

# Diffie-Hellman Problem and Elgamal Encryption Scheme

Yu Zhang

Harbin Institute of Technology

Cryptography, Autumn, 2015

- 1 Cyclic Groups and Discrete Logarithms**
- 2 Diffie-Hellman Assumptions and Applications**
- 3 The Elgamal Encryption Scheme**

**1** Cyclic Groups and Discrete Logarithms

2 Diffie-Hellman Assumptions and Applications

3 The Elgamal Encryption Scheme

# Cyclic Groups and Generators

$\mathbb{G}$  is finite and  $g \in \mathbb{G}$ ,  $\langle g \rangle \stackrel{\text{def}}{=} \{g^0, g^1, \dots\} = \{g^0, g^1, \dots, g^{i-1}\}$ .

- The **order** of  $g$  is the smallest positive integer  $i$  with  $g^i = 1$ .
- $\mathbb{G}$  is a **cyclic group** if  $\exists g$  has order  $m = |\mathbb{G}|$ .  $\langle g \rangle = \mathbb{G}$ ,  $g$  is a **generator** of  $\mathbb{G}$ .

■ Is  $\mathbb{Z}_6^*$ ,  $\mathbb{Z}_7^*$ , or  $\mathbb{Z}_8^*$  with ' $\cdot$ ' cyclic?

- $\langle g \rangle$  is a subgroup of  $\mathbb{G}$ , and  $|\langle g \rangle| \mid |\mathbb{G}|$ .
- If  $p$  is prime, then  $\mathbb{Z}_p^*$  is cyclic.

# Using Prime-Order Groups

## Theorem 1

*If  $\mathbb{G}$  is of prime order, then  $\mathbb{G}$  is cyclic. All  $g \in \mathbb{G}$  except the identity are generators.*

- The discrete logarithm problem is hardest in such groups.
- Finding a generator in such groups is trivial.
- Any non-zero exponent will be invertible modulo the order.
- A necessary condition for the DDH problem to be hard is that  $\text{DH}_g(h_1, h_2)$  by itself should be indistinguishable from a random group element. This is (almost) true for such groups.

# Generating Prime-Order (Sub)Groups in $\mathbb{Z}_p^*$

- $y \in \mathbb{Z}_p^*$  is a **quadratic residue modulo**  $p$  if  $\exists x \in \mathbb{Z}_p^*$  such that  $x^2 \equiv y \pmod{p}$ . (Q: show QRs in  $\mathbb{Z}_7^*$ )
- The set of QR is a subgroup with order  $(p-1)/2$  ( $x^2 \equiv (p-x)^2 \pmod{p}$ ).
- $p$  is a **strong prime** if  $p = 2q + 1$  with  $q$  prime.

---

**Algorithm 1:** A group generation algorithm  $\mathcal{G}$

---

**input** : Security parameter  $1^n$

**output:** Cyclic group  $\mathbb{G}$ , its order  $q$ , and a generator  $g$

- 1 **generate** a random  $(n+1)$ -bit strong prime  $p$
  - 2  $q := (p-1)/2$
  - 3 **choose** an arbitrary  $x \in \mathbb{Z}_p^*$  with  $x \not\equiv \pm 1 \pmod{p}$
  - 4  $g := x^2 \pmod{p}$
  - 5 **return**  $p, q, g$
-

# Discrete Logarithm

If  $\mathbb{G}$  is a cyclic group of order  $q$ , then  $\exists$  a generator  $g \in \mathbb{G}$  such that  $\{g^0, g^1, \dots, g^{q-1}\} = \mathbb{G}$ .

- $\forall h \in \mathbb{G}$ ,  $\exists$  a unique  $x \in \mathbb{Z}_q$  such that  $g^x = h$ .
- $x = \log_g h$  is the **discrete logarithm of  $h$  with respect to  $g$** .
- If  $g^{x'} = h$ , then  $\log_g h = [x' \bmod q]$ .
- $\log_g 1 = 0$  and  $\log_g(h_1 \cdot h_2) = [(\log_g h_1 + \log_g h_2) \bmod q]$ .

Show an instance of DL problem in  $\mathbb{Z}_7^*$

# Overview of Discrete Logarithm Algorithms

- Given a generator  $g \in \mathbb{G}$  and  $y \in \langle g \rangle$ , find  $x$  such that  $g^x = y$ .
- **Brute force:**  $\mathcal{O}(q)$ ,  $q = \text{ord}(g)$  is the order of  $\langle g \rangle$ .
- **Baby-step/giant-step** method [Shanks]:  $\mathcal{O}(\sqrt{q} \cdot \text{polylog}(q))$ .
- **Pohlig-Hellman** algorithm: when  $q$  has small factors.
- **Index calculus** method:  $\mathcal{O}(\exp(\sqrt{n \cdot \log n}))$ .
- The best-known algorithm is the **general number field sieve** with time  $\mathcal{O}(\exp(n^{1/3} \cdot (\log n)^{2/3}))$ .
- Elliptic curve groups vs.  $\mathbb{Z}_p^*$ : more efficient for the honest parties, but that are equally hard for an adversary to break. (Both 1024-bit  $\mathbb{Z}_p^*$  and 132-bit elliptic curve need  $2^{66}$  steps.)



# The Discrete Logarithm Assumption

The discrete logarithm experiment  $\text{DLog}_{\mathcal{A}, \mathcal{G}}(n)$ :

- 1 Run a group-generating algorithm  $\mathcal{G}(1^n)$  to obtain  $(\mathbb{G}, q, g)$ , where  $\mathbb{G}$  is a cyclic group of order  $q$  (with  $\|q\| = n$ ), and  $g$  is a generator of  $\mathbb{G}$ .
- 2 Choose  $h \leftarrow \mathbb{G}$ . ( $x' \leftarrow \mathbb{Z}_q$  and  $h := g^{x'}$ )
- 3  $\mathcal{A}$  is given  $\mathbb{G}, q, g, h$ , and outputs  $x \in \mathbb{Z}_q$ .
- 4  $\text{DLog}_{\mathcal{A}, \mathcal{G}}(n) = 1$  if  $g^x = h$ , and 0 otherwise.

## Definition 2

**The discrete logarithm problem is hard relative to  $\mathcal{G}$**  if  $\forall$  PPT algorithm  $\mathcal{A}$ ,  $\exists$   $\text{negl}$  such that

$$\Pr[\text{DLog}_{\mathcal{A}, \mathcal{G}}(n) = 1] \leq \text{negl}(n).$$

**1** Cyclic Groups and Discrete Logarithms

**2** Diffie-Hellman Assumptions and Applications

**3** The Elgamal Encryption Scheme

# Diffie-Hellman Assumptions

- **Computational Diffie-Hellman (CDH)** problem:

$$\text{DH}_g(h_1, h_2) \stackrel{\text{def}}{=} g^{\log_g h_1 \cdot \log_g h_2}$$

- **Decisional Diffie-Hellman (DDH)** problem:

Distinguish  $\text{DH}_g(h_1, h_2)$  from a random group element  $h'$ .

## Definition 3

DDH problem is hard relative to  $\mathcal{G}$  if  $\forall$  PPT  $\mathcal{A}$ ,  $\exists \text{negl}$  such that

$$\begin{aligned} & |\Pr[\mathcal{A}(\mathbb{G}, q, g, g^x, g^y, g^z) = 1] - \Pr[\mathcal{A}(\mathbb{G}, q, g, g^x, g^y, g^{xy}) = 1]| \\ & \leq \text{negl}(n). \end{aligned}$$

## Intractability of DL, CDH and DDH

DDH is easier than CDH and DL.

# Secure Key-Exchange Experiment

The key-exchange experiment  $\text{KE}_{\mathcal{A}, \Pi}^{\text{eav}}(n)$ :

- 1 Two parties holding  $1^n$  execute protocol  $\Pi$ .  $\Pi$  results in a **transcript** trans containing all the messages sent by the parties, and a **key**  $k$  that is output by each of the parties.
- 2 A random bit  $b \leftarrow \{0, 1\}$  is chosen. If  $b = 0$  then choose  $\hat{k} \leftarrow \{0, 1\}^n$  u.a.r, and if  $b = 1$  then set  $\hat{k} := k$ .
- 3  $\mathcal{A}$  is given trans and  $\hat{k}$ , and outputs a bit  $b'$ .
- 4  $\text{KE}_{\mathcal{A}, \Pi}^{\text{eav}}(n) = 1$  if  $b' = b$ , and 0 otherwise.

## Definition 4

A key-exchange protocol  $\Pi$  is secure in the presence of an eavesdropper if  $\forall$  PPT  $\mathcal{A}$ ,  $\exists$  negl such that

$$\Pr[\text{KE}_{\mathcal{A}, \Pi}^{\text{eav}}(n) = 1] < \frac{1}{2} + \text{negl}(n).$$

# Diffie-Hellman Key-Exchange Protocol



$$(\mathbb{G}, q, g) \leftarrow \mathcal{G}$$

$$\begin{array}{l} x \leftarrow \mathbb{Z}_q \\ h_1 := g^x \end{array} \xrightarrow{\mathbb{G}, q, g, h_1}$$

$$\xleftarrow{h_2} \begin{array}{l} y \leftarrow \mathbb{Z}_q \\ h_2 := g^y \end{array}$$

$$k_A := h_2^x$$

$$k_B := h_1^y$$

Q:  $k_A = k_B = k = ?$

$\widehat{\text{KE}}_{\mathcal{A}, \Pi}^{\text{eav}}$  denote an experiment where if  $b = 0$  the adversary is given  $\hat{k} \leftarrow \mathbb{G}$ .

## Theorem 5

*If DDH problem is hard relative to  $\mathcal{G}$ , then DH key-exchange protocol  $\Pi$  is secure in the presence of an eavesdropper (with respect to the modified experiment  $\widehat{\text{KE}}_{\mathcal{A}, \Pi}^{\text{eav}}$ ).*

## Security

Insecurity against active adversaries (Man-In-The-Middle).

# Proof of Security in DH Key-Exchange Protocol

## Proof.

$$\begin{aligned} \Pr \left[ \widehat{\text{KE}}_{\mathcal{A}, \Pi}^{\text{eav}} = 1 \right] \\ = \frac{1}{2} \cdot \Pr \left[ \widehat{\text{KE}}_{\mathcal{A}, \Pi}^{\text{eav}} = 1 | b = 1 \right] + \frac{1}{2} \cdot \Pr \left[ \widehat{\text{KE}}_{\mathcal{A}, \Pi}^{\text{eav}} = 1 | b = 0 \right] \end{aligned}$$

If  $b = 1$ , then give true key; otherwise give random  $g^z$ .

$$\begin{aligned} &= \frac{1}{2} \cdot \Pr [\mathcal{A}(g^x, g^y, g^{xy}) = 1] + \frac{1}{2} \cdot \Pr [\mathcal{A}(g^x, g^y, g^z) = 0] \\ &= \frac{1}{2} \cdot \Pr [\mathcal{A}(g^x, g^y, g^{xy}) = 1] + \frac{1}{2} \cdot (1 - \Pr [\mathcal{A}(g^x, g^y, g^z) = 1]) \\ &= \frac{1}{2} + \frac{1}{2} \cdot (\Pr [\mathcal{A}(g^x, g^y, g^{xy}) = 1] - \Pr [\mathcal{A}(g^x, g^y, g^z) = 1]) \\ &\leq \frac{1}{2} + \frac{1}{2} \cdot \text{negl}(n) \end{aligned}$$



# Example of DHKE

$$\mathbb{G} = \mathbb{Z}_{11}^*$$

The order  $q = ?$

The set of quadratic residues ?

Is  $g = 3$  a generator?

If  $x > 2$  and  $y = x + 1$ , what is  $x$  and  $y$  ?

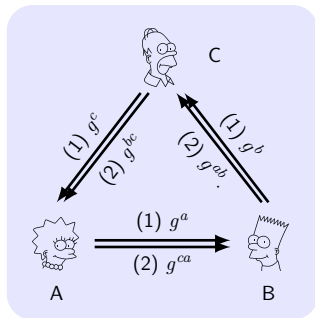
What's the message from Bob to Alice?

How does Alice compute the key?

How does Bob compute the key?

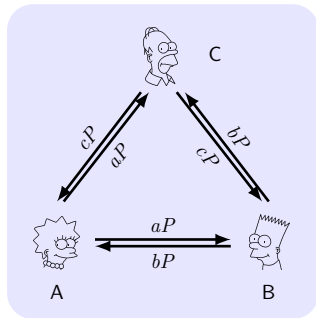
# Triparties Key Exchange

DH-based KE in 2 rounds:



$$\text{Key} = g^{abc}.$$

Joux's KE in 1 round:



$$\text{Key} = e(P, P)^{abc} \text{ in bilinear map.}$$

## Open Problem

How to exchange keys between 4 parties in one round?



**1** Cyclic Groups and Discrete Logarithms

**2** Diffie-Hellman Assumptions and Applications

**3** The Elgamal Encryption Scheme

# Lemma on Perfectly-secret Private-key Encryption

## Lemma 6

$\mathbb{G}$  is a finite group and  $m \in \mathbb{G}$  is an arbitrary element. Then choosing random  $g \leftarrow \mathbb{G}$  and setting  $g' := m \cdot g$  gives the same distribution for  $g'$  as choosing random  $g' \leftarrow \mathbb{G}$ . I.e.,  $\forall \hat{g} \in \mathbb{G}$ :

$$\Pr[m \cdot g = \hat{g}] = 1/|\mathbb{G}|.$$

## Proof.

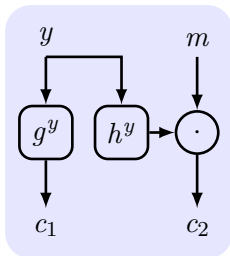
Let  $\hat{g} \in \mathbb{G}$  be arbitrary, then

$$\Pr[m \cdot g = \hat{g}] = \Pr[g = m^{-1} \cdot \hat{g}].$$

Since  $g$  is chosen *u.a.r.*, the probability that  $g$  is equal to the fixed element  $m^{-1} \cdot \hat{g}$  is exactly  $1/|\mathbb{G}|$ . □

# The Elgamal Encryption Scheme

An algorithm  $\mathcal{G}$ , on input  $1^n$ , outputs a description of a cyclic group  $\mathbb{G}$ , its order  $q$  (with  $\|q\| = n$ ), and a generator  $g$ .



## Construction 7

- Gen: run  $\mathcal{G}(1^n)$  to obtain  $(\mathbb{G}, q, g)$ . A random  $x \leftarrow \mathbb{Z}_q$  and  $h := g^x$ .  $pk = \langle \mathbb{G}, q, g, h \rangle$  and  $sk = \langle \mathbb{G}, q, g, x \rangle$
- Enc: a random  $y \leftarrow \mathbb{Z}_q$  and output  $\langle c_1, c_2 \rangle = \langle g^y, h^y \cdot m \rangle$
- Dec:  $m := c_2 / c_1^x$

## Theorem 8

*If the DDH problem is hard relative to  $\mathcal{G}$ , then the Elgamal encryption scheme is CPA-secure.*

# Example of Elgamal Encryption

## Encoding binary strings:

- the subgroup of quadratic residues modulo a strong prime  $p = (2q + 1)$ .
- a string  $\hat{m} \in \{0, 1\}^{n-1}$ ,  $n = \|q\|$ .
- map  $\hat{m}$  to the plaintext  $m = [(\hat{m} + 1)^2 \bmod p]$ .
- The mapping is one-to-one and efficiently invertible.

$q = 83$ ,  $p = 2q + 1 = 167$ ,  $g = 2^2 = 4 \pmod{167}$ ,  $\hat{m} = 011101$

The receiver chooses secret key  $37 \in \mathbb{Z}_{83}$ .

The public key is  $pk = \langle 167, 83, 4, [4^{37} \bmod 167] = 76 \rangle$ .

$\hat{m} = 011101 = 29$ ,  $m = [(29 + 1)^2 \bmod 167] = 65$ .

Choose  $y = 71$ , the ciphertext is

$\langle [4^{71} \bmod 167], [76^{71} \cdot 65 \bmod 167] \rangle = \langle 132, 44 \rangle$ .

Decryption:  $m = [44 \cdot (132^{37})^{-1}] \equiv [44 \cdot 66] \equiv 65 \pmod{167}$ .

65 has the two square roots 30 and 137, and  $30 < q$ , so  $\hat{m} = 29$ .

# Proof of Security of Elgamal Encryption Scheme

## Proof.

**Idea:** Prove that  $\Pi$  is secure in the presence of an eavesdropper by reducing an algorithm  $D$  for DDH problem to the eavesdropper  $\mathcal{A}$ .

Modify  $\Pi$  to  $\tilde{\Pi}$ : the encryption is done by choosing random  $y \leftarrow \mathbb{Z}_q$  and  $z \leftarrow \mathbb{Z}_q$  and outputting the ciphertext:

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

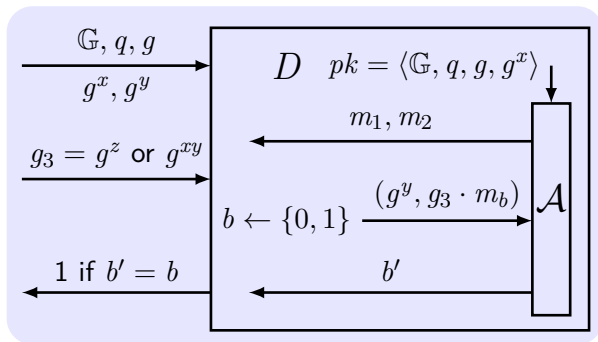
- $\tilde{\Pi}$  is not an encryption scheme.
- $g^y$  is independent of  $m$ .
- $g^z \cdot m$  is a random element independent of  $m$  (Lemma 6).

$$\Pr \left[ \text{PubK}_{\mathcal{A}, \tilde{\Pi}}^{\text{eav}}(n) = 1 \right] = \frac{1}{2}.$$



# Proof (Cont.)

$D$  receives  $(\mathbb{G}, q, g, g^x, g^y, g_3)$  where  $g_3$  equals either  $g^{xy}$  or  $g^z$ , for random  $x, y, z$ :



**Case I:**  $g_3 = g^z$ , ciphertext is  $\langle g^y, g^z \cdot m_b \rangle$ .

$$\Pr[D(g^x, g^y, g^z) = 1] = \Pr[\text{PubK}_{\mathcal{A}, \tilde{\Pi}}^{\text{eav}}(n) = 1] = \frac{1}{2}.$$

**Case II:**  $g_3 = g^{xy}$ , ciphertext is  $\langle g^y, g^{xy} \cdot m_b \rangle$ .

$$\Pr[D(g^x, g^y, g^{xy}) = 1] = \Pr[\text{PubK}_{\mathcal{A}, \Pi}^{\text{eav}}(n) = 1] = \varepsilon(n).$$

Since the DDH problem is hard,

$$\begin{aligned} \text{negl}(n) &\geq |\Pr[D(g^x, g^y, g^z) = 1] - \Pr[D(g^x, g^y, g^{xy}) = 1]| \\ &= \left| \frac{1}{2} - \varepsilon(n) \right|. \end{aligned}$$

## Constructing the ciphertext of the message $m \cdot m'$ .

Given  $pk = \langle g, h \rangle$ ,  $c = \langle c_1, c_2 \rangle$ ,  $c_1 = g^y$ ,  $c_2 = h^y \cdot m$ ,

**Method I:** compute  $c'_2 := c_2 \cdot m'$ , and  $c' = \langle c_1, c'_2 \rangle$ .

$$\frac{c'_2}{c_1^x} = ?$$

**Method II:** compute  $c''_1 := c_1 \cdot g^{y''}$ , and  $c''_2 := c_2 \cdot h^{y''} \cdot m'$ .

$$c''_1 = g^y \cdot g^{y''} = g^{y+y''} \text{ and } c''_2 = ?$$

so  $c'' = \langle c''_1, c''_2 \rangle$  is an encryption of  $m \cdot m'$ .



# Elgamal Implementation Issues

- **Sharing public parameters:**  $\mathcal{G}$  generates parameters  $\mathbb{G}, q, g$ .
  - generated “once-and-for-all”.
  - used by multiple receivers.
  - each receiver must choose their own secret values  $x$  and publish their own public key containing  $h = g^x$ .

## Parameter sharing

In the case of Elgamal, the public parameters can be shared. In the case of RSA, can parameters be shared?

- cyclic group, discrete log., baby-step/giant-step
- CDH, DDH, DHKE protocol
- Elgamal encryption, sharing public parameters