RSA Cryptosystem

Dr. Arup Kumar Pal
Department of Computer Science & Engineering
Indian Institute of Technology (ISM) Dhanbad
Jharkhand-826004
E-mail: cryptography202021@gmail.com

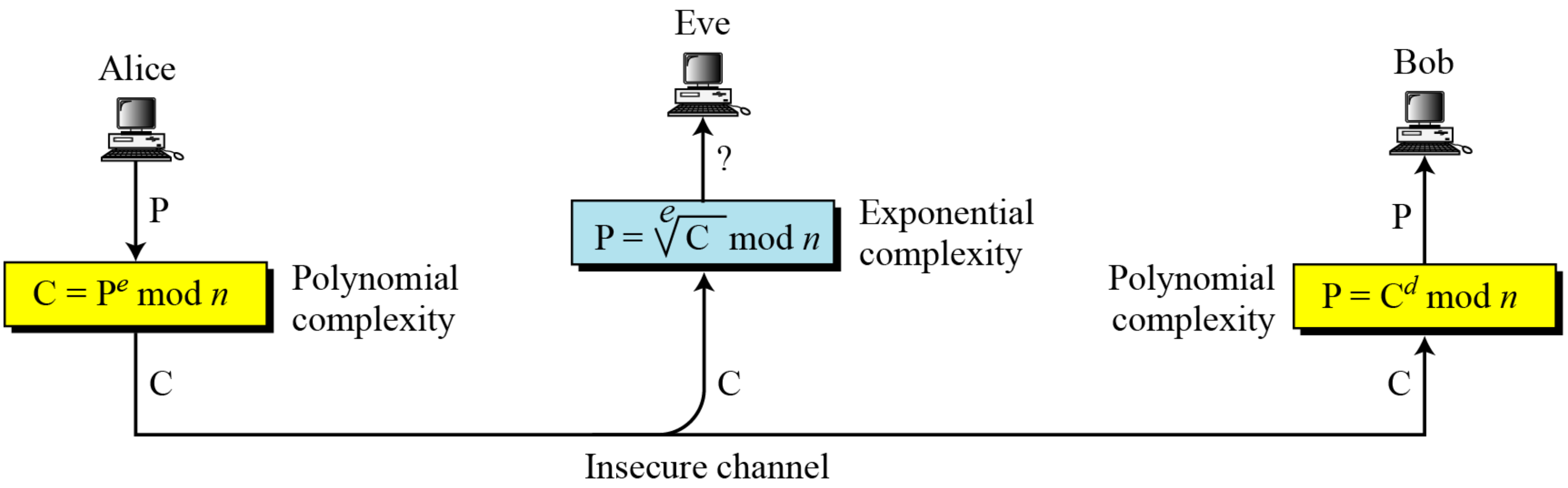
Outline

- To discuss the concepts of RSA cryptosystem
- Practical aspects of RSA
- Attacks on RSA
- Implementation aspects of RSA
- Conclusions

Introduction

- RSA encryption is not meant to replace symmetric ciphers because it is several times slower than ciphers such as AES.
- The main use of the RSA encryption feature is to securely exchange a key for a symmetric cipher.
- RSA is often used together with a symmetric cipher such as AES, where the symmetric cipher does the actual bulk data encryption.
- The underlying one-way function of RSA is the integer factorization problem.

Complexity of operations in RSA



Encryption and Decryption

- Bob chooses two distinct large primes p, and q; and computes n=pq.
- Bob chooses e such that $1 < e < \phi(n)$ and $\gcd(e, \phi(n)) = 1.$
- Bob computes d with $de \equiv 1 \bmod \phi(n).$
- Bob makes n and e public, and keeps d secret.
- Alice encrypts m as $c \equiv m^e \bmod n$ and sends c to Bob.
- Bob decrypts by computing $m \equiv c^d \bmod n$

Public Key: (e, n) and Private Key: (d)

Example

Alice

message $x = 4$

Bob

1. choose $p = 3$ and $q = 11$
2. $n = p \cdot q = 33$
3. $\Phi(n) = (3-1)(11-1) = 20$
4. choose $e = 3$
5. $d \equiv e^{-1} \equiv 7 \bmod 20$

$$\xleftarrow{k_{pub}=(33,3)}$$

$y = x^e \equiv 4^3 \equiv 31 \bmod 33$

$$\xrightarrow{y=31}$$

$y^d = 31^7 \equiv 4 = x \bmod 33$

Contd...

- In practice m, c, n and d are very long numbers, usually 1024 bit long or more.
- The value **e** is referred to as **encryption exponent or public exponent**, and the private key **d** is called **decryption exponent or private exponent.**
- If Alice wants to send an encrypted message to Bob, Alice needs to have Bob's public key (n,e), and Bob decrypts with his private key d.

Proof of Correctness

$$\text{In RSA}: n = pq$$

$$ed \equiv 1 \bmod \phi(n)$$

$$\Rightarrow ed = k\phi(n) + 1$$

$$\text{Encryption}: C = P^e \bmod n$$

$$\text{Decryption}: P = C^d \bmod n$$

$$\text{Now}: C^d \bmod n$$

$$= P^{ed} \bmod n$$

$$= P^{k\phi(n)+1} \bmod n.$$

Contd...

Euler's theorem:

If a and n be integers with gcd(a,n) = 1, then:

$$a^{\Phi(n)} \equiv 1 \pmod{n}$$

Case 1:

If P and n are co-prime then

$$P^{k\phi(n)+1} \bmod n$$

$$= P \times P^{k\phi(n)} \bmod n$$

$$= P \bmod n$$

Contd...

Case 2: *If P is a multiple of p but not of q then*

$$\text{Let } P = i \times p$$

$$P^{k\phi(n)} \bmod q$$

$$= P^{k\phi(p)\phi(q)} \bmod q$$

$$= 1$$

$$\Rightarrow P^{k\phi(n)} = jq + 1$$

Contd...

$$C^d \bmod n$$
$$= P^{k\phi(n)+1} \bmod n$$
$$= P \times P^{k\phi(n)} \bmod n$$
$$= P \times (jq+1) \bmod n$$
$$= \left(jqP + P \right) \bmod n$$
$$= \left(jqip + P \right) \bmod n$$
$$= \left(jin + P \right) \bmod n$$
$$= P \bmod n$$

Fast Exponentiation

- How many multiplications are required to compute the simple exponentiation x^8?

The straightforward method:

$$x \xrightarrow{SQ} x^2 \xrightarrow{MUL} x^3 \xrightarrow{MUL} x^4 \xrightarrow{MUL} x^5 \xrightarrow{MUL} x^6 \xrightarrow{MUL} x^7 \xrightarrow{MUL} x^8$$

Alternatively, we can do something faster:

$$x \xrightarrow{SQ} x^2 \xrightarrow{SQ} x^4 \xrightarrow{SQ} x^8$$

Two basic operations:

SQ: squaring the current result,

MUL: multiplying the current result by the base element x.

Contd...

- How many multiplications are required to compute the simple exponentiation x^{26}?

  The straightforward method: requires 25 multiplications.

Alternatively, we can do something faster:

$$x \xrightarrow{SQ} x^2 \xrightarrow{MUL} x^3 \xrightarrow{SQ} x^6 \xrightarrow{SQ} x^{12} \xrightarrow{MUL} x^{13} \xrightarrow{SQ} x^{26}.$$

Square-and-multiply algorithm

- The algorithm is based on scanning the bit of the exponent from the left (the MSB) to the right (the LSB).
- In every iteration, i.e., for every exponent bit, the current result is squared.
- If and only if the currently scanned exponent bit has the value 1, a multiplication of the current result by x *is executed following the squaring.*

Example

Example We again consider the exponentiation x^{26}. For the square-and-multiply algorithm, the binary representation of the exponent is crucial:

$$x^{26} = x^{11010_2} = x^{(h_4h_3h_2h_1h_0)_2}.$$

The algorithm scans the exponent bits, starting on the left with h_4 and ending with the rightmost bit h_0.

Step	
#0 $x = x^{\mathbf{1}_2}$	inital setting, bit processed: $h_4 = 1$
#1a $(x^1)^2 = x^2 = x^{\mathbf{10}_2}$	SQ, bit processed: h_3
#1b $x^2 \cdot x = x^3 = x^{10_2}x^{1_2} = x^{\mathbf{11}_2}$	MUL, since $h_3 = 1$
#2a $(x^3)^2 = x^6 = (x^{11_2})^2 = x^{\mathbf{110}_2}$	SQ, bit processed: h_2
#2b	no MUL, since $h_2 = 0$
#3a $(x^6)^2 = x^{12} = (x^{110_2})^2 = x^{\mathbf{1100}_2}$	SQ, bit processed: h_1
#3b $x^{12} \cdot x = x^{13} = x^{1100_2}x^{1_2} = x^{\mathbf{1101}_2}$	MUL, since $h_1 = 1$
#4a $(x^{13})^2 = x^{26} = (x^{1101_2})^2 = x^{\mathbf{11010}_2}$	SQ, bit processed: h_0
#4b	no MUL, since $h_0 = 0$

Attacks on RSA

Factorization Attack: The security of RSA is based on the idea that the modulus is so large that it is not feasible to factor it in a reasonable time.

The attacker attempts to factor n. If this can be done then it is a simple way to compute **ϕ(n)** and subsequently d will be derived.

Chosen-Ciphertext Attack

- Eve chooses a random integer X in Z_n^*.
- Eve calculates $Y=C \times X^e \bmod n$
- Eve sends Y to Bob for decrypting and get $Z=Y^d \bmod n$.
- Eve can easily find P because…

Attacks on Encryption Exponent

- To reduce the encryption time, it is tempting to use a small encryption exponent e.
- The common value is considered as e=3.
- In **Broadcast Attack,** If one entity sends the same message to a group of recipients with same low encryption exponent.
- If the public exponent, e=3 and the moduli=n_1,n_2, and n_3 then the problem statement will be:

$$C_1 \equiv P^3 \bmod n_1$$

$$C_2 \equiv P^3 \bmod n_2$$

$$C_3 \equiv P^3 \bmod n_3$$

Chinese Remainder Theorem (CRT)

Chinese Remainder Theorem is used to solve a set of congruent equations with one variable but different moduli, which are relatively prime then the equations have a unique solution.

$$x \equiv 2 \bmod 3$$

$$x \equiv 3 \bmod 5$$

$$x \equiv 2 \bmod 7$$

Step1: $\mathrm{M}=3 \times 5 \times 7=105$

Step2: $\mathrm{M}_1 = \frac{105}{3} = 35; \mathrm{M}_2 = \frac{105}{5} = 21; \mathrm{M}_3 = \frac{105}{7} = 15$

Step3:The inverses are $\mathrm{M}_1^{-1} = 2$ w.r.t.3; $\mathrm{M}_2^{-1} = 1$ w.r.t.5; $\mathrm{M}_3^{-1} = 1$ w.r.t.7;

$$x \equiv (2 \times M_1 \times \mathrm{M}_1^{-1} + 3 \times M_2 \times \mathrm{M}_2^{-1} + 2 \times M_3 \times \mathrm{M}_3^{-1}) \bmod 105 \equiv 23 \bmod 105$$

Attacks on Encryption Exponent

- In **Broadcast Attack,** the problem statement will be:

$$C_1 \equiv P^3 \bmod n_1$$
$$C_2 \equiv P^3 \bmod n_2$$
$$C_3 \equiv P^3 \bmod n_3$$

- Applying the CRT, the attacker can find an equation of the form

$$C' \equiv P^3 \bmod n_1 n_2 n_3$$
$$\Rightarrow P^3 < n_1 n_2 n_3$$
$$\Rightarrow P = \sqrt[3]{C'}$$

Low Decryption Exponent Attacks

- Bob may think that using low decryption exponent would make the decryption process faster.
- Wiener showed that if $d < \frac{1}{3} n^{1/4}$ and $q < p < 2q$ then the attacker can factor n in polynomial time.
- In RSA, the recommendation is to have $d \geq \frac{1}{3} n^{1/4}$ to prevent low decryption exponent attack.

Continued Fractions

- To approximate a real number by a rational number.
- For example π=3.14159… may be approximated as 22/7,333/106, 355/113 etc.
- Approximation process: of π=3.14159…
- Floor(π)=3
- 1/0.14159=7.06251 $$3+\frac{1}{7}=\frac{22}{7}$$
- 1/0.06251=15.9966 $$3+\frac{1}{7+\frac{1}{15}}=\frac{333}{106}$$

Contd…

- 1/0.9966=1

$$3+\cfrac{1}{7+\cfrac{1}{15+\cfrac{1}{1}}}=\frac{355}{113}$$

- The last approximate is more accurate.

Contd…

- In general, $$\frac{p_n}{q_n} = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{\ldots + \cfrac{1}{a_n}}}}$$
- Each rational number $\frac{p_k}{q_k}$ gives a better approximation than the preceding rational numbers.

Theorem

If $\left| x - \frac{r}{s} \right| < \frac{1}{2s^2}$ for integers r,s, then $\frac{r}{s} = \frac{p_i}{q_i}$, for some i.

For example, $\left| \pi - \frac{22}{7} \right| \approx 0.001 < \frac{1}{98}$ and $\frac{22}{7} = \frac{p_2}{q_2}$.

Wiener Attack

Theorem. *Suppose p, q are primes with $q < p < 2q$. Let $n = pq$ and let $1 \le d, e < \phi(n)$ satisfy $de \equiv 1 \pmod{(p-1)(q-1)}$. If $d < \frac{1}{3}n^{1/4}$, then d can be calculated quickly (that is, in time polynomial in $\log n$).*

Proof. Since $q^2 < pq = n$, we have $q < \sqrt{n}$. Therefore, since $p < 2q$,

$$n - \phi(n) = pq - (p-1)(q-1) = p + q - 1 < 3q < 3\sqrt{n}.$$

Write $ed = 1 + \phi(n)k$ for some integer $k \ge 1$. Since $e < \phi(n)$, we have

$$\phi(n)k < ed < \frac{1}{3}\phi(n)n^{1/4},$$

so $k < \frac{1}{3}n^{1/4}$. Therefore,

$$kn - ed = k(n - \phi(n)) - 1 < k(n - \phi(n)) < \frac{1}{3}n^{1/4}(3\sqrt{n}) = n^{3/4}.$$

Also, since $k(n - \phi(n)) - 1 > 0$, we have $kn - ed > 0$. Dividing by dn yields

$$0 < \frac{k}{d} - \frac{e}{n} < \frac{1}{dn^{1/4}} < \frac{1}{3d^2},$$

since $3d < n^{1/4}$ by assumption.

Contd...

We now need a result about continued fractions. Recall from Section 3.12 that if x is a positive real number and k and d are positive integers with

$$\left|\frac{k}{d} - x\right| < \frac{1}{2d^2},$$

then k/d arises from the continued fraction expansion of x. Therefore, in our case, k/d arises from the continued fraction expansion of e/n.

Contd...

Eve does the following:

1. Computes the continued fraction of e/n. After each step, she obtains a fraction A/B.

2. Eve uses $k = A$ and $d = B$ to compute $C = (ed - 1)/k$. (Since $ed = 1 + \phi(n)k$, this value if C is a candidate for $\phi(n)$.)

3. If C is not an integer, she proceeds to the next step of the continued fraction.

4. If C is an integer, then she finds the roots r_1, r_2 of $X^2 - (n - C + 1)X + n$. (Note that this is possibly the equation $X^2 - (n - \phi(n) + 1)X + n = (X - p)(X - q)$ from earlier.) If r_1 and r_2 are integers, then Eve has factored n. If not, then Eve proceeds to the next step of the continued fraction algorithm.

Example

Example. Let $n = 1966981193543797$ and $e = 323815174542919$. The continued fraction of e/n is

$$[0; 6, 13, 2, 3, 1, 3, 1, 9, 1, 36, 5, 2, 1, 6, 1, 43, 13, 1, 10, 11, 2, 1, 9, 5]$$
$$= \cfrac{1}{6 + \cfrac{1}{13 + \cfrac{1}{2 + \cfrac{1}{3 + \cfrac{1}{1 + \cfrac{1}{3 + \cfrac{1}{1 + \cdots}}}}}}}.$$

The first fraction is $1/6$, so we try $k = 1, d = 6$. Since d must be odd, we discard this possibility.

By the remark, we may jump to the third fraction:

$$\cfrac{1}{6 + \cfrac{1}{13 + \frac{1}{2}}} = \frac{27}{164}.$$

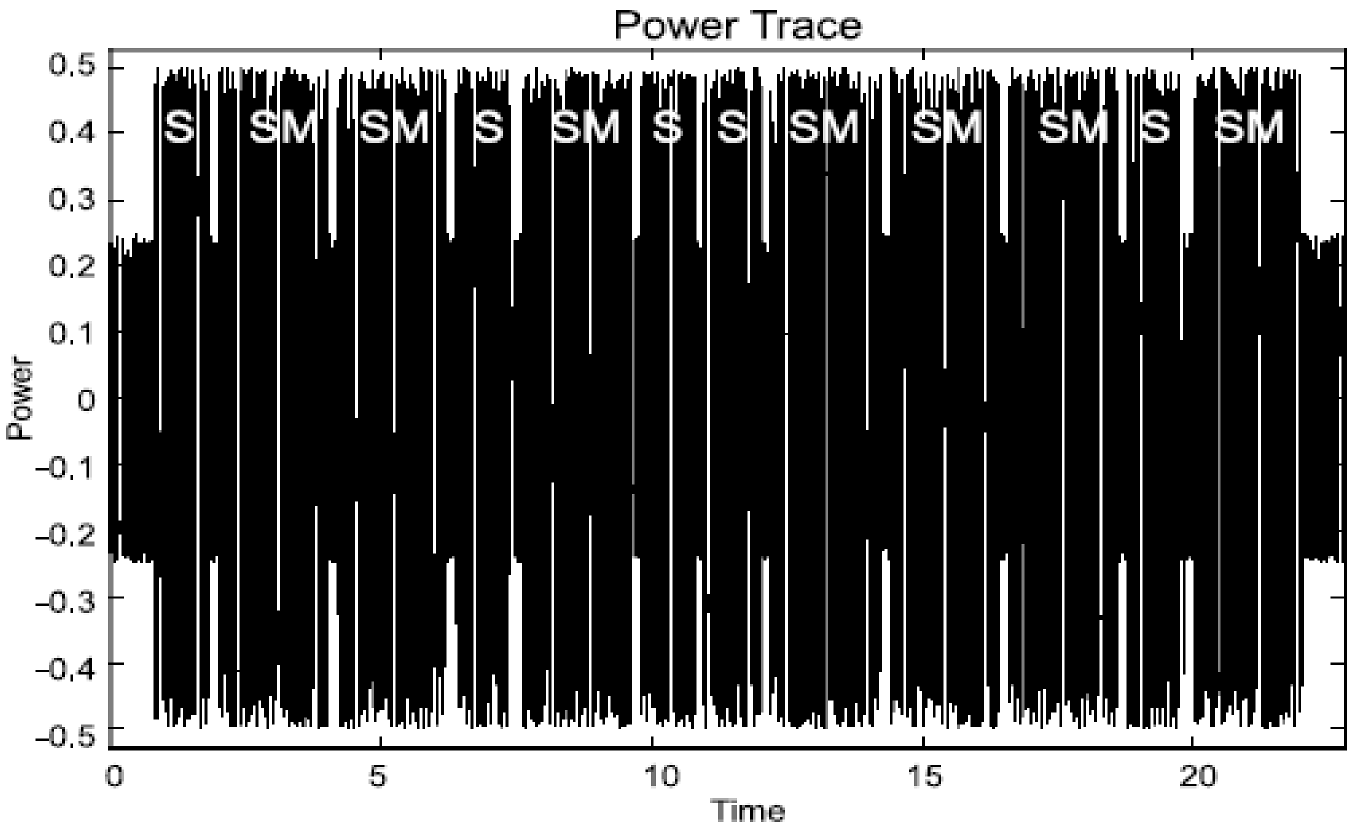
Again, we discard this since d must be odd.

The fifth fraction is $121/735$. This gives $C = (e \cdot 735 - 1)/121$, which is not an integer.

The seventh fraction is $578/3511$ This gives $C = 1966981103495136$ as the candidate for $\phi(n)$. The roots of $n = 37264873 \times 52783789$

Side-Channel Attacks

operations: *S SM SM S SM S S SM SM SM S SM*
private key: 0 1 1 0 1 0 0 1 1 1 0 1



Cycling Attack

- **Idea:** Eve continuously encrypts the intercepted ciphertext C, she eventually get the plaintext.

$$C_1 \equiv C^e \bmod n$$

$$C_2 \equiv C_1^e \bmod n$$

$$C_3 \equiv C_2^e \bmod n$$

$$\vdots$$

$$C_k \equiv C_{k-1}^e \bmod n$$

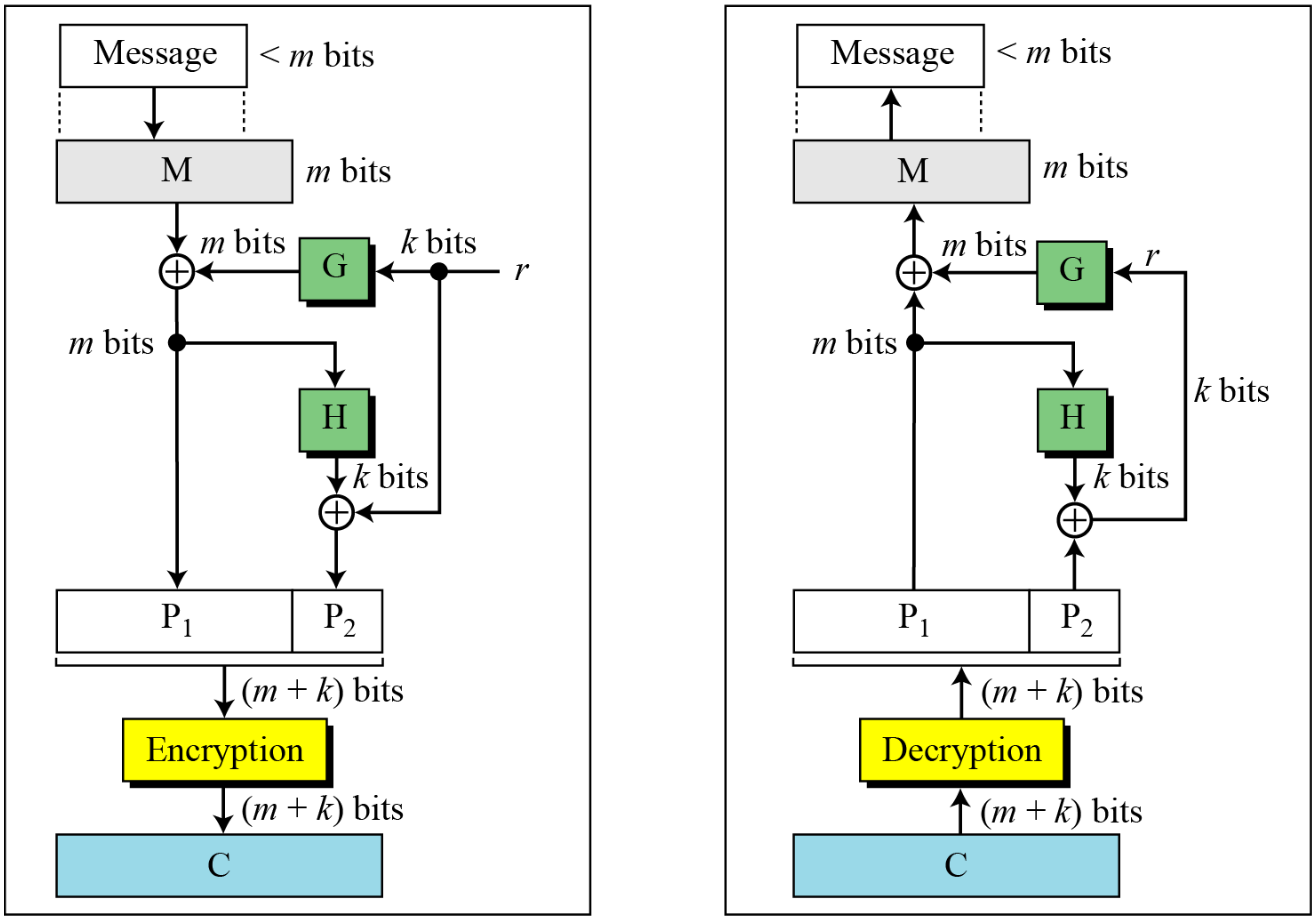
If $\mathrm{C_k} = C$ then stop; the plaintext is $P = C_{k-1}$

Short Message Attack

- Even can encrypt all of the possible message until the result is the same as the ciphertext intercepted.
- For example if it is known that Alice is sending a **four digit** number to Bob, Eve can easily try plaintext numbers from 0000 to 9999 to find the plaintext.
- To defend such kind o attack, **Optimal Asymmetric Encryption Padding (OAEP)** is the standard approach.
- In addition, OAEP mapped the same plaintext with different ciphertext.

Optimal Asymmetric Encryption Padding (OAEP)

M: Padded message
r: One-time random number
P: Plaintext ($P_1 \| P_2$)
C: Ciphertext
G: Public function (k-bit to m-bit)
H: Public function (m-bit to k-bit)



Sender

Receiver

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.

  Not necessary with Elliptic Curve Cryptography.

!!!Thank You!!!