

# E/Gamal Encryption Scheme

It's security is based on DH problem

## Algorithm for the key generation of E/Gamal

Summary: each entity creates a public-key & a corresponding private-key

1. Generate a large random prime " $p$ " & a generator " $\alpha$ " of the multiplicative group  $\mathbb{Z}_p^*$  of the integers mod " $p$ ".

$$\mathbb{Z}_p = \{0, 1, \dots, p-1\}$$

$$\mathbb{Z}_p^* = \{1, 2, \dots, p-1\}$$

Note: you can select a random number (large) & the test it with "primality test alg" to see if it's a prime number.

2. Select a random integer " $a$ ",  $1 \leq a \leq p-2$  & then you compute  $(\underset{\substack{\uparrow \\ \text{generator}}}{\alpha}^a \bmod p)$  using square-&-multiply algorithm.

3. public-key  $(p, \alpha, \underset{\substack{\uparrow \\ \text{perform the reduction} \\ (\bmod p)}}{(\alpha^a)})$ , private-key " $a$ "

# Algorithm for ElGamal Encryption / Decryption

2

summary: B encrypts a message 'm' for A, A decrypts

## 1. Encryption

- obtain A's public key  $(p, \alpha, \alpha^a)$
- represent the message 'm' as an integer in the range  $\{0, 1, \dots, p-1\}$
- select a random integer  $k$ ,  $1 \leq k \leq p-2$
- compute  $\gamma = \alpha^k \pmod{p}$  and  $\delta = m \cdot (\alpha^a)^k \pmod{p}$   
single value
- send ciphertext  $c = (\gamma, \delta)$

## 2. Decryption

- use the private-key 'a' to compute

$$\gamma^{p-1-a} \pmod{p}$$

- Recover 'm' private-key

$$(\gamma^{-a}) \cdot \delta \pmod{p}$$

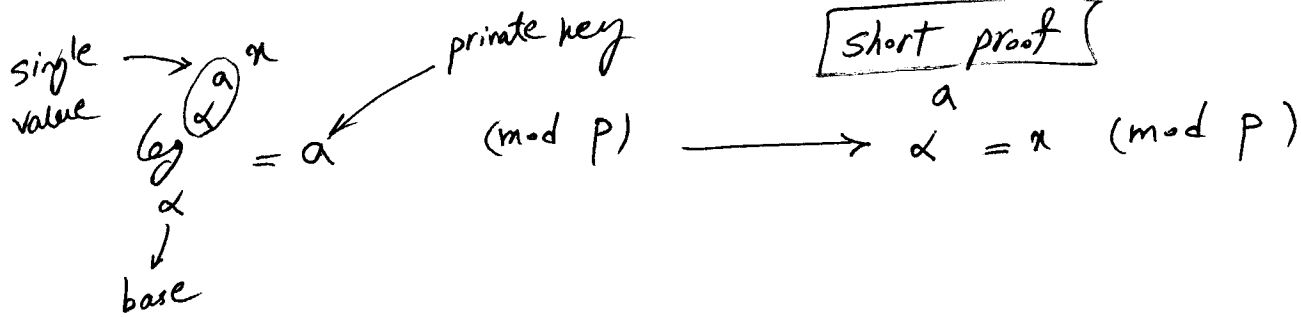
another part of the cipher text

$$\gamma^{p-1-a} \pmod{p} \equiv \underbrace{\gamma^{p-1}}_{\text{Fermat's little Theorem}} \cdot \gamma^{-a} \pmod{p} \equiv \gamma^{-a} \pmod{p}$$

we know:  $\delta = m \cdot (\alpha^a)^k = m \cdot (\underbrace{\alpha^k}_\gamma)^a = m \cdot \gamma^a$

part (b)-dec:  $\gamma^{-a} \cdot \delta = m \cdot \gamma^a \cdot \gamma^{-a} = \boxed{m}$   
inverse

part of correctness



In ElGamal:  $(\alpha, p, \alpha^a)$  public

there is no efficient algorithm to find  $a$  which is the private-key of the sender. that's why ElGamal relies on DH problem.

### Example ElGamal

key gen part { "A" selects  $p=2357$  a generator  $\alpha=2$  of  $\mathbb{Z}_{2357}^*$   
 $a=1751$   
public key  $(p=2357, \alpha=2, \alpha^a = \frac{1185}{x})$

$m=2035$

Encryption

"B" random integer  $k=1520$   
 $\gamma = \alpha^k = 2^{1520} (\text{mod } 2357) = 1430$   
 $\delta = 2035 \cdot 1185^{1520} (\text{mod } 2357) = 697$   
cipher text  $(1430, 697)$

Decryption

"A"

$$\gamma^{p-1-a} = 1430^{605} (\text{mod } 2357) = 872$$

$$\begin{array}{ccc} 2356 & - & 1751 \\ \downarrow & & \downarrow \\ p-1 & & a \end{array} = 605$$

$$m = 872 \cdot 697 (\text{mod } 2357) = 2035$$

$\underbrace{\hspace{1cm}}_{\delta} \qquad \underbrace{\hspace{1cm}}_{\text{message}}$