

Lecture 7. Public-key encryption

Passive adversaries, no shared secret

Bruno Grenet



<https://membres-ljk.imag.fr/Bruno.Grenet/IntroCrypto.html>

Introduction to cryptology
Université Grenoble Alpes – IM²AG
M1 INFO, MOSIG & AM

Introduction

Symmetric (or *private key*) encryption

- ▶ Alice and Bob share a common key k
- ▶ Alice wants to send m to Bob:
 1. Alice computes $c \leftarrow \text{Enc}_k(m)$
 2. Alice sends c to Bob
 3. Bob computes $m' \leftarrow \text{Dec}_k(c)$

and if all goes well: $m = m'$

Key exchange

- ▶ Alice and Bob must agree on a common key k .
- ▶ Diffie-Hellman protocol based on cyclic groups

Public-key (*a.k.a* asymmetric) cryptography: no prior key exchange!

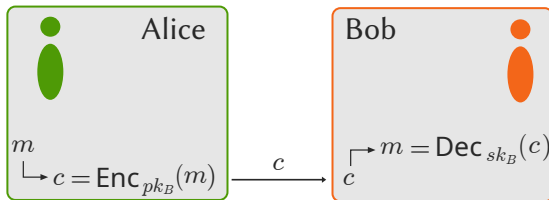
Contents

1. Public-key encryption

2. ElGamal encryption scheme

3. Hybrid encryption

Principle



Encryption Alice encrypts m with Bob's public key: $c \leftarrow \text{Enc}_{pk_B}(m)$

Decryption Bob decrypts c with his private key: $m' \leftarrow \text{Dec}_{sk_B}(c)$

Correctness if $m = m'$

Security if an adversary cannot compute m , knowing both c **and** pk_B

Formalization of public-key encryption

Definition

A public-key encryption scheme is given by 3 algorithms:

$\text{Gen}_n()$ returns a pair of keys (pk, sk) where n is the *security parameter*

$\text{Enc}_{pk}(m)$ returns a ciphertext c for a message $m \in \mathcal{M}_{pk}$

$\text{Dec}_{sk}(c)$ returns a message m or an error

Correctness: for all $(pk, sk) \leftarrow \text{Gen}_n()$ and all $c \leftarrow \text{Enc}_{pk}(m)$, $\text{Dec}_{sk}(c) = m$

Remarks

- ▶ pk is the *public key* and sk the *private (or secret) key*.
- ▶ The public key defines the message space $\mathcal{M}_{pk}$
 - ▶ require a mapping from $\{0, 1\}^*$ to $\mathcal{M}_{pk}$
 - ▶ often obvious
- ▶ The security parameter n sets the lengths of the keys
- ▶ Gen is implicit for symmetric encryption

often implicit
e.g: return $k \leftarrow \{0, 1\}^n$

IND-CPA security

IND-CPA indistinguishability game

Challenger: $(pk, sk) \leftarrow \text{Gen}()$

Adversary: given pk , produces $m_0, m_1 \in \mathcal{M}_{pk}$ of the same size

Challenger: $b \leftarrow \{0, 1\}; c \leftarrow \text{Enc}_{pk}(m_b)$

Adversary: given c , returns a bit b' ; *success* if $b = b'$

Advantages

- ▶ $\text{Adv}_{\text{Enc}}^{\text{IND-CPA}}(A) = \left| \Pr[b' = 1 | b = 1] - \Pr[b' = 1 | b = 0] \right| = |2 \Pr[\text{SUCCESS}] - 1|$
- ▶ $\text{Adv}_{\text{Enc}}^{\text{IND-CPA}}(t) = \max_{A_t} \text{Adv}_{\text{Enc}}^{\text{IND-CPA}}(A_t)$ where A_t has running time $\leq t$

Remarks

- ▶ Extremely similar with IND-CPA for symmetric encryption
 - ▶ No *oracle* access to $\text{Enc}_{pk}(\cdot)$
- ▶ $\text{Enc}_{pk}(\cdot)$ must be randomized: Why?
- ▶ No *perfectly secret* public-key encryption

The public key is... public!

IND-CCA2 security

IND-CCA2 indistinguishability game

Challenger: $(pk, sk) \leftarrow \text{Gen}()$

Adversary: *has oracle access to $\text{Dec}_{sk}(\cdot)$ during the whole experiment*
given pk , produces $m_0, m_1 \in \mathcal{M}_{pk}$ of same size

Challenger: $b \leftarrow \{0, 1\}; c \leftarrow \text{Enc}_{pk}(m_b)$

Adversary: given c , returns a bit b' ; success if $b' = b$ *not allowed to ask $\text{Dec}_{sk}(c)$!*

Advantages

- ▶ $\text{Adv}_{\text{Enc}}^{\text{IND-CCA2}}(A) = \left| \Pr[b' = 1 | b = 1] - \Pr[b' = 1 | b = 0] \right| = |2 \Pr[\text{SUCCESS}] - 1|$
- ▶ $\text{Adv}_{\text{Enc}}^{\text{IND-CCA2}}(q, t) = \max_{A_t} \text{Adv}_{\text{Enc}}^{\text{IND-CCA2}}(A_{q,t})$
where $A_{q,t}$ has running time $\leq t$ and makes $\leq q$ queries to $\text{Dec}_{sk}(\cdot)$

Remarks

- ▶ The security notion needed in practice
- ▶ Equivalent to *non-malleability*: Knowing $c \leftarrow \text{Enc}_{pk}(m)$ but not m , it is *hard* to compute c' such that $\text{Dec}_{sk}(c') = f(m)$ for some chosen $f(\cdot)$

What about *multiple* encryptions?

Two (equivalent) questions

- ▶ What happens if we re-use the same public key several times?
- ▶ Can we encrypt arbitrary long messages?

Reminder in the symmetric case

- ▶ Block ciphers $\rightarrow$ fixed-length deterministic encryption
- ▶ Modes of operations $\rightarrow$ variable-length randomized encryption

Security for multiple encryption

- ▶ The building block is already randomized
- ▶ No need for modes of operations
- ▶ Formally: $\text{IND-CPA} \Rightarrow \text{IND-CPA}$ for multiple encryptions
(idem for IND-CCA2)

$$\text{Enc}_{pk}(m_1) \parallel \dots \parallel \text{Enc}_{pk}(m_B)$$

Encryption: public-key or symmetric + key exchange?

Advantages of symmetric encryption + key exchange

- ▶ Symmetric encryption usually lighter than public-key encryption
 - ▶ Reduced communications
 - ▶ Reduced computations

Advantages of public-key encryption

- ▶ Each user has only one private key to keep in the long run
- ▶ Works in asynchronous situations

Hybrid encryption

- ▶ General idea
 - ▶ Encrypt the message m with a symmetric key $k \rightarrow c$
 - ▶ Encrypt the key k with a public key $pk \rightarrow c'$
 - ▶ Send c and $c' \rightarrow$ decryption in the obvious manner
- ▶ More general framework: we can sometimes do *better* than encrypting the key k
 - ▶ KEM/DEM Paradigm

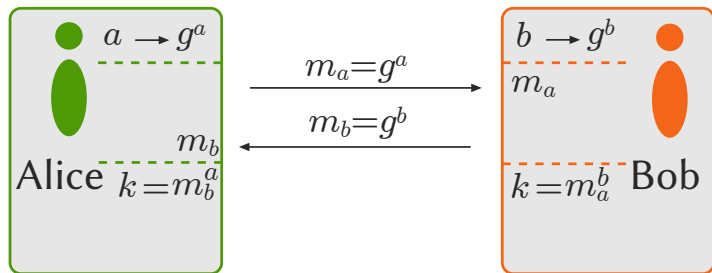
Contents

1. Public-key encryption

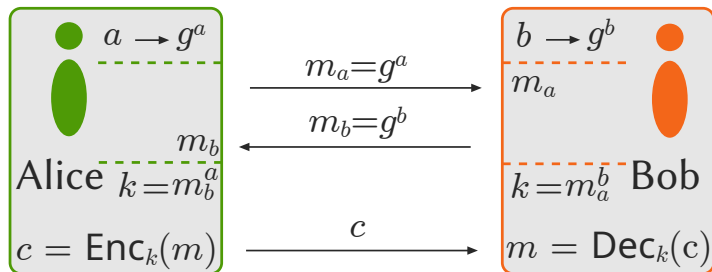
2. ElGamal encryption scheme

3. Hybrid encryption

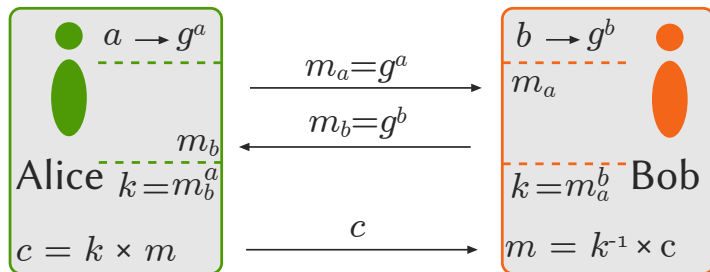
From Diffie-Hellman to ElGamal



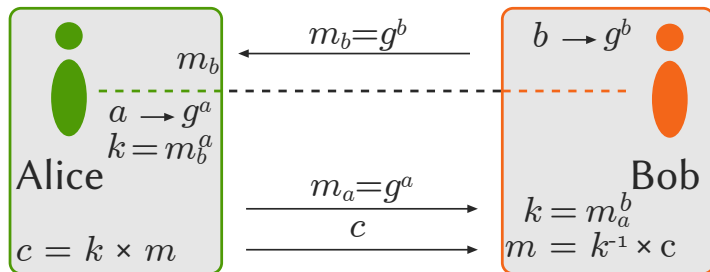
From Diffie-Hellman to ElGamal



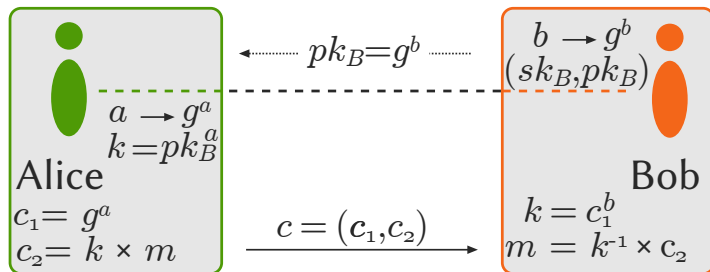
From Diffie-Hellman to ElGamal



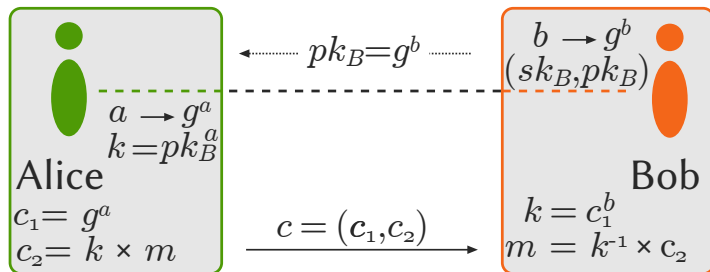
From Diffie-Hellman to ElGamal



From Diffie-Hellman to ElGamal



From Diffie-Hellman to ElGamal



Question

Prove that $\text{Enc}_k(m) = k \times m$ provides a secure encryption scheme

Remark

Several senders can all use Bob's public key:

security for a single encryption $\Rightarrow$ security for multiple encryptions

ElGamal encryption scheme

Construction

Public: a cyclic group $G = \langle g \rangle$ of order $q \simeq 2^n$

Gen():

1. $x \leftarrow \{0, \dots, q-1\}$
2. $h \leftarrow g^x$
3. Return $pk = h$ and $sk = x$

$$(\mathcal{M}_{pk} = G)$$

Enc_{pk}(m):

1. $y \leftarrow \{0, \dots, q-1\}$
2. $c_1 \leftarrow g^y$; $c_2 \leftarrow h^y \cdot m$
3. Return $c = (c_1, c_2)$

Dec_{sk}(c₁, c₂):

1. Return $\hat{m} = c_2 \cdot c_1^{-x}$

Correction

$$\hat{m} = c_2 \cdot c_1^{-x} = (h^y \cdot m) \cdot (g^y)^{-x} = g^{xy} \cdot m \cdot g^{-xy} = m$$

Group multiplication for encryption

Lemma

Let $G = \langle g \rangle$ be a cyclic group of order q and $z \leftarrow \{0, \dots, q-1\}$ (uniformly):

- (i) g^z is a uniform element of G
- (ii) for any $m \in G$, $g^z \cdot m$ is uniform in G

Proof

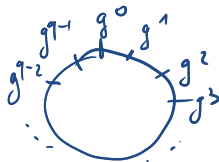
(i) • uniform $\Leftrightarrow \forall h \in G \quad \Pr_z[g^z = h] = 1/q$

• $z \mapsto g^z$ is a bijection $\{0, \dots, q-1\} \rightarrow G$ [its inverse is the discrete log in base g]

$$\hookrightarrow \Pr_z[g^z = h] = \Pr_z[z = \log_g(h)] = 1/q$$

(ii) • Since g^z is uniform, it remains to show that $h \cdot m$ is uniform when $h \leftarrow G$.

• Same proof: $m \mapsto h \cdot m$ is a bijection of inverse $m \mapsto h^{-1} \cdot m$



Security proof

Theorem

Under DDH assumption in G , ElGamal encryption scheme is IND-CPA secure. More precisely, $\text{Adv}_{\text{ElGamal}(G)}^{\text{IND-CPA}}(t) \leq 2 \cdot \text{Adv}_G^{\text{DDH}}(t)$ for all t .

Proof DDH Game

C_{DDH} : simulates the DH protocol

$\hookrightarrow g^a, g^b, g^{ab}$

$\cdot d \leftarrow \{0,1\}$ and $\hat{h} \leftarrow \begin{cases} g^{ab} & \text{if } d=1 \\ h \leftarrow G & \text{if } d=0 \end{cases}$

A_{DDH} : receives $g^a, g^b, \hat{h}$ and outputs $\hat{d}$.

$$\text{Adv}_G^{\text{DDH}}(A_{\text{DDH}}) = |\Pr[\hat{d}=1|d=1] - \Pr[\hat{d}=1|d=0]|$$

Goal - Take the best adv. A_{CPA} with adv. α . (and running time t)

- Use A_{CPA} to build an adv. A_{DDH} with adv. $\frac{\alpha}{2}$. (and run time t)

- then $\text{Adv}_G^{\text{DDH}}(t) \geq \text{Adv}_G^{\text{DDH}}(A_{\text{DDH}}) = \frac{\alpha}{2} = \frac{1}{2} \text{Adv}_{\text{Enc}}^{\text{IND-CPA}}(A_{\text{CPA}}) = \frac{1}{2} \text{Adv}_{\text{Enc}}^{\text{IND-CPA}}(t)$

IND-CPA Game

C_{CPA} : generates keys (pk, sk)

$A_{\text{CPA}}^{(1)}$: creates $m_0, m_1 \in \mathcal{M}_{pk}$

C_{CPA} : $\mu \leftarrow \{0,1\}$ and $c \leftarrow \text{Enc}_{pk}(m_\mu)$

$A_{\text{CPA}}^{(2)}$: gets c and outputs $\hat{\mu}$.

$$\begin{aligned} \text{Adv}_{\text{Enc}}^{\text{IND-CPA}}(A_{\text{CPA}}) &= |\Pr[\hat{\mu}=1|\mu=1] - \Pr[\hat{\mu}=1|\mu=0]| \\ &= |2 \Pr[\text{success}] - 1| \quad (*) \end{aligned}$$

Security proof

Theorem

Under DDH assumption in G , ElGamal encryption scheme is IND-CPA secure. More precisely, $\text{Adv}_{\text{ElGamal}(G)}^{\text{IND-CPA}}(t) \leq 2 \cdot \text{Adv}_G^{\text{DDH}}(t)$ for all t .

Proof Build A_{DDH} :

Input: g^a, g^b and $\hat{k} = \begin{cases} g^{ab} & \text{if } d=1 \\ \text{random } h \leftarrow G & \text{if } d=0 \end{cases}$

Idea: A_{DDH} simulates the IND-CPA game, playing the role of C_{CPA} .

1. A_{DDH} generates keys: $pk = g^a$ and $sk = a$ (unknown to A_{DDH})

2. Call $A_{\text{CPA}}^{(1)} \rightsquigarrow m_0, m_1$

3. A_{DDH} takes $\mu \leftarrow \{0, 1\}$ and computes $c \leftarrow (g^b, \hat{k} \cdot m_\mu)$

4. A_{DDH} calls $A_{\text{CPA}}^{(2)}$ to get $\hat{\mu}$.

5. A_{DDH} outputs $\hat{d} = \begin{cases} 1 & \text{if } \mu = \hat{\mu} \\ 0 & \text{if } \mu \neq \hat{\mu} \end{cases}$

Security proof

Theorem

Under DDH assumption in G , ElGamal encryption scheme is IND-CPA secure. More precisely, $\text{Adv}_{\text{ElGamal}(G)}^{\text{IND-CPA}}(t) \leq 2 \cdot \text{Adv}_G^{\text{DDH}}(t)$ for all t .

Proof

Analysis: two cases: either $\lambda=0$ ($\hat{k}=h \leftarrow G$) or $\lambda=1$ ($\hat{k}=g^{ab}$)

$\lambda=0$ $c = (g^b, h \cdot m_\mu)$ by the lemma, $h \cdot m_\mu$ is uniform in G and $A_{\text{CPA}}^{(c)}$ has no information on m_μ : $\Pr[\hat{\mu} = \mu | \lambda=0] = 1/2$

$\lambda=1$ $c = (g^b, g^{ab} \cdot m_\mu)$ is a valid encryption of m_μ : $\Pr[\hat{\mu} = \mu | \lambda=1] = \frac{1}{2}(1 \pm \alpha)$ (*) by assumption

Conclusion $\text{Adv}_G^{\text{DDH}}(A_{\text{DDH}}) = |\Pr[\hat{\lambda}=1 | \lambda=1] - \Pr[\hat{\lambda}=1 | \lambda=0]| = |\Pr[\hat{\mu} = \mu | \lambda=1] - \Pr[\hat{\mu} = \mu | \lambda=0]|$
 $= |\frac{1}{2}(1 \pm \alpha) - \frac{1}{2}| = |\pm \alpha/2| = \alpha/2$.

Additional remarks

Choice of the group G

- ▶ The order q must be prime, for DDH
- ▶ Several choices (subgroup of $(\mathbb{Z}/p\mathbb{Z})^\times, \dots$)
 - ▶ different security levels
 - ▶ standardization by NIST and other agencies

$\log p$	$\log q$	security
2048	224	112
3072	256	128
7680	384	192
15360	512	256

Message space G ?

- ▶ Solution 1: bijection between G and $\{0, 1\}^\ell$
- ▶ Solution 2: ElGamal-based KEM + key derivation function

for some G

CCA (in)security

- ▶ If $(c_1, c_2) \leftarrow \text{Enc}_{pk}(m)$, then $\text{Dec}_{sk}(c_1, m' \cdot c_2) = m' \cdot c_2 \cdot c_1^{-sk} = m' \cdot m$
 $\Rightarrow$ ElGamal encryption scheme is *malleable*, hence not CCA secure
- ▶ CCA-secure variants exist, mainly using hybrid encryption

Contents

1. Public-key encryption

2. ElGamal encryption scheme

3. Hybrid encryption

Introduction

Observation

- ▶ Public-key encryption scheme designed for small messages
- ▶ Block-by-block encryption possible...
- ▶ ... but expensive

large *ciphertext expansion*

Use of key exchange

1. Agree on a shared key k
2. Use symmetric encryption with k

The idea of hybrid encryption

Sender encrypts the message with a key $k \rightarrow c$

encrypts the key k with the public key of the receiver *encapsulated key*

Receiver decrypts first the encapsulated key with its secret key $\rightarrow k$

decrypts c using $k \rightarrow m$

The KEM/DEM paradigm

Definition

A **Key Encapsulation Mechanism** (KEM) is given by three algorithms:

$\text{Gen}_n()$: produces a pair (pk, sk)

$\text{Encaps}_{pk}()$: produces a pair (c, k)

$\text{Decaps}_{sk}(c)$: returns k

Usage

To send m using public-key pk :

1. $(c, k) \leftarrow \text{Encaps}_{pk}()$
2. $c' \leftarrow \text{Enc}_k(m)$ (with symmetric encryption)

key encapsulation
data encapsulation

Security notions

- ▶ Definitions of IND-CPA / IND-CCA security for KEMs
- ▶ IND-CPA KEM and symmetric encryption $\Rightarrow$ IND-CPA public-key encryption
- ▶ Ditto for IND-CCA

Generic construction from public-key encryption scheme

Definition

Given: Public-key encryption scheme (Enc, Dec)

$\text{Encaps}_{pk}()$:
1. $k \leftarrow \{0, 1\}^n$
2. $c \leftarrow \text{Enc}_{pk}(k)$

$\text{Decaps}_{sk}(c)$:
1. $k \leftarrow \text{Dec}_{sk}(c)$

Security

- ▶ If the public-key scheme is IND-CPA secure, the KEM too
- ▶ Ditto with IND-CCA security

Comments

- ▶ Using ElGamal for instance, must encode k in the group G
- ▶ Not the only nor best solution:
 - ▶ We need: from pk , produce c and k such that k can be recovered from sk and c
 - ▶ We don't need: c to be an actual encryption of k using pk

DDH-based KEM

Construction

Public: a cyclic group $G = \langle g \rangle$ of order q

Gen():

1. $x \leftarrow \{0, \dots, q-1\}$
2. $h \leftarrow g^x$
3. $H \leftarrow$ some hash function from G to $\{0, 1\}^\ell$
4. return $pk = (h, H)$ and $sk = (x, H)$

Encaps_{pk}():

1. $y \leftarrow \{0, \dots, q-1\}$
2. return $c \leftarrow g^y$ and $k \leftarrow H(h^y)$

Decaps_{sk}(c):

1. return $k \leftarrow H(c^x)$

Correction

$$H(c^x) = H(g^{xy}) = H(h^y) = k$$

Security (admitted)

- ▶ If DDH is hard in G and H is regular, the KEM is IND-CPA secure
- ▶ If CDH is hard in G and H is a random oracle, the KEM is IND-CPA secure

Conclusion

Public-key encryption schemes

- ▶ Usually heavier than symmetric encryption schemes
- ▶ Good solution: use hybrid encryption KEM/DEM paradigm
- ▶ Key management can be tricky → *public key infrastructures*

ElGamal encryption scheme

- ▶ Basic idea very close to Diffie-Hellman key exchange protocol
- ▶ Requires other tools to make it IND-CCA secure
- ▶ Security based on DDH or CDH assumption

Other protocols

- ▶ Variant of the DDH-based KEM is standardized as DHIES/ECIES
 - ▶ IND-CPA or IND-CCA security proofs under suitable assumptions
- ▶ Cramer & Shoup protocol: IND-CCA security under DDH assumption
- ▶ Other unrelated protocols using completely different assumptions RSA, LWE, ...