Provable Security
Asymmetric Encryption

DEA - Algorithmique
ENS
20 Février 2003

David Pointcheval
LIENS-CNRS
Ecole normale supérieure

Summary

1. Introduction
2. Computational Assumptions
3. Security Proofs
4. Asymmetric Encryption
5. New Assumptions
6. An Example

Summary

1. Introduction
2. Computational Assumptions
3. Security Proofs
4. Asymmetric Encryption
5. New Assumptions
6. An Example

Two Keys…

Asymmetric Cryptography

Alice $\xleftrightarrow[\text{authenticity}]{\text{secrecy}}$ Bob

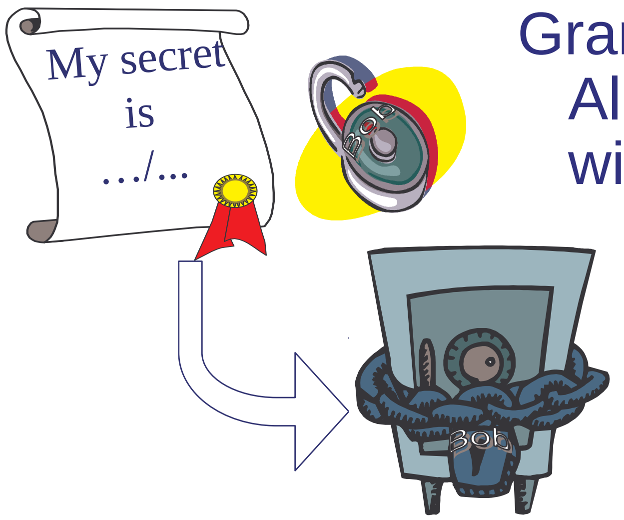
Diffie-Hellman 1976

Asymmetric Encryption:

Bob owns two "keys"

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

Encryption / decryption attack



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

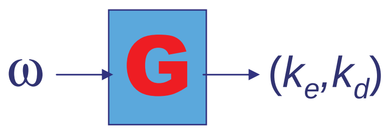
Excepted Bob,
granted his private key
(*Bob can decrypt*)

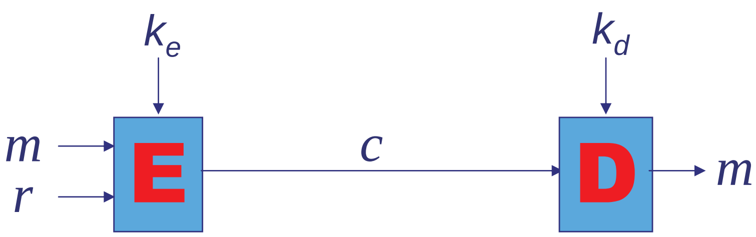
Alice sends the safe to Bob
no one can unlock it
(*impossible to break*)

Encryption Scheme

3 algorithms :

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





Conditional Secrecy

The ciphertext comes from $c = \mathbf{E}_{k_e}(m;r)$

- The encryption key k_e is public
- A unique m satisfies the relation (with possibly several r)

At least exhaustive search on m and r can lead to m, maybe a better attack!

$\Rightarrow$ unconditional secrecy impossible

Algorithmic assumptions

Summary

1. Introduction
2. Computational Assumptions
3. Security Proofs
4. Asymmetric Encryption
5. New Assumptions
6. An Example

Integer Factoring and RSA

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

One-Way Function

- 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)

RSA Problem

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

key

decryption

The Discrete Logarithm

- Let $\mathbf{G} = (<g>, \times)$ be any finite cyclic group
- For any $y \in \mathbf{G}$, one defines

$$\mathrm{Log}_g(y) = \min\{x \geq 0 \mid y = g^x\}$$

- One-way function
  - $x \rightarrow y = g^x$ easy (cubic)
  - $y = g^x \rightarrow x$ difficult (super-polynomial)

$$\boxed{\mathrm{Succ}_g^{\mathrm{dl}}(\mathcal{A}) = \Pr_{x \in \mathbf{Z}_q}\left[\mathcal{A}(y) = x \middle| y = g^x\right]}$$

The Diffie-Hellman Problems

- The Diffie-Hellman Problem (1976):
  - Given $A=g^a$ and $B=g^b$
  - Compute DH$(A,B) = C=g^{ab}$
- The **Decisional Diffie-Hellman Problem**:
  - Given A, B and C in $<g>$
  - Decide whether $C =$ DH(A,B)

$$\mathrm{Adv}_g^{\mathrm{ddh}}(\mathcal{A}) = \left| \begin{array}{l} \Pr_{a,b,c\in \mathbb{Z}_q}\left[\mathcal{A}(A,B,C)=1 \middle| A=g^a, B=g^b, C=g^c\right] \\ \quad - \Pr_{a,b\in \mathbb{Z}_q}\left[\mathcal{A}(A,B,C)=1 \middle| A=g^a, B=g^b, C=g^{ab}\right] \end{array} \right|$$

Complexity Estimates

Estimates for integer factoring Lenstra-Verheul 2000

	Modulus (bits)	Mips-Year ($\log_2$)	Operations (en $\log_2$)
	512	13	58
Mile-stone	1024	35	80
	2048	66	111
	4096	104	149
	8192	156	201

Can be used for RSA too
Lower-bounds for DL in $\mathbb{Z}_p^*$

Summary

1. Introduction
2. Computational Assumptions
3. Security Proofs
4. Asymmetric Encryption
5. New Assumptions
6. An Example

Algorithmic Assumptions *necessary*

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

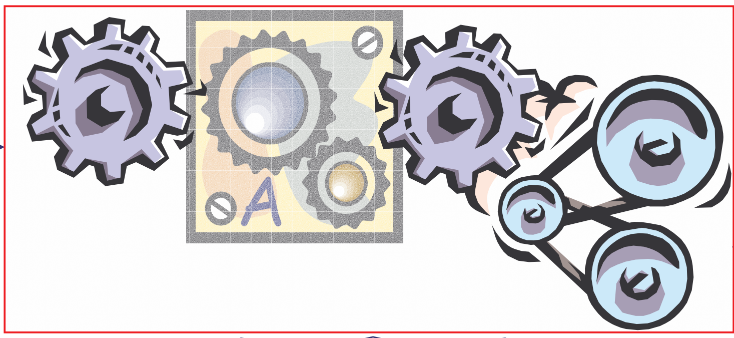
$\Rightarrow$ “reductionist” proof

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** $\rightarrow$



$\rightarrow$ Solution of **I**

P intractable $\Rightarrow$ scheme unbreakable

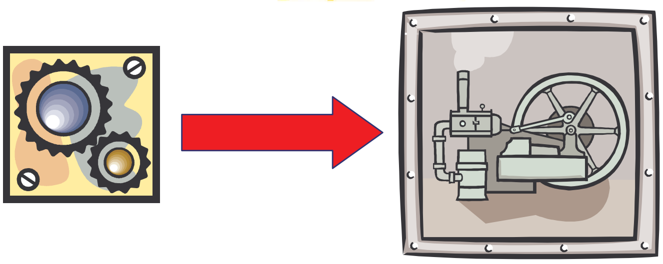
Provably Secure Scheme

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

- the algorithmic assumptions
- the security notions to be guaranteed
- 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)

Eg : $t' = 4t$

P intractable within less than 2^{80} operations
$\Rightarrow$ scheme unbreakable
within less than 2^{78} operations

Security Notions

According to the needs, one defines

- the goals of an adversary
- the means of an adversary,
  i.e. the available information

Summary

1. Introduction
2. Computational Assumptions
3. Security Proofs
4. Asymmetric Encryption
   - Formal Security Model
   - Examples
5. New Assumptions
6. An Example

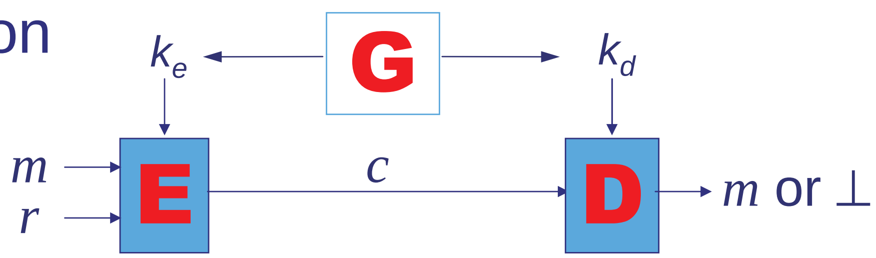
Asymmetric Encryption

- Formal Security Model
- Examples

Encryption Scheme

3 algorithms :

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



OW-Security: it is impossible to get back m just from c, k_e, **E** and **D** (without k_d)

Basic Secrecy

- One-Wayness (OW) :

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

$$\mathrm{Succ}^{ow}(\mathcal{A}) = \Pr_{m,r}\left[\mathcal{A}(k_e, c) = m \middle| c = \mathbf{E}(m;r)\right]$$

Not enough if one already has some information about m :

- "Subject: XXXXX"
- "My answer is XXX" (XXX = Yes/No)

Strong Secrecy

- Semantic Security (IND - Indistinguishability) :

  GM 1984

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

$$\mathrm{Adv}^{ind}(\mathcal{A}) =$$

$$2\Pr_{r,b}\left[\mathcal{A}_2(m_0, m_1, c, s) = b \middle| \begin{array}{c} (m_0, m_1, s) \leftarrow \mathcal{A}_1(k_e) \\ c \leftarrow \mathbf{E}(m_b, r) \end{array}\right] - 1$$

Non-Malleability

- Non-Malleability (NM):

DDN 1991

No polynomial adversary can derive, from a ciphertext $c = \mathbf{E}(m;r)$, a second one $c' = \mathbf{E}(m';r')$ so that the plaintexts m and m' are meaningfully related

non-malleability
$\Downarrow$
semantic security
$\Downarrow$
one-wayness

Basic Attacks

- Chosen-Plaintext Attacks (CPA)

In public-key cryptography setting,
the adversary can encrypt any message of his choice, granted the public key

$\Rightarrow$ the basic attack

Improved Attacks

- More information: oracle access
- Chosen-Ciphertext Attacks (CCA)

  The adversary has access to the strongest oracle: the decryption oracle

  The adversary can obtain the plaintext of any ciphertext of his choice (excepted the challenge)

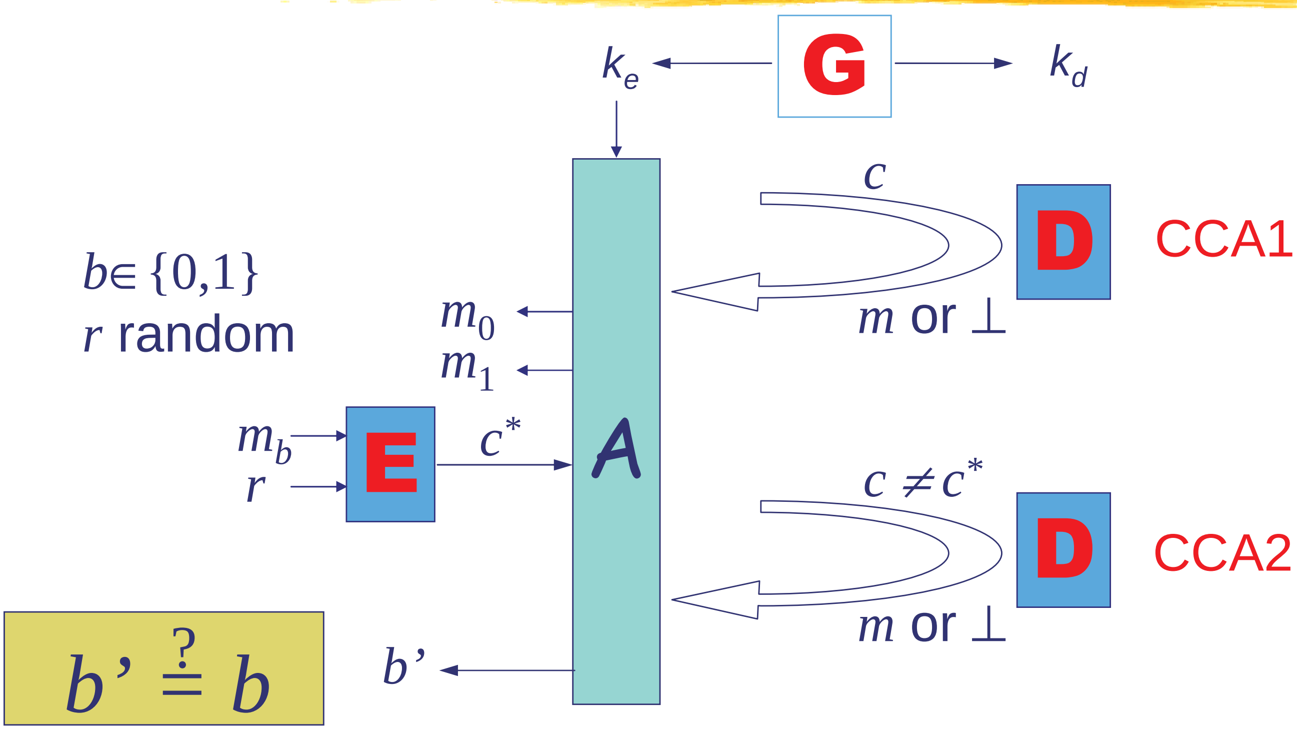
  – non-adaptive (CCA1) NY 1990

  only before receiving the challenge

  – adaptive (CCA2) RS 1991

  unlimited oracle access

IND-CCA2



Relations BDPR C-1998

Implications and separations

$$
\begin{array}{ccccc}
\text{NM-CPA} & \Leftarrow & \text{NM-CCA1} & \Leftarrow & \text{NM-CCA2} \\
\Downarrow & & \Downarrow & & \Updownarrow \\
\text{IND-CPA} & \Leftarrow & \text{IND-CCA1} & \Leftarrow & \text{IND-CCA2} \\
\Downarrow & & & & \\
\text{OW-CPA} & & & &
\end{array}
$$

minimal security

weak security

strong security: CCA

Asymmetric Encryption

- Formal Security Model
- Examples

RSA Encryption

- $n = pq$, product of large primes
- e, relatively prime to $\varphi(n) = (p\text{-}1)(q\text{-}1)$
- n, e : **public** key
- $d = e^{-1} \bmod \varphi(n)$: **private** key

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

OW-CPA = RSA problem

Nothing to prove = definition

El Gamal Encryption

- $\mathbf{G} = (<g>, \times)$ group of order q
- x : **private** key
- $y=g^x$: **public** key

$$\mathbf{E}(m;a) = (g^a, y^a m) \rightarrow (c,d) \qquad \mathbf{D}(c,d) = d / c^x$$

OW-CPA = CDH Assumption

IND-CPA = DDH Assumption

To be proven to see the restrictions

El Gamal: OW-CPA

$$\mathbf{E}(m;a) = (g^a, y^a m) \rightarrow (c,d) \quad \mathbf{D}(c,d) = d / c^x$$

$$\mathrm{Succ}^{ow}(\mathcal{A}) = \Pr_{m,r}\left[\mathcal{A}(y,(c,d)) = m \,\middle|\, (c,d) = \mathbf{E}(m;a)\right]$$

$\mathcal{B}$ is given as input $\mathbf{G} = (<g>, \times)$ and (A,B)

- $y \leftarrow A$ and $c \leftarrow B$
- choose a random value d : $\mathcal{A}(y,(c,d)) \rightarrow m$
- output d/m

If m is correct, $\mathrm{DH}(A,B)=d/m$

$\mathrm{Succ}^{cdh}(\mathcal{B}) = \mathrm{Succ}^{ow}(\mathcal{A})$

El Gamal: IND-CPA

$$\mathrm{Adv}^{ind}(\mathcal{A}) = 2\Pr_{a,b}\left[\mathcal{A}_2(m_0,m_1,(c,d),s) = b \,\middle|\, \begin{array}{c}(m_0,m_1,s) \leftarrow \mathcal{A}_1(y)\\ (c,d) \leftarrow \mathbf{E}(m_b;a)\end{array}\right] - 1$$

$\mathcal{B}$ is given as input $\mathbf{G} = (<g>, \times)$ and (A, B, C)

- $y \leftarrow A$ and $c \leftarrow B$: $\mathcal{A}_1(y) \rightarrow (m_0, m_1)$
- $b \in \{0,1\}$ and $d \leftarrow C\, m_b$: $\mathcal{A}_2(c,d) \rightarrow b'$
- output $\beta = (b=b')$

Let us assume that $m_0, m_1 \in \mathbf{G}$:

- If $C=\mathrm{DH}(A,B)$, $\Pr[b=b'] = \Pr[\mathcal{A}(c,d) = b]$
- If $C\neq\mathrm{DH}(A,B)$, $\Pr[b=b'] = 1/2$

El Gamal: IND-CPA (Cnt'd)

If the messages are encoded into **G**:

- If C=DH(A,B), $\Pr[b=b'] = \Pr[\mathcal{A}(c,d) = b]$
- If C≠DH(A,B), $\Pr[b=b'] = 1/2$

$$\begin{aligned}\mathrm{Adv}^{\mathrm{ddh}}(\mathcal{B}) &= \Pr[\beta = 1 \mid C = \mathrm{CDH}(A,B)] - \Pr[\beta = 1 \mid C \neq \mathrm{CDH}(A,B)] \\ &= \Pr[b' = b\ ] - \frac{1}{2} = \frac{1}{2}\mathrm{Adv}^{ind}(\mathcal{A})\end{aligned}$$

$$\begin{aligned}\mathrm{Adv}(\mathcal{D}) &= 2\Pr[b' = b] - 1 \\ &= \Pr[b' = b \mid b = 1] + \Pr[b' = b \mid b = 0] - 1 \\ &= \Pr[b' = b \mid b = 1] - \Pr[b' \neq b \mid b = 0] \\ &= \Pr[b' = 1 \mid b = 1] - \Pr[b' = 1 \mid b = 0]\end{aligned}$$

Thus,

$\mathrm{Adv}^{ind}(t) \leq 2\ \mathrm{Adv}^{\mathrm{ddh}}(t')$

Summary

1. Introduction
2. Computational Assumptions
3. Security Proofs
4. Asymmetric Encryption
5. New Assumptions
6. An Example

Strong Security Notions

It is very difficult to reach CCA security

Maybe possible, but with inefficient schemes

Inefficient schemes are unuseful in practice:

Everybody wants security,
but only if it is transparent

Ideal Models

$\Rightarrow$ one makes some ideal assumptions:

- ideal random hash function:
  random oracle model
- ideal symmetric encryption:
  ideal cipher model
- ideal group:
  generic model (generic adversaries)

The Random Oracle Model

- Introduced by Bellare-Rogaway ACM-CCS '93
- The most admitted model
- It consists in considering some functions as perfectly random functions, or replacing them by random oracles:
  - each new query is returned a random answer
  - a same query asked twice receives twice the same answer

Modeling a Random Oracle

A usual way to model a random oracle H is to maintain a list Λ_H which contains all the query-answers (x,ρ):

- Λ_H is initially set to an empty list
- A query x to H is answered the following way
  - if for some ρ, $(x,\rho) \in \Lambda_H$, ρ is returned
  - otherwise,

    s a random ρ is drawn from the appropriate range
    s (x,ρ) is appended to Λ_H
    s ρ is returned

Summary

1. Introduction
2. Asymmetric Cryptography
3. Computational Assumptions
4. Security Proofs
5. Encryption and Signature
6. New Assumptions
7. An Example

Generic Construction Bellare-Rogaway '93

Let f be a trapdoor one-way permutation

then (with $G \rightarrow \{0,1\}^\ell$ and $H \rightarrow \{0,1\}^k$)

$$\mathbf{E}(m;r) = f(r) \,\|\, m \oplus G(r) \,\|\, H(m,r)$$

$$\mathbf{D}(a,b,c): \quad r = f^{-1}(a)$$

$$m = b \oplus G(r)$$

$$c = H(m,r)\ ?$$

IND-CCA2: Security Proof

Adversary $\mathcal{A}=(\mathcal{A}_1,\mathcal{A}_2)$

- $\mathcal{A}_1(f) \rightarrow (m_0,m_1)$
- One randomly chooses $\beta \in \{0,1\}$ and r,

  and computes $C = \mathbf{E}(m_\beta,r) = (a,b,c)$:

  $$a = f(r),\ b = m_\beta \oplus G(r),\ c = H(m_\beta,r)$$
- $\mathcal{A}_2(C) \rightarrow \beta'$

with permanent access to

- the decryption oracle $\mathbf{D}$ — $q_{\mathbf{D}}$ queries
- the random oracles G and H — q_G, q_H queries

IND-CCA2: Security Proof (2)

Adversary $\mathcal{A}=(\mathcal{A}_1,\mathcal{A}_2)$ - Simulator $\mathcal{B}$

- $\mathcal{B}(f, y=f(x))$: runs $\mathcal{A}_1(f) \rightarrow (m_0,m_1)$
- randomly chooses $b \in \{0,1\}^\ell$ and $c \in \{0,1\}^k$

  and outputs $C = \mathbf{E}(m_\beta,r) = (y,b,c)$

  this implicitly defines:

  $$r = f^{-1}(y) = x,\ G(r) = m_\beta \oplus b,\ H(m_\beta,r) = c$$
- $\mathcal{A}_2(C) \rightarrow \beta'$

IND-CCA2: Simulation (3)

$\mathcal{B}$ has to answer oracle queries:

- Random oracles G and H

  a new query is answered by

  a new random value in the proper range

  Problem if $G(r)$ (**AskG**) or $H(m_\beta,r)$ (**AskH**)

- Decryption oracle on $C' = (a',b',c')$

  one looks up for $c' = H(m',r')$

  and checks whether $C' = \mathbf{E}(m',r')$

  Problem if $H(m',r')$ not asked: rejection of a valid ciphertext (**BadD**), but with probability 2^{-k}

IND-CCA2: Simulation (4)

Without **AskG**, **AskH** or **BadD:** perfect simulation

New event **ASK**: $G(r)$ or $H(*,r)$

$$\begin{aligned}\Pr_0[\beta' = \beta] &\leq \Pr_1[\beta' = \beta] \\ &\quad + \Pr_1[\mathbf{AskG} \vee \mathbf{AskH}] + \Pr_1[\mathbf{BadD}] \\ &\leq \Pr_1[\beta' = \beta] + \Pr_1[\mathbf{ASK}] + q_{\mathbf{D}}\, 2^{-k}\end{aligned}$$

IND-CCA2: Extraction (5)

Without **ASK** adversary *A* has no information

$$\begin{aligned}\Pr_1[\beta' = \beta] &= \Pr_1[\beta' = \beta \mid \mathbf{ASK}]\, \Pr_1[\mathbf{ASK}] \\ &\quad + \Pr_1[\beta' = \beta \mid \neg\mathbf{ASK}]\, \Pr_1[\neg\mathbf{ASK}] \\ &\leq \Pr_1[\mathbf{ASK}] + \tfrac{1}{2}\end{aligned}$$

$$\begin{aligned}\mathrm{Succ}^{ow}(t') &\geq \Pr_1[\mathbf{ASK}] \geq \Pr_1[\beta' = \beta] - \tfrac{1}{2} \\ &\geq \Pr_0[\beta' = \beta] - \tfrac{1}{2} - \Pr_1[\mathbf{ASK}] - q_{\mathbf{D}}\, 2^{-k}\end{aligned}$$

$$\begin{aligned}\tfrac{1}{2}\,\mathrm{Adv}^{cca}(t) &\leq \Pr_0[\beta' = \beta] - \tfrac{1}{2} \\ &\leq 2\, \mathrm{Succ}^{ow}(t') + q_{\mathbf{D}}\, 2^{-k}\end{aligned}$$

where $t' = t + (q_G + q_H)\, T_f$

IND-CCA2: Result (6)

$$\boxed{\mathrm{Adv}^{cca}(t) \leq 4\mathrm{Succ}_f^{ow}(t + (q_G + q_H)T_f) + \frac{2q_{\mathbf{D}}}{2^k}}$$

If the parameters are properly chosen so that *f* is indeed hard to invert, the encryption scheme is semantically secure against any CCA-adversary, in the random oracle model