = MSB_{k,p}(Tr(\alpha t_i))$ for $i = 1, \cdots, r$, the output of $\mathcal{A}$ on the $2r$ values (t_i, u_i) satisfies $Pr_{t_1,\cdots,t_r \in \mathcal{G}}[\mathcal{A}(t_1, \cdots, t_r; u_1, \cdots, u_r) = \alpha] \geq 1 - p^{-1}$.*

For smaller groups, a weaker result is given by Theorem 1 of [11].

Lemma 2. *Let p be a sufficiently large prime and $\mathcal{G}$ be a subgroup of $\mathbf{F}^*_{p^m}$ of prime order l with $l \geq p^{\rho}$ for some fixed $\rho > 0$. Then for any $\varepsilon > 0$, let $k = \lceil (1 - \frac{\rho}{m} + \varepsilon) logp \rceil$ and $r = \lceil 4m/\varepsilon \rceil$, there is a deterministic polynomial time algorithm $\mathcal{A}$ as follows. For any $\alpha \in \mathbf{F}^*_{p^m}$, if $t_1, \cdots, t_r$ are chosen uniformly and independently at random from $\mathcal{G}$ and if $u_i = MSB_{k,p}(Tr(\alpha t_i))$ for $i = 1, \cdots, r$, the output of $\mathcal{A}$ on the $2r$ values (t_i, u_i) satisfies $Pr_{t_1,\cdots,t_r \in \mathcal{G}}[\mathcal{A}(t_1, \cdots, t_r; u_1, \cdots, u_r) = \alpha] \geq 1 - p^{-m}$.*

Here we can give a generalization of Lemma 1. We first generalize the HNP-MSB problem to be MSB^d hidden number problem (**HNP-MSB**^{d}). **HNP-MSB**^{d} with trace over a subgroup $\mathcal{G} \subseteq \mathbf{F}^*_{p^m}$ can be defined as follows: *Given r elements $t_1, \cdots, t_r \in \mathcal{G}$, chosen independently and uniformly at random, and the values $MSB_{k,p}(Tr(\alpha t_i^d))$ for $i = 1, \cdots, r$, some $k > 0$ and integer $d > 0$, recover the number $\alpha \in \mathbf{F}^*_{p^m}$.* Obviously, when $d = 1$, it is HNP-MSB. We define $\mathcal{O}_{\mathrm{MSB}^1}$ to be an oracle for $\mathrm{MSB}_{k,p}(\mathrm{Tr}(t))$ for any t.

Lemma 3. *Let p be a sufficiently large prime and $\mathcal{G}$ be a subgroup of $\mathcal{N}$ of order l with $l \geq p^{(m-1)/2+\rho}$ for some fixed $\rho > 0$. Then for $k = \lceil 2\sqrt{logp} \rceil$ and $r = \lceil 4(m+1)\sqrt{logp} \rceil$, given an oracle $\mathcal{O}_{MSB^1}$, there is a deterministic polynomial time algorithm $\mathcal{A}^{\mathcal{O}_{MSB^1}}$ for* **HNP-MSB**^{d} *as follows. For any $\alpha \in \mathbf{F}^*_{p^m}$, if $t_1, \cdots, t_r$ are chosen uniformly and independently at random from $\mathcal{G}$ and make r calls to $\mathcal{O}_{MSB^1}$, the output of $\mathcal{A}^{\mathcal{O}_{MSB^1}}$ on the r values t_i satisfies*

$$Pr_{t_1,\cdots,t_r \in \mathcal{G}}[\mathcal{A}^{\mathcal{O}_{MSB^1}}(t_1, \cdots, t_r) = \alpha] \geq \frac{1}{e^r}(1 - p^{-1}) + (1 - \frac{1}{e^r})p^{-m} \ ,$$

where $e = gcd(d, \ l)$.

Proof. Set $u^d(\lambda) := \mathrm{MSB}^d_{k,p}(\mathrm{Tr}(\alpha\lambda)) := \mathrm{MSB}_{k,p}(\mathrm{Tr}(\alpha\lambda^d))$. Let $R : \mathcal{G} \longrightarrow \mathbf{F}^*_p$ be a random function chosen uniformly from the set of all functions from $\mathcal{G}$ to $\mathbf{F}^*_p$, and $S : \mathcal{G}^d \longrightarrow \mathcal{G}$ be a function satisfying $S(\lambda)^d \equiv \lambda \pmod{p^m}$. Here $\mathcal{G}^d$ is the set of d'th powers in $\mathcal{G}$. The function S is simply a function mapping a d'th power $x \in \mathcal{G}^d$ to a randomly chosen d'th root of x. Next, define the following function $\mathrm{MSB}_{k,p}(\mathrm{Tr}(\alpha\lambda))$:

$$u(\lambda) = \mathrm{MSB}_{k,p}(\mathrm{Tr}(\alpha\lambda)) = \begin{cases} u^d(S(\lambda)), & \text{if } \lambda \in \mathcal{G}^d; \\ R(\lambda), & \text{otherwise.} \end{cases}$$

If $\gcd(d,l) = 1$, then $\mathcal{G}^d = \mathcal{G}$. Choose $t_1, \cdots, t_r$ uniformly and independently at random from $\mathcal{G}$, then $t'_1 = t_1^d, \cdots, t'_r = t_r^d$ is also distributed uniformly and independently in $\mathcal{G}$. Calling the oracle $\mathcal{O}_{\text{MSB}^1}$ on t'_i, we get $u_1^d := u^d(t'_1), \cdots,\ u_r^d := u^d(t'_r)$. For the pairs $(t'_1,\ u_1^d),\ \cdots,\ (t'_r,\ u_r^d)$, by Lemma 1, there is a deterministic polynomial time algorithm $\mathcal{B}$ such that

$$\Pr_{t_1,\cdots,t_r \in \mathcal{G}}[\mathcal{B}(t'_1, \cdots, t'_r; u_1^d, \cdots, u_r^d) = \alpha] \geq 1 - p^{-1} .$$

Now we define a algorithm $\mathcal{A}^{\mathcal{O}_{\text{MSB}^1}}$ to call the oracle $\mathcal{O}_{\text{MSB}^1}$ and algorithm $\mathcal{B}$, then $\mathcal{A}^{\mathcal{O}_{\text{MSB}^1}}(t_1, \cdots, t_r) = \mathcal{B}(t'_1, \cdots, t'_r; u_1^d, \cdots, u_r^d)$. So $\mathcal{A}^{\mathcal{O}_{\text{MSB}^1}}$ is a deterministic polynomial time algorithm satisfying

$$\Pr_{t_1,\cdots,t_r \in \mathcal{G}}[\mathcal{A}^{\mathcal{O}_{\text{MSB}^1}}(t_1, \cdots, t_r) = \alpha] \geq 1 - p^{-1} .$$

If $\gcd(d,\ l) = e > 1$, then $\mathcal{G}^d = \mathcal{G}^e$ and $|\mathcal{G}^e| = \frac{l}{e}$. Similar to the above, we have a algorithm $\mathcal{A}^{\mathcal{O}_{\text{MSB}^1}}$ such that

$$\Pr_{t_1,\cdots,t_r \in \mathcal{G}}[\mathcal{A}^{\mathcal{O}_{\text{MSB}^1}}(t_1, \cdots, t_r) = \alpha] = \tfrac{1}{e^r}\Pr_{t_1,\cdots,t_r \in \mathcal{G}}[\mathcal{B}(t'_1, \cdots, t'_r; u_1^d, \cdots, u_r^d) = \alpha]$$
$$+(1 - \tfrac{1}{e^r})\Pr_{t_1,\cdots,t_r \in \mathcal{G}}[\mathcal{B}(t'_1, \cdots, t'_r; u_1^d, \cdots, u_r^d) = \alpha] \geq \tfrac{1}{e^r}(1 - p^{-1}) + (1 - \tfrac{1}{e^r})p^{-m} .$$

This completes the proof. □

2.2 On the least significant bit

We denote by $\text{LSB}(z)$ the least significant bit of an integer $z \geq 0$. When $z \in \mathbf{F}_p$, we let $\text{LSB}(z)$ be $\text{LSB}(x)$ for the unique integer $x \in [0,\ p-1]$ such that $x \equiv z \bmod p$. Now we define the following variant of the Hidden Number Problem (HNP) presented in [4].

HNP-CM^d: Fix an integer $d > 0$ and an $\varepsilon > 0$. Let p be a prime. For an $\alpha \in \mathbf{F}_p^*$, let $L^{(d)} :\ \mathbf{F}_p^* \longrightarrow \{0,\ 1\}$ be a function satisfying

$$\Pr_{t \in \mathbf{F}_p^*}[L^{(d)}(t) = \text{LSB}(\lfloor \alpha t^d \rfloor_p)] \geq \frac{1}{2} + \varepsilon . \qquad (*)$$

The **HNP-CM^d** problem is: *Given an oracle for $L^{(d)}(t)$, find α in polynomial time.* For small ε there might be multiple α satisfying condition $(*)$ (polynomially many in ε^{-1}). In this case the list-**HNP-CM^d** problem is to find all such that $\alpha \in \mathbf{F}_p^*$. Note that it is easy to verify that a given α belongs to the list of solutions by picking polynomially many random samples $x \in \mathbf{F}_p$ (say, $O(1/\varepsilon^2)$ samples suffice) and testing that $L^{(d)}(x) = \text{LSB}(\lfloor \alpha x^d \rfloor_p)$ holds sufficiently often. We usually set $d = 1,\ 2$ or 3. We refer to the above problem as **HNP-CM^d** to denote the fact that we are free to evaluate $L^{(d)}(t)$ at any multiplier t of our choice (the **CM** stands for Chosen Multiplier). When $d = 1$, it is the well-known algorithm (ACGS algorithm) due to Alexi, Chor, Goldreich, and Schnorr [1]. In the original **HNP** studies in [4] one is only given samples $(t,\ L^1(t) = L(t))$ for random t. The following result shows how to solve the **HNP-CM^d** problem for any $\varepsilon > 0$. The proof of it can be found in [1] and [3].

Lemma 4. *Fixed an integer $d > 0$. Let p be a n-bit prime and let $\varepsilon > 0$. Then given ε, the* **HNP-CM**^{d} *problem can be solved in expected polynomial time in logp and d/ε.*

Notice. In particularly, in the following we usually set $\alpha = \mathrm{MSB}_{k,p}(\mathrm{Tr}(ut_i))$.

3 Security of bits of Diffie-Hellman scheme over a finite field

In this section, we make use of solutions, as shown in the preceding, to HNP of the most significant bits and the least significant bit respectively to prove that predicting the LSB of Diffie-Hellman secret is as hard as solving computational Diffie-Hellman problem. The following result show us that predicting LSB is not easier than trying to find MSB. We define $\mathcal{O}^L$ to be an oracle for $L^{(1)}(t) := L(t)$.

Theorem 1. *Given oracle $\mathcal{O}^L$ and a sufficient large prime p. Then, given $\varepsilon > 0$, HNP-MSB can be solved in expected polynomial time $m \cdot T(logp, \frac{1}{\varepsilon})$, where T is a fixed polynomial and m is the degree of extension of finite fields as above.*

Proof. For $\alpha \in \mathbf{F}^*_{p^m}$, we choose independently and uniformly at random r elements $t_1, \cdots, t_r \in \mathcal{G}$. By Lemma 1, if we find the values $u_i = \mathrm{MSB}_{k,p}(\mathrm{Tr}(\alpha t_i))$ for $i = 1, \cdots, r$ and some $k > 0$, then we can recover the number $\alpha \in \mathbf{F}^*_{p^m}$ by a deterministic polynomial time algorithm $\mathcal{A}$, such that

$$\Pr_{t_1,\cdots,t_r \in \mathcal{G}}[\mathcal{A}(t_1, \cdots, t_r; u_1, \cdots, u_r) = \alpha] \geq 1 - p^{-1} .$$

So we get the result.

Now we try to determine u_i for $i = 1, \cdots, r$. By the definition, we know that $0 \leq u_i \leq p - 1$. For any i and any $t \in \mathbf{F}^*_p$, we do with $L(t) = \mathrm{LSB}(\lfloor u_i t \rfloor_p)$. Making use of the ACGS algorithm (the case $d = 1$ of Lemma 3) and oracle $\mathcal{O}^L$, u_i could be found in expected polynomial time in $n = \log p$ and $\frac{1}{\varepsilon}$. After repeating the procedure r times, we could find $u_1, \cdots, u_r$ in expected polynomial time $rT'(n, \frac{1}{\varepsilon})$ for a fixed polynomial T'. This completes the proof. □

In the following, given (g, g^x, g^y), we show that if there is an efficient algorithm for predicting the LSB of $\mathrm{Tr}(g^{ab})t$ for $t \in \mathbf{F}^*_p$, then there is an algorithm for computing the Diffie-Hellman function, i.e., finding g^{ab}.

Corollary 1. *Given (g, g^x, g^y) for $g \in \mathbf{F}^*_{p^m}$, if there is an efficient algorithm for predicting LSB($Tr(g^{ab})t$) for $t \in \mathbf{F}^*_p$, then there is an algorithm for computing the Diffie-Hellman function, i.e., finding g^{ab} in expected polynomial time.*

Proof. It is easily to get the result by Theorem 3.1. □

Remark 1. For smaller group, by Lemma 2, we can get similar results. these results also show that solving the hidden number problem with chosen multiplier (HNP-CM) is as hard as computing DH function. Furthermore, we know that if computing DH function is hard, then the least significant bits are unpredictable.

4 Security of bits of elliptic curve Diffie-Hellman scheme

In this section, we discuss the relations between Diffie-Hellman problem and LSB over elliptic curves. Let $\mathbf{E}$ be an elliptic curve over finite field $\mathbf{F}_{p^m}$ of size p^m, which is the finite extension of $\mathbf{F}_p$, given by an affine Weierstrass equation of the form

$$Y^2 = X^3 + AX + B, \quad 4A^3 + 27B^2 \neq 0 . \tag{4.1}$$

It is well known that the set $\mathbf{E}(\mathbf{F}_{p^m})$ of $\mathbf{F}_{p^m}$-rational points of $\mathbf{E}$ form an Abelian group under an appropriate composition rule and with the point at infinity O as the neutral element.

Let $G \in \mathbf{E}$ be a point of order q for some prime q. Then the common key established at the end of the Diffie-Hellman protocol with respect to the curve $\mathbf{E}$ and the point G is $abG = (x,\ y) \in \mathbf{E}$ for some integers $a, b \in [1,\ q-1]$. Throughout the rest, we use the fact that the representation of $\mathbf{E}$ contains the field of definition of $\mathbf{E}$. With the convention, an algorithm given the representation of $\mathbf{E}/\mathbf{F}_{p^m}$ as input does not need to also be given p^m and p. The algorithm obtains p^m and p from the representation of $\mathbf{E}$.

Diffie-Hellman Function: Let $\mathbf{E}$ be an elliptic curve over $\mathbf{F}_{p^m}$ and let $G \in \mathbf{E}$ be a point of prime order q. We define Diffie-Hellman function as: $\mathrm{DH}_{\mathbf{E},G}(aG,\ bG) = abG$, where a, b are integers in $[1,\ q-1]$. The Diffie-Hellman problem on $\mathbf{E}$ is to compute $\mathrm{DH}_{\mathbf{E},G}(P,\ Q)$ given $\mathbf{E}$, P, G and Q. Usually, we mostly focus on curves in which Diffie-Hellman problem is believed to be hard. Throughout we say that a randomized algorithm $\mathcal{A}$ computes the Diffie-Hellman function if $\mathcal{A}(\mathbf{E},\ G,\ aG,\ bG) = abG$ holds with probability at least $1 - 1/p^m$. The probability is over the random bits used by $\mathcal{A}$.

Twists on elliptic curves: Let $\mathcal{G}$ be a subgroup of $\mathbf{F}_{p^m}$ with $|\mathcal{G}| \geq p^\rho$ for $\rho > 0$. For any $\lambda \in \mathcal{G}$, define $\phi_\lambda(\mathbf{E})$ to be the twisted elliptic curve:

$$Y^2 = X^3 + A\lambda^4 X + B\lambda^6, \quad 4(A\lambda^4)^3 + 27(B\lambda^6)^2 \neq 0 . \tag{4.2}$$

Hence, $\phi_\lambda(\mathbf{E})$ is an elliptic curve for any $\lambda \in \mathcal{G}$. Throughout this section, we are working with the family of curves $\{\phi_\lambda(\mathbf{E}_0)\}_{\lambda\in\mathcal{G}}$ associated with a given curve $\mathbf{E}_0$. It is easy to verify that for any point $P = (x, y) \in \mathbf{E}$ and any $\lambda \in \mathcal{G}$ the point $P_\lambda = (x\lambda^2, y\lambda^3) \in \phi_\lambda(\mathbf{E})$ (see [3]). Moreover, for any points $P, Q, R \in \mathbf{E}$ with $P+Q = R$ we also have $P_\lambda + Q_\lambda = R_\lambda$. In particular, for any $G \in \mathbf{E}$ we have: $xG_\lambda = (xG)_\lambda$, $yG_\lambda = (yG)_\lambda$, $xyG_\lambda = (xyG)_\lambda$. So map $\phi_\lambda : \mathbf{E} \longrightarrow \phi_\lambda(\mathbf{E})$ mapping $P \in \mathbf{E}$ to $P_\lambda \in \phi_\lambda(\mathbf{E})$. Indeed, it is easy to verify that ϕ_λ is an isomorphism of groups. So we also have $\mathrm{DH}_{\phi_\lambda(\mathbf{E}),G_\lambda}(P_\lambda, Q_\lambda) = \phi_\lambda[\mathrm{DH}_{\mathbf{E},G}(P, Q)]$, i.e. if the Diffie-Hellman function is hard to compute in $\mathbf{E}$ then it is also hard to compute for all curves in $\{\phi_\lambda(\mathbf{E})\}_{\lambda\in\mathcal{G}}$.

For any $z \in \mathbf{F}_p$, we let $\mathrm{LSB}(z)$ be $\mathrm{LSB}(x)$ for the unique integer $x \in [0,\ p-1]$ such that $x \equiv z \bmod p$. We say that an algorithm $\mathcal{A}$ has advantage ϵ in predicting the LSB of the trace of the x-coordinate of the Diffie-Hellman function on $\mathbf{E}$ if:

$$\mathrm{Adv}^X_{\mathbf{E},G}(\mathcal{A}) = |\mathrm{Prob}_{a,b}[\mathcal{A}(\mathbf{E}, G, aG, bG) = \mathrm{LSB}(\mathrm{MSB}_{k,p}(\mathrm{Tr}(x)))] - \frac{1}{2}| > \epsilon ,$$

where $abG = (x,\ y) \in \mathbf{E}$, $k = \lceil 2\sqrt{\log p} \rceil$ and a, b are chosen uniformly at random in $[1,\ q-1]$. We write $\mathrm{Adv}^X_{\mathbf{E},G}(\mathcal{A}) > \epsilon$. Similarly, we say that algorithm $\mathcal{A}$ has advantage