
Plan

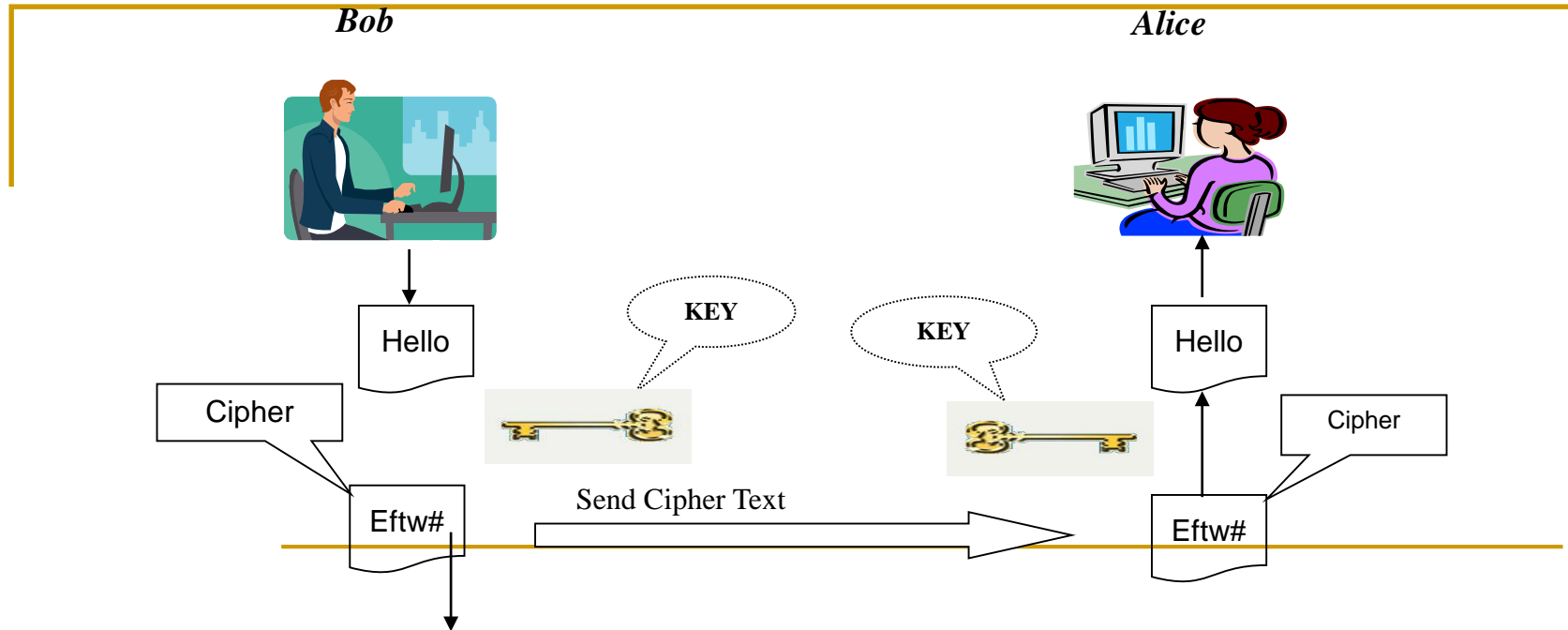
- Introduce Asymmetric Key Cryptography.
 - Diffie-Hellman Protocol
 - RSA Encryption.
 - Attacks on RSA encryption
 - Notions of Security
 - RSA in Practice
-

Plan of Talk

- **Disadvantages of Symmetric key Cryptography.**
 - **Asymmetric Key Cryptography.**
 - **Diffie-Hellman Protocol.**
 - **DH problem**
 - **RSA**
-

Limitations of Classical or traditional (Symmetric key)

cryptography



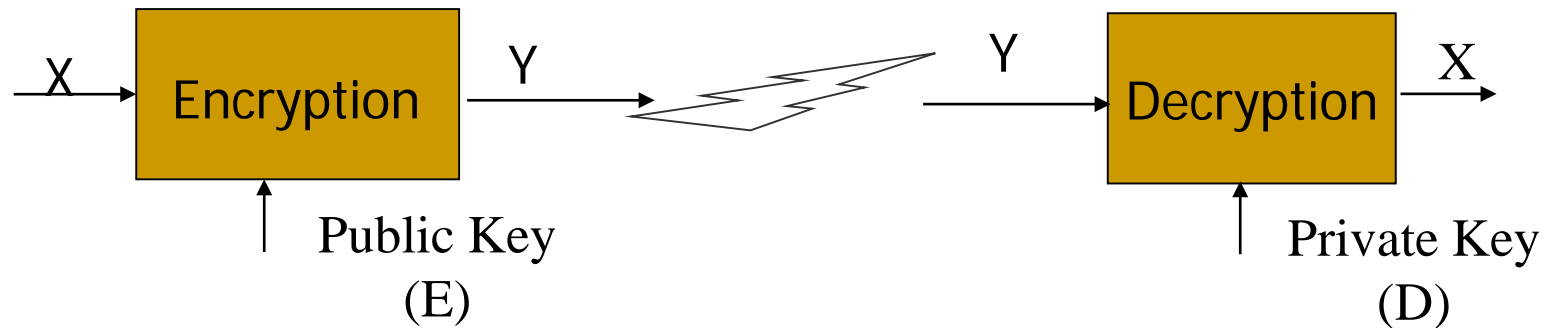
**Fast; Built-in Authentication; But, they need to share the key.
Confidentiality, but cannot protect against each other.**

Disadvantages of Symmetric key Cryptography

- One key is used for both encryption and decryption.
- Further the key need be shared by both sender and receiver.
- If the key is disclosed, the scheme is compromised.
- Non-repudiation is impossible as sender and receiver are equal. One party can forge other party's data. Hence it does not protect the sender from a receiver forging a message and then claiming that it is sent by the sender.
- In networked situation, the requirement for the key storage grows quadratic in n , the numbers of users. The number of common keys is $n(n-1)/2$.

Asymmetric Cryptography

- Communication parties are not equal.
- Uses two keys; a public and a private key.
- From Shannon's analysis of perfect cipher: Encryption transformation should distribute messages to cipher space fairly uniformly. Diffie-Hellman gave a concrete realization of this property without using any secret. This heralded the birth of **public key cryptography**.



PKC

Public Key Cont.

- Modern cryptography; The paper of Diffie-Hellman in 1976 (December 1975 to be precise).
- In a networked situation, the requirement for the key storage grows linearly in n , the number of users.
- Uses two keys; a public and a private key.
- Non-Repudiation is possible.
- Mechanisms differ from the way you lock and unlock.

How it works

- Put (lock, locking key) in the public domain, i.e., in a post office.
- Anyone who wants to send me a letter confidentially can do the following:
 - Buy a strong box
 - Use my lock with the locking key to lock the letter inside.
 - Send me the locked box.

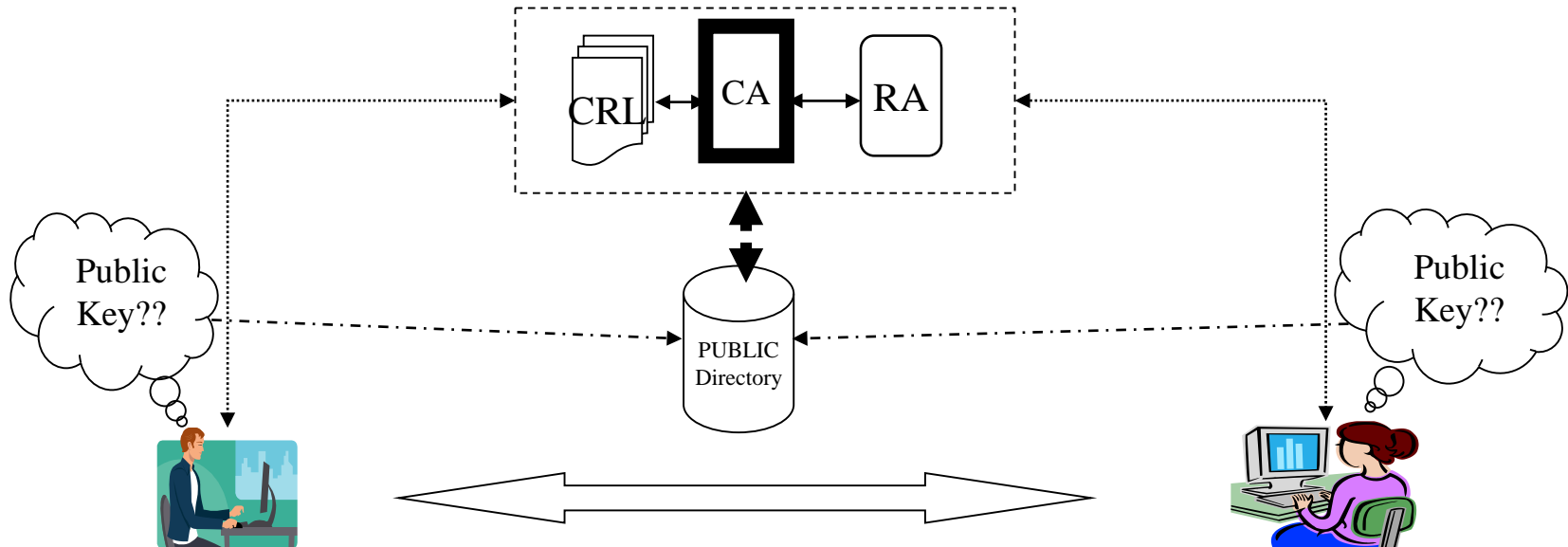


Locking key



Unlocking key

Traditional PKC



Examples:

RSA

ElGamal

ECC based Systems

Typical Uses

Encryption:

- Generally, it involves the use of two keys:
 - A public-key, which may be known by anybody and can be used to encrypt messages.
 - A private-key, known only to the recipient, used to decrypt messages.

Signatures

- Generally, it involves the use of two keys:
 - A private-key, known only to the signer is used to sign messages.
 - A public-key, which may be known by anybody and can be used to verify messages.
- The above methods are asymmetric, because those who encrypt messages or verify signatures cannot decrypt messages or create signatures.

One Way Function

- Let f be a function defined over integers modulo a large number (can be a prime or product of two large primes)
- Computing $f(x)=y$ given ' x ' is easy
- Given $y=f(x)$, computing ' x ' from ' y ' is difficult or hard
- Issues :How hard?
 - Generally the best known algorithm for inverting the function is sub-exponential in number of bits used to represent the elements in function domain or range.

Groups

- Group can be defined as a non-empty set G together with an associative binary operation $*$ satisfying the axioms namely closure, associative, identity and inverse.
- A group G is said to be abelian if it is commutative. That is, it satisfies the commutative property, for all x and y in G , $xy = yx$. In additive notation this translates to $x + y = y + x$. The integers, rational numbers, real numbers and complex numbers under addition are examples of abelian groups.
- A cyclic group is a group which can be generated by a single element called the group generator (set of group elements such that repeated application of the generators on themselves and each other produces all the elements of the group). Cyclic groups are abelian and its generator x satisfies $x^n = I$, where I is the identity element.

Examples

- Example of cyclic groups:
 - \mathbb{Z}_p^* set of non-zero integers modulo a prime p ; $p=5$.

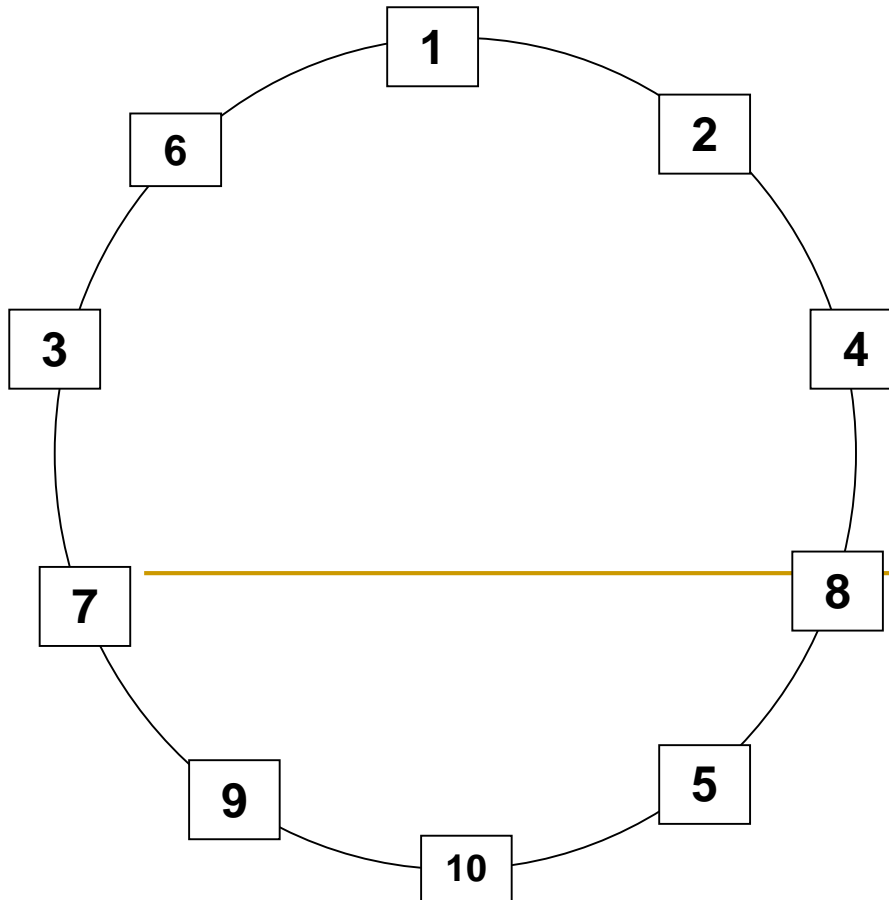
*	1	2	3	4
1	1	2	3	4
2	2	4	1	3
3	3	1	4	2
4	4	3	2	1

- \mathbb{Z}_4^+ additive group of integer modulo m.

+	0	1	2	3
0	0	1	2	3
1	1	2	3	0
2	2	3	0	1
3	3	0	1	2

Discrete Logarithm Problem

- Let 'g' and 'h' be elements of the group G. Then discrete logarithm (DL) problem is the problem of finding 'x' such that $g^x = h$.
 - For example, the solution to the problem $3^x = 13 \pmod{17}$ is 4, because $3^4 = 81 = 13 \pmod{17}$.
- The discrete log problem is believed to be difficult. Therefore it has become the basis of several public key schemes, for example: El-Gamal.



2^1	2	1
2^2	4	2
2^3	8	3
2^4	5	4
2^5	10	5
2^6	9	6
2^7	7	7
2^8	3	8
2^9	6	9
2^{10}	1	10

One Way Function Example

X	$2^x \bmod 11$	$3^x \bmod 11$
0	1	1
1	2	3
2	4	9
3	8	5
4	5	4
5	10 Or -1	1
6	9	3
7	7	9
8	3	5
9	6	4
10	1	1
11	2	3

Example cont.

- 2 is a primitive element.
 - 3 is not a primitive element
 - Given any power of 2, the exponent can be obtained from reading the corresponding index in the table
 - In practice a large modulus is used and hence finding the exponent is difficult. This is one of the important one way functions used in modern cryptography.
 - In general finding primitive element is also an interesting problem. We use the groups where we can easily find generating elements.
-

Diffie-Hellman Key Establishment Protocol

- The diagram illustrates the Diffie-Hellman key exchange process between Alice and Bob. It is organized into two columns, one for Alice and one for Bob, with a central area for the shared key calculation.

 - Alice's Side (Left Column):**
 - Parameters: $p=11, g=2$
 - Step 1: Choose $N_a=2$
 - Calculation: $g^{N_a} = 2^2 = 4 = M_a$
 - Step 2: Compute $K_{ab} = M_b^{N_a} = 9^2 = 4$
 - Bob's Side (Right Column):**
 - Parameters: $p=11, g=2$ (shared with Alice)
 - Step 1: Choose $N_b=6$
 - Calculation: $g^{N_b} = 2^6 = 9 = M_b$
 - Step 2: Compute $K_{ba} = M_a^{N_b} = 4^6 = 4$
 - Shared Key (Bottom Center):**
 - Final result: $K_{ab} = K_{ba} = 4$

Arrows indicate the exchange of public values M_a and M_b between the two parties.

Computational DH problem

- Let G be a cyclic group of size q and g be a generator of the group G .
- Given g^a and g^b , two arbitrary elements of the group G for some integers a and b in the range of

$0 \leq a, b \leq q$, then find

$$g^{ab}$$

Normally G is a group in a suitable finite field.

Relationship between DLOG and DH problems

- Clearly a solution to DLOG implies a solution to DH.
- Is the converse true?
- This is one of the open problems.

Problems with DH Key Exchange

- Man in the middle attack
 - Authentication is missing
 - Need for Key certifying authority
 - If Alice and Bob know each others authenticated public keys, man in the middle attack can be thwarted.
-

Man in the middle Attack

■ Alice

■ Choose N_a

■ g^{N_a}



Choose N_m

g^{N_m}



Choose N_b

g^{N_b}



Gets g^{N_m}

Computes $(g^{N_m})^{N_a}$

Computes $(g^{N_m})^{N_b}$

Malice shares $k_1 = g^{(N_m N_a)}$ with Alice

Malice shares $k_2 = g^{(N_m N_b)}$ with Bob

Cryptosystem

Definition: A cryptosystem is a five-tuple $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$, where the following conditions are satisfied:

\mathcal{P} : a set of possible plaintexts;

\mathcal{C} a set of possible ciphertexts;

\mathcal{K} , the space of keys, a finite set of possible keys;

For each k in \mathcal{K} , there is an encryption rule e_k in \mathcal{E} and a corresponding decryption rule d_k in \mathcal{D} . Each

■ $e_k: \mathcal{P} \rightarrow \mathcal{C}$ and $d_k: \mathcal{C} \rightarrow \mathcal{P}$

are functions such that

■ $d_k(e_k(x)) = x$ for every plaintext x in \mathcal{P} .

How do we create public key encryption?

- Encryption function should be publicly available, eg. from a directory.
- Anyone should be able to encrypt: We need a **one way** function **f**.
- Only the designated user should be able to decrypt
 - The user needs to invert **f** somehow – **f** is one-way.
 - idea is to create a **trapdoor** function which should enable to get the encrypted message.
 - Any public key encryption should have the above features.

One way Functions

- In the last lecture we looked one way function defined over a cyclic group with a generator.
- Let G be multiplicative group of a large order q with a generator g , i.e $G = \{g^0=1, g^1, \dots, g^{q-1}\}$.
- Then given a random a in $\{1, \dots, q\}$, computing g^a is **EASY**
- But given a arbitrary random element y in G it is computationally **HARD** to find a such that $g^a = y$, $a = \text{DLOG of } y \text{ to base } g$.

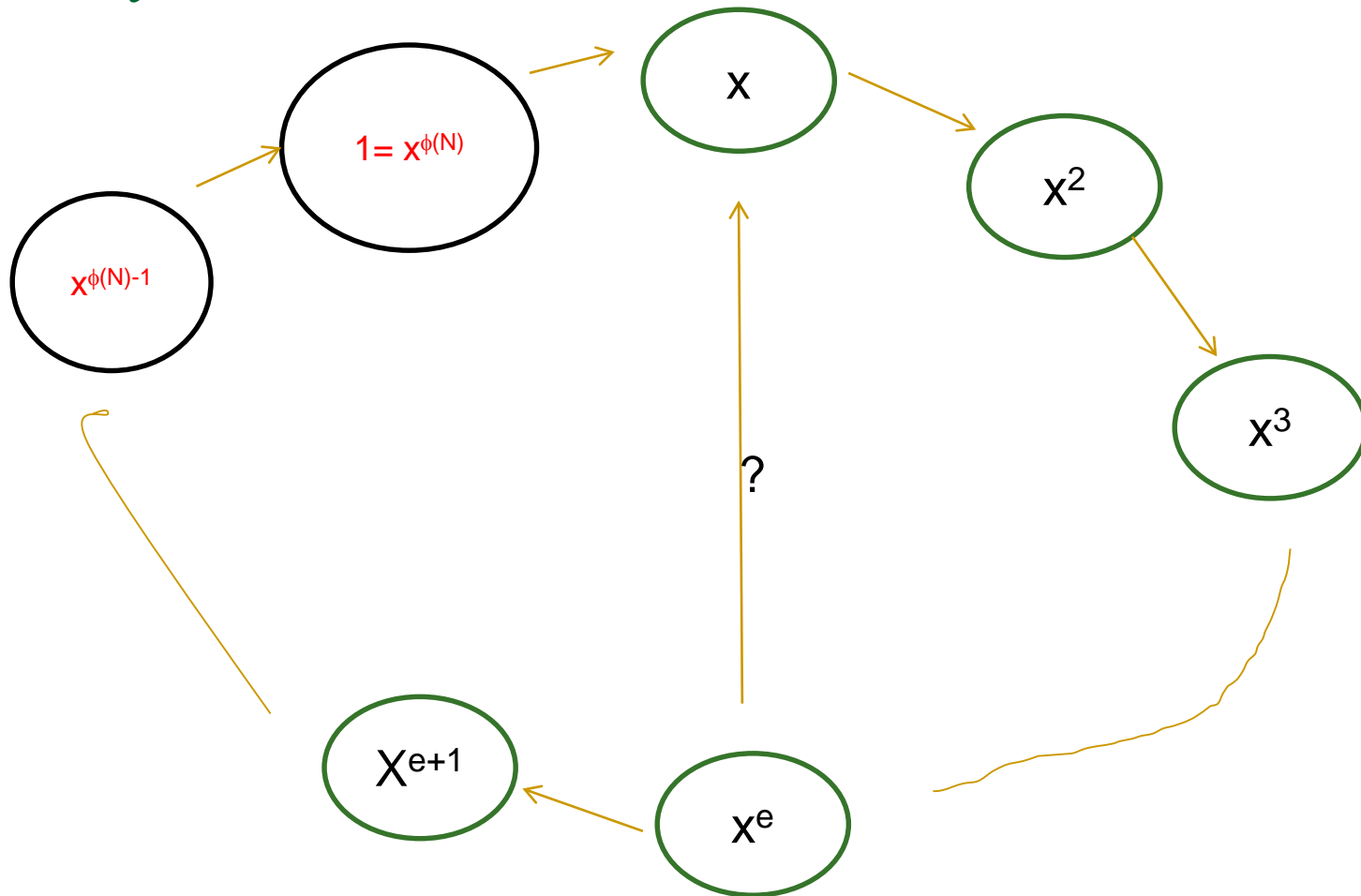
New One way functions

- Are there any other groups whose order could be secret!
- RSA is one such scheme.
- RSA relies on a group of numbers modulo n , which is a product of two large primes.
- I will explain this idea informally. You need to read the slides from Primes.ppt which gives you necessary mathematical background.

RSA Idea

- Alice claims that she knows the factorization of $N = PQ$;
 P, Q Large Primes.
- Currently it is impossible for anyone to get P, Q from N -
Factorization is a hard problem.
- Let us work with some random $x \bmod N$.
- We will assume that $\gcd(x, N) = 1$.
- Consider the group generated by $x \bmod N$.
- We can show that $x^{\phi(N)} = 1 \bmod N$.
- Alice needs to create a public encryption function that
anyone can encrypt, but only she can decrypt.

Why it works?



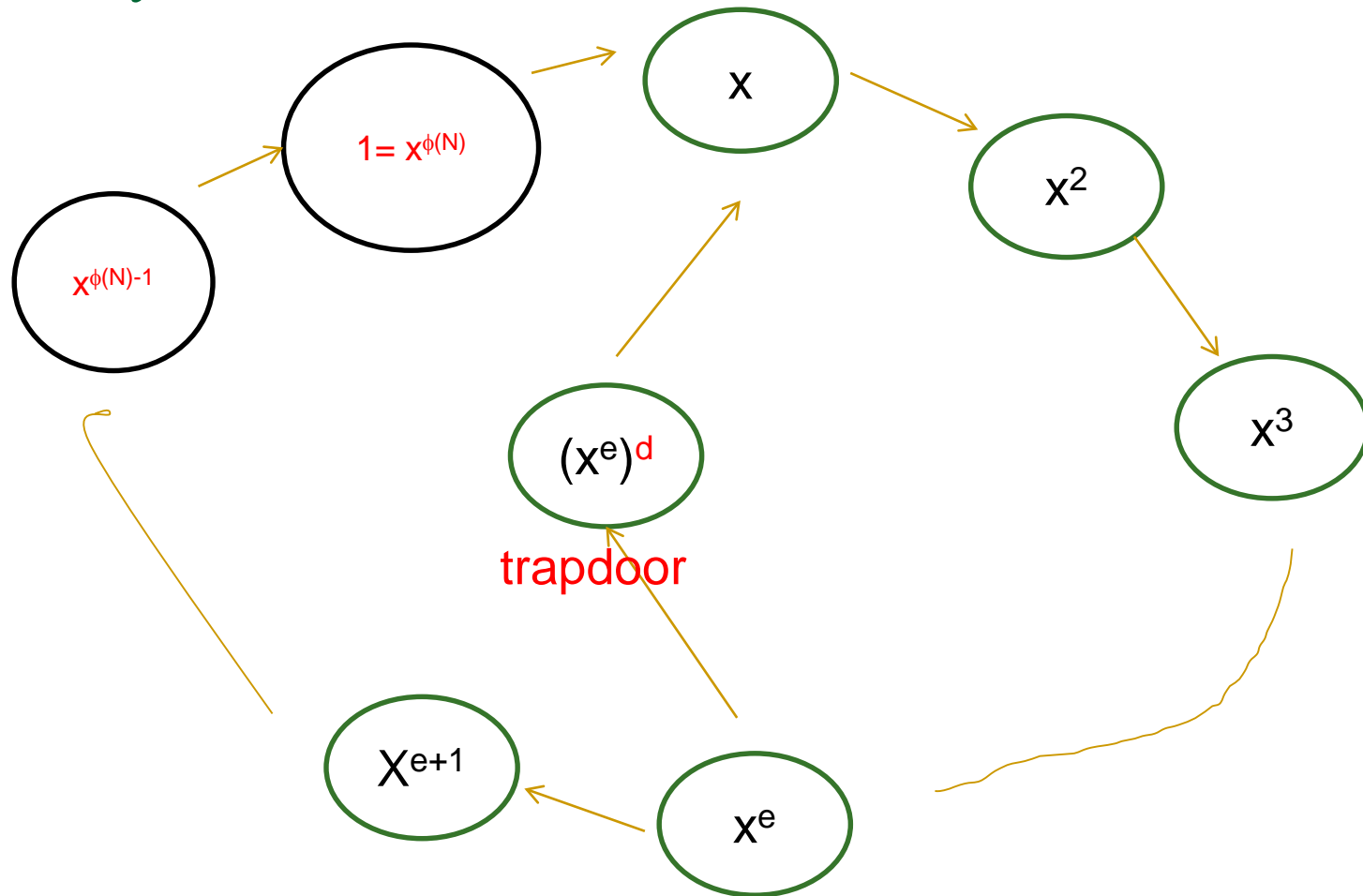
Group of numbers modulo N under multiplication

The order of the group is $\phi(N)$ = number of integers less than N and relatively prime to $N = (P-1)(Q-1)$.

RSA Idea Cont

- Alice will choose e a random number between 1 and $\phi(N)$ and make it public.
- So, Bob can take (e, N) and compute
 - $x^e \bmod N$ as his encryption.
- No one else can work backwards from x^e to x because it is another hard problem-finding e^{th} root of unity mod N (also known as RSA problem).
- How does Alice recover x then?
 - She will create a trapdoor as follows.
 - She will compute d such that $e * d \equiv 1 \bmod \phi(N)$.
 - $(x^e)^d \bmod N = x$;

Why it works?



Group of numbers modulo n under multiplication

The order of the group is $\phi(N)$ = number of integers less than n and relatively prime to $N = (P-1)(Q-1)$. $\phi(N)$ is known to Alice which allows her to find the trapdoor function y^d from the public value .

RSA PKC (1978)

■ Let $n = P \cdot Q$; P, Q are primes. Let $\mathcal{P} = \mathcal{C} = \mathbb{Z}_N$ (residue Integers modulo N). Define

■ $\mathcal{K} = \{ (N, P, Q, e, d) : e \cdot d \equiv 1 \pmod{\phi(N)} \}$

(ϕ : Euler's totient function)

■ For $K = (N, p, q, e, d)$, define

$$E_k(x) = x^e \pmod{N}$$

And

$$D_k(y) = y^d \pmod{N},$$

where $(x, y \in \mathbb{Z}_N)$. The values (N, e) are termed the **public key**, and the values P, Q and d form the private key.

RSA Example

■ Let $N = 91$; $P=13$, $Q=7$ are primes. Let $\mathcal{P} = \mathcal{C} = \mathbb{Z}_{91}$ (residue Integers modulo 91). $\phi(N) = 12*6 = 72$.

■ For $K = (N=91, P=13, Q=7, 5, 29)$, define

$$E_k(x) = x^e \bmod N$$

And

$$D_k(y) = y^d \bmod N,$$

■ Verify $5*29 = 145 \bmod 72 = 1$

■ Message $x = 11$

■ $E_k(11) = C = 11^5 = 72$

■ $D_k(72) = 72^{29} = 11$

RSA is a encryption function

- You need to convince yourself that the RSA decryption function is a one way trapdoor function. If you know d , you can decrypt, otherwise it is impossible.
- It is known that given n, e , $c = M^e \pmod{n}$, it is impossible to determine M . This problem is called RSA problem and also known as determining e^{th} root of $M \pmod{n}$. In general this problem is hard.
- If you determine d from only public parameters, then also you can break RSA. This problem can be solved if you can solve integer factorization problem.

Factorization of numbers

- In general the factorization is hard.
 - Brute force Attack: (infeasible given size of numbers) Brute force algorithm is exponential in b , where b is number of bits in the representation of the number n to be factored.
- Complexity of the best known algorithm for factorization:
$$\exp((c + O(1))b^{1/3} \text{Log}^{2/3}(b)),$$

for some integer $c < 2$
- May be quantum computers come to our rescue; but may not in our life time!

Summary: Hard problems on which RSA is based.

- 1. Integer Factorization problem: Given a large positive integer n , find its prime factorization. (Every number n can be expressed as $p_1^{e_1} p_2^{e_2} \dots p_k^{e_k}$, where the p_i 's are distinct primes and each $e_i > 1$). In particular, if a number n is constructed as a product of two large primes, it is difficult to factor n .
- 2. RSA problem: Given a positive integer n that is a product of two distinct odd primes p and q , ($n=pq$) and a positive integer e such that $\gcd(e, (p-1)(q-1)) = 1$, and an integer c , find an integer m such that
$$m^e = c \pmod{n}.$$

Security of Cryptosystems

- Almost all modern cryptosystems are based on more than one hard problems in mathematics (eg. Discrete logarithms, factorization, RSA problem etc).
- In fact there are no theoretical proofs available stating that these problems are hard.
- On the other hand, there are many instances where the so called hard problems are easy to perform.
- We should ensure that the practical implementation do not use such pathological cases. Hence, we have to address security against any known vulnerability of these hard problems.
- Such attacks based on specific vulnerability of instances of hard mathematical algorithms can be considered as Mathematical attacks. We look for active attacks next.

Security Notions for public key Cryptosystems: Active attacks

- The security of a cryptosystem is defined with respect to the attacks it can withstand.
- The attacker will not be given private or secret information of the cryptographic key whose public cryptosystem he is attacking.
- There are three types of active attacks:
 - **Chosen-plaintext attack(CPA)**
 - Encryption box is available to the attacker before the attack.
 - **Chosen-ciphertext attack(CCA)**
 - Decryption box is available to the attacker before the attack.
 - **Adaptive Chosen-ciphertext attack(CCA2)**
 - Decryption box is available to the attacker except for the challenged ciphertext.

Active attacks: Cont.

- ❑ **Chosen-plaintext attack(CPA)** Here the attacker can obtain cipher texts corresponding any chosen plain texts. The goal is to weaken the crypto system with the obtained plaintext-ciphertext pairs.
- ❑ **Chosen-ciphertext attack(CCA)** Here attacker can obtain plaintexts corresponding any chosen ciphertexts. This means the attacker gets decryption assistance for any chosen ciphertext. The goal for the attacker is to obtain any part of the plaintext after the decryption assistance is terminated.

Active attacks: Cont.

- ❑ **Adaptive Chosen-ciphertext** attack(CCA2) The attacker is challenged with a given ciphertext to decrypt or obtain any part of the plaintext. For this, he/She is provided decryption assistance forever for all chosen cipher texts except for the challenged ciphertext.
- The above notions are not impracticable and hence cryptosystems should be made secure against such attacks.

chosen ciphertext attack

- The basic RSA algorithm is vulnerable to a chosen ciphertext attack (CCA).
- In this scenario, the adversary gets decryption of a number of ciphertexts of his choice.
- Adversary will then be given a challenge cipher text for which he has to produce the decryption (without having access to the private key).



Attack

- Multiplicative property of RSA
 - $E(PU, M1) \times E(PU, M2) = E(PU, [M1 \times M2])$
- The attack procedure
 - 1. Compute $X = (C \times 2^e) \bmod n$
 - 2. Submit X as a chosen ciphertext and obtain corresponding plain text $Y = X^d \bmod n$.
 - 3. Note that Y is in fact $(2M) \bmod n$
 - 4. Compute $M = \text{Inverse}(2) * Y \bmod n$

Optimal Asymmetric Encryption Padding

- To overcome the previous attack, you need somehow break the multiplicative property of the scheme.
- In practice message is introduced with a specific format, which removes the multiplicative property.

OAEP

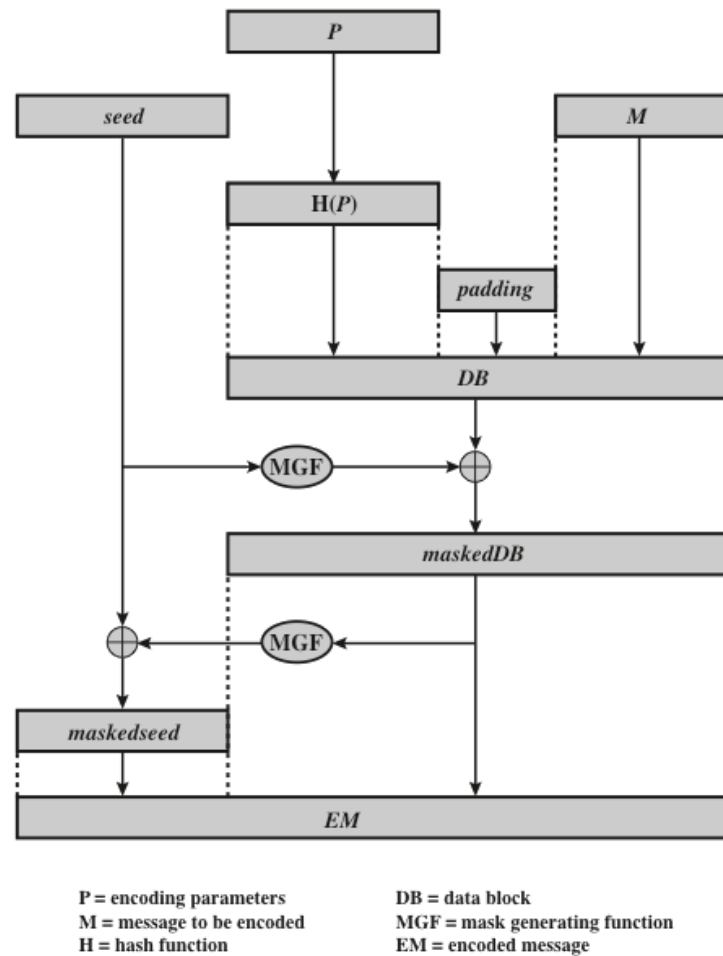


Figure 9.10 Encryption Using Optimal Asymmetric Encryption Padding (OAEP)

RSA Computations

- RSA encryption and decryption involves exponentiation modulo n .
- Exponentiation(a, t, n): $a^t \bmod n$
- What is the complexity of this function?
 - Let n be a b bit binary number
 - Further assume that t is also of order b .
 - Make use of the identity:
 - $[(a \bmod n) \times (b \bmod n)] \bmod n = (a \times b) \bmod n$
- $O(b) = O(\log n)$.

Faster Method in the textbook

```

c ← 0; f ← 1
for i ← k downto 0
    do  c ← 2 × c
        f ← (f × f) mod n
    if bi = 1
        then c ← c + 1
           f ← (f × a) mod n
return f

```

Note: The integer b is expressed as a binary number $b_k b_{k-1} \dots b_0$

Figure 9.8 Algorithm for Computing $a^b \bmod n$

i	9	8	7	6	5	4	3	2	1	0
b_i	1	0	0	0	1	1	0	0	0	0
c	1	2	4	8	17	35	70	140	280	560
f	7	49	157	526	160	241	298	166	67	1

Table 9.4 Result of the Fast Modular Exponentiation Algorithm for $a^b \bmod n$, where $a = 7$, $b = 560 = 1000110000$, and $n = 561$

Key Generation

- Each user (Alice) need to choose two primes p and q of the public modulus securely.
- Then e and d should be computed such that
 - $e d = 1 \bmod \Phi(n)$, $\Phi(n) = (p-1)(q-1)$.
- Primes need to be chosen from a sufficiently large set. ($n = pq$ will be known to any potential adversary)
 - The method should be reasonably efficient
 - Preferably obtained from a true random source.

Choosing Primes: Primality Testing

[We will not be studying mathematical aspects of primality testing]

■ Two methods: Probabilistic and deterministic

■ Probabilistic Test:

- 1. Pick an odd integer m at random (e.g., using a pseudorandom number generator-but should use truly random seed)).
- 2. Pick an integer $a < m$ at random.
- 3. Perform the probabilistic primality test, such as Miller-Rabin, with a as a parameter. If m fails the test, reject the value m and go to step 1.

■ Deterministic Test

- Complexity is polynomial, but still takes more time compared to the time taken by the probabilistic methods

Efficient Operation: Encryption

- For efficiency the RSA algorithm using the public key, a specific choice of e is usually made.
- The most common choice is 65537 ($2^{16} + 1$)
 - Two other popular choices are $e=3$ and $e=17$
 - Each of these choices has only two 1 bits, so the number of multiplications required to perform exponentiation is minimized
 - With a very small public key, such as $e = 3$, RSA becomes vulnerable to a simple attack

Efficient Operation: Decryption

[Please refer to the set of slides for mathematical aspects of RSA]

- Note that decryption uses exponentiation to power d .
 - Avoid a small value of d ; vulnerable to a brute-force attack and to other forms of cryptanalysis
- The Chinese Remainder Theorem (CRT) method
 - The quantities $d_p = d \bmod (p - 1)$ and $d_q = d \bmod (q - 1)$ can be pre calculated
 - Compute $C^{d_p} \bmod p$ and $C^{d_q} \bmod q$ and then use CRT to compute $M = C^d$. You can show that the method is approximately four times as fast as evaluating $M = C^d \bmod n$ directly

Summary

- We considered studying
 - DH Algorithm
 - RSA algorithm
- RSA Algorithm: How it works?
- Some proofs.
- Attacks