One ever wanted to exchange information securely
With the all-digital world, security needs are even stronger…
In your pocket
But also at home

3 Historical Goals
- Confidentiality: The content of a message is concealed
- Authenticity: The author of a message is well identified
- Integrity: Messages have not been altered

between a sender and a recipient, against an adversary.

Also within groups, with insider adversaries

Cannot address availability, but should not affect it!
First Encryption Mechanisms

**The goal of encryption is to hide a message**

Scytale
Permutation

Alberti’s disk
Mono-alphabetical Substitution

Wheel – M 94 (CSP 488)
Poly-alphabetical Substitution

Substitutions and permutations **Security** relies on the secrecy of the mechanism

⇒ How to widely use them?

Use of a (Secret) Key

A shared information *(secret key)* between the sender and the receiver parameterizes the public mechanism

**Enigma:** choice of the connectors and the rotors

Security looks better: but broken (Alan Turing *et al.*)

⇒ Security analysis is required

Modern Cryptography

**Secret Key Encryption**

One *secret key* only shared by Alice and Bob:
this is a *common* parameter for both E and D


**Public Key Cryptography**

- Bob’s public key is used by Alice as a parameter to E
- Bob’s private key is used by Bob as a parameter to D

DES and AES

Still substitutions and permutations, but considering various classes of attacks (statistic)

**DES: Data Encryption Standard**

“Broken” in 1998 by brute force: too short keys (56 bits!)

⇒ No better attack granted a safe design!

New standard since 2001: Advanced Encryption Standard

Longer keys: from 128 to 256 bits

Criteria: Security arguments against many attacks

What does security mean?
Practical Secrecy

**Perfect Secrecy vs. Practical Secrecy**
- No information about the plaintext $m$ can be extracted from the ciphertext $c$, even for a powerful adversary (unlimited time and/or unlimited power): perfect secrecy $\Rightarrow$ information theory
- In practice: adversaries are limited in time/power $\Rightarrow$ complexity theory

We thus model all the players (the legitimate ones and the adversary) as Probabilistic Polynomial Time Turing Machines:

*computers that run programs*

Provable Security

**Symmetric Cryptography**
- The secrecy of the key guarantees the secrecy of communications

**Asymmetric Cryptography**
- The secrecy of the private key guarantees the secrecy of communications

What is a Secure Cryptographic Scheme?

- What does security mean?
  $\rightarrow$ Security notions have to be formally defined
- How to guarantee above security claims for concrete schemes?
  $\rightarrow$ Provable security

General Method

**Computational Security Proofs**

To prove the security of a cryptographic scheme, one needs
- a formal security model (security notions)
- a reduction: if one (Adversary) can break the security notions, then one (Simulator + Adversary) can break a hard problem
- acceptable computational assumptions (hard problems)

Proof by contradiction
**Integer Factoring**

**Records**

Given \( n = pq \) → Find \( p \) and \( q \)

<table>
<thead>
<tr>
<th>Digits</th>
<th>Date</th>
<th>Bit-Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>130</td>
<td>April 1996</td>
<td>431 bits</td>
</tr>
<tr>
<td>140</td>
<td>February 1999</td>
<td>465 bits</td>
</tr>
<tr>
<td>155</td>
<td>August 1999</td>
<td>512 bits</td>
</tr>
<tr>
<td>160</td>
<td>April 2003</td>
<td>531 bits</td>
</tr>
<tr>
<td>200</td>
<td>May 2005</td>
<td>664 bits</td>
</tr>
<tr>
<td>232</td>
<td>December 2009</td>
<td>768 bits</td>
</tr>
</tbody>
</table>

**Complexity**

- 768 bits → \( 2^{64} \) op.
- 1024 bits → \( 2^{80} \) op.
- 2048 bits → \( 2^{112} \) op.
- 3072 bits → \( 2^{128} \) op.
- 7680 bits → \( 2^{192} \) op.
- 15360 bits → \( 2^{256} \) op.

**Reduction**

<table>
<thead>
<tr>
<th>Modulus Bit-length</th>
<th>Adversary Complexity</th>
<th>Algorithm Complexity</th>
<th>Best Known Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>( k = 1024 )</td>
<td>( t &lt; 2^{80} )</td>
<td>( T &lt; 2^{102} )</td>
<td>( 2^{80} ) ×</td>
</tr>
<tr>
<td>( k = 2048 )</td>
<td>( t &lt; 2^{80} )</td>
<td>( T &lt; 2^{113} )</td>
<td>( 2^{112} ) ×</td>
</tr>
<tr>
<td>( k = 3072 )</td>
<td>( t &lt; 2^{80} )</td>
<td>( T &lt; 2^{115} )</td>
<td>( 2^{128} ) ✔</td>
</tr>
</tbody>
</table>

- Lossy reduction: \( T = k^3 \times t \)
- Tight reduction: \( T \approx t \)

With \( k = 1024 \) and \( t < 2^{80} \), one gets \( T < 2^{80} \)

---

**One-Way Functions**

- \( F(1^k) \) generates a function \( f : X \rightarrow Y \)
- From \( x \in X \), it is easy to compute \( y = f(x) \)
- Given \( y \in Y \), it is hard to find \( x \in X \) such that \( y = f(x) \)

**RSA Problem** [Rivest-Shamir-Adleman 1978]

- Given \( n = pq \), \( e \) and \( y \in \mathbb{Z}_n^* \)
- Find \( x \) such that \( y = x^e \mod n \)

This problem is hard without the prime factors \( p \) and \( q \)
It becomes easy with them: if \( d = e^{-1} \mod \varphi(n) \), then \( x = y^d \mod n \)

This problem is assumed as hard as integer factoring:
the prime factors are a trapdoor to find solutions
\( \Rightarrow \) trapdoor one-way permutation

---

**Signature**

**Goal:** Authentication of the sender

\[ \begin{align*}
S \quad & \text{Sign message } m \\
\sigma \quad & \text{Sig} \\
V \quad & \text{Verify} \\
\end{align*} \]
**EUF – NMA: Security Game**

![Diagram](image)

\[ \mathsf{Succ}_{\mathcal{G}}(\mathcal{A}) = \Pr[(k_s, k_v) \leftarrow \mathcal{G}(); (m, \sigma) \leftarrow \mathcal{A}(k_v) : \forall (k_v, m, \sigma) = 1] \]

should be negligible.

\[ \mathcal{A} \text{ knows the public key only} \implies \text{No-Message Attack (NMA)} \]

**EUF – NMA**

**One-Way Function**

- \( \mathcal{G}(1^k): f \overset{R}{\leftarrow} \mathcal{F}(1^k) \) and \( x \overset{R}{\leftarrow} X \), set \( y = f(x) \), \( k_s = x \) and \( k_v = (f, y) \)
- \( S(x, m) = k_s = x \)
- \( \forall ((f, y), m, x') \) checks whether \( f(x') = y \)

Under the one-wayness of \( \mathcal{F} \), \( \mathsf{Succ}^{\text{euf–nma}}(\mathcal{A}) \) is small.

But given one signature, one can “sign” any other message! Signatures are public! \( \implies \text{Known-Message Attacks (KMA)} \)

The adversary has access to a list of messages-signatures.

**EUF – KMA**

**One-Way Functions**

- \( \mathcal{G}(1^k): f \overset{R}{\leftarrow} \mathcal{F}(1^k) \), and \( \vec{x} = (x_{1,0}, x_{1,1}, \ldots, x_{k,0}, x_{k,1}) \overset{R}{\leftarrow} X^{2k} \), \( y_{i,j} = f(x_{i,j}) \) for \( i = 1, \ldots, k \) and \( j = 0, 1 \), \( k_s = \vec{x} \) and \( k_v = (f, \vec{y}) \)
- \( S(\vec{x}, m) = (x_{i,m})_{i=1,\ldots,k} \)
- \( \forall((f, \vec{y}), m, (x'_i)) \) checks whether \( f(x'_i) = y_{i,m} \) for \( i = 1, \ldots, k \)

Under the one-wayness of \( \mathcal{F} \), \( \mathsf{Succ}^{\text{euf–nma}}(\mathcal{A}) \) is small.

With the signature of \( m = 0^k \), I cannot forge any other signature.

With the signatures of \( m = 0^k \) and \( m' = 1^k \), I learn \( \vec{x} \): the secret key Messages can be under the control of the adversary! \( \implies \text{Chosen-Message Attacks (CMA)} \)

**EUF – CMA**

**One-Way Functions**

- \( \mathcal{G}(1^k) \):
- \( \mathcal{G}(1^k): f \overset{R}{\leftarrow} \mathcal{F}(1^k) \), and \( \vec{x} = (x_{1,0}, x_{1,1}, \ldots, x_{k,0}, x_{k,1}) \overset{R}{\leftarrow} X^{2k} \), \( y_{i,j} = f(x_{i,j}) \) for \( i = 1, \ldots, k \) and \( j = 0, 1 \), \( k_s = \vec{x} \) and \( k_v = (f, \vec{y}) \)
- \( S(\vec{x}, m) = (x_{i,m})_{i=1,\ldots,k} \)
- \( \forall((f, \vec{y}), m, (x'_i)) \) checks whether \( f(x'_i) = y_{i,m} \) for \( i = 1, \ldots, k \)

Under the one-wayness of \( \mathcal{F} \), \( \mathsf{Succ}^{\text{euf–nma}}(\mathcal{A}) \) is small.

With the signature of \( m = 0^k \), I cannot forge any other signature.

With the signatures of \( m = 0^k \) and \( m' = 1^k \), I learn \( \vec{x} \): the secret key Messages can be under the control of the adversary! \( \implies \text{Chosen-Message Attacks (CMA)} \)

The adversary has access to any signature of its choice: Chosen-Message Attacks (oracle access):

\[ \mathsf{Succ}^{\text{euf–cma}}(\mathcal{A}) = \Pr[(k_s, k_v) \leftarrow \mathcal{G}(); (m, \sigma) \leftarrow \mathcal{A}(k_v)(k_v) : \forall i, m \neq m_i \land \forall (k_v, m, \sigma) = 1] \]
The RSA Signature

[David Pointcheval – ENS/CNRS/INRIA Collège de France 22/40]

Full-Domain Hash Signature

[Bellare-Rogaway – Eurocrypt ’96]

Full-Domain Hash RSA Signature

The FDH-RSA signature scheme is defined by
- \( \mathcal{G}(1^k) \): p and q, two random primes, and an exponent \( \nu \)
- \( n = pq \), \( k_s \leftarrow s = \nu^{-1} \mod \varphi(n) \) and \( k_v \leftarrow (n, \nu) \)
- \( \mathcal{H} \) is a hash function onto \( \mathbb{Z}_n^* \)
- \( \mathcal{S}(k_s, m) \): the signature is \( \sigma = \mathcal{H}(m)^s \mod n \)
- \( \mathcal{V}(k_v, m, \sigma) \) checks whether \( \mathcal{H}(m) = \sigma^\nu \mod n \)

Theorem (Security of the FDH-RSA)

The FDH-RSA is EUF – CMA under appropriate assumptions on \( \mathcal{H} \),
and assuming the RSA problem is hard

Improved Security

[David Pointcheval – ENS/CNRS/INRIA Collège de France 23/40]
RSA-PSS (PKCS #1 v2.1) [Bellare-Rogaway – Eurocrypt ‘96]

- **m** is the message to encrypt
- **r** is the additional randomness to make encryption probabilistic

After the transformation, \( w \| s \| t \) goes in the plain RSA

**Theorem (EUF-CMA Security)** [Bellare-Rogaway – Eurocrypt ‘96]

RSA-PSS is EUF-CMA secure under the RSA assumption

Security reduction between EUF – CMA and the RSA assumption:

\[ T \approx t \]

\[ \Rightarrow 1024\text{-bit RSA moduli provide } 2^{80} \text{ security} \]

Public-Key Encryption

Goal: Privacy/Secrecy of the plaintext

OW – CPA: Security Game

\[ m^* \text{ random} \]
\[ r^* \text{ random} \]

\[ m^* \overset{?}{=} m \]

OW – CPA: Is it Enough?

**The \( RSA \) Encryption** [Rivest-Shamir-Adleman 1978]

- **\( G(1^k) \)**: \( p \) and \( q \), two random primes, and an exponent \( e \):
  \[ n = pq, \ sk \leftarrow d = e^{-1} \mod \varphi(n) \text{ and } pk \leftarrow (n, e) \]

- **\( \mathcal{E}(pk, m) = c = m^e \mod n \)**;
  \( \mathcal{D}(sk, c) = m = c^d \mod n \)

\( RSA \) encryption is OW – CPA, under the RSA assumption

\[ k_e \]
\[ \mathcal{E} \]
\[ m \]
\[ D \]
\[ k_d \]

OW – CPA Too Weak

- **\( \mathcal{E}'(pk, m = m_1 \| m_2) = \mathcal{E}(pk, m_1) \| m_2 = c_1 \| c_2 \)**
- **\( \mathcal{D}'(sk, c_1 \| c_2) = m_1 = \mathcal{D}(sk, c_1), m_2 = c_2, \) output \( m = m_1 \| m_2 \)**

If \( (G, \mathcal{E}, \mathcal{D}) \) is OW – CPA: then \( (G', \mathcal{E}', \mathcal{D}') \) is OW – CPA too

But this is clearly not enough: **half or more of the message leaks!**


OW – CPA: Is it Enough?

For a “yes/no” answer or “sell/buy” order, one bit of information may be enough for the adversary! How to model that no bit of information leaks?

**Perfect Secrecy vs. Computational Secrecy**

- **Perfect secrecy**: the distribution of the ciphertext is perfectly independent of the plaintext
- **Computational secrecy**: the distribution of the ciphertext is computationally independent of the plaintext

Idea: No adversary can distinguish a ciphertext of $m_0$ from a ciphertext of $m_1$. Probabilistic encryption is required!

ElGamal Encryption

The ElGamal Encryption ($\mathcal{E}_G$)

- $G(1^k): G = \langle g \rangle$ of order $q$, $sk = x \overset{R}{\leftarrow} \mathbb{Z}_q$ and $pk = y = g^x$
- $\mathcal{E}(pk, m, r) = (c_1 = g^r, c_2 = y^r m)$
- $D(sk, (c_1, c_2)) = c_2 / c_1^x$

The ElGamal encryption is IND-CPA, under the DDH assumption

**Decisional Diffie-Hellman Problem**

For $G = \langle g \rangle$ of order $q$, and $x, y \overset{R}{\leftarrow} \mathbb{Z}_q$,

- Given $X = g^x, Y = g^y$ and $Z = g^z$, for either $z \overset{R}{\leftarrow} \mathbb{Z}_q$ or $z = xy$
- Decide whether $z = xy$

This problem is assumed hard to decide in appropriate groups $G$!

ElGamal is IND-CPA: Proof

Let $\mathcal{A}$ be an adversary against $\mathcal{E}_G$: $\mathcal{B}$ is an adversary against DDH: let us be given a DDH instance $(X = g^x, Y = g^y, Z = g^z)$

- $\mathcal{A}$ gets $pk \leftarrow X$ from $\mathcal{B}$, and outputs $(m_0, m_1)$
- $\mathcal{B}$ sets $c_1 \leftarrow Y$
- $\mathcal{B}$ chooses $b \overset{R}{\leftarrow} \{0, 1\}$, sets $c_2 \leftarrow Z \times m_b$, and sends $c = (c_1, c_2)$
- $\mathcal{B}$ receives $b'$ from $\mathcal{A}$ and outputs $d = (b' = b)$

2. $\Pr[b' = b] - 1$
   - $\text{Adv}_{\mathcal{E}_G}^{\text{IND-CPA}}(\mathcal{A})$, if $z = xy$
   - $0$, if $z \overset{R}{\leftarrow} \mathbb{Z}_q$
ElGamal is IND – CPA: Proof

As a consequence,

- \(2 \times \Pr[b' = b|z = xy] - 1 = \text{Adv}_{\text{IND-CPA}}^\text{G}(A)\)
- \(2 \times \Pr[b' = b|z \leftarrow \mathbb{Z}_q] - 1 = 0\)

If one subtracts the two lines:

\[
\text{Adv}_{\text{IND-CPA}}^\text{G}(A) = 2 \times \left( \Pr[d = 1|z = xy] - \Pr[d = 1|z \leftarrow \mathbb{Z}_q] \right) = 2 \times \text{Adv}_{\text{G}}^\text{ddh}(B) \leq 2 \times \text{Adv}_{\text{G}}^\text{ddh}(t)
\]

The adversary can ask any decryption of its choice:

\(\Rightarrow \) Chosen-Ciphertext Attacks (CCA)

Theorem (NM vs. CCA) \([\text{Bellare-Desai-Pointcheval-Rogaway – Crypto '98}]\)
The chosen-ciphertext security implies non-malleability \(\Rightarrow \) the highest security level

IND – CPA: Is it Enough?

The ElGamal Encryption

- \(G(1^k): G = \langle g \rangle \) of order \(q\), \(sk = x \leftarrow \mathbb{Z}_q\) and \(pk = y = g^x\)
- \(E(pk, m, r) = (c_1 = g^{r'}, c_2 = y^{r}m); D(sk, (c_1, c_2)) = c_2/c_1^{r}\)

Private Auctions

All the players \(P_i\) encrypt their bids \(c_i = E(pk, b_i)\) for the authority; the authority opens all the \(c_i\); the highest bid \(b_i\) wins

- IND – CPA guarantees privacy of the bids
- Malleability: from \(c_i = E(pk, b_i)\), without knowing \(b_i\), one can generate \(c' = E(pk, 2b_i)\): an unknown higher bid!

IND – CPA does not imply Non-Malleability

RSA-OAEP (PKCS #1 v2.1) \([\text{Bellare-Rogaway – Eurocrypt '94}]\)
The RSA encryption is OW – CPA, under the RSA assumption, but even not IND – CPA: need of randomness and redundancy

- \(m\) is the message to encrypt
- \(r\) is the additional randomness to make encryption probabilistic
- \(00\ldots00\) is redundancy to be checked at decryption time

After the transformation, \(X || Y\) goes in the plain RSA

Theorem (IND-CCA Security) \([\text{Fujisaki-Okamoto-Pointcheval-Stern – Crypto '01}]\)

RSA-OAEP is IND-CCA secure under the RSA assumption
More precisely, to get information on $m$, encrypted in $c = f(X||Y)$, one must have asked $\mathcal{H}(X) \Rightarrow$ partial inversion of $f$

For RSA: partial inversion and full inversion are equivalent (but at a computational loss)

If there is an adversary that distinguishes, within time $t$, the two ciphertexts with overwhelming advantage (close to 1), one can break RSA within time $T\approx 2^{t+3q_H^2k^3}$

(1024 bits: $k<2^{80}$, $T<2^{152}$)

(2048 bits: $k<2^{112}$, $T<2^{155}$)

(3072 bits: $k<2^{128}$, $T<2^{158}$)

large modulus: $>4096$ bits!

With provable security, one can precisely get:
- the security games one wants to resist against any adversary
- the security level, according to the resources of the adversary

But, it is under some assumptions:
- the best attacks against famous problems (integer factoring, etc)
- no leakage of information excepted from the given oracles

Cryptographers’ goals are thus
- to analyze the intractability of the underlying problems
- to define realistic and strong security notions (games)
- to correctly model the leakage of information (oracle access)
- to design schemes with tight security reductions

Implementations and uses must satisfy the constraints!