

Week 4



Lecture 1

Public Key Cryptography: Diffie-Hellman Protocol and RSA

Lecture 2

Proof of RSA Encryption + Chinese Remainder Theorem, Continued from
Week 3 Lecture 2

Workshop 3: Workshop based on Lectures in Week 3

Quiz 4

Public Key Cryptography: Diffie-Hellman Protocol and RSA

COMP90043
Lecture 1

Public Key Cryptography: Diffie-Hellman and RSA

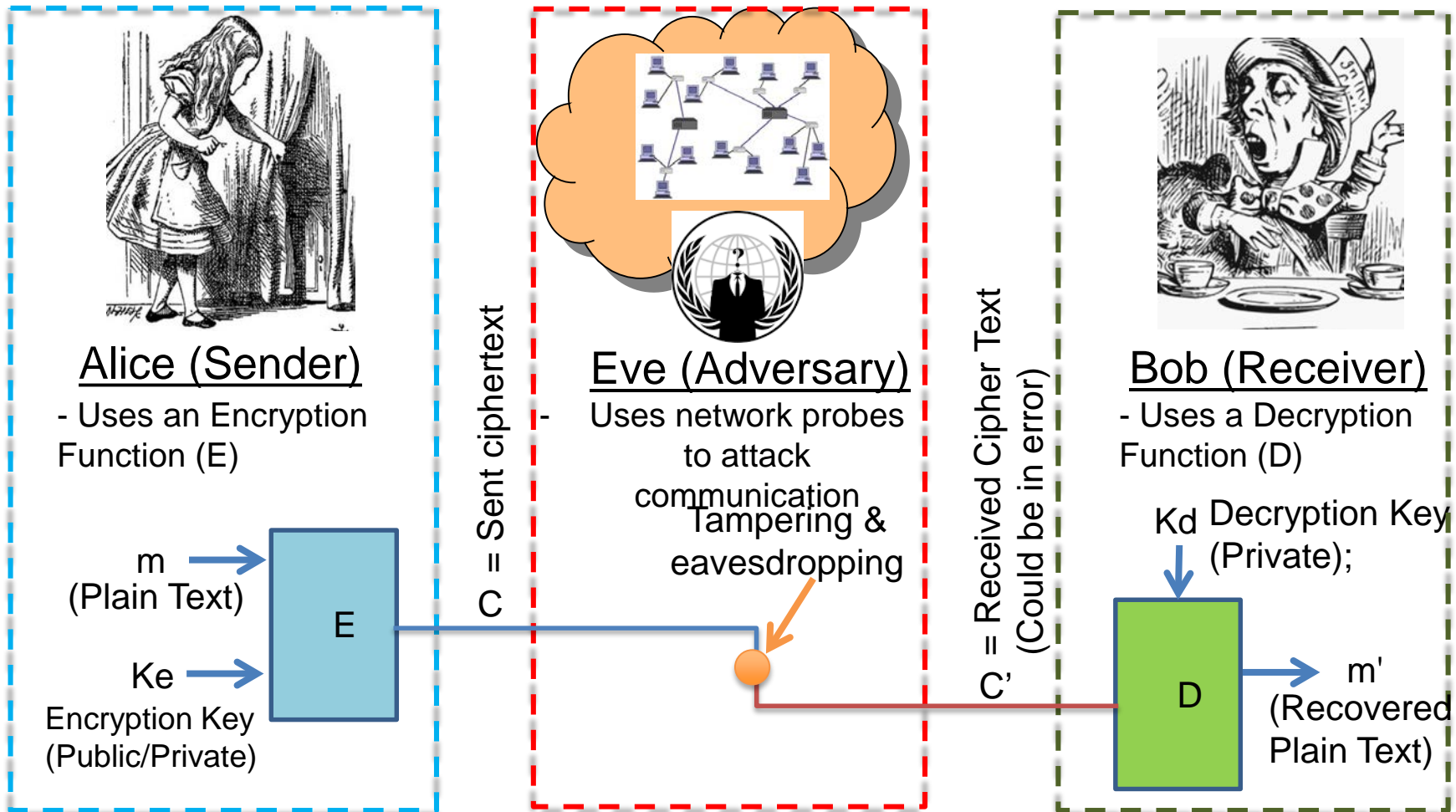
Lecture 1

- 1.1 Concept of Public Key
 - Limitations of Symmetric key system
 - Notations for Public key
- 1.2 Diffie-Hellman Protocol
 - Motivation
 - The protocol and Implications
 - Man in the Middle Attack
- 1.3 RSA Idea
 - Informal Idea
 - RSA Algorithm
 - Attacks on RSA

Recap from Week 1

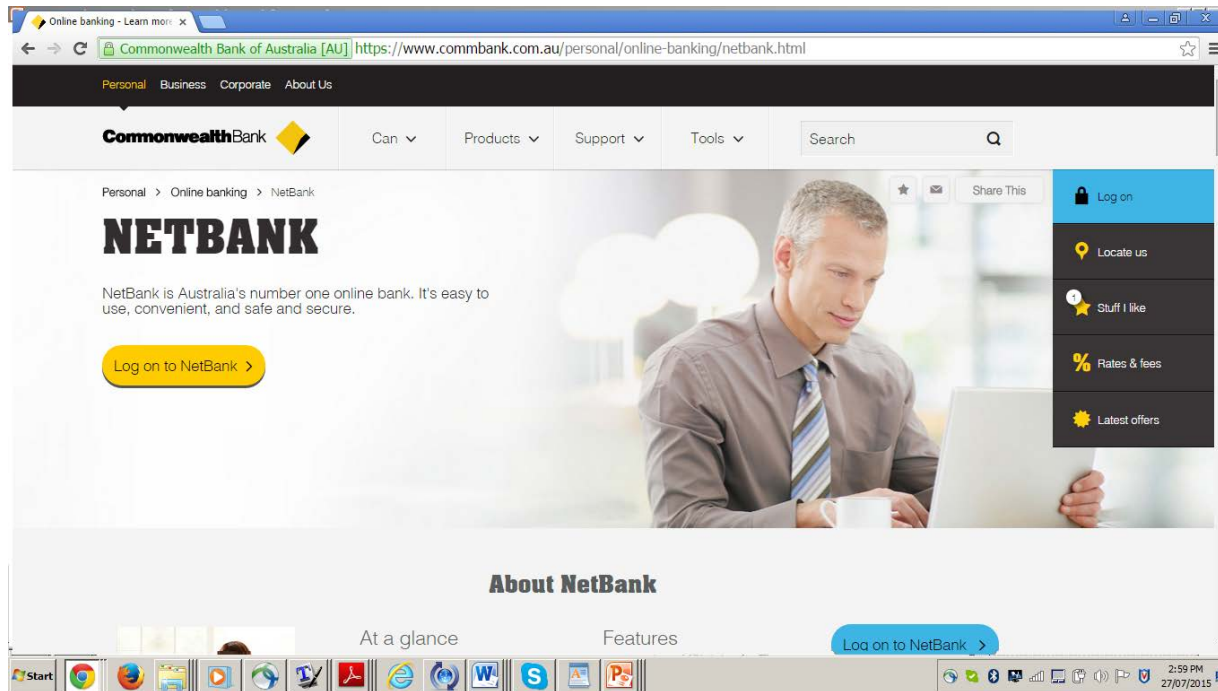
- First we will revisit some of the concepts we dealt in Week 1.

Story of Alice and Bob terms and notations



E , D are public; c is the ciphertext, c' is received ciphertext; ideally $m=m'$;
Cryptography involves many conceptual ideas, we look at the basic functions

Recap Motivating examples

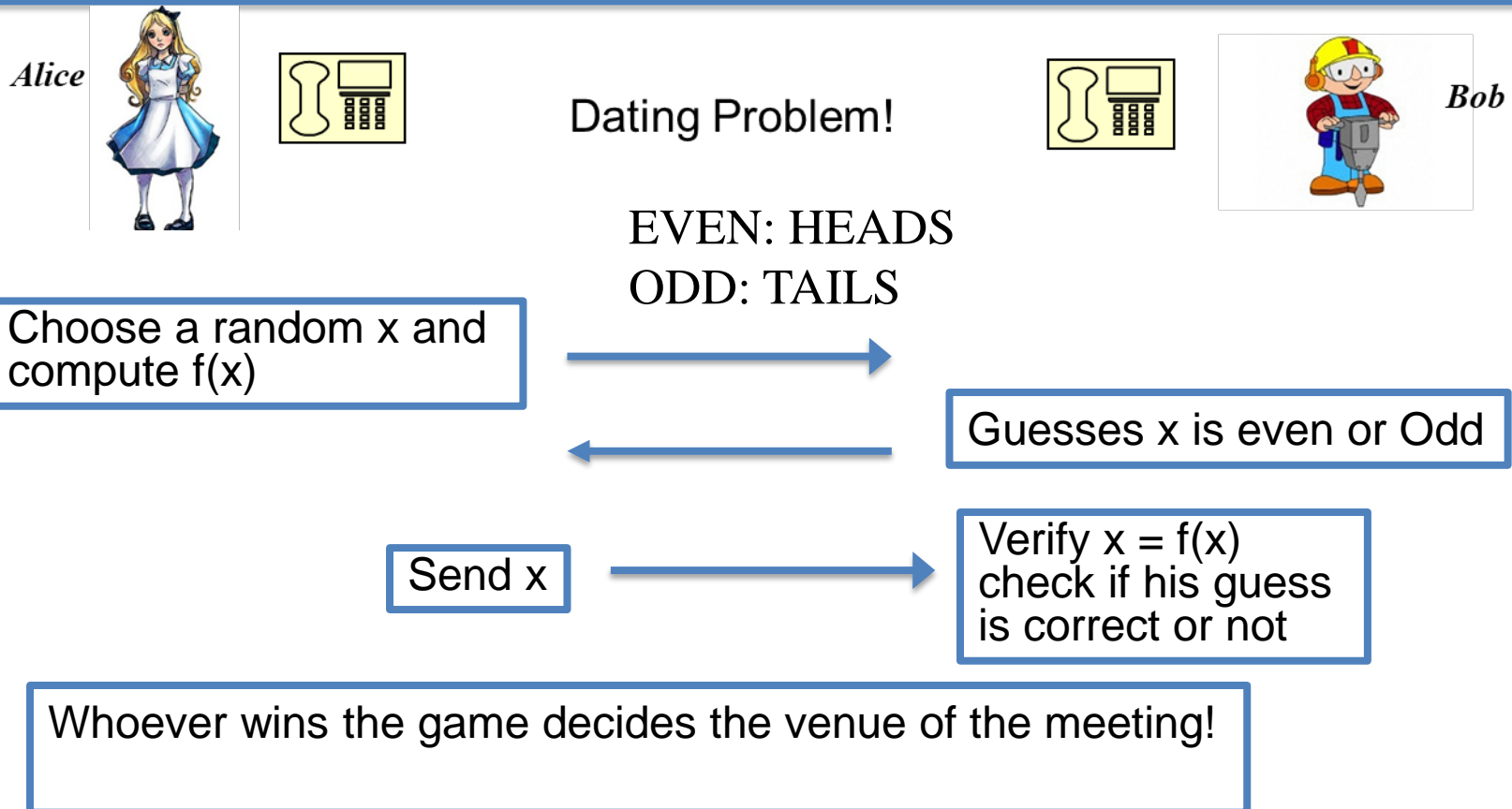


Comm bank Server



Issues in getting your money from the bank.
Should work over Internet
Think, who is Alice, Bob and Eve here.
What tools Cryptography can provide here?

A protocol

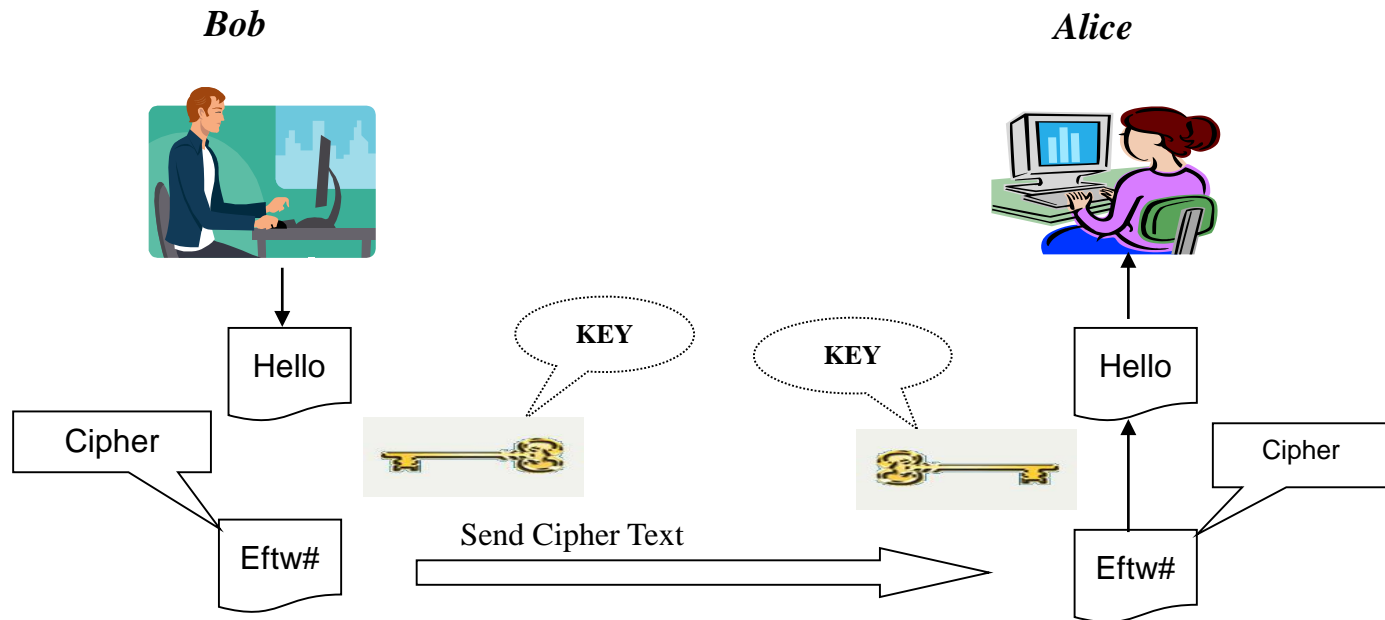


Is this protocol correct and fair (unbiased)?
Can you modify so that both Alice and Bob

1.1 Concept of Public Key

COMP90043
Lecture 1

Limitations of Symmetric Key Systems



- Symmetric key is fast and provides in built in Authentication by virtue of users sharing the key.
- Sharing the key is a huge problem.
- They definitely provide confidentiality, but never project against each other.

Disadvantages of Symmetric key Systems

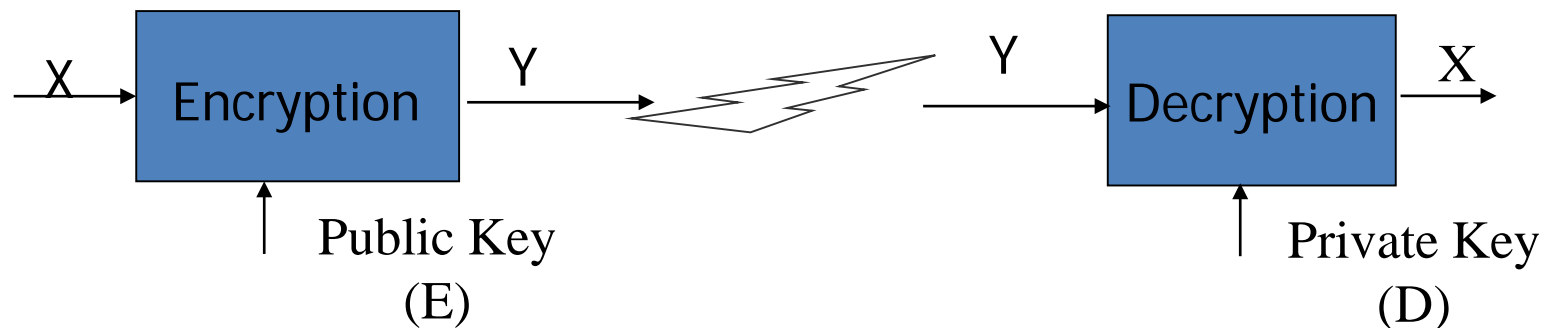
- One key is used for both encryption and decryption.
- Further the key need be shared by both sender and receiver.
- If the key is disclosed, the scheme is compromised.
- Non-repudiation is impossible as sender and receiver are equal. One party can forge other party's data. Hence it does not protect 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 $n(n-1)/2$.

Birth of Public Key Cryptography

- We discussed about Diffie-Hellman protocol in the Introduction lecture. Many consider this development a historical significant. Why?
- Symmetric key system may seem secure, but if keys are compromised, fails completely.
- Another problem is about authentication, can we have useful equivalent of hand written signatures for electronic transactions? We will discuss this concept later in the course, but public key cryptography achieves this property efficiently.
- Please read the original paper by Diffie-Hellman.

Asymmetric Cryptography

- Communication parties are not equal, a precondition for having accountability for their inputs into the conversation.
- 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**.



Asymmetric Cryptography, Continued

- 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, leading to natural accountability to the transactions.
- Mechanisms differ from the way you lock and unlock.
- Eg: Secure staff mail box in the department office



Picture from General Internet Resources

The Figure Illustrates the notations
And use of Public Key functions;
We will use this notation
throughout this semester.

Public key of B : PU_b
Private key of B : PR_b

Encryption and Decryption by A



$$Y = E(PU_b, X)$$

$$X := D(PR_b, Y)$$

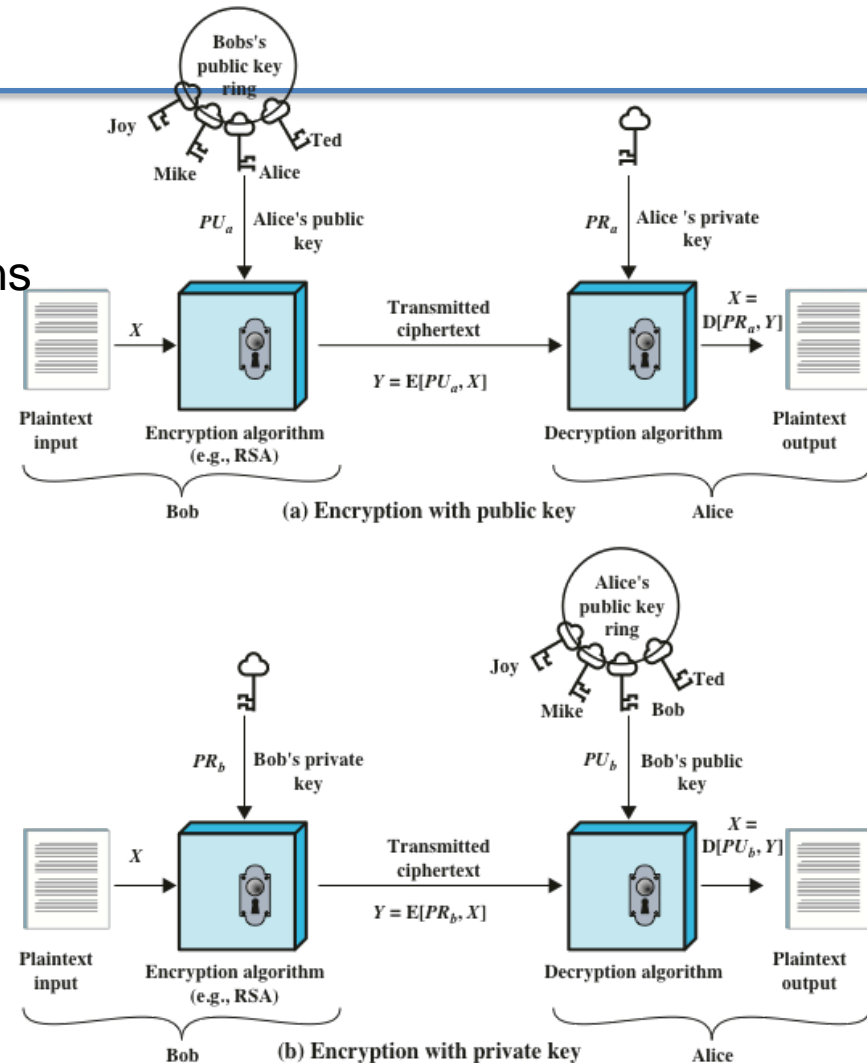
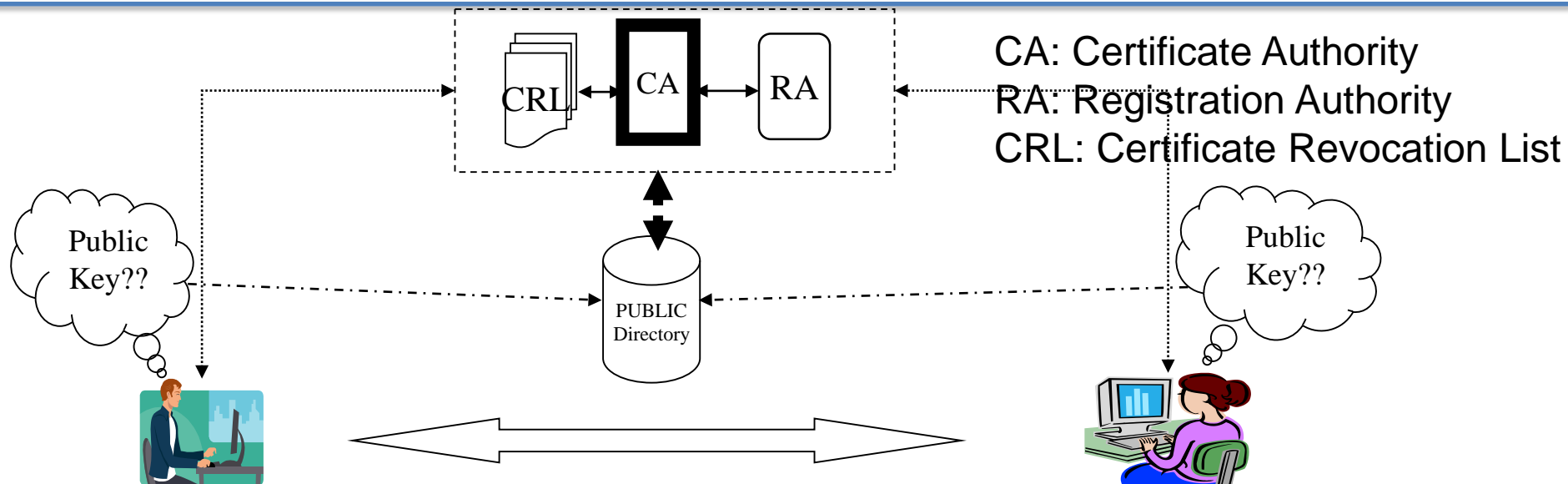


Figure 9.1 Public-Key Cryptography

Table 9.2 from the textbook

Conventional Encryption	Public-Key Encryption
<p><i>Needed to Work:</i></p> <ol style="list-style-type: none">1. The same algorithm with the same key is used for encryption and decryption.2. The sender and receiver must share the algorithm and the key. <p><i>Needed for Security:</i></p> <ol style="list-style-type: none">1. The key must be kept secret.2. It must be impossible or at least impractical to decipher a message if the key is kept secret.3. Knowledge of the algorithm plus samples of ciphertext must be insufficient to determine the key. <p>From Stalling's Textbook</p>	<p><i>Needed to Work:</i></p> <ol style="list-style-type: none">1. One algorithm is used for encryption and a related algorithm for decryption with a pair of keys, one for encryption and one for decryption.2. The sender and receiver must each have one of the matched pair of keys (not the same one). <p><i>Needed for Security:</i></p> <ol style="list-style-type: none">1. One of the two keys must be kept secret.2. It must be impossible or at least impractical to decipher a message if one of the keys is kept secret.3. Knowledge of the algorithm plus one of the keys plus samples of ciphertext must be insufficient to determine the other key.

Traditional or Conventional PKC



- How will Bob locate Alice's Public Key?.
- If he trusts a directory, then he could directly read from it, similar to one gets telephone numbers from White Pages. (White pages now has moved to online)
- An adversary can compromise the public directory. How will you handle such attacks?
- In practice, we need an elaborate arrangement of Public-Key Infrastructure, we will study later (In Chapter 14).

Typical Uses of Public Key Encryption

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.

1.2 Diffie-Hellman Public Key Protocol

COMP90043
Lecture 1

Idea



- We gave a general introduction and motivation for the subject in the first week.
- Recall that the main difficulty with symmetric key scheme is that the key management is going to be hard.
- Public key promises to simplify the key management-any two users who have not met before still be able to obtain a common secret using only Public information.
- However, public key systems also bring in new key management issues-will require a trusted system to distribute public keys. We will study this later.

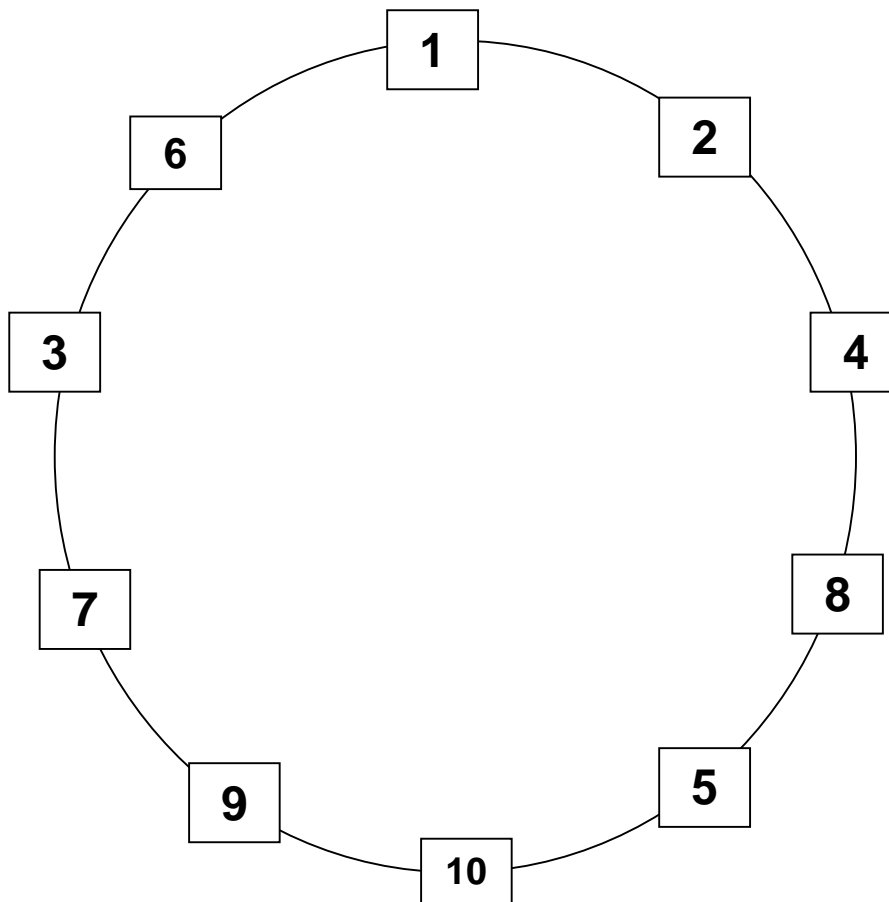
One Way Function

- 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.

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.

An example



g^i $g^i \bmod p$ $Dlog(g^i)$

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

Example of a Cyclic group modulo $p = 11$

g : generator = 2

Order(size) of $G = 10$



What power of 2 is 3?

Example mod 11

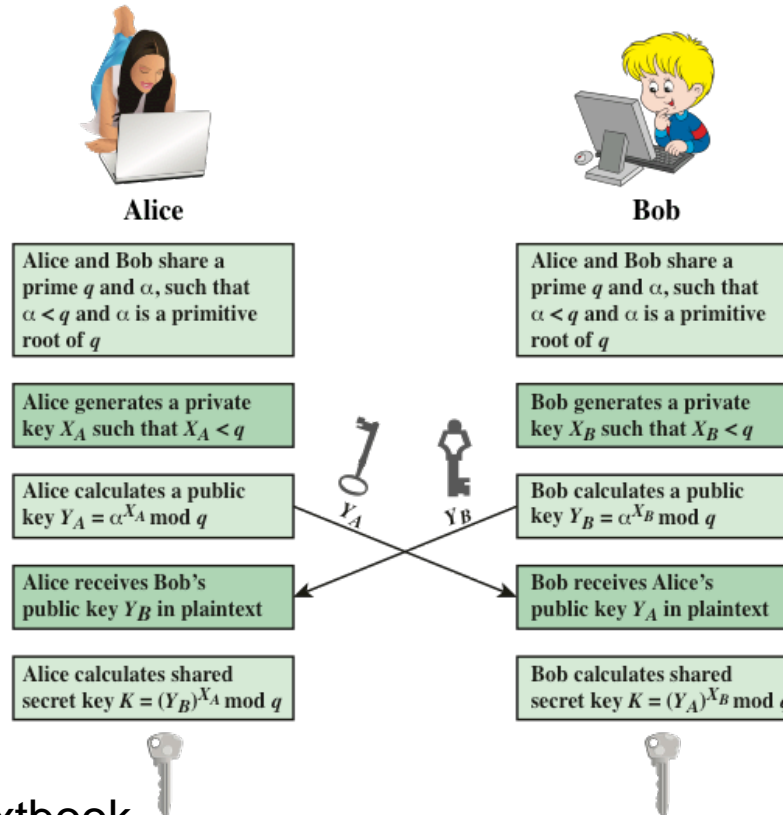
X	$2^x \bmod 11$	$3^x \bmod 11$
0	1	1
1	2	3
2	4	9
3	8	5
4	5	4
5	10 Or -1	1
6	9	3
7	7	9
8	3	5
9	6	4
10	1	1
11	2	3

- 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 Protocol

- Alice
 - Choose $N_a=2$
 - $g^{N_a} = 2^2=4 = M_a$
- 
- Bob
- Choose $N_b=6$
 - $g^{N_b} = 2^6=9=M_b$
- 
- Compute
 - $K_{ab} = M_b^{N_a}$
 - $= 9^2=4$
 -
 -
 -
 -
- $K_{ab} = K_{ba}=4$
- Compute
- $K_{ba} = M_a^{N_b} = 4^6=4$

Diffie-Hellman Protocol



From Stalling's textbook

Figure 10.1 Diffie-Hellman Key Exchange

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 **multiplicative** group in a suitable **finite field**.

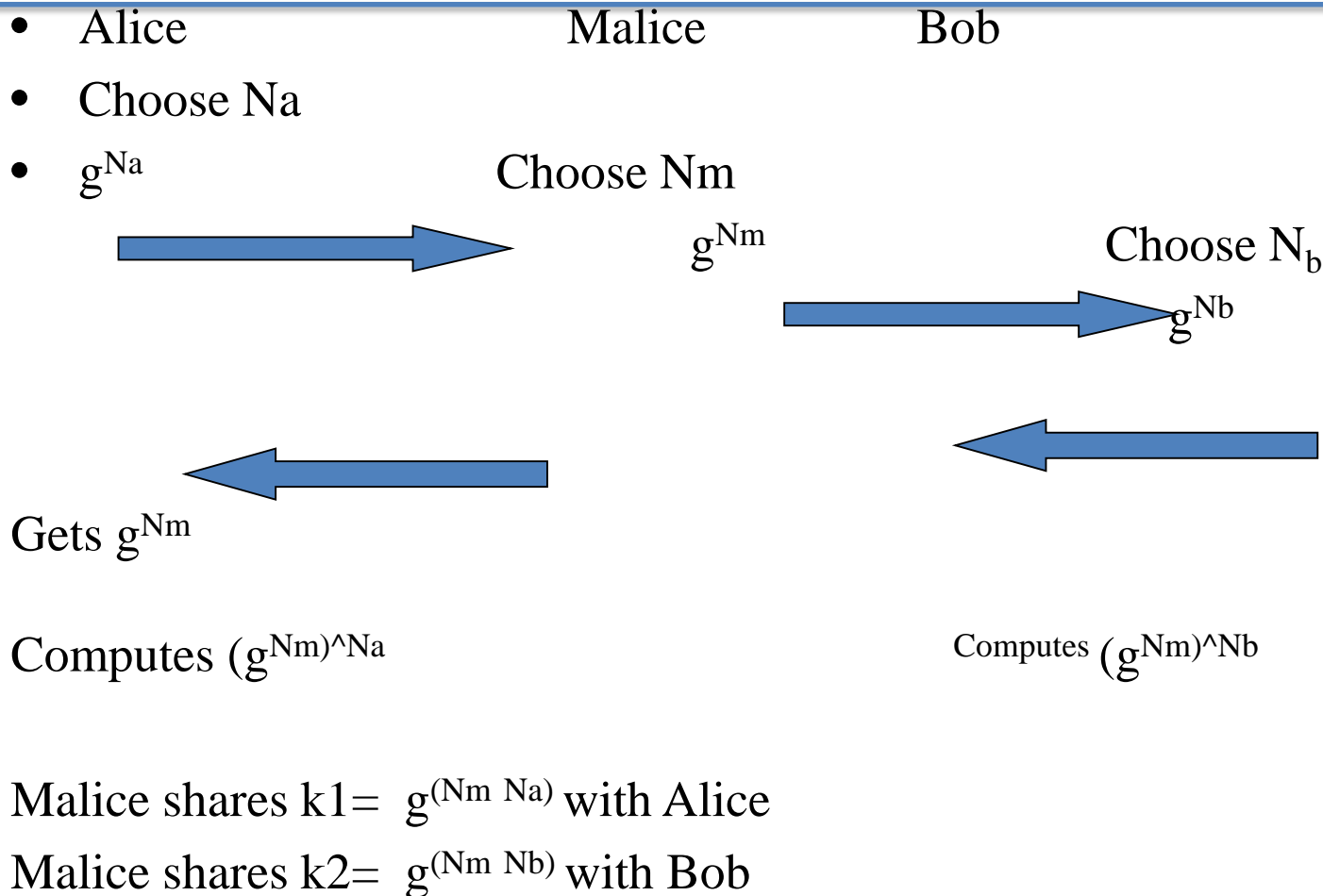
If someone comes with an efficient algorithm for this problem, the protocol is broken!

- 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

- The protocol has a new problem,
- When Alice and Bob exchanging information, how do they know that they are indeed talking to the right individuals?
- With the nature of Internet, someone could masquerade as Alice or Bob and try to fool Bob or Alice. This is because, the public information they exchange is not authenticated.
- The protocol is vulnerable to Man in the Middle Attack.

Man in the middle Attack



Also study the version of the Protocol in the textbook

How do we overcome the MITM Attack

- Main reason for the attack is because of lack of authentication.
- We need Digital Signatures and related concept to tackle this attack.
- We will study this later in the subject.

1.3 RSA Crypto System

COMP90043
Lecture 1

RSA

The first Public key
encryption algorithm



Ron Rivest, Adi Shamir and Leonard Adleman

Based on the assumption that factoring an integer which has an alleged factorization as a product of two prime numbers is a hard problem;

In other words, given an integer n which is constructed by two secret primes p and q , finding the factors is a hard problem.

Message and cipher text belong to Z_n

How to construct a crypto system using the above hard problem?

Basic Facts Again

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?

- Let us try to recreate questions that came to the creators of RSA 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.

New One way functions

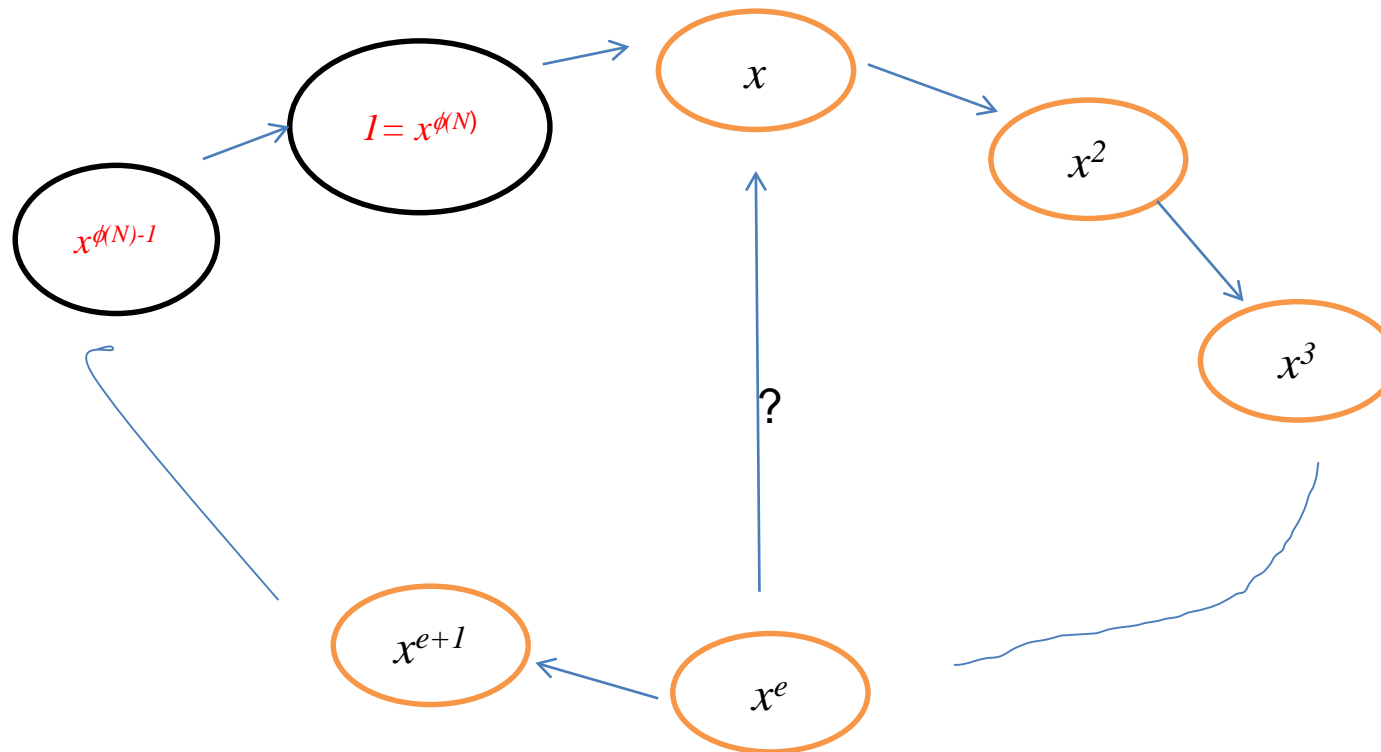
- We looked at Discrete Logarithm problem before.
- We considered a cyclic group G with a generator g
 - Let G be multiplicative group of a large order q with a generator g ,
$$G = \{g^0=1, g^1, \dots, g^{q-1}\}.$$
 - Given a random t in G , computing g^t is easy.
 - However, given an arbitrary y in G , it is computationally hard to obtain Discrete Log of y ; i.e it is hard to find t such that $g^t = y$.
- 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. We will also explain the idea from mathematical results that we have introduced.

RSA Idea

The basic RSA idea begins as follows:

- 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.

$x^{\phi(n)} = 1 \bmod n$: how it works



The operations are in the 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 $n = (p-1)(q-1)$.

$$\begin{aligned} n - p - q + 1 \\ &= pq - p - q + 1 \\ &= (p-1)(q-1) \end{aligned}$$

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 $\bmod n$ (also known as RSA problem).
- But how does Alice recover x ?

-> She will create a trapdoor as follows.
-> She will compute d such that $e \times d \equiv 1 \bmod \phi(n)$.
-> $(x^e)^d \bmod n = x$;

Week 4



Lecture 1

Part -1: Chinese Remainder Theorem, Continued from Week 3 Lecture2

Part II: Introduction to Public Key Cryptography: Diffie-Hellman Protocol and RSA

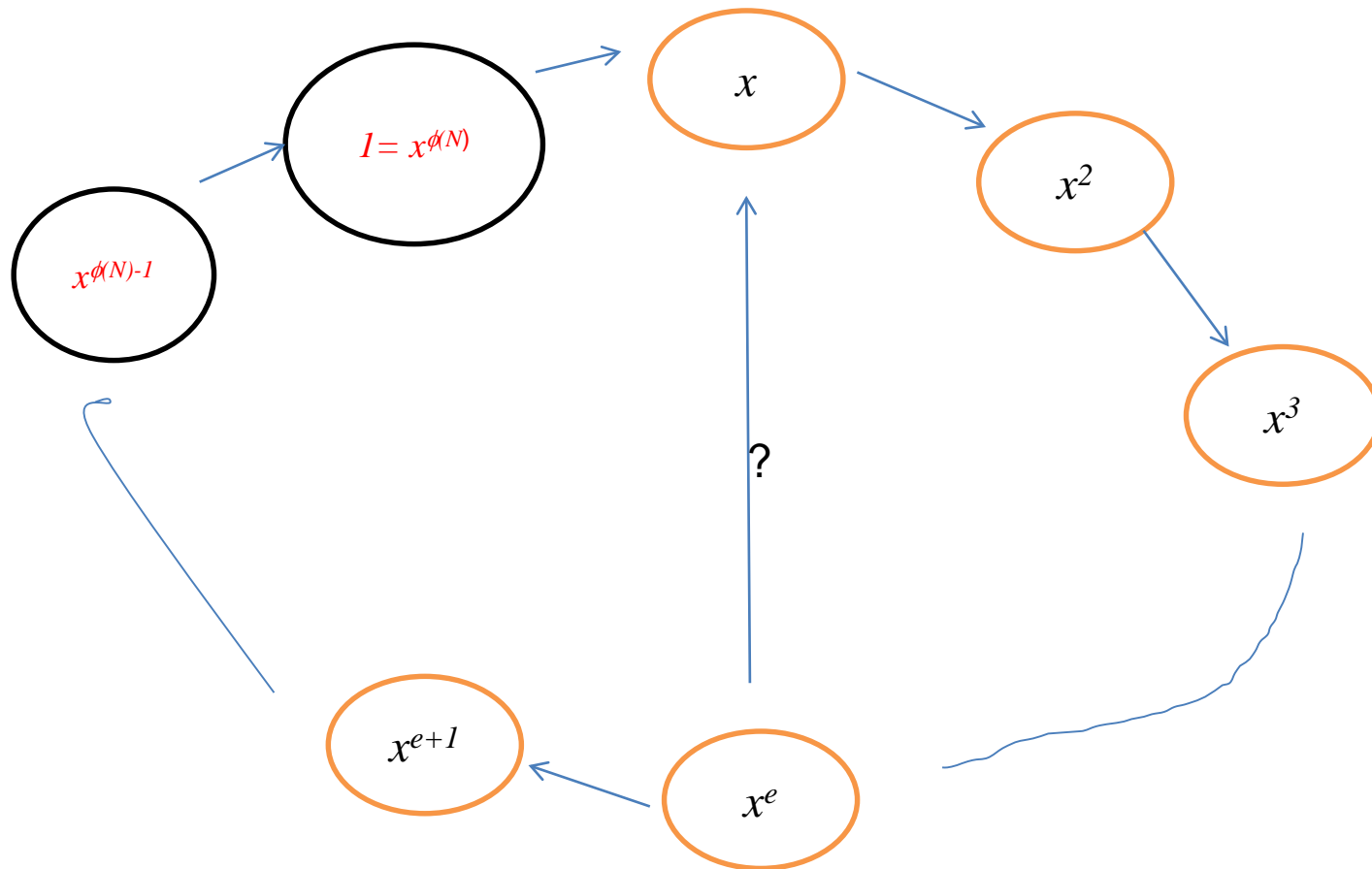
Lecture 2

Proof of RSA Encryption

Workshop 3: Workshop based on Lectures in Week3

Quiz 4

Why does it work? Alice has a trapdoor



The operations are in the 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 $n = (p-1)(q-1)$.

$$\begin{aligned} n - p - q + 1 \\ &= pq - p - q + 1 \\ &= (p-1)(q-1) \end{aligned}$$

RSA PKC (1978)

- Let $n = p \times q$; p, q are primes. Let the plain text and cipher text belong to integers modulo n and let (e, d) pair be computed such that

$$e \times d \equiv 1 \text{ mod } \phi(n)$$

(ϕ : Euler's totient function)

- For the RSA key parameter set $K = (n, p, q, e, d)$, define

$$E_k(x) = x^e \text{ mod } n$$

And

$$D_k(y) = y^d \text{ mod } n,$$

where $(x, y \text{ in } Z_n)$. 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 the plain text and cipher text belong to Z_{91} (residue Integers *modulo* 91). $\phi(n) = 12 \times 6 = 72$.
- 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 $5 \times 29 = 145 \bmod 72 = 1$
- Message $x = 11$
- $E_k(11) = C = 11^5 = 72$
- $D_k(72) = 72^{29} = 11$

Real-World RSA

We only illustrated some toy examples so far.

Let us look at more realistic RSA parameters.

- RSA-768: a 768-bit RSA modulus with 232-digit decimal representation:

$n =$
1230186684530117755130494958384962720772853569595334792197322452151726
40050726
3657518745202199786469389956474942774063845925192557326303453731548268
50791702
6122142913461670429214311602221240479274737794080665351419597459856902
143413.

- RSA Laboratories had issued a challenge to factor the above modulus.
- In 2009, this was broken!*

$n =$ 3347807169895689878604416984821269081770479498371376856891
2431388982883793878002287614711652531743087737814467999489 \times
3674604366679959042824463379962795263227915816434308764267
6032283815739666511279233373417143396810270092798736308917

- The current key size on Internet for secure operations is greater than 4000 bits.

* Thorsten Kleinjung, et. al, **Factorization of a 768-Bit RSA Modulus**. [CRYPTO 2010](#): 333-350

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. We will prove this fact in the coming lecture.
- 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 $c \pmod{n}$. 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.

Security of RSA

Brute Force attacks

Mathematical attacks

- Clearly, the security depends on the hardness of the **factorization problem**.
- If someone can obtain factors p or q , then they can find out $\phi(n)$ and can determine the decryption exponent itself.
- As a consequence of RSA encryption a new problem emerges called the **RSA problem**.
- It is stated as follows: Given $(n, e, c = M^e)$ determine e^{th} root of $c \bmod n$.
- This problem is also considered to be hard. The complexity is sub exponential on the key size.
- Quantum computing can help to factor n efficiently, however it may take some years before they are developed.

Complexity of Factorization

- 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:
 $\exp((c + O(1)b^{1/3} \log^{2/3}(b)))$,
for some integer $c < 2$
- May be quantum computers come to our rescue; earlier people were thinking it might thousands of years. But with rapid development of Quantum computing, the risk has been moved from “long term” to “medium term”.

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
$$m^e = c \pmod{n}.$$

Comment on Security of Known Schemes

- 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

- 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.
 - 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)**
 - 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.
 - 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.

Efficient Computation

- RSA requires an algorithm for exponentiation in mod n .
- I will give you an extended workshop sheet where we workout some mathematical results pertaining to RSA operations.
- You will have an opportunity to work on RSA key generation exercises in next week workshop.
- Can you write your own fast algorithm for RSA encryption and decryption?
- How do you choose primes for RSA?
- You need to know a bit more mathematics to understand the theory. We will not study in this topic in this subject.

Week 4



Lecture 1

Public Key Cryptography: Diffie-Hellman Protocol and RSA

Lecture 2

Proof of RSA Encryption + Chinese Remainder Theorem, Continued from
Week 3 Lecture 2

Workshop 3: Workshop based on Lectures in Week 3

Quiz 4