14 Years of Chosen Ciphertext Security: A Survey of Public Key Encryption

Victor Shoup
New York University

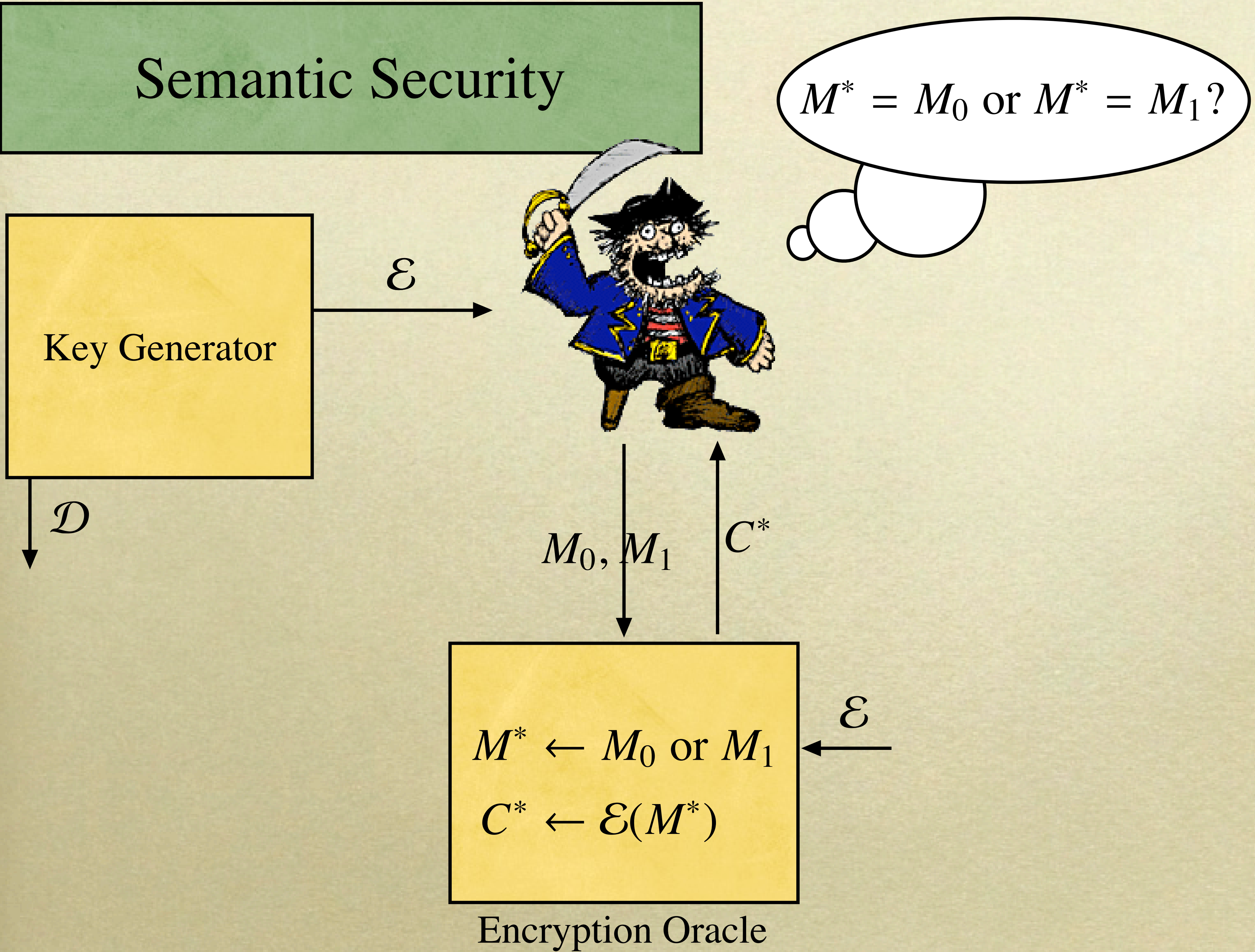
A Historical Perspective

- The **wild** years (mid 70's-mid 80's):
  - Diffie-Hellman, RSA, ElGamal
- The **rigorous** years (mid 80's-early 90's)
  - Definitions, Definitions, Definitions, Definitions, ...
- The **practical** years (early 90's-present)

Notions of Secure Encryption

- Semantic Security [GM84]
- Security Against Non-adaptive Chosen Ciphertext Attack [NY90]
- Security Against Adaptive Chosen Ciphertext Attack [RS91,DDN91]

Semantic Security



$M^* = M_0$ or $M^* = M_1$?

Key Generator

$\mathcal{E}$

$\mathcal{D}$

M_0, M_1

C^*

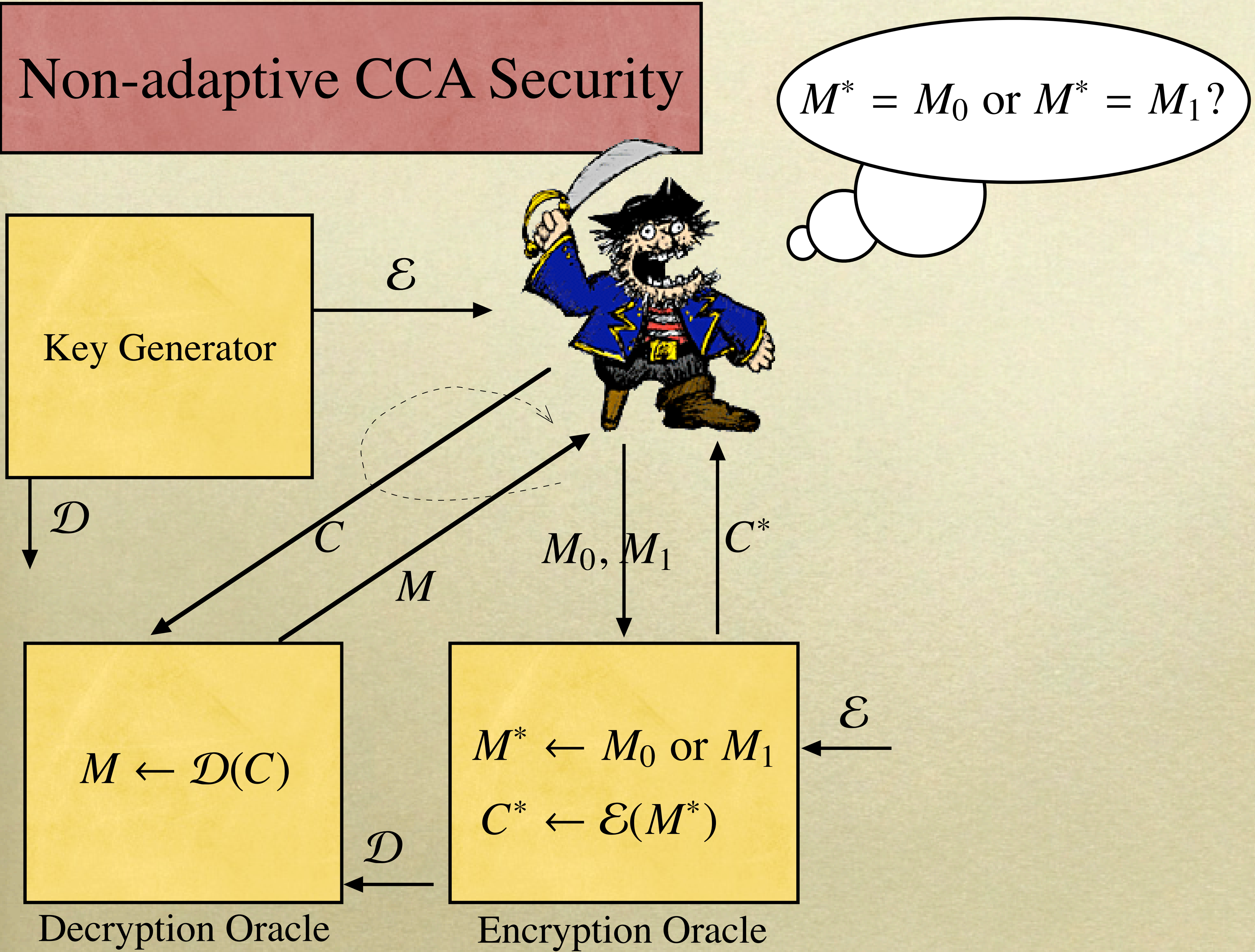
$M^* \leftarrow M_0$ or M_1

$C^* \leftarrow \mathcal{E}(M^*)$

$\mathcal{E}$

Encryption Oracle

Non-adaptive CCA Security



$M^* = M_0$ or $M^* = M_1$?

Key Generator

$\mathcal{E}$

$\mathcal{D}$

C

M

M_0, M_1

C^*

$M \leftarrow \mathcal{D}(C)$

$M^* \leftarrow M_0$ or M_1

$C^* \leftarrow \mathcal{E}(M^*)$

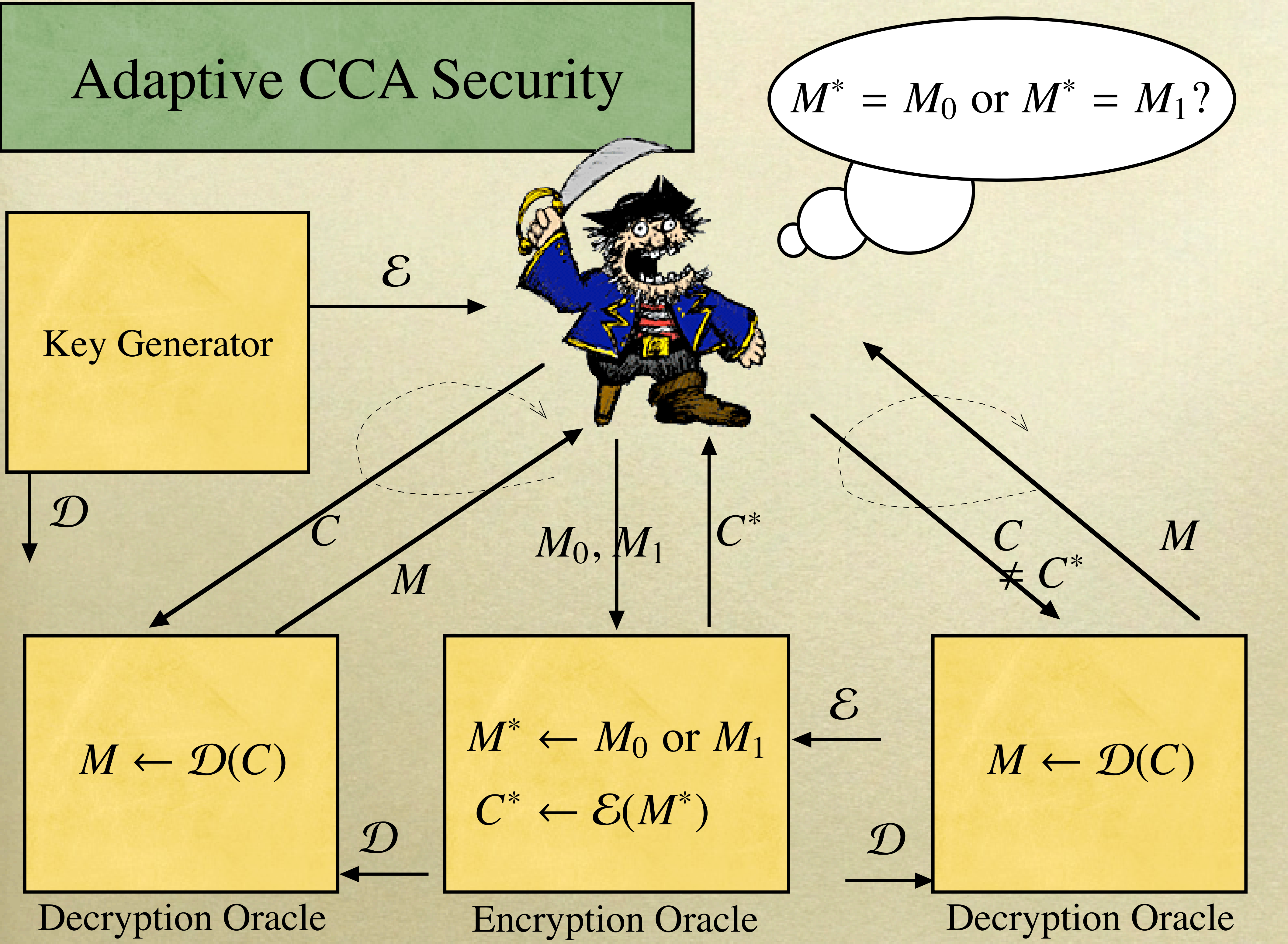
$\mathcal{E}$

$\mathcal{D}$

Decryption Oracle

Encryption Oracle

Adaptive CCA Security



$M^* = M_0$ or $M^* = M_1$?

Key Generator

$\mathcal{E}$

$\mathcal{D}$

C

M

M_0, M_1

C^*

$C \neq C^*$

M

$M \leftarrow \mathcal{D}(C)$

$M^* \leftarrow M_0$ or M_1

$C^* \leftarrow \mathcal{E}(M^*)$

$M \leftarrow \mathcal{D}(C)$

$\mathcal{E}$

$\mathcal{D}$

$\mathcal{D}$

Decryption Oracle

Encryption Oracle

Decryption Oracle

Case Study: RSA

n — RSA modulus

e — encryption exponent

d — decryption exponent

Encrypt $M \in \mathbb{Z}_n$:

$$C \leftarrow M^e$$

Decrypt $C \in \mathbb{Z}_n$:

$$M \leftarrow C^d$$

Case Study: RSA

n — RSA modulus

e — encryption exponent

d — decryption exponent

Encrypt $M \in \mathbb{Z}_n$:

$$C \leftarrow M^e$$

Decrypt $C \in \mathbb{Z}_n$:

$$M \leftarrow C^d$$

Not even semantically secure!

Compare C^* to M_0^e and M_1^e

RSA with Random Padding

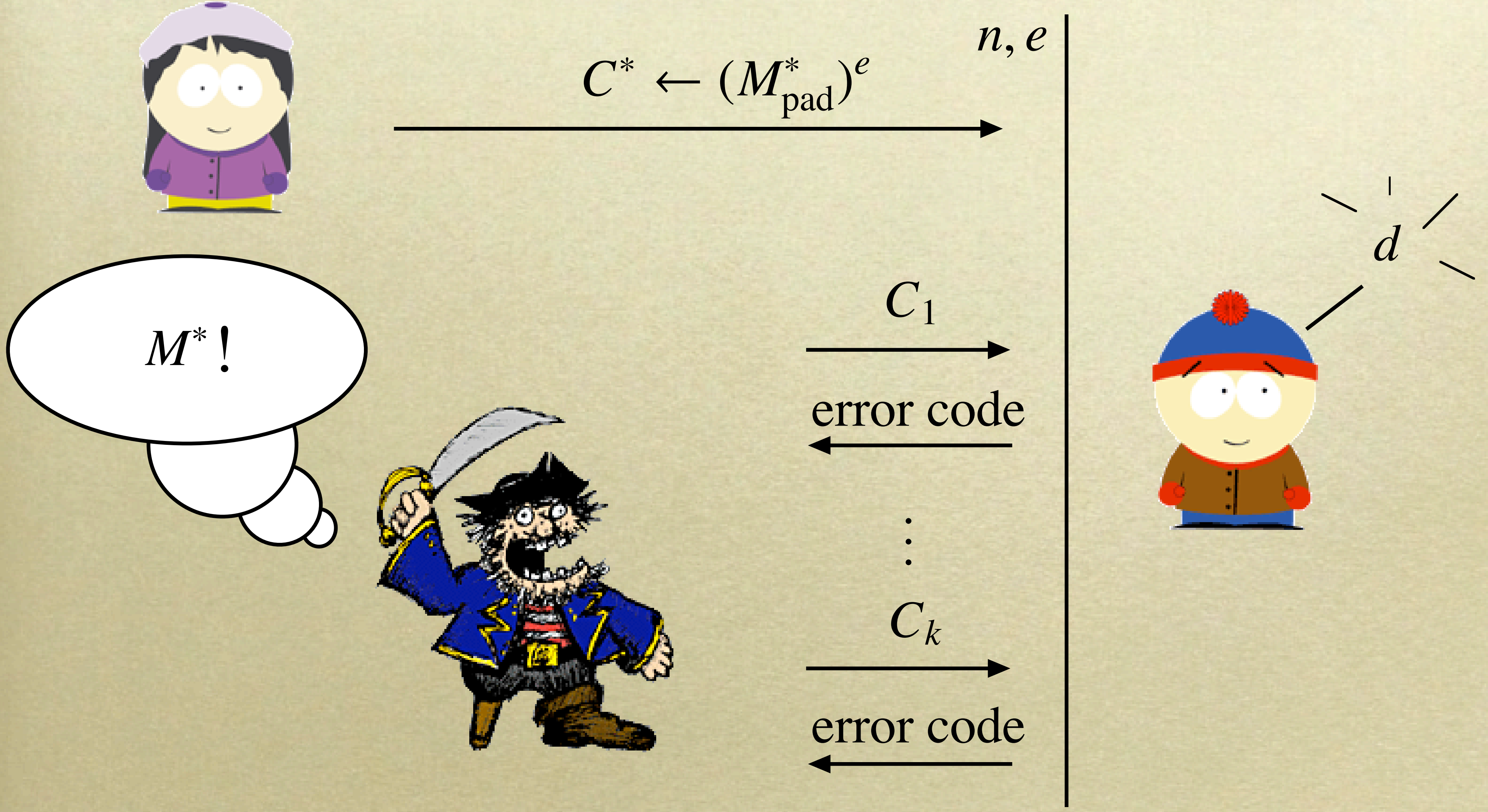
RSA PKCS#1 padding:

00	02	padding string	00	M

M_{pad}

$$C = (M_{\text{pad}})^e$$

Bleichenbacher's Attack on RSA PKCS#1 [B98]



n, e

$C^* \leftarrow (M^*_{\mathrm{pad}})^e$

d

C_1

error code

$\vdots$

C_k

error code

M^* !

Case Study: ElGamal

G — group of prime order q

$\boldsymbol{g}$ — generator for G

KeyGen: $\underbrace{z \xleftarrow{\text{¢}} \mathbb{Z}_q}_{\mathcal{D}}, \quad \underbrace{\boldsymbol{h} \leftarrow \boldsymbol{g}^z}_{\mathcal{E}}$

Encrypt $\boldsymbol{M} \in G$:

$w \xleftarrow{\text{¢}} \mathbb{Z}_q$

$\boldsymbol{a} \leftarrow \boldsymbol{g}^w$

$\boldsymbol{e} \leftarrow \boldsymbol{M}\boldsymbol{h}^w$

$C \leftarrow (\boldsymbol{a}, \boldsymbol{e})$

Decrypt $C = (\boldsymbol{a}, \boldsymbol{e})$:

$M \leftarrow \boldsymbol{e}/\boldsymbol{a}^z$

ElGamal Encryption: Security

Semantically Secure under DDH

(Decisional Diffie-Hellman)

$(\boldsymbol{g}^{r}, \boldsymbol{g}^{s}, \boldsymbol{g}^{rs})$ “looks like” $(\boldsymbol{g}^{r}, \boldsymbol{g}^{s}, \boldsymbol{g}^{t})$

ElGamal Encryption: Security

Semantically Secure under DDH

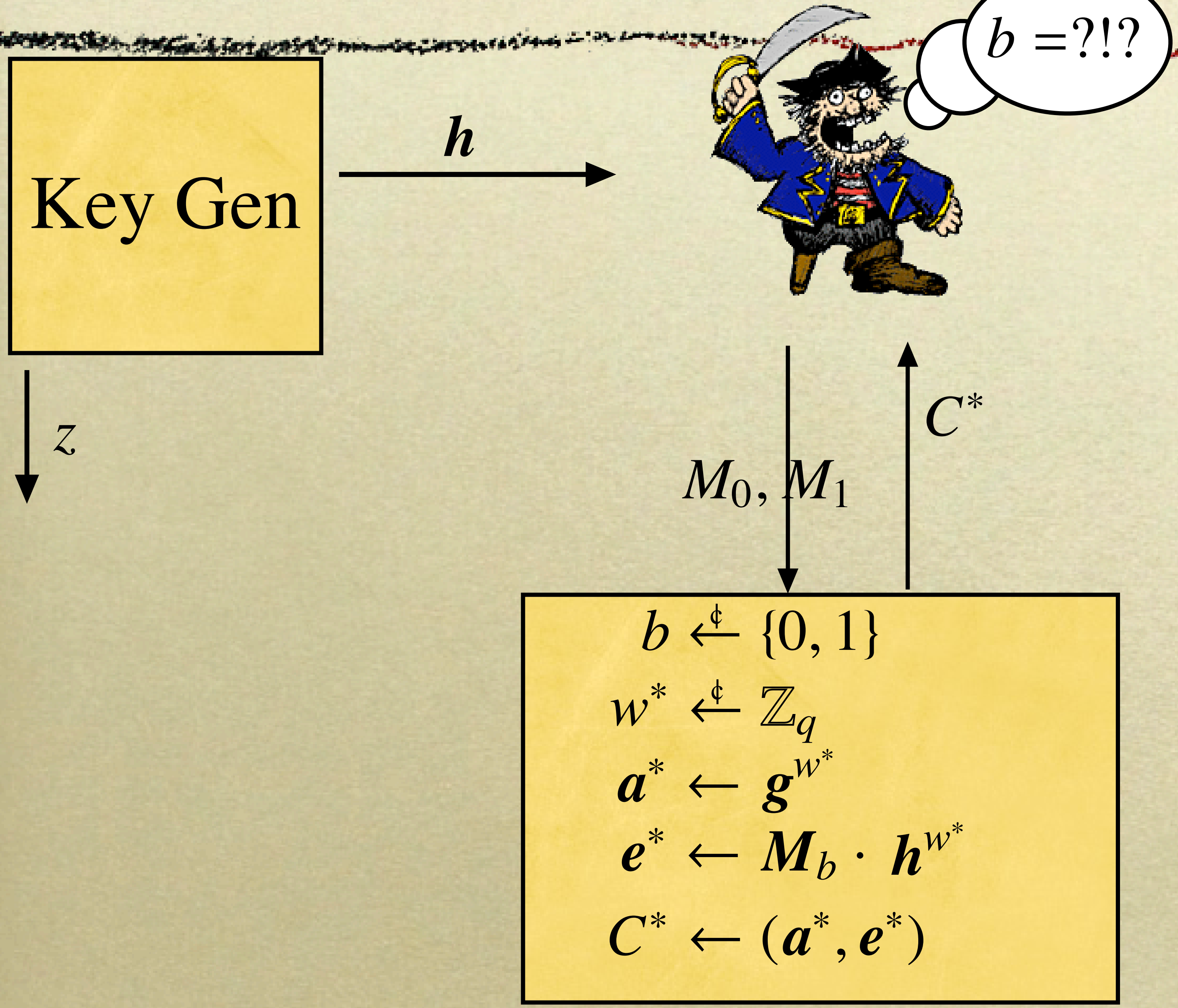
(Decisional Diffie-Hellman)

$(\boldsymbol{g}^r, \boldsymbol{g}^s, \boldsymbol{g}^{rs})$ "looks like" $(\boldsymbol{g}^r, \boldsymbol{g}^s, \boldsymbol{g}^t)$

Insecure against *Adaptive* CCA

Security

G0: Original Game



$b \overset{¢}{\leftarrow} \{0,1\}$

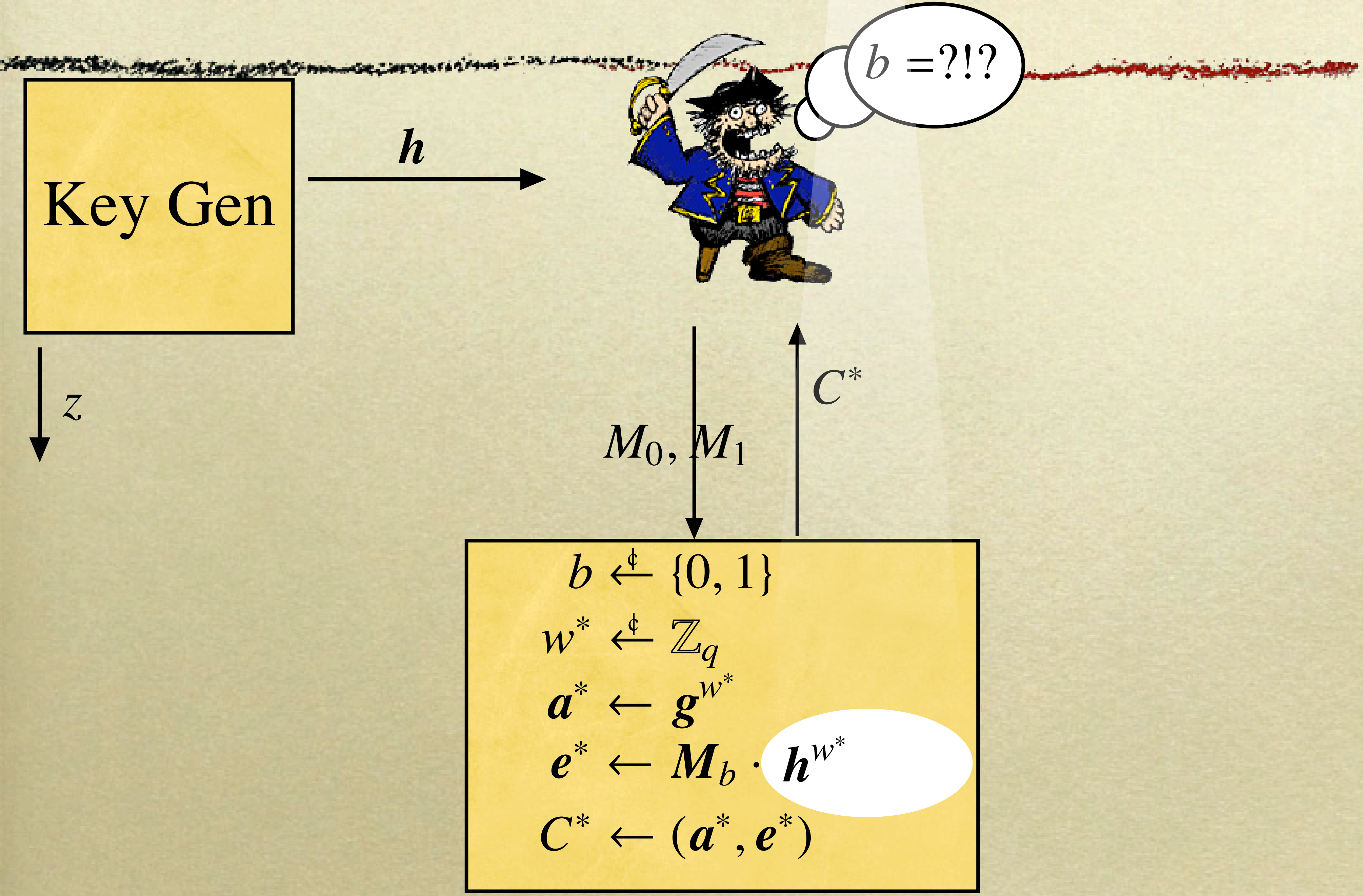
$w^* \overset{¢}{\leftarrow} \mathbb{Z}_q$

$\boldsymbol{a}^* \leftarrow \boldsymbol{g}^{w^*}$

$\boldsymbol{e}^* \leftarrow \boldsymbol{M}_b \cdot \boldsymbol{h}^{w^*}$

$C^* \leftarrow (\boldsymbol{a}^*, \boldsymbol{e}^*)$

Security



$$b \xleftarrow{¢} \{0,1\}$$
$$w^* \xleftarrow{¢} \mathbb{Z}_q$$
$$\boldsymbol{a}^* \leftarrow \boldsymbol{g}^{w^*}$$
$$\boldsymbol{e}^* \leftarrow \boldsymbol{M}_b \cdot \boldsymbol{h}^{w^*}$$
$$C^* \leftarrow (\boldsymbol{a}^*, \boldsymbol{e}^*)$$

Security

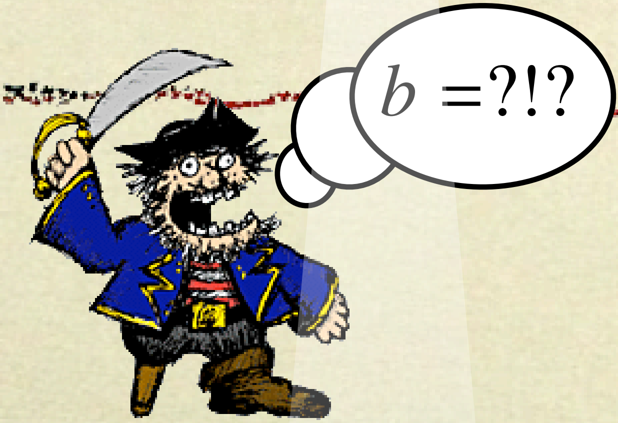
$$\text{G1: DDH} - (\boldsymbol{g}^{w^*}, \boldsymbol{g}^{z}, \boldsymbol{g}^{w^*z}) \approx (\boldsymbol{g}^{w^*}, \boldsymbol{g}^{z}, Random)$$

$\boldsymbol{h}^{w^*}$

Key Gen

$\boldsymbol{h}$

z



M_0, M_1

C^*

$b \xleftarrow{\text{¢}} \{0,1\}$

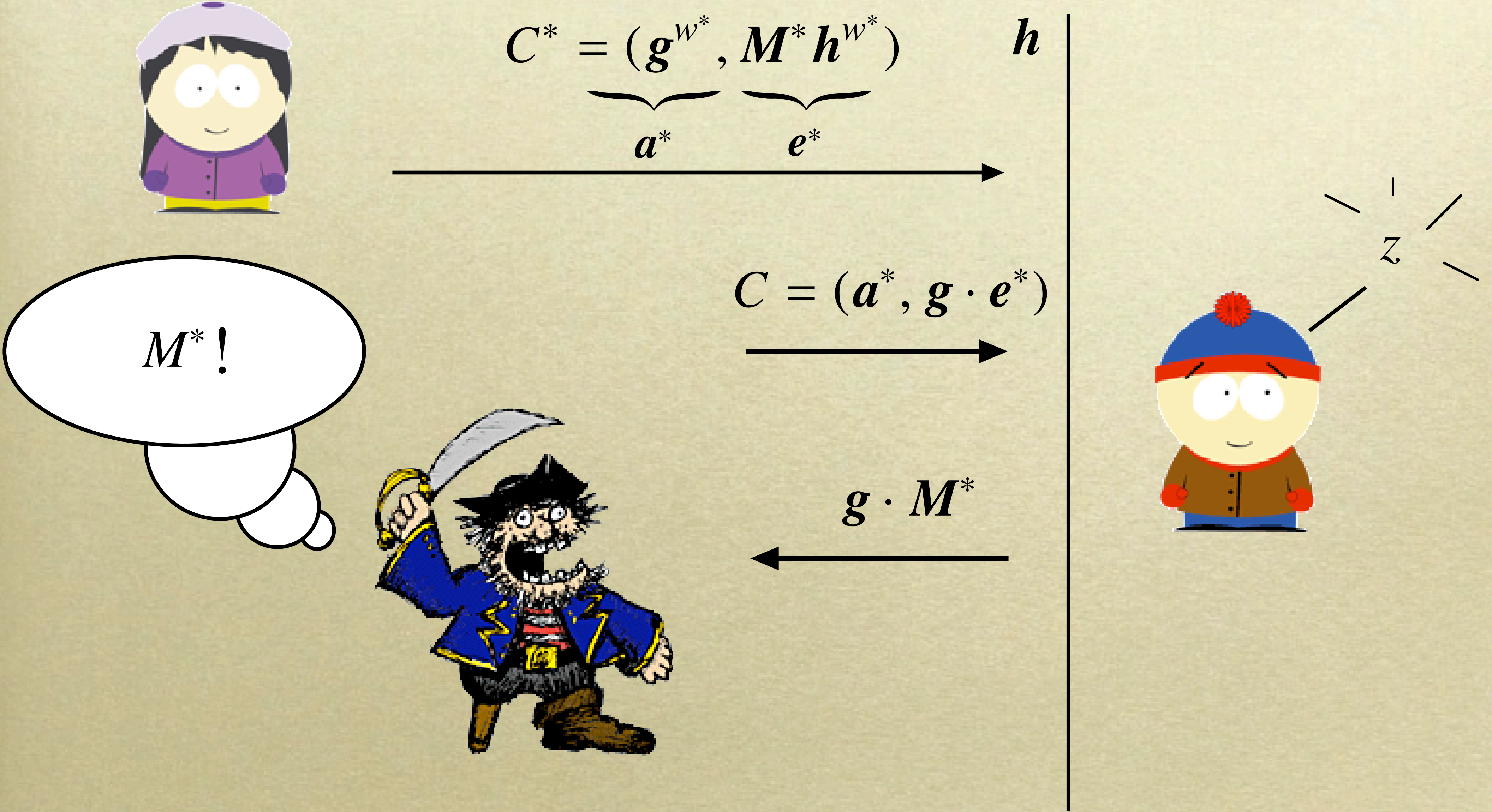
$w^* \xleftarrow{\text{¢}} \mathbb{Z}_q$

$\boldsymbol{a}^* \leftarrow \boldsymbol{g}^{w^*}$

$\boldsymbol{e}^* \leftarrow \boldsymbol{M}_b \cdot Random$

$C^* \leftarrow (\boldsymbol{a}^*, \boldsymbol{e}^*)$

CCA against ElGamal



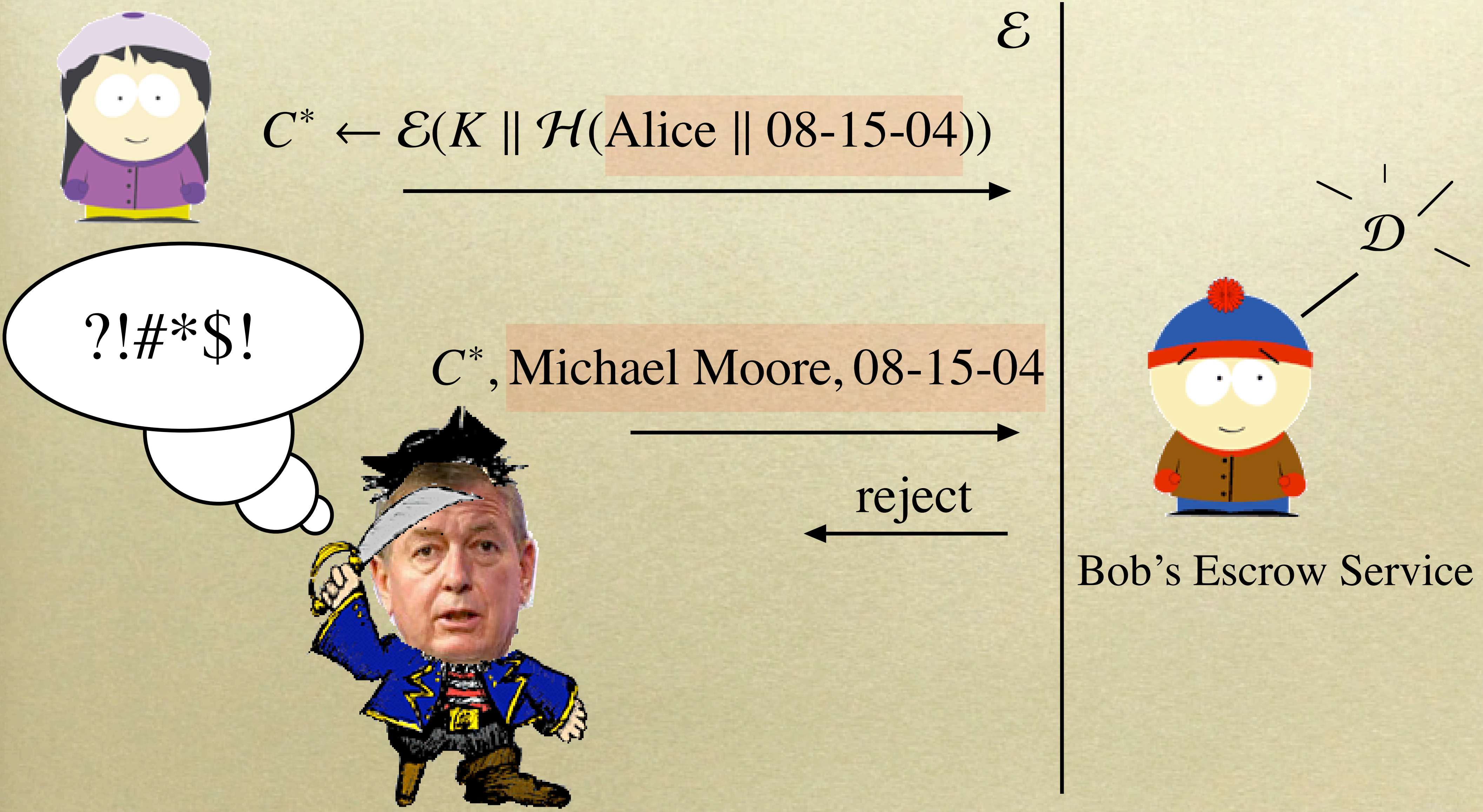
$$C^* = (\underbrace{g^{w^*}}_{a^*}, \underbrace{M^* h^{w^*}}_{e^*})$$

h

$$C = (a^*, g \cdot e^*)$$

$$g \cdot M^*$$

Example: Key Escrow



$\mathcal{E}$

$C^* \leftarrow \mathcal{E}(K \,\|\, \mathcal{H}(\text{Alice} \,\|\, \text{08-15-04}))$

$\mathcal{D}$

C^*, Michael Moore, 08-15-04

reject

Bob's Escrow Service

Labeled Encryption

A label L is input to $\mathcal{E}$ and $\mathcal{D}$

Security:

Adverary submits L^*, M_0, M_1 to encryption oracle

Adversary submits (C, L) to decryption oracle

Restriction: $(C, L) \neq (C^*, L^*)$

Two Key Construction

[NY90,DDN91,S01,L03]

Ingredients:

- NIZK
- semantically secure PKE

Public Key:

- NIZK reference string
- public keys $\mathcal{E}_L$ and $\mathcal{E}_R$ for PKE

Private Key:

- corresponding private keys $\mathcal{D}_L$ and $\mathcal{D}_R$

Two Key Construction

[NY90,DDN91,S01,L03]

Ingredients:

- NIZK
- semantically secure PKE

Public Key:

- NIZK reference string
- public keys $\mathcal{E}_L$ and $\mathcal{E}_R$ for PKE

Private Key:

- corresponding private keys $\mathcal{D}_L$ and $\mathcal{D}_R$

Two Key Construction

[NY90,DDN91,S01,L03]

Encryption of M: (C_L, C_R, π)

where $C_L = \mathcal{E}_L(M)$, $C_R = \mathcal{E}_R(M)$,

π is a proof that C_L and C_R encrypt the same message

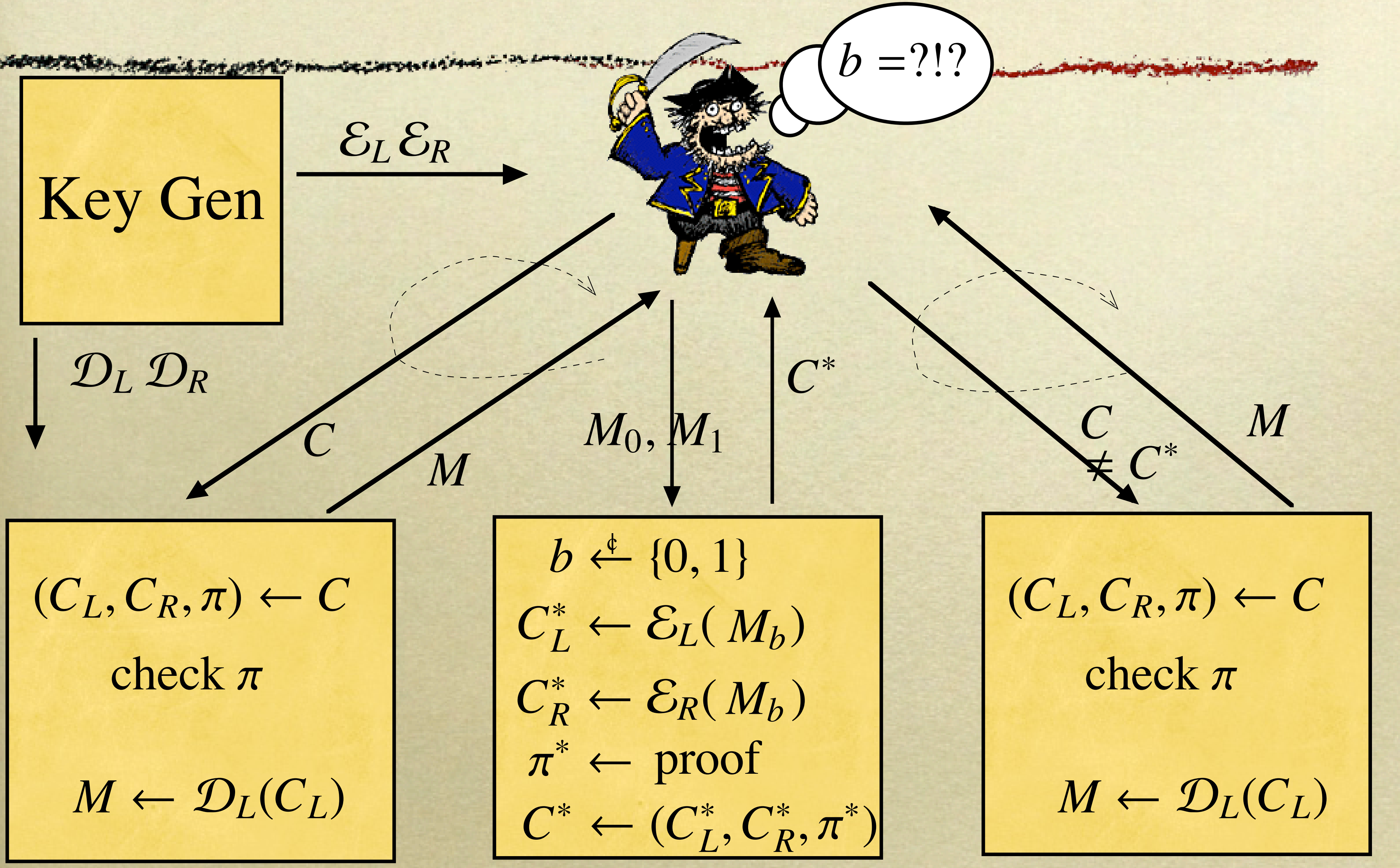
Decryption of (C_L, C_R, π):

check π

output $\mathcal{D}_L(C_L)$

Security

G0: Original Game



b =?!?

Key Gen

$\mathcal{E}_L\ \mathcal{E}_R$

$\mathcal{D}_L\ \mathcal{D}_R$

C

M

M_0, M_1

C^*

$C \neq C^*$

M

$(C_L, C_R, \pi) \leftarrow C$

check π

$M \leftarrow \mathcal{D}_L(C_L)$

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

$C^*_L \leftarrow \mathcal{E}_L(M_b)$

$C^*_R \leftarrow \mathcal{E}_R(M_b)$

$\pi^* \leftarrow$ proof

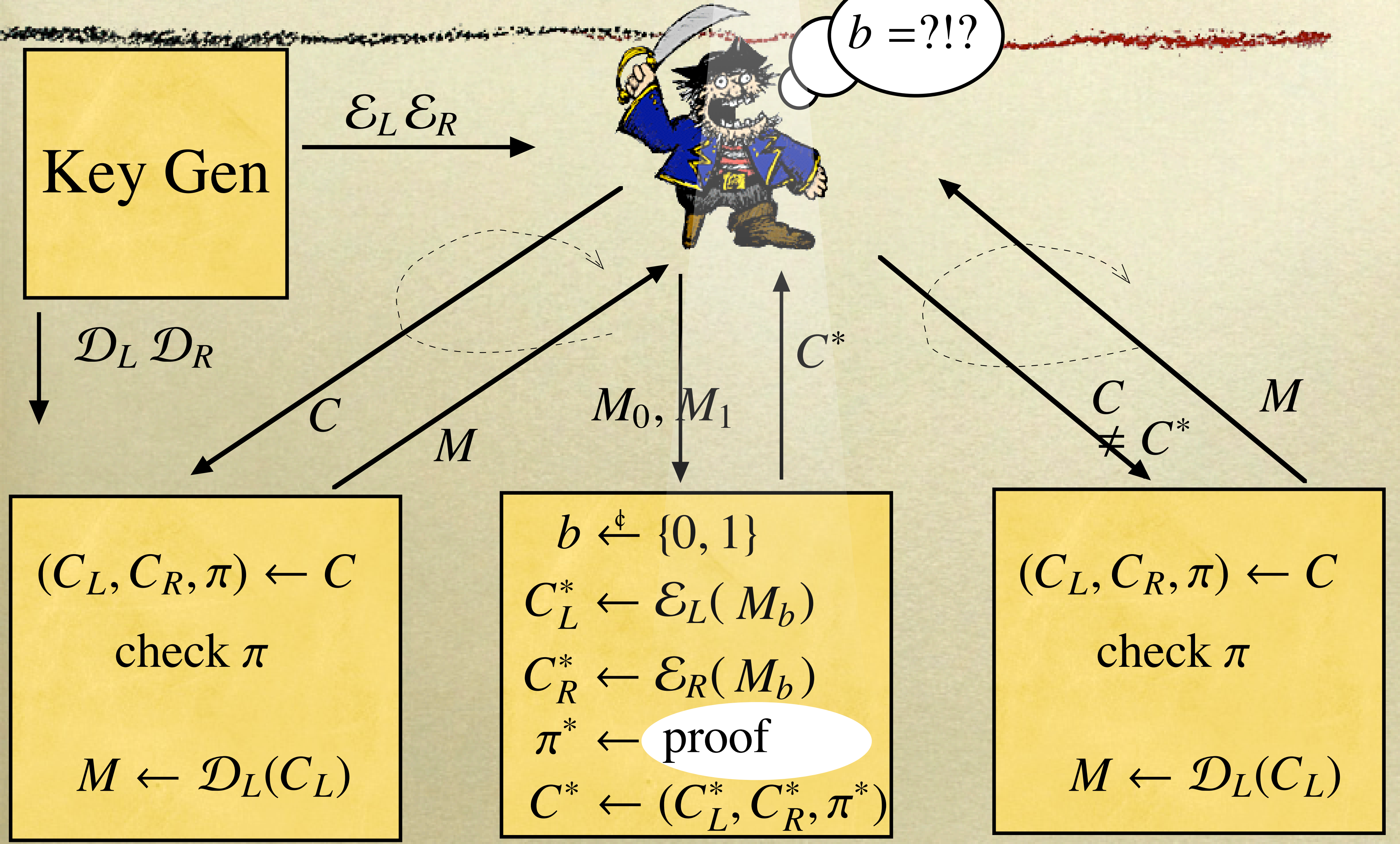
$C^* \leftarrow (C^*_L, C^*_R, \pi^*)$

$(C_L, C_R, \pi) \leftarrow C$

check π

$M \leftarrow \mathcal{D}_L(C_L)$

Security



b =?!?

Key Gen

$\mathcal{E}_L \, \mathcal{E}_R$

$\mathcal{D}_L \, \mathcal{D}_R$

C

M

M_0, M_1

C^*

$C \neq C^*$

M

$(C_L, C_R, \pi) \leftarrow C$

check π

$M \leftarrow \mathcal{D}_L(C_L)$

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

$C_L^* \leftarrow \mathcal{E}_L(M_b)$

$C_R^* \leftarrow \mathcal{E}_R(M_b)$

$\pi^* \leftarrow$ proof

$C^* \leftarrow (C_L^*, C_R^*, \pi^*)$

$(C_L, C_R, \pi) \leftarrow C$

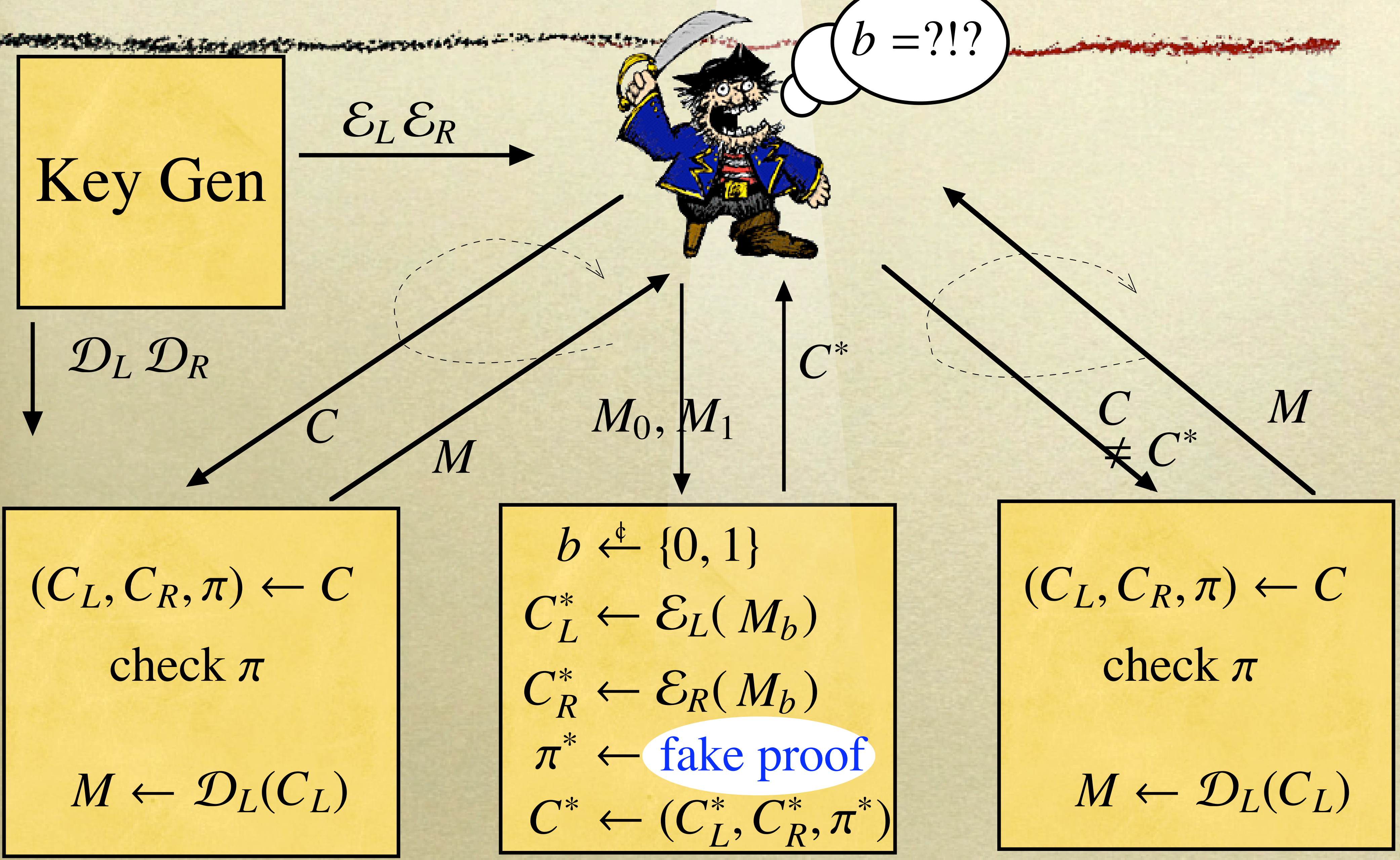
check π

$M \leftarrow \mathcal{D}_L(C_L)$

Security

proof

G1: Zero Knowledge



Key Gen

$\mathcal{E}_L\ \mathcal{E}_R$

$\mathcal{D}_L\ \mathcal{D}_R$

$b = ?!?$

C

M

M_0, M_1

C^*

$C \neq C^*$

M

$(C_L, C_R, \pi) \leftarrow C$

check π

$M \leftarrow \mathcal{D}_L(C_L)$

$b \xleftarrow{¢} \{0, 1\}$

$C_L^* \leftarrow \mathcal{E}_L(\, M_b\,)$

$C_R^* \leftarrow \mathcal{E}_R(\, M_b\,)$

$\pi^* \leftarrow$ fake proof

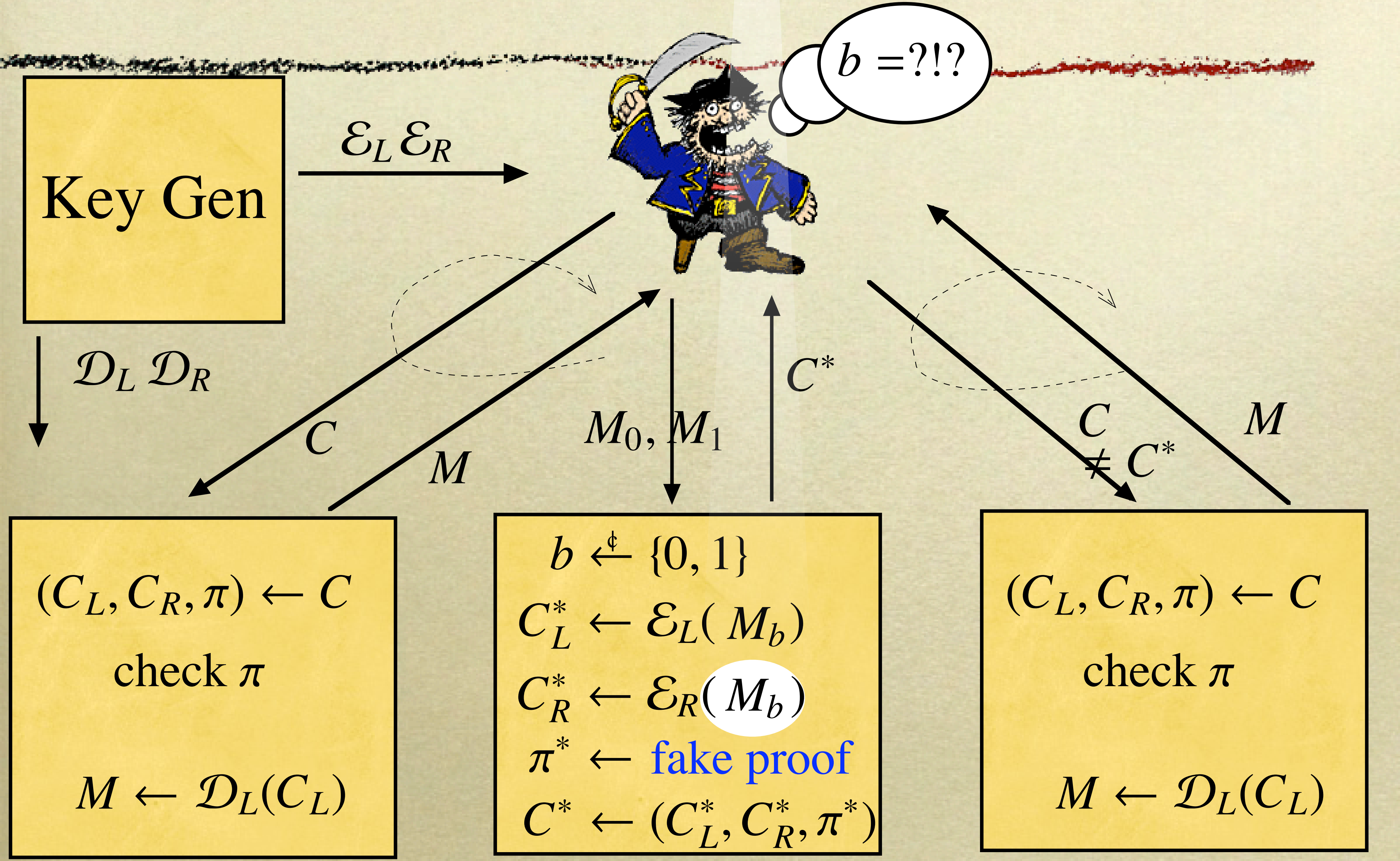
$C^* \leftarrow (C_L^*, C_R^*, \pi^*)$

$(C_L, C_R, \pi) \leftarrow C$

check π

$M \leftarrow \mathcal{D}_L(C_L)$

Security



Key Gen

$(C_L, C_R, \pi) \leftarrow C$

check π

$M \leftarrow \mathcal{D}_L(C_L)$

$b \overset{\text{¢}}{\leftarrow} \{0,1\}$

$C_L^* \leftarrow \mathcal{E}_L(M_b)$

$C_R^* \leftarrow \mathcal{E}_R(M_b)$

$\pi^* \leftarrow$ fake proof

$C^* \leftarrow (C_L^*, C_R^*, \pi^*)$

$(C_L, C_R, \pi) \leftarrow C$

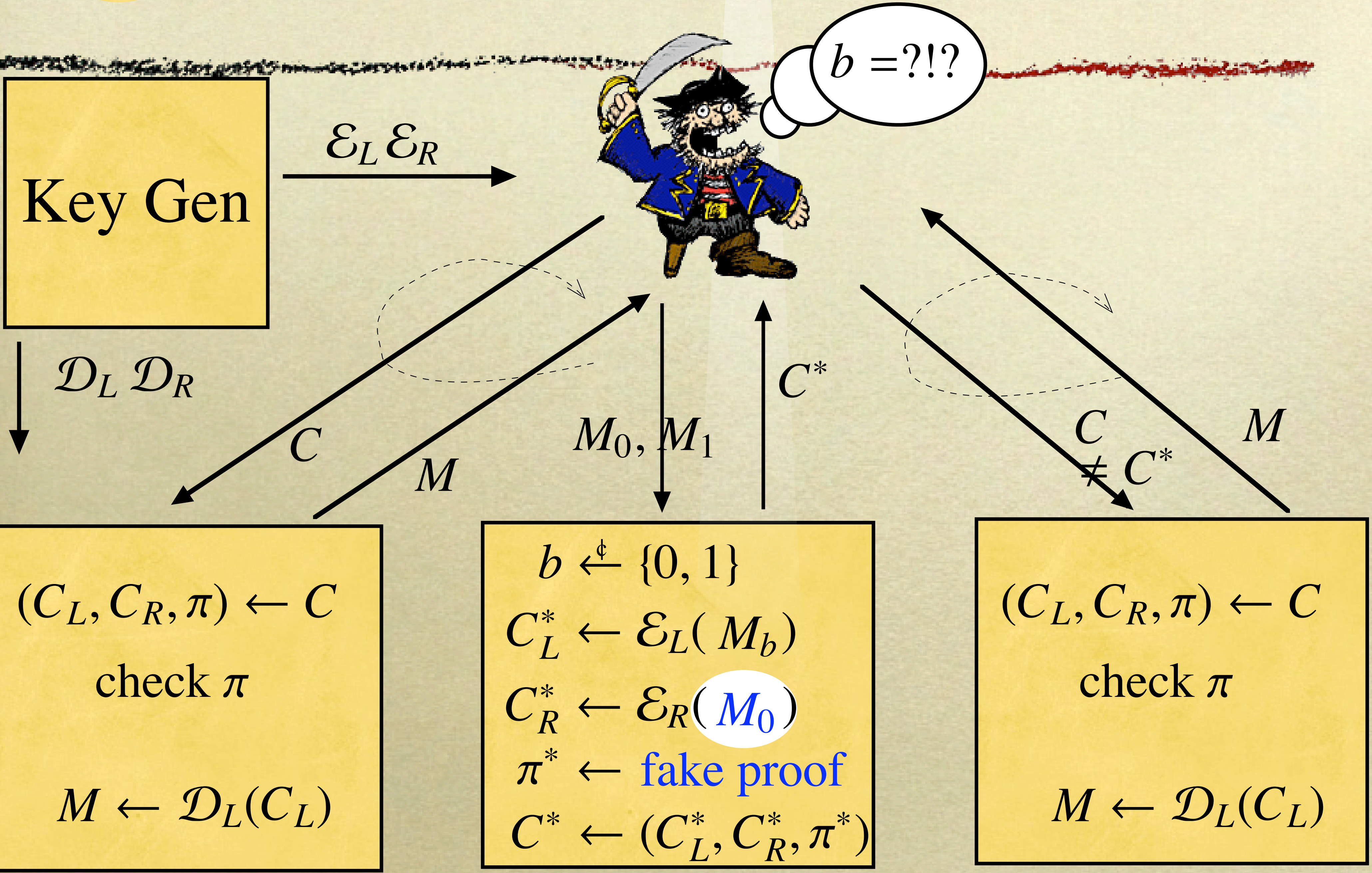
check π

$M \leftarrow \mathcal{D}_L(C_L)$

Security

M_b

G2: Semantic Security (right)



b =?!?

Key Gen

$\mathcal{E}_L\ \mathcal{E}_R$

$\mathcal{D}_L\ \mathcal{D}_R$

C

M

M_0, M_1

C^*

C
$\neq C^*$

M

$(C_L, C_R, \pi) \leftarrow C$

check π

$M \leftarrow \mathcal{D}_L(C_L)$

$b \overset{\text{¢}}{\leftarrow} \{0, 1\}$

$C_L^* \leftarrow \mathcal{E}_L(M_b)$

$C_R^* \leftarrow \mathcal{E}_R(M_0)$

$\pi^* \leftarrow$ fake proof

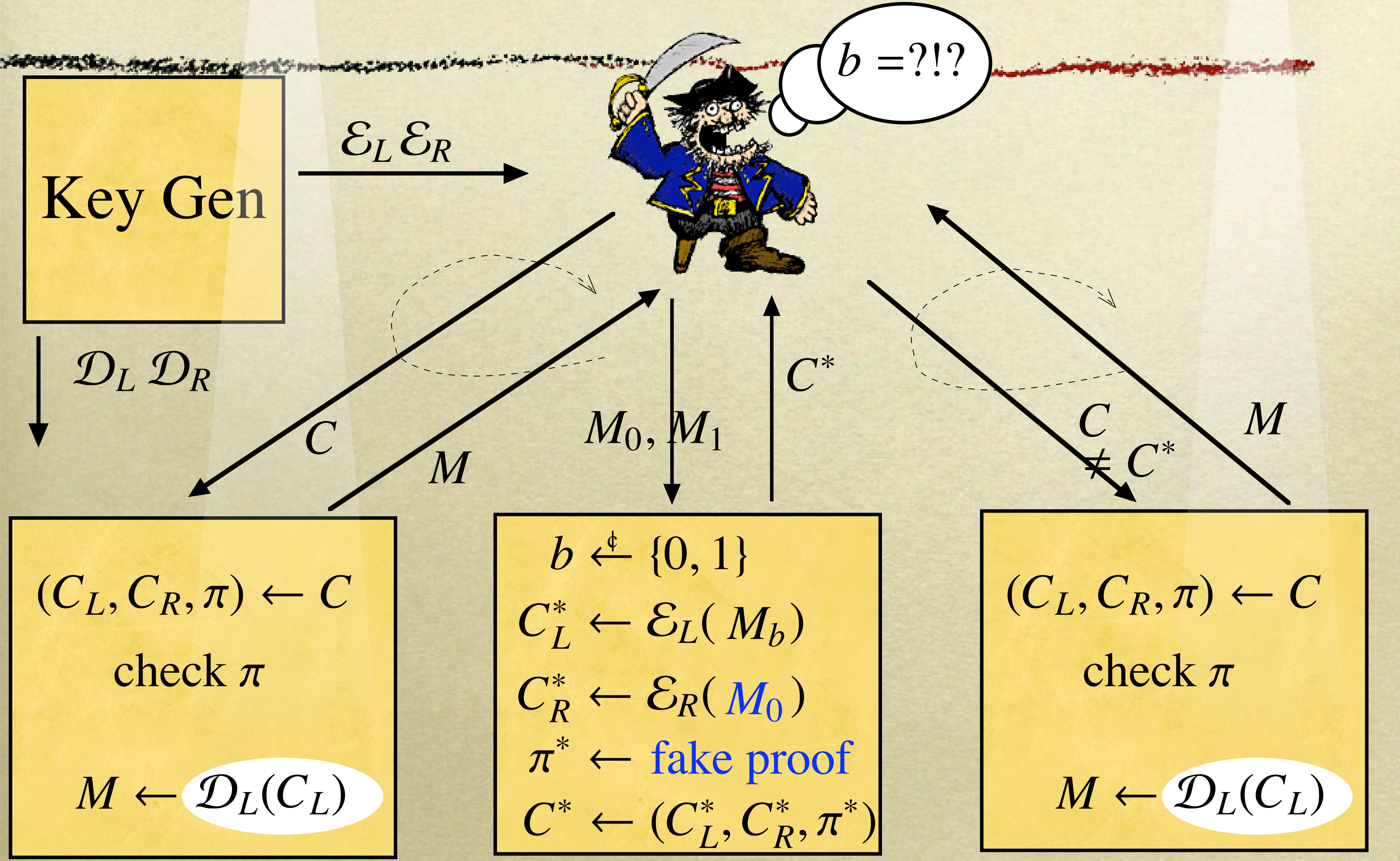
$C^* \leftarrow (C_L^*, C_R^*, \pi^*)$

$(C_L, C_R, \pi) \leftarrow C$

check π

$M \leftarrow \mathcal{D}_L(C_L)$

Security



$b = ?!?$

Key Gen

$\mathcal{E}_L\ \mathcal{E}_R$

$\mathcal{D}_L\ \mathcal{D}_R$

C

M

M_0, M_1

C^*

$C \neq C^*$

M

$(C_L, C_R, \pi) \leftarrow C$

check π

$M \leftarrow \mathcal{D}_L(C_L)$

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

$C_L^* \leftarrow \mathcal{E}_L(M_b)$

$C_R^* \leftarrow \mathcal{E}_R(M_0)$

$\pi^* \leftarrow$ fake proof

$C^* \leftarrow (C_L^*, C_R^*, \pi^*)$

$(C_L, C_R, \pi) \leftarrow C$

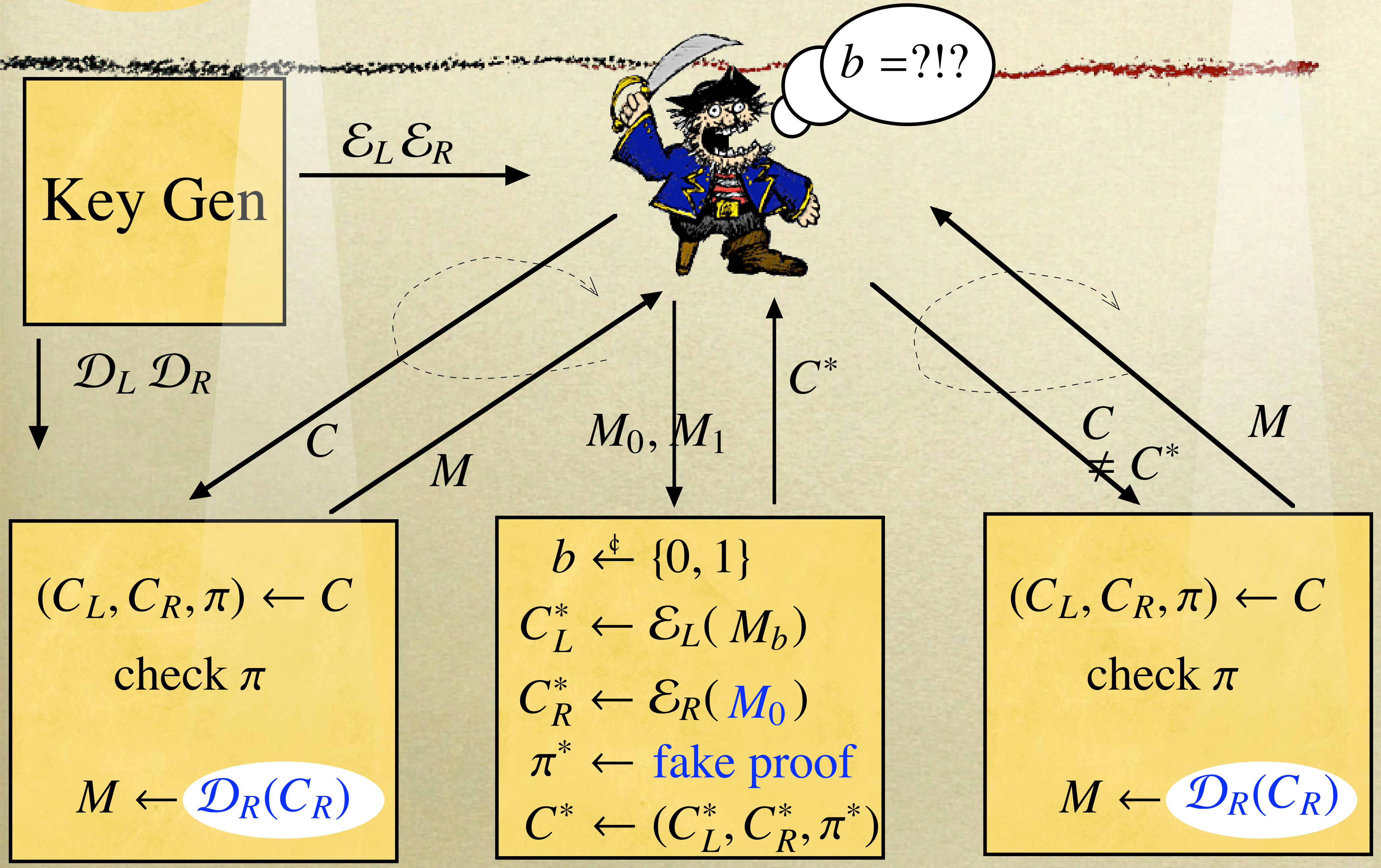
check π

$M \leftarrow \mathcal{D}_L(C_L)$

Security

$\mathcal{D}_L(C_L)$

G3: Simulation Soundness



Key Gen

$\mathcal{E}_L\ \mathcal{E}_R$

$\mathcal{D}_L\ \mathcal{D}_R$

C

M

M_0, M_1

C^*

$C \neq C^*$

M

$(C_L, C_R, \pi) \leftarrow C$

check π

$M \leftarrow \mathcal{D}_R(C_R)$

$b \overset{¢}{\leftarrow} \{0, 1\}$

$C_L^* \leftarrow \mathcal{E}_L(\ M_b\ )$

$C_R^* \leftarrow \mathcal{E}_R(\ M_0\ )$

$\pi^* \leftarrow$ fake proof

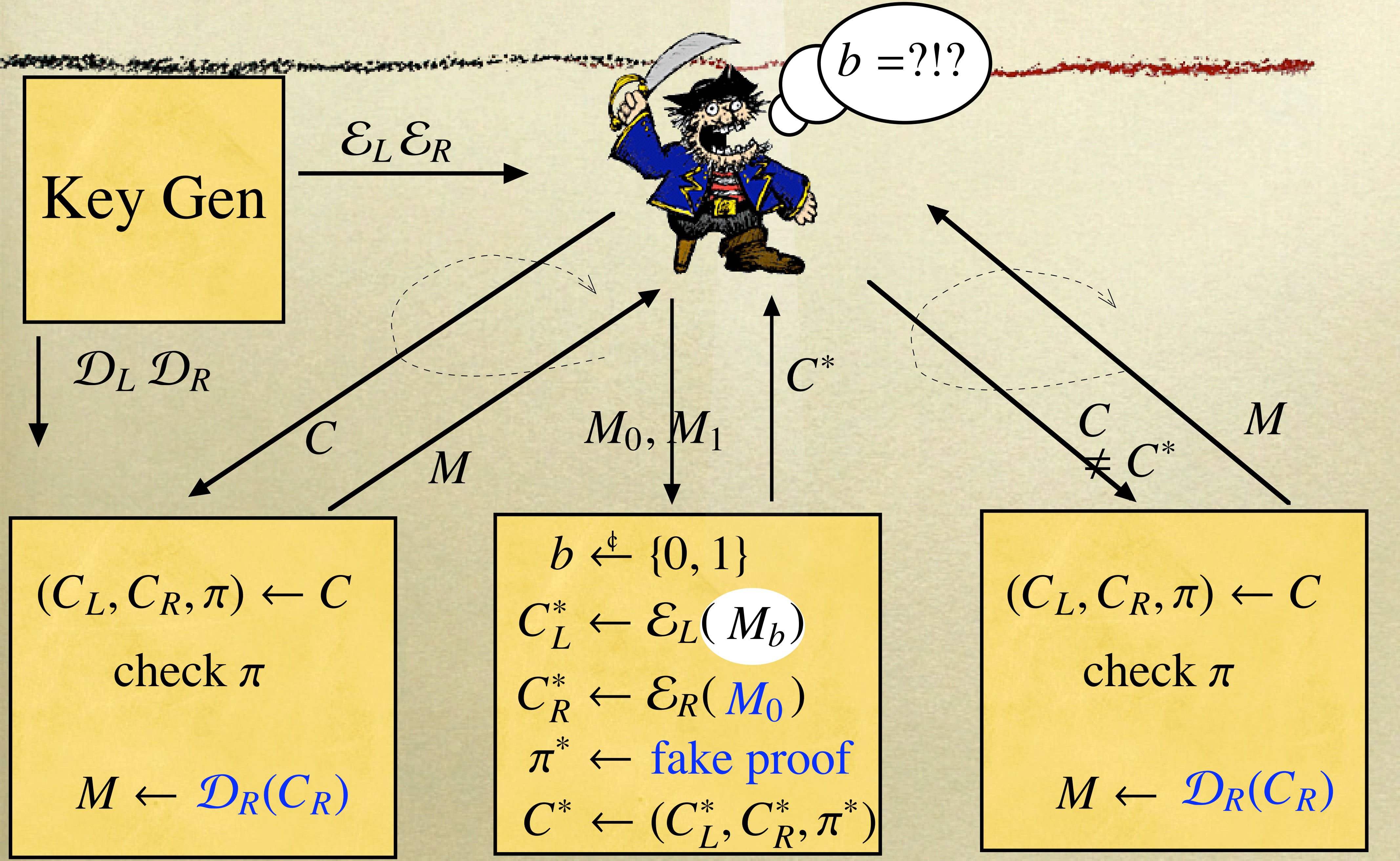
$C^* \leftarrow (C_L^*, C_R^*, \pi^*)$

$(C_L, C_R, \pi) \leftarrow C$

check π

$M \leftarrow \mathcal{D}_R(C_R)$

Security



Key Gen

$\mathcal{E}_L\ \mathcal{E}_R$

$\mathcal{D}_L\ \mathcal{D}_R$

C

M

M_0, M_1

C^*

$C \neq C^*$

M

$(C_L, C_R, \pi) \leftarrow C$

check π

$M \leftarrow \mathcal{D}_R(C_R)$

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

$C_L^* \leftarrow \mathcal{E}_L(M_b)$

$C_R^* \leftarrow \mathcal{E}_R(M_0)$

$\pi^* \leftarrow$ fake proof

$C^* \leftarrow (C_L^*, C_R^*, \pi^*)$

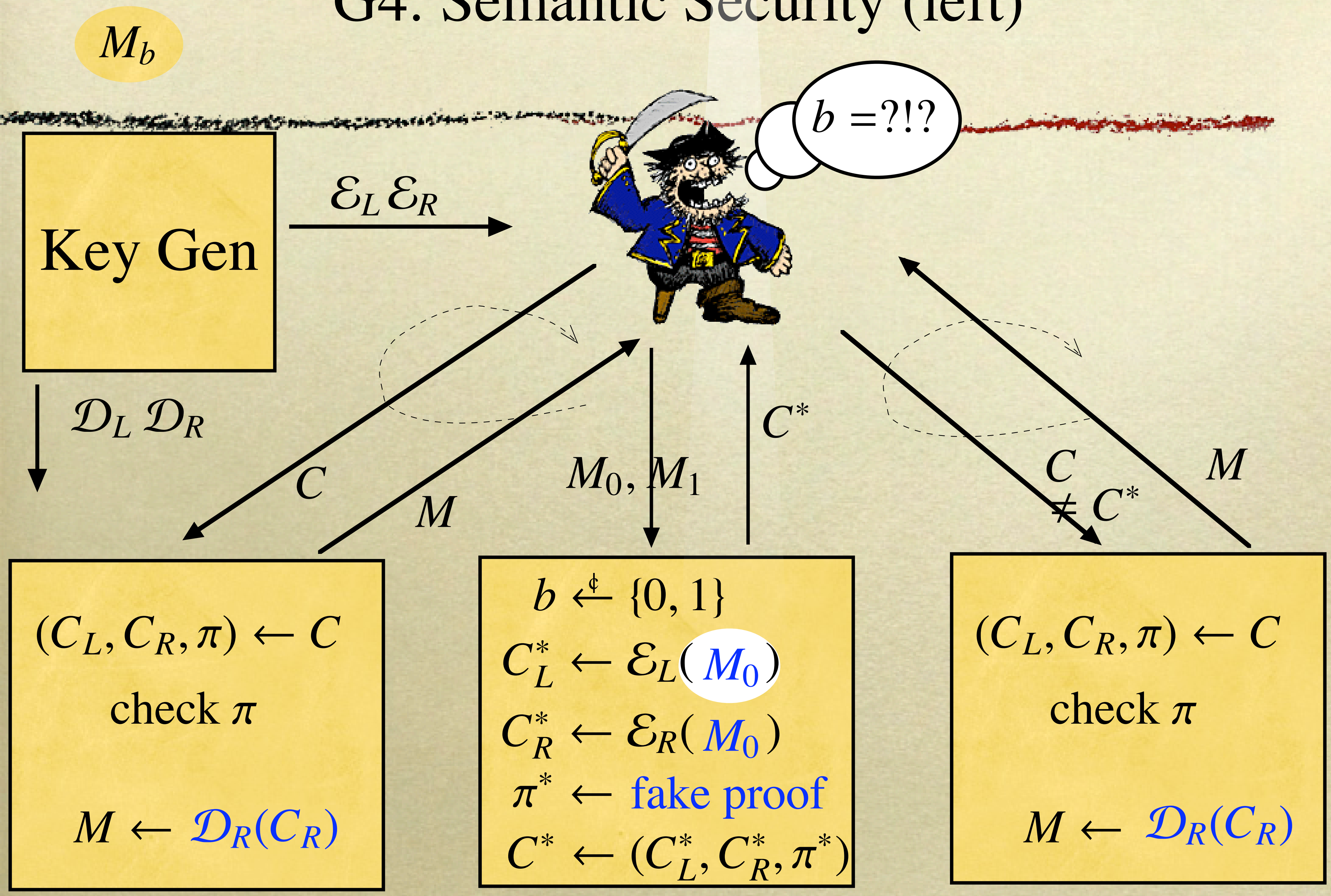
$(C_L, C_R, \pi) \leftarrow C$

check π

$M \leftarrow \mathcal{D}_R(C_R)$

Security

G4: Semantic Security (left)



M_b

$b = ?!?$

Key Gen

$\mathcal{E}_L\ \mathcal{E}_R$

$\mathcal{D}_L\ \mathcal{D}_R$

C

M

M_0, M_1

C^*

$C \neq C^*$

M

$(C_L, C_R, \pi) \leftarrow C$

check π

$M \leftarrow \mathcal{D}_R(C_R)$

$b \xleftarrow{¢} \{0,1\}$

$C_L^* \leftarrow \mathcal{E}_L(M_0)$

$C_R^* \leftarrow \mathcal{E}_R(M_0)$

$\pi^* \leftarrow$ fake proof

$C^* \leftarrow (C_L^*, C_R^*, \pi^*)$

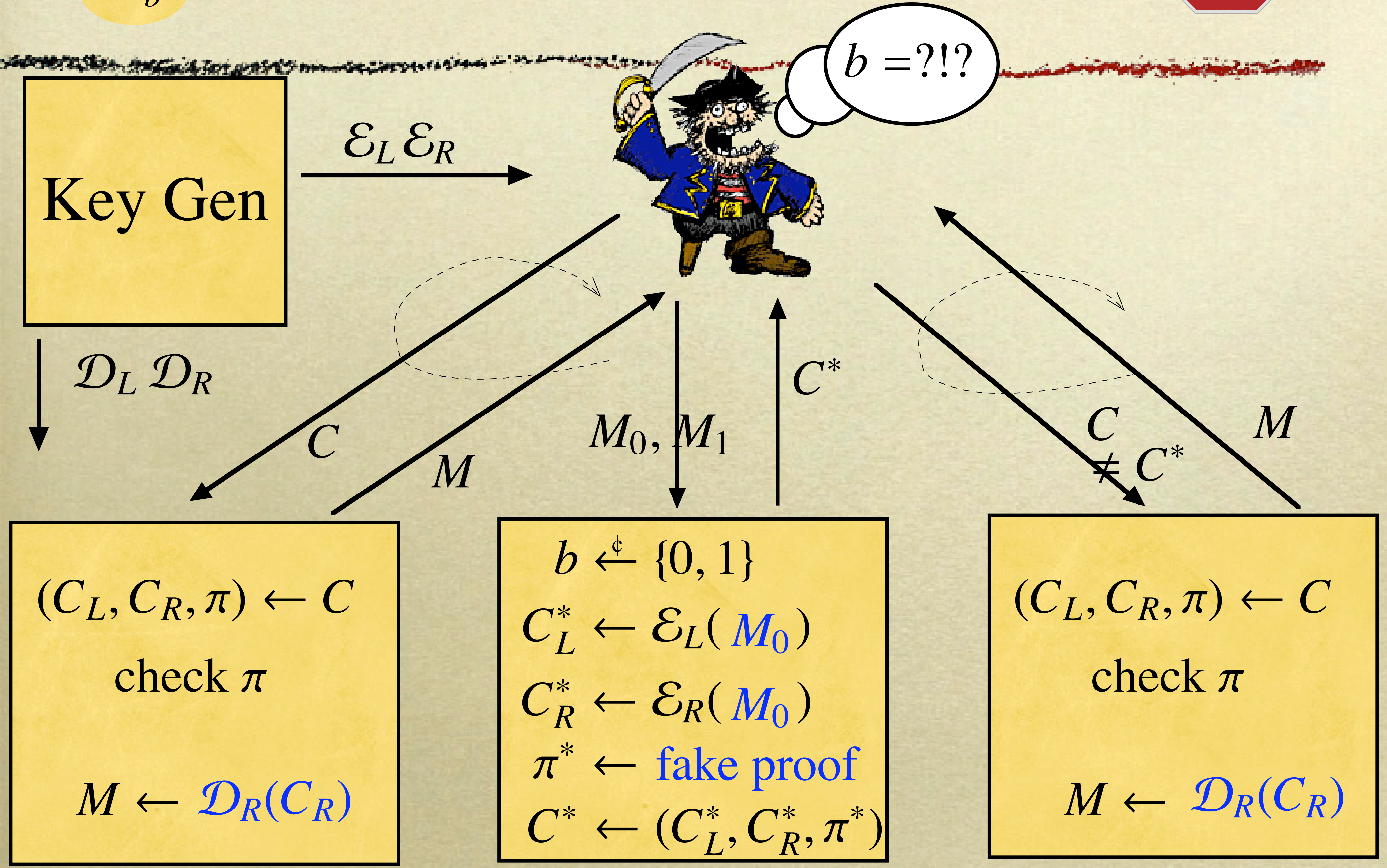
$(C_L, C_R, \pi) \leftarrow C$

check π

$M \leftarrow \mathcal{D}_R(C_R)$

G4: Semantic Security (left)

M_b



Key Gen

$\mathcal{E}_L\ \mathcal{E}_R$

$\mathcal{D}_L\ \mathcal{D}_R$

C

M

M_0, M_1

C^*

$C \neq C^*$

M

$(C_L, C_R, \pi) \leftarrow C$

check π

$M \leftarrow \mathcal{D}_R(C_R)$

$b \overset{¢}{\leftarrow} \{0,1\}$

$C_L^* \leftarrow \mathcal{E}_L(M_0)$

$C_R^* \leftarrow \mathcal{E}_R(M_0)$

$\pi^* \leftarrow$ fake proof

$C^* \leftarrow (C_L^*, C_R^*, \pi^*)$

$(C_L, C_R, \pi) \leftarrow C$

check π

$M \leftarrow \mathcal{D}_R(C_R)$

Efficient NIZKs

- General NIZKs are impractical, but...
- Proofs for special languages
- Designated verifier proofs

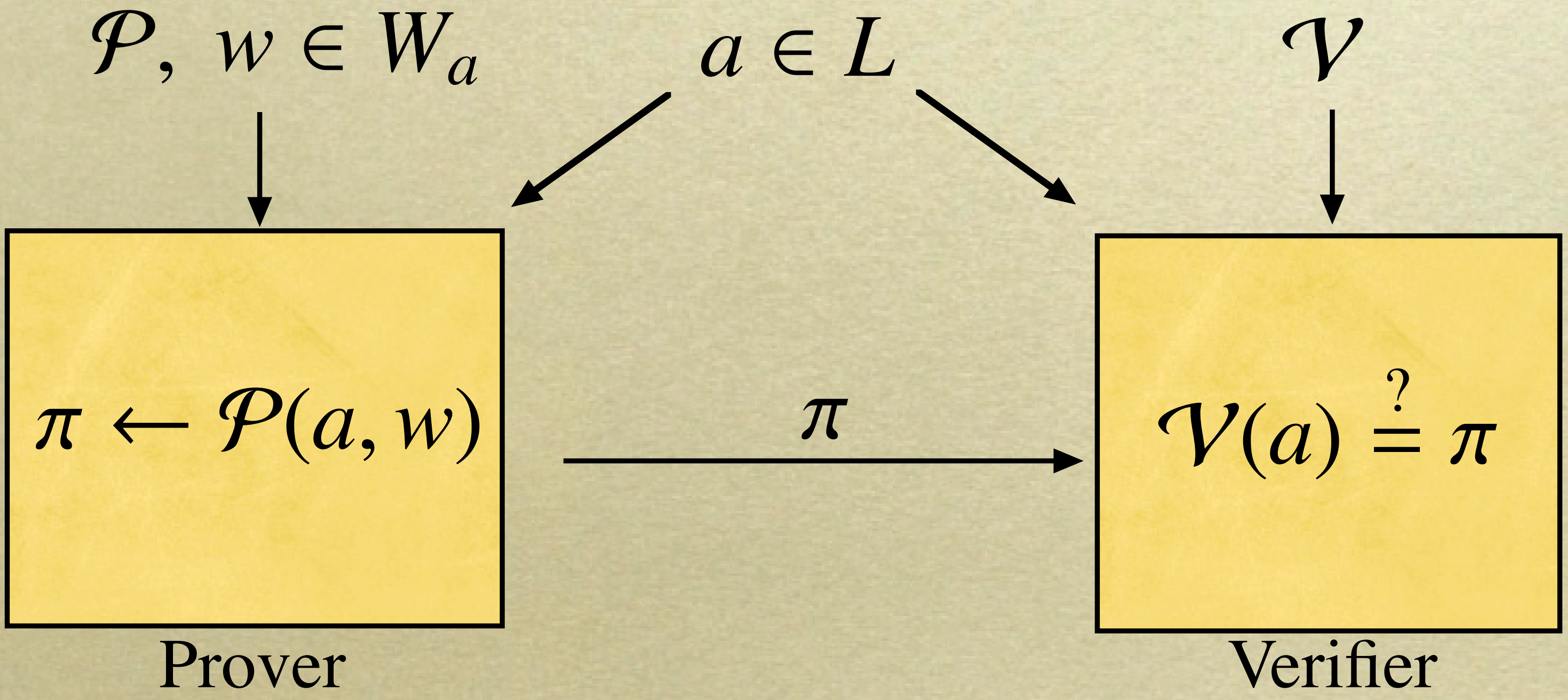
Hash Proof Systems

$L \subset U$; for $a \in L$, $W_a = \{\text{witnesses for } a\} \subset W$

KeyGen $\mapsto (\mathcal{P}, \mathcal{V})$

Proof Function $\mathcal{P} : L \times W \to \Pi$

Verification Function $\mathcal{V} : U \to \Pi$



$\mathcal{P},\ w \in W_a$

$a \in L$

$\mathcal{V}$

$\pi \leftarrow \mathcal{P}(a, w)$

π

$\mathcal{V}(a) \stackrel{?}{=} \pi$

Prover

Verifier

Completeness:

$$\forall a \in L, w \in W_a : \mathcal{P}(a, w) = \mathcal{V}(a)$$

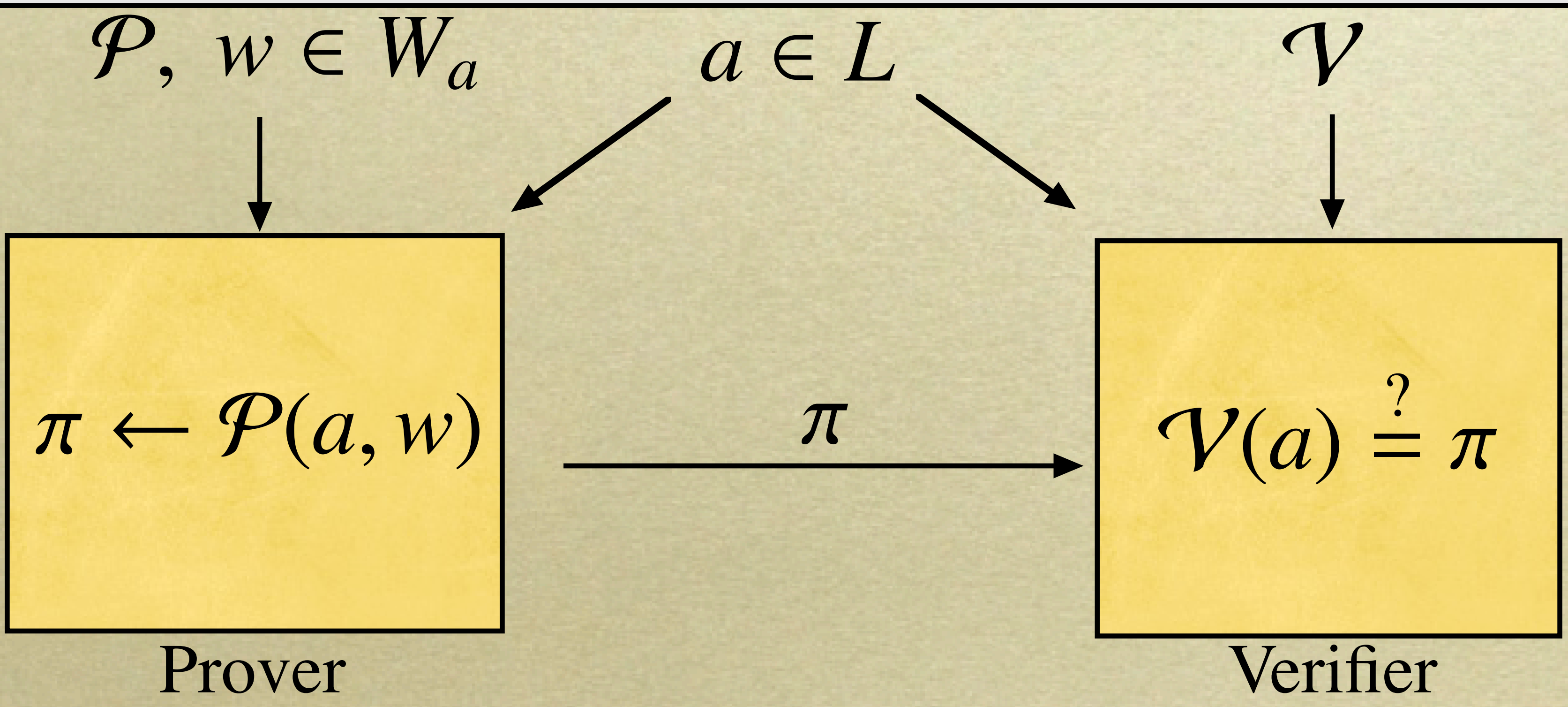
Soundness:

$\forall a \in \overline{L} : \mathcal{V}(a)$ looks random, given $\mathcal{P}$

Simulation Soundness:

$\forall a, b \in \overline{L}$ with $a \neq b$:

$\mathcal{V}(b)$ looks random, given $\mathcal{V}(a)$ and $\mathcal{P}$



$\mathcal{P},\ w \in W_a$

$a \in L$

$\mathcal{V}$

$\pi \leftarrow \mathcal{P}(a, w)$

π

$\mathcal{V}(a) \overset{?}{=} \pi$

Prover

Verifier

Example: Equality of DL

G — a group of large prime order q

$\boldsymbol{g}_1, \boldsymbol{g}_2$ — generators for G

$U = G \times G,\ L = \{(\boldsymbol{g}_1^w, \boldsymbol{g}_2^w) : w \in \mathbb{Z}_q\}$

w is the witness

Example: Equality of DL

G — a group of large prime order q

$\boldsymbol{g}_1, \boldsymbol{g}_2$ — generators for G

$U = G \times G,\ L = \{(\boldsymbol{g}_1^w, \boldsymbol{g}_2^w) : w \in \mathbb{Z}_q\}$

KeyGen:

$$\underbrace{x_1, x_2 \xleftarrow{\$} \mathbb{Z}_q,}_{\mathcal{V}} \qquad \underbrace{\boldsymbol{c} \leftarrow \boldsymbol{g}_1^{x_1} \boldsymbol{g}_2^{x_2}}_{\mathcal{P}}$$

Example: Equality of DL

G — a group of large prime order q

$\boldsymbol{g}_1, \boldsymbol{g}_2$ — generators for G

$$U = G \times G,\ L = \{(\boldsymbol{g}_1^{w}, \boldsymbol{g}_2^{w}) : w \in \mathbb{Z}_q\}$$

KeyGen:

$$\underbrace{x_1, x_2}_{\mathcal{V}} \xleftarrow{\$} \mathbb{Z}_q, \qquad \underbrace{\boldsymbol{c}}_{\mathcal{P}} \leftarrow \boldsymbol{g}_1^{x_1} \boldsymbol{g}_2^{x_2}$$

Verification Function:

$$\mathcal{V}(\boldsymbol{a}_1, \boldsymbol{a}_2) = \boldsymbol{a}_1^{x_1} \boldsymbol{a}_2^{x_2}$$

Example: Equality of DL

G — a group of large prime order q

$\boldsymbol{g}_1, \boldsymbol{g}_2$ — generators for G

$U = G \times G,\ L = \{(\boldsymbol{g}_1^w, \boldsymbol{g}_2^w) : w \in \mathbb{Z}_q\}$

KeyGen:

$$\underbrace{x_1, x_2 \overset{\textcent}{\leftarrow} \mathbb{Z}_q,}_{\mathcal{V}} \qquad \underbrace{\boldsymbol{c} \leftarrow \boldsymbol{g}_1^{x_1} \boldsymbol{g}_2^{x_2}}_{\mathcal{P}}$$

Proof Function:

for $(\boldsymbol{a}_1, \boldsymbol{a}_2) \in L$ with witness w,

$\mathcal{P}(\boldsymbol{a}_1, \boldsymbol{a}_2; w) = \boldsymbol{c}^w$

Verification Function:

$\mathcal{V}(\boldsymbol{a}_1, \boldsymbol{a}_2) = \boldsymbol{a}_1^{x_1} \boldsymbol{a}_2^{x_2}$

Example: Equality of DL

G — a group of large prime order q

$\boldsymbol{g}_1, \boldsymbol{g}_2$ — generators for G

$$U = G \times G,\ L = \{(\boldsymbol{g}_1^{w}, \boldsymbol{g}_2^{w}) : w \in \mathbb{Z}_q\}$$

KeyGen:

$$\begin{array}{ll} x_1, x_2 \xleftarrow{\text{¢}} \mathbb{Z}_q, & \boldsymbol{c} \leftarrow \boldsymbol{g}_1^{x_1}\, \boldsymbol{g}_2^{x_2} \\ \underbrace{y_1, y_2}_{\mathcal{V}} \xleftarrow{\text{¢}} \mathbb{Z}_q, & \underbrace{\boldsymbol{d}}_{\mathcal{P}} \leftarrow \boldsymbol{g}_1^{y_1}\, \boldsymbol{g}_2^{y_2} \end{array}$$

Simulation Soundness

Proof Function:

for $(\boldsymbol{a}_1, \boldsymbol{a}_2) \in L$ with witness w,

$$\mathcal{P}(\boldsymbol{a}_1, \boldsymbol{a}_2; w) = \boldsymbol{c}^{w} \cdot \boldsymbol{d}^{w\mathcal{H}(\boldsymbol{a}_1, \boldsymbol{a}_2)}$$

Verification Function:

$$\mathcal{V}(\boldsymbol{a}_1, \boldsymbol{a}_2) = \boldsymbol{a}_1^{x_1}\, \boldsymbol{a}_2^{x_2} \cdot (\boldsymbol{a}_1^{y_1}\, \boldsymbol{a}_2^{y_2})^{\mathcal{H}(\boldsymbol{a}_1, \boldsymbol{a}_2)}$$

From Hash Proofs to CCA Security

Efficient Semantically Secure PKE

Efficient Hash Proof for Plaintext Equality

\+ Two Key Construction

Efficient and CCA Secure Encryption

Example: Equal ElGamal Plaintexts

Simulation Soundness

$$C_1 = (\boldsymbol{a}_1, \boldsymbol{e}_1) = (\boldsymbol{g}^{w_1}, \boldsymbol{h}_1^{w_1} \boldsymbol{M})$$

$$C_2 = (\boldsymbol{a}_2, \boldsymbol{e}_2) = (\boldsymbol{g}^{w_2}, \boldsymbol{h}_2^{w_2} \boldsymbol{M})$$

$$\underbrace{x_1, x_2, x_3, y_1, y_2, y_3}_{\mathcal{V}} \xleftarrow{\text{¢}} \mathbb{Z}_q$$

$$\underbrace{\begin{array}{ll} \boldsymbol{c}_1 \leftarrow \boldsymbol{g}^{x_1} \boldsymbol{h}_1^{x_3} & \boldsymbol{d}_1 \leftarrow \boldsymbol{g}^{y_1} \boldsymbol{h}_1^{y_3} \\ \boldsymbol{c}_2 \leftarrow \boldsymbol{g}^{x_2} \boldsymbol{h}_2^{-x_3} & \boldsymbol{d}_2 \leftarrow \boldsymbol{g}^{y_2} \boldsymbol{h}_2^{-y_3} \end{array}}_{\mathcal{P}}$$

$$\mathcal{V}(C_1, C_2) = \boldsymbol{a}_1^{x_1} \boldsymbol{a}_2^{x_2} \boldsymbol{f}^{x_3} \cdot (\boldsymbol{a}_1^{y_1} \boldsymbol{a}_2^{y_2} \boldsymbol{f}^{y_3})^{\alpha}$$

$$\mathcal{P}(C_1, C_2; w_1, w_2) = \boldsymbol{c}_1^{w_1} \boldsymbol{c}_2^{w_2} \cdot (\boldsymbol{d}_1^{w_1} \boldsymbol{d}_2^{w_2})^{\alpha}$$

where $\boldsymbol{f} = \boldsymbol{e}_1 / \boldsymbol{e}_2$, $\alpha = \mathcal{H}(C_1, C_2)$

Improvements and Extensions

- More efficient schemes [CS98,S00,KD04]
- Extensions [CS02]:
  - Quadratic Residuosity
  - Paillier’s Decisional Composite Residuosity

Standards: ISO 18033-2

- RSA-OAEP
- Hybrid Encryption Schemes:
  - RSA based
  - ElGamal based

A Hybrid Encryption Paradigm

Secure Key Encapsulation

\+ Secure Data Encapsulation

Secure Hybrid Encryption

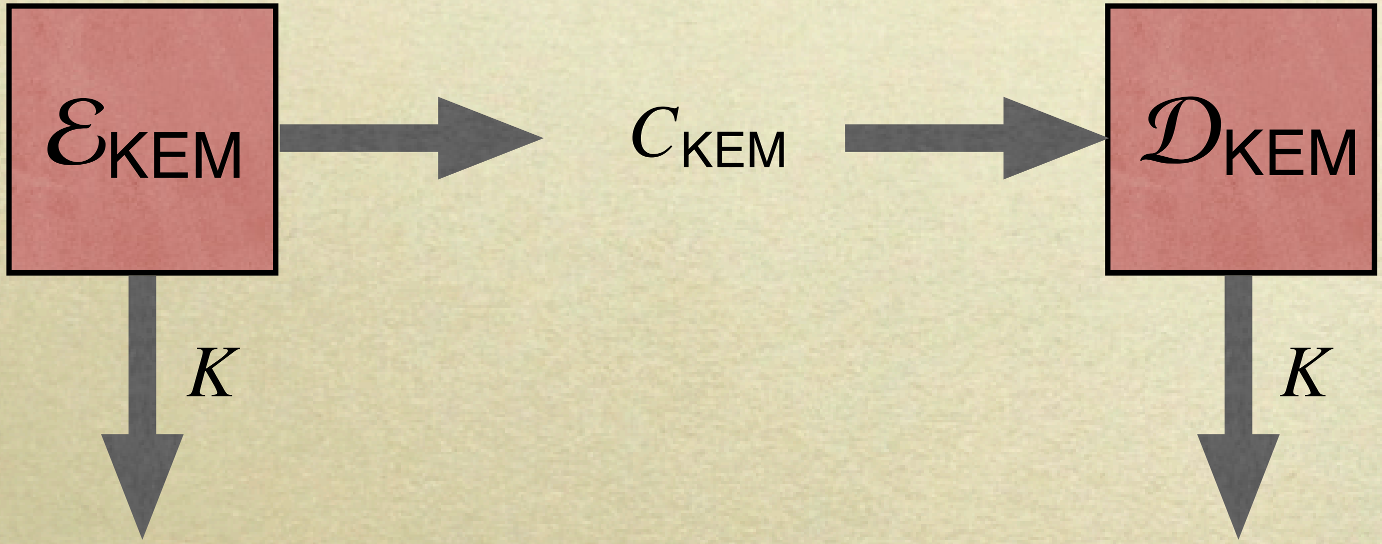
A Hybrid Encryption Paradigm

Secure Key Encapsulation

\+ Secure Data Encapsulation

Secure Hybrid Encryption

A Hybrid Encryption Paradigm



$\mathcal{E}_{\mathrm{KEM}}$

C_{KEM}

$\mathcal{D}_{\mathrm{KEM}}$

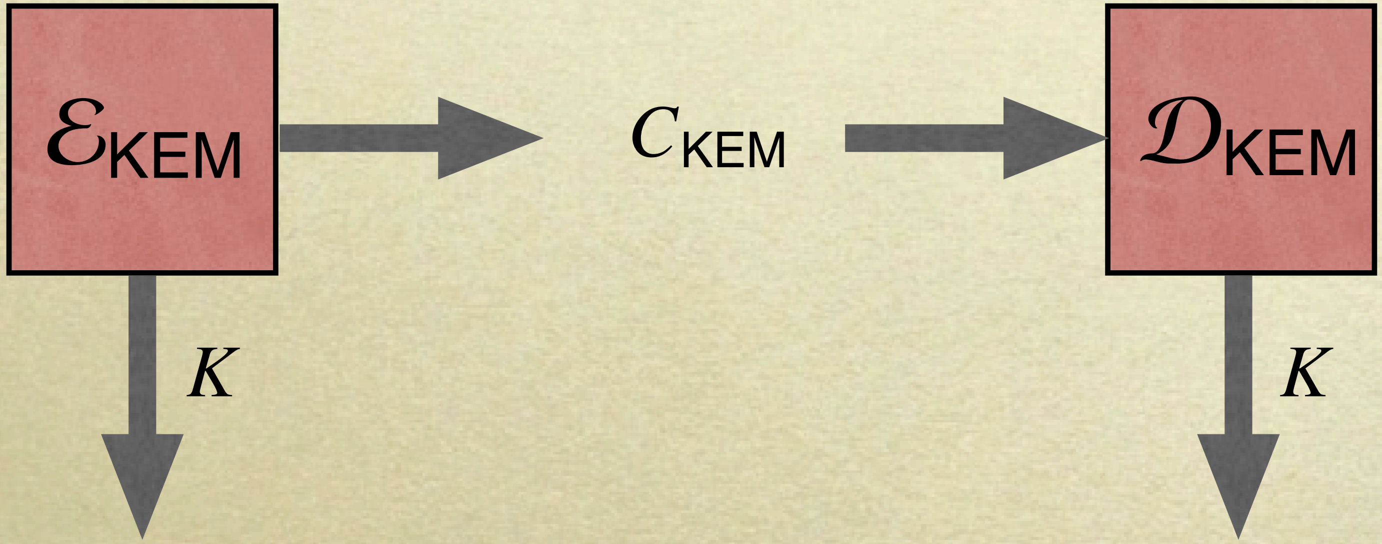
K

K

Key Encapsulation Mechanism (KEM)

Security: *K* looks random after a CCA

A Hybrid Encryption Paradigm



$\mathcal{E}_{\text{KEM}}$ C_{KEM} $\mathcal{D}_{\text{KEM}}$

K K

Key Encapsulation Mechanism (KEM)

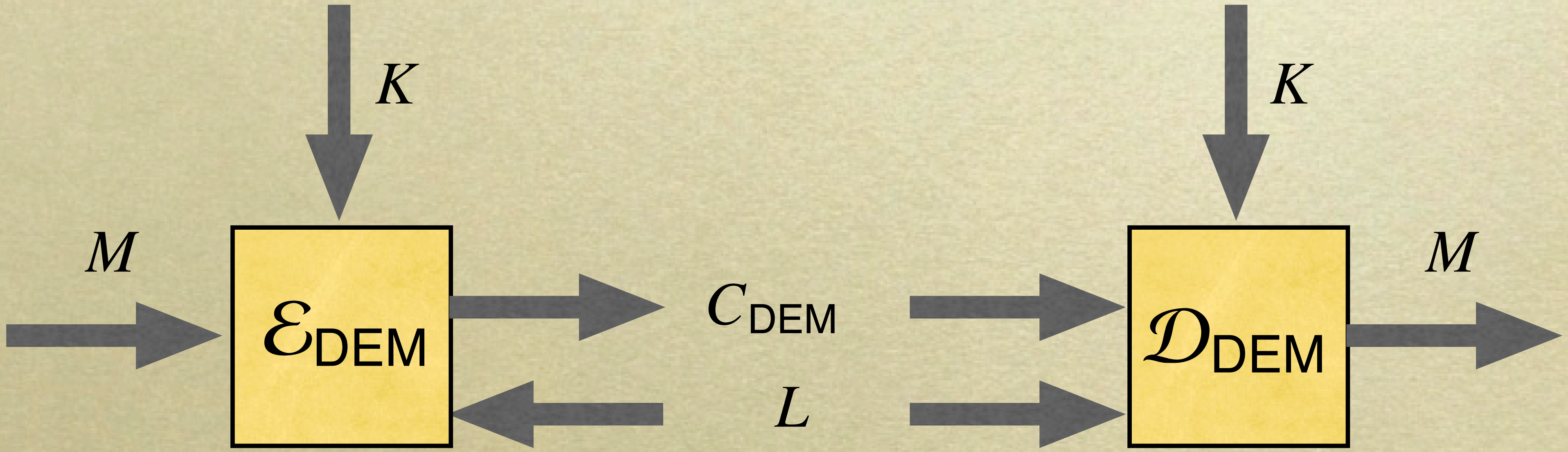
Security: *K* looks random after a CCA

A Hybrid Encryption Paradigm

Data Encapsulation Mechanism (DEM)

Security: M is hidden after a CCA

Implementation: Encrypt then MAC, OCB Mode

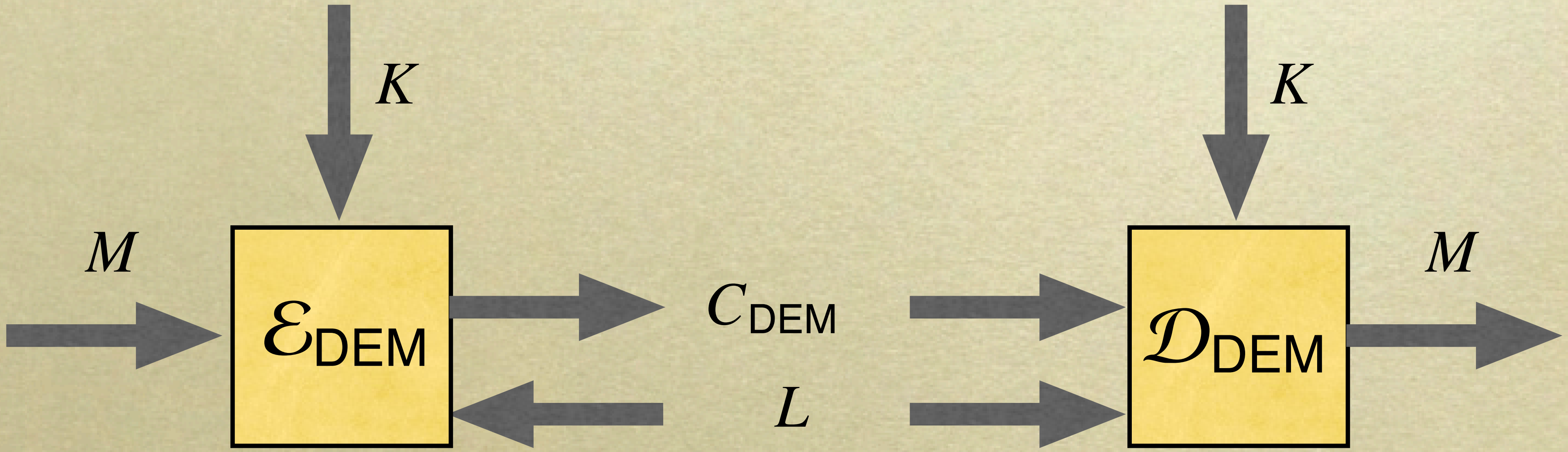


A Hybrid Encryption Paradigm

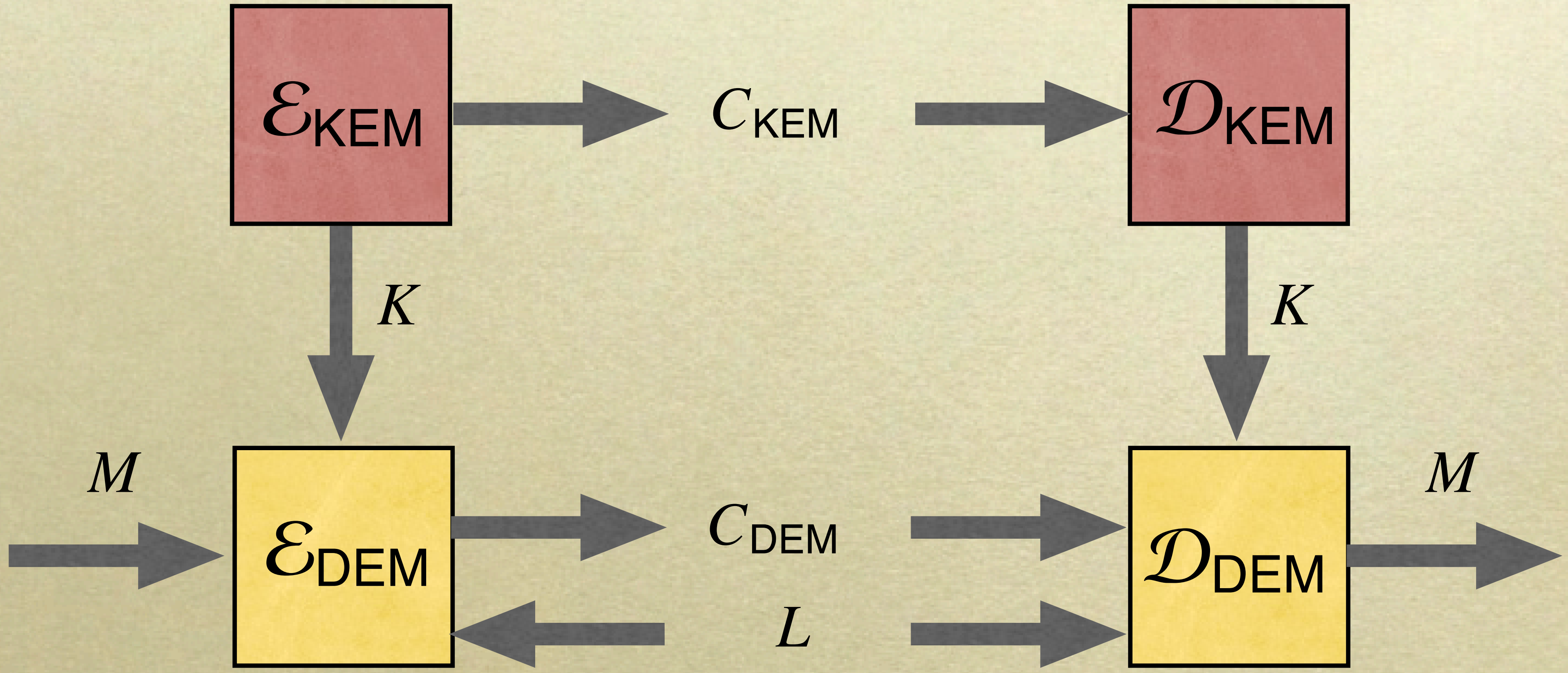
Data Encapsulation Mechanism (DEM)

Security: M is hidden after a CCA

Implementation: Encrypt then MAC, OCB Mode



A Hybrid Encryption Paradigm



Secure KEM + Secure DEM = Secure Hybrid

RSA KEM

n — RSA modulus

e — encryption exponent

d — decryption exponent

Encrypt:

$$w \xleftarrow{\text{¢}} \mathbb{Z}_n$$
$$a \leftarrow w^e$$
$$K \leftarrow \mathsf{KDF}(w)$$
$$C \leftarrow a$$

Decrypt $C = a$:

$$w \leftarrow a^d$$
$$K \leftarrow \mathsf{KDF}(w)$$

secure in ROM under RSA

RSA KEM

n — RSA modulus

e — encryption exponent

d — decryption exponent

Encrypt:

$w \xleftarrow{\cent} \mathbb{Z}_n$

$a \leftarrow w^e$

$K \leftarrow \mathsf{KDF}(w)$

$C \leftarrow a$

Decrypt $C = a$:

$w \leftarrow a^d$

$K \leftarrow \mathsf{KDF}(w)$

ElGamal KEM

G — a group of large prime order q

$\boldsymbol{g}$ — generator for G

$\mathcal{D} \left\{ \; z \xleftarrow{\cent} \mathbb{Z}_q, \right.$

$\mathcal{E} \left\{ \; \boldsymbol{h} \leftarrow \boldsymbol{g}^z \right.$

Encrypt:

$w \xleftarrow{\cent} \mathbb{Z}_q, \boldsymbol{a} \leftarrow \boldsymbol{g}^w$
$K \leftarrow \mathsf{KDF}(\boldsymbol{h}^w)$
$C \leftarrow a$

Decrypt $C = \boldsymbol{a}$:

[check $1 = \boldsymbol{a}^q$]
$K \leftarrow \mathsf{KDF}(\boldsymbol{a}^z)$

secure in ROM under GapDH

ElGamal KEM

G — a group of large prime order q

$\boldsymbol{g}$ — generator for G

$\mathcal{D} \left\{ \; z \overset{\text{¢}}{\leftarrow} \mathbb{Z}_q, \right.$

$\mathcal{E} \left\{ \; \boldsymbol{h} \leftarrow \boldsymbol{g}^z \right.$

Encrypt:

$w \overset{\text{¢}}{\leftarrow} \mathbb{Z}_q, \boldsymbol{a} \leftarrow \boldsymbol{g}^w$

$K \leftarrow \mathsf{KDF}(\boldsymbol{h}^w)$

$C \leftarrow a$

Decrypt $C = \boldsymbol{a}$:

[check $1 = \boldsymbol{a}^q$]

$K \leftarrow \mathsf{KDF}(\boldsymbol{a}^z)$

Hash Proof KEM

G — a group of large prime order q

$\boldsymbol{g}_1$ — generator for G

$$\mathcal{D}\ \left\{\ t, x, y, z \overset{¢}{\leftarrow} \mathbb{Z}_q, \right.$$

$$\mathcal{E}\ \left\{\ \boldsymbol{g}_2 \leftarrow \boldsymbol{g}_1^t, \boldsymbol{c} \leftarrow \boldsymbol{g}_1^x,\ \boldsymbol{d} \leftarrow \boldsymbol{g}_1^y,\ \boldsymbol{h} \leftarrow \boldsymbol{g}_1^z \right.$$

Encrypt:

$w \overset{¢}{\leftarrow} \mathbb{Z}_q,\ \boldsymbol{a}_1 \leftarrow \boldsymbol{g}_1^w,\ \boldsymbol{a}_2 \leftarrow \boldsymbol{g}_2^w$

$\boldsymbol{v} \leftarrow \boldsymbol{c}^w \boldsymbol{d}^{w\mathcal{H}(\boldsymbol{a}_1,\boldsymbol{a}_2)}$

$K \leftarrow \mathsf{KDF}(\boldsymbol{h}^w)$

$C \leftarrow (\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{v})$

Decrypt $(\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{v})$:

$$\text{check} \begin{cases} [\ 1 = \boldsymbol{a}_1^q\ ] \\ \boldsymbol{a}_2 = \boldsymbol{a}_1^t \\ \boldsymbol{v} = \boldsymbol{a}_1^{x+y\mathcal{H}(\boldsymbol{a}_1,\boldsymbol{a}_2)} \end{cases}$$

$K \leftarrow \mathsf{KDF}(\boldsymbol{a}_1^z)$

secure under DDH

Hash Proof KEM

G — a group of large prime order q

$\boldsymbol{g}_1$ — generator for G

$$\mathcal{D} \left\{ t, x, y, z \xleftarrow{\$} \mathbb{Z}_q, \right.$$

$$\mathcal{E} \left\{ \boldsymbol{g}_2 \leftarrow \boldsymbol{g}_1^t, \boldsymbol{c} \leftarrow \boldsymbol{g}_1^x, \ \boldsymbol{d} \leftarrow \boldsymbol{g}_1^y, \ \boldsymbol{h} \leftarrow \boldsymbol{g}_1^z \right.$$

Encrypt:

$w \xleftarrow{\$} \mathbb{Z}_q, \ \boldsymbol{a}_1 \leftarrow \boldsymbol{g}_1^w, \ \boldsymbol{a}_2 \leftarrow \boldsymbol{g}_2^w$

$\boldsymbol{v} \leftarrow \boldsymbol{c}^w \boldsymbol{d}^{w\mathcal{H}(\boldsymbol{a}_1,\boldsymbol{a}_2)}$

$K \leftarrow \mathsf{KDF}(\boldsymbol{h}^w)$

$C \leftarrow (\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{v})$

Decrypt $(\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{v})$:

$$\text{check} \left\{ \begin{array}{c} [\, 1 = \boldsymbol{a}_1^q \,] \\ \boldsymbol{a}_2 = \boldsymbol{a}_1^t \\ \boldsymbol{v} = \boldsymbol{a}_1^{x+y\mathcal{H}(\boldsymbol{a}_1,\boldsymbol{a}_2)} \end{array} \right.$$

$K \leftarrow \mathsf{KDF}(\boldsymbol{a}_1^z)$

Has

secure under DDH

no less secure than ElGamal KEM

G — a group of large prime order q

$\boldsymbol{g}_1$ — generator for G

$$\mathcal{D}\ \Big\{\ t, x, y, z \xleftarrow{\text{¢}} \mathbb{Z}_q,$$

$$\mathcal{E}\ \Big\{\ \boldsymbol{g}_2 \leftarrow \boldsymbol{g}_1^t,\ \boldsymbol{c} \leftarrow \boldsymbol{g}_1^x,\ \boldsymbol{d} \leftarrow \boldsymbol{g}_1^y,\ \boldsymbol{h} \leftarrow \boldsymbol{g}_1^z$$

Encrypt:

$w \xleftarrow{\text{¢}} \mathbb{Z}_q,\ \boldsymbol{a}_1 \leftarrow \boldsymbol{g}_1^w,\ \boldsymbol{a}_2 \leftarrow \boldsymbol{g}_2^w$

$\boldsymbol{v} \leftarrow \boldsymbol{c}^w \boldsymbol{d}^{w\mathcal{H}(\boldsymbol{a}_1,\boldsymbol{a}_2)}$

$K \leftarrow \mathsf{KDF}(\boldsymbol{h}^w)$

$C \leftarrow (\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{v})$

Decrypt $(\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{v})$:

$$\text{check} \begin{cases} [\ 1 = \boldsymbol{a}_1^q\ ] \\ \boldsymbol{a}_2 = \boldsymbol{a}_1^t \\ \boldsymbol{v} = \boldsymbol{a}_1^{x+y\mathcal{H}(\boldsymbol{a}_1,\boldsymbol{a}_2)} \end{cases}$$

$K \leftarrow \mathsf{KDF}(\boldsymbol{a}_1^z)$

Has

secure under DDH

no less secure than ElGamal KEM

secure in ROM under CDH

G — a group of large prime order q

$\boldsymbol{g}_1$ — generator for G

$$\mathcal{D}\left\{\; t, x, y, z \overset{¢}{\leftarrow} \mathbb{Z}_q, \right.$$

$$\mathcal{E}\left\{\; \boldsymbol{g}_2 \leftarrow \boldsymbol{g}_1^t, \boldsymbol{c} \leftarrow \boldsymbol{g}_1^x,\ \boldsymbol{d} \leftarrow \boldsymbol{g}_1^y,\ \boldsymbol{h} \leftarrow \boldsymbol{g}_1^z \right.$$

Encrypt:

$w \overset{¢}{\leftarrow} \mathbb{Z}_q,\ \boldsymbol{a}_1 \leftarrow \boldsymbol{g}_1^w,\ \boldsymbol{a}_2 \leftarrow \boldsymbol{g}_2^w$

$\boldsymbol{v} \leftarrow \boldsymbol{c}^w \boldsymbol{d}^{w\mathcal{H}(\boldsymbol{a}_1,\boldsymbol{a}_2)}$

$K \leftarrow \mathsf{KDF}(\boldsymbol{h}^w)$

$C \leftarrow (\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{v})$

Decrypt $(\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{v})$:

$$\text{check} \begin{cases} [\ 1 = \boldsymbol{a}_1^q\ ] \\ \boldsymbol{a}_2 = \boldsymbol{a}_1^t \\ \boldsymbol{v} = \boldsymbol{a}_1^{x+y\mathcal{H}(\boldsymbol{a}_1,\boldsymbol{a}_2)} \end{cases}$$

$K \leftarrow \mathsf{KDF}(\boldsymbol{a}_1^z)$

Has

secure under DDH

no less secure than ElGamal KEM

secure in ROM under CDH

exponentiation with pre-processing

G — a group of

$\boldsymbol{g}_1$ — generator for G

$$\mathcal{D}\left\{\ t, x, y, z \xleftarrow{\text{¢}} \mathbb{Z}_q,\right.$$

$$\mathcal{E}\left\{\ \boldsymbol{g}_2 \leftarrow \boldsymbol{g}_1^t, \boldsymbol{c} \leftarrow \boldsymbol{g}_1^x,\ \boldsymbol{d} \leftarrow \boldsymbol{g}_1^y,\ \boldsymbol{h} \leftarrow \boldsymbol{g}_1^z\right.$$

Encrypt:

$w \xleftarrow{\text{¢}} \mathbb{Z}_q,\ \boldsymbol{a}_1 \leftarrow \boldsymbol{g}_1^w,\ \boldsymbol{a}_2 \leftarrow \boldsymbol{g}_2^w$

$\boldsymbol{v} \leftarrow \boldsymbol{c}^w \boldsymbol{d}^{w\mathcal{H}(\boldsymbol{a}_1,\boldsymbol{a}_2)}$

$K \leftarrow \mathsf{KDF}(\boldsymbol{h}^w)$

$C \leftarrow (\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{v})$

Decrypt $(\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{v})$:

$$\text{check}\left\{\begin{array}{r}[\ 1 = \boldsymbol{a}_1^q\ ]\\ \boldsymbol{a}_2 = \boldsymbol{a}_1^t\\ \boldsymbol{v} = \boldsymbol{a}_1^{x+y\mathcal{H}(\boldsymbol{a}_1,\boldsymbol{a}_2)}\end{array}\right.$$

$K \leftarrow \mathsf{KDF}(\boldsymbol{a}_1^z)$

Has

secure under DDH

no less secure than ElGamal KEM

secure in ROM under CDH

exponentiation with pre-processing

confounds patent lawyers

G — a group of

$\boldsymbol{g}_1$ — generator

$$\mathcal{D}\left\{\ t, x, y, z \xleftarrow{\$} \mathbb{Z}_q,\right.$$

$$\mathcal{E}\left\{\ \boldsymbol{g}_2 \leftarrow \boldsymbol{g}_1^t,\ \boldsymbol{c} \leftarrow \boldsymbol{g}_1^x,\ \boldsymbol{d} \leftarrow \boldsymbol{g}_1^y,\ \boldsymbol{h} \leftarrow \boldsymbol{g}_1^z\right.$$

Encrypt:

$w \xleftarrow{\$} \mathbb{Z}_q,\ \boldsymbol{a}_1 \leftarrow \boldsymbol{g}_1^w,\ \boldsymbol{a}_2 \leftarrow \boldsymbol{g}_2^w$

$\boldsymbol{v} \leftarrow \boldsymbol{c}^w \boldsymbol{d}^{w\mathcal{H}(\boldsymbol{a}_1,\boldsymbol{a}_2)}$

$K \leftarrow \mathsf{KDF}(\boldsymbol{h}^w)$

$C \leftarrow (\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{v})$

Decrypt $(\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{v})$:

$$\text{check}\left\{\begin{array}{l} [\ 1 = \boldsymbol{a}_1^q\ ] \\ \boldsymbol{a}_2 = \boldsymbol{a}_1^t \\ \boldsymbol{v} = \boldsymbol{a}_1^{x+y\mathcal{H}(\boldsymbol{a}_1,\boldsymbol{a}_2)} \end{array}\right.$$

$K \leftarrow \mathsf{KDF}(\boldsymbol{a}_1^z)$

And now for something completely different... [CHK04, BB04]

CCA Secure Encryption using
elliptic curves, equipped with
bilinear maps

Different assumptions and techniques

Hashed decisional (or even computational) assumption

Distribute decryption service, without interaction
(Improves [CG99])

Conclusions

- Adaptive CCA security is now widely accepted as the "right" definition
- Demanded by standards bodies
- Science fiction is becoming reality
- Progress continues!

Conclusions

- A
- Scienc y
- Progress continues!

Thanks for Listening!