CS557: Cryptography

Public-key Cryptography-III

S. Tripathy
IIT Patna

RSA Security

- Three major approaches to attacking RSA:
  - brute force key search:
    - infeasible given size of numbers
  - mathematical attacks (based on difficulty of computing ø(N), by factoring modulus N)

- mathematical approach to find d takes 3 forms:
  - factor N=p.q, hence find ø(N) and then d
  - determine ø(N) directly and find d
  - find d directly
- timing attacks (on running of decryption)

Complexity of Factoring Problem

- Trial division
  - Complexity √n
- Pollard p-1 method

input : an integer n , and a prespecified "bound" B

output : factors of n

$$a \leftarrow 2$$

$$\text{for } \; j \leftarrow 2 \; \text{ to } \; B$$

$$\quad \text{do } \; a \leftarrow a^j \mod n$$

$$d \leftarrow \gcd(a-1, n)$$

$$\text{if } \; 1 < d < n$$

$$\quad \text{then } \; \text{return}(d)$$

$$\quad \text{else } \; \text{return}(\text{"} \mathit{failure} \text{"})$$

The Pollard's rho algorithm

- 2. The Pollard's rho algorithm

input： an integer *n*

output：factors of *n*

(1) Selecting a "random" function *f* with integer coefficients , and any $x_0 \in Z_n$.

Begin with x=x_0 and y=y_0.

(2) Repeat the two calculations

$$x \leftarrow f(x) \,\mathbf{mod}\, n \quad \mathbf{and} \quad y \leftarrow f(f(y)) \,\mathbf{mod}\, n$$

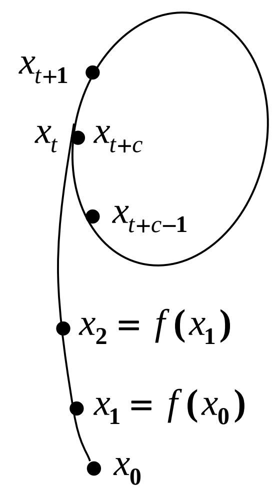
until d=gcd(x-y,n)>1.

(3) Do the following compare

3.1 If d<n, we have succeeded.

3.2 If d=n, the method is failed. Goto (1).

(*) A typical choice of f(x)=x^2+1, with a seed x_0=2.



Pollard Rho method for factorization

Introduction:

Let us prepare a sequence (S) as follows

$$S = \begin{cases} x_0 & \text{initialize a value} \quad i = 0 \\ x_i & x_i = f(x_{i-1}) \bmod n \quad i > 0 \end{cases}$$

If we find x_i, x_j s.t. $p \mid (x_i - x_j)$
as $p \mid n$

$(x_i - x_j) \mid n$

i.e. for any pair

$d \leftarrow \gcd((x_i - x_j), n) > 1$

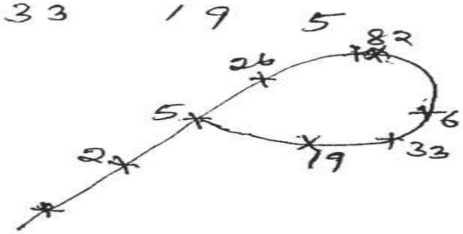
$\Rightarrow d$ is a factor $= p$.

$\boxed{f(z) = z^2 + 1}$ can be chosen

Ex: $N = 119$, $x_0 = 2$

i	0	1	2	3	4	5	6	7
x_i	2	5	26	82	61	33	19	5

The value seems like a ρ



To reduce the no. of GCD computations you can use Floyd's cycle detection.

choose $x_0, y_0 \leftarrow \mathbb{Z}_n$

$x_i = f(x_i)$

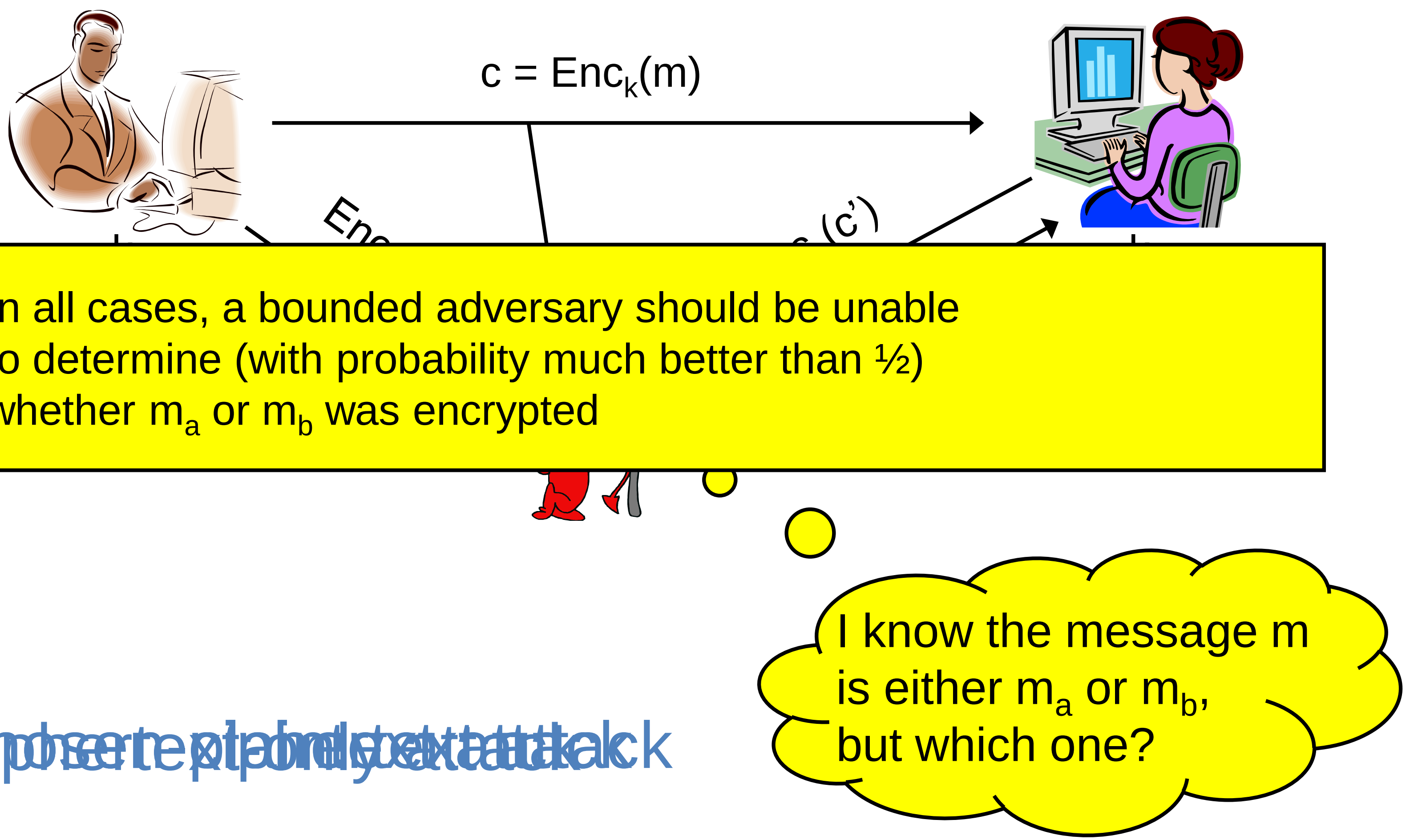
$y_i = f(f(x_i))$

if $d = \gcd((x_i - y_i), n) > 1$ return d

$\boxed{\gcd(61 - 26, 119) = 7}$

i	0	1	2	3
x_i	2	5	$\boxed{26}$	82
y_i	2	26	$\boxed{61}$	19

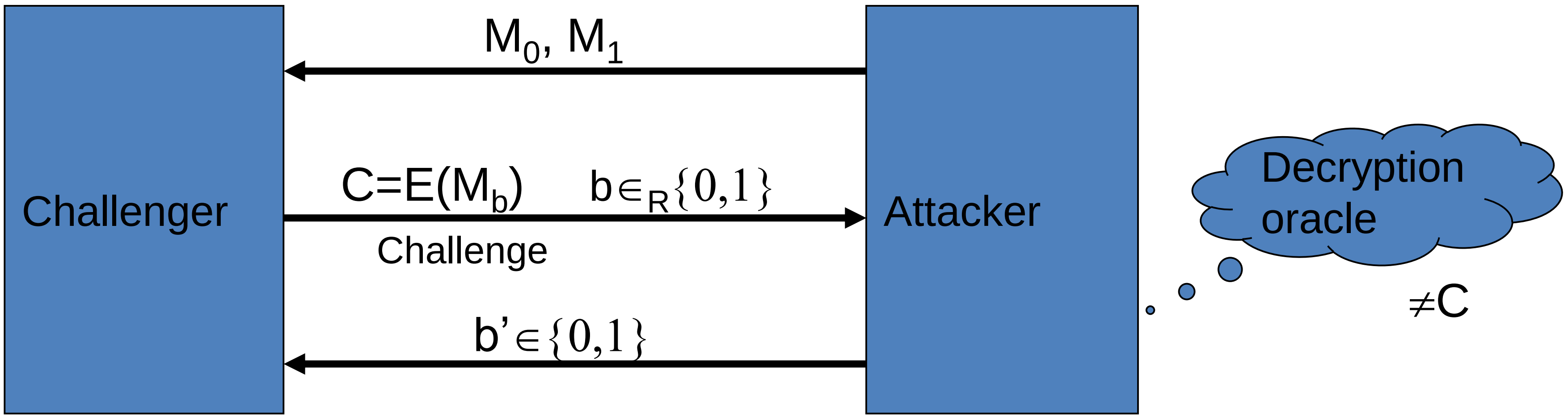
Cipher text only, CPA and CCA



In all cases, a bounded adversary should be unable to determine (with probability much better than ½) whether m_a or m_b was encrypted

Chosen ciphertext security (CCS)

- No efficient attacker can win the following game:
  (with non-negligible advantage)



Attacker wins if b=b'

- Chosen ciphertext:

Attacker intercepts the ciphertext c (A→B)

and compute $y = c*r^e \bmod n$ chosing a random r. Send Y to B.

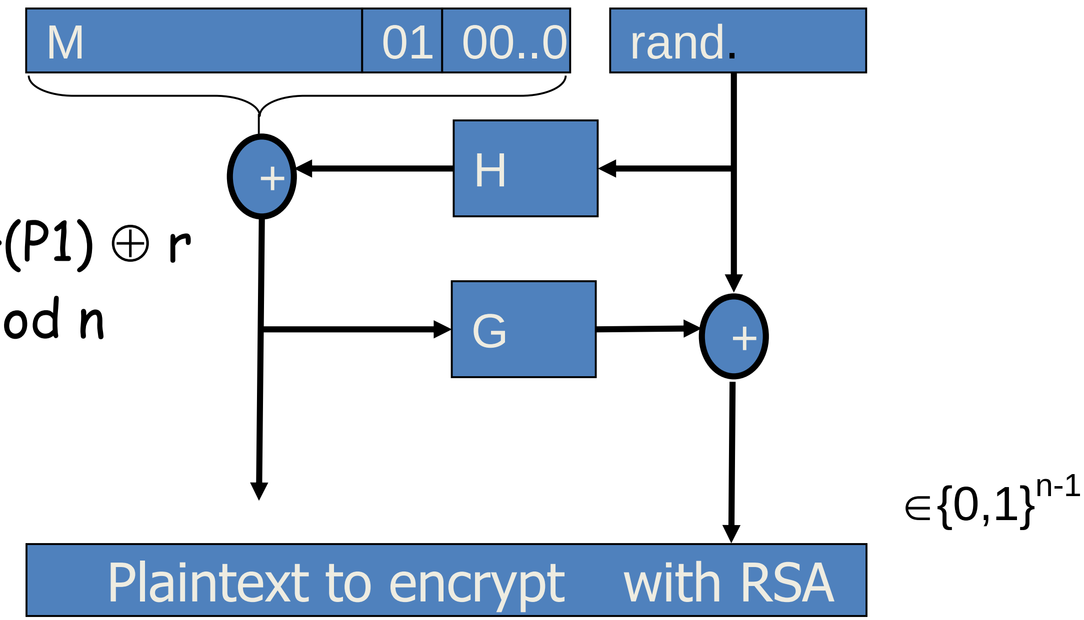
B decrypts y as $z = y^d \bmod n = M.r \bmod n$ sends to the attacker.

Attacker can find M from z easily as he knows r and hence r^{-1}.

- Attacks on smaller exponent:
  - If system uses smaller value of e (=3 say) for simpler it is easier to solve for d and obtaining the plain text
  - If attacker can obtain 3 different ciphers (c1, c2, c3) of same plain text (P) with different modulus.
    - $C1 = P^3 \bmod n1$, $C2 = P^3 \bmod n2$ and $C3 = P^3 \bmod n3$
    - $C' = c1.c2.c3 = P^3 \bmod n1.n2.n3$ (now $P^3 < n1.n2.n3$)
    - $P = (C')^{1/3}$
  - Related Message attack: if two linearly related plain texts (P1 and P2) are enciphered attacker can retrieve P2 if P1 is known

OAEP

- New preprocessing function: OAEP (BR94).



$P1 = M \oplus H(r) \qquad P2 = G(P1) \oplus r$

$C = (P1 \,||\, P2)^e \bmod n$

Thm: RSA is trap-door permutation $\Rightarrow$ OAEP is CCS
when H,G are "*random oracles*".

Timing Attack

- developed in mid-1990's
- exploit timing variations in operations
  - eg. multiplying by small vs large number
  - or Integer Factors varying which instructions executed
- infer operand size based on time taken
- RSA exploits time taken in exponentiation
- countermeasures
  - use constant exponentiation time
  - add random delays
  - blind values used in calculations

Notes on RSA

- Too smaller e is undesirable
- If $d< (n^{1/4})/3$ then d can be efficiently computed from e and n
- If $n=p.q$ is of t digits, knowing $t/4$ digits of p one can factor n efficiently
- Sharing the modulus is bed

Key lengths

- Security of public key system should be comparable to security of block cipher.

NIST:

Cipher key-size	Modulus size
$\leq$ 64 bits	512 bits.
80 bits	1024 bits
128 bits	3072 bits.
256 bits (AES)	**15360** bits

- High security $\Rightarrow$ very large moduli.

ELGAMAL PUBLIC KEY CRYPTOGRAPHY
BASED ON
DIFFIE HELLMAN KEY EXCHANGE

Diffie-Hellman key exchange (**D-H**) is a cryptographic protocol that allows two parties that have no prior knowledge of each other to jointly establish a shared secret key over an insecure communications channel. This key can then be used to encrypt subsequent communications using a symmetric key cipher.

Diffie-Hellman Key Exchange

- Public: prime (1024 bit), generator g of group Z_p^*

Alice		Bob
Pick random $1 \le r_A \le p-1$	$\xrightarrow{g^{r_A}}$	Pick random $1 \le r_B \le p-1$
Output: $g^{r_A r_B}$	$\xleftarrow{g^{r_B}}$	Output: $g^{r_A r_B}$
Erase r_A		Erase r_B

Diffie-Hellman problem: Obtain g^{ab} with given (g^a, g^b)

Easy if we can compute x from g^x
No better way known
implicit key authentication (only if attacker is passive)

CDH and DDH

- Discrete Log problem
  - Given y and g in Zp where p is prime, find the unique x in Zp , such that $y = g^x \bmod p$.
  - No efficient algorithm
- Computational Diffie-Hellman (CDH)
  - Given a multiplicative group (G, *), an element $g \in G$ having order q, given g^x and g^y, find g^xy
- Decision Diffie-Hellman (DDH)
  - Given a multiplicative group (G, *), an element $g \in G$ having order q, given g^x, g^y, and g^z, determine

    if g^xy $\equiv$ g^z mod n

- Discrete Log is at least as hard as CDH, which at least as hard as DDH

- Thanks