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 $c \leftarrow \mathrm{Enc}_k(m)$
  2. Alice sends c to Bob
  3. Bob computes $m' \leftarrow \mathrm{Dec}_k(c)$

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

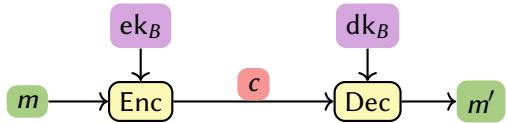
Key exchange

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

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

Contents

Principle



Encryption Alice encrypts m with Bob's encryption key: $c \leftarrow \mathrm{Enc}_{\mathrm{ek}_B}(m)$
Decryption Bob decrypts c with his decryption key: $m' \leftarrow \mathrm{Dec}_{\mathrm{dk}_B}(c)$

Properties

Correctness: m' should equal m
Security: an adversary should not be able to compute m, knowing both c **and** ek_B

Public and private/secret keys

- The encryption key is often called the public key — pk
- The decryption key is often called the private or secret key — sk

Formalization of public-key encryption

Definition

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

$\mathsf{Gen}_n()$ returns a pair of keys $(\mathsf{ek}, \mathsf{dk})$ where n is the *security parameter*

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

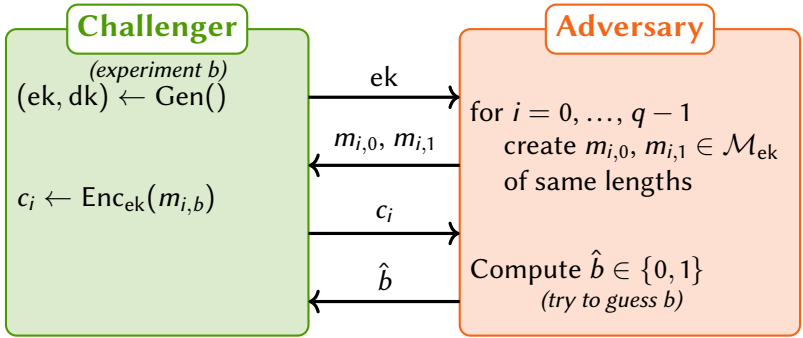
$\mathsf{Dec}_{\mathsf{dk}}(c)$ returns a message m or an error

Correctness: for all $(\mathsf{ek}, \mathsf{dk}) \leftarrow \mathsf{Gen}_n()$ and all $c \leftarrow \mathsf{Enc}_{\mathsf{ek}}(m)$, $\mathsf{Dec}_{\mathsf{dk}}(c) = m$

Remarks

- The message space $\mathcal{M}_{\mathsf{ek}}$ depends on the encryption key
  - require a(n often obvious) mapping from $\{0,1\}^*$ to $\mathcal{M}_{\mathsf{ek}}$
- The security parameter n sets the lengths of the keys *often implicit*
- Gen is implicit for symmetric encryption *e.g*: return $k \twoheadleftarrow \{0,1\}^n$

IND-CPA security



Remarks

- Almost the same definition as for symmetric encryption
  - Need to send ek to define $\mathcal{M}_{\mathrm{ek}}$
  - The adversary can encrypt messages
- $\mathrm{Enc}_{\mathrm{ek}}(\cdot)$ must be randomized: Why?
- No *perfectly secret* public-key encryption

Advantages

Advantage of an adversary

$$\mathrm{Adv}_{\mathrm{Enc}}^{\text{IND-CPA}}(\mathcal{A}) = \left|\Pr\left[\mathcal{A}^{\mathrm{Enc}} \to 1 \middle| b = 1\right] - \Pr\left[\mathcal{A}^{\mathrm{Enc}} \to 1 \middle| b = 0\right]\right|$$

Advantage function

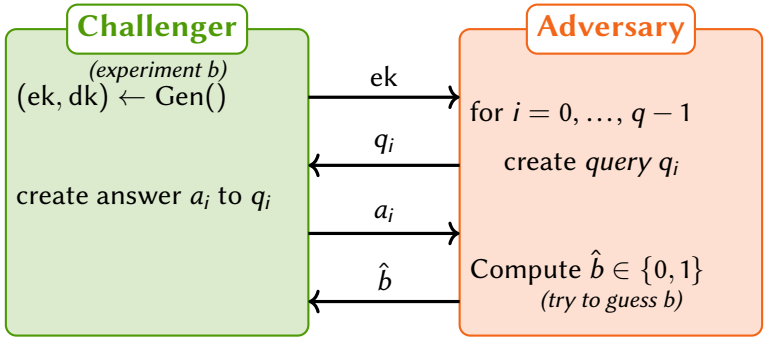
$$\mathrm{Adv}_{\mathrm{Enc}}^{\text{IND-CPA}}(q, t) = \max_{\mathcal{A}_{q,t}} \mathrm{Adv}_{\mathrm{Enc}}^{\text{IND-CPA}}(\mathcal{A}_{q,t})$$

where $\mathcal{A}_{q,t}$ makes q queries and runs in time $\leq t$

You've already heard that...

- No *formal* definition of IND-CPA secure in this course — concrete security
- The advantage function is a measure on the IND-CPA security

IND-CCA2 security



Two kinds of queries

Encryption query: $q_i = (m_{i,0}, m_{i,1})$, $a_i = \mathsf{Enc}_{\mathsf{ek}}(m_{i,b})$ *same lengths*

Decryption query: $q_i = \hat{c}_i$, $a_i = \mathsf{Dec}_{\mathsf{dk}}(\hat{c}_i)$ $\hat{c}_i \notin \{c_0, \ldots, c_{i-1}\}$

Advantages, etc.

- As in IND-CPA security

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

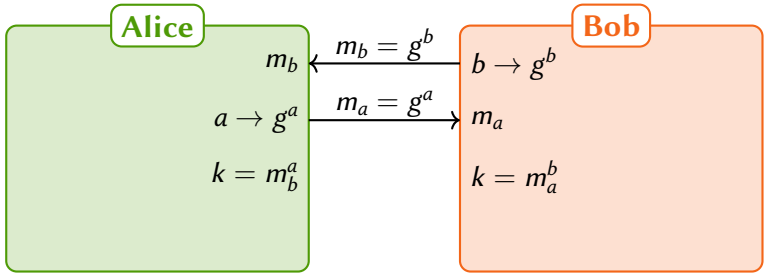
- Each user has only one private key to keep in the long run
- Works in asynchronous situations

Hybrid encryption

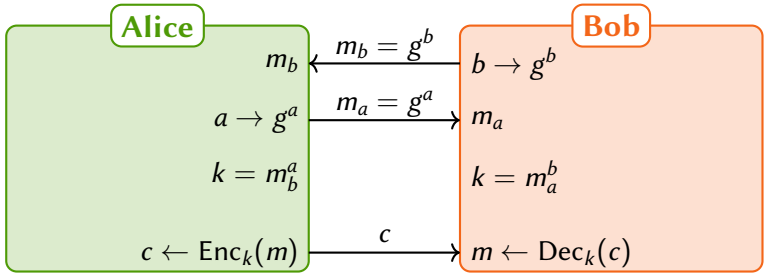
- General idea
  - Encrypt the message m with a symmetric key $k \rightarrow c$
  - Encrypt the key k with a public encryption key ek $\rightarrow c'$
  - Send c and $c' \rightarrow$ decryption in the obvious manner
- More general framework: KEM/DEM Paradigm
  - We can sometimes do *better* than encrypting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

Contents

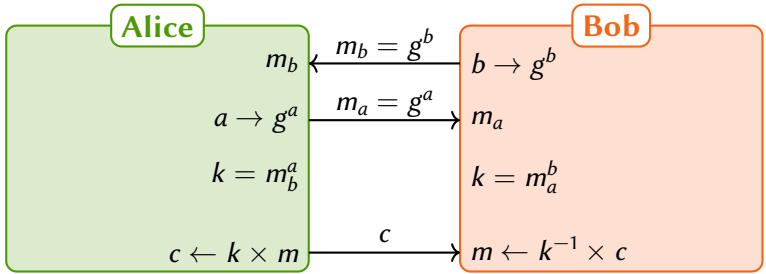
From Diffie-Hellman to ElGamal



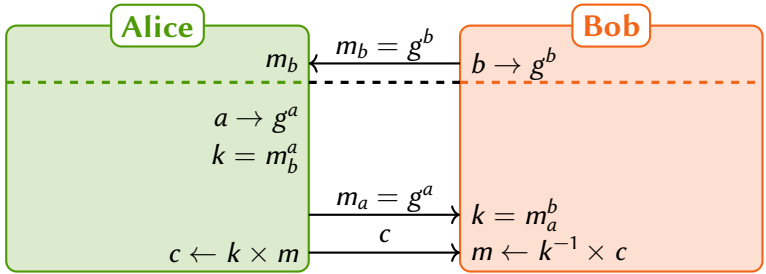
From Diffie-Hellman to ElGamal



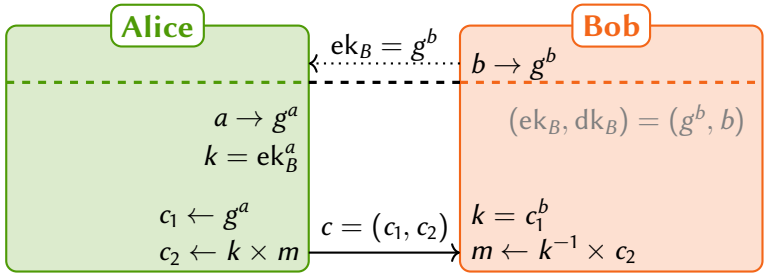
From Diffie-Hellman to ElGamal



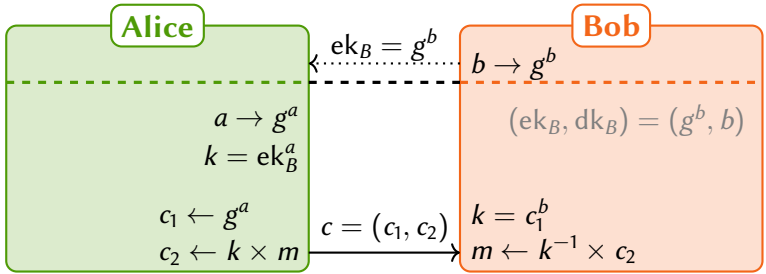
From Diffie-Hellman to ElGamal



From Diffie-Hellman to ElGamal



From Diffie-Hellman to ElGamal



Question

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

ElGamal encryption scheme

Construction

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

Gen(): 1. $x \twoheadleftarrow \{0, ..., q-1\}$
2. $h \leftarrow g^x$
3. Return $\mathsf{ek} = h$ and $\mathsf{dk} = x$ $(\mathcal{M}_{\mathsf{ek}} = G)$

$\mathsf{Enc}_{\mathsf{ek}}(m)$: 1. $y \twoheadleftarrow \{0, ..., q-1\}$
2. $c_1 \leftarrow g^y$; $c_2 \leftarrow \mathsf{ek}^y \cdot m$
3. Return $c = (c_1, c_2)$

$\mathsf{Dec}_{\mathsf{dk}}(c_1, c_2)$: 1. Return $\hat{m} = c_2 \cdot c_1^{-\mathsf{dk}}$

Correction

$$\hat{m} = c_2 \cdot c_1^{-dk} = ek^y \cdot m \cdot (g^y)^{-dk} = (g^{dk})^y \cdot m \cdot (g^y)^{-dk} = m$$

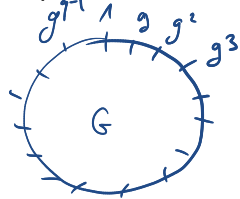
Group multiplication for encryption

Lemma

Let $G = \langle g \rangle$ be a cyclic group of order q and $z \twoheadleftarrow \{0, \ldots, q-1\}$ (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 means: $\forall h \in G$, $\Pr_z[g^z = h] = 1/q$.

There is a bijection $\{0, \ldots, q-1\} \longrightarrow G$

$z \longmapsto g^z$

Therefore $\Pr[g^z = h] = \Pr[z = \log_g h] = 1/q$.

(ii) We want to prove: if $h \leftarrow G$ is uniform, then $\forall m \in G$, $h \cdot m$ is uniform

Same proof because $h \longmapsto h \cdot m$ is a bijection in G (inverse $h \longmapsto h \cdot m^{-1}$)

Security proof

Theorem

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

Proof

DDH Game

- C_{DDH} playing exp. $\lambda \in \{0,1\}$

($\lambda=0$: $a, b, c \leftarrow \{0, \dots, q-1\}$

$\lambda=1$: $a, b \leftarrow \{0, \dots, q-1\}$, $c \leftarrow a \cdot b$

Sends g^a, g^b, g^c

- A_{DDH}: receives g^a, g^b, g^c and computes $\hat{\lambda}$

$$\mathrm{Adv}^{DDH}_{G}(A_{DDH}) = \left| Pr[A_{DDH} \rightarrow 1 \mid \lambda=1] - Pr[A_{DDH} \rightarrow \lambda \mid \lambda=0] \right|$$

IND-CPA Game

- C_{CPA} playing exp. $\mu \in \{0,1\}$

Creates (g^x, x) and sends g^x

- for $i=0$ to $q-1$:

A_{CPA} creates $m_{i,0}, m_{i,1}$

C_{CPA} computes $(g^y, g^{xy} \cdot m_{i,\mu})$

- A_{CPA} computes $\hat{\mu}$

$$\mathrm{Adv}^{\text{IND-CPA}}_{\text{EG}(G)}(A_{CPA}) = \left| Pr[\hat{\mu}=1 \mid \mu=1] - Pr[\hat{\mu}=1 \mid \mu=0] \right|$$

Security proof

Theorem

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

Proof

Idea

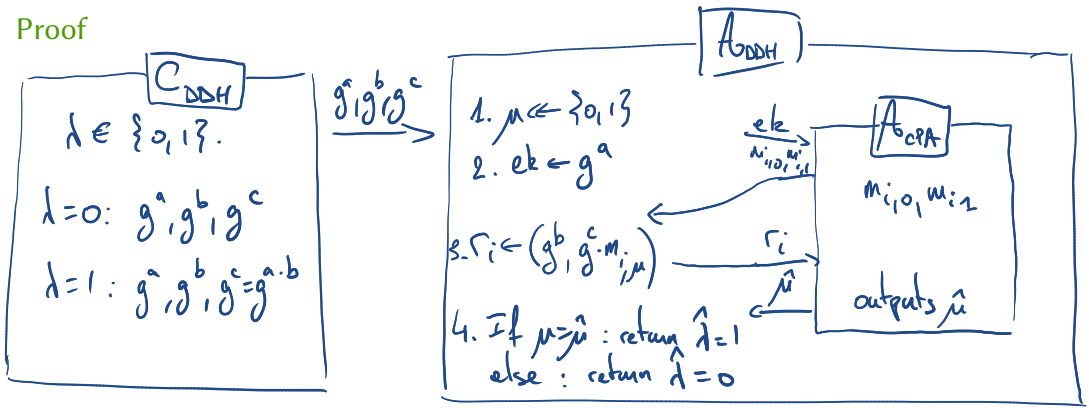
- Start from A_{CPA} that has advantage $\alpha = \mathrm{Adv}^{\text{IND-CPA}}_{\text{EG}(G)}(q, t)$
- Build A_{DDH} from A_{CPA} and prove that
  - $\mathrm{Adv}^{\text{DDH}}_{G}(A_{DDH}) \geq \frac{1}{2}\alpha$.
  - A_{DDH} has run time $\Theta(t)$
- Conclude: $\mathrm{Adv}^{\text{DDH}}_{G}(t) \geq \mathrm{Adv}^{\text{DDH}}_{G}(A_{DDH}) \geq \frac{\alpha}{2} = \frac{1}{2}\mathrm{Adv}^{\text{IND-CPA}}_{\text{EG}(G)}(q, t)$

Security proof

Theorem

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

Proof



Security proof

Theorem

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

Proof

Analysis

- $\beta = \mathrm{Adv}^{\text{DDH}}_{G}(A_{\text{DDH}}) = \left| \Pr\left[\hat{d}=1 \mid d=1\right] - \Pr\left[\hat{d}=1 \mid d=0\right] \right| = \left| \Pr\left[\hat{\mu}=\mu \mid d=1\right] - \Pr\left[\hat{\mu}=\mu \mid d=0\right] \right|$
- $\Pr\left[\hat{\mu}=\mu \mid d=0\right] = 1/2$ since $g^{c} \cdot m_{i,\mu}$ is uniform in G and $\mu \leftarrow \{0,1\}$.
- $\Pr\left[\hat{\mu}=\mu \mid d=1\right] = \Pr\left[\hat{\mu}=1 \mid d=1, \mu=1\right]\Pr[\mu=1] + \Pr\left[\hat{\mu}=0 \mid d=1, \mu=0\right]\Pr[\mu=0]$
  $= \frac{1}{2}\left(\Pr\left[\hat{\mu}=1 \mid d=1, \mu=1\right] + 1 - \Pr\left[\hat{\mu}=1 \mid d=1, \mu=0\right]\right)$
  $= \frac{1}{2}\left(1 \pm \alpha\right)$
- $\beta = \left| \frac{1}{2}(1 \pm \alpha) - 1/2 \right| = \left| \pm \alpha/2 \right| = \alpha/2$.

Additional remarks

Choice of the group G

- The order q must be prime, for DDH
- Several choices (subgroup of $(\mathbb{Z}/p\mathbb{Z})^\times, \ldots$)
  - 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 $\{0,1\}^\ell$ *for some G*
- Solution 2: ElGamal-based KEM + key derivation function

CCA (in)security

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

Contents

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 encryption key of the receiver *encapsulated key*

Receiver decrypts first the encapsulated key with its decryption key $\rightarrow k$
decrypts c using $k \rightarrow m$

The KEM/DEM paradigm

Definition

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

$\mathsf{Gen}_n()$: produces a pair $(\mathsf{ek}, \mathsf{dk})$

$\mathsf{Encaps}_{\mathsf{ek}}()$: produces a pair (c, k)

$\mathsf{Decaps}_{\mathsf{dk}}(c)$: returns k

Usage

To send m using encryption key ek:

1. $(c, k) \leftarrow \mathsf{Encaps}_{\mathsf{ek}}()$ — *key encapsulation*
2. $c' \leftarrow \mathsf{Enc}_k(m)$ (with symmetric encryption) — *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 $(\mathsf{Enc}, \mathsf{Dec})$

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

$\mathsf{Decaps}_{\mathsf{dk}}(c)$:
1. $k \leftarrow \mathsf{Dec}_{\mathsf{dk}}(c)$

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 ek, produce c and k such that k can be recovered from dk and c
  - We don't need: c to be an actual encryption of k using ek

DDH-based KEM

Construction

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

Gen():
1. $x \twoheadleftarrow \{0, \ldots, q-1\}$
2. $h \leftarrow g^x$
3. $H \leftarrow$ some hash function from G to $\{0,1\}^\ell$
4. return $\mathsf{ek} = (h, H)$ and $\mathsf{dk} = (x, H)$

$\mathrm{Encaps}_{\mathsf{ek}}()$:
1. $y \twoheadleftarrow \{0, \ldots, q-1\}$
2. return $c \leftarrow g^y$ and $k \leftarrow H(h^y)$

$\mathrm{Decaps}_{\mathsf{dk}}(c)$:
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 $\rightarrow$ *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, …