Topic 4: Public-key Cryptography (PKC)

Understand the principles of Public-Key Cryptography (PKC)

Source: Stallings' book, chapter 9

Overview

- Part 1
  - Introduction
- Part 2
  - Mathematical Basics for RSA
- Part 3
  - RSA Algorithm
  - Hybrid Cryptosystems
  - Conclusion

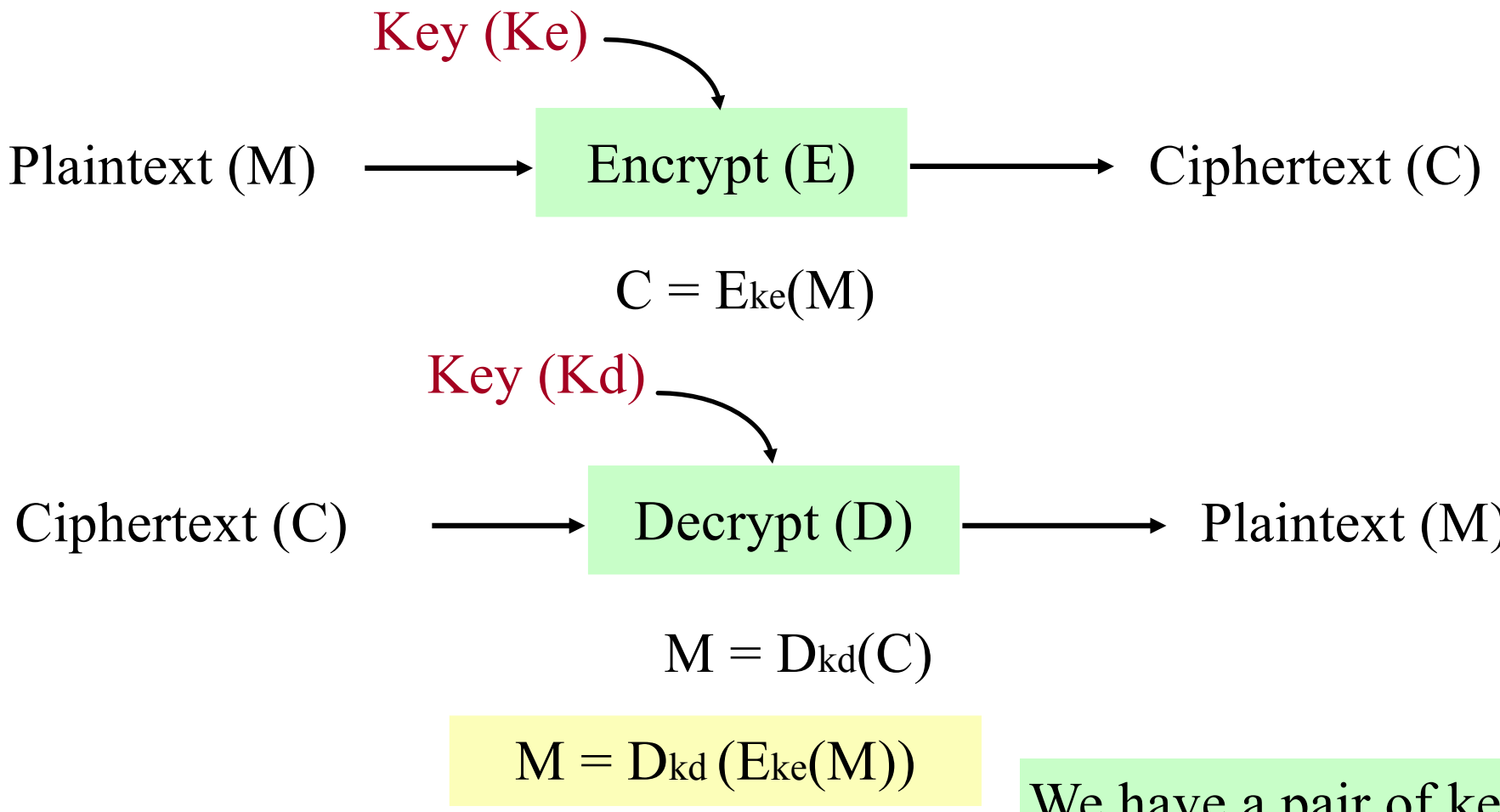
Introduction - Motivation

- Up to this point, all cryptographic schemes are based on shared secret keys - symmetric (or conventional) cryptography.
- The problems with symmetric cryptography - motivations
  - As two/more parties share the same key, non-repudiation can not be achieved without the involvement of a trusted third party.
  - A separate key is needed for each pair of users (or even for each ciphertext encryption – session key).
    - So an n-user system requires $n*(n-1)/2$ keys - the n^2 problem.
    - Generating and distributing these keys are a challenging problem.
    - Maintaining security for the keys already distributed is also challenging - can one remember so many keys?

Introduction - PKC Features

- In 1976, Diffie and Hellman (DH) first presented the notion of public key cryptography.
  - Keys could come in pairs - one public and one private; and it is infeasible to generate one key from the other; encryption produced by using one of the keys could only be reversed by the other key in the pair.

Introduction - PKC Features



We have a pair of keys here: {Ke, Kd}

Introduction - PKC Features

- PKC is based on the idea of a trapdoor function, or mathematically "hard" problems.
- The pair of private and public keys are related mathematically.
- Easy to generate keys (public and private).
- Hard to compute private key from public key.
- Easy to encrypt and decrypt if the right key is known.
- Hard to recover plaintext from ciphertext without the right key.

One-way function, f

$C = f(M)$ "Easy"

$M = f^{-1}(C)$ "Infeasible"

Trap-door one-way function, f

$C = f(K, M)$ "Easy" if K & M known

$M = f^{-1}(K, C)$ "Easy" if K & C known

$M = f^{-1}(K, C)$ "Infeasible" if K not known, C known

Introduction - Commonly used One-Way Function

- Integer factorization (used in RSA)
  - Finding prime factors of a large integer: $n=p*q$
  - n is known
  - find p and q

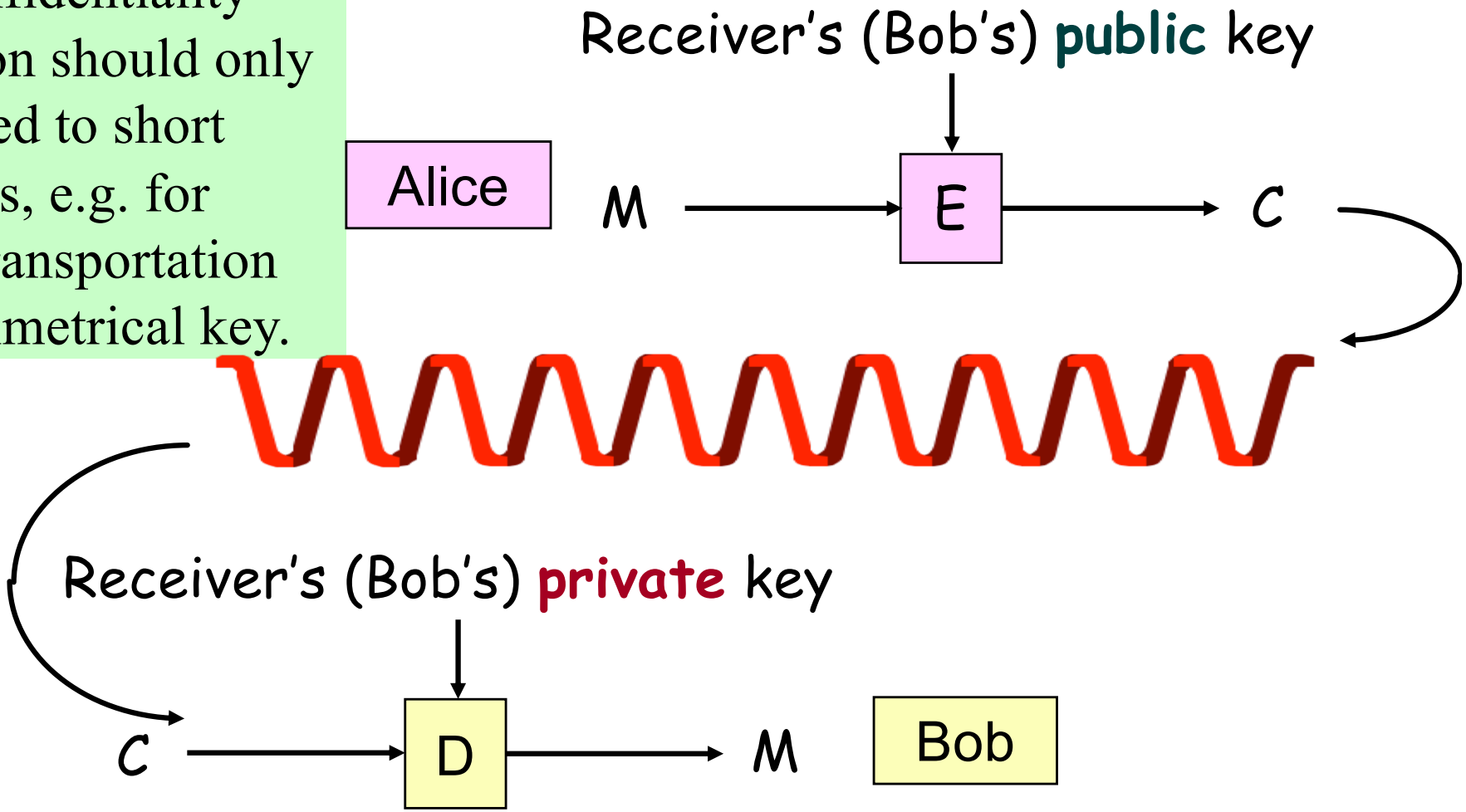
- Discrete logarithm (used in DSS and DH)
  - $a^x=b \bmod p$
  - a, b and p are known
  - finding an integer, x, satisfying this equation

Public-key Cryptographic Algorithms

- Since 1976, numerous public-key cryptographic algorithms have been proposed. Among those secure and practical public-key algorithms
  - some are suitable for **encryption** (+ **key distribution**);
  - some are only useful for **digital signatures, e.g. DSA** (Digital Signature Algorithm; or DSS - Digital Signature Standard);
  - some are for **key agreement,** e.g. **DH** algorithm;
  - only three algorithms, **RSA,** ElGamal and Rabin, works well for both encryption and digital signatures.

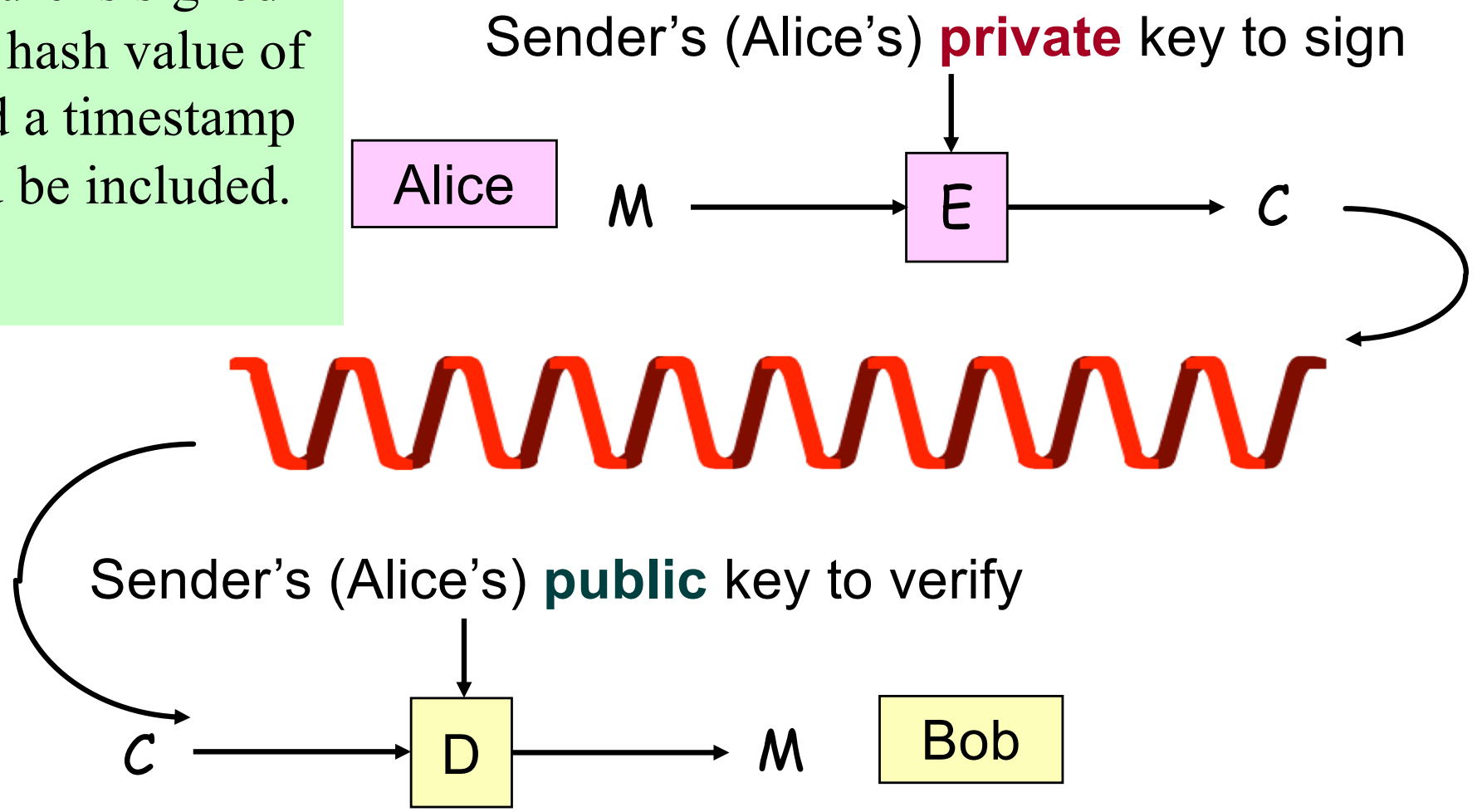
Introduction - Achieve Confidentiality (Secrecy)

- This confidentiality protection should only be applied to short messages, e.g. for secure transportation of a symmetrical key.



Introduction - Achieve Authenticity

- Signature is signed on the hash value of *M*, and a timestamp should be included.



Introduction - RSA Applications

- RSA is commonly used for
  - Confidentiality
    - Encrypt the plaintext *M* using **recipient's public key;**
    - As only the recipient has the corresponding private key, so *M* can only be read by the recipient.
  - Digital signature - message authenticity (message authentication or integrity) and non-repudiation of message origin
    - Sign *M* (actually the hash of *M*) using **sender's private key;**
    - As only the sender has this private key, so the message must have been signed by the sender.

Part 2 Overview

- Mathematical Basics for RSA

Mathematical Basics - Modular arithmetic

- Given some integer, n, the set of integers [$0, 1, \ldots, n-1$] is the set of possible remainders when one divides any integer by n.
- This set is called the set of residues/remainders modulo n.

- **Mathematical definition**

$$a = b \bmod n$$

means there exists an integer number k such that a can be represented as

$$a = k \cdot n + b$$

with the condition that: $0 \leq b \leq n-1$

Here we are not interested in the value of k; the important thing is its existence.

Mathematical Basics - Modular arithmetic

- Given integers, a, b, and $n \neq 0$, *a is congruent to b modulo n if and only if*

$$a-b = k \cdot n$$

for some integer k, *i.e. n divides $(a-b)$.*

- We call n the modulus, and b is remainder or residue of a modulo n.
- Examples:
  - 9 mod 5 = 4
  - 20 mod 9 = 2
  - $17 = 2 \ mod \ 5$ *since* $17-2 = 3 \cdot 5$
- $x = y$ if and only if

$$(x \bmod n) = (y \bmod n)$$

An example

The modulo operator is commutative with the basic arithmetic operations. For example, it does not matter whether you **first multiply**

$$18 \cdot 13 = 234 = 4 \ mod \ 10$$

or first calculate the modulus **and then multiply:**

$$18 \cdot 13 = 8 \cdot 3 \ mod \ 10$$

$$= 24 \ mod \ 10 = 4 \ mod \ 10$$

Mathematical Basics - Modular arithmetic

- Properties
  - $a = a \ mod \ n$
  - $a = b \ mod \ n \Leftrightarrow b = a \ mod \ n$
  - $a = b \bmod n \ \& \ b = c \bmod n \Rightarrow a = c \bmod n$
  - $(a + b) \bmod n = ((a \bmod n) + (b \bmod n)) \bmod n$
  - $(a \cdot b) \bmod n = ((a \bmod n) \cdot (b \bmod n)) \bmod n$
  - $a \cdot (b + c) \bmod n = (a \cdot b + a \cdot c) \bmod n$
  - $a \cdot x \bmod n = 1$ where x is an integer and called the multiplicative inverse of a; in this case, x can be written as a^{-1}, i.e. $a \cdot a^{-1} \bmod n = 1$.

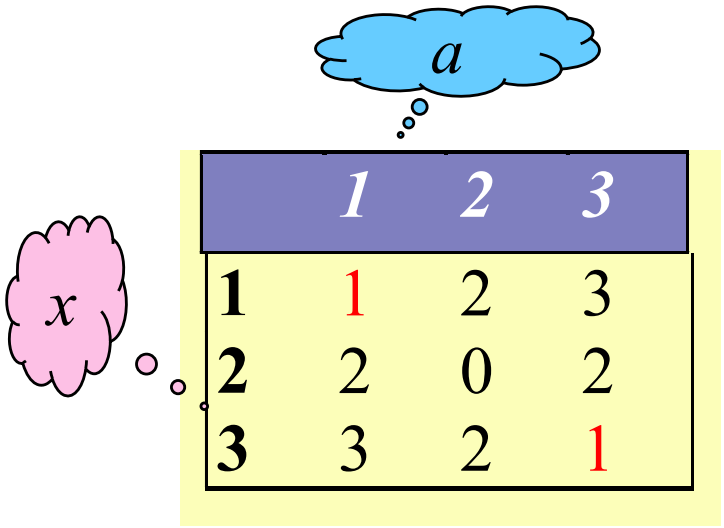
Mathematical Basics - Modular arithmetic

- Existence of **multiplicative inverse**
  - Given $a \in [0, n\text{-}1]$, find $x \in [0, n\text{-}1]$ such that $a \cdot x \bmod n = 1$;
  - E.g. as $3 \cdot 4 \bmod 11 = 12 \bmod 11 = 1$, so we say, 3 and 4 are each other's multiplicative inverse mod 11.

- iff n is a prime, a and n are relative prime, i.e. $\gcd(a, n)=1$, then $a \in (0, n\text{-}1]$ has a unique inverse modulo n.

- An integer $p > 1$ is a **prime number** if it is divisible only by itself and 1, e.g. 7.
- a and b are said to be **relatively prime** if only 1 can divide each of them, e.g. are 8 and 15 relatively prime?

Mathematical Basics - Multiplication table *Mod 4*



The inverse of 2 (mod 4) does not exist, because there isn't another number x in the finite field that can satisfy

$a.x = 1 \bmod 4.$

Not surprising!

There are two common divisors between *a* (2) and *n* (4), as 2 and 4 are not relatively prime.

Mathematical Basics - Multiplication table *Mod 11*

a

x

	1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*
1	1	2	3	4	5	6	7	8	9	10
2	2	4	6	8	10	1	3	5	7	9
3	3	6	9	1	4	7	10	2	5	8
4	4	8	1	5	9	2	6	10	3	7
5	5	10	4	9	3	8	2	7	1	6
6	6	1	7	2	8	3	9	4	10	5
7	7	3	10	6	2	9	5	1	8	4
8	8	5	2	10	7	4	1	9	6	3
9	9	7	5	3	1	10	8	6	4	2
10	10	9	8	7	6	5	4	3	2	1

Mathematical Basics - Multiplication Table Mod 11

- The *Table* gives the multiplication results for mod 11 (11 is a prime), the following can be noted:
  - In each row/column,
    - we can find *ALL* the integers in this set, and
    - each integer only appears **ONCE.**
  - For every integer, a, we can find another integer, x, that satisfies this equation:

    $a.x=1 \bmod 11, \quad (a.x=1 \bmod n)$

    That is, *every integer* in this set has its multiplicative inverse.
  - For example, 1 and 1; 2 and 6; 3 and 4; etc.

Mathematical Basics - Multiplication Table Mod 11

- Here $n=11$ is a prime,
  - Given a and b, there is a unique solution, x, that satisfies:

$$a.x = b \bmod n$$

  - All the integers in the set, $\{1, 2, \ldots, (n-1)\}$, are relative prime to n, i.e. there are $(n-1)=10$ such integers.

- **Euler's Totient/Phi Function (Euler's Theorem):**
  - If n is a prime, all of the positive integers, from 1 through to $(n-1)$, are relative prime to n and this is written as $\varphi(n)=(n-1)$.
  - If we have two different prime numbers, p and q, then for $n=p.q$,
    - $\varphi(n)=\varphi(p.q)=(p-1).(q-1)$
    - $a^{\varphi(n)}=1 \bmod n$

Part 3 Overview

- RSA Algorithm
- Hybrid Cryptosystems
- Conclusion

RSA Algorithm

- The algorithm was invented by Ron **R**ivest, Ali **S**hamir, and Leonard **A**dleman.
- It is by far the easiest to understand and implement.
- It has withstood years of cryptanalysis - remains by far most popular and well trusted scheme.
- The algorithm actually consists of two numbers, the modulus (represented by the letter ***n***) and the public exponent (represented by the letter ***e***).
- The modulus is the product of two very large prime numbers (100 to 400 digits), represented by the letters ***p*** and ***q***. ***p*** and ***q*** must be kept secret.

RSA Algorithm

- It is a block cipher. The block length is limited by some integer *n;* the plaintext and ciphertext, when being converted to integers, are between 0 and *n*-1.

- The algorithm has three functions:
  - **Key generation**
  - **Encryption**
  - **Decryption**

  } **Use the same mathematical function, but different keys.**

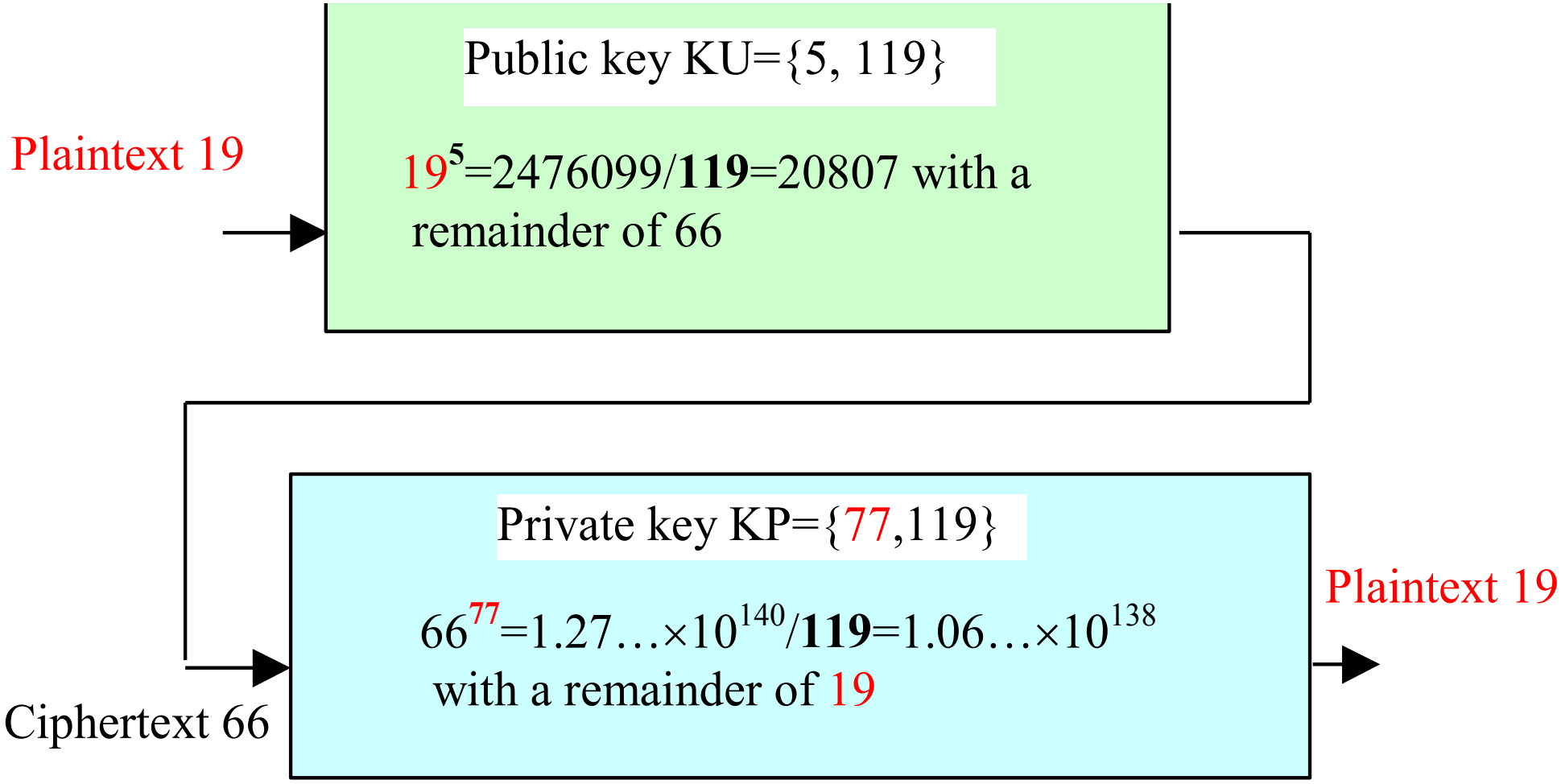
RSA Algorithm

- Key generation:
  - select two large primes (e.g. 200 digits) p and q
  - calculate $n = p.q$ and $\varphi(n) = (p - 1).(q - 1)$
  - select integer e relatively prime to $\varphi(n)$ & $1<e<\varphi(n)$
  - calculate $d = e^{-1} \bmod \varphi(n)$ (or $d.e = 1 \bmod \varphi(n)$)
  - **public key = {e, n }**
  - **private key ={d, n}**
  - *To summarise*:
    - p, q are private & chosen;
    - $n = p.q$ is public & calculated (but keep p, q secret);
    - e is public & chosen, and d is private & calculated.

RSA Algorithm

- Encryption:
  - represent the plaintext as an integer M in [0, n-1], i.e. $M < n$;
  - ciphertext: $C = M^e \bmod n$
- Decryption:
  - ciphertext: C
  - plaintext: $M = C^d \bmod n$
- An example of using RSA to encrypt a message
  - select p=7 and q=17
  - calculate $n = p.q = 119$ and $\varphi(n) = (p-1).(q-1) = 96$
  - select $e = 5$, relatively prime to $\varphi(n)$=96 and less than $\varphi(n)$
  - calculate d=77, such that $de = 1 \bmod \varphi(n)$ (= 96) and $d < 96$
  - let $M = 19$, then ciphertext $C = 19^5 \bmod 119 = 66$.

RSA Algorithm



Why RSA Works

- Euler's Theorem:
  - $a^{\phi(n)} \bmod n = 1$ where $\gcd(a,n)=1$
- In RSA, we have:
  - $n=p.q$
  - $\phi(n)=(p-1)(q-1)$
  - chose e & d to be inverses $\bmod \phi(n)$
  - hence $e.d=1+k.\phi(n)$ for some k
- hence:

$$C^d = M^{e.d} = M^{1+k.\phi(n)} = M^1.(M^{\phi(n)})^k$$
$$= M^1.(1)^k = M^1 = M \bmod n$$

RSA Algorithm - Standard

- PKCS#1 standard defines the use of RSA algorithm. It defines the key generation, encryption, decryption, digital signatures, verification, public key format, padding, and several other issues with RSA. It is probably the most widely used RSA standard, and most of the security protocols using RSA are also compatible with the PKCS#1 standard.
  - PKCS#1 standard - http://www.rsasecurity.com/rsalabs/pkcs/pkcs-1/index.html

RSA Algorithm - Some facts for the RSA

- Security of RSA relies on difficulty of finding d given $\{e, n\}$.
  - the problem of computing d from $\{e, n\}$ is computationally equivalent to the problem of factoring n
    - If one can factorise n, then he can find p and q, and hence calculated d;

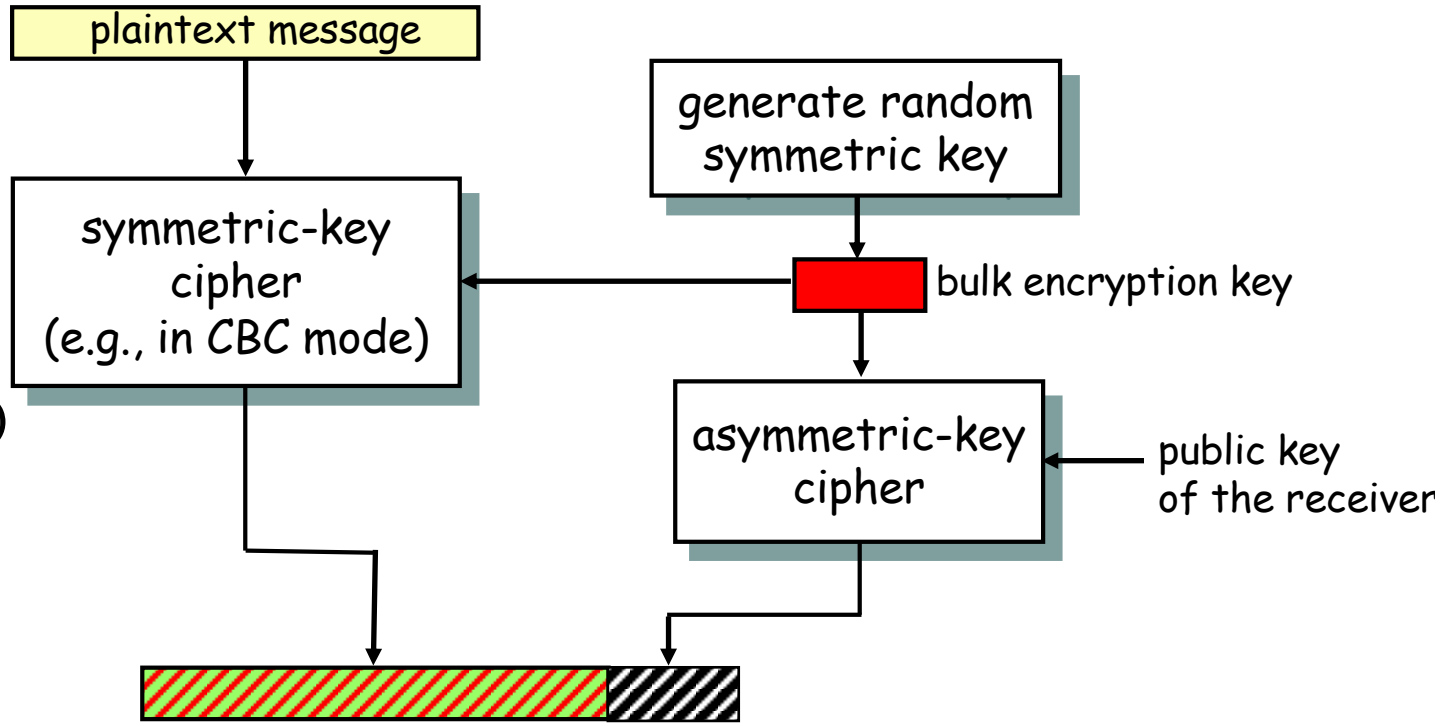
- p and q should differ in length by only a few digits, and both should be on the order of 100 - 200 digits or even larger.
  - n with 150 digits could be factored in about 1 year.
  - factoring n with 200 digits could take about 1000 years (assuming about 10^{12} operations per second).

Hybrid Cryptosystems

- Public key ciphers are much slower than symmetric key ciphers.
  - E.g. 1000 times slower in hardware, and 100 times slower in software, than DES.
- Symmetric ciphers
  - have key management problem.
  - can not provide non-repudiation service without the involvement of a trusted third party.
- So, usually, we combine them to get the strengths of both – this leads to the hybrid cryptosystem
  - Public cipher for symmetric key establishment/transportation and/or for digital signature generation.
  - Symmetric cipher for bulk encryption.

Hybrid Cryptosystems

- To speed the things up, hybrid encryption is used.
  - A symmetrical algorithm with a random session key (bulk encryption key) is used to encrypt the message;
  - A public-key algorithm is used to encrypt the random session (symmetric) key - digital enveloping.



What is the strength, and what is the remaining problem of this system?

Exercise Question – E4.1

- Name three application scenarios or cases where using RSA is preferable than using AES and name one application scenario where the use of AES is necessary.

Exercise Question – E4.2

- You are a recipient of p = 5, q = 7. You make the modulus n = 35 public. You also choose an exponent e = 5 and make that public too.
  Messages are sent to you, one letter at a time. Letters are coded into numbers as: A -> 0, B -> 1, and so on.

  Now, the following message has arrived for you:
  17 19 7 9 0 12 24

  Decrypt this message.

Conclusions

- Different from a symmetric-key cipher, in a public-key cipher, a pair of mathematically related keys are used, one for encryption and the other for decryption.
- Public key cryptography provides capabilities that can not be attained with symmetric cryptography, but it is too inefficient to be used alone - for large text encryption.

- PCK ciphers: RSA, DSS (Digital Signature Standard), DH (Diffie-Hellman), ...