



# 现代密码学

## Modern Cryptography

张方国

中山大学计算机学院

Office: Room 305, IM School Building

E-mail: [isszhfg@mail.sysu.edu.cn](mailto:isszhfg@mail.sysu.edu.cn)

HomePage: <https://cse.sysu.edu.cn/content/2460>





# 第二十讲 DLP (三)

## 第6章 公钥密码学和离散对数

- 实际中的离散对数算法
- 离散对数的比特安全性
- **ElGamal**体制的安全性





# 实际中的离散对数算法

下列 $(G, \alpha)$ 情形中的离散对数问题是最为重要的:

- $G = (Z_p^*, \cdot)$ ,  $p$ 是素数,  $\alpha$ 是模 $p$ 的一个本原元。
- $G = (Z_p^*, \cdot)$ ,  $p, q$ 是素数,  $p \equiv 1 \pmod q$ ,  $\alpha$ 是 $Z_p$ 中的一个 $q$ 阶元素。
- $G = (F_{2^n}^*, \cdot)$ ,  $\alpha$ 是 $F_{2^n}^*$ 中的一个本原元素。
- $G = (E, +)$ , 其中 $E$ 是模素数 $p$ 的一个椭圆曲线,  $\alpha \in E$ 是一个具有素数 $q = \#E/h$ 阶的点, 这里 $h = 1, 2$ 或 $4$ 。
- $G = (E, +)$ , 其中 $E$ 是有限域 $F_{2^n}$ 上的椭圆曲线,  $\alpha \in E$ 是一个具有素数 $q = \#E/h$ 阶的点, 这里 $h = 2$ 或 $4$ 。

对情形1、2和3, 利用 $(Z_p^*, \cdot)$ 或 $(F_{2^n}^*, \cdot)$ 上的指数演算法的适当形式可能形成攻击。对于情形2、4和5, 利用 $q$ 阶子群中的Pollard  $\rho$ 算法可能形成攻击。



# 实际中的离散对数算法

安全性结果 Lenstra和Verheul估计的相对安全性结果。

要使基于椭圆曲线离散对数的密码体制保持安全到2020年，对情形4，建议取 $p \approx 2^{160}$  (或者在情形5中取 $n = 160$ )。

而对于情形1和2，要达到相同程度的安全水平， $p$ 至少应取 $2^{1880}$ 。

原因：主要是对椭圆曲线离散对数没有已知的指数演算法攻击。







# Recent 10 years work on ECDLP

- Weil Descent+ Summation(Index calculus )

A Joux, V Vitse(2012), **151ecdip over GF(p<sup>6</sup>)**;

J.-C. Faugère et.al. EUROCRYPT 2012: **Improving the Complexity of Index Calculus Algorithms Elliptic Curves over Binary Fields**

Semaev I.2015, New algorithm for ECDLP

Huang-Kosters-Yeo2015, **First or Last fall degree need more research!**

**Steven Galbraith**-Pierrick Gaudry 2016

**ECDLP is in time  $2^{c\sqrt{n}\ln n}$ !**

We believe that elliptic curves over characteristic 2 fields of prime degree  $n$  are not threatened by such methods



- Improving Square Root Attacks

**Improved Baby-step Giant-step Algorithm :**

Bernstein-Lange grumpy giants(2013); Steven Galbraith, Ping Wang and Fangguo Zhang(2017) new speed up; Cheon J.H (2010) DLP with auxiliary inputs.

**Pollard rho Algorithm:**

Bernstein-Lange(2013); Hong J., Lee H.(2015) pre-computing; Fangguo Zhang and Ping Wang(2013) Point Halving.

Fangguo Zhang and Shengli Liu(2019), List Decoding.

**There is no practical attack on random curve on GF (2<sup>n</sup>) and GF(p)!**

**ECDLP on GF(p) is more secure than that on GF (2<sup>n</sup>) !**



# ECDLP record

- 2009.07, EPFL, SECp-112

Joppe W. Bos, Marcelo E. Kaihara, Thorsten Kleinjung, Arjen K. Lenstra and Peter L. Montgomery, **Solving a 112-bit Prime Elliptic Curve Discrete Logarithm Problem on Game Consoles using Sloppy Reduction**, International Journal of Applied Cryptography, 2(3), 2012, pp. 212-228

- SAC2014

**Solving the Discrete Logarithm of a 113-bit Koblitz Curve with an FPGA Cluster**

Erich Wenger and Paul Wolfger

Graz University of Technology  
Institute for Applied Information Processing and Communications  
Inffeldgasse 16a, 8010 Graz, Austria

Erich Wenger and Paul Wolfger:  
**ECC2-113**,  
<https://eprint.iacr.org/2015/143>

- ECDLP<sub>2</sub>-117(over **GF** ( $2^{127}$ )):

Bernstein, D. J., Engels, S., Lange, T., Niederhagen, R., Paar, C., Schwabe, P., Zimmermann, R.: Faster discrete logarithms on FPGAs. <http://eprint.iacr.org/2016/382>

- **Breaking ECC<sub>2</sub>K-130??**

**From 2012 to now:** <http://ecc-challenge.info>

**ECC<sub>2</sub>K-130 on NVIDIA GPUs** Daniel J. Bernstein and Hsieh-Chung Chen and Chen-Mou Cheng and Tanja Lange and Ruben Niederhagen and Peter Schwabe and Bo-Yin Yang

- **How about ECC<sub>2</sub>-131?**

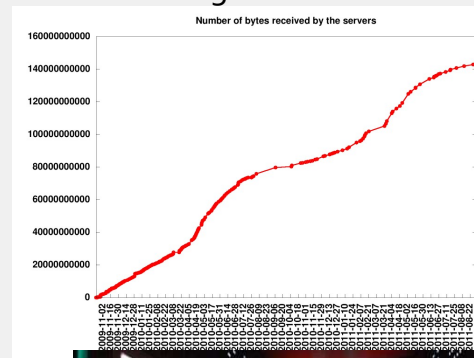
Test1: 4582 DPs using 20 nodes(480 kers) in 51.45 hs over Tianhe2

⇒ ECC<sub>2</sub>-131 is in 120 days using Tianhe2(17920 nodes)

Test2: 561 DPs using i7-7940X @3.10GHz CPU(14 kers) in 54.93 hs

⇒ ECC<sub>2</sub>-131 is in 93.5 days using 10000 such CPUs

关沛冬-罗玉琴-张方国-田海博, **ECC<sub>2</sub>-131** 的并行 Pollard rho 算法实现分析, 密码学报





# Number/Function Field Sieve for DLP in $GF(p^n)$

Sub-exponential time:  $q=p^n$ ,  $c$  is a constant,

$$L_q[1/3, c] = \exp((c+o(1))(\log q)^{1/3}(\log \log q)^{2/3}).$$

- Practical Improvements on FFS [Joux-Lercier@ANTS2002]
- FFS in the medium prime case [Joux-Lercier@Eurocrypt2006]
- Antoine Joux, A new index calculus algorithm with complexity  $L(1/4+o(1))$  in very small characteristic [Eprint2013:95]
- BGJT: A **quasi-polynomial algorithm** for discrete logarithm in finite fields of small characteristic. CoRR, abs/1306.4244, 2013.
- Joux, 24/12/2012,  $GF(33553771^{47})$ ; 06/01/2013,  $GF(33341353^{57})$ ;
- Joux et al., 21/05/2013,  $GF(2^{6168})$ ; 01/2014,  $GF(3^{5 \cdot 479})$ ; 10/2014,  $GF(2^{1279})$ , (1279 is prime); 2016,  $GF(3^{6 \cdot 509})$ ;  $GF(2^{30750})$  (GKZ2019);
- ....

1) Small characteristic finite fields are not recommended for DLP cryptosystem;

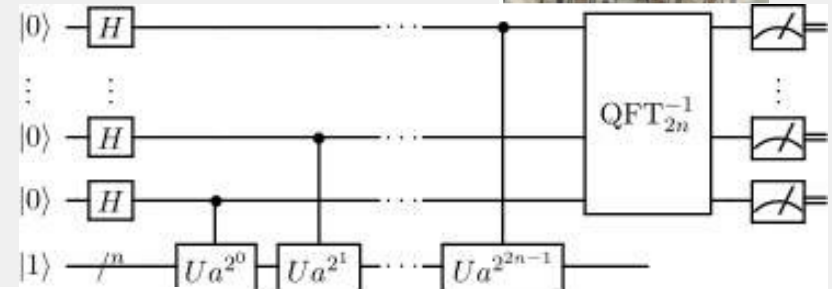
2) Bilinear pairings over finite fields with characteristic 2,3 or small and medium prime are not safe!





# Quantum computing on ECDLP

- Shor 94: Quantum computers can
  - Factor integers
  - Calculate DLP (in any group)
- **This breaks two PKCs:**  
**RSA, ECC**



ECDLP in $E(\mathbb{F}_p)$ simulation results					Factoring of RSA modulus $N$ interpolation from [21]		
$\lceil \log_2(p) \rceil$ bits	#Qubits	#Toffoli gates	Toffoli depth	Sim time sec	$\lceil \log_2(N) \rceil$ bits	#Qubits	#Toffoli gates
110	1014	$9.44 \cdot 10^9$	$8.66 \cdot 10^9$	273	512	1026	$6.41 \cdot 10^{10}$
160	1466	$2.97 \cdot 10^{10}$	$2.73 \cdot 10^{10}$	711	1024	2050	$5.81 \cdot 10^{11}$
192	1754	$5.30 \cdot 10^{10}$	$4.86 \cdot 10^{10}$	1 149	—	—	—
224	2042	$8.43 \cdot 10^{10}$	$7.73 \cdot 10^{10}$	1 881	2048	4098	$5.20 \cdot 10^{12}$
256	2330	$1.26 \cdot 10^{11}$	$1.16 \cdot 10^{11}$	3 848	3072	6146	$1.86 \cdot 10^{13}$
384	3484	$4.52 \cdot 10^{11}$	$4.15 \cdot 10^{11}$	17 003	7680	15362	$3.30 \cdot 10^{14}$
521	4719	$1.14 \cdot 10^{12}$	$1.05 \cdot 10^{12}$	42 888	15360	30722	$2.87 \cdot 10^{15}$

Table 2: Resource estimates of Shor's algorithm for computing elliptic curve discrete logarithms in  $E(\mathbb{F}_p)$  versus Shor's algorithm for factoring an RSA modulus  $N$ .

Martin Roetteler, Michael Naehrig, Krysta M. Svore, and Kristin Lauter, **Quantum Resource Estimates for Computing Elliptic Curve Discrete Logarithms**, ASIACRYPT 2017, Part II, LNCS 10625, pp. 241–270, 2017.





# 比特安全性或硬核比特

- $f: \{0,1\}^n \rightarrow \{0,1\}^n$  (**one-way function**)
- $h: \{0,1\}^n \rightarrow \{0,1\}$  is a hard-core predicate for  $f$  if given  $y=f(x)$  it is hard to guess  $h(x)$  with probability significantly higher than  $\frac{1}{2}$ .
- A case:  **$h(x)$** :  
any bit of  
binary  
representation  
of  $x$

Silvio Micali



Manuel Blum



- 1984: defined  
notion of pseudo-  
random generator



# Bit security of DLP in any group

- $G$  is a group with **prime order** and  $\langle g \rangle = G$ , given  $(g, g^x)$ , if an adversary can compute certain bits of  $x = x_{n-1}2^{n-1} + \dots + x_12 + x_0$ ?
- Blum and Micali : "finding some bits (aka. hard-core bits) is as hard as computing discrete log in any group  $G$ "
- DL-LSB  $\Leftrightarrow$  DLP (Divide and Square-root method)



# 离散对数的比特安全性

问题**6.2** 离散对数第*i*比特问题

实例:  $I = (p, \alpha, \beta, i)$ , 其中 $p$ 是素数,  $\alpha \in Z_p^*$ 是一个本原元,  $\beta \in Z_p^*$ ,  $i$ 是一个整数, 满足 $1 \leq i \leq \lceil \log_2(p-1) \rceil$ 。

计算: 计算 $L_i(\beta)$ , 这是 $\log_\alpha \beta$ 二进制表示的第*i*个比特。





计算一个离散对数的最低比特是容易的。即如果 $i = 1$ ，离散对数的第1比特问题可以有效地计算。主要是利用模 $p$ 二次剩余的Euler准则， $p$ 是素数。

在 $Z_p^*$ 中，恰有一半的元素是平方剩余，一半不是。

假设 $\alpha$ 是 $Z_p$ 的一个本原元。那么如果 $a$ 是偶数则 $\alpha^a \in QR(p)$ 。因为 $(p-1)/2$ 个元素 $\alpha^0 \bmod p, \alpha^2 \bmod p, \dots, \alpha^{p-3} \bmod p$ 互不相同，因此

$$QR(p) = \{\alpha^{2i} \bmod p : 0 \leq i \leq (p-3)/2\}$$

所以， $\beta$ 是二次剩余当且仅当 $\log_\alpha \beta$ 是偶数，即当且仅当 $L_1(\beta) = 0$ 。

由Euler准则可知， $\beta$ 是一个二次剩余，当且仅当

$$\beta^{(p-1)/2} \equiv 1 \pmod{p}$$

所以有下列计算公式计算 $L_1(\beta)$ ：

$$L_1(\beta) = \begin{cases} 0 & \beta^{(p-1)/2} \equiv 1 \pmod{p} \\ 1 & \text{others} \end{cases}$$







# 离散对数的比特安全性

现在考虑如何计算 $i$ 大于1时 $L_1(\beta)$ 的值。假设 $p-1=2^s t$ ,  $t$ 是奇数。可以证明对 $i \leq s$ , 容易计算 $L_1(\beta)$ 。另一方面, 计算 $L_{s+1}(\beta)$ 或许是困难的, 因为任何计算 $L_{s+1}(\beta)$ 的假设算法都可以用于计算 $\mathbb{Z}_p$ 上的离散对数。





# ElGamal体制的安全性

密码体制6.1所描述的基本的ElGamal密码体制不是语义安全的。

用Euler准则容易判定 $Z_p$ 的元素是否是模 $p$ 的二次剩余。由6.7.1知道， $\beta$ 是一个模 $p$ 的二次剩余，当且仅当 $a$ 是偶数。类似地， $y_1$ 是一个模 $p$ 的二次剩余，当且仅当 $k$ 是偶数。我们可以确定 $a$ 和 $k$ 的奇偶性来计算 $ak$ 的奇偶性。所以我们可以确定是否 $\beta^k (= \alpha^k)$ 是一个二次剩余。

假设要区分 $x_1$ 的加密与 $x_2$ 的加密，这里 $x_1$ 是二次剩余， $x_2$ 不是模 $p$ 的二次剩余。确定 $y_2$ 的二次剩余性是很简单的事情，我们已经讨论了确定 $\beta^k$ 的二次剩余性的判定方法。那么， $(y_1, y_2)$ 是 $x_1$ 的加密，当且仅当 $\beta^k$ 和 $y_2$ 二者同为二次剩余或者同为非二次剩余。

如果 $\beta$ 是一个二次剩余，每个明文 $x$ 也要求均为二次剩余，上述攻击就失效了。





# ElGamal体制的安全性

事实上, 如果 $p = 2q + 1$ ,  $q$ 是素数, 可以证明, 限制 $\beta, y_1$ 和 $x$ 为二次剩余等价于在模 $p$ 的二次剩余子群中实现ElGamal密码体制。如果在 $\mathbb{Z}_p^*$ 中离散对数问题是难解的, 这个版本的ElGamal密码体制被猜测是语义安全的。





Let  $\mathcal{G}$  be a polynomial-time algorithm that, on input  $1^n$ , outputs a (description of a) cyclic group  $\mathbb{G}$ , its order  $q$  (with  $\|q\| = n$ ), and a generator  $g$ . (As usual, we also require that the group operation in  $\mathbb{G}$  can be computed in time polynomial in  $n$ ). The El Gamal encryption scheme is defined as follows:

### **CONSTRUCTION 10.19**

Let  $\mathcal{G}$  be as in the text. Define a public-key encryption scheme as follows:

- $\text{Gen}(1^n)$  runs  $\mathcal{G}(1^n)$  to obtain  $\mathbb{G}, q, g$ , and then chooses a random  $x \leftarrow \mathbb{Z}_q$ . The public key is  $\langle \mathbb{G}, q, g, g^x \rangle$  and the private key is  $\langle \mathbb{G}, q, g, x \rangle$ .
- To encrypt a message  $m \in \mathbb{G}$  with respect to the public key  $pk = \langle \mathbb{G}, q, g, h \rangle$ , choose a random  $y \leftarrow \mathbb{Z}_q$  and output the ciphertext

$$\langle g^y, h^y \cdot m \rangle.$$

- To decrypt a ciphertext  $\langle c_1, c_2 \rangle$  using the private key  $sk = \langle \mathbb{G}, q, g, x \rangle$ , compute

$$m := c_2 / c_1^x.$$





# ElGamal的安全性

- 引理 如果DDH问题是困难的，那么ElGamal加密体制在选择明文攻击下是多项式安全的。

证明：为了显示ElGamal是多项式安全的，我们首先假设存在一个能够攻破ElGamal多项式安全性的多项式时间算法A，然后我们给出一个使用算法A作为子程序的算法B来解决DDH问题。

我们首先来回忆多项式安全性的攻击游戏：

在寻找阶段，输入一个公钥，输出两个消息和一些状态信息。

在猜测阶段，输入一个挑战密文、一个公钥、两个消息和一些状态信息，猜测挑战密文对应的明文是哪个消息。

ElGamal密文为：

$(g^k, mh^k)$

其中 $k$ 是一个随机整数， $h$ 是公钥。

给定 $g^x$ 、 $g^y$ 和 $g^z$ ，解决DDH问题的算法B执行如下步骤：

- ① 令 $h=g^x$ 。
- ②  $(m_0, m_1, s) = A$ （寻找阶段,  $h$ ）。
- ③ 设置 $c_1=g^y$ 。
- ④ 从 $\{0,1\}$ 中随机选择一个数 $b$ 。
- ⑤ 设置 $c_2=mbg^z$ 。
- ⑥  $b' = A$ （猜测阶段,  $(c_1, c_2), h, m_0, m_1, s$ ）。
- ⑦ 如果 $b = b'$ ，输出“TRUE”，否则输出“FALSE”。



# CDHP 与 DDHP

问题**6.3** 计算Diffie-Hellman问题(CDH)

实例：一个乘法群 $(G, \cdot)$ ，一个 $n$ 阶元素 $\alpha \in G$ ，两个元素 $\beta, \gamma \in \langle \alpha \rangle$ 。

问题：找出 $\delta \in \langle \alpha \rangle$ ，满足 $\log_{\alpha} \delta \equiv \log_{\alpha} \beta \times \log_{\alpha} \gamma \pmod{n}$ （等价地，给定 $\alpha^b$ 和 $\alpha^c$ ，找出 $\alpha^{bc}$ ）。

问题**6.4** 判定Diffie-Hellman问题(DDH)

实例：一个乘法群 $(G, \cdot)$ ，一个 $n$ 阶元素 $\alpha \in G$ ，三个元素 $\beta, \gamma, \delta \in \langle \alpha \rangle$ 。

问题：是否有 $\log_{\alpha} \delta \equiv \log_{\alpha} \beta \times \log_{\alpha} \gamma \pmod{n}$ ？（等价地，给定 $\alpha^b$ 、 $\alpha^c$ 和 $\alpha^d$ ，判定是否 $d \equiv bc \pmod{n}$ ）。





# CDHP 与 DDHP

对于CDH和DDH, 存在图灵归约

$$DDH \infty_T CDH$$

且

$$CDH \infty_T \text{Discrete Logarithm}$$

这些归约说明, 假设DDH难解与假设CDH难解至少有相同的程度。假设CDH难解与假设离散对数难解至少有着相同的程度。

EIGamal密码体制的语义安全性等价于DDH的难解性; EIGamal解密等价于求解CDH。因此, 要证明EIGamal密码体制的安全性所必需的假设强于仅假设离散对数是难解的。





# CDHP 与 ElGamal 体制

任何解CDH的算法，都可以用于解密ElGamal密文，反之亦然。

证明：“ $\Rightarrow$ ” 假设OracleCDH是解CDH的一个算法，

设 $(y_1, y_2)$ 是ElGamal密码体制的密文，具有公钥 $\alpha$ 和 $\beta$ 。计

算 $\delta = \text{OracleCDH}(\alpha, \beta, y_1)$ 然后定义 $x = y_2 \delta^{-1}$ 容易看出， $x$ 是密文 $(y_1, y_2)$ 的解密。

“ $\Leftarrow$ ” 假设Oracle-Elgamal-Decrypt是解密ElGamal密文的一个算法。

设 $\alpha, \beta, \gamma$ 如CDH中的假设给定。定义 $\alpha$ 和 $\beta$ 是ElGamal密码体制的公钥。那么，定义 $y_1 = \gamma$ ，令 $y_2 \in \langle \alpha \rangle$ 为随机取定。计

算 $x = \text{Oracle-Elgamal-Decrypt}(\alpha, \beta, (y_1, y_2))$ ，这是密文 $(y_1, y_2)$ 的解密。最后，计算 $\delta = y_2 x^{-1}$ 是CDH给定实例的解。即

$$\delta = \alpha^{\log_{\alpha} \beta \log_{\alpha} \gamma}$$



