Cryptography

Lecture 9

Arpita Patra

Quick Recall and Today's Roadmap

>> Assumptions in Cyclic Groups (of prime order); how to construct such creatures using NT and GT

>> DH Key Agreement

>> Intro to PKE. Plus and Minus

>> PKE Security Definition

>> CPA Security

>> CPA Multi-message Security

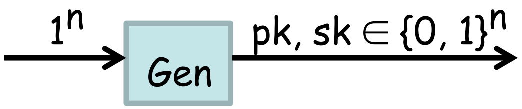
>> CPA Single Message Security Implies CPA Multi-message Security Proof: Fantastic application of hybrid arguments

>> El Gamal CPA Secure Scheme

>> RSA (maybe)

Public-key Cryptography: Syntax

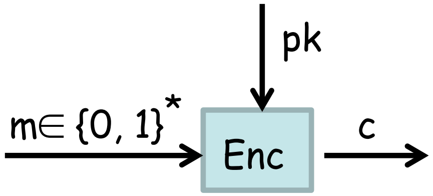
- A public-key cryptosystem is a collection of 3 PPT algorithms (Gen, Enc, Dec)



1^n Gen $pk, sk \in \{0, 1\}^n$

Syntax: $(pk, sk) \leftarrow Gen(1^n)$

Randomized Algo

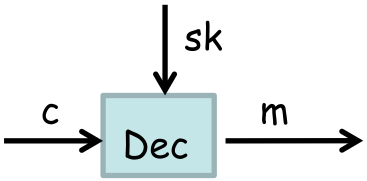


pk

$m \in \{0, 1\}^*$ Enc c

Syntax: $c \leftarrow Enc_{pk}(m)$

Most often randomized to achieve meaningful notion of security



sk

c Dec m

Syntax: $m := Dec_{sk}(c)$

Deterministic (w.l.o.g)

Except with a negligible probability over (pk, sk) output by $Gen(1^n)$, we require the following for every (legal) plaintext m

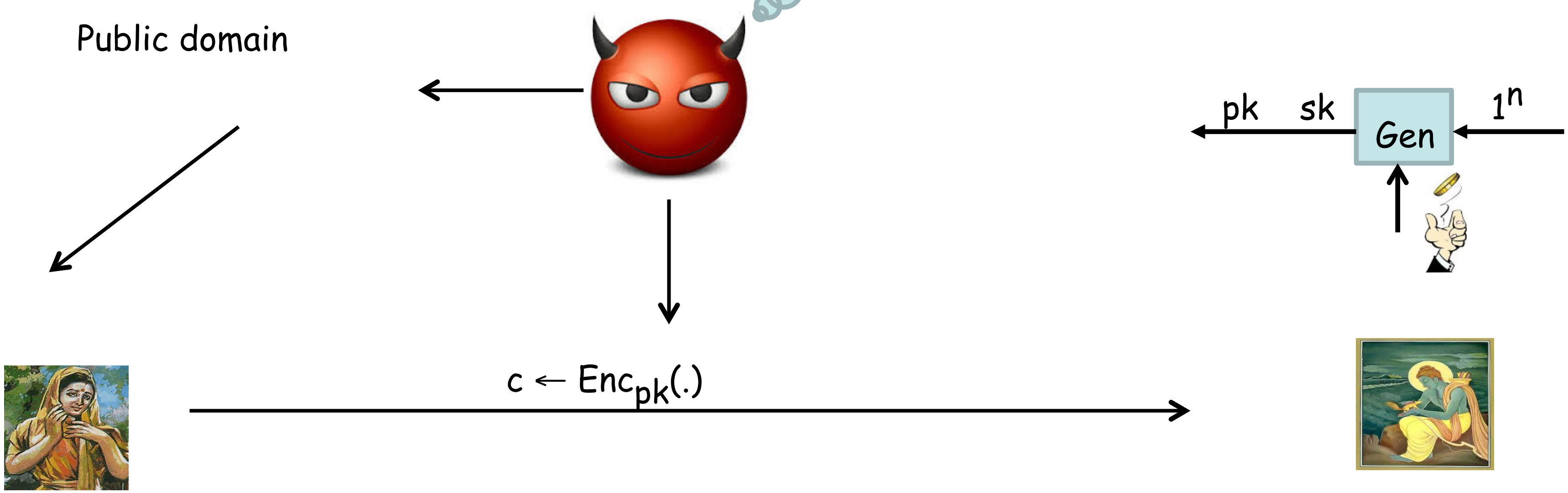
$Dec_{sk}(Enc_{pk}(m)) := m$

Public-key Encryption: Security Definition

Let Π = (Gen, Enc, y

- ❑ What is the least possible security guar

I know that the message is either “I am not fine” or “I am fine Ram”



- ❑ We expect that even after seeing the ciphertext c, the adversary should not be able to find out the password, except with probability negligibly better than $\frac{1}{2}$
  - ➢ Semantic security/IND security

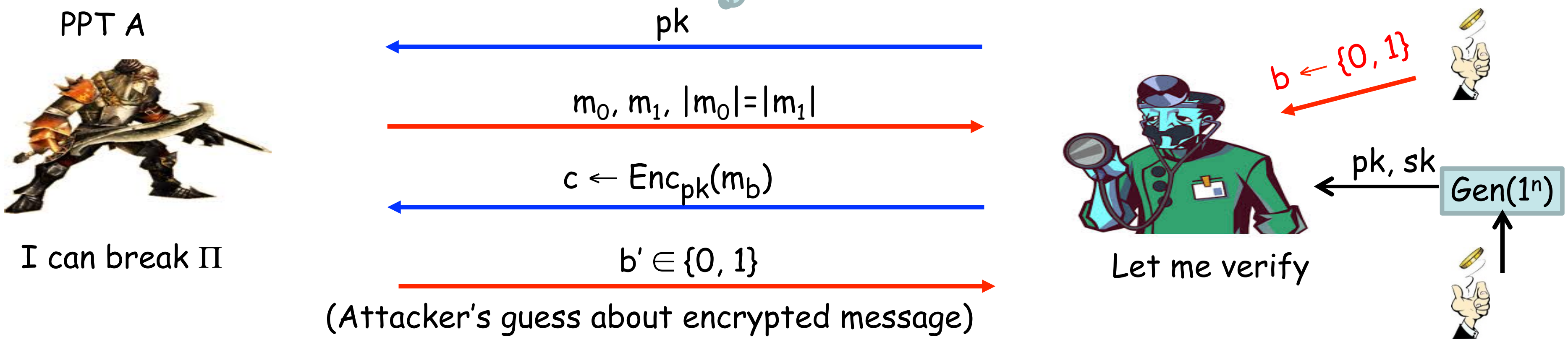
Indistinguishability Experiment for PKE (Ciphertext-only Attack)

Indistinguishability experiment

Pub

In the real-world, everyone including the attacker will have the public key pk

en, Enc, Dec)



PPT A

I can break Π

pk

$m_0, m_1, |m_0| = |m_1|$

$c \leftarrow \mathrm{Enc}_{pk}(m_b)$

$b' \in \{0, 1\}$

(Attacker's guess about encrypted message)

$b \leftarrow \{0, 1\}$

pk, sk

$\mathrm{Gen}(1^n)$

Let me verify

Game Output

$b = b'$

1 --- attacker won

$b \neq b'$

2 st

How is the above experiment different from the corresponding symmetric-key encryption experiment ?

Π COA-secure if for every PPT attacker , the probability that A wins the experiment is at most negligib

$$\Pr\left[\mathrm{PubK}^{\mathrm{coa}}_{A,\Pi}(n) = 1\right] \leq \frac{1}{2} + \mathrm{negl}(n)$$

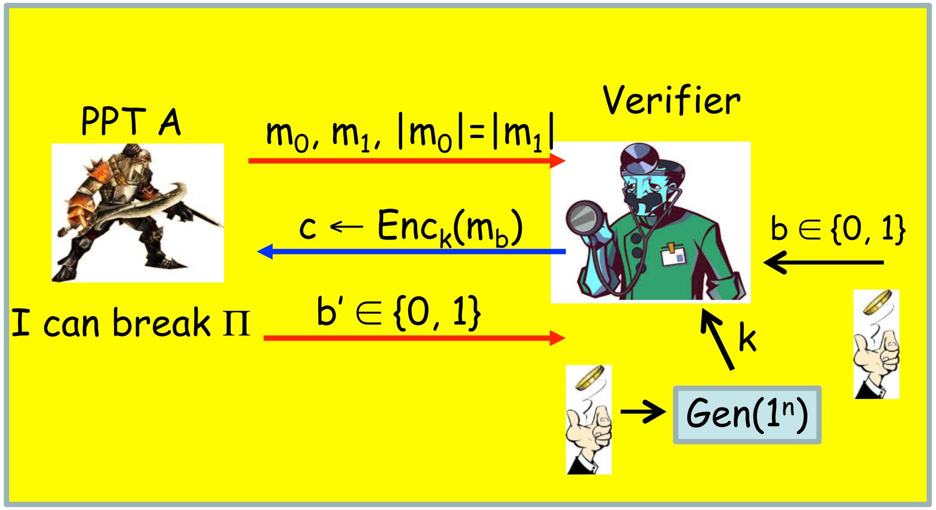
Ciphertext-only Attack: Symmetric-key vs Asymmetric-key World

Π = (Gen, Enc, Dec)

Symmetric-key Encryption

Π' = (Gen', Enc', Dec')

Asymmetric-key Encryption



PPT A

Verifier

$m_0, m_1, |m_0|=|m_1|$

$c \leftarrow Enc_k(m_b)$

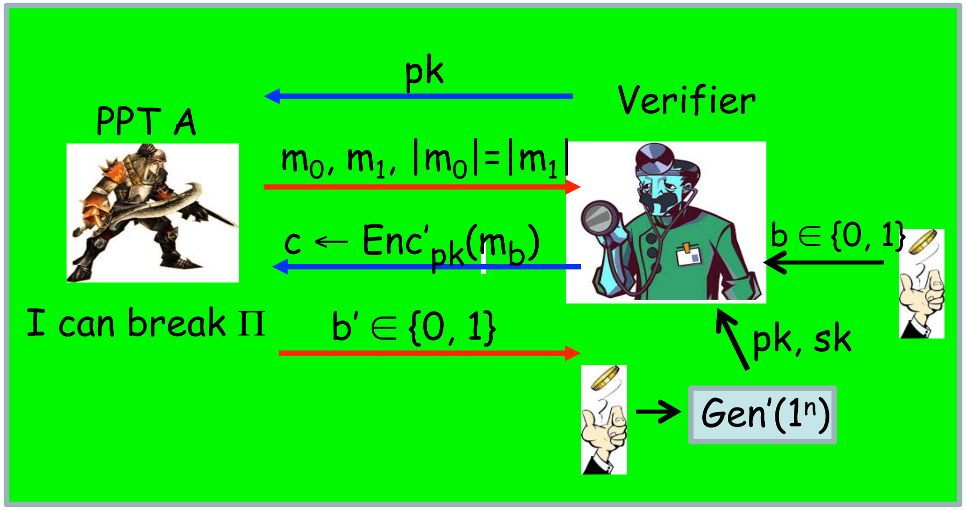
$b \in \{0, 1\}$

I can break Π

$b' \in \{0, 1\}$

k

Gen(1^n)



pk

PPT A

Verifier

$m_0, m_1, |m_0|=|m_1|$

$c \leftarrow Enc'_{pk}(m_b)$

$b \in \{0, 1\}$

I can break Π

$b' \in \{0, 1\}$

pk, sk

Gen'(1^n)

- ❑ Consequence of giving the public-key pk to the attacker ?
  - ➢ Attacker can encrypt any message of its ch
  - ➢ Free-encryption oracle for the attacker
    - ❖ Not possible in the symmetric-key wor
- ❑ Already captures CPA!!
- ❑ COA is equivalent to CPA security for PKE

Attention: No deterministic public-key encryption can be even COA-secure, whereas we have seen deterministic scheme to be COA-secure in SKE

Extremely dangerous for small message space. Adv can keep a table of encryptions of all the message and then compares to find the message encrypted.

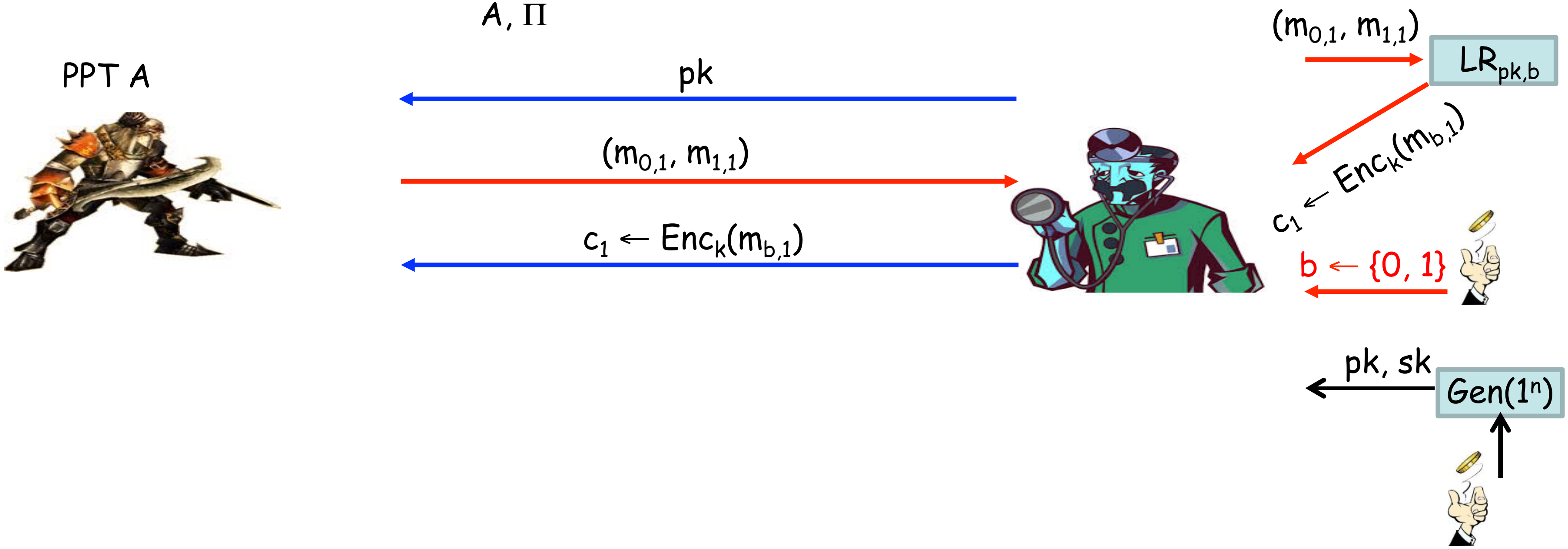
Multi-message CPA Security

>> Important to see the effect of using the same key for multiple messages

>> In reality multiple messages are encrypted under the same public key.

Multi-CPA experiment $\mathrm{PubK}^{\text{cpa-mult}}_{A,\Pi}(n)$

Π = (Gen, Enc, Dec)

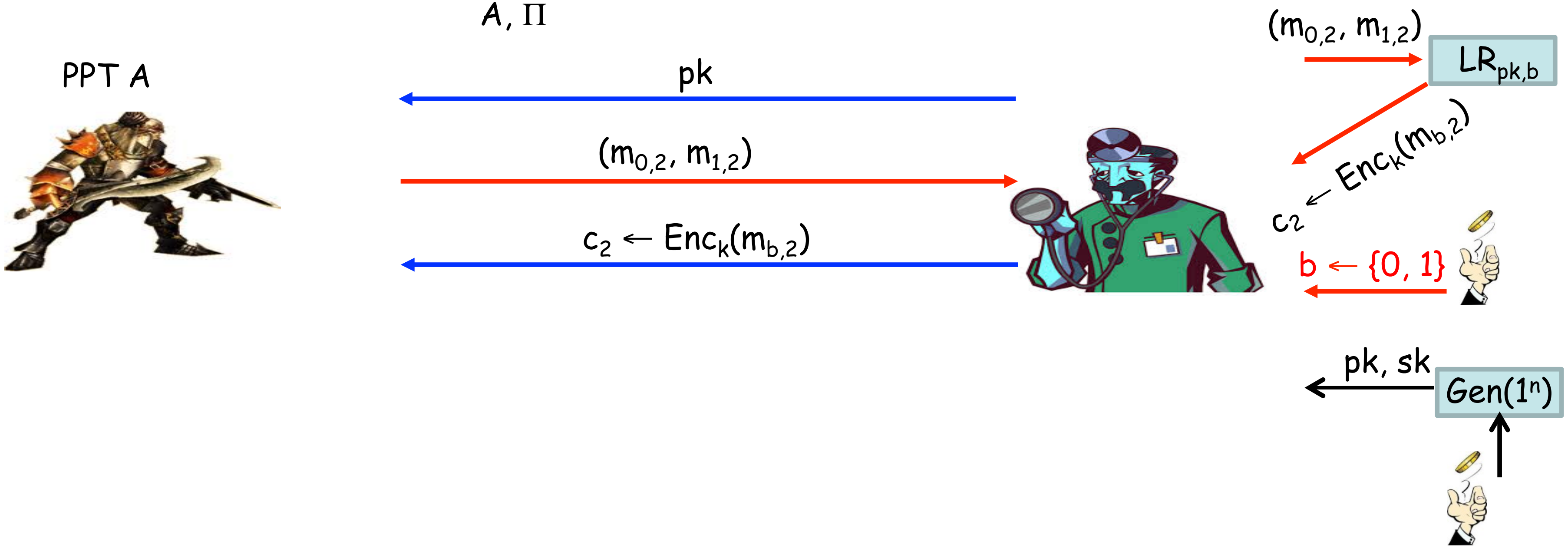


Multi-message CPA Security

>> Important to see the effect of using the same key for multiple messages
>> In reality multiple messages are encrypted under the same public key.

Multi-CPA experiment $\mathsf{PubK}^{\text{cpa-mult}}_{A,\Pi}(n)$

$\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$



PPT A

pk

$(m_{0,2}, m_{1,2})$

$c_2 \leftarrow \mathsf{Enc}_k(m_{b,2})$

$(m_{0,2}, m_{1,2})$

$\mathsf{LR}_{pk,b}$

$c_2 \leftarrow \mathsf{Enc}_k(m_{b,2})$

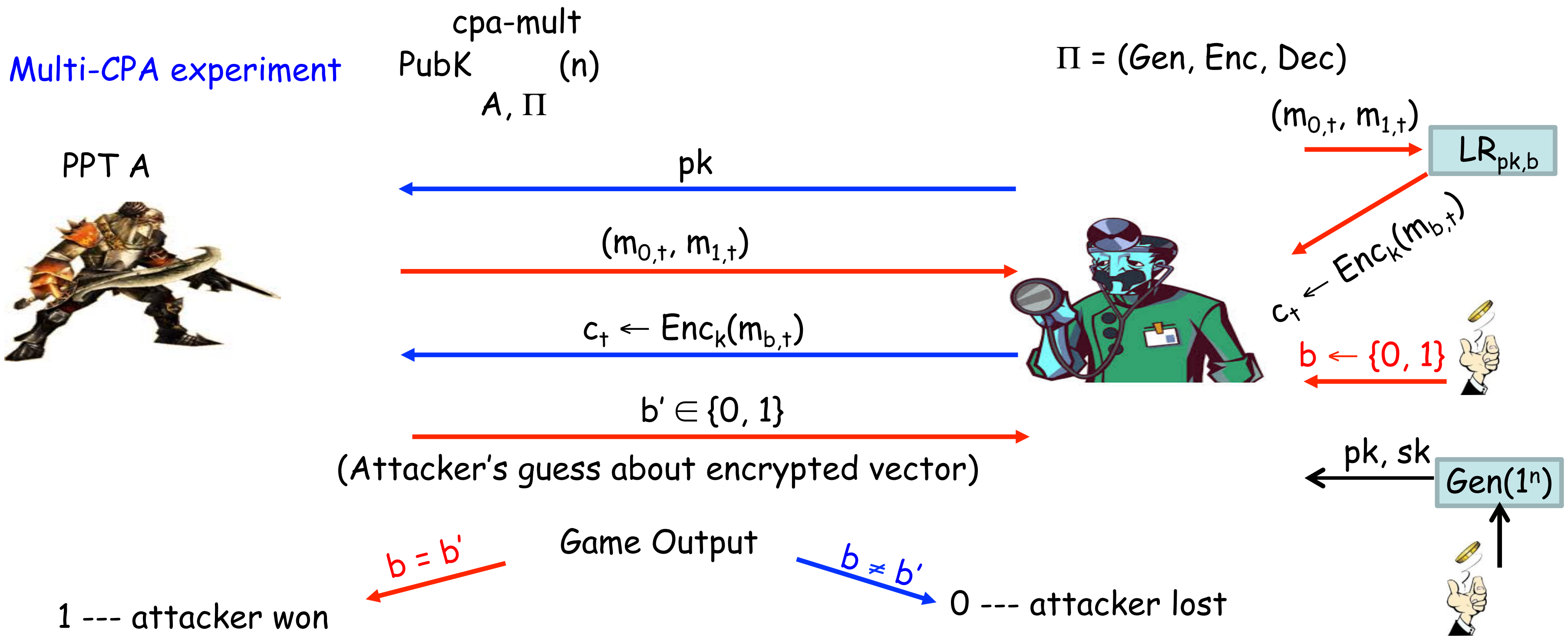
$b \leftarrow \{0, 1\}$

pk, sk

$\mathsf{Gen}(1^n)$

Multi-message CPA Security

>> Important to see the effect of using the same key for multiple messages
>> In reality multiple messages are encrypted under the same public key.



Multi-CPA experiment $\mathrm{PubK}^{\text{cpa-mult}}_{A,\Pi}(n)$

Π = (Gen, Enc, Dec)

PPT A

pk

$(m_{0,t}, m_{1,t})$

$c_t \leftarrow \mathrm{Enc}_k(m_{b,t})$

$b' \in \{0, 1\}$

(Attacker's guess about encrypted vector)

$(m_{0,t}, m_{1,t})$

$\mathrm{LR}_{pk,b}$

$c_t \leftarrow \mathrm{Enc}_k(m_{b,t})$

$b \leftarrow \{0, 1\}$

pk, sk

Gen(1^n)

Game Output

$b = b'$

1 --- attacker won

$b \neq b'$

0 --- attacker lost

Π has mult-CPA secure if for every PPT attacker A taking part in the above experiment, the probability that A wins the experiment is at most negligibly better than ½

$$\Pr\left(\mathrm{PubK}^{\text{cpa-mult}}_{A,\Pi}(n) = 1\right) \leq \frac{1}{2} + \mathrm{negl}(n)$$

(Single vs Multi-message CPA Security)

Theorem: single-message CPA security) → multi-message CPA security).

Proof: On the board (power of hybrid argument)

Hybrid Arguments

Polynomially Many

World/View 1

$|\Pr[A(\text{View1}) = 1 - \Pr[A(\text{View1.1}) = 1]| < \text{negl}(n)$

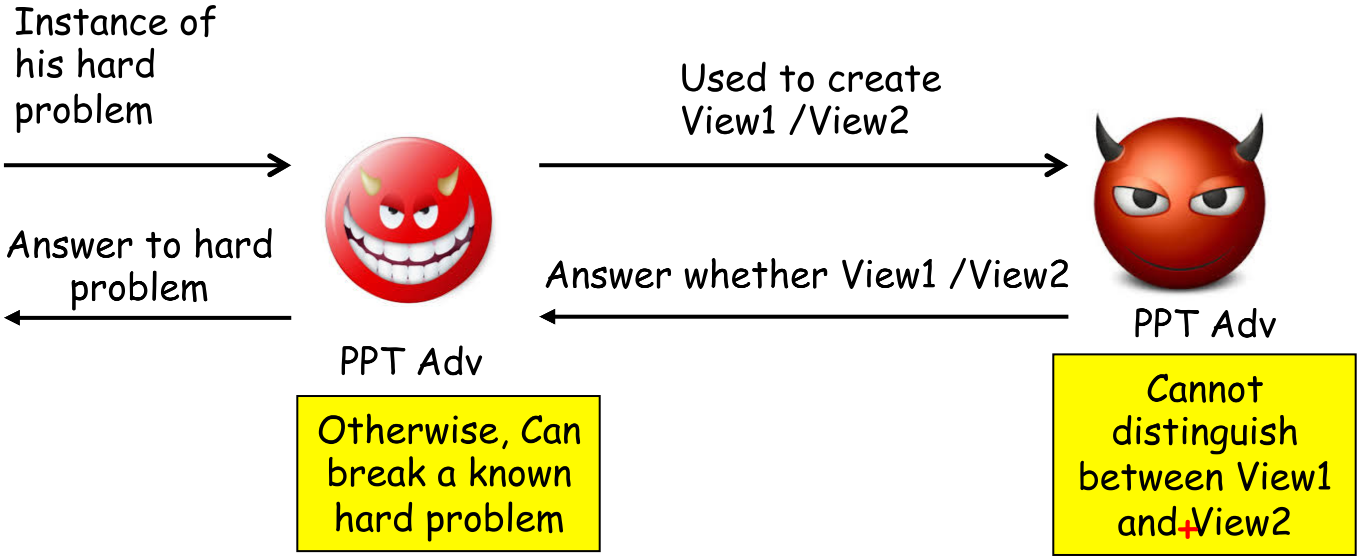
+

World/View 1.1

$|\Pr[A(\text{View1.1}) = 1 - \Pr[A(\text{View1.2}) = 1]| < \text{negl}(n)$

+

World/View 1.2



Instance of his hard problem

Used to create View1 /View2

Answer to hard problem

Answer whether View1 /View2

PPT Adv

PPT Adv

Otherwise, Can break a known hard problem

Cannot distinguish between View1 and View2

World/View 1.i

World/View 1.t

+

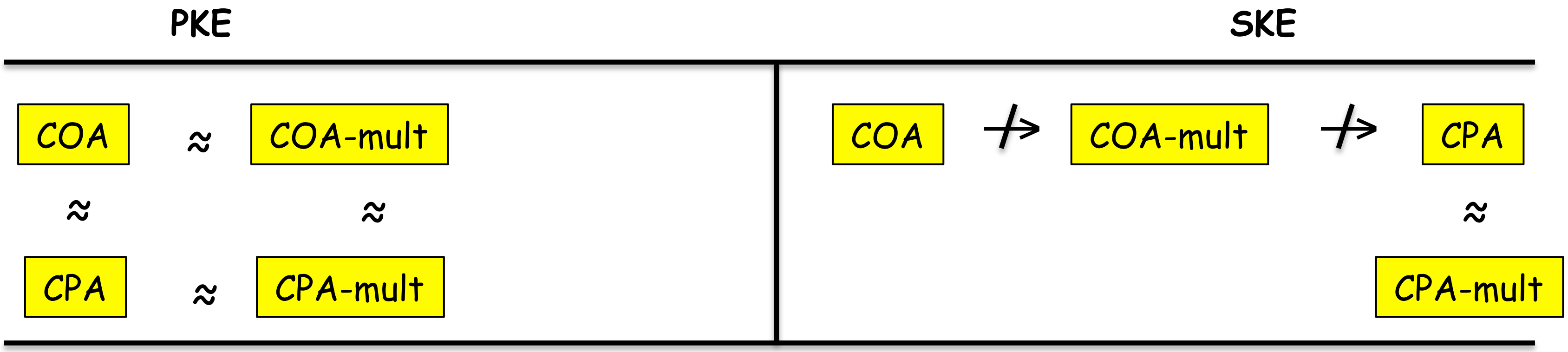
$|\Pr[A(\text{View1.t}) = 1 - \Pr[A(\text{View2}) = 1]| < \text{negl}(n)$

World/View 2

$|\Pr[A(\text{View1}) = 1 - \Pr[A(\text{View2}) = 1]| < t \cdot \text{negl}(n)$

The intermediate views are called hybrids

Implications of Single-message CPA security → Multi-message CPA Security



PKE

COA ≈ COA-mult

≈ ≈

CPA ≈ CPA-mult

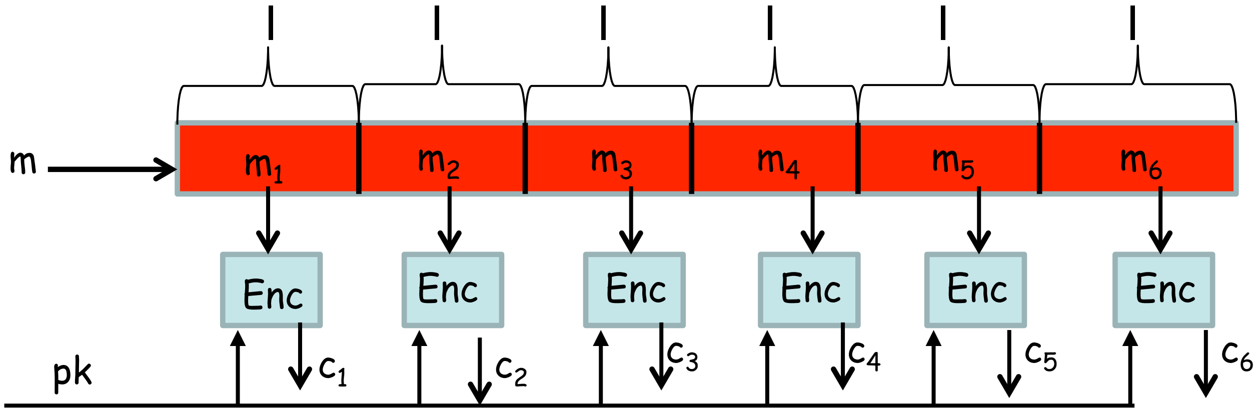
SKE

COA ↛ COA-mult ↛ CPA

≈

CPA-mult

- ❑ Given CPA secure scheme Π for bit/small messages, constructing CPA-secure PKE for long message is not an issue.



Heads-up; Surprize: Sames does not hold for CCA security. Term paper

$c_1 c_2 \ldots c_6 \leftarrow \mathrm{Enc}_{pk}(m)$

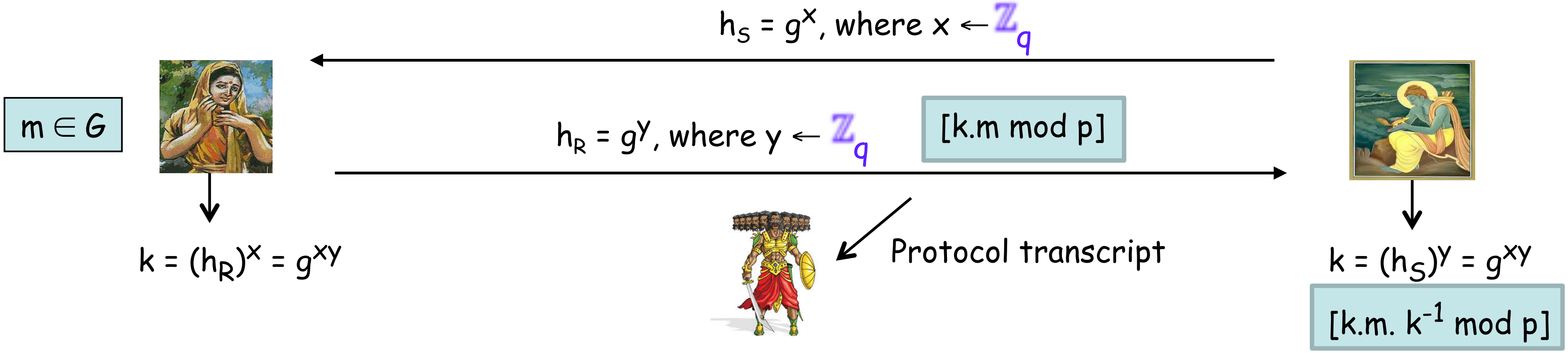
- ❑ Why the above PKE, say Π' is CPA-secure ?
  - ➢ The above construction is equivalent to encrypting a vector of message $\vec{M} = (m_1, \ldots, m_6)$
  - ➢ Reduction of CPA-security of Π' for LARGE single message → CPA-security for Π for multi messages

CPA-secure Public-key Encryption Based on DDH (El Gamal Encryption Scheme)

- Invented by Taher El Gamal in 1985
  - Based on the observation that the DH key-exchange protocol can be "converted" into a public-key encryption algorithm by incorporating an additional step
- Recall the DH key-exchange protocol

Public Info: Cyclic group of prime order q, (G, ., q, g)

(For concreteness, consider ($\mathbb{Z}_p^*$, * mod p) and the subgroup (G, * mod p), with $G = \{x^2 \bmod p\}$)



$h_S = g^x$, where $x \leftarrow \mathbb{Z}_q$

$m \in G$

$h_R = g^y$, where $y \leftarrow \mathbb{Z}_q$

[k.m mod p]

Protocol transcript

$k = (h_R)^x = g^{xy}$

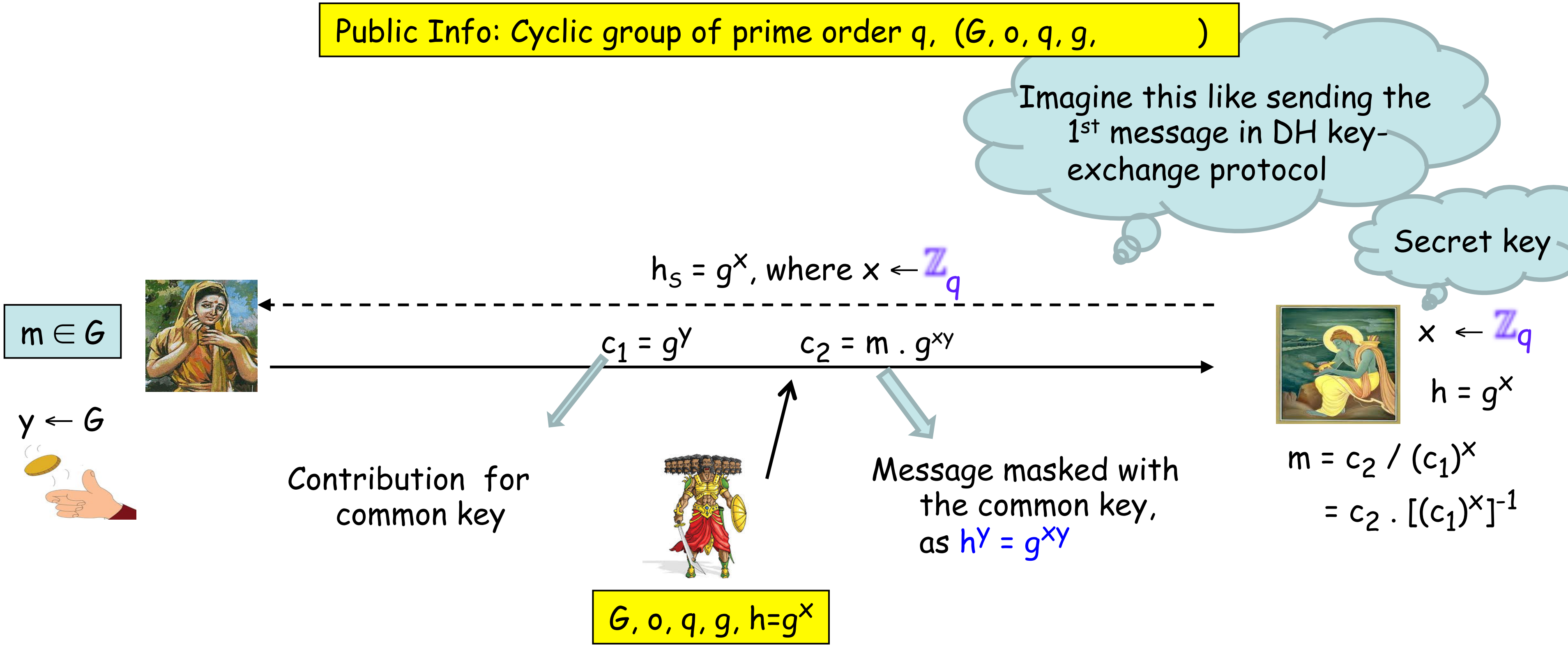
$k = (h_S)^y = g^{xy}$

[k.m. k^{-1} mod p]

Unable to distinguish $k = g^{xy}$ from a random element g^z in G (if DDH is hard in G)

- How to convert this protocol into a public-key encryption scheme ?
  - The encryptor can use the agreed upon key k to mask its message !!

El Gamal Public-key Encryption



Public Info: Cyclic group of prime order q, (G, o, q, g,)

Imagine this like sending the 1st message in DH key-exchange protocol

Secret key

$h_S = g^x$, where $x \leftarrow \mathbb{Z}_q$

$m \in G$

$c_1 = g^y$ $c_2 = m \,.\, g^{xy}$

$x \leftarrow \mathbb{Z}_q$

$h = g^x$

$y \leftarrow G$

Contribution for common key

Message masked with the common key, as $h^y = g^{xy}$

$m = c_2 / (c_1)^x$

$= c_2 \,.\, [(c_1)^x]^{-1}$

$G, o, q, g, h=g^x$

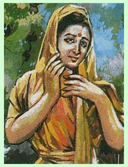
Theorem: If the DDH problem is hard relative to (G, o), then El Gamal encryption scheme is CPA-secure

- Adversary will be unable to distinguish the mask g^{xy} from a random group element g^z, given $h=g^x$, $c_1 = g^y$. Otherwise, we can use him to break DDH assumption.

- If an random element g^z was used for masking, then the encryption perfectly hides m (it is an OTP in fact). So even an unbounded powerful adversary will have no clue about the message

Security Proof of El Gamal

Public Key pk = $(G,o,q,g,h=g^x)$



m,pk= $(G,o,q,g,h=g^x)$ →

$\mathrm{Enc}_{pk}(m)$

$c_1 = g^y$ for random y

$c_2 = g^{xy}.\ m$

c →

c,sk=x →

$\mathrm{Dec}_{sk}(c)$

$c_2 / (c_1)^x = c_2\ .\ [(c_1)^x]^{-1}$

For any z', $\Pr[g^z.m = g^{z'}] = 1/|G|$ when z is chosen uniformly from G

Theorem. If DDH is hard, then Π is a CPA-secure scheme.

Proof: Assume Π is not CPA-secure

A, p(n): $\Pr\left(\mathrm{PubK}^{cpa}_{A,\Pi}(n) = 1\right) > \frac{1}{2} + 1/p(n)$ $\qquad \Pr\left(\mathrm{PubK}^{cpa}_{A,\overline{\Pi}}(n) = 1\right) = \frac{1}{2}$

$$\left|\ \Pr[D(\text{DDH tuple}) = 1] - \Pr[D(\text{non-DDH tuple}) = 1]\ \right| > 1/p(n)$$

Let us run $\mathrm{PubK}^{cpa}_{A,\Pi}(n)$

DDH or non-DDH tuple?

$(G,o,q,g,\ g^x,\ g^y,\ g^z)$ → D

← 1 if b = b'

0 otherwise

b

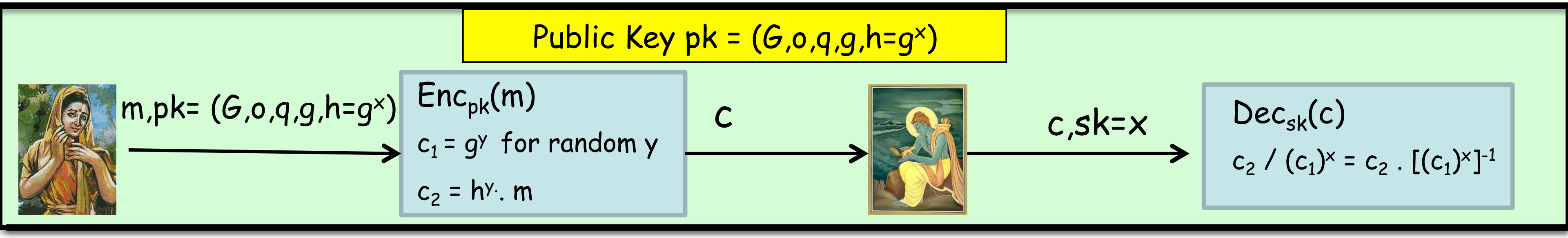
D → A: pk = (G,o,q,g,g^x)

A → D: $m_0, m_1 \in_R \mathcal{M}$, $|m_0| = |m_1|$

D → A: c = $(g^y, g^z .m_b)$

A → D: $b' \in \{0, 1\}$

El Gamal Implementation Issues



Public Key pk = $(G, o, q, g, h = g^x)$

$m, pk = (G, o, q, g, h = g^x)$

$Enc_{pk}(m)$

$c_1 = g^y$ for random y

$c_2 = h^y . m$

c

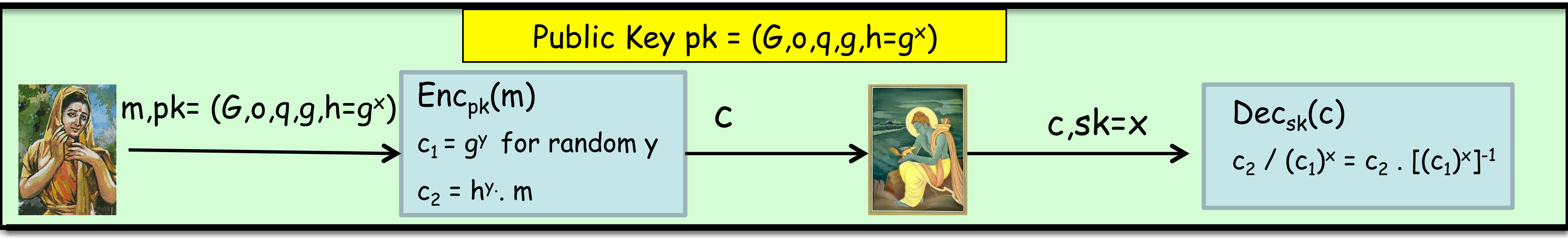
c,sk=x

$Dec_{sk}(c)$

$c_2 / (c_1)^x = c_2 . [(c_1)^x]^{-1}$

- ❑ Sharing public parameters
  - ➢ The public parameters (G, q, g, h) can be publicly shared once-and-for-all
  - ➢ NIST has published standard parameters suitable for El Gamal encryption scheme
  - ➢ Sharing public parameters does not hamper security --- contrast to RSA
- ❑ Choice of groups
  - ➢ Option I: prime order subgroup (G, * mod p) of $\mathbb{Z}_p^*$, where p = 2q+1 and $G = \{x^2 \bmod p \mid x \in \mathbb{Z}_p^*\}$
  - ➢ Option II (Practically popular): groups based on points on elliptic curves
- ❑ Message Space --- not bit strings, but rather group elements. Two possible solutions to deal with this
  - ➢ Option I: Use some efficient reversible encoding mechanism from bit strings to group elements
  - ➢ Option II: Use the El Gamal encryption scheme as a part of a Hybrid encryption scheme

El Gamal Implementation Issues



Public Key pk = $(G,o,q,g,h=g^x)$

m,pk= $(G,o,q,g,h=g^x)$

$Enc_{pk}(m)$

$c_1 = g^y$ for random y

$c_2 = h^y . m$

c

c,sk=x

$Dec_{sk}(c)$

$c_2 / (c_1)^x = c_2 . [(c_1)^x]^{-1}$

❑ Mapping bit strings to group elements

- For concreteness, consider prime order subgroup G of $\mathbb{Z}^*_p$, where p = 2q+1 and G = $\{x^2 \bmod p \mid x \in \mathbb{Z}^*_p\}$

$\mathbb{Z}^*_{11}$:	1	2	3	4	5	6	7	8	9	10	p = 11, q = 5
Squares modulo 11:	1^2	2^2	3^2	4^2	5^2	6^2	7^2	8^2	9^2	10^2	Group G
Values:	1	4	9	5	3	3	5	9	4	1	Plaintext and ciphertext space

- $\mathbb{Z}^*_p$ = {1, 2, …, q, q+1, …, 2q}

- Consider the mapping f: {1, …, q} → G

$$f(x) \overset{def}{=} [x^2 \bmod p]$$

- Let || q || = n bits

- Function f is a bijection
  - ❖ A quadratic residue $[x^2 \bmod p]$ has two modular square roots: [x mod p], [-x mod p]
  - ❖ Only one square root lies in the range {1, …, q}
  - ❖ Function f is efficiently invertible

- Given an (n-1)-bit string $x \in \{0, 1\}^{n-1}$, map it to an element of G as follows:
  - ❖ Compute f(1 || x) --- 1 || x will be an n-bit string, will be an integer in the range {1, …, q}

7th Chalk and Talk topic

Goldwasser-Micali Cryptosystem based on Quadratic Residuacity

8th Chalk and Talk topic

Miller-Rabin Primality Testing

Thank You!