What Can Cryptography Guarantee?

Que peut nous garantir la cryptographie ?

David Pointcheval
Ecole normale supérieure

Fondation Sciences Mathématiques de Paris
September 27th, 2011

Security of Communications

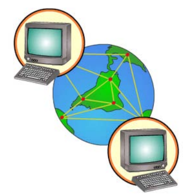
One ever wanted to exchange information securely

With the all-digital world, security needs are even stronger...

In your pocket



But also at home



First Encryption Mechanisms

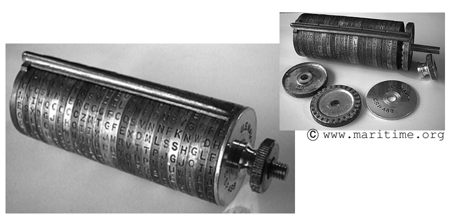
The goal of encryption is to hide a message

Scytale
Permutation

Substitutions and permutations
Security relies on
the secrecy of the mechanism

⇒ **How to widely use them?**

Alberti's disk
Mono-alphabetical Substitution

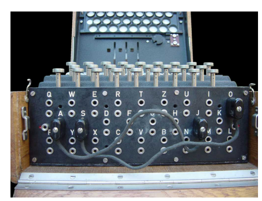


Wheel – M 94 (CSP 488)
Poly-alphabetical Substitution

Common Parameter

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**

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:

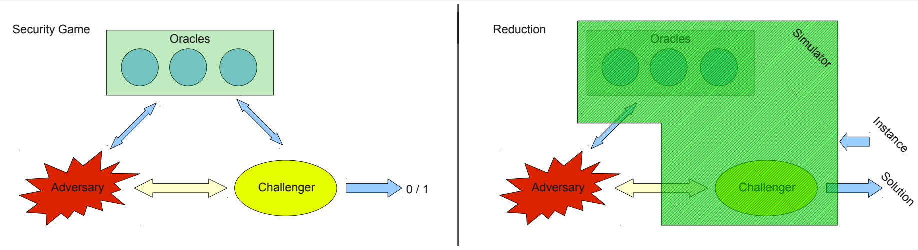
computers that run programs

What is a Secure Cryptographic Scheme?

- What does security mean? $\rightarrow$ Formal security notions
- How to guarantee above security claims? $\rightarrow$ Provable security

Computational Security Proofs

- 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

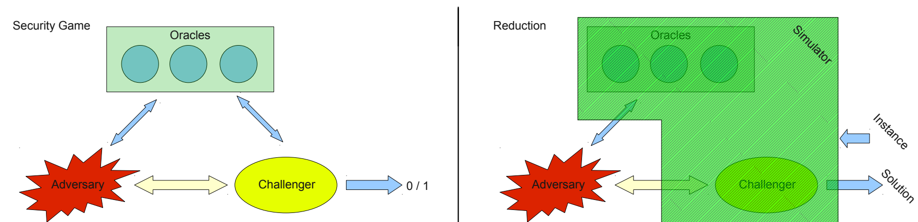
Given $n = pq$ $\longrightarrow$ Find p and q

Digits	Date	Bit-Length
130	April 1996	431 bits
140	February 1999	465 bits
155	August 1999	512 bits
160	April 2003	531 bits
200	May 2005	664 bits
232	December 2009	768 bits

Complexity

768 bits $\rightarrow 2^{64}$ op.	3072 bits $\rightarrow 2^{128}$ op.
1024 bits $\rightarrow 2^{80}$ op.	4096 bits $\rightarrow 2^{150}$ op.
2048 bits $\rightarrow 2^{112}$ op.	7680 bits $\rightarrow 2^{192}$ op.

Reduction



Adversary running time t | Algorithm running time $T = f(t)$

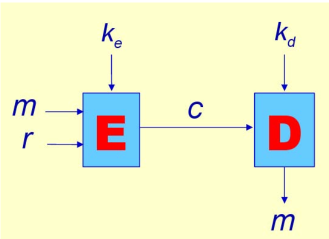
- Lossy reduction: $T = k^3 \times t$

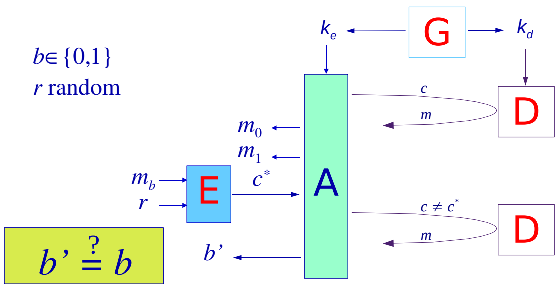
Modulus Bit-length	Adversary Complexity	Algorithm Complexity	Best Known Complexity	
$k = 2048$	$t < 2^{110}$	$T < 2^{143}$	2^{112}	✗
$k = 3072$	$t < 2^{110}$	$T < 2^{146}$	2^{128}	✗
$k = 4096$	$t < 2^{110}$	$T < 2^{146}$	2^{150}	✓

- Tight reduction: $T \approx t$
  With $k = 2048$ and $t < 2^{110}$, one gets $T < 2^{110}$ ✓

Public-Key Encryption

Goal: Privacy/Secrecy of the plaintext





No adversary can distinguish a ciphertext of m_0 from a ciphertext of m_1. **IND-CPA**

Even with an access to the decryption oracle (to model leakage of information). **IND-CCA**

RSA-OAEP (PKCS #1 v2.1)

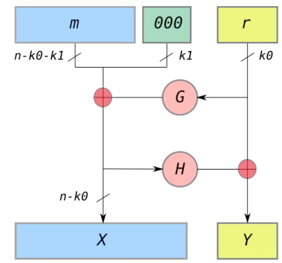
[Bellare-Rogaway – Eurocrypt '94]

The Plain $\mathcal{RSA}$ Encryption

[Rivest-Shamir-Adleman 1978]

- $\mathcal{G}(1^k)$: $n = pq$, $sk \leftarrow d = e^{-1} \bmod \varphi(n)$ and $pk \leftarrow (n, e)$
- $\mathcal{E}(pk, m) = c = m^e \bmod n$; $\mathcal{D}(sk, c) = m = c^d \bmod n$

Deterministic and malleable: **randomness and redundancy**



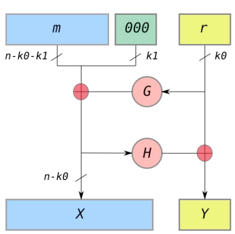
- m is the message to encrypt
- r is the additional randomness to make encryption probabilistic
- $00\ldots00$ is redundancy to be checked at decryption time

Then, $c = RSA(X\|Y)$

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

RSA-OAEP is IND-CCA secure under the RSA assumption

RSA-OAEP Security Proof [Fujisaki-Okamoto-Pointcheval-Stern – Crypto '01]



$c = f(X\|Y)$

To get information on m, $H(X)$ queried

$\Longrightarrow$ partial inversion of f

$c = RSA(X\|Y)$

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

If an adversary breaks **IND-CCA** within time t, one can break RSA within time $T \approx 2t + 3{q_H}^2 k^3$ (q_H = number of Hashing queries $\approx 2^{60}$)

$k = 2048$	(2^{112})	$t < 2^{110}$	$T < 2^{155}$	✘	$\Longrightarrow$ large modulus:
$k = 4096$	(2^{150})	$t < 2^{110}$	$T < 2^{158}$	✘	> 4096 bits!

REACT-RSA

[Okamoto-Pointcheval – CT-RSA '01]

$\mathcal{E}(pk, m, r) = (c_1 = r^e \bmod n, c_2 = G(r) \oplus m, c_3 = H(r, m, c_1, c_2))$

Security reduction between $\mathbf{IND}-\mathbf{CCA}$ and the RSA assumption:

$T \approx t \Longrightarrow$ 2048-bit RSA moduli provide 2^{110} security

Classical Assumptions

Main Assumptions

- Integer Factoring
- Modular Roots (Square roots and e-th roots)
- Discrete Logarithm (in Finite Fields and in Elliptic Curves)

Properties

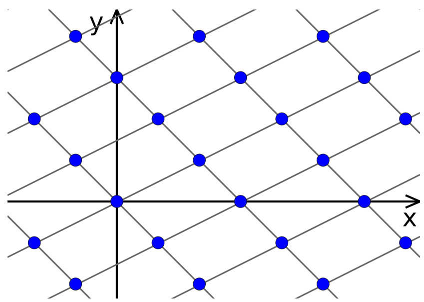
- Advantages: easy to implement, and widely used
- Drawbacks:
  - Factoring and DL in finite fields require larger and larger keys
  - They are all subject to quantum attacks [Shor 1997]

Alternatives: Post-Quantum Cryptography

- Error-Correcting Codes
- Systems of Multi-Variate Equations
- Lattices

Lattice-Based Cryptography

Lattice Problems
- Shortest Vector
- Small Basis (Reduced)
- Closest Vector



Properties
- Worst-case/Average-case Reductions
- No quantum attack known

Related Problems
- Learning With Errors
- Knapsack Problem

Cryptographic Primitives
- Identity Based Encryption
- Fully Homomorphic Encryption

Conclusion

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 the underlying problems
- no leakage of information excepted from the given oracles

Cryptographers' goals are thus
- analysis of the underlying problems / new problems
- realistic and strong security notions (games)
- accurate model for leakage of information (oracle access)
- tight security reductions

Implementations and uses must satisfy the constraints!