Public-key encryption, IND-CPA/CCA, ElGamal, RSA

Basic goals of cryptography

	Message privacy	**Message integrity / authentication**
Symmetric keys	Symmetric encryption	Message authentication codes (MAC)
Asymmetric keys	Asymmetric encryption (a.k.a. public-key encryption)	Digital signatures

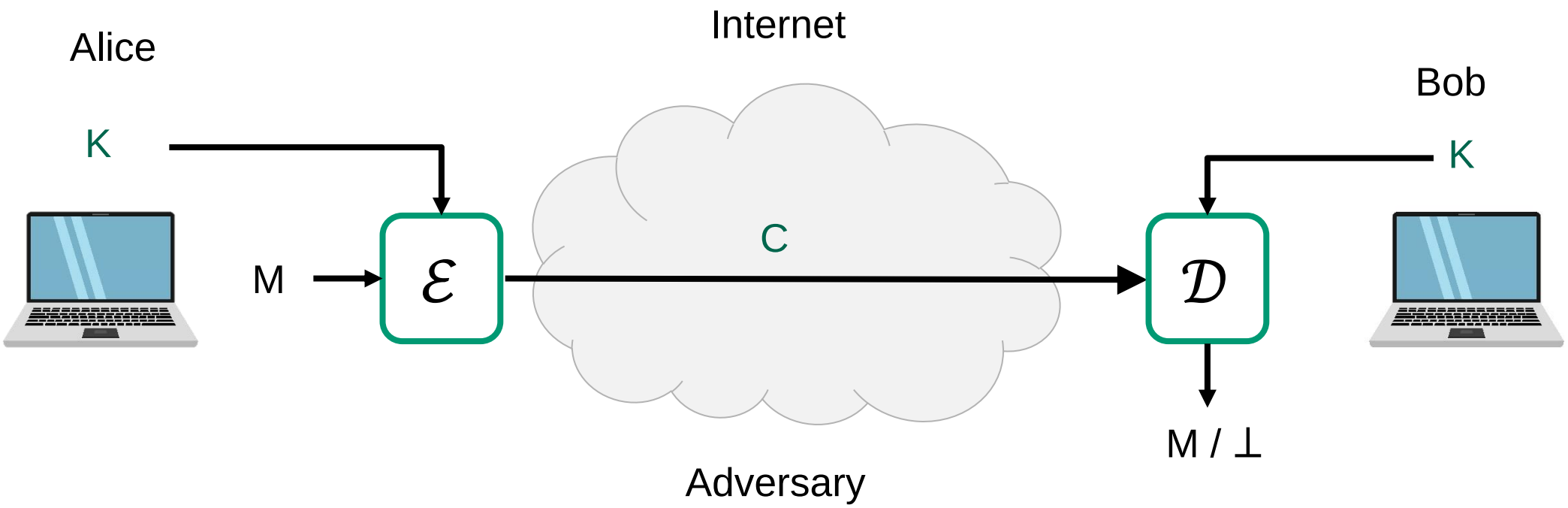
(Key exchange)

Basic goals of cryptography

	Message privacy	**Message integrity / authentication**
Symmetric keys	Symmetric encryption	Message authentication codes (MAC)
Asymmetric keys	Asymmetric encryption (a.k.a. public-key encryption)	Digital signatures

(Key exchange)

Creating secure channels: encryption schemes

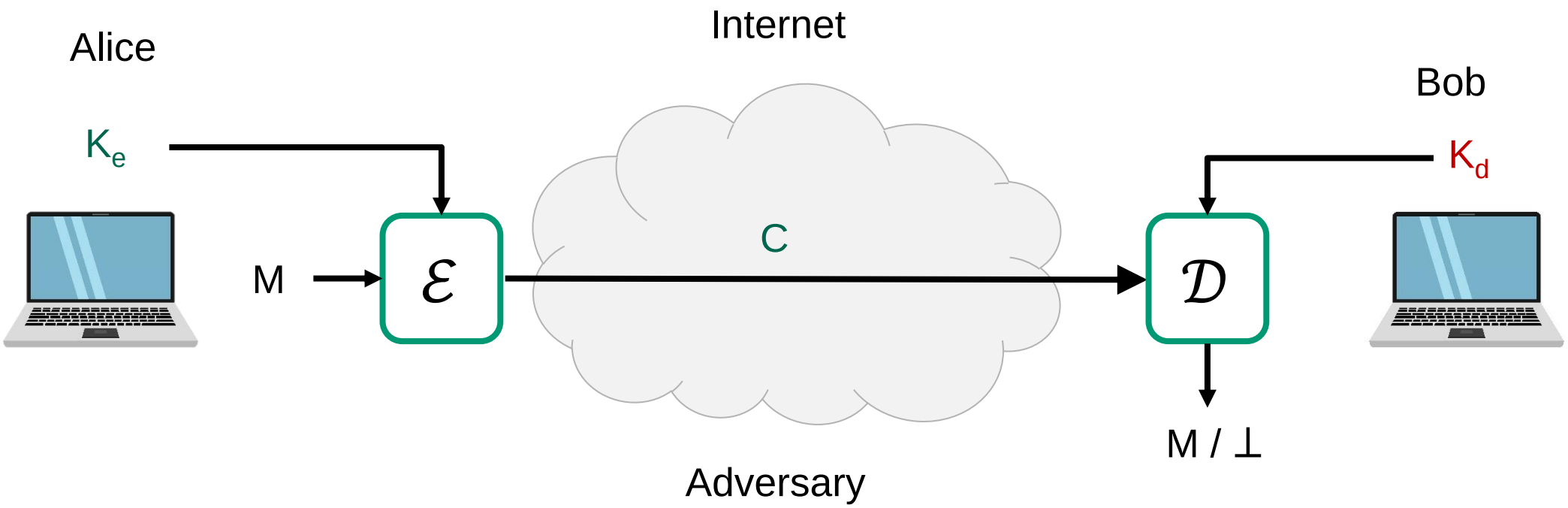


$\mathcal{E}$: encryption algorithm (public)

K: encryption /decryption key (secret)

$\mathcal{D}$: decryption algorithm (public)

Creating secure channels: encryption schemes



$\mathcal{E}$: encryption algorithm (public)

K_e : encryption key (public)

$\mathcal{D}$: decryption algorithm (public)

K_d : decryption key (private)

Public-key encryption

Public-key encryption – syntax

A **public-key encryption scheme** is a tuple $\Sigma = (\mathrm{KeyGen}, \mathrm{Enc}, \mathrm{Dec})$ of algorithms

$(sk, pk) \xleftarrow{\$} \mathrm{KeyGen}$

$\mathrm{Enc} : \mathcal{PK} \times \mathcal{M} \to \mathcal{C}$

$\mathrm{Enc}(pk, M) = \mathrm{Enc}_{pk}(M) = C$

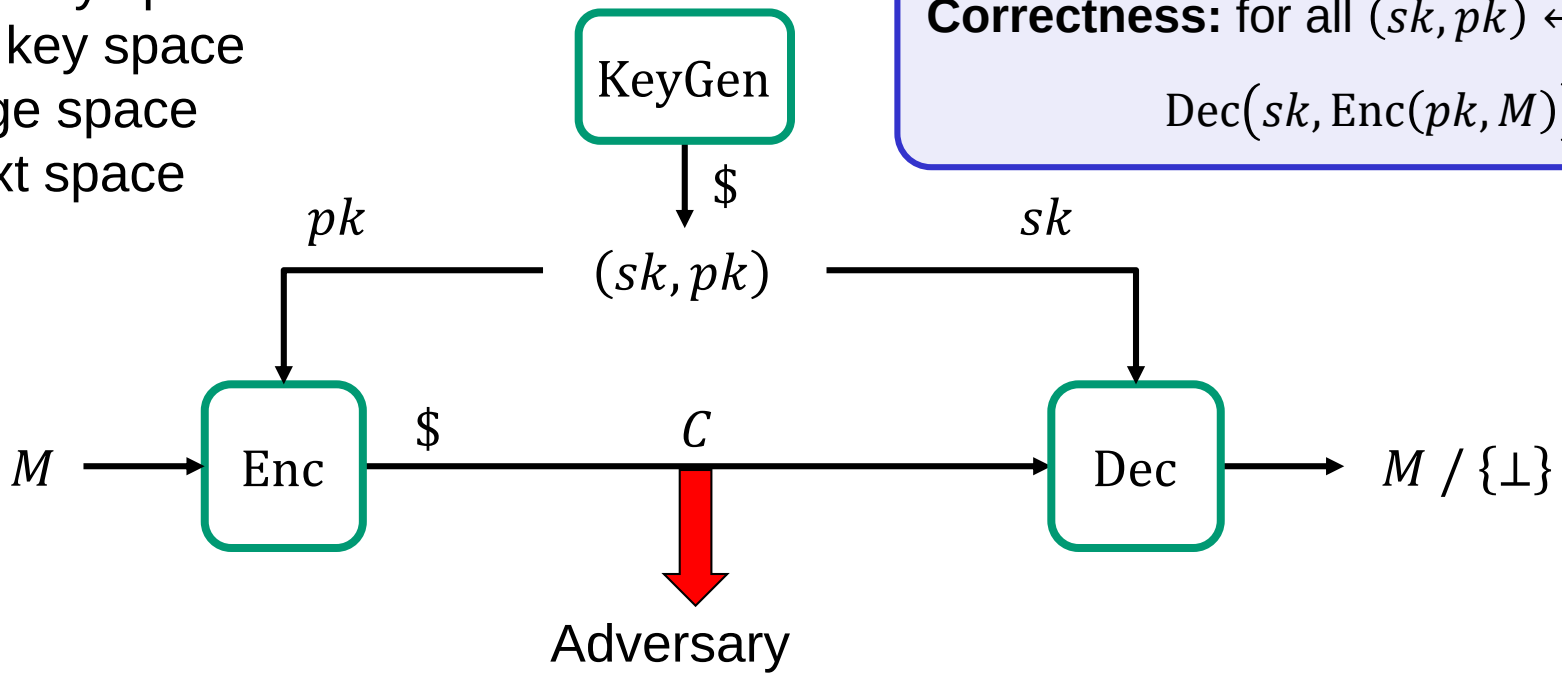
$\mathrm{Dec} : \mathcal{SK} \times \mathcal{C} \to \mathcal{M} \cup \{\perp\}$

$\mathrm{Dec}(sk, C) = \mathrm{Dec}_{SK}(C) = M/\perp$

- $\mathcal{SK}$ – private key space
- $\mathcal{PK}$ – public key space
- $\mathcal{M}$ – message space
- $\mathcal{C}$ – ciphertext space

Correctness: for all $(sk, pk) \leftarrow \mathrm{KeyGen}$:

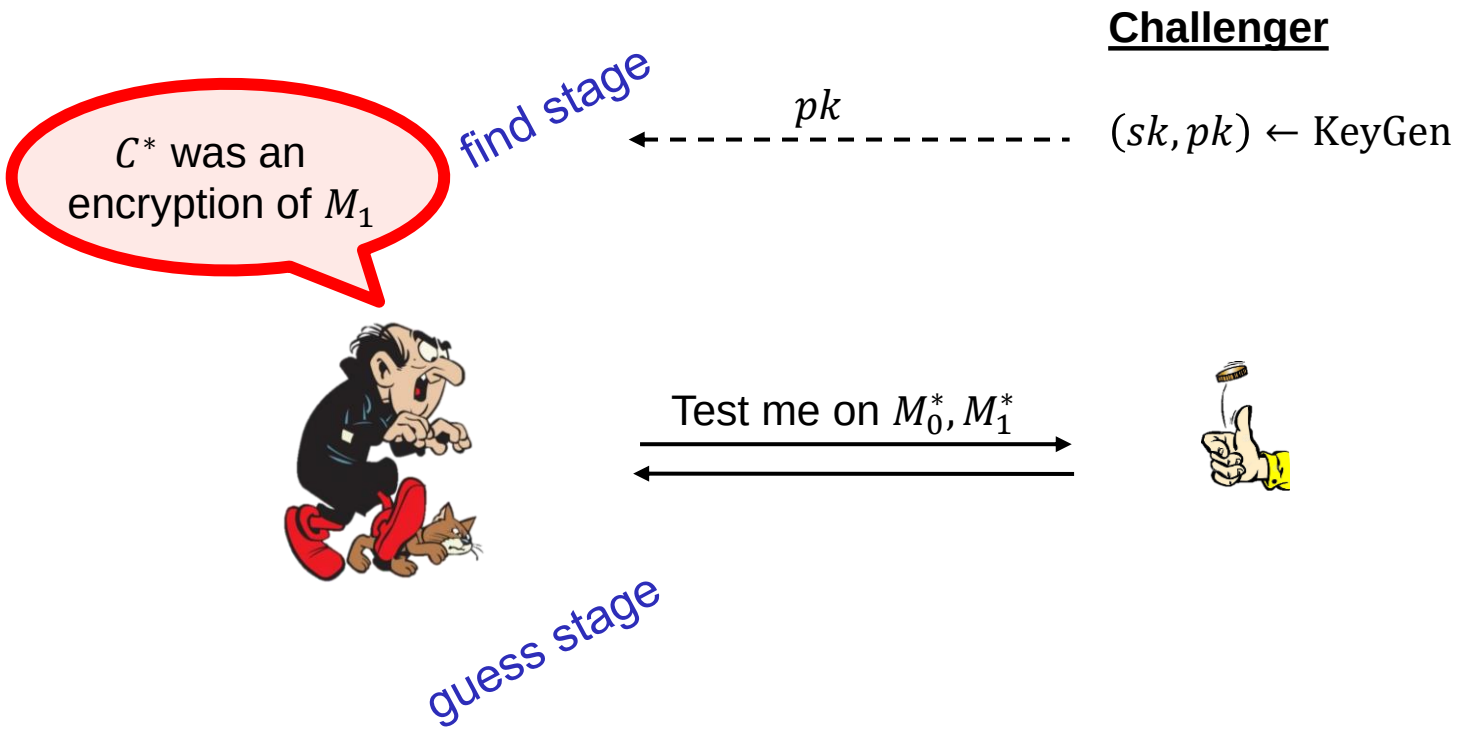
$$\mathrm{Dec}\big(sk, \mathrm{Enc}(pk, M)\big) = M$$



Public-key encryption – security: IND-CPA

$\mathbf{Exp}_{\Sigma}^{\text{ind-cpa}}(A)$

1. $b \xleftarrow{\$} \{0,1\}$
2. $(sk, pk) \xleftarrow{\$} \Sigma.\text{KeyGen}$
3. $M_0^*, M_1^* \leftarrow A(pk)$ // find stage
4. **if** $|M_0^*| \neq |M_1^*|$ **then**
5. **return** $\perp$
6. $C^* \leftarrow \Sigma.\text{Enc}(pk, M_b^*)$
7. $b' \leftarrow A(pk, C^*)$ // guess stage
8. **return** $b' \stackrel{?}{=} b$



Definition: The **IND-CPA-advantage** of an adversary A is

$$\mathbf{Adv}_{\Sigma}^{\text{ind-cpa}}(A) = \left|2 \cdot \Pr\left[\mathbf{Exp}_{\Sigma}^{\text{ind-cpa}}(A) \Rightarrow \text{true}\right] - 1\right|$$

Public-key encryption – security: IND-CCA

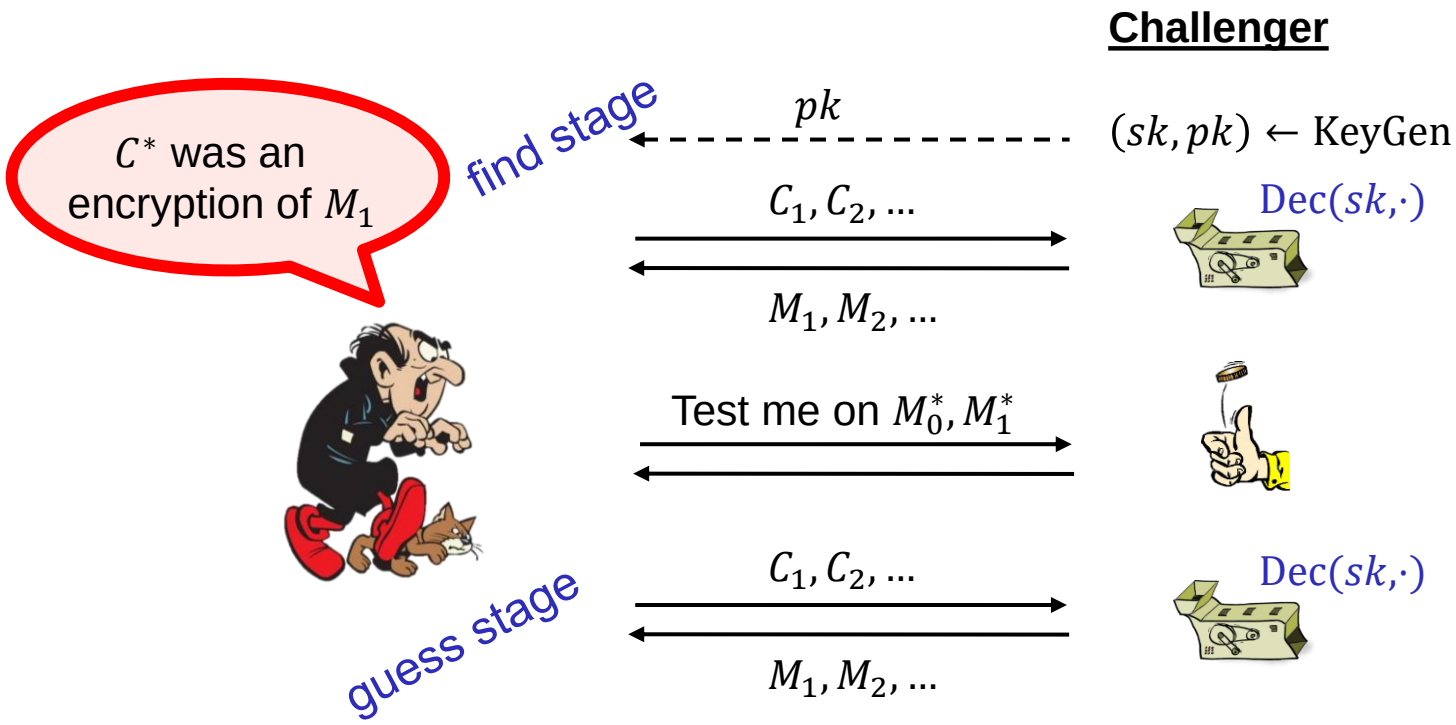
$\mathbf{Exp}_{\Sigma}^{\text{ind-cca}}(A)$

1. $b \xleftarrow{\$} \{0,1\}$
2. $(sk, pk) \xleftarrow{\$} \Sigma.\text{KeyGen}$
3. $M_0^*, M_1^* \leftarrow A^{\mathcal{D}_{sk}(\cdot)}(pk)$ // find stage
4. **if** $|M_0^*| \neq |M_1^*|$ **then**
5. **return** $\perp$
6. $C^* \leftarrow \Sigma.\text{Enc}(pk, M_b^*)$
7. $b' \leftarrow A^{\mathcal{D}_{sk}(\cdot)}(pk, C^*)$ // guess stage
8. **return** $b' \stackrel{?}{=} b$

$\mathcal{D}_{sk}(C)$

---

1. **if** $C = C^*$ **the** // cheating!
2. **return** $\perp$
3. **return** $\Sigma.\text{Dec}(sk, C)$



Definition: The **IND-CCA-advantage** of an adversary A is

$$\mathbf{Adv}_{\Sigma}^{\text{ind-cca}}(A) = \left|2 \cdot \Pr\left[\mathbf{Exp}_{\Sigma}^{\text{ind-cca}}(A) \Rightarrow \text{true}\right] - 1\right|$$

Diffie-Hellman key exchange

- Discovered in the 1970's

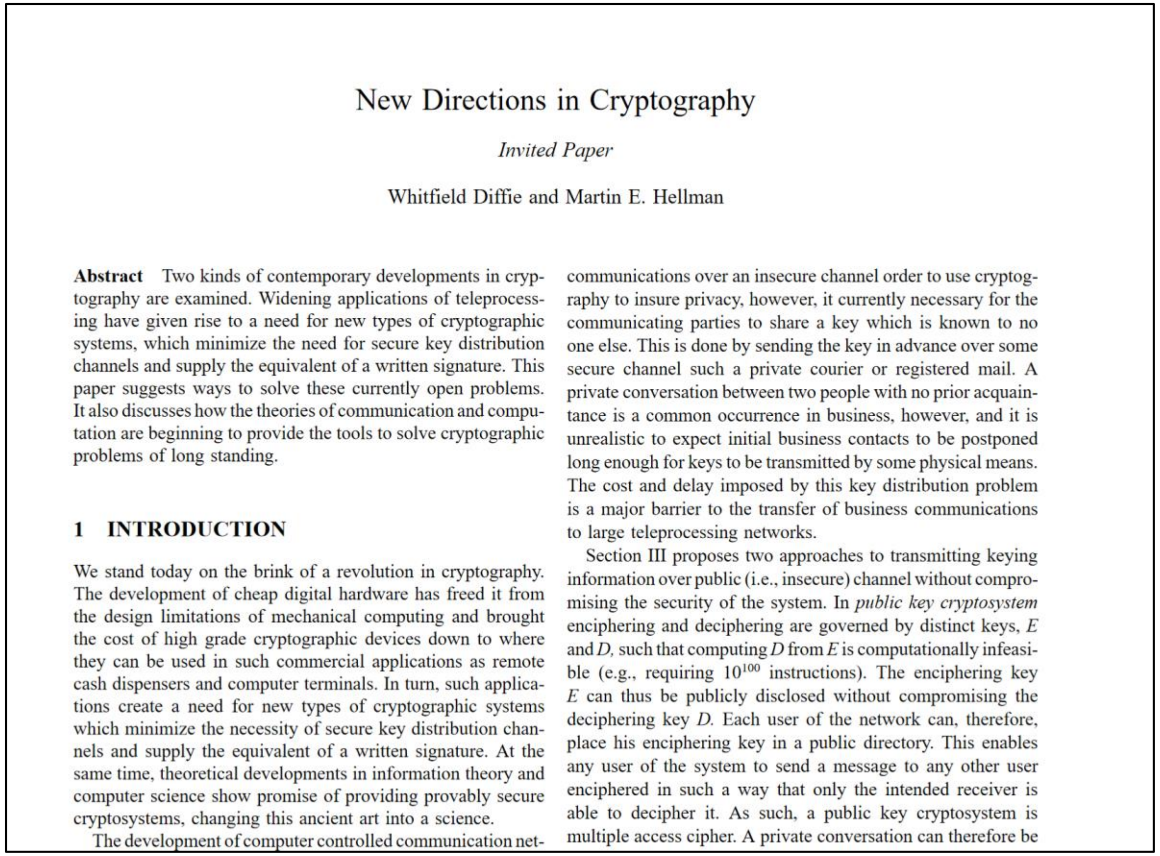
- Allows two parties to establish a shared secret without ever having met

- Diffie & Hellman paper also introduced the idea of:

  - Public-key encryption
    - But didn't figure out how to do it
    - 1978: ElGamal encryption scheme

  - Digital signatures
    - But didn't figure out how to do it

Ralph Merkle Whitfield Diffie
Martin Hellman



New Directions in Cryptography

Invited Paper

Whitfield Diffie and Martin E. Hellman

Abstract Two kinds of contemporary developments in cryptography are examined. Widening applications of teleprocessing have given rise to a need for new types of cryptographic systems, which minimize the need for secure key distribution channels and supply the equivalent of a written signature. This paper suggests ways to solve these currently open problems. It also discusses how the theories of communication and computation are beginning to provide the tools to solve cryptographic problems of long standing.

1 INTRODUCTION

We stand today on the brink of a revolution in cryptography. The development of cheap digital hardware has freed it from the design limitations of mechanical computing and brought the cost of high grade cryptographic devices down to where they can be used in such commercial applications as remote cash dispensers and computer terminals. In turn, such applications create a need for new types of cryptographic systems which minimize the necessity of secure key distribution channels and supply the equivalent of a written signature. At the same time, theoretical developments in information theory and computer science show promise of providing provably secure cryptosystems, changing this ancient art into a science.

The development of computer controlled communication networks communications over an insecure channel order to use cryptography to insure privacy, however, it currently necessary for the communicating parties to share a key which is known to no one else. This is done by sending the key in advance over some secure channel such a private courier or registered mail. A private conversation between two people with no prior acquaintance is a common occurrence in business, however, and it is unrealistic to expect initial business contacts to be postponed long enough for keys to be transmitted by some physical means. The cost and delay imposed by this key distribution problem is a major barrier to the transfer of business communications to large teleprocessing networks.

Section III proposes two approaches to transmitting keying information over public (i.e., insecure) channel without compromising the security of the system. In *public key cryptosystem* enciphering and deciphering are governed by distinct keys, E and D, such that computing D from E is computationally infeasible (e.g., requiring 10^{100} instructions). The enciphering key E can thus be publicly disclosed without compromising the deciphering key D. Each user of the network can, therefore, place his enciphering key in a public directory. This enables any user of the system to send a message to any other user enciphered in such a way that only the intended receiver is able to decipher it. As such, a public key cryptosystem is multiple access cipher. A private conversation can therefore be

ElGamal

$$G = \langle g \rangle$$

A

B

$a \xleftarrow{\$} \{1, \dots, |G|\}$

$A \leftarrow g^a$

$Z \leftarrow B^a = g^{ab}$

$b \xleftarrow{\$} \{1, \dots, |G|\}$

$B \leftarrow g^b$

$Z \leftarrow A^b = g^{ab}$

ElGamal

$G = \langle g \rangle$

B

A

$b \xleftarrow{\$} \{1, \dots, |G|\}$

$B \leftarrow g^b$

$a \xleftarrow{\$} \{1, \dots, |G|\}$

$A \leftarrow g^a$

$Z \leftarrow B^a = g^{ab}$

$Z \leftarrow A^b = g^{ab}$

ElGamal

$G = \langle g \rangle$

B

A, C

$b \xleftarrow{\$} \{1, \dots, |G|\}$

$B \leftarrow g^b$

$a \xleftarrow{\$} \{1, \dots, |G|\}$

$A \leftarrow g^a$

$Z \leftarrow B^a = g^{ab}$

$C \leftarrow \Sigma.\mathrm{Enc}(Z, M)$

$Z \leftarrow A^b = g^{ab}$

$\Sigma = (\mathrm{KeyGen}, \mathrm{Enc}, \mathrm{Dec})$

$\Sigma.\mathrm{Enc} : G \times \mathcal{M} \to \mathcal{C}$

$\Sigma.\mathrm{Dec} : G \times \mathcal{C} \to \mathcal{M}$

ElGamal

$G = \langle g \rangle$

B

A, C

$b \xleftarrow{\$} \{1, \dots, |G|\}$

$B \leftarrow g^b$

$a \xleftarrow{\$} \{1, \dots, |G|\}$

$A \leftarrow g^a$

$Z \leftarrow B^a = g^{ab}$

$C \leftarrow \Sigma.\mathrm{Enc}(Z, M)$

$Z \leftarrow A^b = g^{ab}$

$M \leftarrow \Sigma.\mathrm{Dec}(Z, C)$

$\Sigma = (\mathrm{KeyGen}, \mathrm{Enc}, \mathrm{Dec})$

$\Sigma.\mathrm{Enc} : G \times \mathcal{M} \to \mathcal{C}$

$\Sigma.\mathrm{Dec} : G \times \mathcal{C} \to \mathcal{M}$

ElGamal

$G = \langle g \rangle$

KeyGen

1. $sk = b \xleftarrow{\$} \{1, \dots, |G|\}$
2. $pk = B \leftarrow g^b$
3. **return** (sk, pk)

B

A, C

$a \xleftarrow{\$} \{1, \dots, |G|\}$

$A \leftarrow g^a$

$Z \leftarrow B^a = g^{ab}$

$C \leftarrow \Sigma.\mathrm{Enc}(Z, M)$

$Z \leftarrow A^b = g^{ab}$

$M \leftarrow \Sigma.\mathrm{Dec}(Z, C)$

$\Sigma = (\mathrm{KeyGen}, \mathrm{Enc}, \mathrm{Dec})$

$\Sigma.\mathrm{Enc} : G \times \mathcal{M} \to \mathcal{C}$

$\Sigma.\mathrm{Dec} : G \times \mathcal{C} \to \mathcal{M}$

ElGamal

$G = \langle g \rangle$

KeyGen

1. $sk = b \xleftarrow{\$} \{1, \dots, |G|\}$
2. $pk = B \leftarrow g^b$
3. **return** (sk, pk)

B

A, C

Enc(pk, M)

1. $a \xleftarrow{\$} \{1, \dots, |G|\}$
2. $A \leftarrow g^a$
3. $Z \leftarrow B^a = g^{ab}$
4. $C \leftarrow \Sigma.\mathrm{Enc}(Z, M)$
5. **return** (A, C)

$Z \leftarrow A^b = g^{ab}$

$M \leftarrow \Sigma.\mathrm{Dec}(Z, C)$

$\Sigma = (\mathrm{KeyGen}, \mathrm{Enc}, \mathrm{Dec})$

$\Sigma.\mathrm{Enc} : G \times \mathcal{M} \to \mathcal{C}$

$\Sigma.\mathrm{Dec} : G \times \mathcal{C} \to \mathcal{M}$

ElGamal

$\text{ElGamal.Enc} : G \times \mathcal{M} \rightarrow G \times \mathcal{C}$

$\text{ElGamal.Dec} : \boldsymbol{Z}_p \times G \times \mathcal{C} \rightarrow \mathcal{M}$

$G = \langle g \rangle$

B

A, C

KeyGen

1. $sk = b \xleftarrow{\$} \{1, \dots, |G|\}$
2. $pk = B \leftarrow g^b$
3. **return** (sk, pk)

Enc(pk, M)

1. $a \xleftarrow{\$} \{1, \dots, |G|\}$
2. $A \leftarrow g^a$
3. $Z \leftarrow B^a = g^{ab}$
4. $C \leftarrow \Sigma.\text{Enc}(Z, M)$
5. **return** (A, C)

Dec(sk, C)

1. $Z \leftarrow A^b = g^{ab}$
2. $M \leftarrow \Sigma.\text{Dec}(Z, C)$
3. **return** M

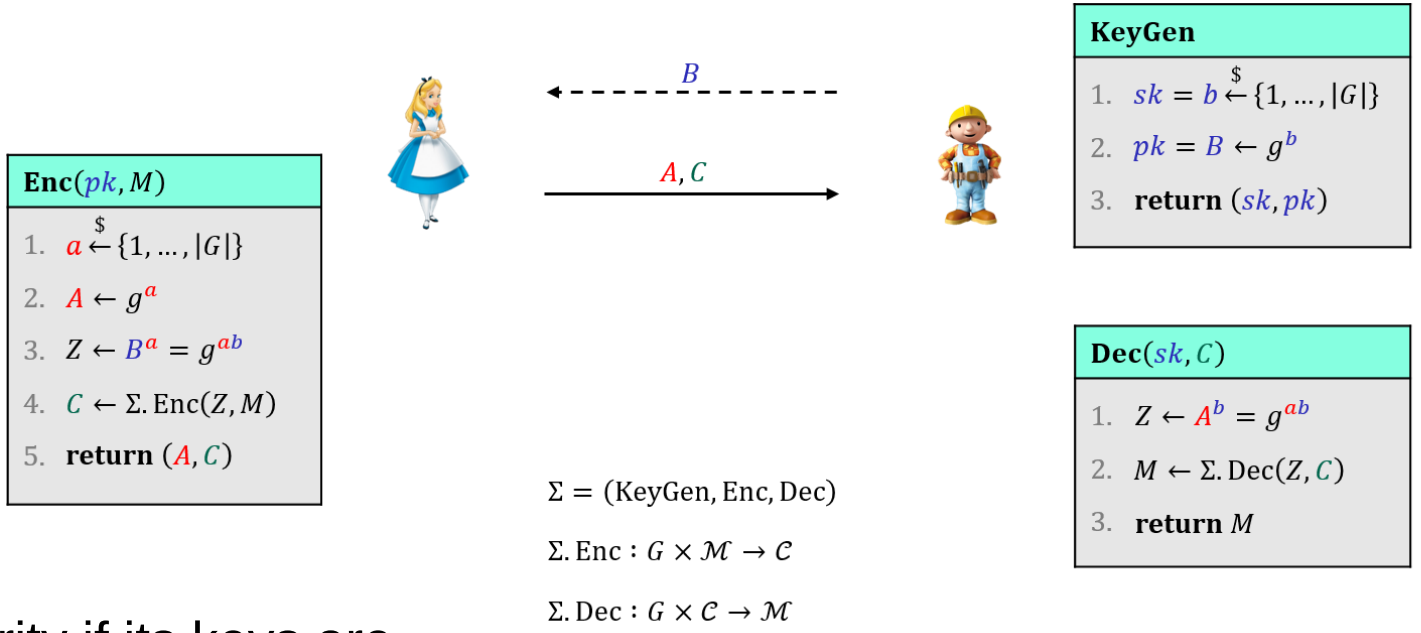
$\Sigma = (\text{KeyGen}, \text{Enc}, \text{Dec})$

$\Sigma.\text{Enc} : G \times \mathcal{M} \rightarrow \mathcal{C}$

$\Sigma.\text{Dec} : G \times \mathcal{C} \rightarrow \mathcal{M}$

ElGamal – IND-CPA security

- Clearly DLOG and DH must be hard
- Clearly Σ must be secure (IND-CPA? IND-CCA?)
- But is this enough?



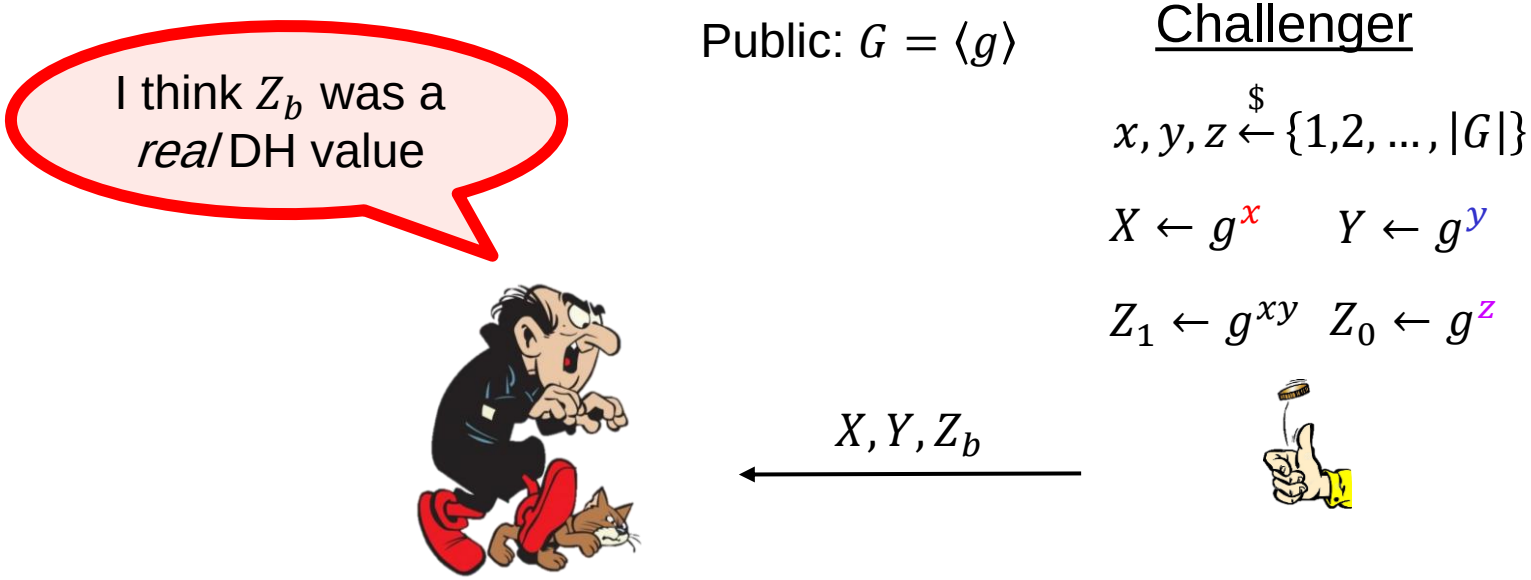
- **No:** Σ only guarantees security if its keys are *independent* and *uniformly* distributed (in the group G)

but g^{ab} *isn't* uniformly distributed in G!

Decisional Diffie-Hellman (DDH) problem

$\mathbf{Exp}^{\mathrm{ddh}}_{G,g}(A)$

1. $b \xleftarrow{\$} \{0,1\}$
2. $x, y, z \xleftarrow{\$} \{1,2,\ldots,|G|\}$
3. $X \leftarrow g^x \quad Y \leftarrow g^y$
4. $Z_1 \leftarrow g^{xy}$
5. $Z_0 \leftarrow g^z$
6. $b' \leftarrow A(X, Y, Z_b)$
7. **return** $b' \stackrel{?}{=} b$



Adversary wins if it can *distinguish* a real DH value g^{xy} from a random group element g^z

Definition: The **DDH-advantage** of an adversary A is

$$\mathbf{Adv}^{\mathrm{ddh}}_{G,g}(A) = \left|2 \cdot \Pr\left[\mathbf{Exp}^{\mathrm{ddh}}_{G,g}(A) \Rightarrow \mathrm{true}\right] - 1\right|$$

DLOG vs. DH vs. DDH

$\mathbf{Exp}^{\mathrm{dlog}}_{G,g}(A)$

1. $x \xleftarrow{\$} \{1,2,\dots,|G|\}$
2. $X \leftarrow g^x$
3. $x' \leftarrow A(X)$
4. **return** $x \stackrel{?}{=} x$

$\mathbf{Exp}^{\mathrm{dh}}_{G,g}(A)$

1. $x, y \xleftarrow{\$} \{1,2,\dots,|G|\}$
2. $X \leftarrow g^x$
3. $Y \leftarrow g^y$
4. $z \leftarrow A(X,Y)$
5. **return** $g^z \stackrel{?}{=} g^{xy}$

$\mathbf{Exp}^{\mathrm{ddh}}_{G,g}(A)$

1. $b \xleftarrow{\$} \{0,1\}$
2. $x, y, z \xleftarrow{\$} \{1,2,\dots,|G|\}$
3. $X \leftarrow g^x \quad Y \leftarrow g^y$
4. $Z_1 \leftarrow g^{xy}$
5. $Z_0 \leftarrow g^z$
6. $b' \leftarrow A(X,Y,Z_b)$
7. **return** $b' \stackrel{?}{=} b$

DLOG security $\Leftarrow$ DH security $\Leftarrow$ DDH security

DLOG security $\stackrel{?}{\Rightarrow}$ DH security $\not\Rightarrow$ DDH security

ElGamal – IND-CPA security: proof idea

Theorem: For *any* IND-CPA adversy A against ElGamal, there are adversaries B and C such that

$$\mathbf{Adv}^{\text{ind-cpa}}_{\text{ElGamal}}(A) \leq 2 \cdot \mathbf{Adv}^{\text{ddh}}_{G,g}(B) + \mathbf{Adv}^{\text{ind-cpa}}_{\Sigma}(C)$$

B

A, C

KeyGen

1. $sk = b \xleftarrow{\$} \{1, \ldots, |G|\}$
2. $pk = B \leftarrow g^b$
3. **return** (sk, pk)

Enc(pk, M)

1. $a \xleftarrow{\$} \{1, \ldots, |G|\}$
2. $A \leftarrow g^a$
3. $Z \leftarrow B^a = g^{ab}$
4. $C \leftarrow \Sigma.\text{Enc}(Z, M)$
5. **return** (A, C)

Dec(sk, C)

1. $Z \leftarrow A^b = g^{ab}$
2. $M \leftarrow \Sigma.\text{Dec}(Z, C)$
3. **return** M

$\Sigma = (\text{KeyGen}, \text{Enc}, \text{Dec})$

$\Sigma.\text{Enc} : G \times \mathcal{M} \to \mathcal{C}$

$\Sigma.\text{Dec} : G \times \mathcal{C} \to \mathcal{M}$

ElGamal – IND-CPA security: proof idea

Theorem: For *any* IND-CPA adversy A against ElGamal, there are adversaries B and C such that

$$\mathbf{Adv}^{\text{ind-cpa}}_{\text{ElGamal}}(A) \leq 2 \cdot \mathbf{Adv}^{\text{ddh}}_{G,g}(B) + \mathbf{Adv}^{\text{ind-cpa}}_{\Sigma}(C)$$

B

A, C

KeyGen

1. $sk = b \xleftarrow{\$} \{1, \ldots, |G|\}$
2. $pk = B \leftarrow g^b$
3. **return** (sk, pk)

Enc(pk, M)

1. $a \xleftarrow{\$} \{1, \ldots, |G|\}$
2. $A \leftarrow g^a$
3. $Z \leftarrow g^z$
4. $C \leftarrow \Sigma.\text{Enc}(Z, M)$
5. **return** (A, C)

can't tell due to DDH-security

Dec(sk, C)

1. $Z \leftarrow g^z$
2. $M \leftarrow \Sigma.\text{Dec}(Z, C)$
3. **return** M

$\Sigma = (\text{KeyGen}, \text{Enc}, \text{Dec})$

$\Sigma.\text{Enc} : G \times \mathcal{M} \to \mathcal{C}$

$\Sigma.\text{Dec} : G \times \mathcal{C} \to \mathcal{M}$

ElGamal – IND-CPA security: proof idea

Theorem: For *any* IND-CPA adversy A against ElGamal, there are adversaries B and C such that

$$\mathbf{Adv}^{\text{ind-cpa}}_{\text{ElGamal}}(A) \leq 2 \cdot \mathbf{Adv}^{\text{ddh}}_{G,g}(B) + \mathbf{Adv}^{\text{ind-cpa}}_{\Sigma}(C)$$

B

A, C

KeyGen

1. $sk = b \xleftarrow{\$} \{1, \ldots, |G|\}$
2. $pk = B \leftarrow g^b$
3. **return** (sk, pk)

Enc(pk, M)

1. $a \xleftarrow{\$} \{1, \ldots, |G|\}$
2. $A \leftarrow g^a$
3. $Z \leftarrow g^z$
4. $C \leftarrow \Sigma.\text{Enc}(Z, M)$
5. **return** (A, C)

can't tell due to DDH-security

Dec(sk, C)

1. $Z \leftarrow g^z$
2. $M \leftarrow \Sigma.\text{Dec}(Z, C)$
3. **return** M

$\Sigma = (\text{KeyGen}, \text{Enc}, \text{Dec})$

$\Sigma.\text{Enc} : G \times \mathcal{M} \to \mathcal{C}$

$\Sigma.\text{Dec} : G \times \mathcal{C} \to \mathcal{M}$

ElGamal – IND-CPA security: proof idea

Theorem: For *any* IND-CPA adversy A against ElGamal, there are adversaries B and C such that

$$\mathbf{Adv}^{\text{ind-cpa}}_{\text{ElGamal}}(A) \leq 2 \cdot \mathbf{Adv}^{\text{ddh}}_{G,g}(B) + \mathbf{Adv}^{\text{ind-cpa}}_{\Sigma}(C)$$

B

A, C

KeyGen

1. $sk = b \xleftarrow{\$} \{1, \ldots, |G|\}$
2. $pk = B \leftarrow g^b$
3. **return** (sk, pk)

Enc(pk, M)

1. $a \xleftarrow{\$} \{1, \ldots, |G|\}$
2. $A \leftarrow g^a$
3. $Z \xleftarrow{\$} G$
4. $C \leftarrow \Sigma.\text{Enc}(Z, M)$
5. **return** (A, C)

Dec(sk, C)

1. $Z \xleftarrow{\$} G$
2. $M \leftarrow \Sigma.\text{Dec}(Z, C)$
3. **return** M

$\Sigma = (\text{KeyGen}, \text{Enc}, \text{Dec})$

$\Sigma.\text{Enc} : G \times \mathcal{M} \to \mathcal{C}$

$\Sigma.\text{Dec} : G \times \mathcal{C} \to \mathcal{M}$

ElGamal – IND-CPA security: proof idea

Theorem: For *any* IND-CPA adversy A against ElGamal, there are adversaries B and C such that

$$\mathbf{Adv}^{\text{ind-cpa}}_{\text{ElGamal}}(A) \leq 2 \cdot \mathbf{Adv}^{\text{ddh}}_{G,g}(B) + \mathbf{Adv}^{\text{ind-cpa}}_{\Sigma}(C)$$

Enc(pk, M)

1. $a \xleftarrow{\$} \{1, \ldots, |G|\}$
2. $A \leftarrow g^a$
3. $Z \xleftarrow{\$} G$
4. $C \leftarrow \Sigma.\text{Enc}(Z, M)$
5. **return** (A, C)

B

A, C

secure by IND-CPA

KeyGen

1. $sk = b \xleftarrow{\$} \{1, \ldots, |G|\}$
2. $pk = B \leftarrow g^b$
3. **return** (sk, pk)

Dec(sk, C)

1. $Z \xleftarrow{\$} G$
2. $M \leftarrow \Sigma.\text{Dec}(Z, C)$
3. **return** M

$\Sigma = (\text{KeyGen}, \text{Enc}, \text{Dec})$

$\Sigma.\text{Enc} : G \times \mathcal{M} \to \mathcal{C}$

$\Sigma.\text{Dec} : G \times \mathcal{C} \to \mathcal{M}$

ElGamal – IND-CPA security: proof idea

Theorem: For *any* IND-CPA adversy A against ElGamal, there are adversaries B and C such that

$$\mathbf{Adv}^{\text{ind-cpa}}_{\text{ElGamal}}(A) \leq 2 \cdot \mathbf{Adv}^{\text{ddh}}_{G,g}(B) + \mathbf{Adv}^{\text{ind-cpa}}_{\Sigma}(C)$$

KeyGen

1. $sk = b \xleftarrow{\$} \{1, \ldots, |G|\}$
2. $pk = B \leftarrow g^b$
3. **return** (sk, pk)

Enc(pk, M)

1. $a \xleftarrow{\$} \{1, \ldots, |G|\}$
2. $A \leftarrow g^a$
3. $Z \xleftarrow{\$} G$
4. $C \leftarrow \Sigma.\text{Enc}(Z, M)$
5. **return** (A, C)

B

A, C

secure by IND-CPA

Dec(sk, C)

1. $Z \xleftarrow{\$} G$
2. $M \leftarrow \Sigma.\text{Dec}(Z, C)$
3. **return** M

$\Sigma = (\text{KeyGen}, \text{Enc}, \text{Dec})$

$\Sigma.\text{Enc} : G \times \mathcal{M} \to \mathcal{C}$

$\Sigma.\text{Dec} : G \times \mathcal{C} \to \mathcal{M}$

Hashed ElGamal

ElGamal. Enc : $G \times \mathcal{M} \rightarrow G \times \mathcal{C}$

ElGamal. Dec : $\boldsymbol{Z}_p \times G \times \mathcal{C} \rightarrow \mathcal{M}$

$G = \langle g \rangle$

B

A, C

KeyGen

1. $sk = b \xleftarrow{\$} \{1, \ldots, |G|\}$
2. $pk = B \leftarrow g^b$
3. **return** (sk, pk)

Enc(pk, M)

1. $a \xleftarrow{\$} \{1, \ldots, |G|\}$
2. $A \leftarrow g^a$
3. $Z \leftarrow B^a = g^{ab}$
4. $C \leftarrow \Sigma.\mathrm{Enc}(Z, M)$
5. **return** (A, C)

$\Sigma = (\mathrm{KeyGen}, \mathrm{Enc}, \mathrm{Dec})$

$\Sigma.\mathrm{Enc} : G \times \mathcal{M} \rightarrow \mathcal{C}$

$\Sigma.\mathrm{Dec} : G \times \mathcal{C} \rightarrow \mathcal{M}$

Dec(sk, C)

1. $Z \leftarrow A^b = g^{ab}$
2. $M \leftarrow \Sigma.\mathrm{Dec}(Z, C)$
3. **return** M

Hashed ElGamal

ElGamal. Enc : $G \times \mathcal{M} \rightarrow G \times \mathcal{C}$

ElGamal. Dec : $\boldsymbol{Z}_p \times G \times \mathcal{C} \rightarrow \mathcal{M}$

$G = \langle g \rangle$

B

A, C

KeyGen

1. $sk = b \xleftarrow{\$} \{1, \ldots, |G|\}$
2. $pk = B \leftarrow g^b$
3. **return** (sk, pk)

Enc(pk, M)

1. $a \xleftarrow{\$} \{1, \ldots, |G|\}$
2. $A \leftarrow g^a$
3. $Z \leftarrow B^a = g^{ab}$
4. $C \leftarrow \Sigma.\mathrm{Enc}(Z, M)$
5. **return** (A, C)

$H: G \rightarrow \{0,1\}^k$

$\Sigma = (\mathrm{KeyGen}, \mathrm{Enc}, \mathrm{Dec})$

$\Sigma.\mathrm{Enc} : \{0,1\}^k \times \mathcal{M} \rightarrow \mathcal{C}$

$\Sigma.\mathrm{Dec} : \{0,1\}^k \times \mathcal{C} \rightarrow \mathcal{M}$

Dec(sk, C)

1. $Z \leftarrow A^b = g^{ab}$
2. $M \leftarrow \Sigma.\mathrm{Dec}(Z, C)$
3. **return** M

Hashed ElGamal

$\text{ElGamal.Enc}: G \times \mathcal{M} \rightarrow G \times \mathcal{C}$

$\text{ElGamal.Dec}: \boldsymbol{Z}_p \times G \times \mathcal{C} \rightarrow \mathcal{M}$

$G = \langle g \rangle$

B

A, C

KeyGen

1. $sk = b \xleftarrow{\$} \{1, \ldots, |G|\}$
2. $pk = B \leftarrow g^b$
3. **return** (sk, pk)

Enc(pk, M)

1. $a \xleftarrow{\$} \{1, \ldots, |G|\}$
2. $A \leftarrow g^a$
3. $K \leftarrow H(B^a) = H(g^{ab})$
4. $C \leftarrow \Sigma.\text{Enc}(Z, M)$
5. **return** (A, C)

$H: G \rightarrow \{0,1\}^k$

$\Sigma = (\text{KeyGen}, \text{Enc}, \text{Dec})$

$\Sigma.\text{Enc}: \{0,1\}^k \times \mathcal{M} \rightarrow \mathcal{C}$

$\Sigma.\text{Dec}: \{0,1\}^k \times \mathcal{C} \rightarrow \mathcal{M}$

Dec(sk, C)

1. $K \leftarrow H(A^b) = H(g^{ab})$
2. $M \leftarrow \Sigma.\text{Dec}(Z, C)$
3. **return** M

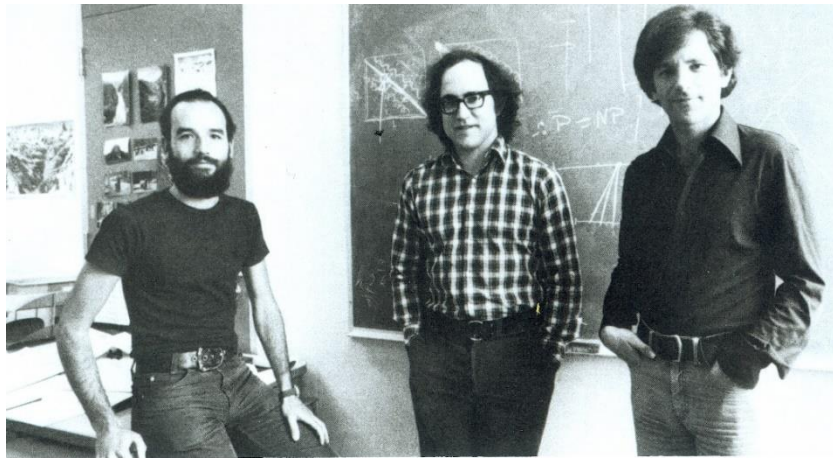
$$C = M^e \pmod{n}$$

The RSA encryption scheme

RSA

- Designed by Rivest, Shamir and Adleman in 1977
  - One year before ElGamal

- Used both for public key *encryption* and *digital signatures*



Adi Shamir Ron Rivest Leonard Adleman

The group $(Z_n^*,\cdot)$

$Z_p = \{0, 1, \dots, p-1\}$ $(Z_p,\cdot)$ is *not* a group!

$Z_p^* = \{1, \dots, p-1\}$ $(Z_p^*,\cdot)$ *is* a group!

$Z_n = \{0, 1, \dots, n-1\}$ $(Z_n,\cdot)$ is *not* a group!

$Z_n^* \neq \underbrace{\{1, \dots, n-1\}}_{Z_n^+}$ $(Z_n^+,\cdot)$ is *also not* a group!

$$Z_n^* = \underbrace{\text{invertible elements in } Z_n}_{(Z_n^*,\cdot) \text{ is a group!}} \overset{\text{theorem}}{=} \{a \in Z_n \mid \gcd(a,n) = 1\}$$

Not invertible	Invertible
2, 4, 5, 6, 8	1, 3, 7, 9

$Z_{10}^+ = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$

$2 \cdot 1 = 2 \mod 10$
$2 \cdot 2 = 4 \mod 10$
$2 \cdot 3 = 6 \mod 10$
$2 \cdot 4 = 8 \mod 10$
$2 \cdot 5 = 0 \mod 10$
$2 \cdot 6 = 2 \mod 10$
$2 \cdot 7 = 4 \mod 10$
$2 \cdot 8 = 6 \mod 10$
$2 \cdot 9 = 8 \mod 10$

$1 \cdot 1 = 1 \mod 10$

$3 \cdot 7 = 21 = 1 \mod 10$

$9 \cdot 9 = 81 = 1 \mod 10$

$10 = 2 \cdot 5$

$2 = 2$
$4 = 2 \cdot 2$
$5 = 5$
$6 = 2 \cdot 3$
$8 = 2 \cdot 2 \cdot 2$

Proof: let $d = \gcd(a,n)$

- a invertible $\Rightarrow \exists b \in Z_n$ such that $ab = 1 \bmod n \Rightarrow \exists k$: $ab = 1 + kn \Rightarrow ab - kn = 1 \Rightarrow d(a'b - kn') = 1 \Rightarrow d = 1$
- $d = 1 \Rightarrow$ Claim: $\exists s, t \in Z$ such that $sa + tn = d = 1 \Rightarrow sa = 1 - tn \Rightarrow sa = 1 \bmod n \Rightarrow a$ is invertible

Euler's $\phi(n)$ function

- $\phi(n) \overset{\text{def}}{=} |\boldsymbol{Z}_n^*| = |\{\, a \in \boldsymbol{Z}_n \mid \gcd(a,n) = 1 \,\}|$
- $\phi(p) = p - 1$
- $\phi(p \cdot q) = (p-1) \cdot (q-1)$
- **Note:** $\phi(n) \approx n - 2\sqrt{n} \approx n$ ($p \cdot q$)
  - i.e.: *almost all* integers are invertible for large p, q

$$\overbrace{1 \cdot p,\ 2 \cdot p,\ 3 \cdot p,\ \cdots\ (q-1) \cdot p}^{q-1}$$
$$\underbrace{1 \cdot q,\ 2 \cdot q,\ 3 \cdot q,\ \cdots\ (p-1) \cdot q}_{p-1}$$

$$\phi(pq) = \#\text{numbers less than } pq - \#\text{numbers less than } pq \text{ with } \gcd(x, pq) \neq 1$$
$$= (pq - 1) - (q - 1 + p - 1)$$
$$= pq - q - p + 1$$
$$= (p-1) \cdot (q-1)$$

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$\phi(n)$	**1**	**1**	**2**	**2**	**4**	**2**	**6**	**4**	**6**	**4**	**10**	**4**	**12**	**6**	**8**

Euler's Theorem

Theorem: if $(G,\circ)$ is a finite group, then for all $g \in G$:

$$g^{|G|} = e$$

- $(\boldsymbol{Z}_p^*,\cdot)$: $|\boldsymbol{Z}_p^*| = p - 1 \quad e = 1$

Fermat's theorem: if p is prime, then for all $a \neq 0 \pmod p$:

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

- $(\boldsymbol{Z}_n^*,\cdot)$: $|\boldsymbol{Z}_n^*| = \phi(n) \quad e = 1$

Euler's theorem: for all positive integers n, if $\gcd(a,n) = 1$ then

$$a^{\phi(n)} \equiv 1 \pmod n$$

Textbook RSA

$$\text{RSA.Enc}: \overbrace{\mathbb{Z}^+ \times \mathbb{Z}^*_{\phi(n)}}^{\mathcal{PK}} \times \mathbb{Z}^*_n \to \underset{\mathcal{M}}{\mathbb{Z}^*_n} \quad \underset{\mathcal{C}}{}$$

$$\text{RSA.Dec}: \underset{\mathcal{SK}}{\mathbb{Z}^*_{\phi(n)}} \times \underset{\mathcal{C}}{\mathbb{Z}^*_n} \to \underset{\mathcal{M}}{\mathbb{Z}^*_n}$$

KeyGen

1. $p, q \xleftarrow{\$}$ two random prime numbers
2. $n \leftarrow p \cdot q$
3. $\phi(n) = (p-1)(q-1)$
4. **choose** e such that $\gcd(e, \phi(n)) = 1$
5. $d \leftarrow e^{-1} \bmod \phi(n)$
6. $sk \leftarrow d \qquad pk \leftarrow (n, e)$
7. **return** (sk, pk)

Enc$(pk = (n, e), M \in \mathbb{Z}^*_n)$

1. $C \leftarrow M^e \bmod n$
2. **return** C

pk

C

Dec$(sk = d, C \in \mathbb{Z}^*_n)$

1. $M \leftarrow C^d \bmod n$
2. **return** M

Common choices of e: 3, 11_2 17, 10001_2 65 537, $1\ 0000\ 0000\ 0000\ 0001_2$

Textbook RSA – correctness

Theorem: if $(G,\circ)$ is a finite group, then for all $g \in G$:

$$g^{|G|} = e$$

Euler's theorem: for all $a \in \boldsymbol{Z}_n^*$

$$a^{\phi(n)} \equiv 1 \pmod{n}$$

Corollary I: $g^i = g^{i \bmod |G|} = a^{i \bmod \phi(n)}$

$\mathrm{Dec}\big(sk, \mathrm{Enc}(pk, M)\big) = M$ $\qquad d = e^{-1} \bmod \phi(n) \iff ed = 1 \bmod \phi(n)$

$C^d = M^{ed} = M^{ed \bmod \phi(n)} = M^1 = M \bmod n$

Fact: RSA also works for $M \in \boldsymbol{Z}_n$

KeyGen

1. $p, q \xleftarrow{\$}$ two random prime numbers
2. $n \leftarrow p \cdot q$
3. $\phi(n) = (p-1)(q-1)$
4. **choose** e such that $\gcd\big(e, \phi(n)\big) = 1$
5. $d \leftarrow e^{-1} \bmod \phi(n)$
6. $sk \leftarrow d \qquad pk \leftarrow (n, e)$
7. **return** (sk, pk)

Enc$(pk = (n,e), M \in \boldsymbol{Z}_n^*)$

1. $C \leftarrow M^e \bmod n$
2. **return** C

Dec$(sk = d, C \in \boldsymbol{Z}_n^*)$

1. $M \leftarrow C^d \bmod n$
2. **return** M

Textbook RSA – security

- Textbook RSA is *not* IND-CPA secure!
  - Deterministic
  - Malleable

Enc$(pk = (n, e), M \in \mathbf{Z}_n)$

1. $C \leftarrow M^e \bmod n$
2. **return** C

pk

C

KeyGen

1. $p, q \xleftarrow{\$}$ two random prime numbers
2. $n \leftarrow p \cdot q$
3. $\phi(n) = (p-1)(q-1)$
4. **choose** e such that $\gcd(e, \phi(n)) = 1$
5. $d \leftarrow e^{-1} \bmod \phi(n)$
6. $sk \leftarrow d \quad pk \leftarrow (n, e)$
7. **return** (sk, pk)

Dec$(sk = d, C \in \mathbf{Z}_n)$

1. $M \leftarrow C^d \bmod n$
2. **return** M

- Many other attacks as well*

- Textbook RSA is *not* an encryption scheme!

- So what is it? Answer: a *one-way (trapdoor) permutation*

* https://crypto.stackexchange.com/questions/20085/which-attacks-are-possible-against-raw-textbook-rsa

The RSA-problem

$\text{RSA.Enc}_{pk}: \; \boldsymbol{Z}_n^* \to \boldsymbol{Z}_n^*$

$\text{RSA.Enc}_{pk}: \; M \mapsto M^e \bmod n$

Enc$(pk = (n, e), M \in \boldsymbol{Z}_n)$

1. $C \leftarrow M^e \bmod n$
2. **return** C

pk

C

KeyGen

1. $p, q \xleftarrow{\$}$ two random prime numbers
2. $n \leftarrow p \cdot q$
3. $\phi(n) = (p-1)(q-1)$
4. **choose** e such that $\gcd(e, \phi(n)) = 1$
5. $d \leftarrow e^{-1} \bmod \phi(n)$
6. $sk \leftarrow d \quad pk \leftarrow (n, e)$
7. **return** (sk, pk)

Dec$(sk = d, C \in \boldsymbol{Z}_n)$

1. $M \leftarrow C^d \bmod n$
2. **return** M

- **RSA-problem:** given $pk = (e, n)$ and $C = M^e \bmod n$
  find M

  - **RSA-assumption:** the RSA-problem is hard for large random primes p, q (≈ 1024 bits each)

  - Easy is d is known: simply compute $C^d \bmod n$
  - Easy if $\phi(n)$ is known: simply compute $1/e = e^{-1} = d \bmod \phi(n)$ and do the above
  - Easy if p and q are known: compute $\phi(n) = (p-1) \cdot (q-1)$ and do the above

  - Hence: RSA-assumption $\Longrightarrow$ must be hard to find d, $\phi(n)$ and p, q given e, n (**factoring problem**)
    - What about the converse: factoring is hard $\Longrightarrow$ RSA-problem is hard? *Open problem!*
    - Note: given $\phi(n)$ and n $\Longrightarrow$ easy to find p and q (exercise)

How hard is factoring?

- In practice factoring is only known way to break RSA
- Factoring $n = pq$ believed to be hard for large p and q
- Naïve $\text{Factor}(n)$:
  - 3 divides n? return $3 \cdot \text{Factor}(n/3)$
  - 5 divides n? return $5 \cdot \text{Factor}(n/5)$
  - 7 divides n? return $7 \cdot \text{Factor}(n/7)$
  - ⋮
  - $\lfloor\sqrt{n}\rfloor$ divides n? return $\lfloor\sqrt{n}\rfloor \cdot \text{Factor}(n/\lfloor\sqrt{n}\rfloor)$
  - Return n
  - *Very* inefficient: $n \approx 2^k \implies \pi(n) \approx \frac{2^k}{\ln 2^k} \approx \frac{2^k}{k}$

Algorithm	**Time to factor** $n \approx 2^k$
Naive	$\approx \mathcal{O}(2^k)$
Quadratic sieve	$\approx \mathcal{O}\left(e^{c\sqrt{k}\sqrt{\ln k}}\right)$
Number field sieve	$\approx \mathcal{O}\left(e^{1.92\sqrt[3]{k}\cdot\sqrt[1.5]{\ln k}}\right)$

- *Much* faster factoring algorithms known: current record 829-bit number (Heninger et al.'20)

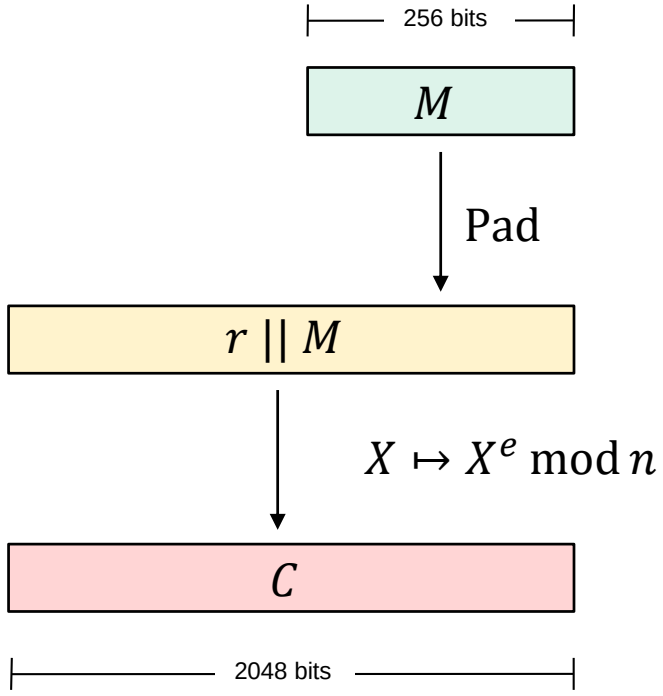
RSA in practice

- Textbook RSA is deterministic ⟹ cannot be IND-CPA secure

- How to achieve IND-CPA?
  - *pad* message with random data before applying RSA function
  - PKCS#1v1.5
  - RSA-OAEP

- RSA *encryption* not used much in practice anymore
  - Mostly **key transport**: message is a random 256-bit key used in a subsequent symmetric encryption scheme

- RSA *digital signatures* still very common



Summary

- Public-key encryption security goals: IND-CPA and IND-CCA

- ElGamal
  - Public-key encryption from Diffie-Hellman key exchange
  - IND-CPA secure if the DDH assumption holds in the group and the symmetric encryption scheme is (one-time) IND-CPA secure
  - Hashed-ElGamal: hash the derived Diffie-Hellman key to obtain a symmetric key $K \in \{0,1\}^k$
    - DDH assumption can be weakened to DH assumption (but stronger requirement on hash function)

- RSA
  - *Not* a public-key encryption scheme directly
  - Can (with random padding) be used to create a IND-CPA secure public-key encryption scheme
  - Must be hard: RSA-problem and factoring problem