

# Provably Secure and Efficient Cryptography

Tsuyoshi TAKAGI

TU Darmstadt

ttakagi@cdc.informatik.tu-darmstadt.de  
<http://www.informatik.tu-darmstadt.de/TI/>

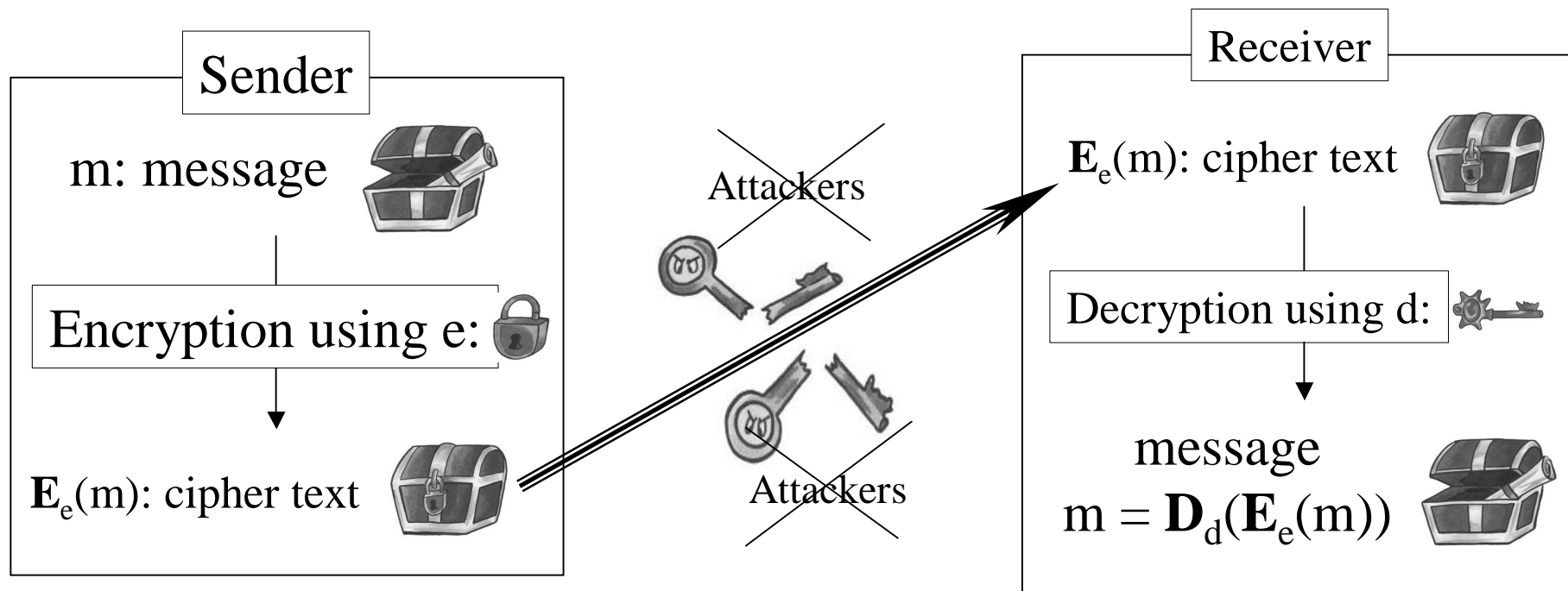
# Contents

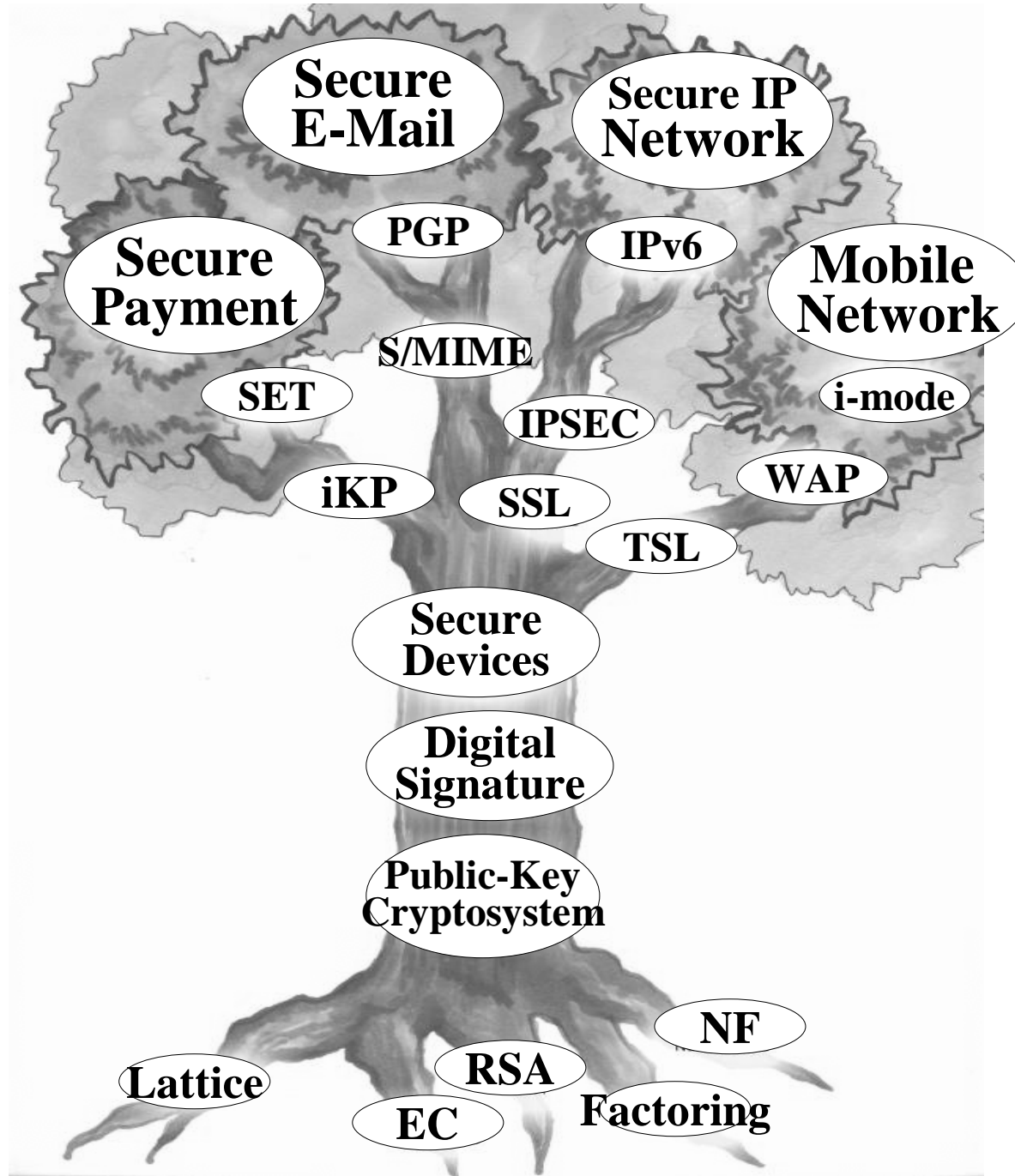
- Overview
- NICE Cryptosystem
- Provable Security
- New Practical Attacks

# Public-Key Cryptosystem

( $e$ : public key ,  $d$ : secret key ) of the receiver

$E_e$ : encryption function,  $D_d$ : decryption function,  $m = D_d(E_e(m))$





17.05.2003

# RSA Cryptosystem

de facto standard of public-key cryptosystems

$p, q$ : primes,  $n = pq$ ,  $ed = 1 \pmod{(p-1)(q-1)}$ ,

$e, n$ : public key,  $d$ : secret key, (factoring,  $n$ : 1024 bits)

$M$ : message,  $M \in \{0, 1, 2, \dots, n-1\}$ .

**Encryption:  $C = M^e \pmod n$**

$e$ : small ( $2^{16}+1$ )

**Decryption:  $M = C^d \pmod n$**

$d$ : large ( $d > n^{1/2}$ )

# Fast Exponentiation

The binary representation of  $d = d[k-1]2^{k-1} + d[k-2]2^{k-2} + \dots + d[1]2^1 + d[0]2^0$ , where  $d[k-1]=1$ .

## Left-to-right binary method

**Input**  $C, n, d$

**Output**  $C^d \bmod n$

$X = C;$

For  $i=k-2$  to  $0$

$X = X^2 \bmod n;$

if  $d[i]=1$ , then  $X=X*C \bmod n;$

**Return**  $X$

cubic complexity  $O((\log n)^3)$ .  
 - we need about 1500 modular multiplications for 1024-bit  $n, d$  on average.

$d =$  179769313486231590772930519078902473361797697894230657273430081157732639445209167262771634937140456477800995856  
 4863673560357494227785840418926558467439899258695049140360821770965996851973903412635215659390188627764072341203  
 1668285970266526289737711820513944871376325649575655785893257302729658745304709432808

# Efficiency is Important

## Smart Cards

- tamper-resistant
- mobility



➔ We need efficient cryptographic algorithms

# How to improve efficiency?

## (1) Software technique

- optimization of compiler, memory management

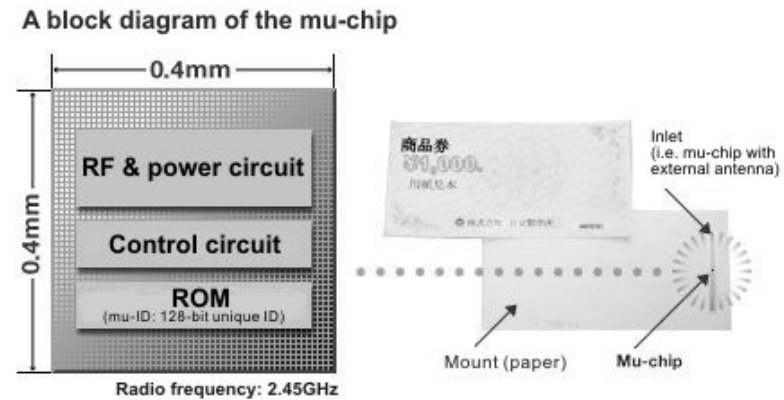
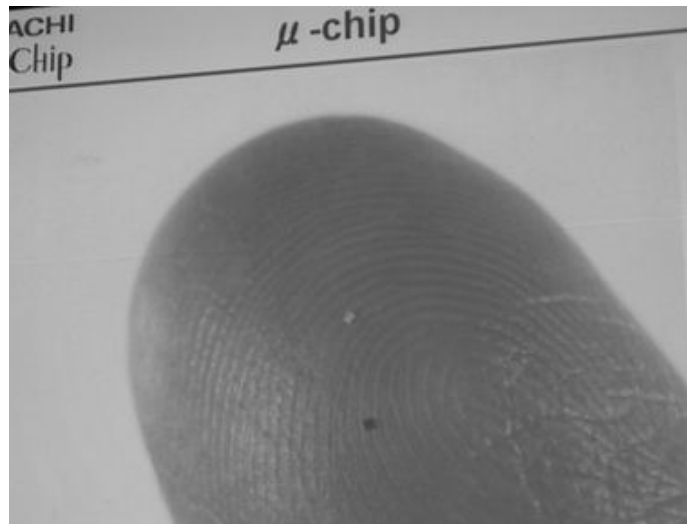
## (2) Hardware technology

- improvement of CPU, memory (RAM, ROM) access

## (3) Mathematical improvements

- developing efficient algorithms

# $\mu$ -chip (Hitachi)



Contact-less chip card,  $0.4 \times 0.4 \text{ mm}^2$ , Radio Frequency 2.45 GHz, 128-bit ROM.



RFID (Radio Frequency Identification)  
Ubiquitous Computing, Pervasive Computing

# Other Standard PKCs

## (1) EPOC, ESIGN

- factoring based cryptosystems, provably secure

## (2) (Hyper-) Elliptic Curve Cryptosystem

- discrete logarithm problem of elliptic curve over finite fields

## (3) NTRU

- lattice reduction based cryptosystem

## (4) SFLASH

- hidden monomial problem

# Three Classes

## Cryptanalysis

- factoring (number field sieve, elliptic curve method)
- discrete logarithm problem of EC (rho method, BSGS)
- SVP/CVP (lattice reduction problem)
- other weak keys (Silver-Pohlig-Hellmann, MOV-FR reduction)

## Key Generation

- primality test (Miller-Rabin, Agrawal-Kayal-Saxena, etc)
- order counting algorithm of elliptic curves over finite fields (CM method, Schoof method, Satoh method, etc)

## Encryption & Decryption

- modular multiplication (Montgomery multiplication, etc)
- addition chain (window based method, etc)
- addition formula of ECC (Jacobian coordinate, etc)

# SECG (Standards for Efficient Cryptography Group)

secp160r1

```

p = FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF 7FFFFFFF
a = FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF 7FFFFFFF
b = 1C97BEFC 54BD7A8B 65ACF89F 81D4D4AD C565FA
x = 4A96B568 8EF57328 46646989 68C38BB9 13CBFC
y = 23A62855 3168947D 59DCC912 04235137 7AC5FE
n = 01 00000000 00000000 0001F4C8 F927AED3 CA7
h = 01
    
```

# NICE Cryptosystem

- NICE cryptosystem is constructed over class groups of quadratic discriminants  $Cl(D)$ , where  $D = -pq^2$ .
- NICE cryptosystem is based on the factoring problem.
- Decryption time is of quadratic complexity  $O((\log D)^2)$ .

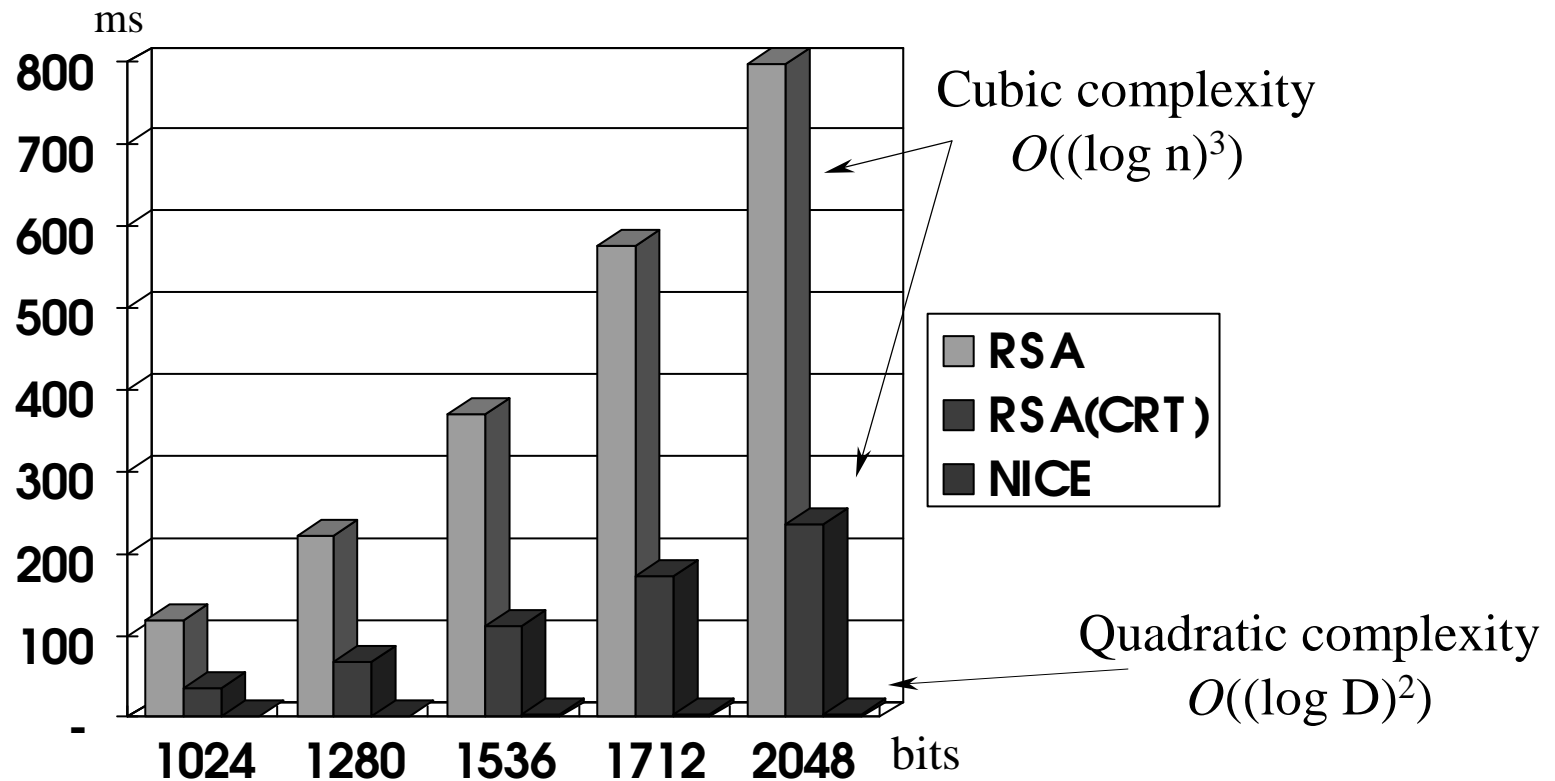
## Publications:

Journal of Cryptology'00, Eurocrypt'98, CHES'99,  
Mathematics of Public-Key Cryptography '99, ICISC'01

## Related works:

Asiacrypt'99, PKC'00, SAC'00, CT-RSA'01, ACISP'02.

# Efficiency of Quadratic Complexity



The decryption of NICE is fast even for large keys.

1.7 ms for 1024-bit public-keys.

4.3 ms for 2048-bit public-keys.

# Encryption and Decryption Algorithm

## RSA

integer ring  $\mathbf{Z}/n\mathbf{Z}$ ,  $n = pq$   
 $a \bullet \mathbf{Z}/n\mathbf{Z}$ ,  $a = 0, 1, 2, \dots, n-1$ .

Encryption: message  $M \bullet \mathbf{Z}/n\mathbf{Z}$   
 $C = M^e \text{ mod } n$ , for exponent  $e$ .

Decryption:  $M = C^d \text{ mod } n$ .

## NICE

class group  $Cl(D)$ ,  $D = -pq^2$   
 $(a,b) \bullet Cl(D)$ ,  $a,b = 0, 1, 2, \dots, D^{1/2}$ .

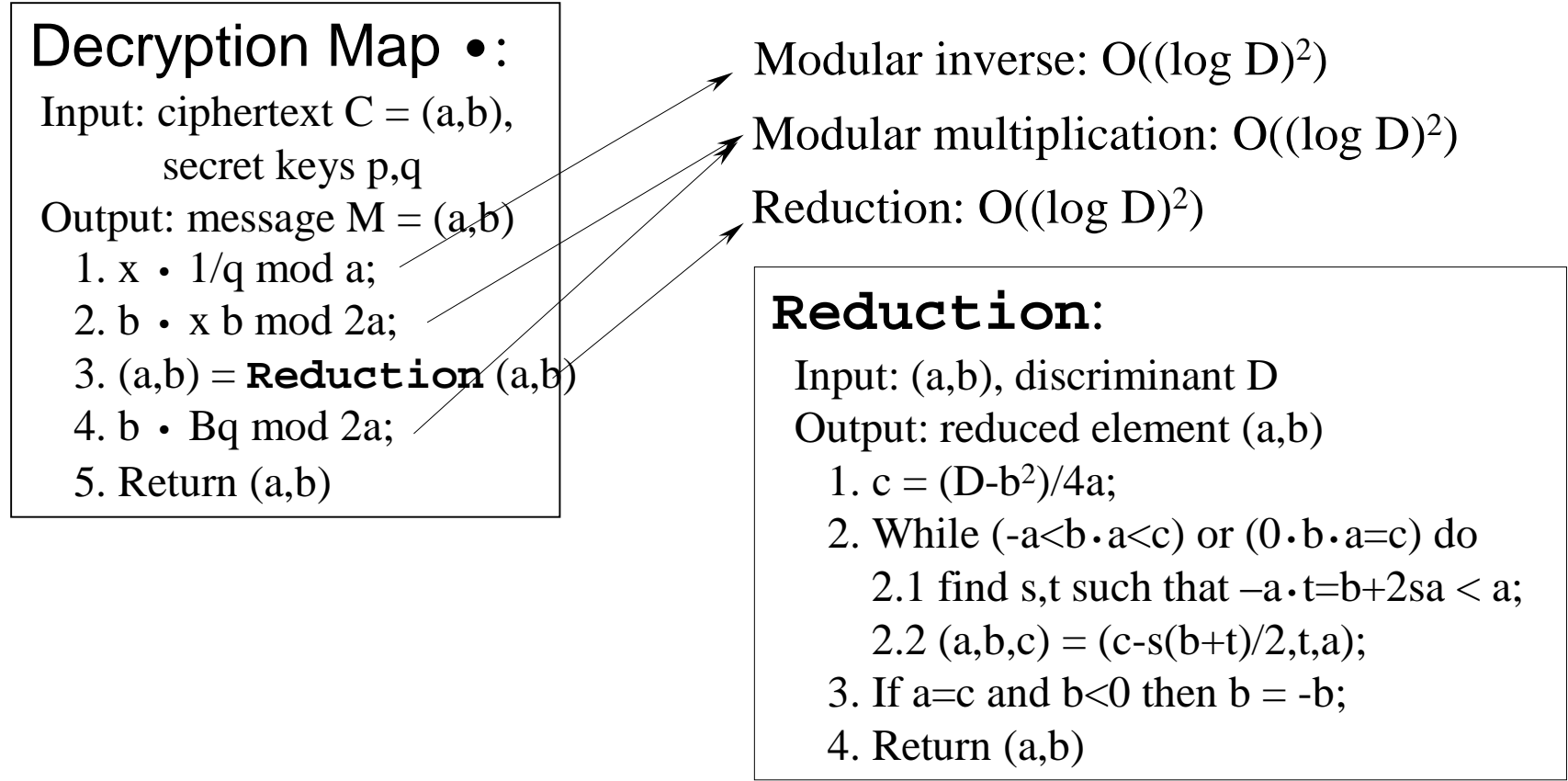
Encryption: message  $M \bullet SI(D)$   
 $C = MQ$ , for random  $Q \bullet Ker(D)$ .

Decryption:  $M = \bullet(C)$ .

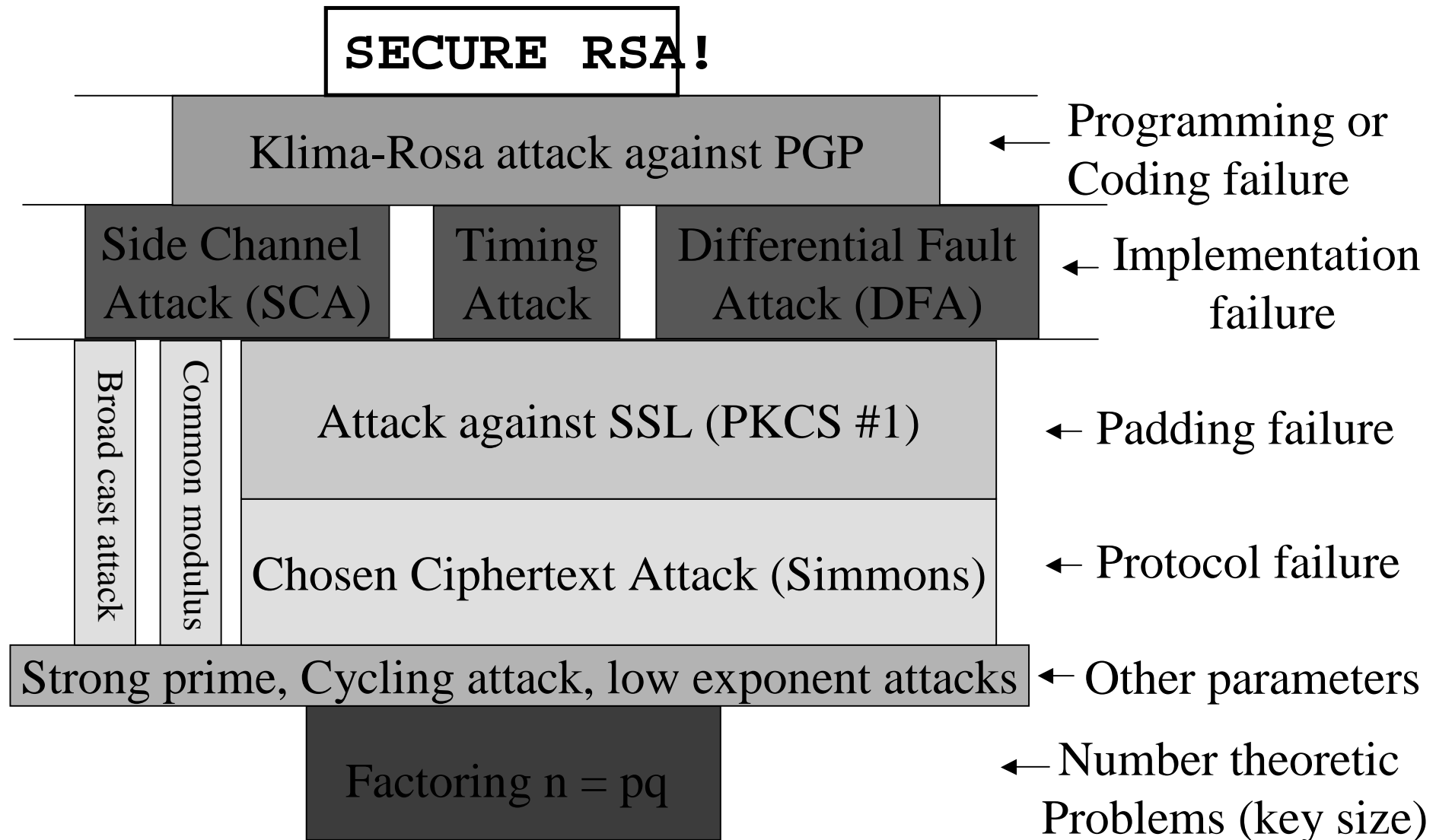
$Ker(D) =$  the kernel of map  $Cl(D) \rightarrow Cl(-p)$

Security of NICE is based on Hidden Kernel Problem (HKP).

# Why Quadratic Complexity $O((\log D)^2)$ ?



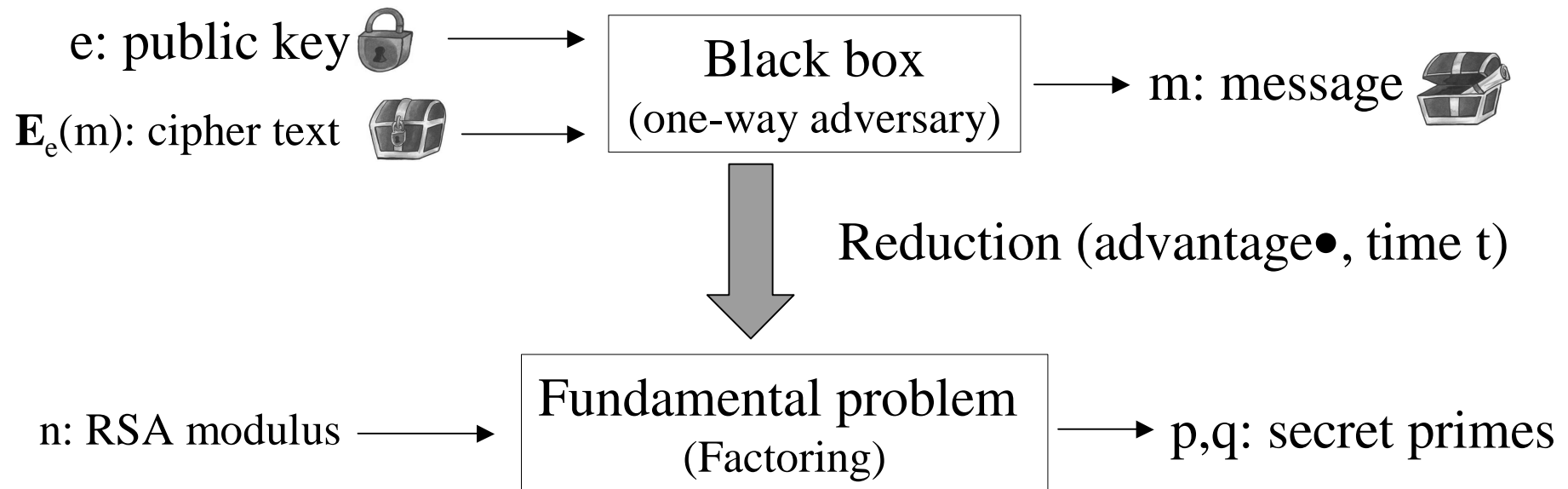
# Why Provable Security?



# What is Provably Secure?

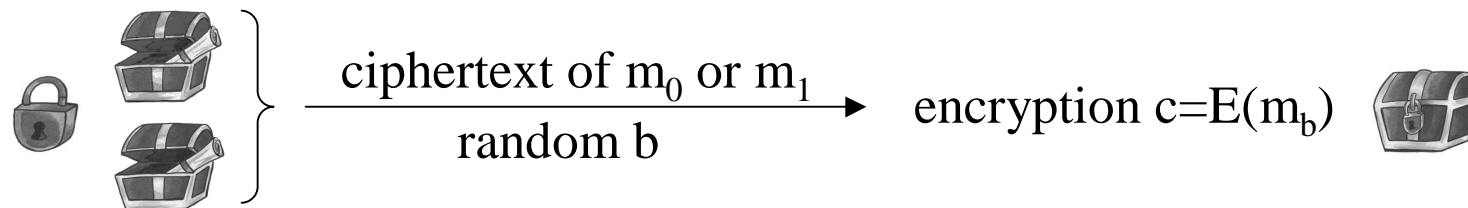
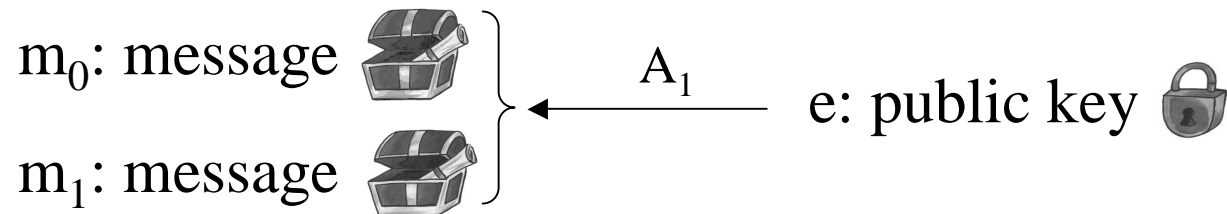
Methodology of the proof:

- To define the model of an attack (black box)
- To reduce the attack to a fundamental problem
- To estimate the advantage and time of the reduction



# Semantic Security

(1) Algorithm  $A_1$ , on input  $pk$ , finds two message  $m_0, m_1$  (find stage).



(2) Algorithm  $A_2$ , on input  $m_0, m_1, c = E(m_b)$ , guesses  $b$  (guess stage).



$A$  is allowed to ask the decryption oracle during this game.

# NICE-X Cryptosystem

Hash functions,  $g: \text{Ker} \bullet \{0,1\}^{512}$ ,  $h: \{0,1\}^{512} \times \text{Ker} \bullet \{0,1\}^{160}$ .

## Encryption:

random  $R \bullet \text{SI}(D)$ , random  $Q \bullet \text{Ker}(D)$ ,

$C = RQ$ ,  $B = m \oplus g(Q)$ ,  $H = h(m, Q)$

$(C, B, H)$  is the cipher text of a message  $m \bullet \{0,1\}^{512}$

## Decryption:

$R = \bullet(C)$ ,  $Q = C R^{-1}$ ,  $m = B \oplus g(Q)$ ,

Check  $H = h(m, Q)$ , if not reject.

NICE-X is semantically secure under the hardness of HKP.

# New Attacks against Semantically Secure PKC

- Timing Attack '96
- Side Channel Attacks '97
- Differential Fault Attacks '98
- Klima-Rosa Attack against PGP '99
- Reject Timing Attack '01
- Exceptional Procedure Attacks '02
- Zero-Value Point Attacks '03

# RSA with CRT

Algorithm RSA\_CRT\_Decryption

Input  $C, n, p, q, dp, dq, p\_inv\_q$

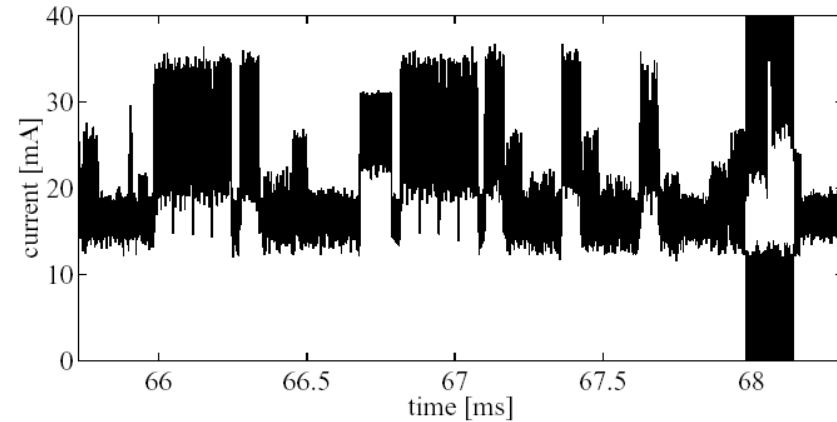
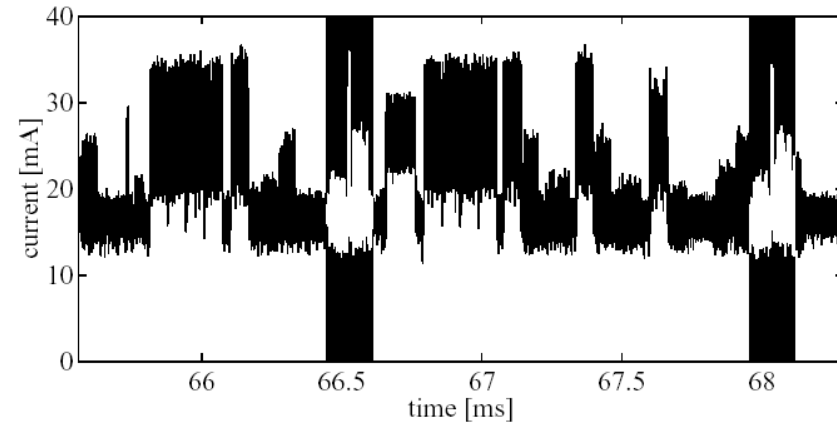
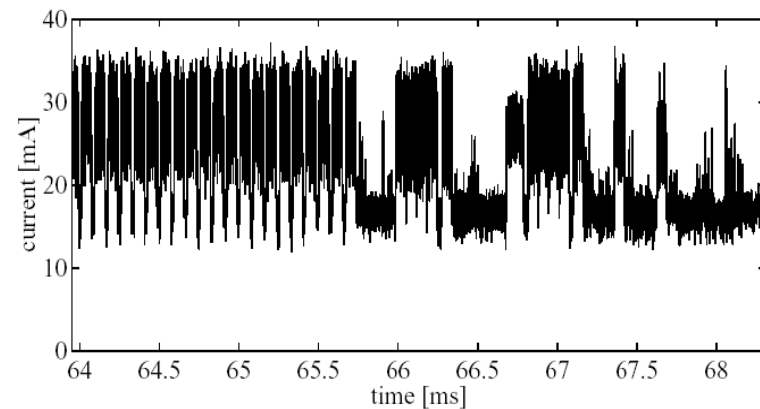
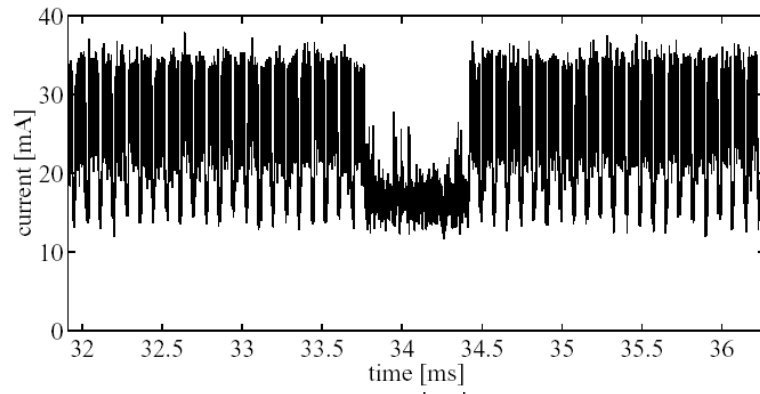
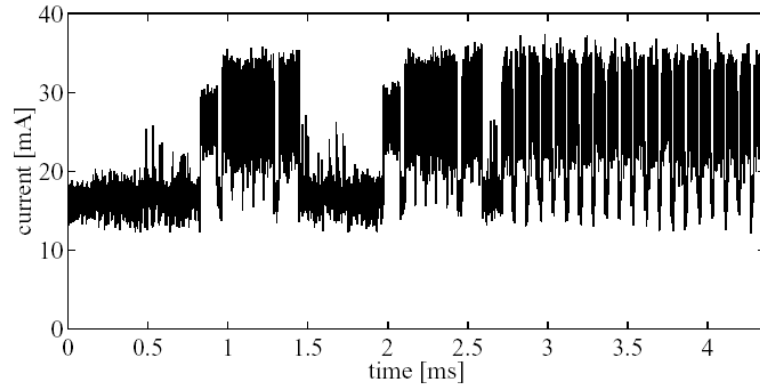
Output  $M$

- 1:  $M_p = C^{dp} \bmod p$ ;
- 2:  $M_q = C^{dq} \bmod q$ ;
- 3:  $v = (M_q - M_p) p\_inv\_q \bmod q$ ;
- 4:  $M = M_p + pv$ ;
- 5: Return  $M$

PKCS #1 – RSA Cryptography Standard

<http://www.rsasecurity.com/rsalabs/pkcs/pkcs-1/index.html>

# Power Consumption



Cited from the paper: R. Novak, "SPA-Based Adaptive Chosen Ciphertext Attack on RSA Implementation," PKC 2002, LNCS 2274, pp.252-262, 2002.