Cryptography

Corso di Laurea Magistrale in Informatica

Public-Key Encryption

Ugo Dal Lago

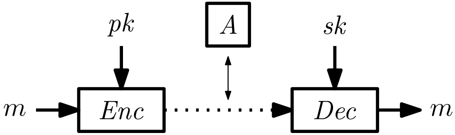
Academic Year 2021-2022

Ciphers in an Asymmetrical Framework

- In asymmetric cryptography, anyone who wants to *receive* messages generates not a key but a *pair* of keys (pk, sk) where:
  - pk is a *public key*, used by the sender when encoding messages and must reach as many users as possible (through authenticated channels, even if not private).
  - sk is a *private key*.

Ciphers in an Asymmetrical Framework

- In asymmetric cryptography, anyone who wants to *receive* messages generates not a key but a *pair* of keys (pk, sk) where:
  - pk is a *public key*, used by the sender when encoding messages and must reach as many users as possible (through authenticated channels, even if not private).
  - sk is a *private key*.
- The framework then becomes the following one:



Symmetric Key vs. Asymmetric Key

- In asymmetric cryptography:
  - Only *one part* of the key is kept secret, while the other is made public.
  - (Portions of) different keys are used in the encryption and decryption phases.

Symmetric Key vs. Asymmetric Key

- In asymmetric cryptography:
  - Only *one part* of the key is kept secret, while the other is made public.
  - (Portions of) different keys are used in the encryption and decryption phases.
- **Advantages** of the Asymmetric Key:
  - It is no longer necessary to distribute keys on *private* channels.
  - Each user must manage the secrecy of *only one* key.

Symmetric Key vs. Asymmetric Key

- In asymmetric cryptography:
  - Only *one part* of the key is kept secret, while the other is made public.
  - (Portions of) different keys are used in the encryption and decryption phases.
- **Advantages** of the Asymmetric Key:
  - It is no longer necessary to distribute keys on *private* channels.
  - Each user must manage the secrecy of *only one* key.
- **Disadvantages** of the Asymmetric Key:
  - The performance of asymmetric-key schemes is usually orders of magnitude lower than that of symmetric-key ones.
  - Public keys must be distributed over *authenticated* channels, without which a very simple attack is possible.

Public-Key Encryption Scheme

- The definition of the encryption scheme $\Pi = (Gen, Enc, Dec)$ needs to be suitably modified:
  - Gen takes a string in the form 1^n as input and outputs a pair of keys (pk, sk), such that that $|pk|, |sk| \geq n$ and such that n can be inferred by pk or sk.
  - The Enc algorithm takes as input a message m and a public key pk and outputs a ciphertext.
  - The algorithm Dec can be probabilistic, it takes as input a ciphertext c and a secret key sk and outputs either a message or a special symbol $\perp$.
- Let us assume that the scheme is **correct**, this time in the *probabilistic* sense: there must exist a negligible function ε such that for every pair (pk, sk) produced by $Gen(1^n)$ and for every n,

$$Pr(Dec_{sk}(Enc_{pk}(m)) \neq m) \leq \varepsilon(n)$$

- Often, Enc_k is defined only for messages of length equal to n, or over the whole space $\{0, 1\}^*$.

Security of a Public-Key Encryption Scheme

- The notion of experiment should be modified:

$\mathsf{PubK}^{eav}_{A,\Pi}(n)$:
$(pk, sk) \leftarrow Gen(1^n)$;
$(m_0, m_1) \leftarrow A(1^n, pk)$;
if $|m_0| \neq |m_1|$ **then**
  Result: 0
$b \leftarrow \{0,1\}$; $c \leftarrow Enc(k, m_b)$;
$b^* \leftarrow A(c)$;
Result: $\neg(b \oplus b^*)$

Security of a Public-Key Encryption Scheme

- The notion of experiment should be modified:

$\mathsf{PubK}^{eav}_{A,\Pi}(n)$:
$(pk, sk) \leftarrow Gen(1^n)$;
$(m_0, m_1) \leftarrow A(1^n, pk)$;
if $|m_0| \neq |m_1|$ **then**
  Result: 0
$b \leftarrow \{0,1\}$; $c \leftarrow Enc(k, m_b)$;
$b^* \leftarrow A(c)$;
Result: $\neg(b \oplus b^*)$

Definition

A public key encryption scheme Π is said to be *secure against passive attacks* iff for every adversary PPT A there exists a function $\varepsilon \in \mathcal{NGL}$ such that

$$Pr(\mathsf{PubK}^{eav}_{\Pi,A}(n) = 1) = \frac{1}{2} + \varepsilon(n)$$

Comments on the Definition

- The definition of security we have just given is imperceptibly different from that seen in a symmetrical context: A obviously has also access to pk.
- This small difference has *important* consequences:
  1. The fact that A has access to pk implies that A can encrypt any message, even without access to oracles.
  2. Given pk and $c = Enc_{pk}(m)$, it is always possible to reconstruct m having arbitrary time available.

Theorem

If Π is secure against passive attacks, then it is CPA-secure.

Comments on the Definition

- The definition of security we have just given is imperceptibly different from that seen in a symmetrical context: A obviously has also access to pk.
- This small difference has *important* consequences:
  1. The fact that A has access to pk implies that A can encrypt any message, even without access to oracles.
  2. Given pk and $c = Enc_{pk}(m)$, it is always possible to reconstruct m having arbitrary time available.

Theorem

If Π is secure against passive attacks, then it is CPA-secure.

Theorem

There are no asymmetric ciphers that are secure in a perfect sense.

Insecurity of Deterministic Encryption

- We know that every *passive* adversary, having access to pk, is actually also *active*.
  - Therefore, many properties that we have seen for the symmetrical case and for CPA attacks hold also in this case.

Insecurity of Deterministic Encryption

- We know that every *passive* adversary, having access to pk, is actually also *active*.
  - Therefore, many properties that we have seen for the symmetrical case and for CPA attacks hold also in this case.

Theorem

No public key scheme in which Enc is deterministic can be secure with respect to PubK^{eav}.

Insecurity of Deterministic Encryption

- We know that every *passive* adversary, having access to pk, is actually also *active*.
  - Therefore, many properties that we have seen for the symmetrical case and for CPA attacks hold also in this case.

Theorem

No public key scheme in which Enc is deterministic can be secure with respect to PubK^{eav}.

- Historically, a large number of public-key encryption schemes are such that *Enc* is deterministic.
  - This had (and still has) disastrous consequences.

On Multiple Encryptions

- Similarly to what we have seen in the symmetrical case, we can talk about security for *multiple encryptions*.
  - We just define a new experiment PubK^{mult} in which the adversary outputs not a pair of messages (m_0, m_1) but a pair of tuple of messages $(\mathbf{m}_0, \mathbf{m}_1)$ where $\mathbf{m}_0 = (m_0^1, \ldots, m_0^t)$, $\mathbf{m}_1 = (m_1^1, \ldots, m_1^t)$, and $|m_0^j| = |m_1^j|$.

On Multiple Encryptions

- Similarly to what we have seen in the symmetrical case, we can talk about security for *multiple encryptions*.
  - We just define a new experiment PubK^{mult} in which the adversary outputs not a pair of messages (m_0, m_1) but a pair of tuple of messages $(\mathbf{m}_0, \mathbf{m}_1)$ where $\mathbf{m}_0 = (m_0^1, \ldots, m_0^t)$, $\mathbf{m}_1 = (m_1^1, \ldots, m_1^t)$, and $|m_0^j| = |m_1^j|$.
- As usual, a public-key encryption scheme Π is said to be secure with respect to multiple encodings iff for every PPT A there exists ε with

$$Pr(\mathsf{PubK}_{\Pi,A}^{mult}(n) = 1) = \frac{1}{2} + \varepsilon(n)$$

On Multiple Encryptions

- Similarly to what we have seen in the symmetrical case, we can talk about security for *multiple encryptions*.
  - We just define a new experiment PubK^{mult} in which the adversary outputs not a pair of messages (m_0, m_1) but a pair of tuple of messages $(\mathbf{m}_0, \mathbf{m}_1)$ where $\mathbf{m}_0 = (m_0^1, \ldots, m_0^t)$, $\mathbf{m}_1 = (m_1^1, \ldots, m_1^t)$, and $|m_0^j| = |m_1^j|$.
- As usual, a public-key encryption scheme Π is said to be secure with respect to multiple encodings iff for every PPT A there exists ε with

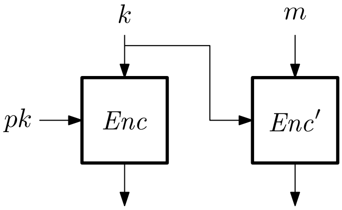
$$Pr(\mathsf{PubK}_{\Pi,A}^{mult}(n) = 1) = \frac{1}{2} + \varepsilon(n)$$

Theorem

If an encryption scheme Π is secure with respect to PubK^{eav}, then it is secure with respect to PubK^{mult}.

Hybrid Encryption

- We have already mentioned that public-key encryption schemes are *less performing* than private-key ones.
- With hybrid encryption we simply try to put together *the positive aspects* of public-key and private-key encryptions.
- Given $\Pi = (Gen, Enc, Dec)$ with a public key and $\Pi' = (Gen', Enc', Dec')$ with a private key, we can construct Π^{Hy} in which the encryption is more or less as follows:



Hybrid Encryption

- When defining the hybrid encryption, we will make the assumption that Gen' returns a random string in $\{0,1\}^n$ and Π includes $\{0,1\}^n$ in the message space.
- Formally, the scheme Π^{Hy} is defined from Π and Π', as follows :

$Gen^{Hy}(1^n)$:
Result: $Gen(1^n)$

$Enc^{Hy}(pk, m)$:
$k \leftarrow \{0,1\}^n$;
$c \leftarrow Enc_{pk}(k)$;
$d \leftarrow Enc_k(m)$;
Result: (c, d)

$Dec^{Hy}(sk, (c, d))$:
$k \leftarrow Dec_{sk}(c)$;
$m \leftarrow Dec_k(d)$;
Result: m

Theorem

If Π is CPA-secure and Π' has indistinguishable encryptions, then Π^{Hy} is secure.

Hybrid Encryption: Why?

1. **Encryption Time.**
   - Suppose that the encryption of the key takes time α and that the encryption of the message takes time β for each bit.
   - Therefore, the average time taken by Enc^{Hy} for each bit will be , for messages t long, equal to $TIME(t) = (\alpha + \beta t)/t$.
   - Note that
   $$\lim_{t \to \infty} \frac{\alpha + \beta t}{t} = \beta$$

Hybrid Encryption: Why?

1. **Encryption Time.**
   - Suppose that the encryption of the key takes time α and that the encryption of the message takes time β for each bit.
   - Therefore, the average time taken by Enc^{Hy} for each bit will be , for messages t long, equal to $TIME(t) = (\alpha + \beta t)/t$.
   - Note that
   $$\lim_{t \to \infty} \frac{\alpha + \beta t}{t} = \beta$$
2. **Ciphertexts' Length**
   - A very similar reasoning to that made for the encryption time can be made for the length of the ciphertexts.
   - As $|m|$ increases, the quantity $|c|$ stays constant, while there are private-key encryption schemes such that $|d| = |m| + n$.
   - Therefore, as $|m|$ increases, the length of (c, d) is linear.

The RSA Encryption Scheme

- We have considered the security of public-key encryption schemes, giving interesting results.
- However, we have not dealt with any concrete encryption scheme.
  - Hybrid Encryption cannot be used in this sense, as it requires the existence of a public-key encryption scheme to start from.
- We will first present a scheme call **Textbook RSA**:

$Gen(1^n)$:
$(N, e, d) \leftarrow \mathsf{GenRSA}(1^n)$;
Result: $((N, e), (N, d))$

$Enc(((N, e), m)$:
$c \leftarrow m^e \mod N$;
Result: c

$Dec((N, d), c)$:
$m \leftarrow c^d \mod N$;
Result: m

- The correctness of the scheme follows from the fact that if the pair $((N, e), (N, d))$ is obtained from Gen, then f_d is the inverse of f_e.

Textbook RSA: Problems

- First of all, it should be noted that Textbook RSA is **insecure** with respect to our definition.
  - To realise this, it is sufficient to observe that Enc is deterministic!
  - However, a very weak security notion holds: given the public key (N, e) and $c = m^e \mod N$, it is not possible to determine the message m in its entirety, at least when the RSA Assumption holds.

Textbook RSA: Problems

- First of all, it should be noted that Textbook RSA is **insecure** with respect to our definition.
  - To realise this, it is sufficient to observe that *Enc* is deterministic!
  - However, a very weak security notion holds: given the public key (N, e) and $c = m^e \mod N$, it is not possible to determine the message m in its entirety, at least when the RSA Assumption holds.
- From a theoretical point of view, it would be necessary to guarantee that $m \in \mathbb{Z}_N^*$. Also when $m \in \mathbb{Z}_N$, encryption and decryption work.
  - It can also be shown that $\phi(N)/N$, considered as a function of n, is in the form $1 - \varepsilon(n)$.

Textbook RSA: Problems

- First of all, it should be noted that Textbook RSA is **insecure** with respect to our definition.
  - To realise this, it is sufficient to observe that *Enc* is deterministic!
  - However, a very weak security notion holds: given the public key (N, e) and $c = m^e \mod N$, it is not possible to determine the message m in its entirety, at least when the RSA Assumption holds.
- From a theoretical point of view, it would be necessary to guarantee that $m \in \mathbb{Z}_N^*$. Also when $m \in \mathbb{Z}_N$, encryption and decryption work.
  - It can also be shown that $\phi(N)/N$, considered as a function of n, is in the form $1 - \varepsilon(n)$.
- In the literature, there are many examples of attacks against Textbook RSA.
  - If, as is often the case, e is chosen as a *fixed* and very *small* value (e.g. 3), then m is the cube root of m (modulo N), which can be easily computed.
  - The complexity of the brute force attack can be reduced from N to $\sqrt{N}$.

Padded RSA

- Is there any way to make RSA secure?

Padded RSA

- Is there any way to make RSA secure?
- The answer is yes. Consider the following diagram, called **Padded RSA**:

Gen(1^n):
$(N, e, d) \leftarrow$ $\mathsf{GenRSA}(1^n)$;
Result: $((N, e), (N, d))$

Enc$(((N, e), m)$:
$r \leftarrow \{0,1\}^{|N|-\ell(n)-1}$;
$c \leftarrow (r||m)^e \bmod N$;
Result: c

Dec$((N, d), c)$:
$m \leftarrow LSB(c^d \bmod N, \ell(n))$;
Result: m

where ℓ is a function such that $|m| \leq \ell(n) \leq 2n - 2$ and LSB returns the least significant bits.

- It is necessary to choose $\ell(n)$ sufficiently small, less than linear.

Theorem

If the RSA Assumption holds with respect to GenRSA *and if* $\ell(n) = O(\lg n)$*, then Padded RSA is secure with respect to passive attacks.*

The Elgamal Encryption Scheme

- In addition to RSA, there is another secure encryption scheme based on the assumptions we talked about few lessons ago.
- In particular, there is one encryption scheme, due to Elgamal, which can be proved secure from the DDH Assumption.
- The observation to start from is that, when fixed two elements $m, c \in \mathbb{G}$ of a finite group, the probability that a random element $k \in \mathbb{G}$ is such that $m \cdot k = g$ is equal to $\frac{1}{|\mathbb{G}|}$.
  - All this can be easily proved by observing that

$$Pr(m \cdot k = c) = Pr(k = m^{-1} \cdot c) = \frac{1}{|\mathbb{G}|}$$

- In other words, we are in a situation similar to the one we saw in OTP.

The Elgamal Encryption Scheme

- Formally, the Elgamal scheme is defined as follows:

$Gen(1^n)$:
$(\mathbb{G}, q, g) \leftarrow \mathsf{GenCG}(1^n)$;
$x \leftarrow \mathbb{Z}_q$;
$sk \leftarrow (\mathbb{G}, q, g, x)$;
$pk \leftarrow (\mathbb{G}, q, g, g^x)$;
Result: (sk, pk)

$Enc((\mathbb{G}, q, g, h), m)$:
$y \leftarrow \mathbb{Z}_q$;
Result: $(g^y, h^y \cdot m)$

$Dec((\mathbb{G}, q, g, x), (c, d))$:
Result: d/c_1^x

- The correctness of the scheme is easy to prove:

$$\frac{d}{c_1^x} = \frac{h^y \cdot m}{g^{yx}} = \frac{(g^x)^y \cdot m}{g^{xy}} = m$$

Theorem

If Assumption DDH holds with respect to GenCG, *then the Elgamal scheme is secure.*