

# Diffie-Hellman Problem and Elgamal Encryption Scheme

Yu Zhang

HIT/CST/NIS

Cryptography, Autumn, 2014

- 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 a finite group and  $g \in \mathbb{G}$ , the set

$$\langle g \rangle \stackrel{\text{def}}{=} \{g^0, g^1, \dots\} = \{g^0, g^1, \dots, g^{i-1}\}.$$

- 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  $\mathbb{G}$ .
- $\langle g \rangle$  is a subgroup of  $\mathbb{G}$ , and  $|\langle g \rangle| \mid |\mathbb{G}|$ .
- If  $\mathbb{G}$  is a group of prime order  $p$ , then  $\mathbb{G}$  is cyclic. All elements of  $\mathbb{G}$  except the identity are generators of  $\mathbb{G}$ .

## Theorem 1

*If  $p$  is prime, then  $\mathbb{Z}_p^*$  is cyclic.*

# Examples of Cyclic Groups

- $\mathbb{Z}_{15}$  with '+' is cyclic and the '1' is a generator.
- $\mathbb{Z}_7$  with '+' is cyclic and all non-zero elements are generators.
- For  $\mathbb{Z}_7^*$ , 2 is not a generator, but 3 is.

$\mathbb{G}$  is a cyclic group of order  $n$ , and  $g$  is a generator of  $\mathbb{G}$ . Then the mapping  $f : \mathbb{Z}_n \rightarrow \mathbb{G}$  given by  $f(a) = g^a$  is an isomorphism. For  $a, a' \in \mathbb{Z}_n$ ,

$$f(a + a') = g^{[a+a' \bmod n]} = g^{a+a'} = g^a \cdot g^{a'} = f(a) \cdot f(a').$$

All cyclic groups of the same order are “the same” in an algebraic sense, but this is not true in a computational sense.

# 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  $\log_g(h_1 \cdot h_2) = [(\log_g h_1 + \log_g h_2) \bmod q]$ .

# 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

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

# 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.)



**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.

# Recall 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



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

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

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

$$k_A := h_2^x$$

$$k_B := h_1^y$$

$$k_A = k_B = k = g^{xy}.$$

$\widehat{\text{KE}}_{\mathcal{A}, \Pi}^{\text{eav}}$  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  $g^z$ .

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



**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  $1/|\mathbb{G}|$ .  $\square$

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

## Theorem 8

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



# Example of Elgamal Encryption

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

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  $\hat{m} = 29$ .

## 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  $c' = \langle c_1, c'_2 \rangle$ .

$$\frac{c'_2}{c_1^x} = \frac{h^y \cdot m \cdot m'}{g^{xy}} = \frac{g^{xy} \cdot m \cdot m'}{g^{xy}} = m \cdot m'.$$

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

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

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

# Elgamal Implementation Issues

- **Encoding binary strings:** depends on the particular type of group.
  - 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.
- **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, but in the case of RSA, parameters cannot be shared.

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