What Can Cryptography Guarantee?
Que peut nous garantir la cryptographie ?

David Pointcheval
Ecole normale supérieure
Fondation Sciences Mathématiques de Paris
September 27th, 2011

Cryptography
Provable Security
Encryption
Assumptions

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**
  - ⇒ information theory
- In practice: adversaries are limited in time/power  
  ⇒ **complexity theory**

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

| computers that run programs |

### 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

<table>
<thead>
<tr>
<th>Bit-length</th>
<th>2^{54} op.</th>
<th>2^{128} op.</th>
<th>2^{30} op.</th>
<th>2^{150} op.</th>
<th>2^{112} op.</th>
</tr>
</thead>
<tbody>
<tr>
<td>768 bits</td>
<td>1024 bits</td>
<td>2048 bits</td>
<td>3072 bits</td>
<td>4096 bits</td>
<td>7680 bits</td>
</tr>
</tbody>
</table>

#### Complexity

| $k = 2048$ | $t \leq 2^{110}$ | $T \leq 2^{143}$ | $2^{112}$ | $\times$ |
| $k = 3072$ | $t \leq 2^{110}$ | $T \leq 2^{146}$ | $2^{128}$ | $\times$ |
| $k = 4096$ | $t \leq 2^{110}$ | $T \leq 2^{146}$ | $2^{150}$ | $\checkmark$ |

- **Lossy reduction:** $T = k^3 \times t$
- **Tight reduction:** $T \approx t$
  - With $k = 2048$ and $t \leq 2^{110}$, one gets $T \leq 2^{110}$

### What is a Secure Cryptographic Scheme?
- What does security mean? → **Formal security notions**
- How to guarantee above security claims? → **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)
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 Security Proof [Fujisaki-Okamoto-Pointcheval-Stern – Crypto '01]

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

- $k = 2048$ (2^{112})
- $k = 4096$ (2^{150})

REACT-RSA [Okamoto-Pointcheval – CT-RSA '01]

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

Security reduction between IND – CCA and the RSA assumption:

$T \approx t \implies 2048$-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
- **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!*