Security Proofs

Asymmetric Encryption without Redundancy

Rennes – January 2004

Joint work with Duong Hieu Phan

David Pointcheval
CNRS-ENS, Paris, France

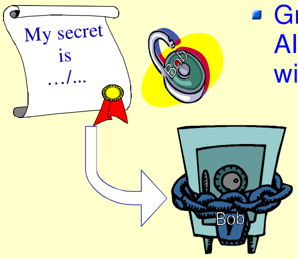
Summary

- Introduction
- Provable Security
- Asymmetric Encryption
- New Schemes

Summary

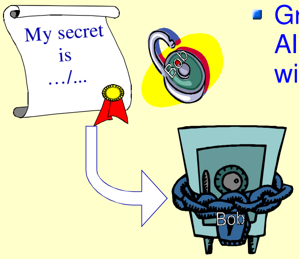
- Introduction
- Provable Security
- Asymmetric Encryption
- New Schemes

Encryption / decryption attack



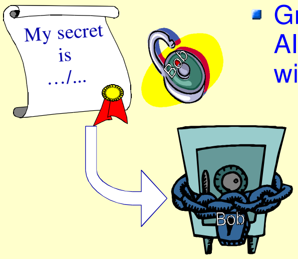
- Granted Bob's public key, Alice can lock the safe, with the message inside
  (*encrypt the message*)

Encryption / decryption attack



- Granted Bob's public key, Alice can lock the safe, with the message inside
  (*encrypt the message*)
- Alice sends the safe to Bob no one can unlock it
  (*impossible to break*)

Encryption / decryption attack



- Granted Bob's public key, Alice can lock the safe, with the message inside
  (*encrypt the message*)
  - *Excepted Bob, granted his private key (Bob can decrypt)*
- Alice sends the safe to Bob no one can unlock it
  (*impossible to break*)

Kerckhoffs' Principles (1)

In 1883, in "La Cryptographie Militaire" Kerckhoffs wrote:

- *Le système doit être matériellement, sinon mathématiquement, indéchiffrable*
  - The system should be, if not theoretically unbreakable, unbreakable in practice

Kerckhoffs' Principles (2)

- *Il faut qu'il n'exige pas le secret, et qu'il puisse sans inconvénient tomber entre les mains de l'ennemi*
  - Compromise of the system should not inconvenience the correspondents

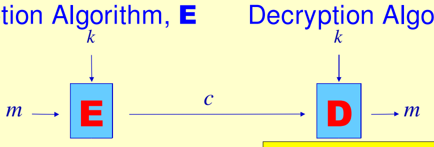
Kerckhoffs' Principles (3)

- *La clef doit pouvoir en être communiquée et retenue sans le secours de notes écrites, et être changée ou modifiée au gré des correspondants*
  - the key should be rememberable without notes and should be easily changeable
- etc …

Symmetric Encryption

- Principles 2 and 3 define the concept of the *symmetric* cryptography:

Encryption Algorithm, **E** Decryption Algorithm, **D**



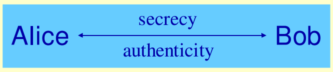
Security = secrecy: impossible to recover m from c only (without k)

Security : heuristic

1st Principle

Asymmetric Cryptography

Extends 2nd principle



Diffie-Hellman 1976

Asymmetric Encryption: Bob owns two "keys"

- A public key (encryption k_e) ⇒ known by everybody (included Alice)
  - so that anybody can encrypt a message for him
- A private key (decryption k_d) ⇒ known by Bob only
  - to help him to decrypt

Integer Factoring and RSA

One-Way Function

- Multiplication/Factorization:
  - $p, q \mapsto n = p.q$ easy (quadratic)
  - $n = p.q \mapsto p, q$ difficult (super-polynomial)

Integer Factoring and RSA

- Multiplication/Factorization:
  - $p, q \mapsto n = p.q$ easy (quadratic)
  - $n = p.q \mapsto p, q$ difficult (super-polynomial)
- RSA Function, from $\mathbf{Z}_n$ in $\mathbf{Z}_n$ (with $n=pq$)

  for a fixed exponent e Rivest-Shamir-Adleman 1978
  - $x \mapsto x^e \bmod n$ easy (cubic)
  - $y=x^e \bmod n \mapsto x$ difficult (without p or q)

    $x = y^d \bmod n$ where $d = e^{-1} \bmod\ (n)$

One-Way Function

RSA Problem

Integer Factoring and RSA

- Multiplication/Factorization:
  - $p, q \mapsto n = p.q$ easy (quadratic)
  - $n = p.q \mapsto p, q$ difficult (super-polynomial)
- RSA Function, from $\mathbf{Z}_n$ in $\mathbf{Z}_n$ (with $n=pq$)

  for a fixed exponent e Rivest-Shamir-Adleman 1978
  - $x \mapsto x^e \bmod n$ easy (cubic)
  - $y=x^e \bmod n \mapsto x$ difficult (without p or q)

    $x = y^d \bmod n$ where $d = e^{-1} \bmod\ (n)$

One-Way Function

encryption

Integer Factoring and RSA

- Multiplication/Factorization:
  - $p, q \mapsto n = p.q$ easy (quadratic)
  - $n = p.q \mapsto p, q$ difficult (super-polynomial)
- RSA Function, from $\mathbf{Z}_n$ in $\mathbf{Z}_n$ (with $n=pq$)

  for a fixed exponent e Rivest-Shamir-Adleman 1978
  - $x \mapsto x^e \bmod n$ easy (cubic)
  - $y=x^e \bmod n \mapsto x$ difficult (without p or q)

    $x = y^d \bmod n$ where $d = e^{-1} \bmod\ (n)$

One-Way Function

difficult to break

Integer Factoring and RSA

- Multiplication/Factorization:
  - $p, q \mapsto n = p.q$ easy (quadratic)
  - $n = p.q \mapsto p, q$ difficult (super-polynomial)
- RSA Function, from $\mathbf{Z}_n$ in $\mathbf{Z}_n$ (with $n=pq$)

  for a fixed exponent e Rivest-Shamir-Adleman 1978
  - $x \mapsto x^e \bmod n$ easy (cubic)
  - $y=x^e \bmod n \mapsto x$ difficult (without p or q)

    $x = y^d \bmod n$ where $d = e^{-1} \bmod\ (n)$

One-Way Function

trapdoor

key

decryption

Summary

- Introduction
- Provable Security
- Asymmetric Encryption
- New Schemes

Algorithmic Assumptions *necessary*

- $n=pq$: **public modulus**
- e : **public exponent**
- $d=e^{-1} \bmod\ (n)$: **private**

RSA Encryption

- $\mathbf{E}(m) = m^e \bmod n$
- $\mathbf{D}(c) = c^d \bmod n$

If the RSA problem is easy,
secrecy is not satisfied:
anybody may recover m from c

Algorithmic Assumptions *sufficient?*

Security proofs give the guarantee that the assumption is ***enough*** for secrecy:

- if an adversary can break the secrecy
- one can break the assumption

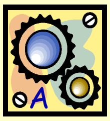
“reductionist” proof

Extends the 1st Principle

Proof by Reduction

Reduction of a problem **P** to an attack *Atk*:

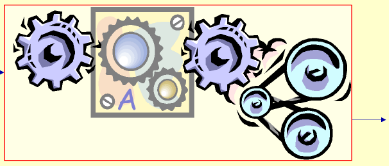
- Let *A* be an adversary that breaks the scheme
- Then *A* can be used to solve **P**



Proof by Reduction

Reduction of a problem **P** to an attack *Atk*:

- Let *A* be an adversary that breaks the scheme
- Then *A* can be used to solve **P**



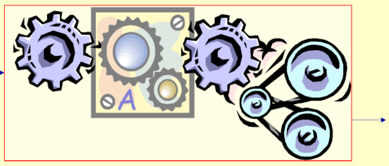
Instance **I** of **P** →

→ Solution of **I**

Proof by Reduction

Reduction of a problem **P** to an attack *Atk*:

- Let *A* be an adversary that breaks the scheme
- Then *A* can be used to solve **P**



Instance **I** of **P** →

→ Solution of **I**

P intractable ⟹ scheme unbreakable

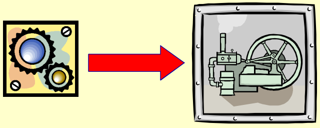
Provably Secure Scheme

To prove the security of a cryptographic scheme, one has to make precise

- the algorithmic assumptions
  - some have been presented
- the security notions to be guaranteed
  - depends on the scheme
- a reduction:
  an adversary can help
  to break the assumption

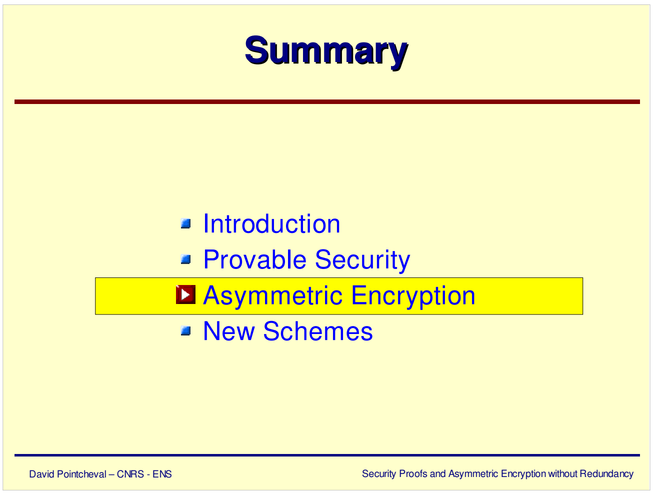
Practical Security

Adversary within t



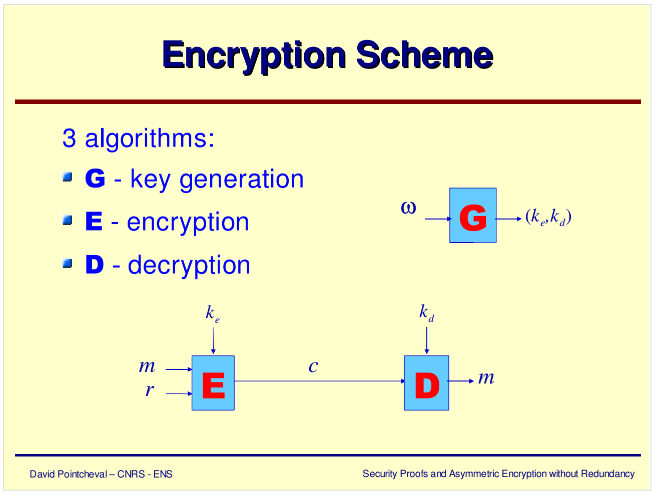
Algorithm against **P** within $t' = T(t)$

- Complexity theory: T polynomial
- Exact Security: T explicit
- Practical Security: T small (linear)



Summary

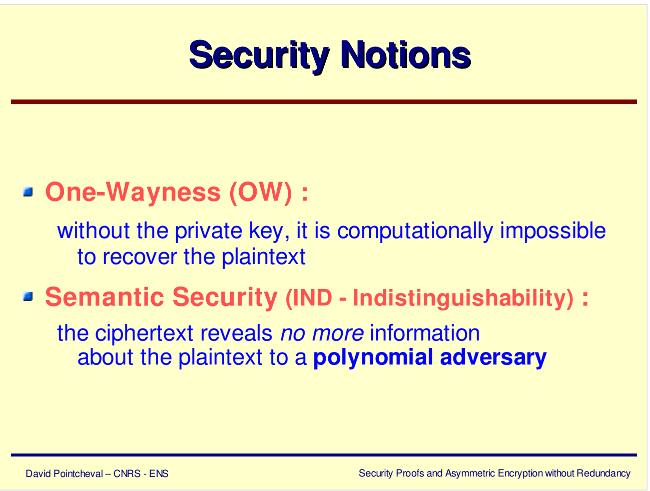
- Introduction
- Provable Security
- Asymmetric Encryption
- New Schemes



Encryption Scheme

3 algorithms:

- **G** - key generation
- **E** - encryption
- **D** - decryption



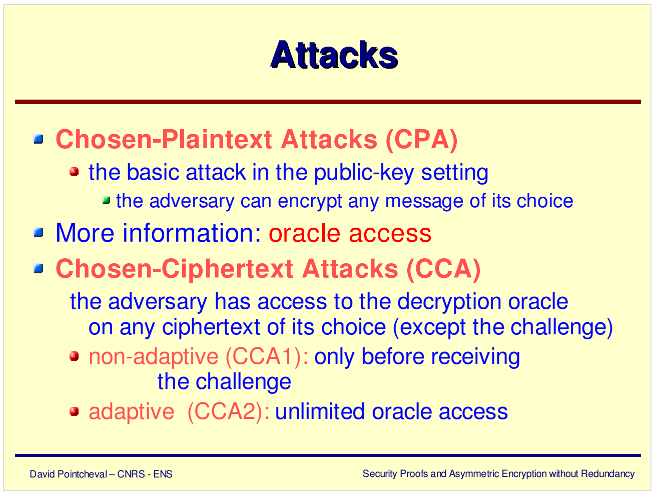
Security Notions

- **One-Wayness (OW) :**

  without the private key, it is computationally impossible to recover the plaintext

- **Semantic Security (IND - Indistinguishability) :**

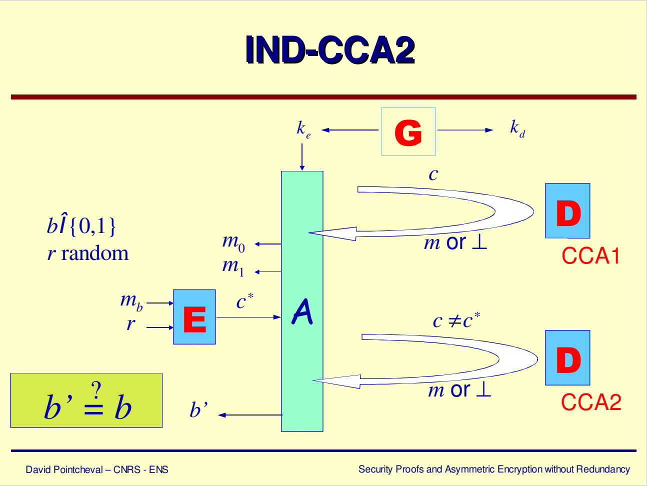
  the ciphertext reveals *no more* information about the plaintext to a **polynomial adversary**



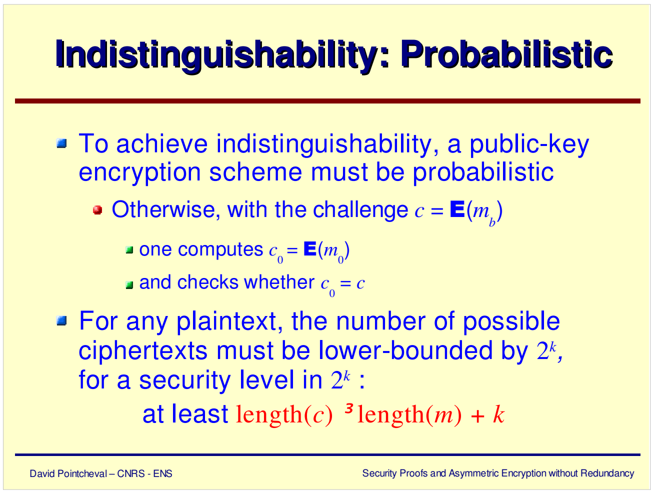
Attacks

- Chosen-Plaintext Attacks (CPA)
  - the basic attack in the public-key setting
    - the adversary can encrypt any message of its choice
- More information: oracle access
- Chosen-Ciphertext Attacks (CCA)

  the adversary has access to the decryption oracle on any ciphertext of its choice (except the challenge)
  - non-adaptive (CCA1): only before receiving the challenge
  - adaptive (CCA2): unlimited oracle access



IND-CCA2



Indistinguishability: Probabilistic

- To achieve indistinguishability, a public-key encryption scheme must be probabilistic
  - Otherwise, with the challenge c = **E**(m_b)
    - one computes c_0 = **E**(m_0)
    - and checks whether $c_0 = c$
- For any plaintext, the number of possible ciphertexts must be lower-bounded by 2^k, for a security level in 2^k :

  at least length(c) ³ length(m) + k

Chosen-Ciphertext Security: Redundancy

- To resist chosen-ciphertext attacks, one makes the decryption oracle unuseful:
  - Very few ciphertexts are valid
  - For building a valid ciphertext, the adversary necessarily knows the corresponding plaintext
- Examples
  - Zero-knowledge proof of knowledge of the plaintext
  - Zero-knowledge proof of validity (CCA1 - Naor-Yung 90)
    - $C = (c_1, c_2, p)$ where $c_1 = \mathbf{E}_{pk_1}(m_1)$, $c_2 = \mathbf{E}_{pk_2}(m_2)$ and p is a proof that $m_1 = m_2$

CCA: Redundancy (Cont'd)

Practical constructions:

- OAEP: redundancy in the padding
- REACT: MAC in the ciphertext
- Cramer-Shoup: Proof of validity = redundancy

Such a redundancy makes that a random ciphertext is valid (a possible output of the encryption algorithm) with a very small probability, less than 2^{-k}:

in practice: at least $\mathrm{length}(c)$ ³ $\mathrm{length}(m) + 2k$

Optimal Size = No Redundancy

- No redundancy = any ciphertext is valid:
  - is a possible output of $\mathbf{E}(m,r)$
  - the function $\mathbf{E}: \mathcal{M}$ ´ $\mathcal{R} \to \mathcal{C}$
    $(m,r) \to c$ is a surjection
- Advantages:
  - optimal bandwidth
  - no reaction attack / implementation issues
  - easier distribution of the decryption process

Summary

- Introduction
- Provable Security
- Asymmetric Encryption
- New Schemes

Full-Domain Permutation Encryption

- First candidate: in the same vein as the Full-Domain Hash Signature
- Public permutation $\mathbf{P}$ (Random Permutation Model) onto $\mathcal{M}$ ´$\mathcal{R} \approx \mathcal{C} \approx \{0,1\}^n$ ´$\{0,1\}^k \approx \{0,1\}^l$
- Trapdoor one-way permutation f onto $\{0,1\}^l$

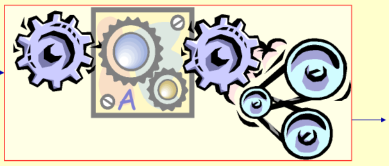
$$\mathbf{E}: \quad \mathcal{M} \text{ ´} \mathcal{R} \to \mathcal{C}$$
$$(m,r) \to c = f(\mathbf{P}(m,r))$$

  - the public key is the pair $(f, \mathbf{P})$ which includes $\mathbf{P}^{-1}$
  - the private key is the trapdoor f^{-1}

FDP Encryption is IND-CCA2 Secure

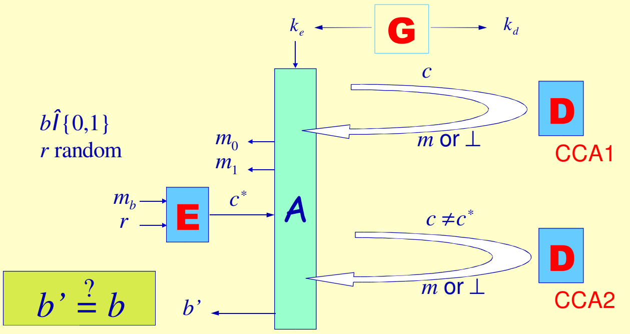
- In the RPM, a (t,ε)-IND-CCA2 adversary helps to invert f within almost the same time t, and with success probability greater than $\varepsilon - q/2^k$

Instance $y = f(x)$



Solution x

Game IND-CCA2



FDP Encryption is IND-CCA2 Secure

Simulation of the oracles

- **G**, for generating f and **E**, outputting y
- **P**, **P**$^{-1}$ and **D** using a list of tuples $\{(m, r, p, c)\}$

$$p = \mathbf{P}(m,r),\ c = f(p) = \mathbf{E}(m,r)$$

- problem if (m,r) is assumed to correspond to **P**$^{-1}(f^{-1}(c))$ from the **D**-simulation, and *A* asks for **P**(m,r): the simulation should output $p = f^{-1}(c)$, which is unknown but **D** outputs m only: r is unpredictable

  unless there are collisions on m, the probability of such an event is less than $q_{\mathbf{P}}/2^k$

FDP Encryption: Properties

- No redundancy
- Optimal bandwidth: length(c) = length(m) + k
- High security level: IND-CCA2
  - with efficient reduction
  - but in the Random-Permutation Model

Can we weaken the assumptions?

The Random-Oracle Model

- A weaker model : the random-oracle model
  - access to a truly random function
- How to build a random permutation from a random function?
  - Luby-Rackoff: a Feistel construction
  - not that easy: here, one has access to the internal function...

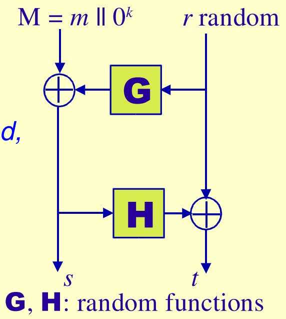
  Let us try anyway:
  - OAEP, a 2-round Feistel Network

2-round OAEP

- **E**(m) : $c = f(s \,\|\, t)$
- **D**(c) : $s \,\|\, t = f^{-1}(c)$

then invert OAEP,

if the redundancy is satisfied, one returns m



G, **H**: random functions

2-round OAEP (cont'd)

- In the random-oracle model
- If *f is a trapdoor partial-domain OW permutation*:
  - (s,t) ® $f(s \,\|\, t)$ trapdoor one-way
  - $f(s \,\|\, t)$ ® s also hard to compute
- With a redundancy 0^k and random of size k_0
- The encryption scheme f-OAEP:
- IND-CCA2 with quadratic lost (in $q_{\mathbf{D}} q_{\mathbf{G}} / 2^{k_0}$: $k_0 = 2k$)

length(c) = length(m) + 3k

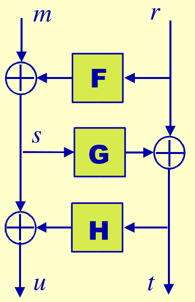
What About the Redundancy?

- For IND-CCA2: redundancy
  Plaintext-awareness = invalid ciphertexts
- ***Without redundancy... is it still IND-CCA2?***
  - 2-round OAEP: no known attack, but no proof either
    - Any simulation seems to be subject
      to the Shoup's attack (malleability of OAEP)
  - 3-round OAEP: can be proven

3-round OAEP

- $\mathbf{E}(m) : c = f\,(t \parallel u)$
- $\mathbf{D}(c) \;: t \parallel u = f^{-1}(c)$

then invert OAEP,
and return m



F, **G** and **H**: random functions

Idea of the Security

- 2-round OAEP: as in the Shoup's attack,
  - the adversary can forge a ciphertext c,
    with the same r as in the challenge ciphertext
  - the simulator cannot check it
  - the adversary can always distinguish the simulation
- With one more round:
  - the adversary is stuck!

    one can simulate everything
    in a consistent way
  - at random when not already known
  - anticipating some future answers, when determined

Tightness of the Reduction

- Everything works well with lists, $_{\mathbf{F}}$, $_{\mathbf{G}}$, $_{\mathbf{H}}$, $_{\mathbf{D}}$
- But for $g = \mathbf{G}(s)$, which implies
  - $\mathbf{F}(r) = m \oplus s$ for $r = t \oplus g$
  - for any $(t, h) \in {}_{\mathbf{H}}$, and $(m,c) \in {}_{\mathbf{D}}$
    such that $c = f\,(t, h \oplus s)$

  in case such a query is asked later
- Problem if such a query has already been asked...

Since g is random, the overall probability of such a bad event is upper-bounded by $q_{\mathbf{D}}\, q_{\mathbf{F}} / 2^k$.

Security Result

- With a random of size k_0, but no redundancy
- In the ROM, a $(t,\ )$-IND-CCA2 adversary helps to partially invert f within $t' \approx t + q_{\mathbf{G}} q_{\mathbf{H}} T_f$, and with success probability greater than $\ - q_{\mathbf{D}} Q / 2^{k_0}$
- The 3-round OAEP is:
- IND-CCA2 with quadratic lost $(k_0 = 2k)$

$$\mathrm{length}(c) = \mathrm{length}(m) + 2k$$

Conclusion

- We have proposed the first IND-CCA2 encryption schemes, without redundancy:
  - the FDP encryption is optimal
    - based on the OW of the trapdoor permutation
    - optimal bandwidth
    - but in the Random-Permutation Model
  - the 3-round OAEP
    - with similar characteristics
      as the 2-round OAEP
    - but without redundancy