CS 6355/4355: Cryptanalysis and Database Security

Topic 1: Foundations of Security and Classical Encryption Techniques

Lecturer: Rongxing LU

Email: RLU1@unb.ca Office: GE 114

Website: http://www.cs.unb.ca/~rlu1/

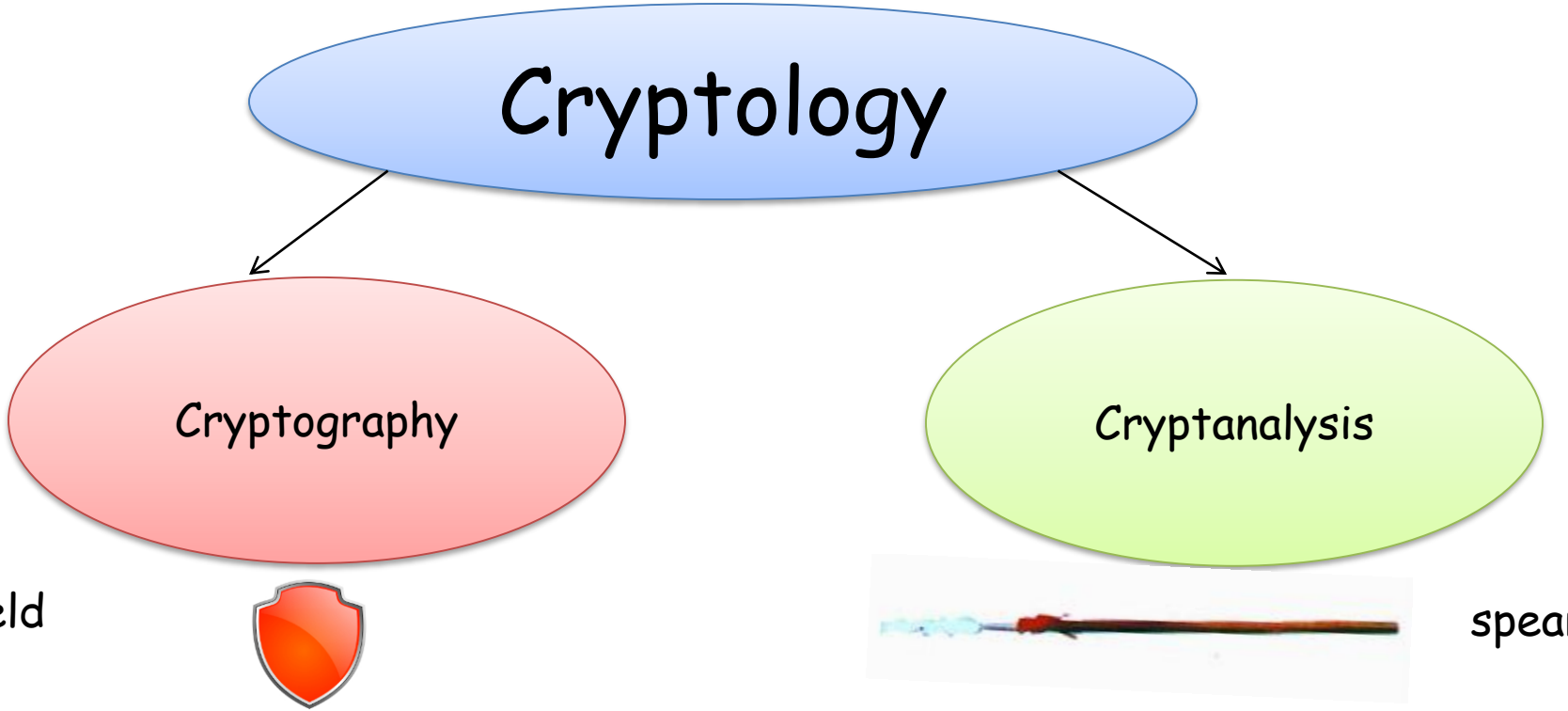
Faculty of Computer Science, University of New Brunswick

Introduction

- What is Cryptology?
  - The term cryptology is derived from the Greek words "kryptÓs", standing for "hidden", and "lÓgos", standing for "word". Consequently, the meaning of the term cryptology is best paraphrased as "hidden word". This paraphrase refers to the original intent of cryptology, namely to hide the meaning of specific words and to protect their confidentiality accordingly.

Cryptology = Cryptography + Cryptanalysis

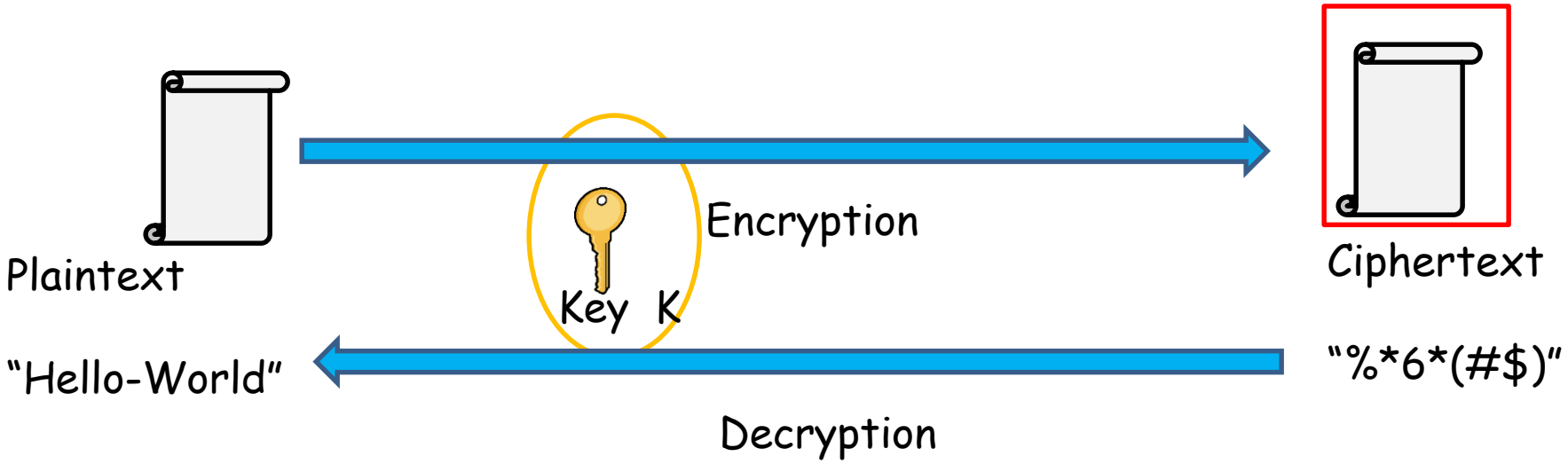
- Cryptology refers to the mathematical science and field of study that comprises both cryptography and cryptanalysis.



Cryptography

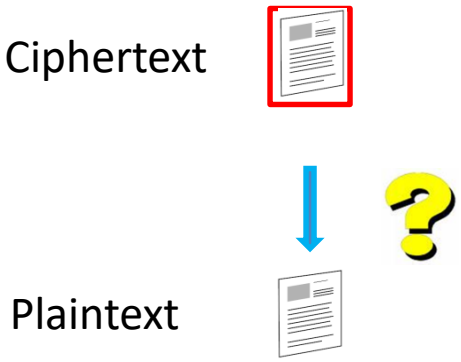
shield

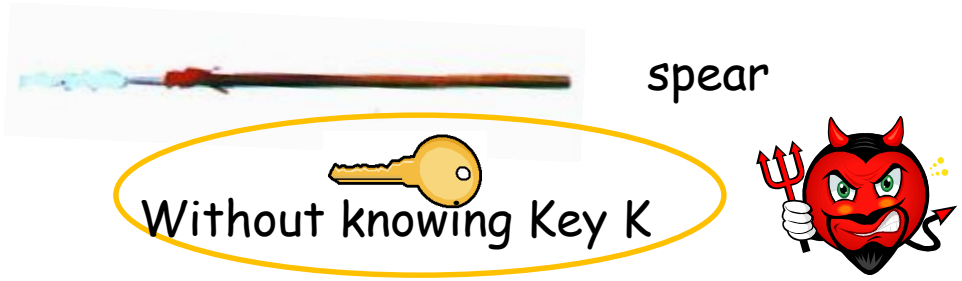
- The science of "Secret" writing
  - A cipher is a function which transforms a plaintext message into a ciphertext (cryptogram) by the process of encipherment
  - Plaintext is recovered from the ciphertext by the process of deciphering



Cryptanalysis

- The science and study of breaking ciphers, i.e., the process of determining the plaintext message from the ciphertext





Cryptographic algorithms and protocols can be grouped into four main areas:

- Symmetric encryption
  - Used to conceal the contents of blocks or streams of data of any size, including messages, files, encryption keys, and passwords
- Asymmetric (public key) encryption
  - Used to conceal small blocks of data, such as encryption keys and hash function values, which are used in digital signatures
- Data integrity algorithms
  - Used to protect blocks of data, such as messages, from alteration
- Authentication protocols
  - Schemes based on the use of cryptographic algorithms designed to authenticate the identity of entities

Cryptographic algorithms and protocols can be applied to Network Security

- The field of network and Internet security consists of:
  - measures to deter, prevent, detect, and correct security violations that involve the transmission of information

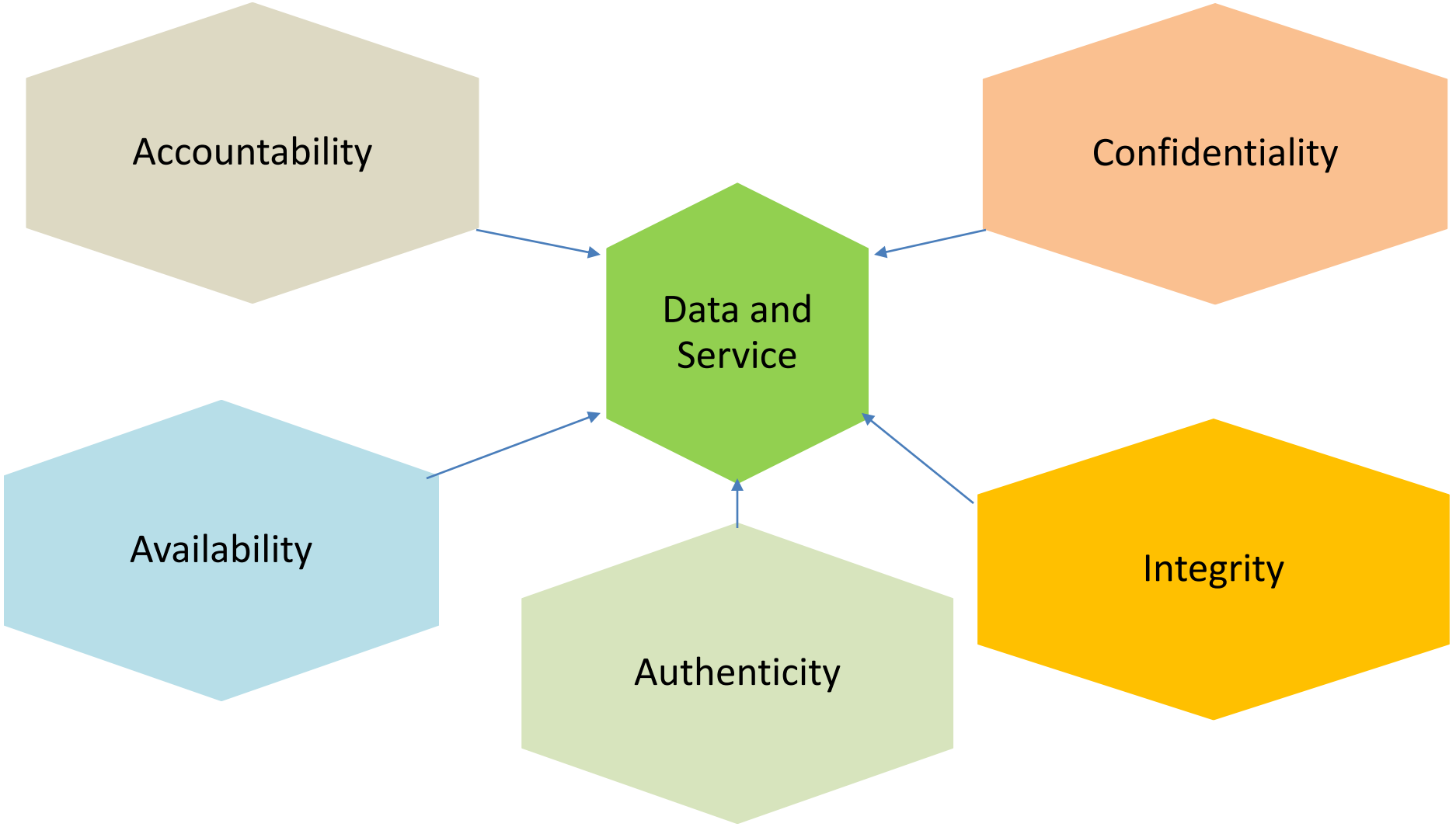
Cryptographic algorithms and protocols can be applied to Computer Security

- The NIST Computer Security Handbook defines the term computer security as:
  - "the protection afforded to an automated information system in order to attain the applicable objectives of preserving the integrity, availability and confidentiality of information system resources" (includes hardware, software, firmware, information/data, and telecommunications)

Computer Security Objectives

- Confidentiality
  - Data confidentiality
    - Assures that private or confidential information is not made available or disclosed to unauthorized individuals
  - Privacy
    - Assures that individuals control or influence what information related to them may be collected and stored and by whom and to whom that information may be disclosed
- Integrity
  - Data integrity
    - Assures that information and programs are changed only in a specified and authorized manner
  - System integrity
    - Assures that a system performs its intended function in an unimpaired manner, free from deliberate or inadvertent unauthorized manipulation of the system
- Availability
  - Assures that systems work promptly and service is not denied to authorized users

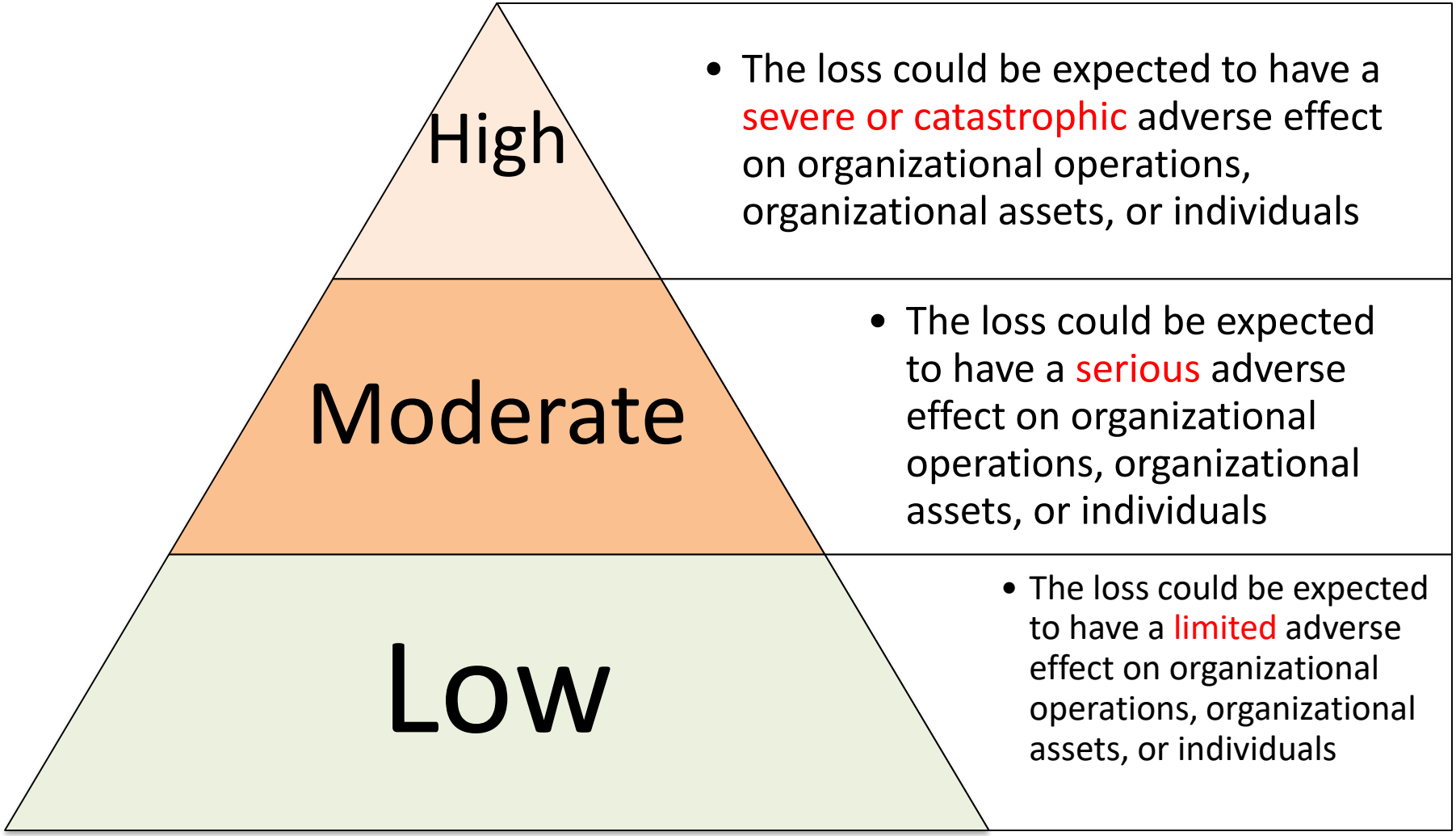
Essential Network and Computer Security Requirement



Authenticity & Accountability

- Authenticity:
  - The property of being genuine and being able to be verified and trusted; confidence in the validity of a transmission, a message, or a message originator.
- Accountability:
  - The security goal that generates the requirement for actions of an entity to be traced uniquely to that entity.
    - Nonrepudiation, intrusion detection and prevention, legal action

Breach of Security Levels of Impact



Computer Security Challenges

- Security is not simple
- Potential attacks on the security features need to be considered
- Procedures used to provide particular services are often counter-intuitive
- It is necessary to decide where to use the various security mechanisms
- Requires constant monitoring
- Is too often an afterthought

- Security mechanisms typically involve more than a particular algorithm or protocol
- Security is essentially a battle of wits between a perpetrator and the designer
- Little benefit from security investment is perceived until a security failure occurs
- Strong security is often viewed as an impediment to efficient and user-friendly operation

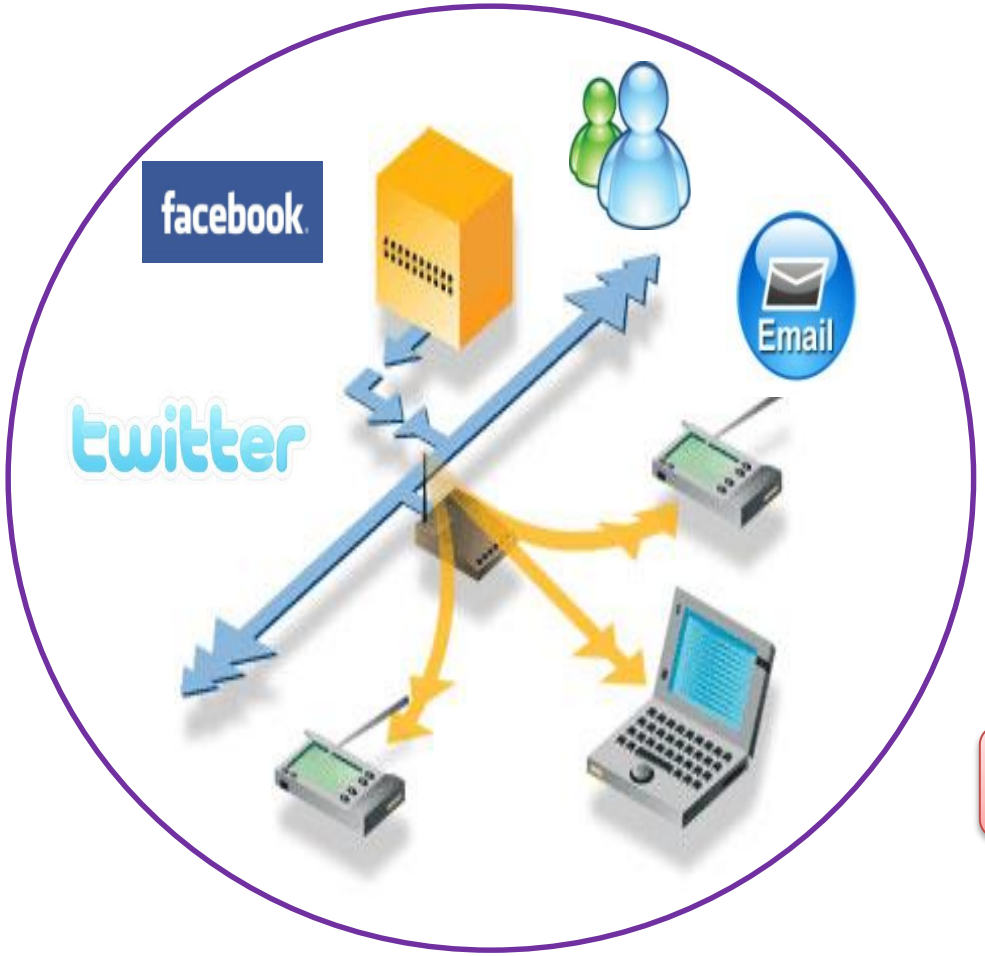
OSI Security Architecture

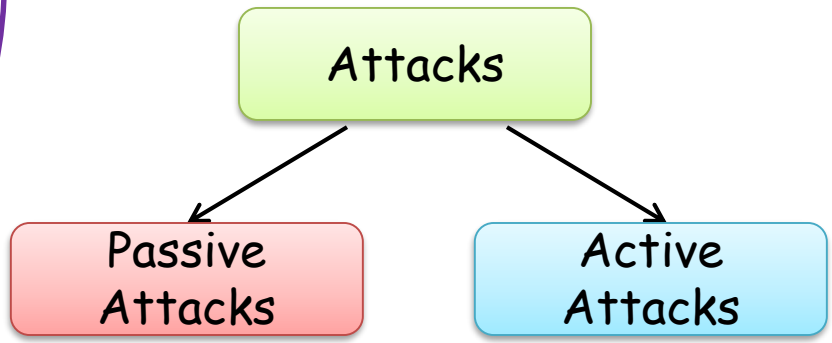
- Security attack
  - Any action that compromises the security of information owned by an organization
- Security mechanism
  - A process (or a device incorporating such a process) that is designed to detect, prevent, or recover from a security attack
- Security service
  - A processing or communication service that enhances the security of the data processing systems and the information transfers of an organization
  - Intended to counter security attacks, and they make use of one or more security mechanisms to provide the service

Threat & Attack

- Definitions taken from RFC 4949, Internet Security Glossary.
- **Threat:**
  - A potential for violation of security, which exists when there is a circumstance, capability, action, or event that could breach security and cause harm. That is, a threat is a possible danger that might exploit a vulnerability.
- **Attack:**
  - An assault on system security that derives from an intelligent threat; that is, an intelligent act that is a deliberate attempt (especially in the sense of a method or technique) to evade security services and violate the security policy of a system.

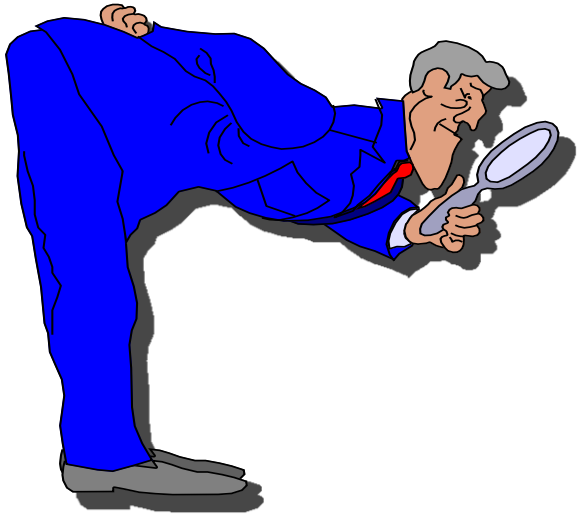
Security Attacks

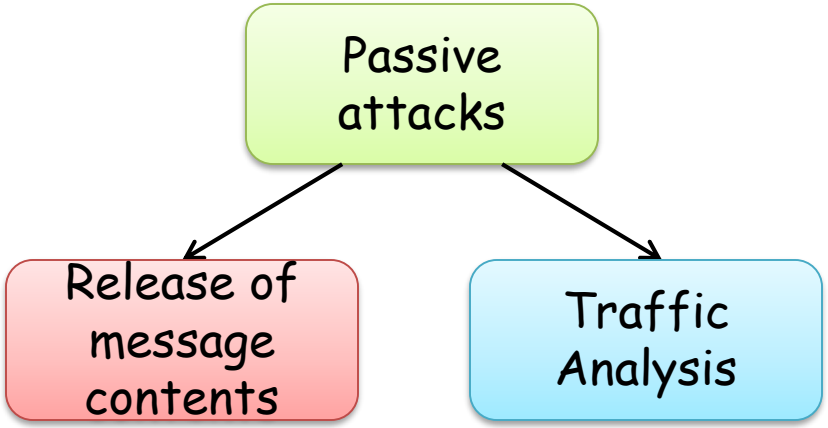




Passive Attacks

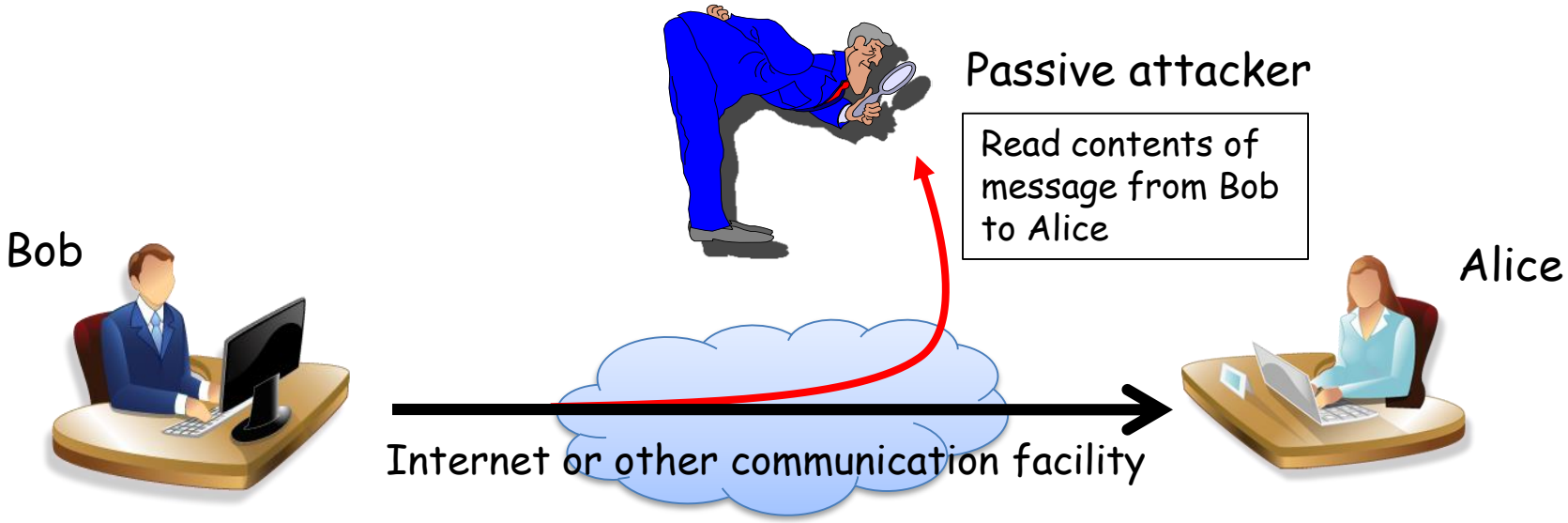
- **Passive attacks** are in the nature of eavesdropping on, or monitoring of, transmissions. The goal of the opponent is to obtain information that is being transmitted.
- Two types of passive attacks are **release of message contents** and **traffic analysis**.





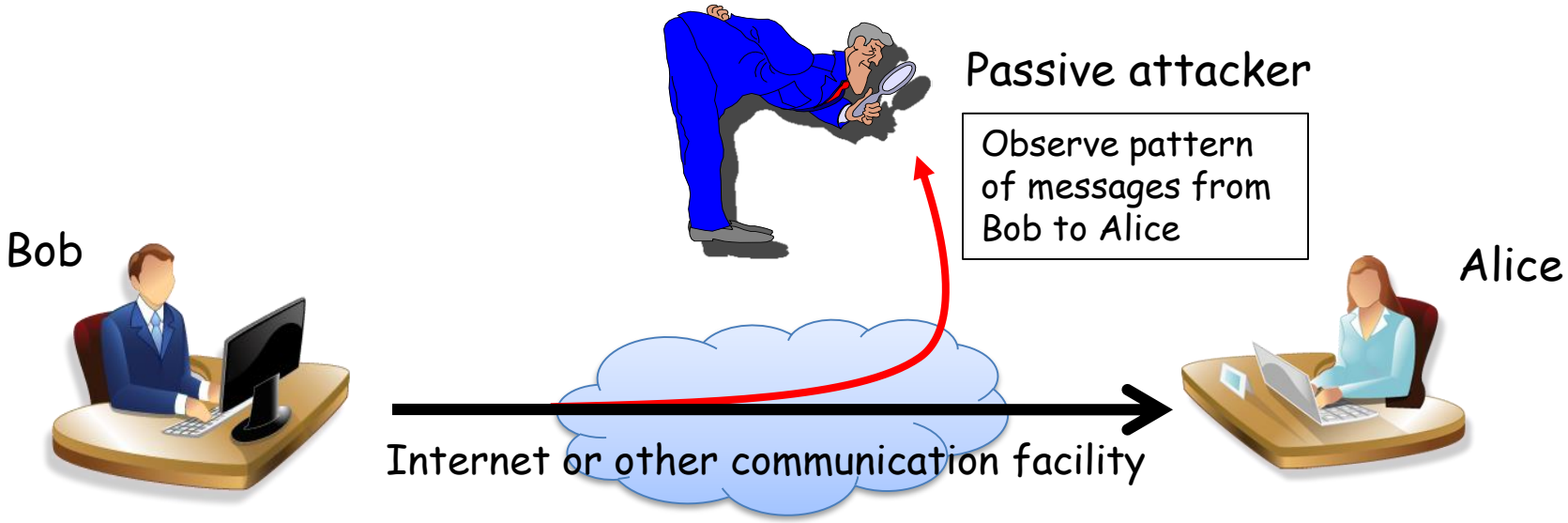
Release of message contents

- The release of message contents: A telephone conversation, an electronic mail message, and a transferred file may contain sensitive or confidential information.
- Our goal is to prevent a passive attacker from learning the contents of these transmissions.



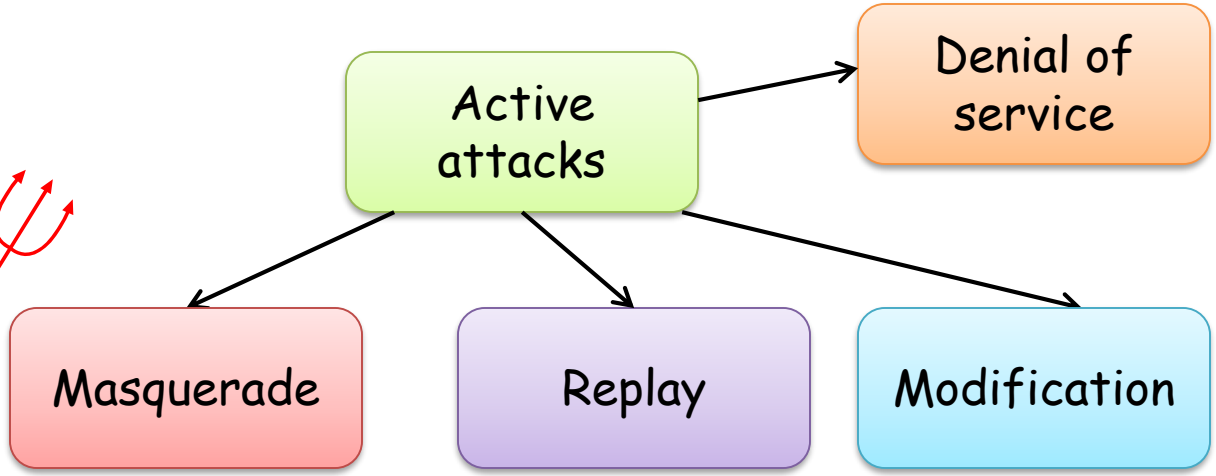
Traffic Analysis

- **Traffic Analysis:** If we had encryption protection in place, a passive attacker might still be able to observe the pattern of these messages. The attacker could determine the location and identity of communicating hosts and could observe the frequency and length of messages being exchanged, where the information might be useful in guessing the nature of the communication that was taking place.



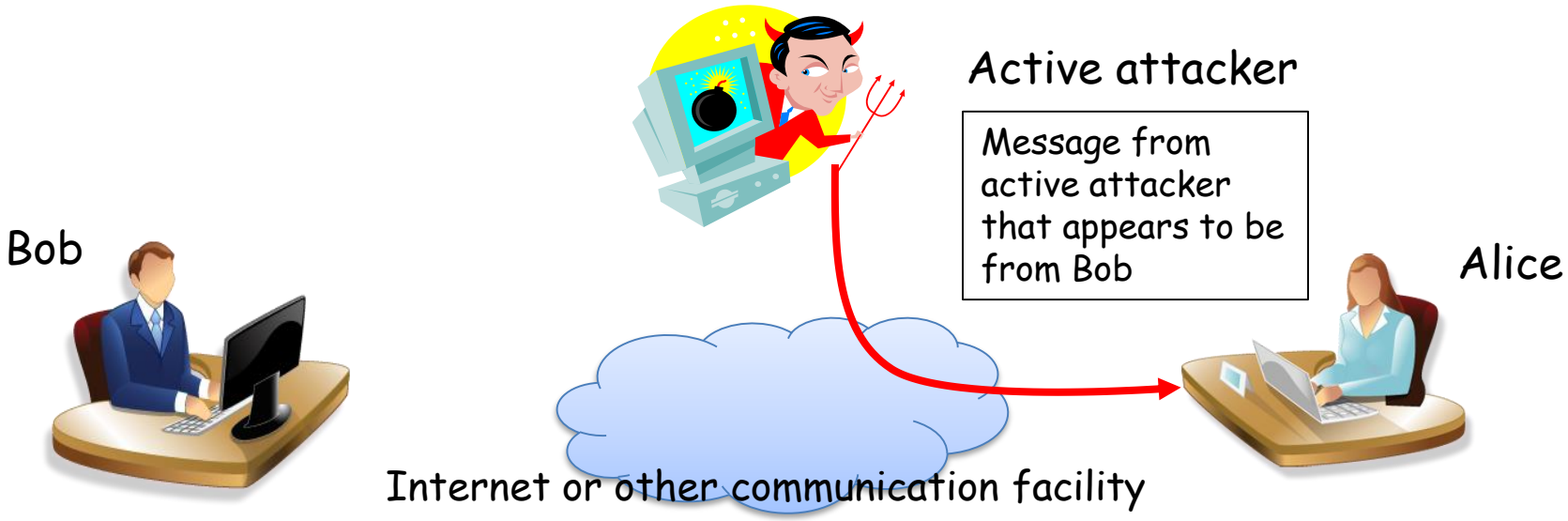
Active Attacks

- **Active attacks** involve some modification of the data stream or the creation of a false stream and can be subdivided into four categories: masquerade, replay, modification of messages, and denial of service.



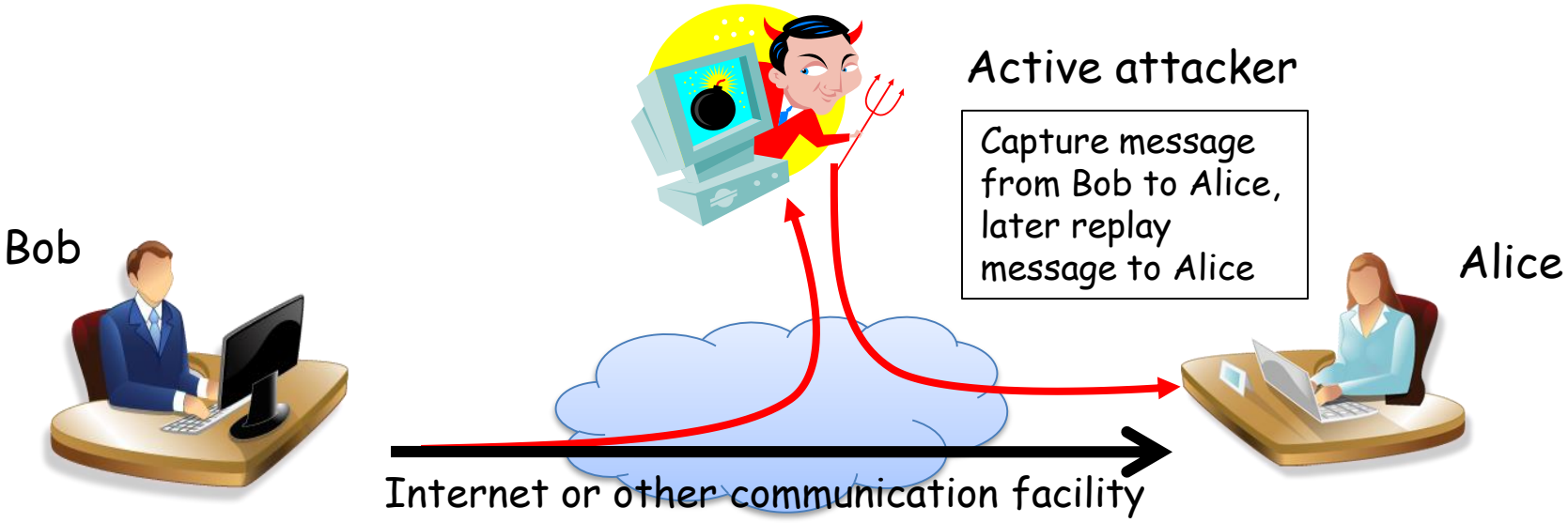
Masquerade

- **Masquerade:** takes place when one entity pretends to be a different entity



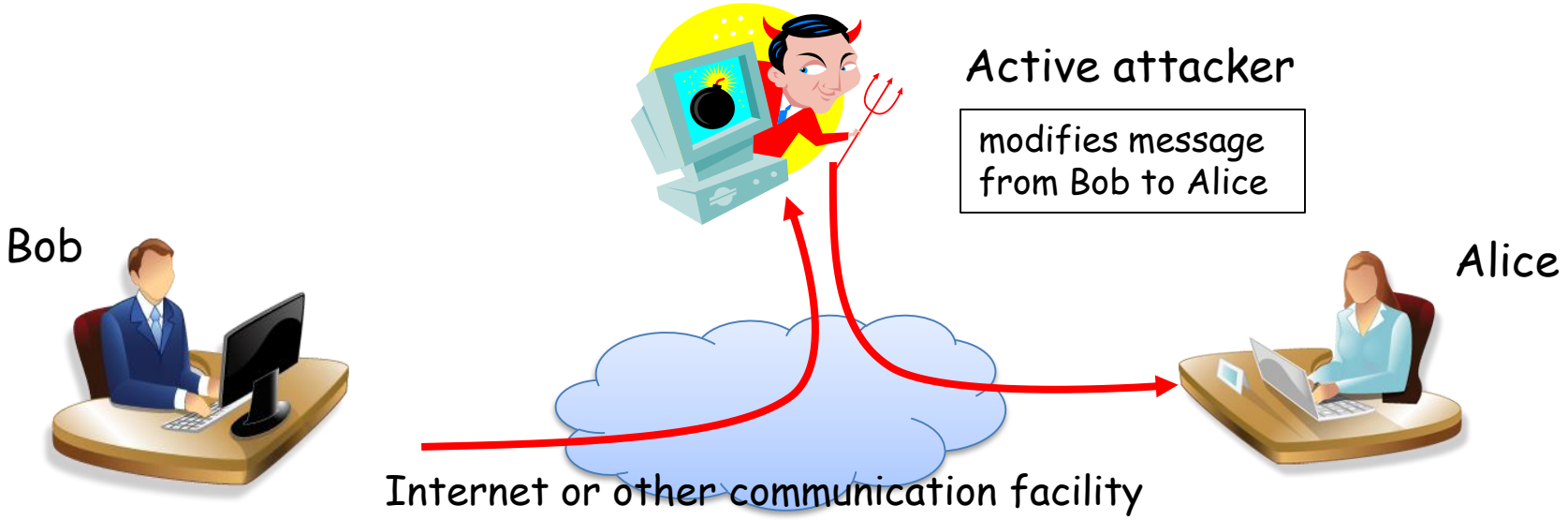
Replay

- **Replay:** involves the passive capture of a data unit and its subsequent retransmission to produce an unauthorized effect



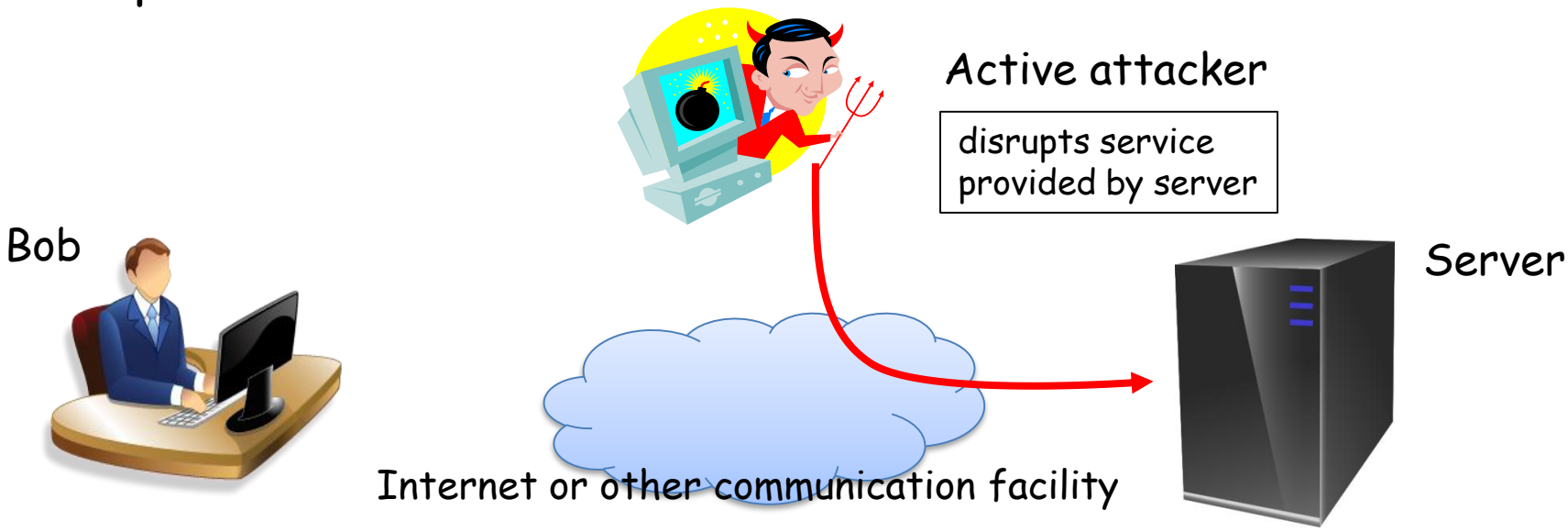
Modification

- **Modification:** means that some portion of a legitimate message is altered, or that messages are delayed or reordered, to produce an unauthorized effect. For example, a message meaning "Allow Alice to read confidential file accounts" is modified to mean "Allow Alice to delete confidential file accounts."



Denial of Service

- **Denial of Service (DoS):** prevents the normal use or management of communications facilities.
  - DoS attack may have a specific target; for example, an entity may suppress all messages directed to a particular destination (e.g., the security audit service).
  - Another form of DoS is the disruption of an entire network, either by disabling the network or by overloading it with messages so as to degrade performance.



Comparisons

• Passive Attacks

- Passive attacks are very difficult to detect because they do not involve any alteration of the data.
- Typically, the message traffic is sent and received in an apparently normal fashion and neither the sender nor receiver is aware that a third party has read the messages or observed the traffic pattern.
- However, it is feasible to prevent the success of these attacks, usually by means of encryption. Thus, the emphasis in dealing with passive attacks is on prevention rather than detection.

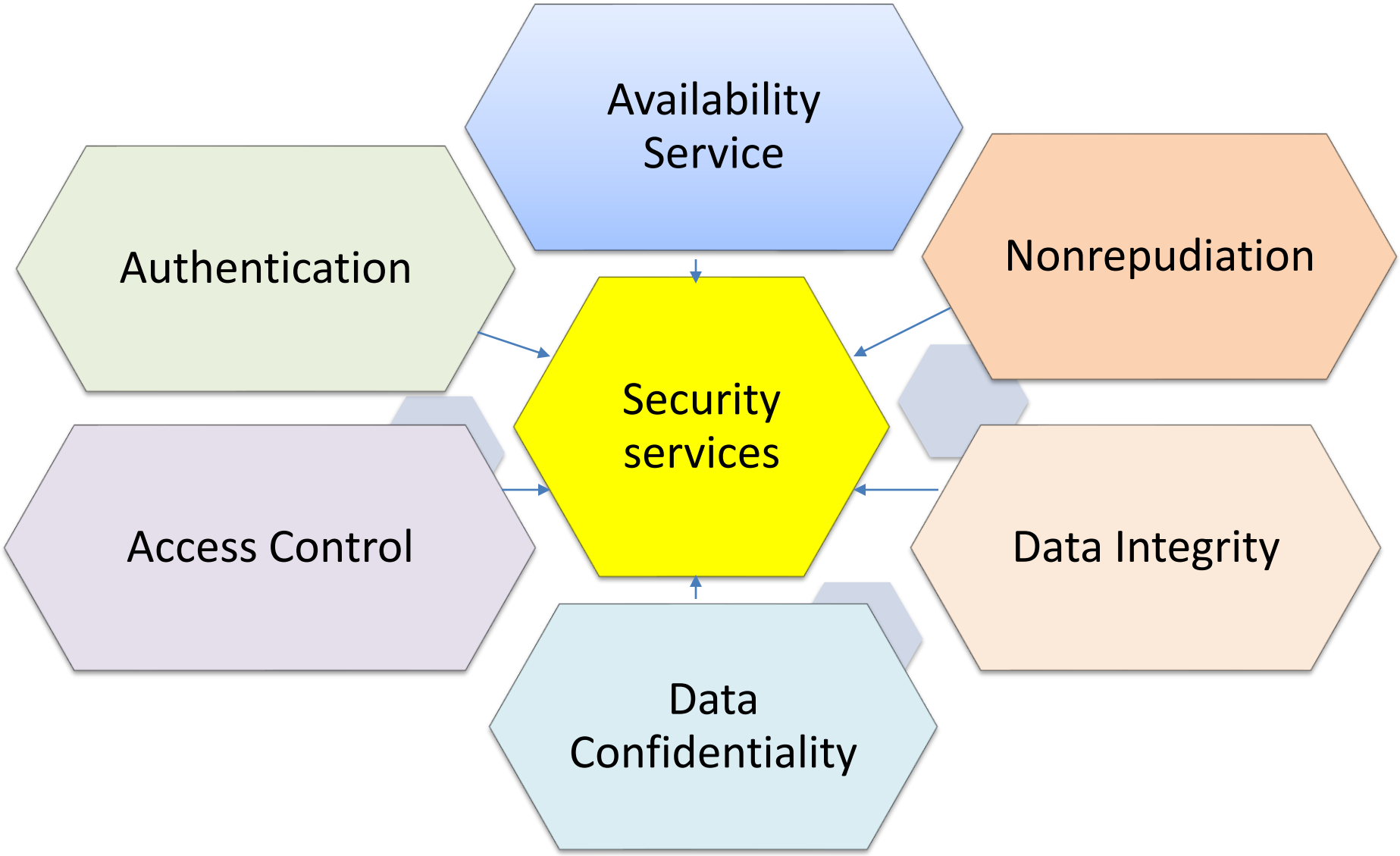
• Active Attacks

- Active attacks present the opposite characteristics of passive attacks. Whereas passive attacks are difficult to detect, measures are available to prevent their success.
- On the other hand, it is quite difficult to prevent active attacks absolutely, because of the wide variety of potential physical, software, and network vulnerabilities.
- Instead, the goal is to detect active attacks and to recover from any disruption or delays caused by them. If the detection has a deterrent effect, it may also contribute to prevention.

Security Services

- Defined by X.800 as:
  - A service provided by a protocol layer of communicating open systems and that ensures adequate security of the systems or of data transfers
- Defined by RFC 4949 as:
  - A processing or communication service provided by a system to give a specific kind of protection to system resources

Security Services



Authentication

- Concerned with assuring that a communication is authentic
  - In the case of a single message, assures the recipient that the message is from the source that it claims to be from
  - In the case of ongoing interaction, assures the two entities are authentic and that the connection is not interfered with in such a way that a third party can masquerade as one of the two legitimate parties

Two specific authentication services are defined in X.800:

- Peer entity authentication
- Data origin authentication

Access Control

- The ability to limit and control the access to host systems and applications via communications links
- To achieve this, each entity trying to gain access must first be identified, or authenticated, so that access rights can be tailored to the individual

Data Confidentiality

- The protection of transmitted data from passive attacks
  - Broadest service protects all user data transmitted between two users over a period of time
  - Narrower forms of service includes the protection of a single message or even specific fields within a message
- The protection of traffic flow from analysis
  - This requires that an attacker not be able to observe the source and destination, frequency, length, or other characteristics of the traffic on a communications facility

Data Integrity

- Can apply to a stream of messages, a single message, or selected fields within a message
- Connection-oriented integrity service, one that deals with a stream of messages, assures that messages are received as sent with no duplication, insertion, modification, reordering, or replays
- A connectionless integrity service, one that deals with individual messages without regard to any larger context, generally provides protection against message modification only

Nonrepudiation

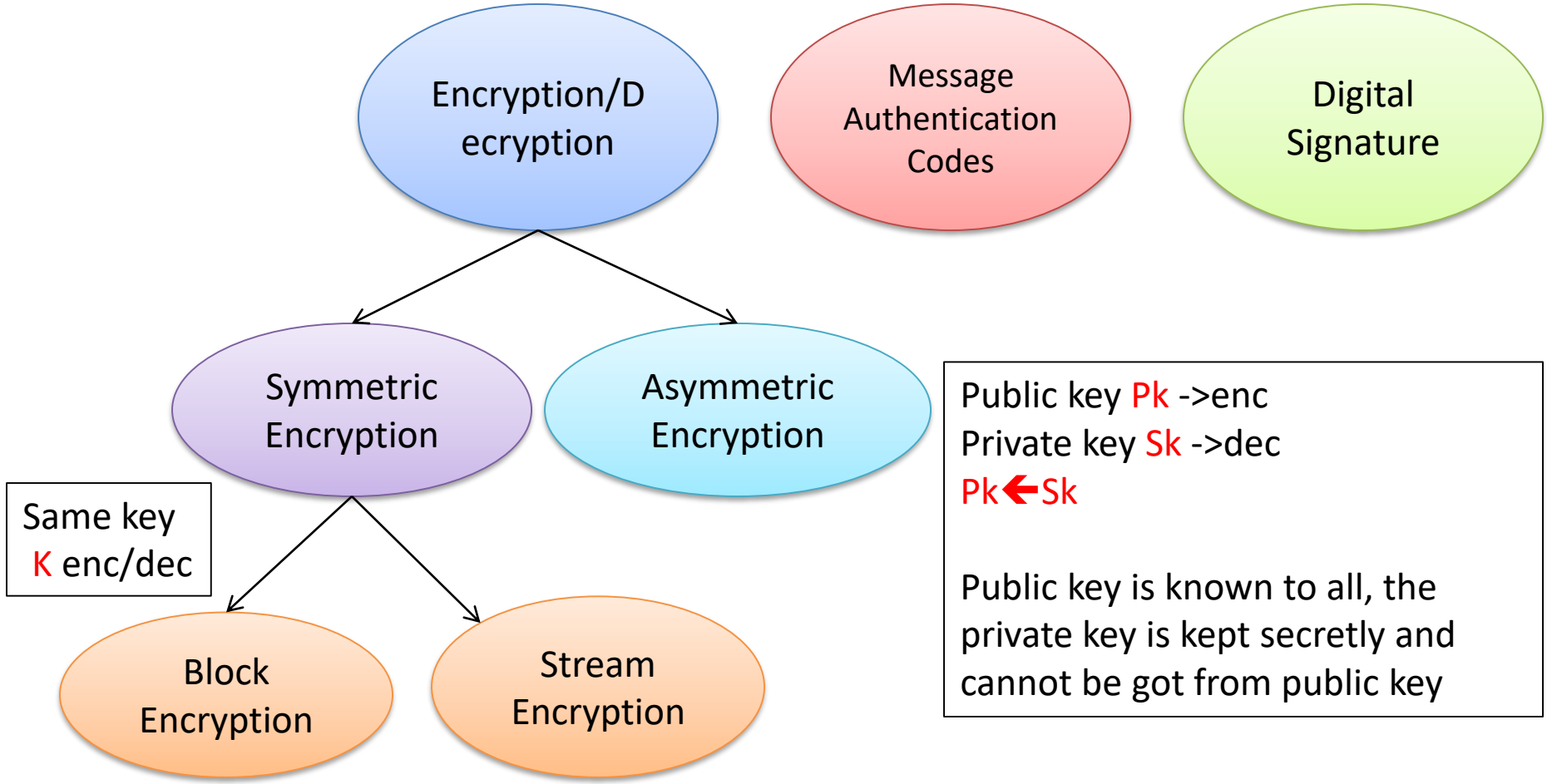
- Prevents either sender or receiver from denying a transmitted message
- When a message is sent, the receiver can prove that the alleged sender in fact sent the message
- When a message is received, the sender can prove that the alleged receiver in fact received the message

Availability Service

- Protects a system to ensure its availability
- This service addresses the security concerns raised by denial-of-service attacks
- It depends on proper management and control of system resources and thus depends on access control service and other security services

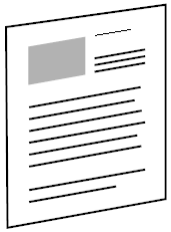
Security Mechanisms

- Cryptographic Tools



Encryption/Decryption

Symmetric Encryption



message M

C = Enc(M, K)

M = Dec(C, K)

Key K

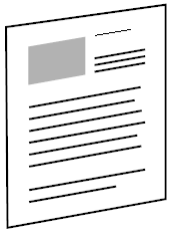
ciphertext C

Message Space

Key Space

recover M or K/Sk in a reasonable time

Asymmetric Encryption



message M

Public key Pk

C = Enc(M, Pk)

M = Dec(C, Sk)

Private key Sk

ciphertext C

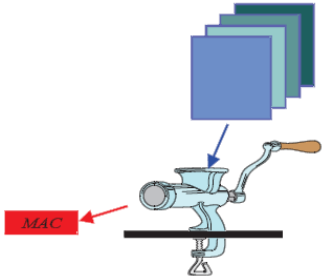
Message Authentication/Hashing

MD5: Message Digest 5
SHA1: Secure Hash Algorithm 1

message M

H(M)

digest
Allow to detect any modification of M



M1

H(M1)

M2

H(M2)

No source authentication

With 1-bit difference

With unpredictable 50% change

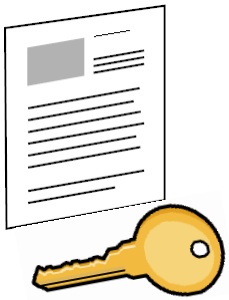
M1 H(M1)

M2

No-collision
Given H(M1), it is impossible to find another message M2, whose hash value H(M2)=H(M1)

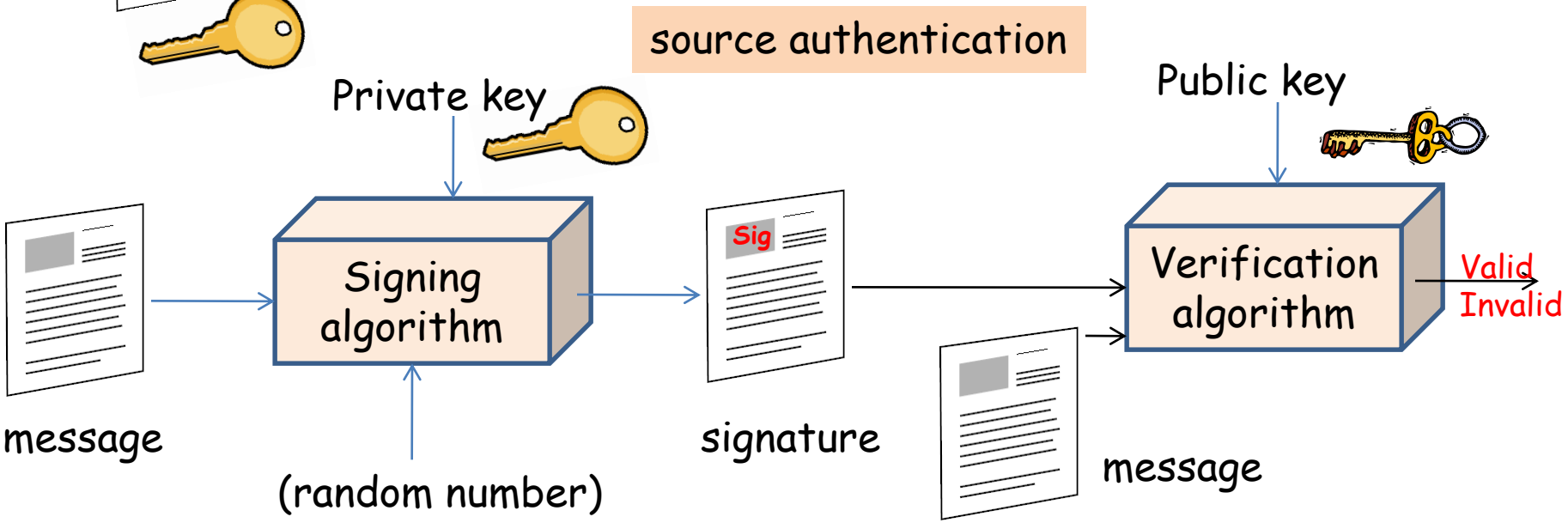
Digital Signatures

Handwriting signature: prove the authors and content of a message to a third party in a later time

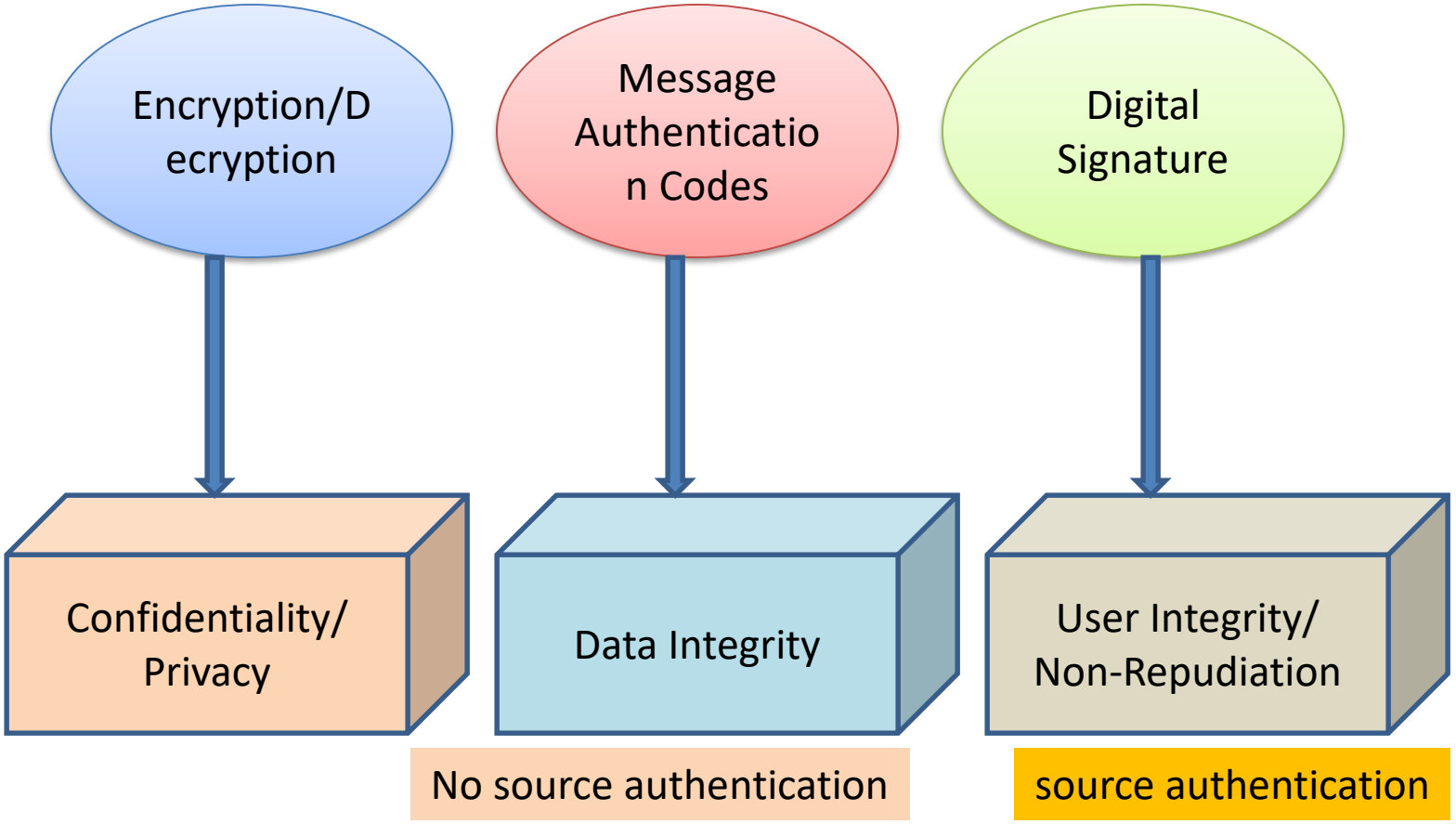


Digital Analogy of handwriting signature

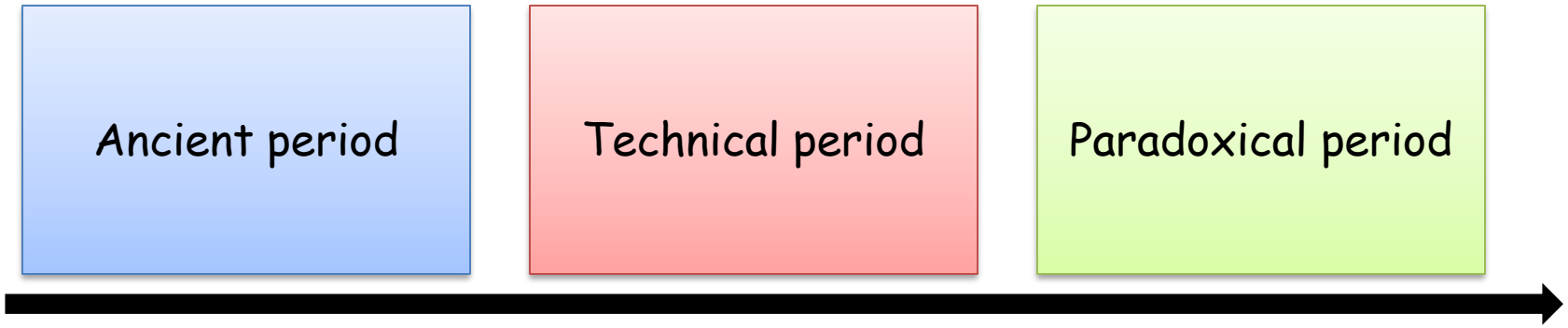
Digital signature uses private key Sk to sign an electronic file



Cryptographic Objectives



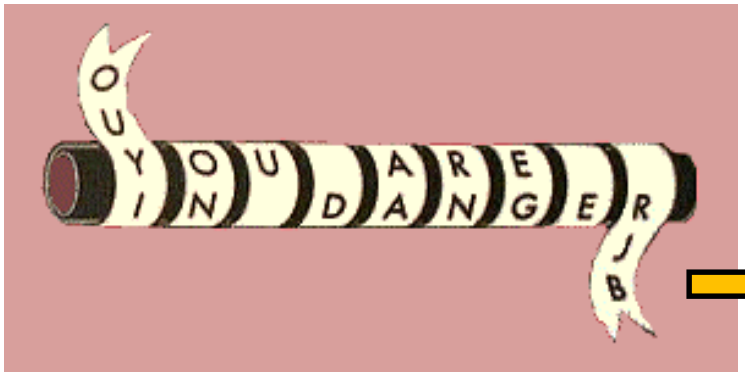
History of Cryptography



- **Ancient period:** (– until 1918)
  - with relatively simple algorithms that were designed and implemented manually.
- **Technical period:** (from 1919 until 1975)
  - Extensive use of encrypting electro-mechanical machines, especially in the period of the Second World War (cipher machine).
- **Paradoxical period:** (from Mid-1970s until -)
  - More pervasive use of computers in recent decades, supported by solid mathematical basis (number theory, group, ring, field theory, …) (cryptography on computer)

Ancient period

transposition of letters
substitution (replace letters)

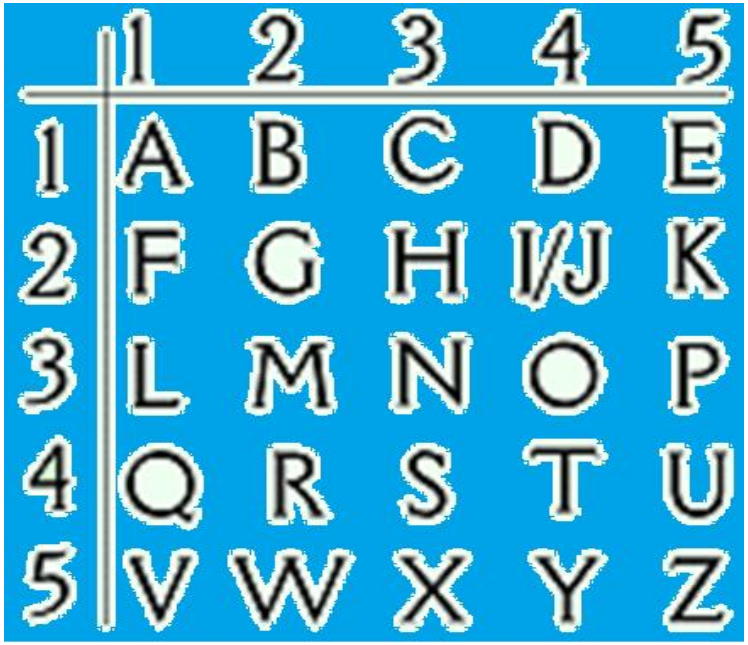


Historical Story 1

YOU ARE IN DANGER

In 405 BC the Greek general LYSANDER OF SPARTA was sent a coded message written on the inside of a servant's belt. When LYSANDER wound the belt around a wooden baton the message was revealed. The message warned LYSANDER that Persia was about to go to war against him. He immediately set sail and defeated the Persians.

Historical Story 2



	1	2	3	4	5
1	A	B	C	D	E
2	F	G	H	I/J	K
3	L	M	N	O	P
4	Q	R	S	T	U
5	V	W	X	Y	Z

The Greeks also invented a code which changed letters into numbers. A is written as 11, B is 12, and so on. So WAR would read 52 11 42.

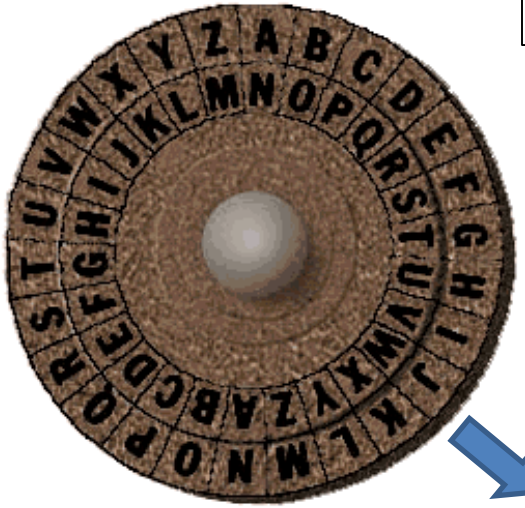
This kind of code cipher was still being used two thousand years later during the First World War.

Plaintext	H	E	L	L	O	W	O	R	L	D
Ciphertext	23	15	31	31	34	52	34	42	31	14

Historical Story 3

Caesar Shift Cipher:

The Roman Emperor CAESAR invented his own simple code:
Each letter substituted by shifting n=3 places
His famous phrase VENI, VIDI, VICI ("I came, I saw, I conquered") would have read YHQL YLGL YLFL
V—Y, E—H, N—Q, I—L, D—G, C—I



Cipher Disk

Plaintext	H	E	L	L	O	W	O	R	L	D
Ciphertext	U	R	Y	Y	B	J	B	E	Y	Q

Technical Period

Machine used for substitution

Historical Story 4

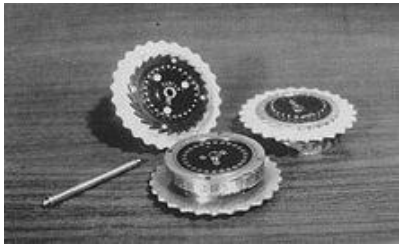
During the Second World War

Before war broke out in 1939 the Germans had planned a special way of keeping their communications secret. The army, navy and air force were told to encode their messages using cipher machines called ENIGMA

ENIGMA

https://www.youtube.com/watch?v=mXZNayEPFKc

ENIGMA



Performing substitutions

key space = a total of 10^{17} combinations, i.e., ENIGMA can put a message into code in around Million, Million, Million different ways.

- The Enigma cipher machine looked like a traditional typewriter in a wooden box. An electric current went from the keyboard through a set of rotors and a plugboard to light up the 'code' alphabet.
- At least once a day the Germans changed the order of the rotors, their starting positions and the plugboard connections.
- To decipher a message sent using Enigma, you had to work out exactly how all of these had been set.
- In the 1930's Polish cipher experts secretly began to try to crack the code. Obtained information on its usage: 1) daily code book indicated rotors and orientation; 2) a different orientation key for each message. Just before war broke out they managed to pass models and drawings of Enigma to British and French code-breakers. Finally, Enigma was broken.

Paradoxical Period

Symmetric Cryptography

Asymmetric Cryptography

- The draft *Data Encryption Standard* (DES) was published in the U.S. Federal Register on 17 March 1975, which is an effort to develop secure electronic communication facilities for businesses such as banks and other large financial organizations. DES was adopted and published as a Federal Information Processing Standard Publication in 1977. The release of its specification stimulated an explosion of public and academic interest in cryptography.
- In 1976, *New Directions in Cryptography* by Whitfield Diffie and Martin Hellman was published, which is a very significant step in the history of cryptography. In the paper, it introduced a radically new method of distributing cryptographic keys, which went far toward solving one of the fundamental problems of cryptography, key distribution, and has become known as Diffie-Hellman key exchange. In addition, it also stimulated the birth of a new class of enciphering algorithms, the asymmetric cryptography.

Kerckhoffs Principles

1. The system must be substantially, if not mathematically, undecipherable;
2. The system must not require secrecy and can be stolen by the enemy without causing trouble;
3. It must be easy to communicate and remember the keys without requiring written notes, it must also be easy to change or modify the keys with different participants;
4. The system ought to be compatible with telegraph communication;
5. The system must be portable, and its use must not require more than one person;
6. Finally, regarding the circumstances in which such system is applied, it must be easy to use and must neither require stress of mind nor the knowledge of a long series of rules.

- A cryptosystem should be secure even if everything about the system, except the key, is public knowledge.
- Kerckhoffs' principle was reformulated by Claude Shannon as "The enemy knows the system." In that form, it is called Shannon's maxim.

RSA Public Key Cryptosystem

- R. Rivest, A. Shamir, L. Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Communications of the ACM, Vol. 21 (2), pp.120-126. 1978.
- **History:** Famous Paper Rejection (anonymous reviewer's comments)
  - *"R.L. RIVEST, A. SHAMIR, AND L. ADELMAN " A Method for Obtaining Digital Signatures and Public-Key" Cryptosystems." According to the (very short) introduction, this paper purports to present a practical implementation of Diffie and Hellman's public-key cryptosystem for applications in the electronic mail realm. If this is indeed the premise, the paper should be rejected both for a failure to live up to it and for its irrelevance. I doubt that a system such as this one will ever be practical. The paper does a poor job of convincing the reader that practicality is attainable.*

P & Q PRIME
$N = PQ$
$ED \equiv 1 \text{ MOD } (P-1)(Q-1)$
$C = M^E \text{ MOD } N$
$M = C^D \text{ MOD } N$

Simple Classical Encryption Techniques

- Transposition
- Substitution
- Polyalphabetic Substitution
- Vigenere

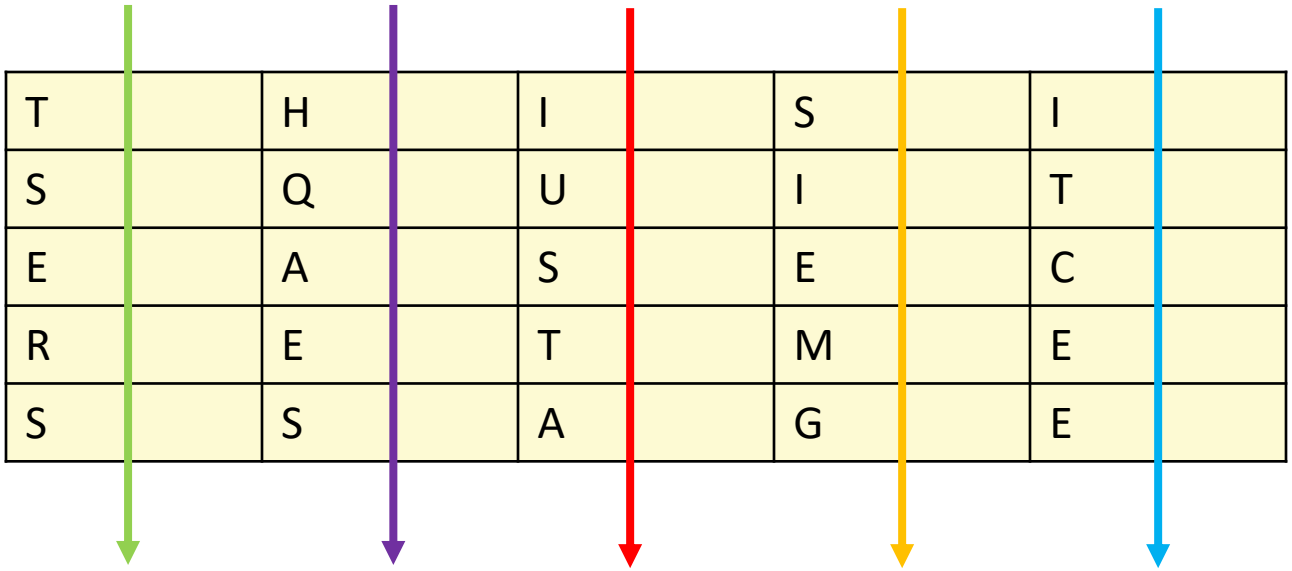
Transposition

- Transposition Ciphers rely on rearranging the order of letter according to some predetermined pattern
- Common method is Columnar Transposition – Write message in a matrix then rearrange columns
- Example:
  - "THIS IS QUITE A SECRET MESSAGE"
  - Represent as 5 X 5 matrix

T	H	I	S	I
S	Q	U	I	T
E	A	S	E	C
R	E	T	M	E
S	S	A	G	E

Columnar Transposition

- "Key" is the order in which columns are read, — choose 3-1-5-4-2
- Ciphertext is now "IUSTATSERSITCEESIEMGHQAES"



T	H	I	S	I
S	Q	U	I	T
E	A	S	E	C
R	E	T	M	E
S	S	A	G	E

Cryptanalysis of Cipher

- "IUSTATSERSITCEESIEMGHQAES" – "Looks" complex, but …
- Observation - letters do not appear equally in English text
- Frequency Analysis

Single Letter Frequency

- Frequency Analysis
- In English

a	8.2%	j	0.2	s	6.3
b	1.5	k	0.8	t	9.1
c	2.8	l	4.0	u	2.8
d	4.3	m	2.4	v	1.0
e	12.7	n	6.7	w	2.4
f	2.2	o	7.5	x	0.2
g	2.0	p	1.9	y	2.0
h	6.1	q	0.1	z	0.1
i	7.0	r	6.0		

- Thus, letters ciphering e, t, and a are easily discovered
- Subsequently can look for the rest of the letters and letter pairs

Diagram Frequency and Trigram Frequency

The most frequent diagrams in English on a relative scale of 1 to 10:

Diagram	Frequency	Diagram	Frequency
TH	10.00	HE	9.05
IN	7.17	ER	6.65
RE	5.92	ON	5.70
AN	5.63	EN	4.76
AT	4.72	ES	4.24
ED	4.12	TE	4.04
TI	4.00	OR	3.98
ST	3.81	AR	3.54
ND	3.52	TO	3.50
NT	3.44	IS	3.43
OF	3.38	IT	3.26
AL	3.15	AS	3.00

The most frequent trigrams in English:

ENT	ION	AND	ING	IVE	TIO
FOR	OUR	THI	ONE		

Substitution Ciphers

- Message symbols are mapped into permuted set of symbols

A =>B	B =>W	C=>E	D=>K	E=>Q
F=>F	G=>M	H=>V	I=>Y	J=>A
K=>L	L=>U	M=>C	N=>O	O=>N
P=>P	Q=>H	R=>S	S=>I	T=>D
U=>X	V=>T	W=>R	X=>G	Y=>Z

Cryptanalysis of Substitutions

- Brute force attacks: try all 26! decipherments - if one decipherment per microsecond, it would take more that 103 years!
- Analyze a large volume of ciphertext for letter frequency
- If frequencies are close to natural English only mapped to different letters, try replacement
- Digram and Trigram frequencies will also be preserved.

Shifted Alphabets

- Cipher formed by shifting letters of the alphabet k positions modulo $|l|$ (the size of the alphabet)
- For English

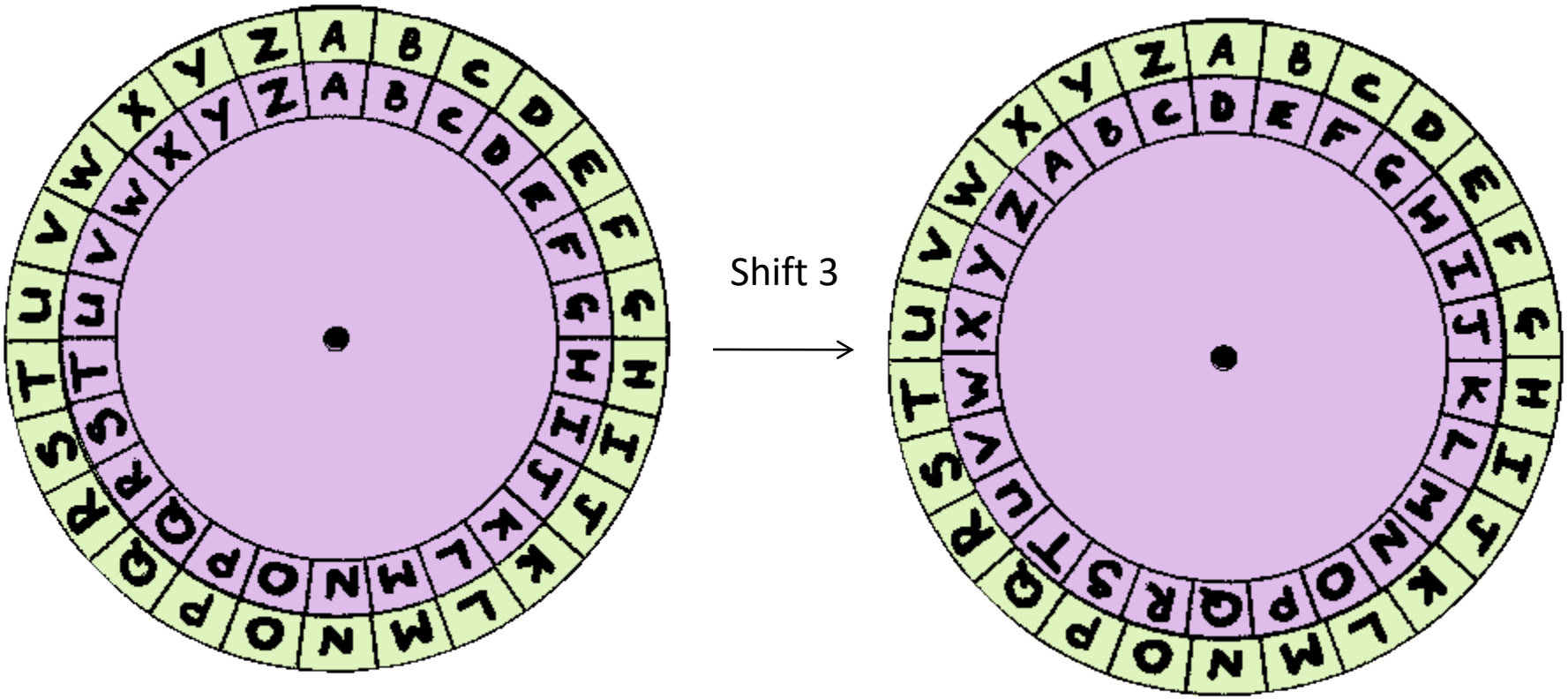
$$f(\blacksquare) = (\blacksquare + k) \, mod \, 26$$

- Multiplicative functions could also be used if k and l are relatively prime

$$f(\blacksquare) = \blacksquare \times k \, mod \, 26$$

Example: Caesar Cipher

- Very simple method is simply to shift alphabets — Caesar Ciphers (shift of 3)



T	H	I	S	I	S	A	C	A	E	S	A	R	C	I	P	H	E	R
W	K	L	V	L	V	D	F	D	H	V	D	U	F	L	S	K	H	U

Other Transforms

- Affine Transforms (Linear plus a constant)

$$f(\blacksquare) = (\blacksquare \cdot k + j) \; mod \; n$$

- Higher Order (define a poly of degree l)

$$f(\blacksquare) = \blacksquare^{l} \cdot k_l + \blacksquare^{l-1} \cdot k_{l-1} + \cdots + \blacksquare \cdot k_1 + k_0 \; mod \; n$$

Cryptanalysis

- All previous methods create a one-to-one mapping of plaintext to ciphertext.
- This is vulnerable to frequency analysis even for relatively small amounts of intercepted ciphertext
- Objective is to "flatten" symbol distribution in ciphertext.

Polyalphabetic Substitutions

- Reduce probability of successful correlation attacks by smearing statistics
- Use multiple substitutions
- $E_K(M) = f_1(m_1), f_2(m_2), \cdots, f_d(m_d), f_{d+1}(m_{d+1}), \cdots$

Vigenere Example

- Vigenere Cipher uses a repeated "Key Word" to perform substitutions

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

T	H	I	S		I	S		A		V	I	G	E	N	E	R	E		C	I	P	H	E	R	Plaintext
I	C	R	I		C	R		I		C	R	I	C	R	I	C	R		I	C	R	I	C	R	Key
B	J	Z	A		K	J		I		X	Z	O	G	E	M	T	V		K	K	G	P	G	I	Ciphertext

Ciphertext = Plaintext + Key mod 26

Coding in Cryptography

Bit, Byte and the ASCII code

- Definition: A bit is either 0 or 1.
- Definition: A byte is a string of 0's and/or 1's with length 8. So one byte equals 8 bits.
- Example:
  - 00000000, 11111111, 11110000, 10101010
- Example:
  - 1001111000011101 is a data of two bytes

Modulo-2 Addition

- Definition: The exclusive-or (XOR) (also called as modulo-2 addition) is defined as follows:

x	y	$x \oplus y$
0	0	0
1	0	1
0	1	1
1	1	0

- Remark: $x \oplus x = 0$ for any $x \in \{0,1\}$
- Remark: if $x \oplus y = z$ then $x = z \oplus y$

Modulo-2 Addition

- Bitwise exclusive-or: Let $x = x_1 x_2 \cdots x_n$ and $y = y_1 y_2 \cdots y_n$. The bitwise exclusive-or of x and y is
$$x \oplus y = (x_1 \oplus y_1)(x_2 \oplus y_2) \cdots (x_n \oplus y_n)$$

- Example:

	1	0	0	1	1
$\oplus$	1	0	1	1	0
	0	0	1	0	1

Encoding Message: ASCII Code

- America Standard Code for Inform. Interchanges
  1. Blank 1,
  2. letters (capital + lower) 26+26=52,
  3. Digits (0,1,…,9) 10,
  4. Symbols 32 [punctuation symbols, accents, brackets, operators]
  5. Controls 33 (non-printable characters)

- Altogether 128 characters. Each is encoded into a string of 7 bits.
  - Example: e= 01100101, "=00100010

- Remark: The ASCII encoding rule is a 1-to-1 function **F** from the set of 128 characters to a subset of the set $\{0,1\}^8$, i.e., consisting of all binary strings of length 8 whose first bit is always 0.
- Remark: There are other ways to encode English messages and data. (Huffman coding)

Binary Representation of Numbers

$$i = i_{t-1} i_{t-2} \cdots i_1 i_0 = i_{t-1} \times 2^{t-1} + i_{t-2} \times 2^{t-2} + \cdots + i_1 \times 2 + i_0$$

where each $i_j \in \{0,1\}$

Example: $1011 = 2^3 + 2^1 + 2^0 = 11$

Decimal Representation of Numbers

$$i = i_{t-1}i_{t-2}\cdots i_1 i_0 = i_{t-1}\times 10^{t-1} + i_{t-2}\times 10^{t-2} + \cdots + i_1 \times 10 + i_0$$

where each $i_j \in \{0,1,2,\cdots,9\}$

Example: $7019 = 7\times 10^3 + 1\times 10^1 + 9\times 10^0 = 7019$

Hexadecimal Representation of Numbers

Define $a=10, b=11, c=12, d=13, e=14, f=15$

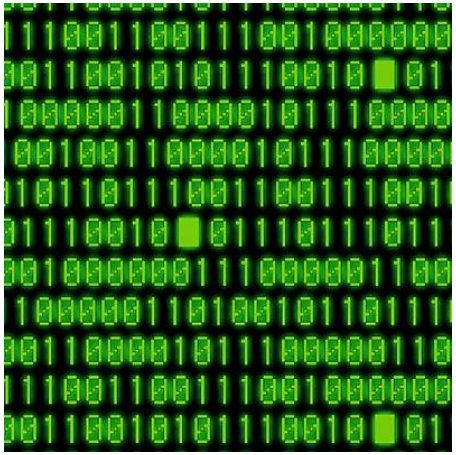
$$i = i_{t-1}i_{t-2}\cdots i_1 i_0 = i_{t-1}\times 16^{t-1} + i_{t-2}\times 16^{t-2} + \cdots + i_1 \times 16 + i_0$$

where each $i_j \in \{0,1,2,\cdots,9,a,b,c,d,e,f\}$

Example: $8ad2 = 8\times 16^3 + a\times 16^2 + d\times 16 + 2\times 16^0 = 35536$

Encoding Message

- The purpose of encoding is not for data confidentiality, but to represent data or a message as binary string for information processing and interchanging.
- We can always encode messages to binary strings for processing.




Thank
you

University of New Brunswick, Faculty of Computer Science

CS4355/6355: Topic 1 – Additional Note

1 Simple Substitution Ciphers

As Julius Caesar surveys the unfolding battle from his hilltop outpost, an exhausted and disheveled courier bursts into his presence and hands him a sheet of parchment containing gibberish:

j s j r d k f q q n s l g f h p g w j f p y m w t z l m n r r n s j s y q z h n z x

Within moments, Julius sends an order for a reserve unit of charioteers to speed around the left flank and exploit a momentary gap in the opponentŠs formation.
How did this string of seemingly random letters convey such important information?

Please use the simple substitution cipher: $ciphertext = plaintext + key \mod 26$ to recover the plaintext of the string and the used key, and explain why.

CS 6355/4355:
Cryptanalysis and Database Security

Topic 2: Symmetric Encryption

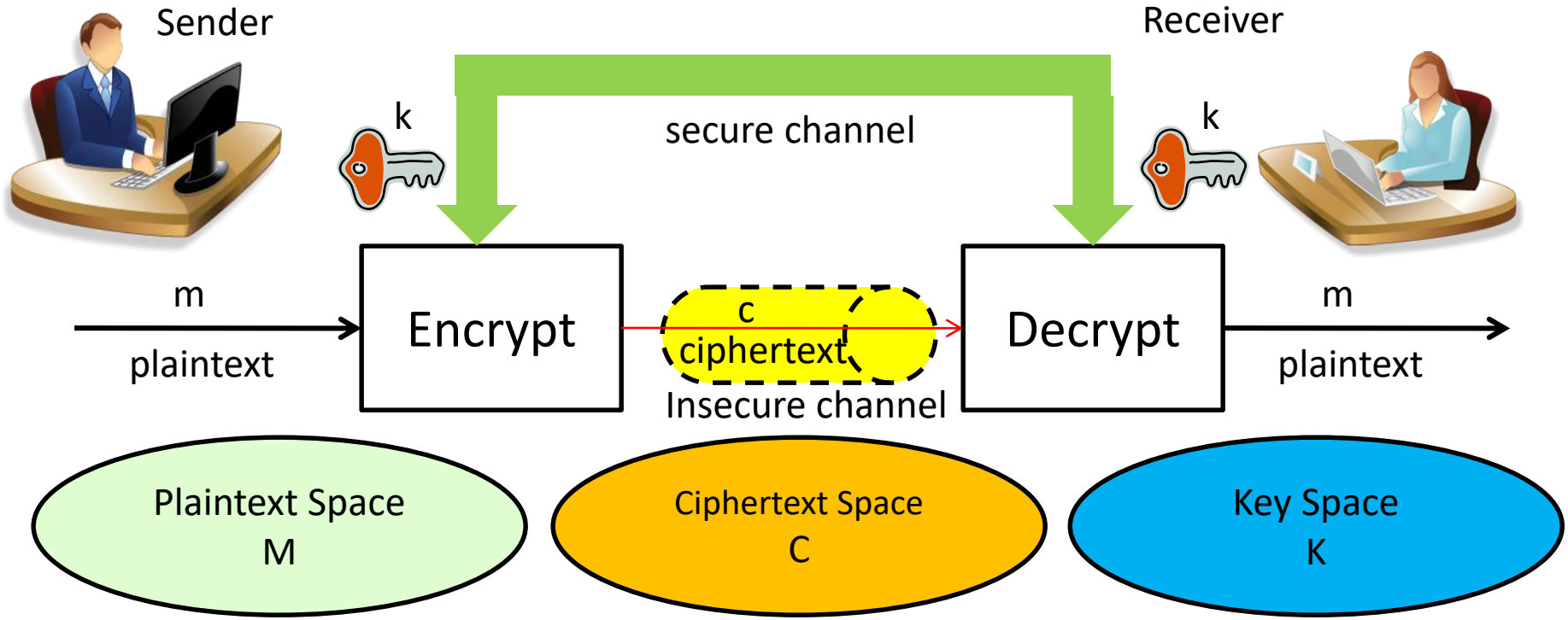
Lecturer: Rongxing LU

Email: RLU1@unb.ca Office: GE 114

Website: http://www.cs.unb.ca/~rlu1/

Faculty of Computer Science, University of New Brunswick

Symmetric Encryption



Symmetric Encryption

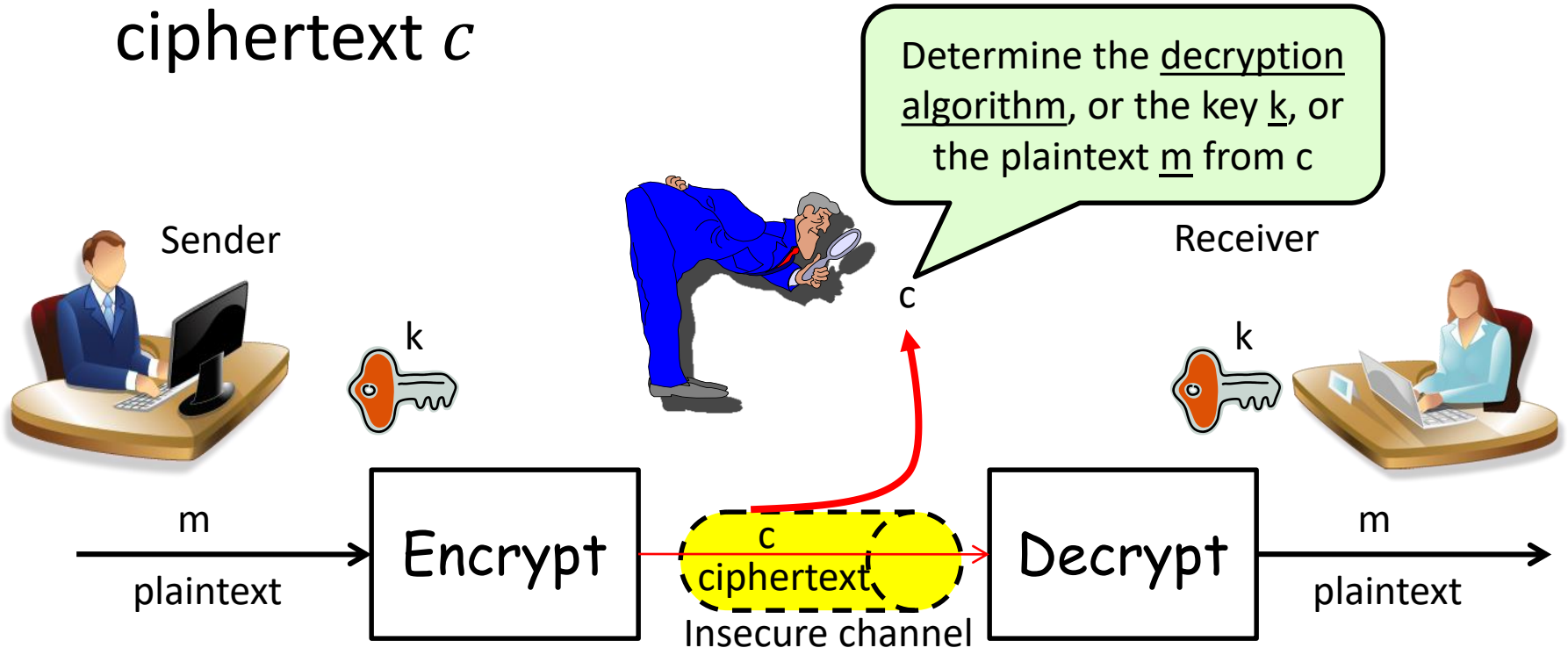
- A 5-tuple (M, C, K, Enc_k, Dec_k), where
  - M, C, K are the message space, ciphertext space, and key space, respectively.
- Any key $k \in K$ could be the encryption and decryption key,
- Encryption: $c = Enc_k(m)$, where Enc_k is usually applied to blocks of or characters of a plaintext $m \in M$
- Decryption: $m = Dec_k\ (c)$, where Dec_k is usually applied to blocks or characters of the ciphertext $c \in C$
- Enc_k, Dec_k are encryption and decryption algorithms, with $Dec_k\big(Enc_k(m)\big) = m$ for each $m \in M$

Brute-Force Attack and Cryptanalysis on Symmetric Encryption

- Brute-force attack
  - Attacker tries every possible key on a piece of ciphertext until an intelligible translation into plaintext is obtained
  - On average, half of all possible keys must be tried to achieve success
  - To supplement the brute-force approach, some degree of knowledge about the expected plaintext is needed, and some means of automatically distinguishing plaintext from garble is also needed
- Cryptanalysis
  - Attack relies on the nature of the algorithm plus some knowledge of the general characteristics of the plaintext
  - Attack exploits the characteristics of the algorithm to attempt to deduce a specific plaintext or to deduce the key being used

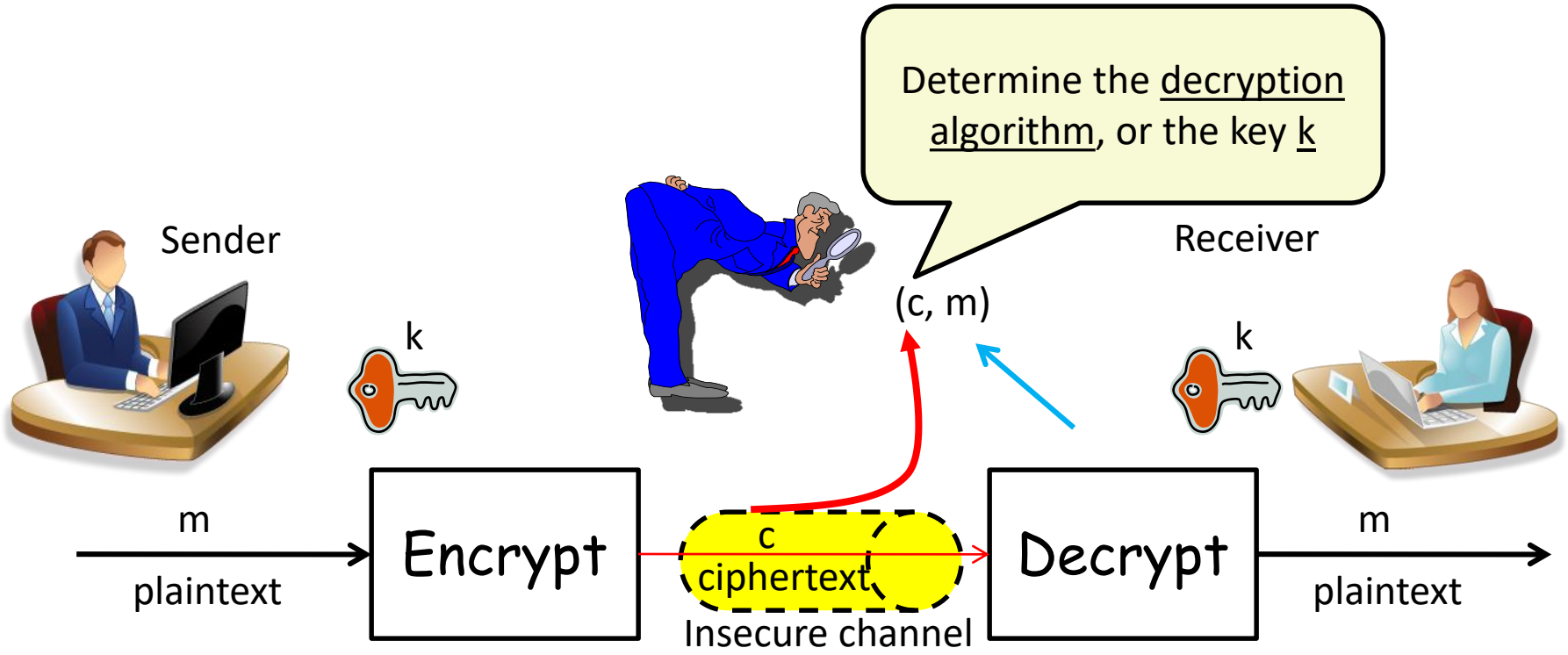
Attacks on Symmetric Encryption

- **Ciphertext-only attack:** An adversary determines the decryption algorithm Dec_k or key k, or the plaintext from intercepted ciphertext c



Attacks on Symmetric Encryption (2)

- **Known-plaintext attack:** An adversary determines the decryption algorithm Dec_k or key k, from a ciphertext-plaintext (c, m)

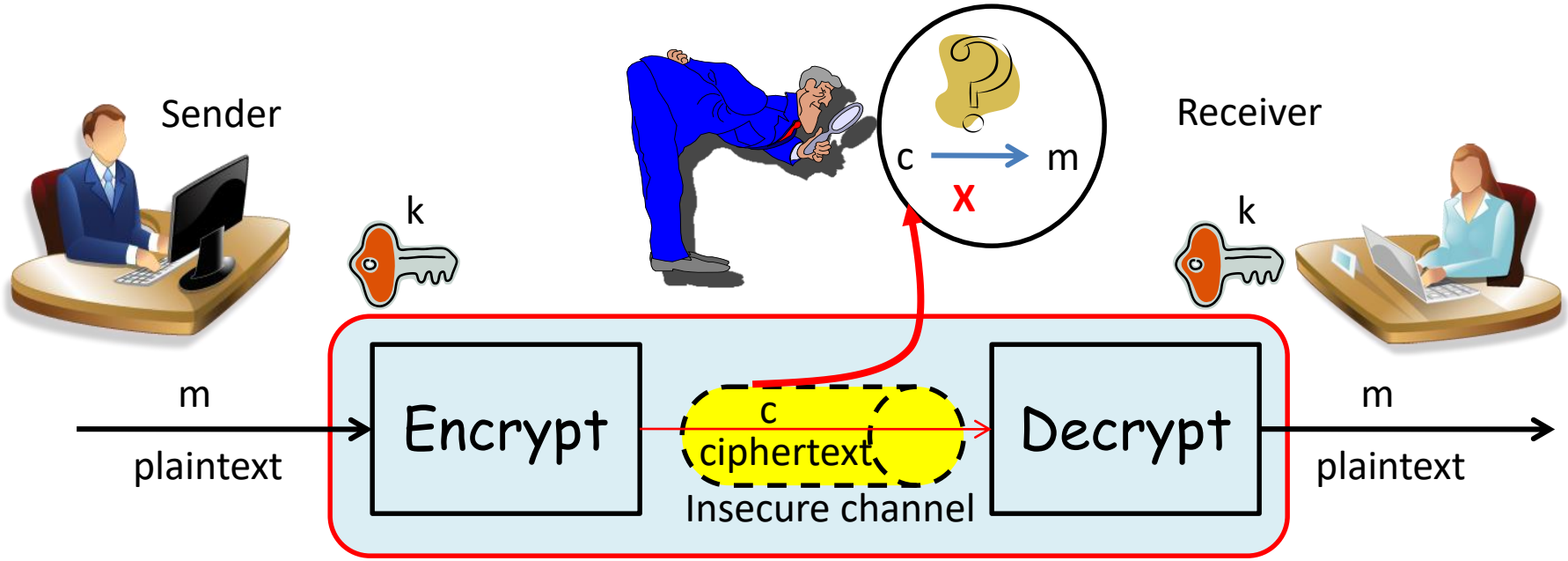


More Attacks on Symmetric Encryption

Type	Known to the Cryptanalyst
Ciphertext Only	• Encryption algorithm; Ciphertext
Known Plaintext	• Encryption algorithm; Ciphertext • One or more plaintext-ciphertext pairs formed with the secret key
Chosen Plaintext	• Encryption algorithm; Ciphertext **Differential cryptanalysis** • Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key
Chosen Ciphertext	• Encryption algorithm; Ciphertext • Ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret
Chosen Text	• Encryption algorithm; Ciphertext • Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key • Ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret

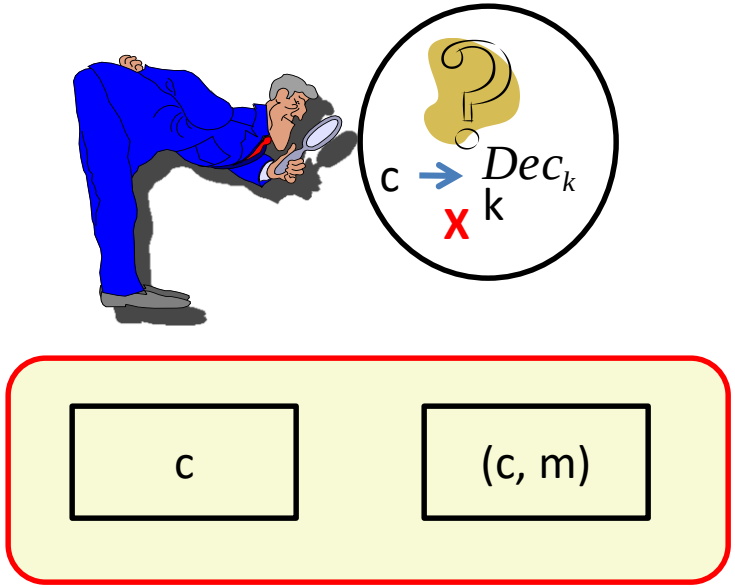
Security Requirements

- According to Kerckhoffs Principles, the security should depend on the confidentiality of the key, so it is usually assumed that the algorithms Enc_k and Dec_k are known to an adversary.
- It should be computationally infeasible for an adversary to determine the plainext $m \in M$, given a ciphertext $c \in C$.



Security Requirements (2)

- It should be computationally infeasible for an adversary to systematically determine the decryption algorithm Dec_k or key k from intercepted ciphertext c, even if the corresponding plaintext m is known.



Properties of Symmetric Encryption

- How to design a symmetric cryptosystem to meet the above requirement?
- Two properties are desirable
  - Confusion
  - Diffusion

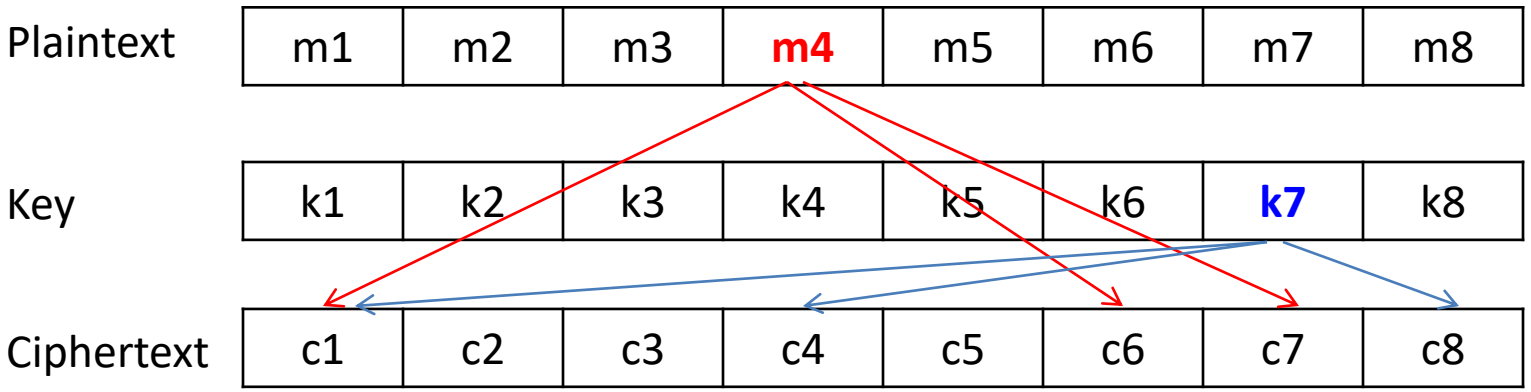
Confusion

- **Confusion:**
  - Process of substituting characters or symbols to make the relationship between ciphertext and key as complex as possible.
  - Attackers' uncertainty as to the contents of a message or the key used for encryption and decryption.

Plaintext	1	1	0	1	1	0	1	0
Key	0	1	0	1	1	1	0	1
Cipertext = Plain $\oplus$ Key	1	0	0	0	0	1	1	1

Diffusion

- **Diffusion:**
  - Process of spreading effect of plaintext or key as widely as possible over ciphertext.
  - Dispersion of the effect of individual key or message bits over the plaintext



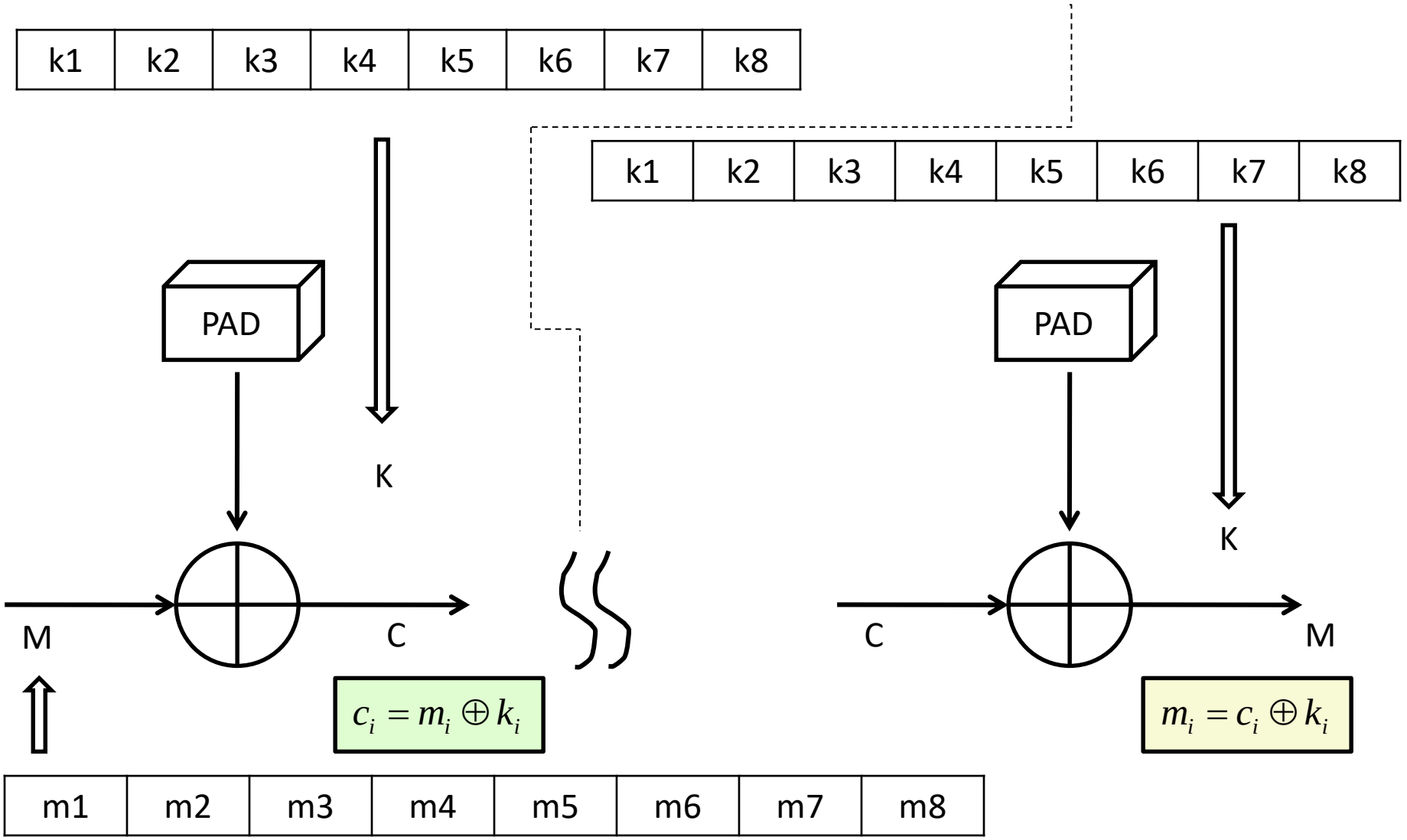
Classification of Secure Ciphers

- Unconditionally secure
  - No matter how much time an opponent has, it is impossible for him or her to decrypt the ciphertext simply because the required information is not there
  - Cannot be broken regardless of attackers computational abilities
- Computationally secure
  - The cost of breaking the cipher exceeds the value of the encrypted information
  - The time required to break the cipher exceeds the useful lifetime of the information
  - Secure against attacker with “reasonable” resources

Unconditionally secure

- Only system known to be provably unconditionally secure are “one-time pads”
- Each message is encoded into a binary string using the ASCII code;
- The secret key is a random binary string with the same length as the message;
- A secret key is used only for one message and is then discarded.

One-Time Pad


k1 k2 k3 k4 k5 k6 k7 k8
k1 k2 k3 k4 k5 k6 k7 k8
PAD
K
PAD
K
M
C
C
M
$c_i = m_i \oplus k_i$
$m_i = c_i \oplus k_i$
m1 m2 m3 m4 m5 m6 m7 m8

Why One-Time Pad is provably secure?

- The security depends on the randomness of the key, but it is hard to define randomness.
- In cryptographic context, we seek two fundamental properties in a binary random key sequence:
  - **Unpredictability:** Independent of the number of the bits of a sequence observed, the probability of guessing the next bit is not better than 1/2. Therefore, the probability of a certain bit being 1 or 0 is exactly equal to 1/2.
  - **Balanced** (Equal Distribution): The number of 1 and 0 should be equal.

Mathematical Proof

- the probability of a key bit being 1 or 0 is exactly equal to ½
- The plaintext bits are not balanced. Let the probability of 0 be x and then the probability of 1 turns out to be $1 - x$.
- We can calculate the probability of ciphertext bits.

m_i	Prob. m	k_i	Prob. k	c_i	Prob. c
0	x	0	$1/2$	0	$x/2$
0	x	1	$1/2$	1	$x/2$
1	$1-x$	0	$1/2$	1	$(1-x)/2$
1	$1-x$	1	$1/2$	0	$(1-x)/2$

- We find out the probability of a ciphertext bit being 1 or 0 is equal to $\frac{1}{2} \cdot x + \frac{1}{2} \cdot (1-x) = \frac{1}{2}$, and the ciphertext looks like a random sequence.

Computationally Secure

- "Useful" Cryptosystem
- Observe the message have only limited "secret" lifetime;
- Balance between size of cipher (speed and efficiency) and strength;
- Evolution driven by improvements in computational power;
- An encryption scheme is said to be computationally secure if:
  - The cost of breaking the cipher exceeds the value of the encrypted information or
  - The time required to break the cipher exceeds the useful lifetime of the information.

Average time required for exhaustive key search

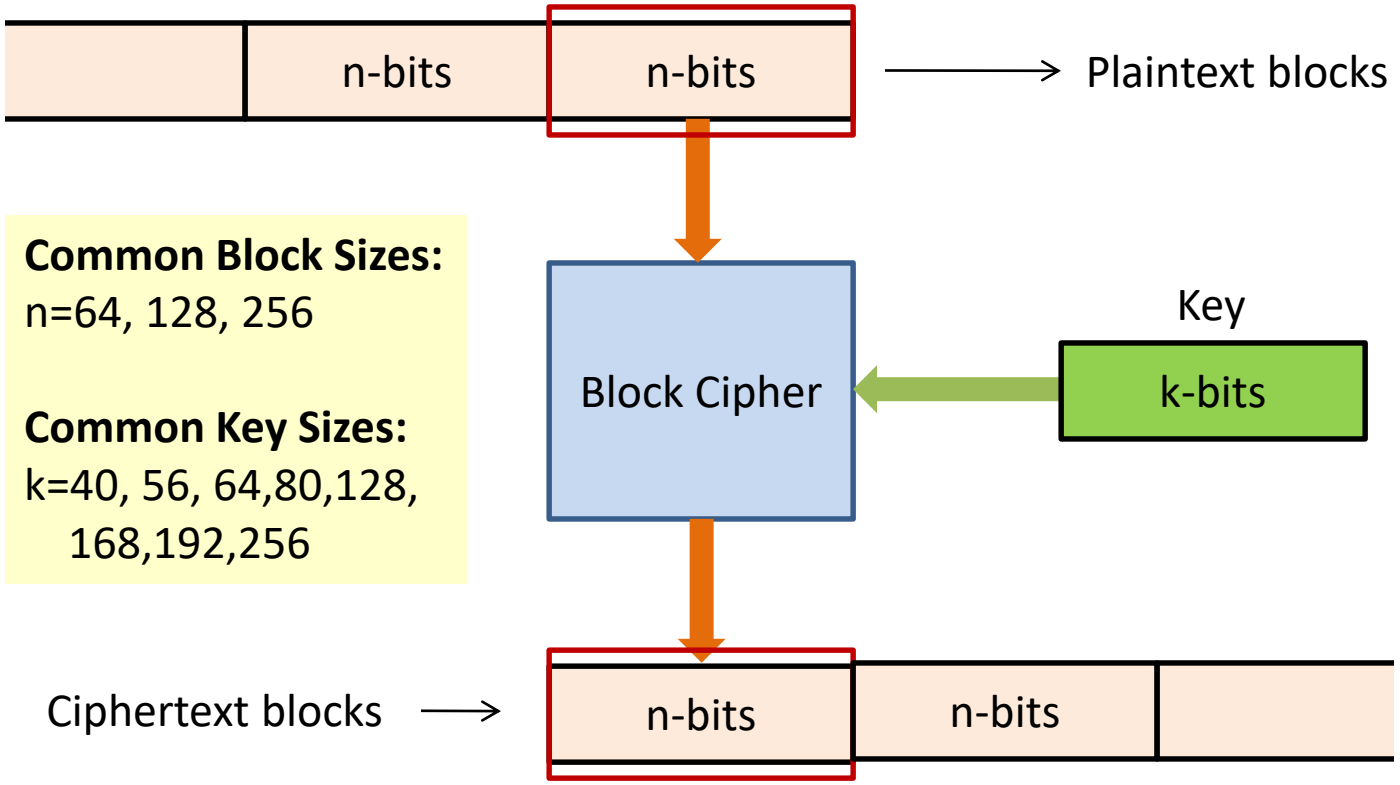
Key Size (bits)	Number of Alternatives Keys	Time required at 10^6 Decryption / μs
32	$2^{32} = 4.3 \times 10^9$	2.15 milliseconds
56	$2^{56} = 7.2 \times 10^{16}$	10 hours
128	$2^{128} = 3.4 \times 10^{38}$	5.4×10^{18} years
168	$2^{168} = 3.7 \times 10^{50}$	5.9×10^{30} years

Type of Ciphers

- Two basic formats:
  - Block ciphers
    - Block ciphers break messages into fixed length blocks, and encrypt each block using the same key.
    - The Data Encryption Standard (DES) is an example of a block cipher, where blocks of 64 bits are encrypted using a 56-bit key.
  - Stream ciphers
    - Stream ciphers, like block ciphers, break message into fixed length blocks, but use a sequence of keys to encrypt the blocks.
    - The Vigenere cipher is an example of a stream cipher.
      - Key = $k_1 k_2 k_3 k_4$ (random, used one-time only)
      - Plaintext = $m_1 m_2 m_3 m_4$; Ciphertext = $c_1 c_2 c_3 c_4$, where $c_i = m_i \oplus k_i$

Block Ciphers

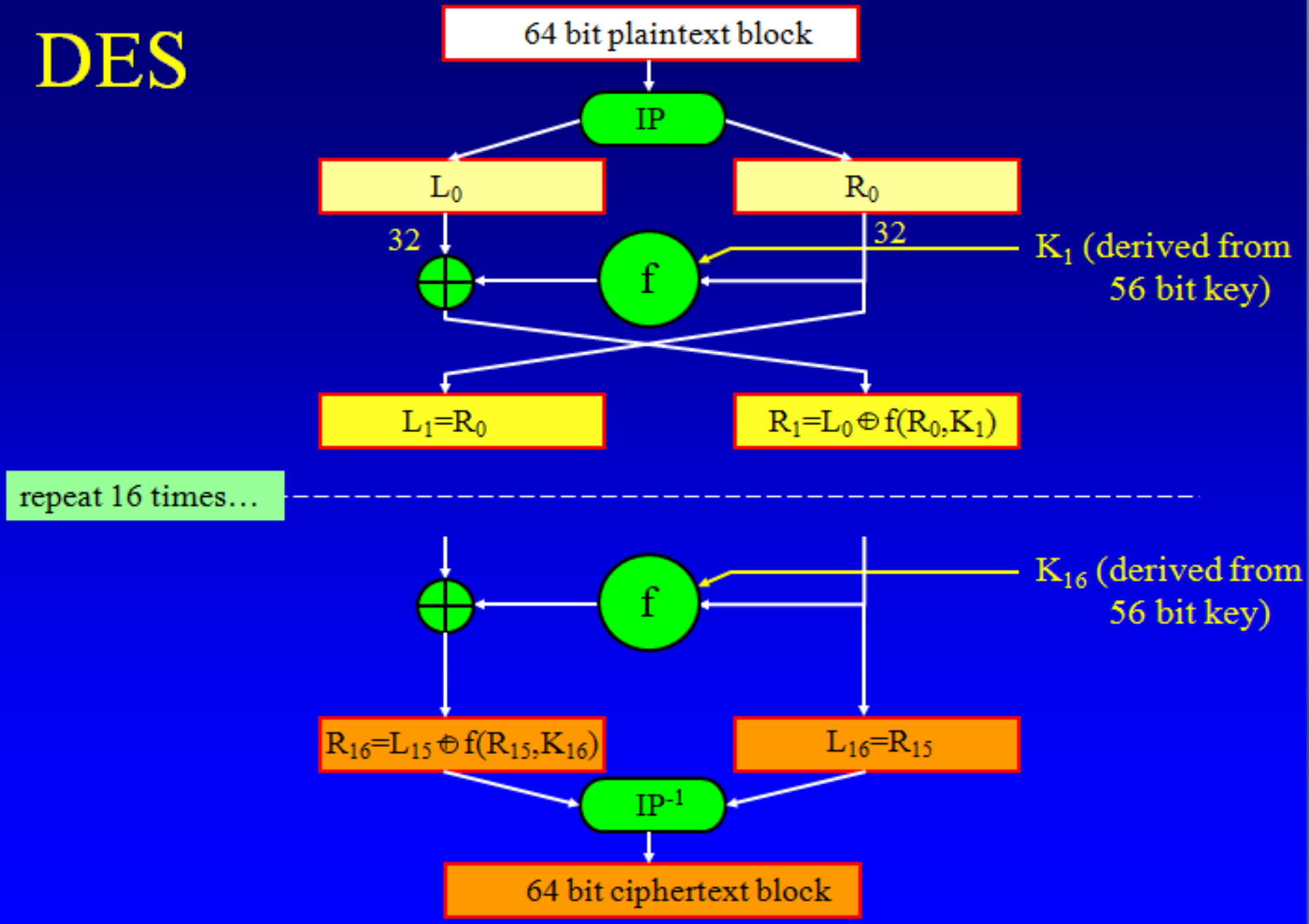
- Message is divided into fixed size blocks (block size) using padding if necessary
- Ciphertext is block of (usually) the same size



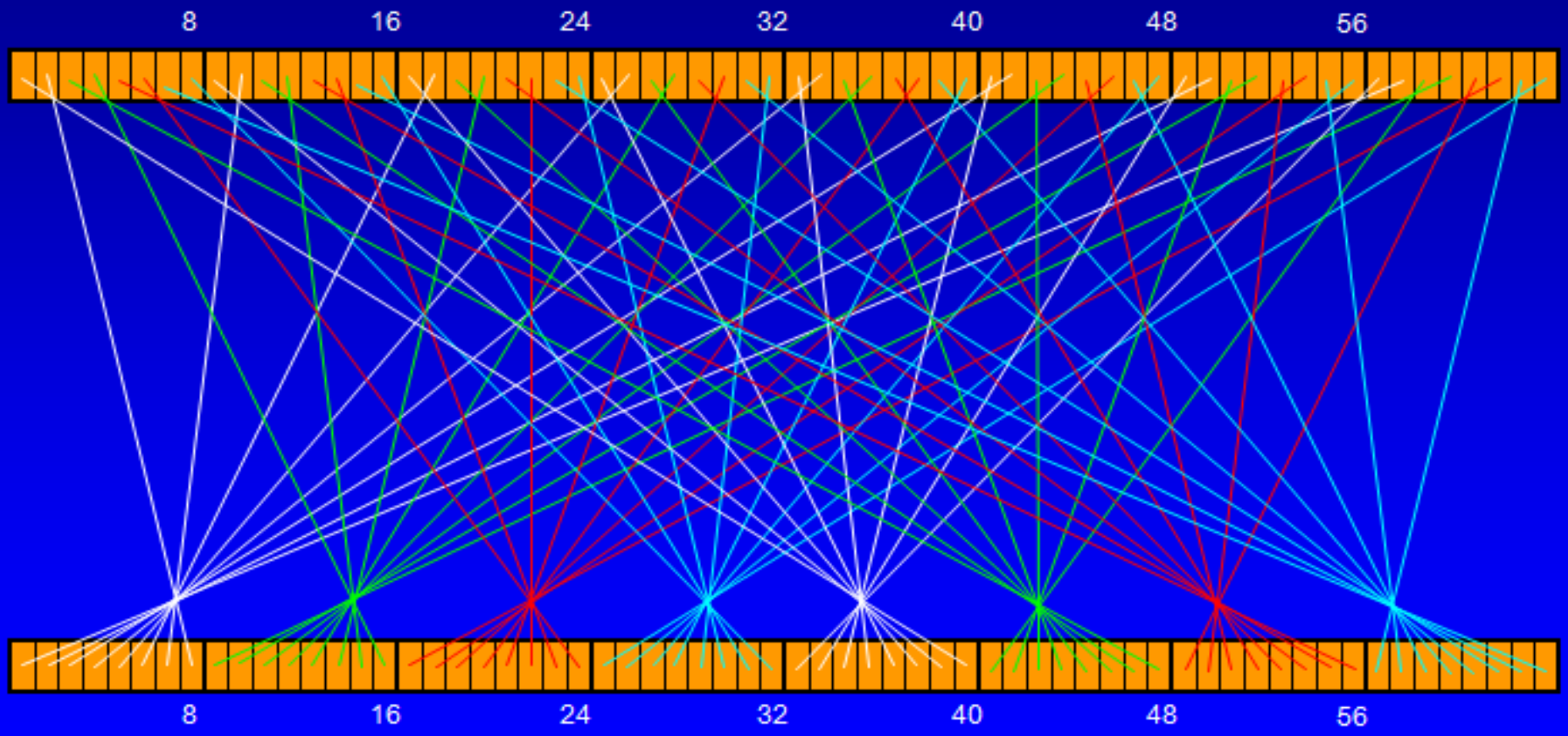
Block Ciphers (2)

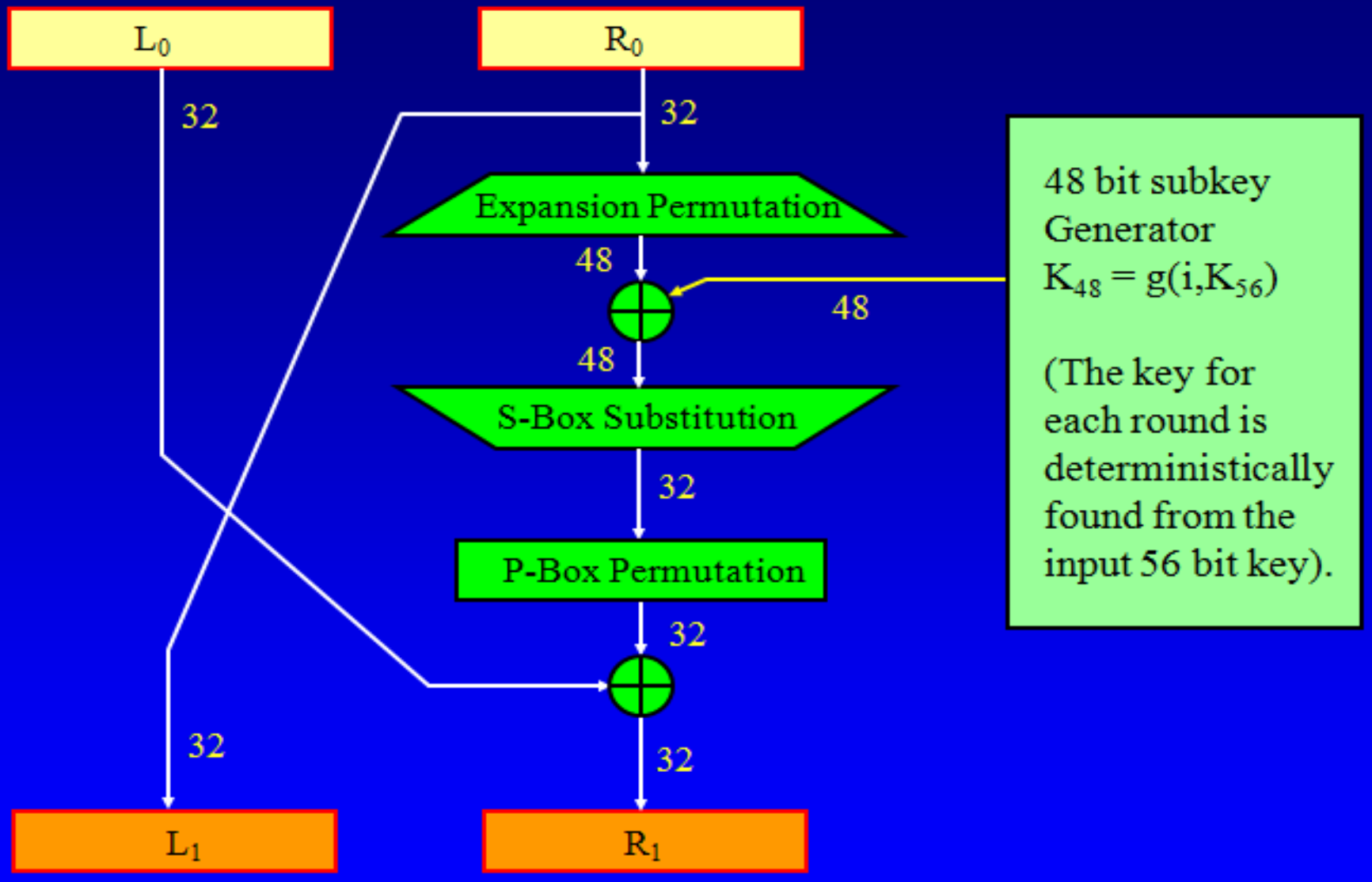
- Formal definition of Block Cipher
  - Let E be an encryption algorithm, and let $E_k(b)$ be the ciphertext of the message b with key k.
  - Let a message $m = b_1 b_2 \cdots$ where each b_i is of a fixed length.
  - A block cipher is a cipher for which $E_k(m) = E_k(b_1)E_k(b_2)\cdots$
- Properties of Block Ciphers
  - Adds Confusion about message and key
  - Should have good Diffusion Properties
    - Single bit change in input plaintext should produce changes in aprox. 50% of output bits (at random)
    - Single bit change in key should produce changes in approx. 50% of output bits (at random)
- Example: DES encrypts 64-bit blocks
  - Key size 56 bits plus 8 parity bits

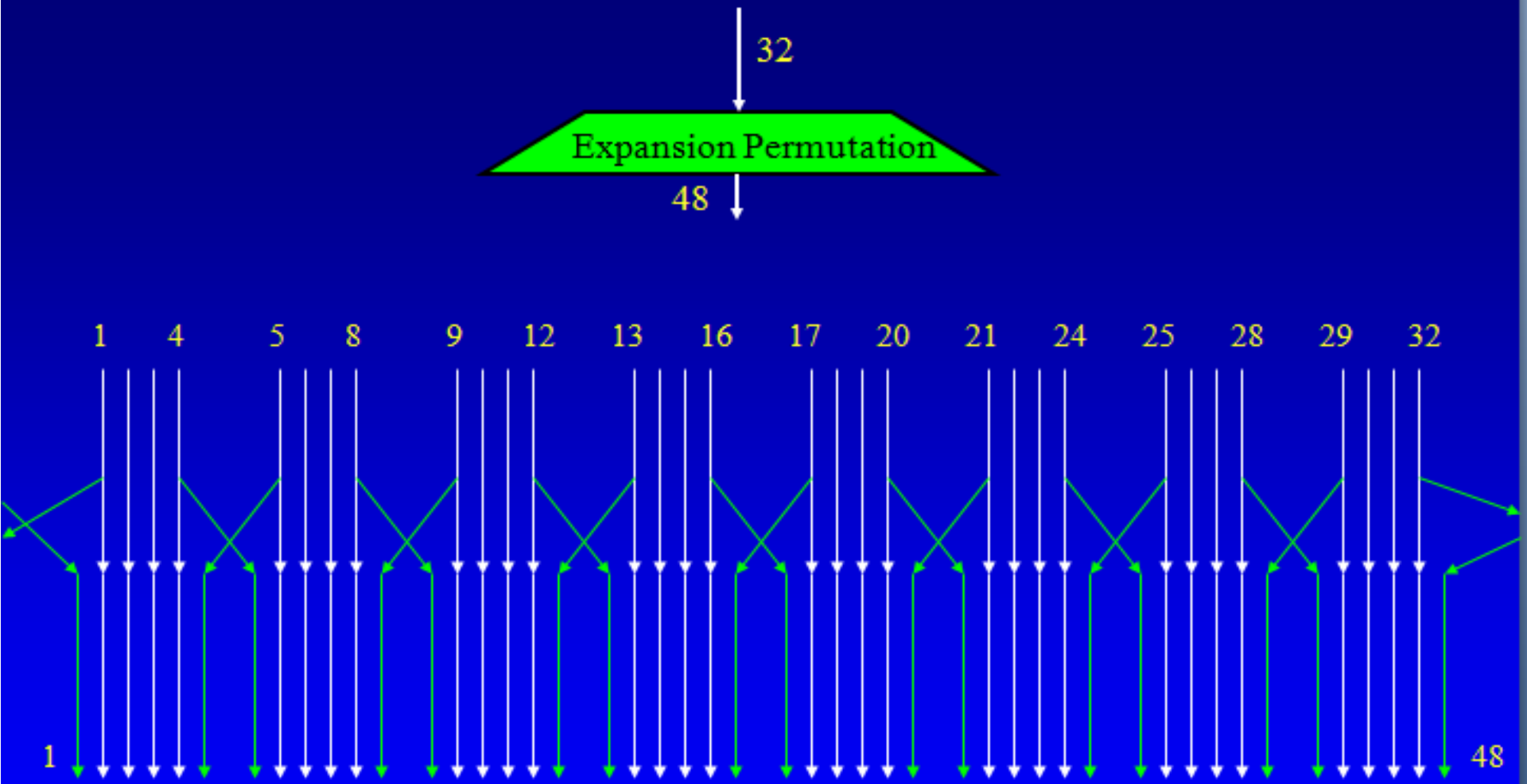
DES

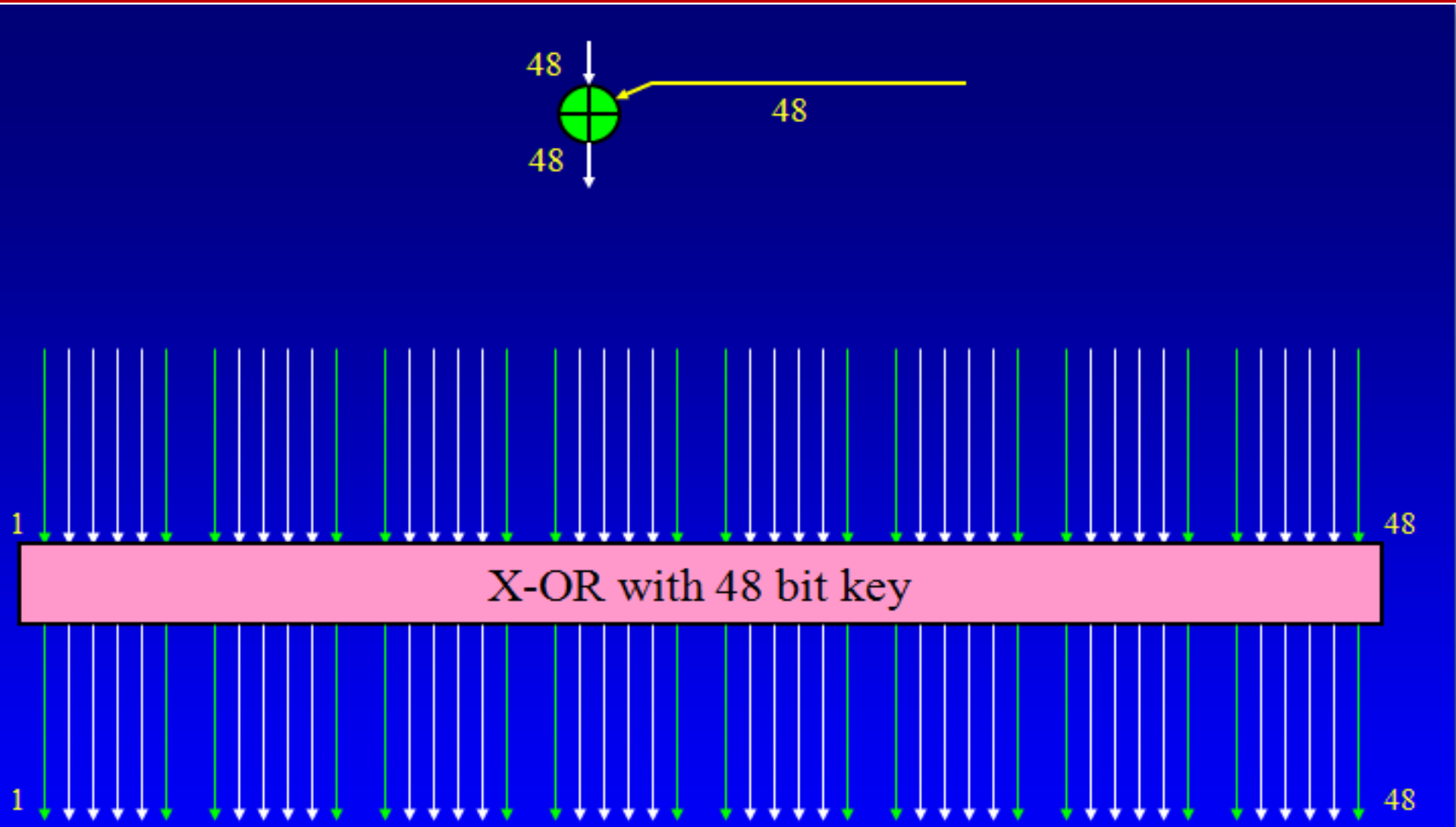

64 bit plaintext block
IP
L_0
R_0
32
32
f
K_1 (derived from 56 bit key)
$L_1=R_0$
$R_1=L_0 \oplus f(R_0,K_1)$
repeat 16 times…
f
K_{16} (derived from 56 bit key)
$R_{16}=L_{15} \oplus f(R_{15},K_{16})$
$L_{16}=R_{15}$
IP^{-1}
64 bit ciphertext block

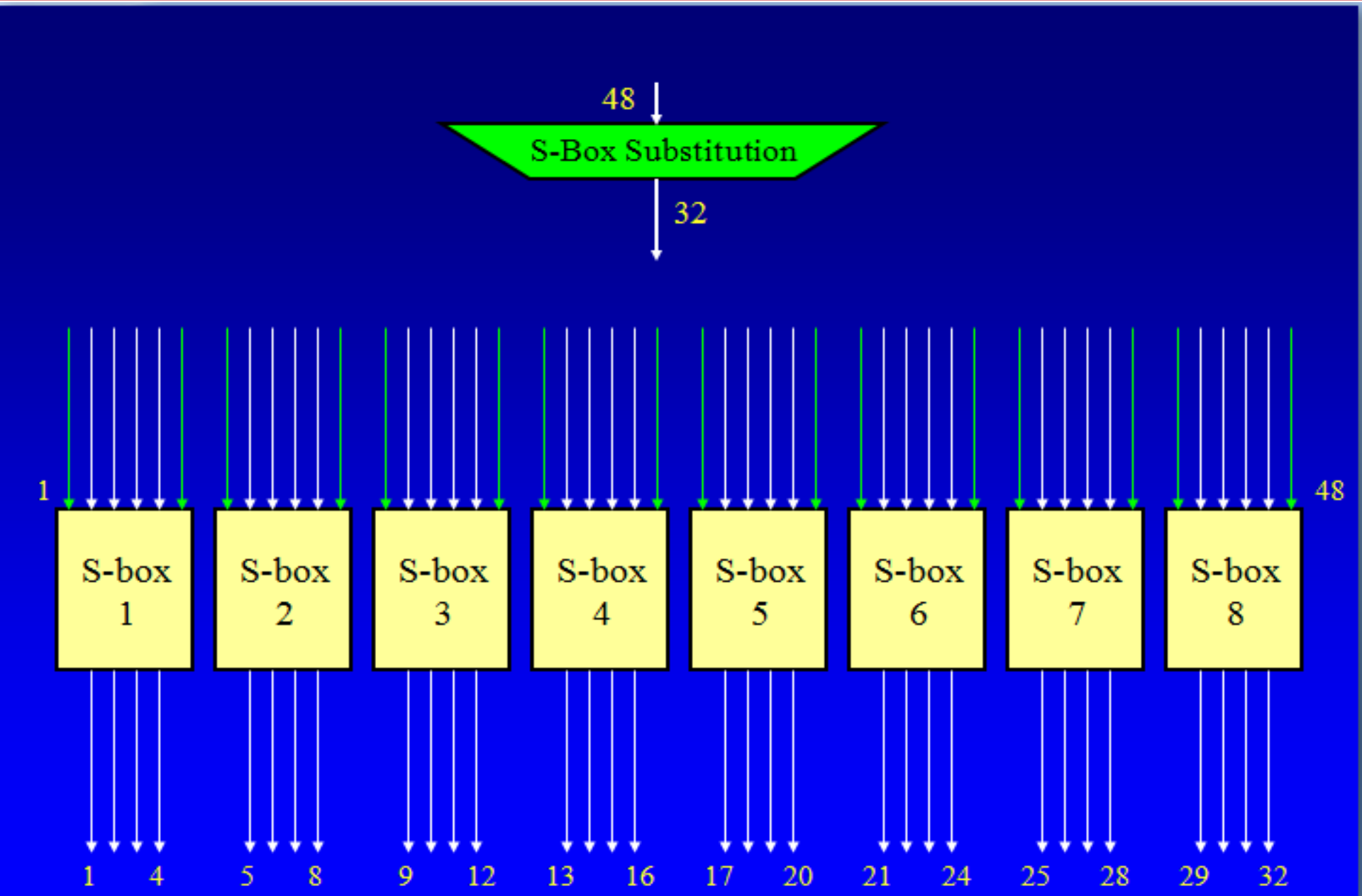
IP (Initial Permutation):



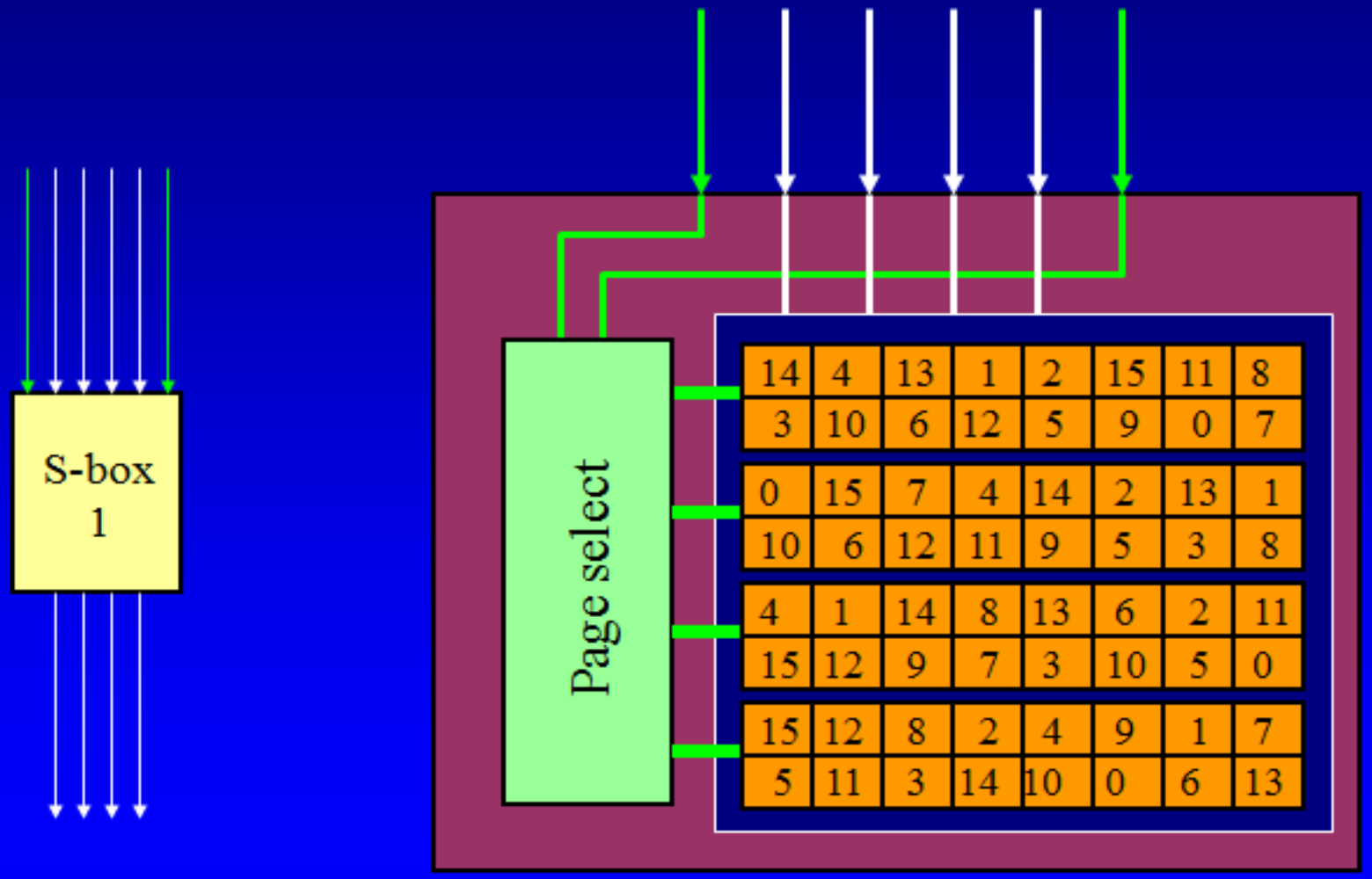

L_0
R_0
32
32
Expansion Permutation
48
48
48
S-Box Substitution
32
P-Box Permutation
32
32
32
L_1
R_1
48 bit subkey Generator
$K_{48} = g(i,K_{56})$
(The key for each round is deterministically found from the input 56 bit key).


32
Expansion Permutation
48
1 4 5 8 9 12 13 16 17 20 21 24 25 28 29 32
1
48

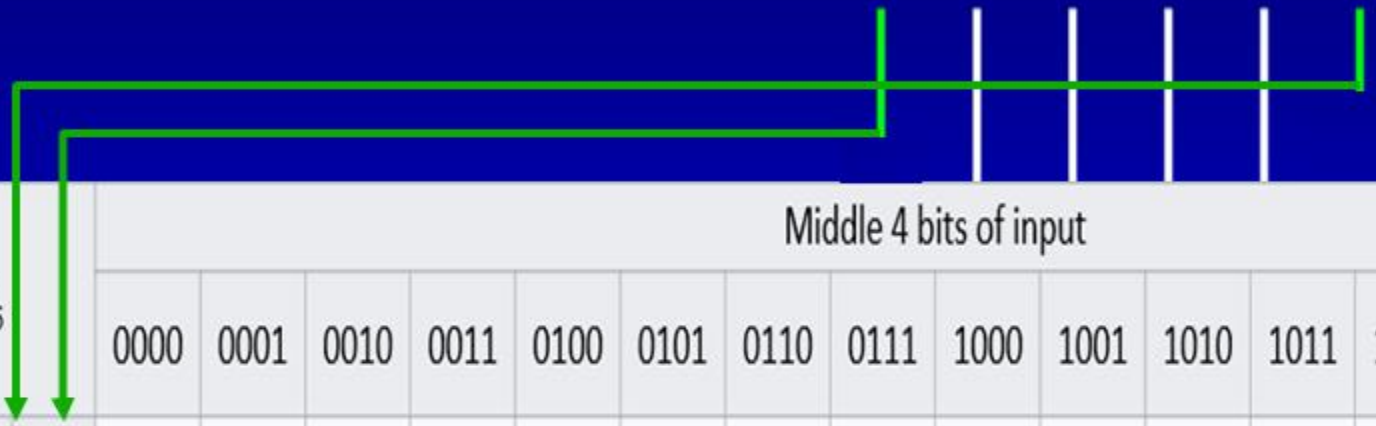

48
48
48
1
48
X-OR with 48 bit key
1
48


48
S-Box Substitution
32
1
48
S-box 1
S-box 2
S-box 3
S-box 4
S-box 5
S-box 6
S-box 7
S-box 8
1 4
5 8
9 12
13 16
17 20
21 24
25 28
29 32

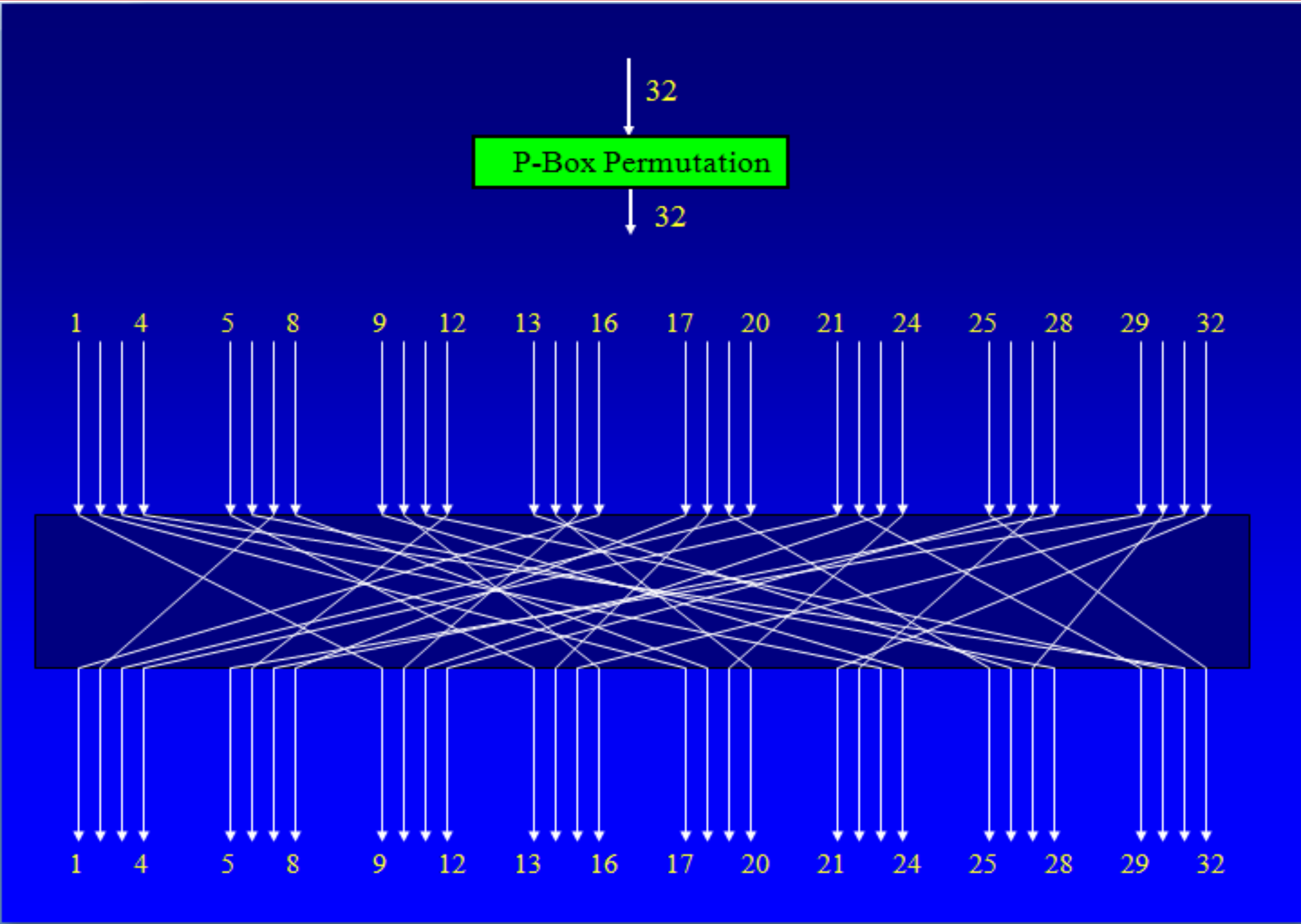
How an S-Box works



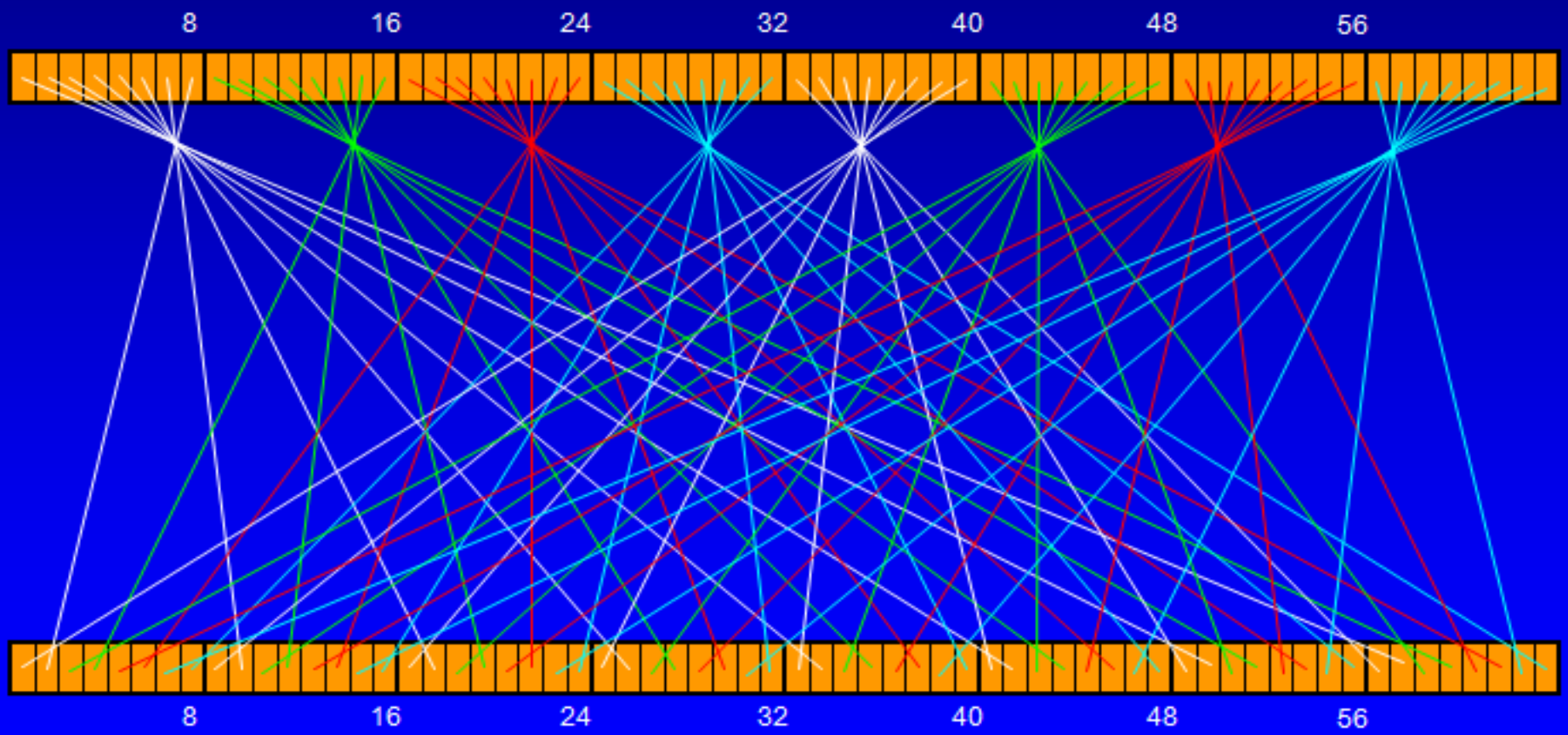
How an S-Box works



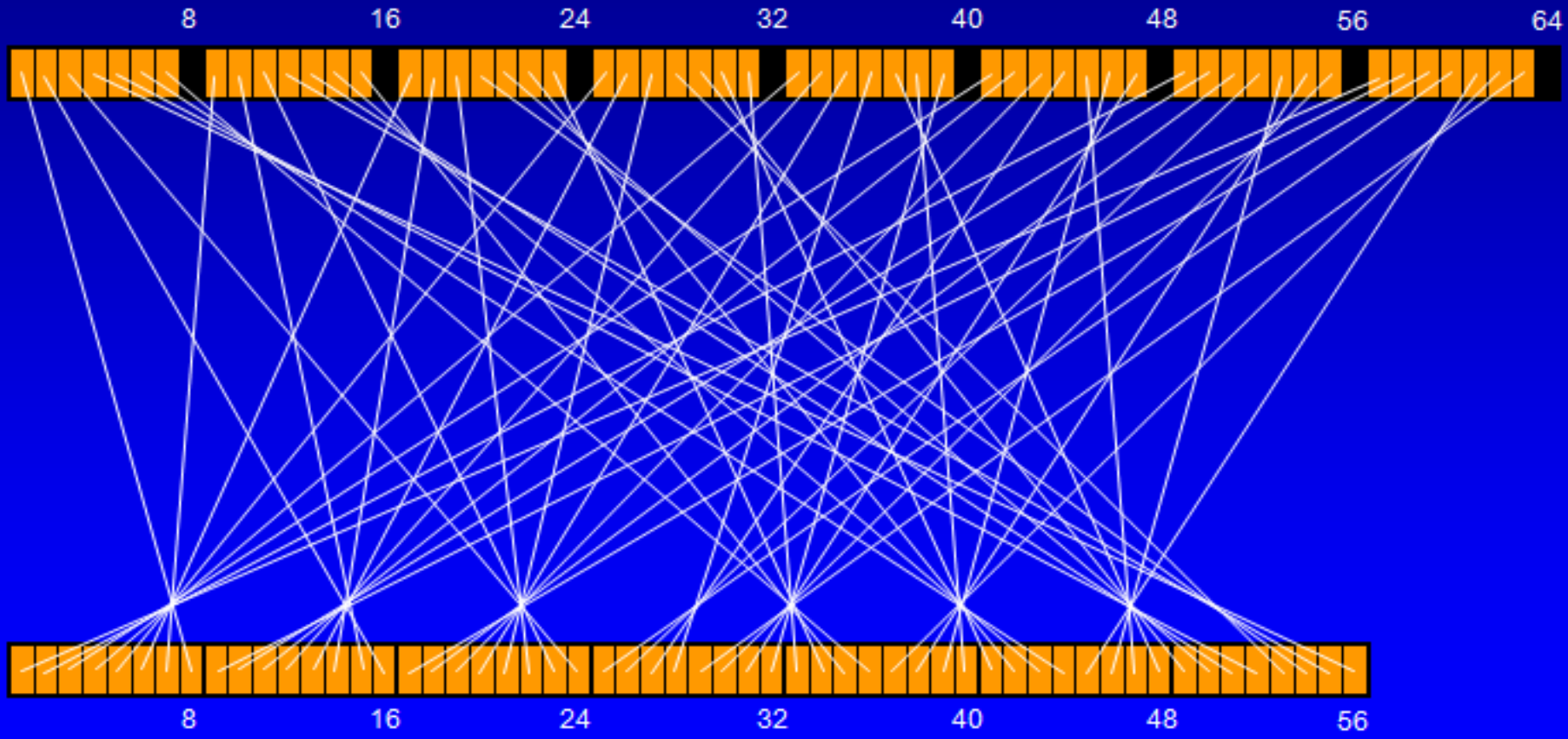
S_5	Middle 4 bits of input															
	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
Outer bits 00	0010	1100	0100	0001	0111	1010	1011	0110	1000	0101	0011	1111	1101	0000	1110	1001
Outer bits 01	1110	1011	0010	1100	0100	0111	1101	0001	0101	0000	1111	1010	0011	1001	1000	0110
Outer bits 10	0100	0010	0001	1011	1010	1101	0111	1000	1111	1001	1100	0101	0110	0011	0000	1110
Outer bits 11	1011	1000	1100	0111	0001	1110	0010	1101	0110	1111	0000	1001	1010	0100	0101	0011


32
P-Box Permutation
32
1 4 5 8 9 12 13 16 17 20 21 24 25 28 29 32
1 4 5 8 9 12 13 16 17 20 21 24 25 28 29 32

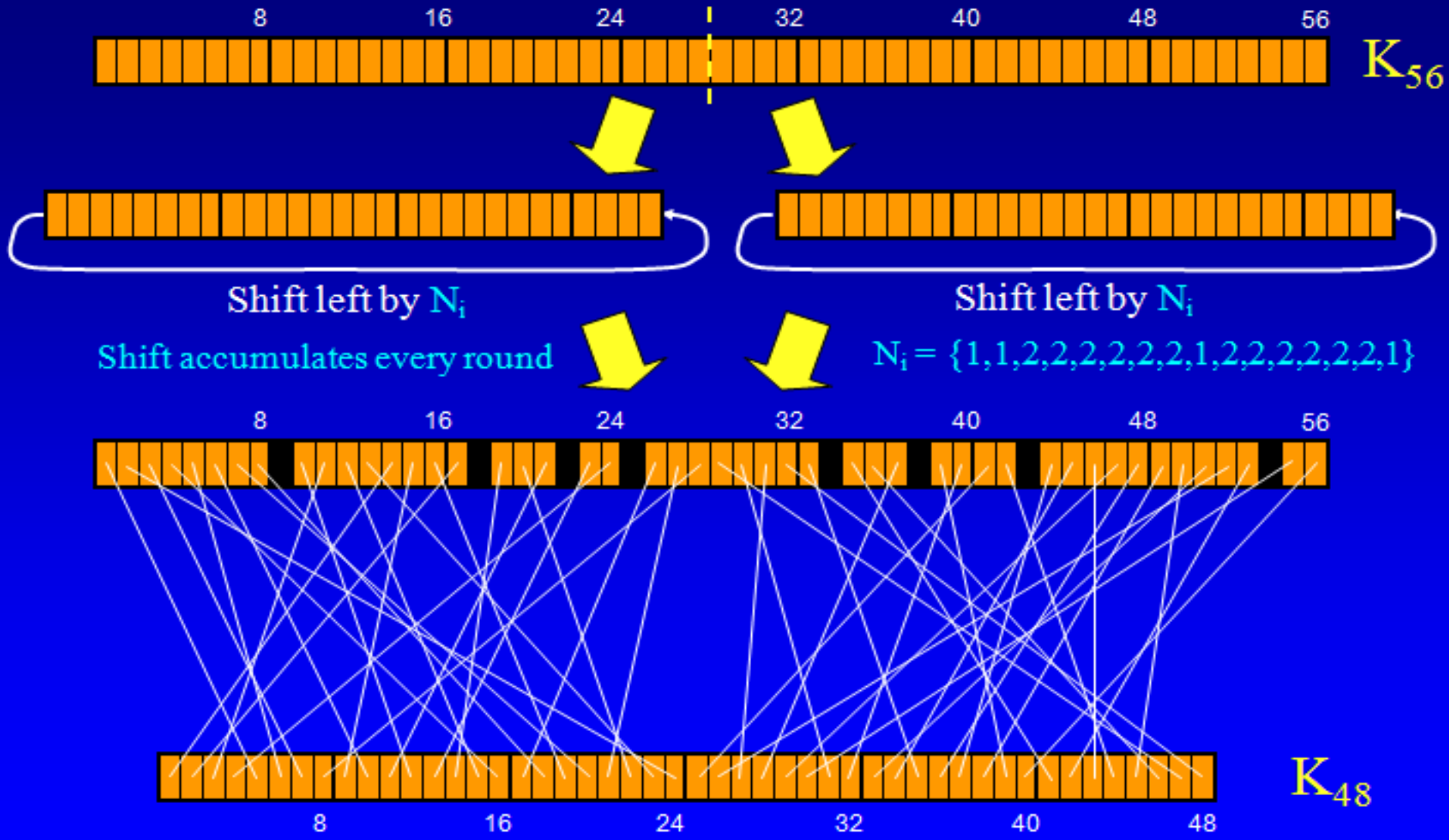
IP^{-1} (Final Permutation):



Initial Key Permutation



Key Split & Shift & Compress



Avalanche Effect

- key desirable property of encryption algorithm
  - where a change of one input or key bit results in changing approx half output bits
  - making attempts to "home-in" by guessing keys impossible
- DES exhibits strong avalanche

Strength of DES – Key Size

- 56-bit keys have $2^{56} = 7.2 \times 10^{16}$ values
- brute force search looked hard
- advances have shown is possible
  - in 1997 on Internet in a few months
  - in 1998 on dedicated h/w (EFF) in a few days
  - in 1999 above combined in 22hrs!
  - still must be able to recognize plaintext
    - Otherwise, have no idea on the correct key
- Forced to consider alternatives to DES

Strength of DES – Analytic Attacks

- Currently, we have several analytic attacks on DES
  - utilize some deep structure of the cipher by gathering information about encryptions
  - can eventually recover some/all of the sub-key bits
  - if necessary then exhaustively search for the rest
- generally these are statistical attacks
  - differential cryptanalysis
  - linear cryptanalysis
  - related key attacks

Strength of DES – Timing Attacks

- attacks actual implementation of cipher
- use knowledge of consequences of implementation to derive information about some/all subkey bits
- specifically use fact that calculations can take varying times depending on the value of the inputs to it
- particularly problematic on smartcards

Differential Cryptanalysis

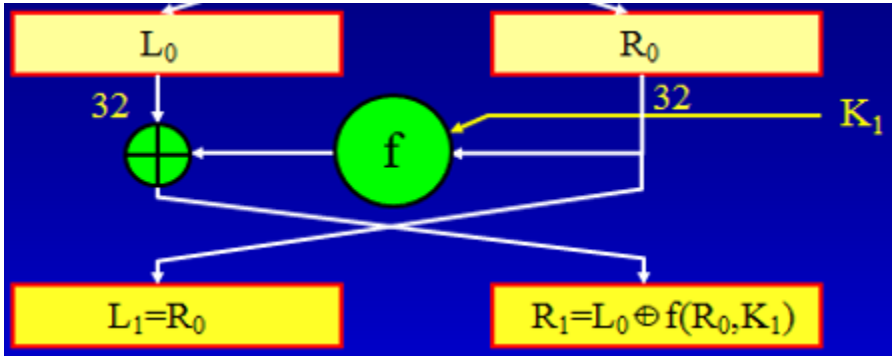
- one of the most significant recent (public) advances in cryptanalysis
- known by NSA in 70's cf DES design
- Murphy, Biham & Shamir published in 90's
- powerful method to analyse block ciphers
- used to analyze most current block ciphers with varying degrees of success
- DES reasonably resistant to it, cf Lucifer

Differential Cryptanalysis (2)

- a statistical attack against Feistel ciphers
- uses cipher structure not previously used
- design of S-P networks has output of function f influenced by both input & key
- hence cannot trace values back through cipher without knowing value of the key
- differential cryptanalysis compares two related pairs of encryptions (differential)

Differential Cryptanalysis (3)

- Differential cryptanalysis compares two related pairs of encryptions
- with known difference in the input m0||m1
- searching for a known difference in output
- when same subkeys are used



$$\Delta m_{i+1} = m_{i+1} \oplus m'_{i+1}$$
$$= [m_{i-1} \oplus f(m_i, K_i)] \oplus [m'_{i-1} \oplus f(m'_i, K_i)]$$
$$= \Delta m_{i+1} \oplus [f(m_i, K_i) \oplus f(m'_i, K_i)]$$

$$\Delta R_1 = R_1 \oplus R'_1$$
$$= [L_0 \oplus f(R_0, K_1)] \oplus [L'_0 \oplus f(R'_0, K_1)]$$
$$= \Delta L_0 \oplus [f(R_0, K_1) \oplus f(R'_0, K_1)]$$

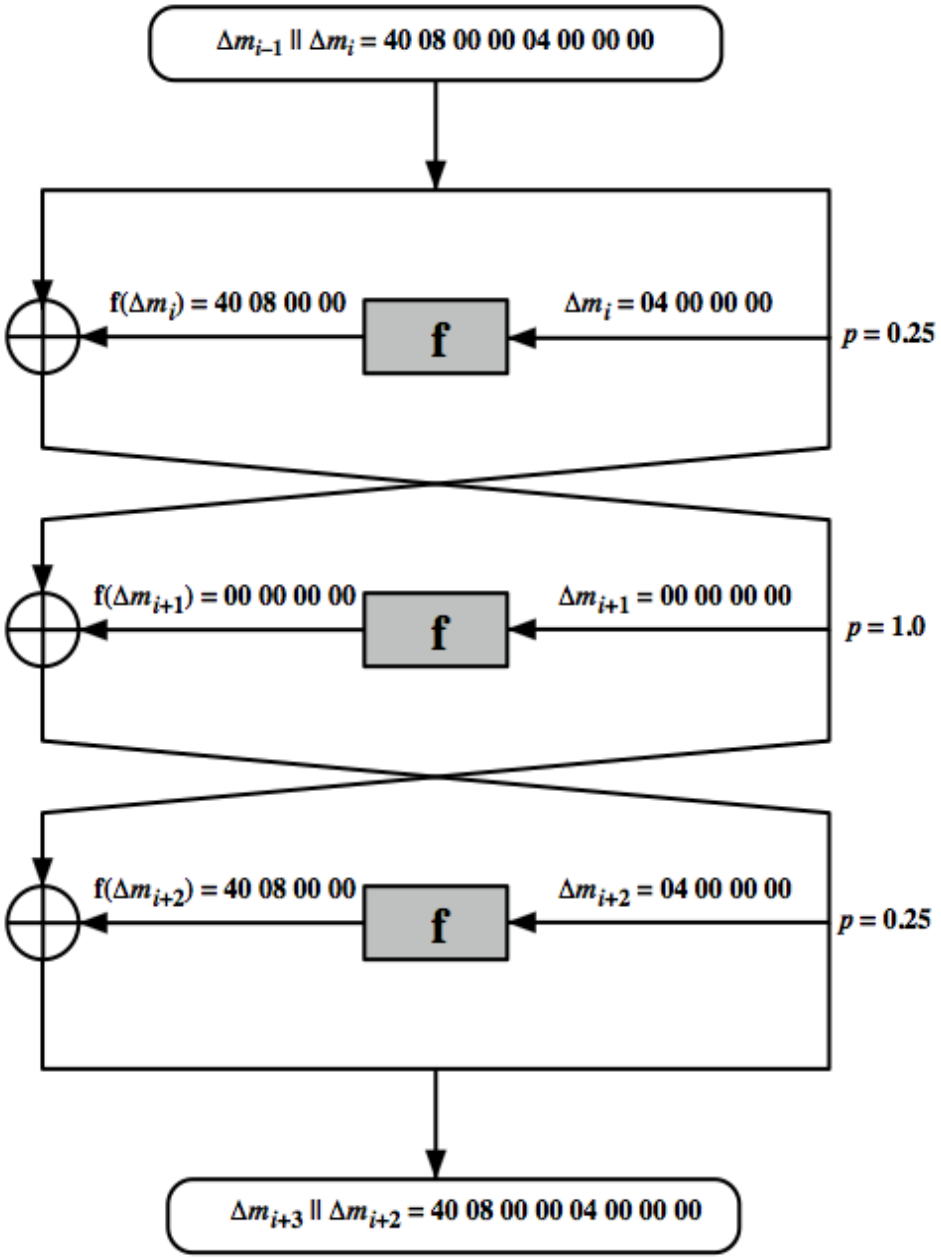
Differential Cryptanalysis (4)

- have some input difference giving some output difference with probability p
- if find instances of some higher probability input / output difference pairs occurring
- can infer subkey that was used in round
- then must iterate process over many rounds (with decreasing probabilities)
- This procedure must be repeated many times to determine all the key bits.

Differential Cryptanalysis (5)

Overall probability of given output difference is

$$0.25 \times 1.0 \times 0.25 = 0.0625$$



Linear Cryptanalysis

- another fairly recent development
- also a statistical method
- must be iterated over rounds, with decreasing probabilities
- developed by Matsui et al in early 90's
- based on finding linear approximations
- can attack DES with 2^{43} known plaintexts, easier but still in practice infeasible

Block Cipher Design Principles: Number of Rounds

- The greater the number of rounds, the more difficult it is to perform cryptanalysis
- In general, the criterion should be that the number of rounds is chosen so that known cryptanalytic efforts require greater effort than a simple brute-force key search attack
- If DES had 15 or fewer rounds, differential cryptanalysis would require less effort than a brute-force key search

Block Cipher Design Principles: Design of Function F

- The heart of a Feistel block cipher is the function F
- The more nonlinear F, the more difficult any type of cryptanalysis will be
- The algorithm should have good avalanche properties
  - Strict avalanche criterion (SAC): **States that any output bit j of an S-box should change with probability 1/2 when any single input bit i is inverted for all i , j**
  - Bit independence criterion (BIC) : **States that output bits j and k should change independently when any single input bit i is inverted for all i , j , and k**
- The SAC and BIC criteria appear to strengthen the effectiveness of the confusion function

Block Cipher Design Principles: Key Schedule Algorithm

- With any Feistel block cipher, the key is used to generate one subkey for each round
- In general, we would like to select subkeys to maximize the difficulty of deducing individual subkeys and the difficulty of working back to the main key
- It is suggested that, at a minimum, the key schedule should guarantee key/ciphertext Strict Avalanche Criterion and Bit Independence Criterion

Advanced Encryption Standard (AES)

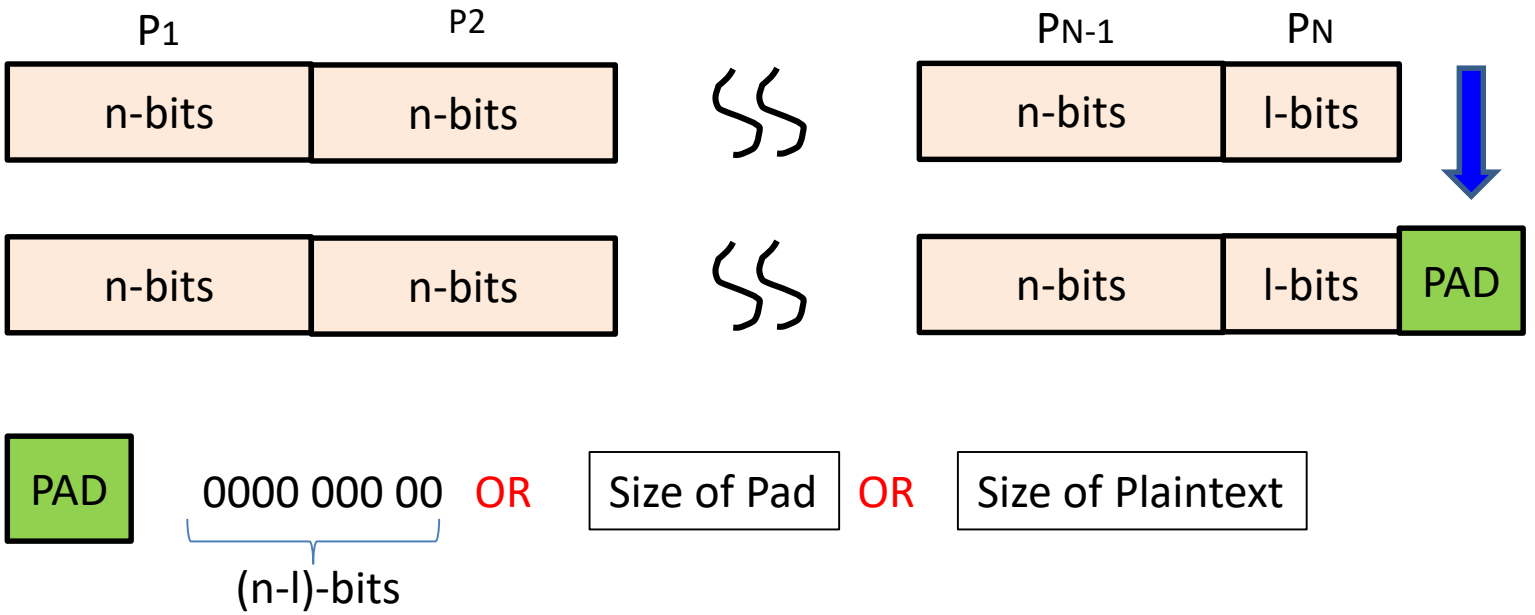
- NIST (National Institute of Standards and Technology) created a program for the development of Advanced Encryption Standard (AES) (first call Sept. 97)
- "Winner" – **Rijndael** announced Oct. 2000
- Rijndael (Daemen and Rijmen) supports keys of 128, 192, or 256 bits and messages of 128, 192, or 256 bits (AES uses only 128 bit blocks)
- Designed to be resistant to linear or differential cryptanalysis
- Fast and efficient in hardware and software implementations

How to use a block cipher?

- Block ciphers encrypt fixed-size blocks
  - e.g. DES encrypts 64-bit blocks
- We need some way to encrypt a message of arbitrary length
  - e.g. a message of 1000 bytes
- NIST (National Institute of Standards and Technology) defines several ways to deal with it, called modes of operation
  - ECB (Electronic Code Book)
  - CBC (Cipher Block Chaining)

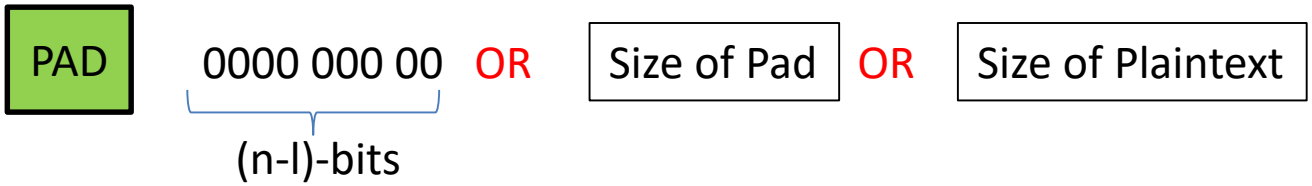
Message Padding

- The plaintext message is broken into blocks, P_1, P_2, P_3
- The last block may be short of a whole block and needs padding.

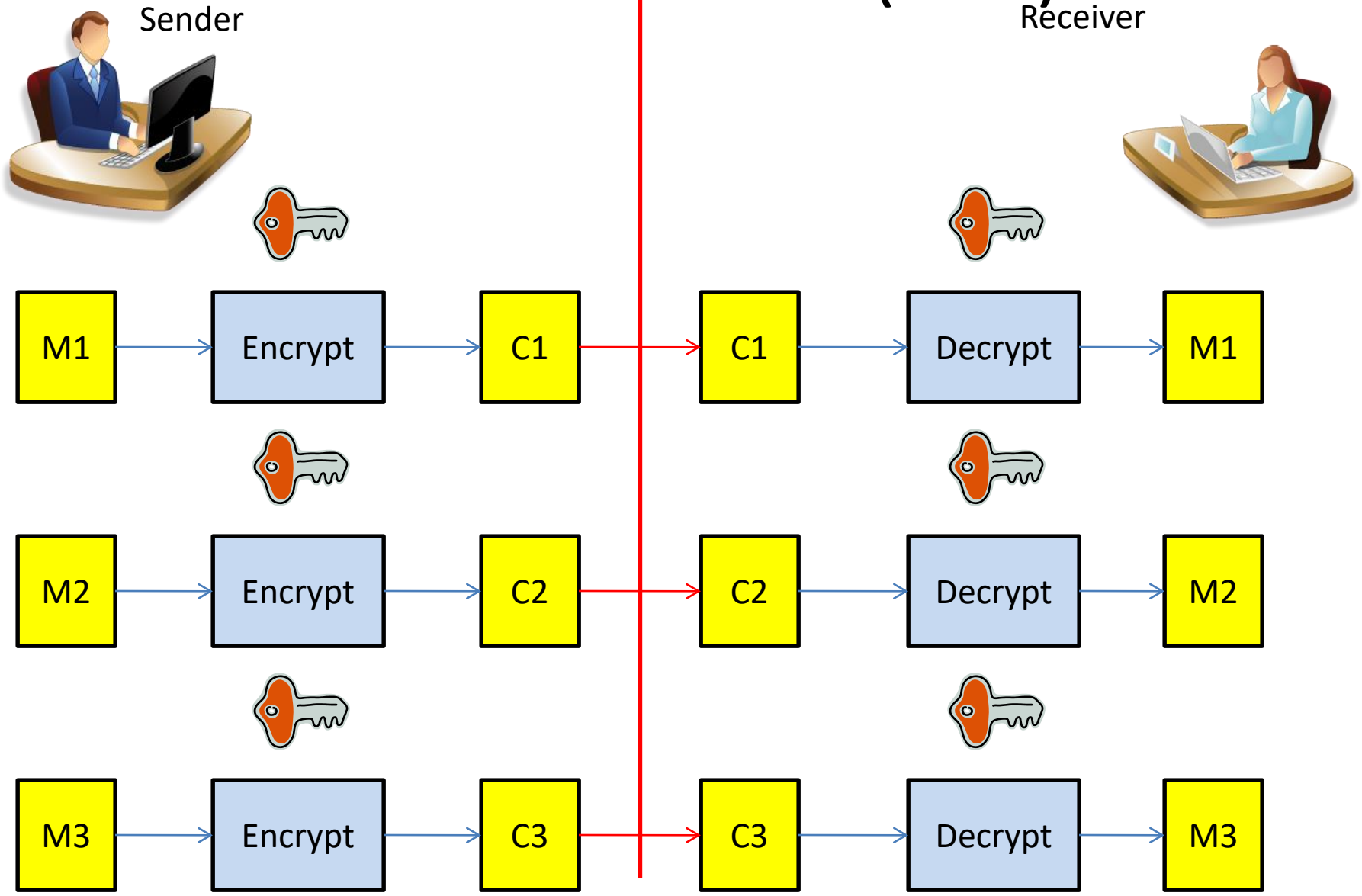


Message Padding (2)

- Possible padding:
  - Known non-data values (e.g. nulls)
  - Or a number indicating the size of the pad
  - Or a number indicating the size of the plaintext
  - The last two may require an extra block.



Electronic Code Book (ECB) Mode



Electronic Code Book (ECB) Mode (2)

- Cipher acts as simple block substitution determined by key
- For a given key, this mode behaves like we have a gigantic codebook, in which each plaintext block has an entry, hence the name Electronic Code Book
- Fast and simple but repeated input block creates repeated ciphertext block
- Vulnerable to replay attacks: if an attacker thinks block C_2 corresponds to $ amount, then substitute another C_k
- Attacker can also build a codebook of $< C_k;$ guessed $M_k >$ pairs
- Application: secure transmission of short pieces of information (e.g. a temporary encryption key)

Cipher Block Chaining (CBC) Mode

Cipher Block Chaining (CBC) Mode (2)

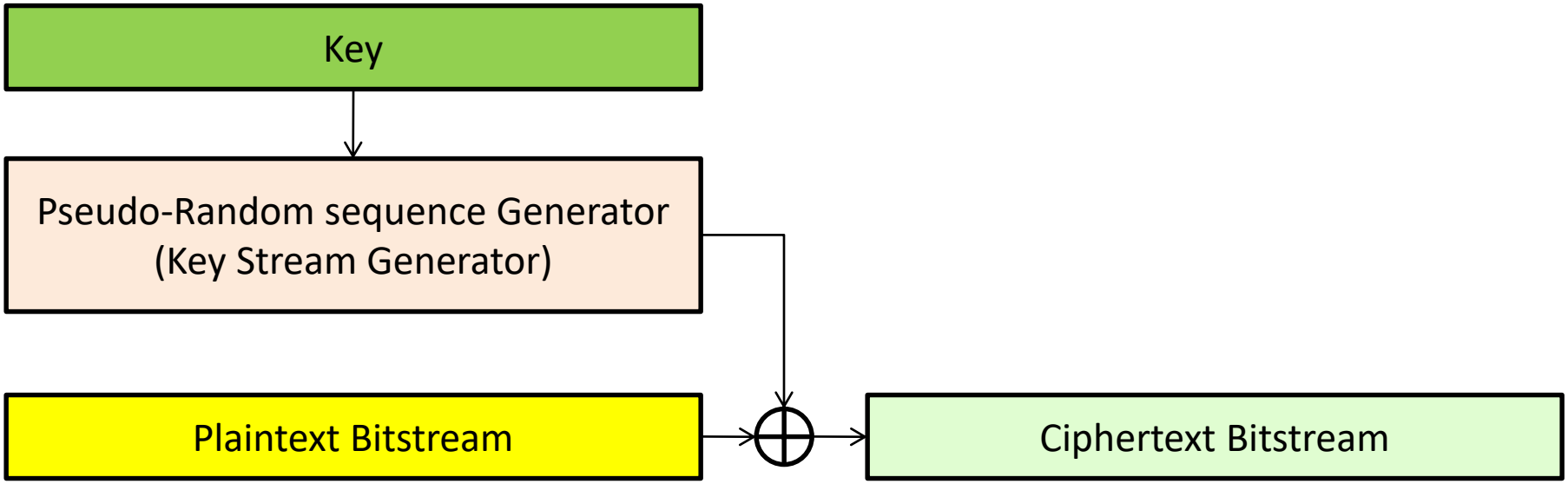
- The plaintext is broken into blocks M_1, M_2, M_3
- Each plaintext block is XORed (chained) with the previous ciphertext block before encryption (hence the name CBC)

$$C_i = E_k(C_{i-1} \oplus M_i); \;\; C_0 = IV$$

  - The encryption of a block depends on the current and all blocks before it. Then, the input plaintext $M_i = M_k$ will not result in the same output code due to memory-based chaining
- Use an Initial Vector (IV) (Use only once) to start the process
- Decryption: $M_i = C_{i-1} \oplus D_k(C_i)$
- Application: general block-oriented transmission.

Stream Ciphers

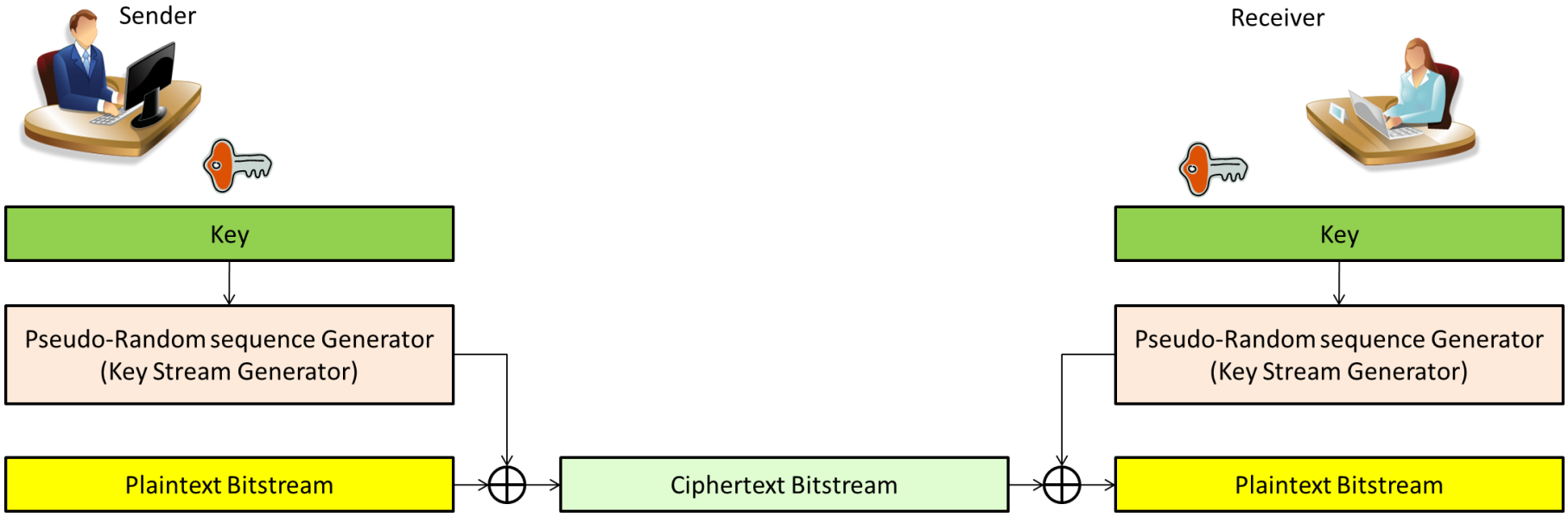
- Many times data is transmitted in serial form (one bit at a time)
- Cipher generates "Key-Stream" which is combined with "Message-Stream" to produce "Cipher-Stream"



Plaintext Stream	1	1	1	1	1	1	0	0	0	0	0	…
Pseudo-Random Stream	1	0	0	1	1	0	1	1	0	1	0	…
Ciphertext Stream	0	1	1	0	0	1	1	1	0	1	0	…

Stream Ciphers (2)

- Inverted at receiver by combining the same Key Stream
- Better than Block Ciphers for Serial Communication Channels
- Repeated input patterns do not produce repeated cipher stream sequences



Plaintext Stream	1	1	1	1	1	1	0	0	0	0	0	...
Pseudo-Random Stream	1	0	0	1	1	0	1	1	0	1	0	...
Ciphertext Stream	0	1	1	0	0	1	1	1	0	1	0	...

Stream Ciphers (3)

- Only adds Confusion (no Diffusion)
- If Synchronization lost between sender and receiver (Cryptosync) – must resync

- "no Diffusion" → What shall we know about the key?

Dangers of Keystream Reuse in Stream Cipher

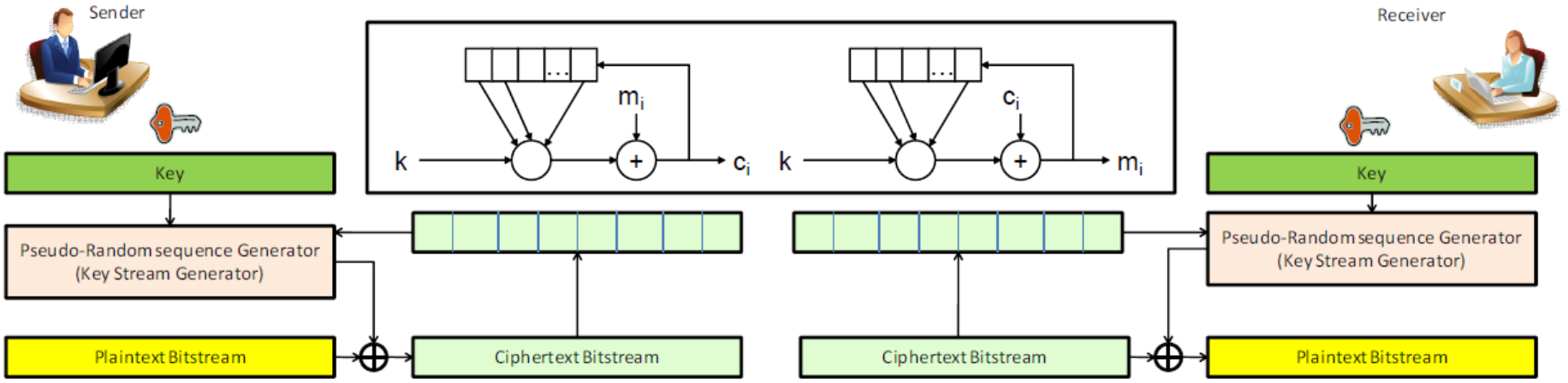
- If we have a random stream, why don't we just reuse it?

$$(M_1 \oplus K) \oplus (M_2 \oplus K) = M_1 \oplus M_2$$

- If M_1 or M_2 is known or discovered, then the other one will also be known.

Self-synchronous Stream Cipher

- Use ciphertext as feedback into the stream generation process. Key-stream is generated as a function of the key and a fixed number of previous ciphertext digits.
- If Cryptosync lost, resynchronization will be re-established once bad bits have passed through feedback register



Plaintext Stream	1	1	1	1	1	1	0	0	0	0	0	...
Pseudo-Random Stream	1	0	0	1	1	0	1	1	0	1	0	...
Ciphertext Stream	0	1	1	0	0	1	1	1	0	1	0	...


Thank
you

University of New Brunswick, Faculty of Computer Science

CS4355/6355: Topic 2 – Additional Note

1 DES and AES Problems

1. Let K be a 56-bit DES key, and let M be a 64-bit plaintext, given the ciphertext

$$C = DES(K, M) \tag{1.1}$$

how to recover the key K and the plaintext M?

Solutions.

- Case 1: If M is meaningless, e.g., password, secret key, we cannot verify whether a key is correct or not.
- Case 2: If M is meaningful,
  - For each key $k \in \{0,1\}^{56}$ do
  - $M = DES^{-1}(k, C)$
  - if M is meaningful, return $k||M$

2. Let K be a 56-bit DES key, let L be a 64-bit string, and let M be a 64-bit plaintext, check the following two algorithms derived from DES are secure or not.

$$case\ 1: \quad C = DES(K, L \oplus M) \tag{1.2}$$

$$case\ 2: \quad C = L \oplus DES(K, M) \tag{1.3}$$

For each algorithm, three pairs of plaintext-ciphertext $(M_1, C_1), (M_2, C_2), (M_3, C_3)$ are available for your cryptanalysis.

Solutions.

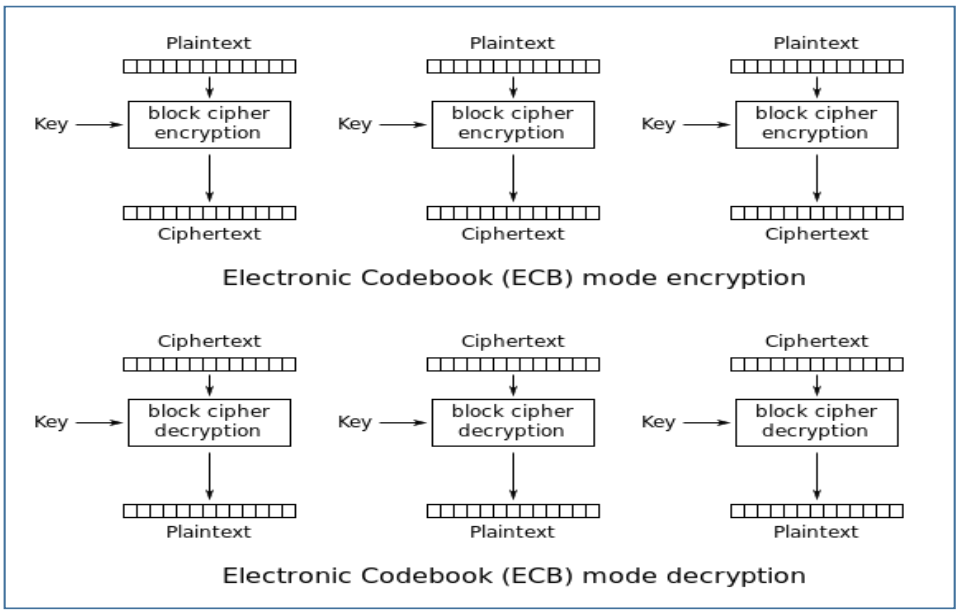
- Case 1
  - For each key $k \in \{0,1\}^{56}$ do
  - $L_1 = DES^{-1}(k, C_1) \oplus M_1$, $L_2 = DES^{-1}(k, C_2) \oplus M_2$, and $L_3 = DES^{-1}(k, C_3) \oplus M_3$
  - if $L_1 = L_2 = L_3$, return $k||L_1$
- Case 2
  - For each key $k \in \{0,1\}^{56}$ do
  - $L_1 = DES(k, M_1) \oplus C_1$, $L_2 = DES(k, M_2) \oplus C_2$, and $L_3 = DES(k, M_3) \oplus C_3$
  - if $L_1 = L_2 = L_3$, return $k||L_1$

3. Assume AES is a secure PRF (Pseudorandom Function), define a function $F(K, M) = AES(M, K)$. Is $F(K, M)$ is a secure PRF?

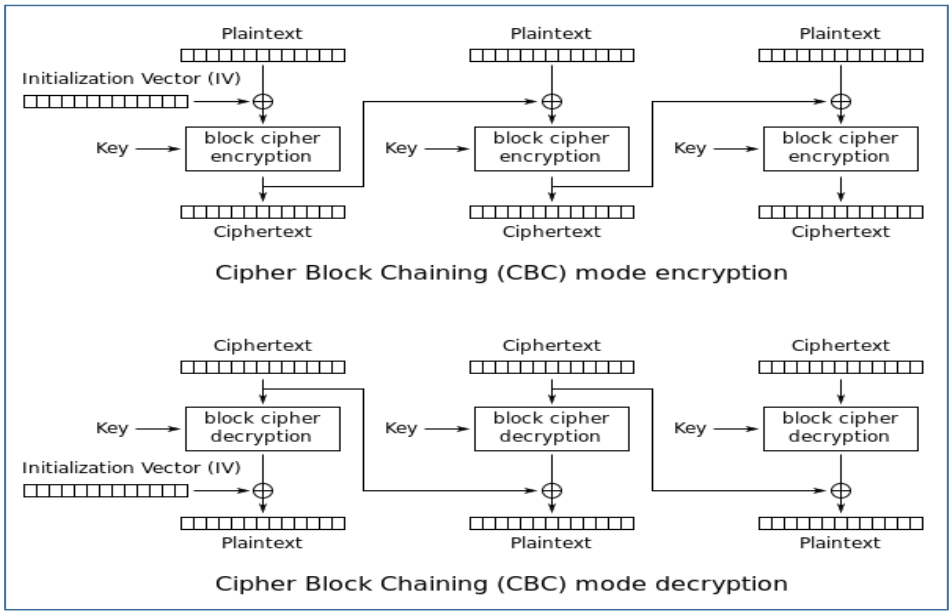
   Solution. Once we are given a pair of plaintext-ciphertext (M, C), we can easily recover the key K as $AES^{-1}(M, C) = K$. Thus, $F(K, M)$ is not a secure PRF.

2 Block Cipher Modes

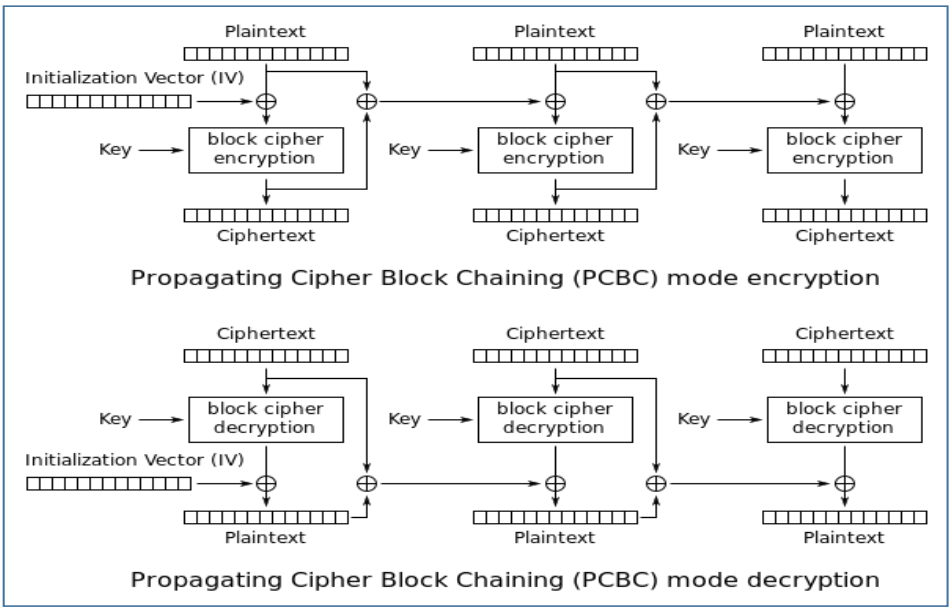
- Electronic Codebook (ECB)



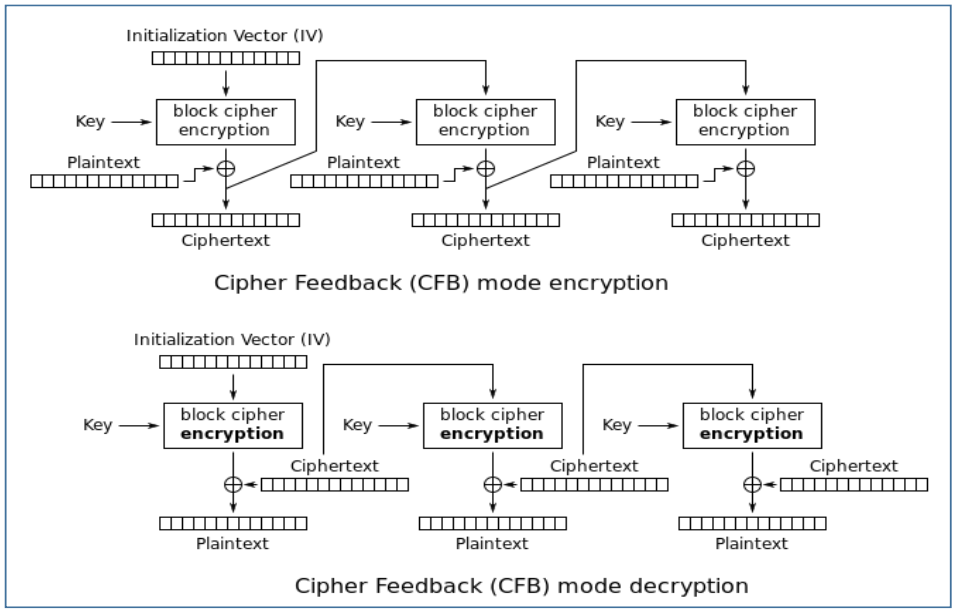
- Cipher Block Chaining (CBC)



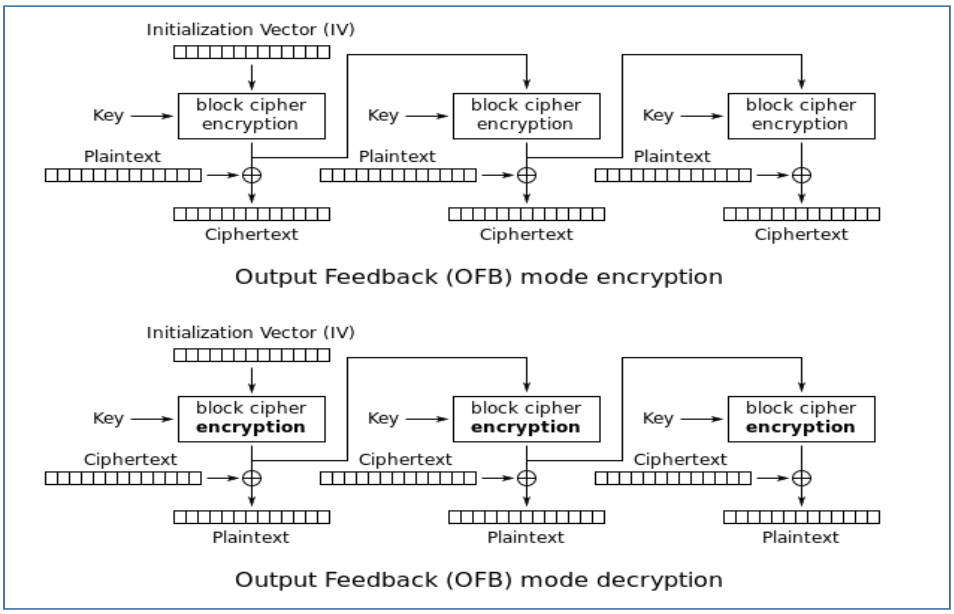
- Propagating Cipher Block Chaining (PCBC)



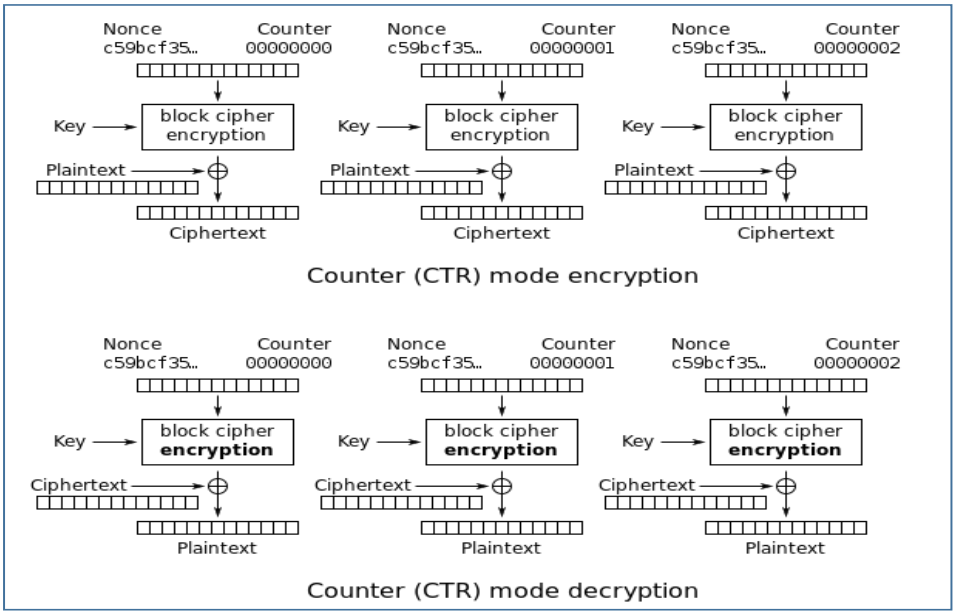
- Cipher Feedback (CFB)



- Output Feedback (OFB)


Initialization Vector (IV)
Key
block cipher encryption
Plaintext
Ciphertext
Output Feedback (OFB) mode encryption
Initialization Vector (IV)
Key
block cipher encryption
Ciphertext
Plaintext
Output Feedback (OFB) mode decryption

- Counter (CTR)


Nonce
c59bcf35...
Counter
00000000
00000001
00000002
Key
block cipher encryption
Plaintext
Ciphertext
Counter (CTR) mode encryption
Counter (CTR) mode decryption

CS 6355/4355:
Cryptanalysis and Database Security

Topic 3: Finite Fields and Number Theory

Lecturer: Rongxing LU

Email: RLU1@unb.ca Office: GE 114

Website: http://www.cs.unb.ca/~rlu1/

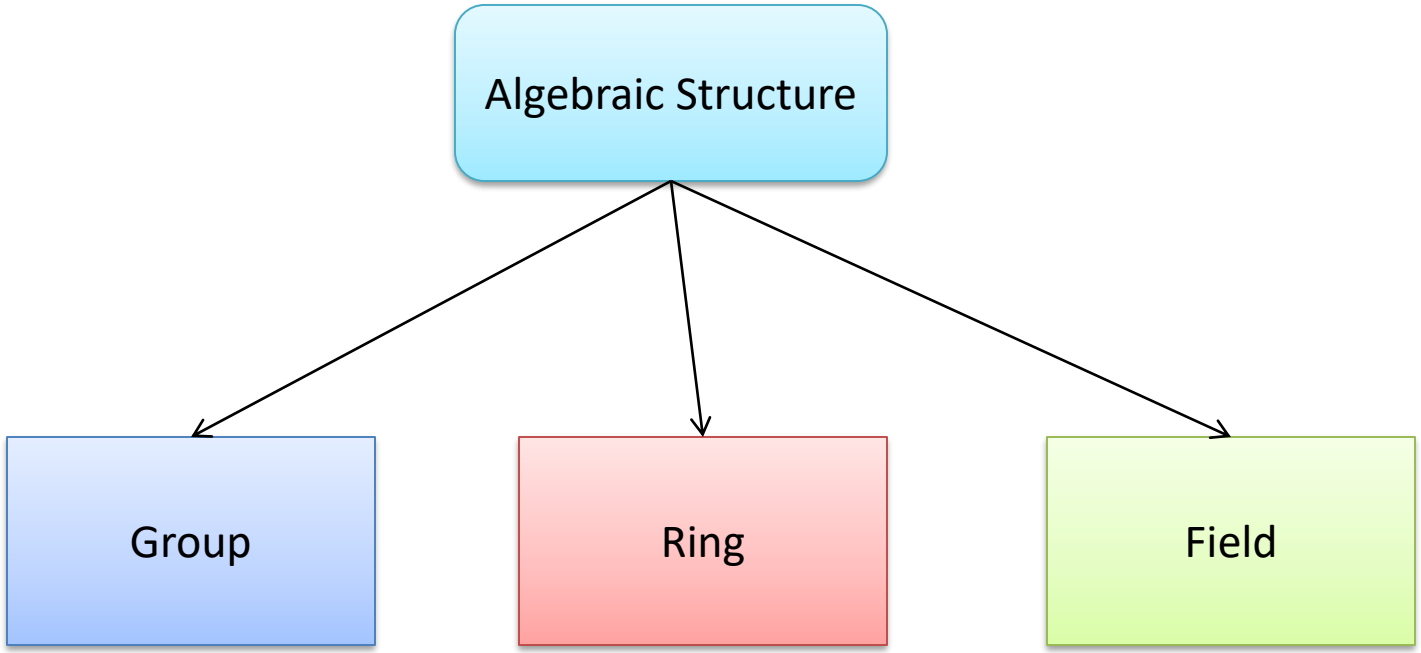
Faculty of Computer Science, University of New Brunswick

Mathematical Fundamentals

- Algebraic Structure (Group, Ring, and Finite Fields)
- Number Theory

Algebraic Structure

- Public key cryptography requires sets of integers and specific operations that are defined for those sets, including Group, Ring and Field.



https://www.youtube.com/watch?v=WRhFW6p832o

Evariste Galois

Basic Modular Arithmetic

- define **modulo operator** "a mod n" to be remainder when a is divided by n
- use the term **congruence** for: a ≡ b mod n
  - when divided by *n,* a and b have same remainder
  - eg. 100 ≡ 34 mod 11
  - 1:00 and 13:00 hours are the same 13 ≡ 1 mod 12
- b is called a **residue** of a mod n
  - since with integers can always write: a = qn + b
  - usually chose smallest positive remainder as residue
    - ie. 0 <= b <= n-1
  - process is known as **modulo reduction**
    - eg. -12 mod 7 = -5 mod 7 = 2 mod 7 = 9 mod 7

Modulo 8 Addition Example

+	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	0
2	2	3	4	5	6	7	0	1
3	3	4	5	6	7	0	1	2
4	4	5	6	7	0	1	2	3
5	5	6	7	0	1	2	3	4
6	6	7	0	1	2	3	4	5
7	7	0	1	2	3	4	5	6

→ X

↓ Y

↘ Z=X+Y mod 8

Group

- A group (G) is a set of elements with an operation (•) that satisfies four properties (or axioms). A commutative group satisfies an extra property, commutativity:
  - **Closure:** with some operation (•) whose result is also in the set
  - **Associativity:** (a.b).c = a.(b.c)
  - **Existence of identity:** e: e.a = a.e = a
  - **Existence of inverse:** a^{-1}: $a.a^{-1} = e$
  - **Commutativity:** a.b = b.a
    - => forms an **abelian (commutative) group**

Group (cont.)

- A group (G, •) involves a single operation, the properties imposed on the operation allow the use of a pair of operations as long as they are inverses of each other.
- Example 1.
  - The set of residue integers with the addition operator, G = < Z_n , +>, is a commutative group. We can perform addition and subtraction on the elements of this set without moving out of the set.
  - Z_8={0,1,2,3,4,5,6,7}, Z_3={0,1,2}

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

Z_3

Group (cont.)

+	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	0
2	2	3	4	5	6	7	0	1
3	3	4	5	6	7	0	1	2
4	4	5	6	7	0	1	2	3
5	5	6	7	0	1	2	3	4
6	6	7	0	1	2	3	4	5
7	7	0	1	2	3	4	5	6

Z_8

Group (cont.)

- Example 2.
  - The set Zn^* with the multiplication operator, G = <Zn*, ×>, is also an abelian group.
  - $Z_7^* = \{1,2,3,4,5,6\}$, $Z_3^* = \{1,2\}$, Z_8^* ?

×	1	2	3	4	5	6
1	1	2	3	4	5	6
2	2	4	6	1	3	5
3	3	6	2	5	1	4
4	4	1	5	2	6	3
5	5	3	1	6	4	2
6	6	5	4	3	2	1

Z_7^*

×	1	2
1	1	2
2	2	1

Z_3^*

$Z_8^* = \{1,3,5,7\}$

Since 2, 4, 6 have no inverses in Z_8^*

Group (cont.)

- Example 3.
  - Let us define a set G = < {a, b, c, d}, •> and the operation as shown in

•	a	b	c	d
a	a	b	c	d
b	b	c	d	a
c	c	d	a	b
d	d	a	b	c

  - It is an **abelian group**
    - Closure, Associativity, Existence of identity, Existence of inverse, Commutativity

Discussion

- Check whether the following sets can form group under the given operation?
- Case 1: the set of real numbers R, for the operation $a \circ b = 2(a + b)$

Case 2

- G={1,-1}, for the ordinary multiplication operation.

Case 3

- Non-Zero Real Number Set R^*, for operation $a \circ b = 2ab$

Case 4

- Let G={(a,b)|a,b are real numbers and a‡0}, for the operation $(a,b)^{o}(c,d)=(ac,ad+b)$.

Case 4…

Exercise

- Let G={e, a, b}, the operation is defined as

	e	a	b
e	e	a	b
a	a	b	e
b	b	e	a

- Prove G is a group for the operation.

Finite Group

- Finite Group: A group having a finite number of elements.
  - Infinite Group: A group having an infinite number of elements. Some infinite groups, such as the integers or rationals.
- Finite Groups, such as Z_7^* , Z_3^* , and G = < {a, b, c, d}, •>
- The number of elements in a group

  is called the **order** of the group.

•	a	b	c	d
a	a	b	c	d
b	b	c	d	a
c	c	d	a	b
d	d	a	b	c

Subgroup

- A subset H of <G, *> is called a subgroup of G if H also forms a group under **the same operation *.** More precisely, H is a subgroup of G if the restriction of * to H x H is a group operation on H. This is usually represented by H ≤ G, read as "H is a subgroup of G".
- Trivial subgroup G ≤ G
- Nontrivial subgroups:

  J={0,4} and H={0,2,4,6},

  where J is also a

  subgroup of H
- **operation:**

  addition modulo 8

Z_8

+	0	2	4	6	1	3	5	7
0	0	2	4	6	1	3	5	7
2	2	4	6	0	3	5	7	1
4	4	6	0	2	5	7	1	3
6	6	0	2	4	7	1	3	5
1	1	3	5	7	2	4	6	0
3	3	5	7	1	4	6	0	2
5	5	7	1	3	6	0	2	4
7	7	1	3	5	0	2	4	6

Subgroup (Cont.)

- Example.
  - Is the group H = $<Z_{10}, +>$ a subgroup of the group G = $<Z_{12}, +>$?

Cyclic Subgroups

- Cyclic subgroups
  - If a subgroup of a group can be generated using the power of an element, the subgroup is called the **cyclic subgroup.**

$$a^n \rightarrow \underbrace{a \bullet a \bullet \cdots \bullet a}_{n-times}$$

Cyclic Subgroups (Cont.)

- Example.
  - Four cyclic subgroups can be made from the group G = <Z6, +>. They are H1 = <{0}, +>, H2 = <{0, 2, 4}, +>, H3 = <{0, 3}, +>, and H4 = G.

$0^0 \bmod 6 = 0$

$2^0 \bmod 6 = 0$
$2^1 \bmod 6 = 2$
$2^2 \bmod 6 = (2 + 2) \bmod 6 = 4$

$4^0 \bmod 6 = 0$
$4^1 \bmod 6 = 4$
$4^2 \bmod 6 = (4 + 4) \bmod 6 = 2$

$3^0 \bmod 6 = 0$
$3^1 \bmod 6 = 3$

$1^0 \bmod 6 = 0$
$1^1 \bmod 6 = 1$
$1^2 \bmod 6 = (1 + 1) \bmod 6 = 2$
$1^3 \bmod 6 = (1 + 1 + 1) \bmod 6 = 3$
$1^4 \bmod 6 = (1 + 1 + 1 + 1) \bmod 6 = 4$
$1^5 \bmod 6 = (1 + 1 + 1 + 1 + 1) \bmod 6 = 5$

$5^0 \bmod 6 = 0$
$5^1 \bmod 6 = 5$
$5^2 \bmod 6 = 4$
$5^3 \bmod 6 = 3$
$5^4 \bmod 6 = 2$
$5^5 \bmod 6 = 1$

Cyclic Subgroups (Cont.)

- Example.
  - Three cyclic subgroups can be made from the group G = $<Z_{10}^*, \times>$.

Cyclic Groups

- Cyclic Group.
  - A cyclic group is a group that is its own cyclic subgroup.

$$\{e, g, g^2, \cdots, g^{n-1}\}, where \sim g^n = e$$

Cyclic Groups

- Cyclic Group.
  - A cyclic group is a group that is its own cyclic subgroup.
  - Example. $\{e, g, g^2, \cdots, g^{n-1}\}, where \sim g^n = e$
    - Three cyclic subgroups can be made from the group G = $<Z_{10}^*, \times>$. G has only four elements: 1, 3, 7, and 9. The cyclic subgroups are H1 = <{1}, ×>, H2 = <{1, 9}, ×>, and H3 = G.
    - The group G = $<Z_{10}^*, \times>$ is a cyclic group with two generators, g = 3 and g = 7.

×	1	3	7	9
1	1	3	7	9
3	3	9	1	7
7	7	1	9	3
9	9	7	3	1

$7^0 \bmod 10 = 1$
$7^1 \bmod 10 = 7$
$7^2 \bmod 10 = 9$
$7^3 \bmod 10 = 3$

$3^0 \bmod 10 = 1$
$3^1 \bmod 10 = 3$
$3^2 \bmod 10 = 9$
$3^3 \bmod 10 = 7$

Cyclic Groups (Cont.)

- Cyclic Group.
  - Example.
    - The group G = <Z6, +> is a cyclic group with two generators, g = 1 and g = 5.

+	0	1	2	3	4	5
0	0	1	2	3	4	5
1	1	2	3	4	5	0
2	2	3	4	5	0	1
3	3	4	5	0	1	2
4	4	5	0	1	2	3
5	5	0	1	2	3	4

	$g = 1$	$g = 5$
g	1	5
$g+g$	2	4
$g+g+g$	3	3
$g+g+g+g$	4	2
$g+g+g+g+g$	5	1
$g+g+g+g+g+g$	0	0

Lagrange's Theorem

- Assume that G is a group, and H is a subgroup of G. If the orders of G and H are |G| and |H|, respectively, then, based on Lagrange's theorem, |H| divides |G|.
- Example.
  - Four cyclic subgroups can be made from the group G = <Z6, +>. They are H1 = <{0}, +>, H2 = <{0, 2, 4}, +>, H3 = <{0, 3}, +>, and H4 = G.
- Order of an Element
  - The order of an element is the order of the cyclic group it generates.

$0^0 \bmod 6 = 0$

$2^0 \bmod 6 = 0$
$2^1 \bmod 6 = 2$
$2^2 \bmod 6 = (2 + 2) \bmod 6 = 4$

$4^0 \bmod 6 = 0$
$4^1 \bmod 6 = 4$
$4^2 \bmod 6 = (4 + 4) \bmod 6 = 2$

$3^0 \bmod 6 = 0$
$3^1 \bmod 6 = 3$

$1^0 \bmod 6 = 0$
$1^1 \bmod 6 = 1$
$1^2 \bmod 6 = (1 + 1) \bmod 6 = 2$
$1^3 \bmod 6 = (1 + 1 + 1) \bmod 6 = 3$
$1^4 \bmod 6 = (1 + 1 + 1 + 1) \bmod 6 = 4$
$1^5 \bmod 6 = (1 + 1 + 1 + 1 + 1) \bmod 6 = 5$

$5^0 \bmod 6 = 0$
$5^1 \bmod 6 = 5$
$5^2 \bmod 6 = 4$
$5^3 \bmod 6 = 3$
$5^4 \bmod 6 = 2$
$5^5 \bmod 6 = 1$

Ring

- A ring, R = <{a,b,c,…}, ●, ■>, is an algebraic structure with two operations.

Set	Operation •	Operation ▪
{a,b,c,…}	1. Closure 2. Associativity 3. Commutativity 4. Existence of identity 5. Existence of inverse	1. Closure 2. Associativity 3. Commutativity (The third property is only satisfied for a commutative ring)

distributivity

$$a \blacksquare (b \cdot c) = a \blacksquare b \cdot a \blacksquare c \quad , \qquad (b \cdot c) \blacksquare a = b \blacksquare c \cdot c \blacksquare a$$

Ring (Cont.)

- The set Z with two operations, addition and multiplication, is a commutative ring. We show it by R = <Z, +, ×>. Addition satisfies all of the five properties; multiplication satisfies only three properties.
- Z={…, -2, -1, 0, 1, 2, …}

+	…	-2	-1	0	1	…
…	…	…	…	…	…	…
-2	…	-4	-3	-2	-1	…
-1	…	-3	-2	-1	0	…
0	…	-2	-1	0	1	…
1	…	-1	0	1	2	…
…	…	…	…	…	…	…

×	…	-2	-1	0	1	…
…	…	…	…	…	…	…
-2	…	4	2	0	-2	…
-1	…	2	1	0	-1	…
0	…	0	0	0	0	…
1	…	-2	-1	0	1	…
…	…	…	…	…	…	…

Example

- Let R be a ring with identity (denoted as 1). Prove R is also a ring with identity under the operations $a \oplus b = a + b - 1, a^{\circ}b = a + b - ab$

Field

- A field, denoted by F = <{a,b,c,…}, ●, ■>, is a commutative ring in which the second operation satisfies all five properties defined for the first operation except that the identity of the first operation has no inverse.

Set	Operation •	Operation ▪
{a,b,c,…}	1. Closure 2. Associativity 3. Commutativity 4. Existence of identity 5. Existence of inverse	1. Closure 2. Associativity 3. Commutativity 4. Existence of identity 5. Existence of inverse*

* The identity element of the first operation has no inverse with respect to the second operation

Finite Fields

- Galois showed that for a field to be finite, the number of elements should be p^n, where p is a prime and n is a positive integer.
- A Galois field, GF(p^n), is a finite field with p^n elements.
- When n = 1, we have GF(p) field. This field can be the set Z_p, {0, 1, …, p − 1}, with two arithmetic operations.
- A very common field in this category is GF(2) with the set {0, 1} and two operations, addition and multiplication.

GF(2)	
{0, 1}	(+, x)

+	0	1
0	0	1
1	1	0

x	0	1
0	0	0
1	0	1

a	0	1
-a	0	1

a	0	1
a^{-1}	--	1

Finite Fields (Cont.)

- We can define GF(5) on the set Z_5 (5 is a prime) with addition and multiplication operators

GF(5)	
{0, 1,2,3,4}	(+, x)

+	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

x	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	1	4	2
4	0	4	3	2	1

a	0	1	2	3	4
-a	0	4	3	2	1

a	0	1	2	3	4
a^{-1}	--	1	3	2	4

Finite Fields (Cont.)

Arithmetic in GF(2^3) -- Addition

	+	000	001	010	011	100	101	110	111
		0	1	2	3	4	5	6	7
000	0	0	1	2	3	4	5	6	7
001	1	1	0	3	2	5	4	7	6
010	2	2	3	0	1	6	7	4	5
011	3	3	2	1	0	7	6	5	4
100	4	4	5	6	7	0	1	2	3
101	5	5	4	7	6	1	0	3	2
110	6	6	7	4	5	2	3	0	1
111	7	7	6	5	4	3	2	1	0

(a) Addition

Finite Fields (Cont.)

Arithmetic in GF(2^3) - Multiplication

		000	001	010	011	100	101	110	111
	×	0	1	2	3	4	5	6	7
000	0	0	0	0	0	0	0	0	0
001	1	0	1	2	3	4	5	6	7
010	2	0	2	4	6	3	1	7	5
011	3	0	3	6	5	7	4	1	2
100	4	0	4	3	7	6	2	5	1
101	5	0	5	1	4	2	7	3	6
110	6	0	6	7	1	5	3	2	4
111	7	0	7	5	2	1	6	4	3

Finite Fields (Cont.)

Arithmetic in GF(2^3) – Identity, Inverse

w	0	1	2	3	4	5	6	7
$-w$	0	1	2	3	4	5	6	7
w^{-1}	---	1	5	6	7	2	3	4

Finite Fields (Cont.) GF(2^3)

Polynomial Arithmetic Modulo ($x^3 + x + 1$)

	+	000 0	001 1	010 x	011 $x+1$	100 x^2	101 x^2+1	110 x^2+x	111 x^2+x+1
000	0	0	1	x	$x+1$	x^2	x^2+1	x^2+x	x^2+x+1
001	1	1	0	$x+1$	x	x^2+1	x^2	x^2+x+1	x^2+x
010	x	x	$x+1$	0	1	x^2+x	x^2+x+1	x^2	x^2+1
011	$x+1$	$x+1$	x	1	0	x^2+x+1	x^2+x	x^2+1	x^2
100	x^2	x^2	x^2+1	x^2+x	x^2+x+1	0	1	x	$x+1$
101	x^2+1	x^2+1	x^2	x^2+x+1	x^2+x	1	0	$x+1$	x
110	x^2+x	x^2+x	x^2+x+1	x^2	x^2+1	x	$x+1$	0	1
111	x^2+x+1	x^2+x+1	x^2+x	x^2+1	x^2	$x+1$	x	1	0

(a) Addition

	×	000 0	001 1	010 x	011 $x+1$	100 x^2	101 x^2+1	110 x^2+x	111 x^2+x+1
000	0	0	0	0	0	0	0	0	0
001	1	0	1	x	$x+1$	x^2	x^2+1	x^2+x	x^2+x+1
010	x	0	x	x^2	x^2+x	$x+1$	1	x^2+x+1	x^2+1
011	$x+1$	0	$x+1$	x^2+x	x^2+1	x^2+x+1	x^2	1	x
100	x^2	0	x^2	$x+1$	x^2+x+1	x^2+x	x	x^2+1	1
101	x^2+1	0	x^2+1	1	x^2	x	x^2+x+1	$x+1$	x^2+x
110	x^2+x	0	x^2+x	x^2+x+1	1	x^2+1	$x+1$	x	x^2
111	x^2+x+1	0	x^2+x+1	x^2+1	x	1	x^2+x	x^2	$x+1$

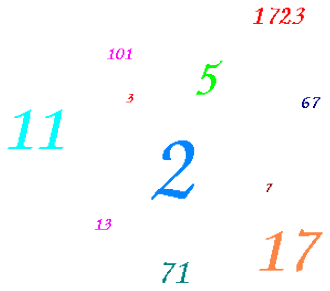
(b) Multiplication

Group, Ring, Field

Algebraic Structure	Supported Typical Operations	Supported Typical Sets of Integers
Group	(+,-) or (×,÷)	Zn or Zn*
Ring	(+,-) and (×)	Z
Field	(+,-) and (×, ÷)	Z_p

Number Theory

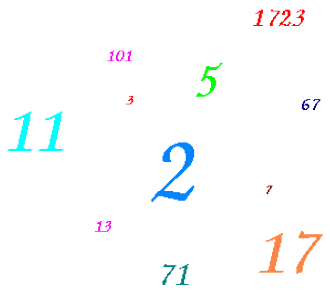
- Fundamental Number Theorem
- GCD, Euclid's algorithm
- Extended Euclid's Algorithm
- Modular Arithmetic
- Euler's Totient Function
- Fermat Theorem
- Euler's Theorem



Prime Numbers

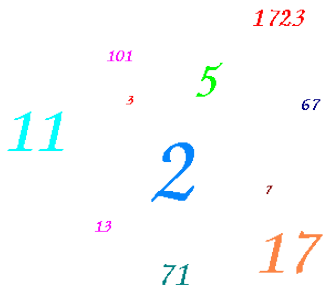
- prime numbers only have divisors of 1 and self
- they cannot be written as a product of other numbers
- note: 1 is prime, but is generally not of interest
- eg. 2,3,5,7 are prime, 4,6,8,9,10 are not
- prime numbers are central to number theory
- list of prime number less than 200 is:

2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97 101 103 107 109 113 127 131 137 139 149 151 157 163 167 173 179 181 191 193 197 199



Fundamental Theorem of Arithmetic

- All numbers can be expressed as a unique products of primes
  - 10 = 2 * 5, 20=2*2*5 , 60 = 2 * 2 * 3 * 5
- Proof in two parts
  - 1. All numbers are expressible as products of primes
  - 2. There is only one such product sequence per number



Fundamental Theorem of Arithmetic

- First part of proof
  - All numbers are products of primes

Let S = {x | x is not expressible as a product of primes}

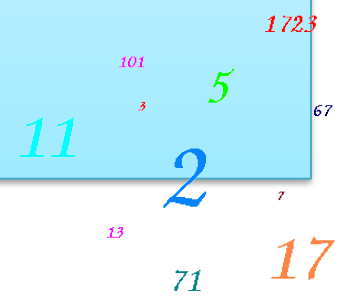
Let c = min{S}. c cannot be prime

Let $c = c_1 \cdot c_2$

$c_1, c_2 < c \Longrightarrow c_1, c_2 \notin S$ (because c is min{S})

$\therefore$ c_1, c_2 are products of primes $\Longrightarrow$ c is too

$\therefore$ S is an empty set



Fundamental Theorem of Arithmetic

- Second part of proof
  - The product of primes is unique

Let $n = p_1p_2p_3p_4\ldots = q_1q_2q_3q_4\ldots$

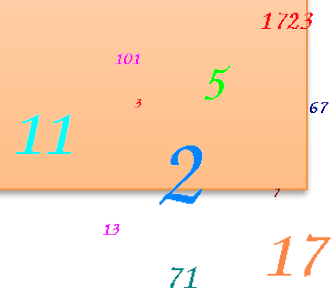
Cancel common primes. Now unique primes on both sides

Now, $p_1 \mid p_1p_2p_3p_4$ "|" divide

$\Rightarrow p_1 \mid q_1q_2q_3q_4\ldots$

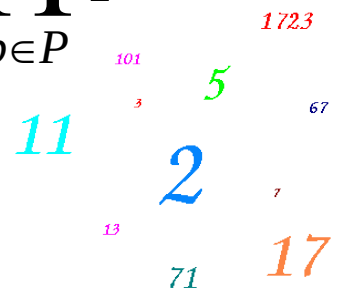
$\Rightarrow p_1 \mid$ one of $q_1, q_2, q_3, q_4\ldots$

$\Rightarrow p_1 = q_i$ which is a contradiction



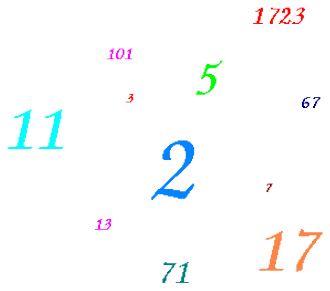
Prime Factorization

- to **factor** a number n is to write it as a product of other numbers: n=a x b x c
- note that factoring a number is relatively hard compared to multiplying the factors together to generate the number
- the **prime factorisation** of a number n is when its written as a product of primes $a = \prod_{p \in P} p^{a_p}$
  - eg. 91=7x13 ; 3600=2^4x3^2x5^2

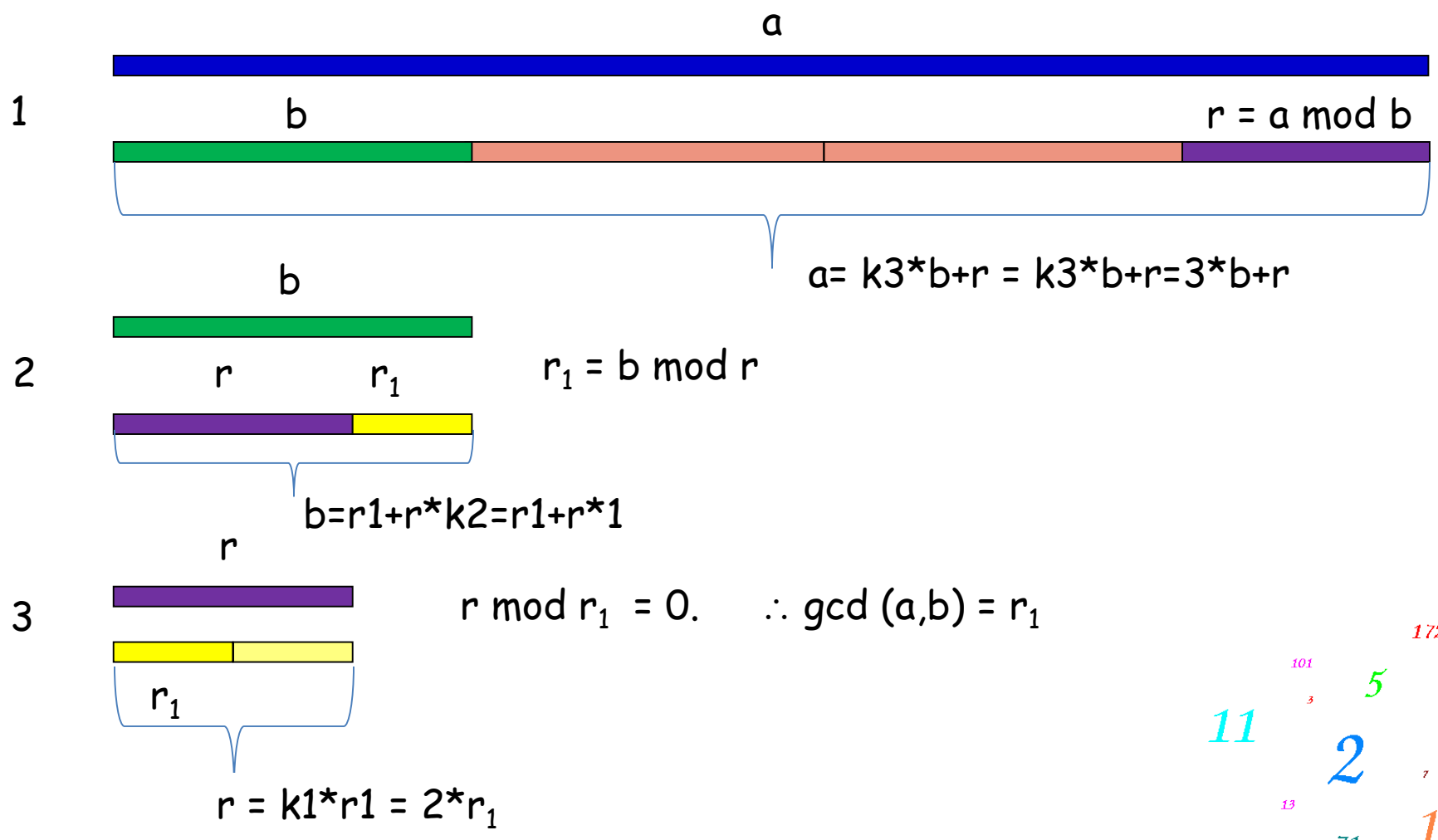


Relatively Prime Numbers & GCD

- two numbers a, b are **relatively prime** if have **no common divisors** apart from 1
  - eg. 8 & 15 are relatively prime since factors of 8 are 1,2,4,8 and of 15 are 1,3,5,15 and 1 is the only common factor
- conversely can determine the greatest common divisor (GCD) by comparing their prime factorizations and using least powers
  - eg. $300=2^1x3^1x5^2$ $18=2^1x3^2$ hence $GCD(18,300)=2^1x3^1x5^0=6$



GCD(a,b)- Euclid's algorithm



Euclid's algorithm proof

- Proof that r1 divides a and b
- Proof that r1 is the greatest divisor

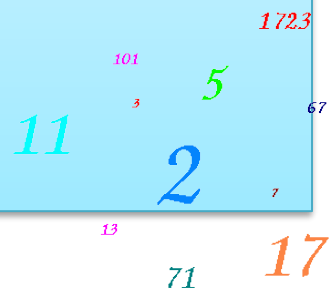
Since r1|r, we have
r = k1*r1 for some k1,
Since b=r1+r*k2 for some k2,
we have
b=r1+k1*r1*k2=r1*(1+k1*k2)
As a result, we have r1|b

Since a= k3*b+r for some k3
a= k3*b+r=k3*r1*(1+k1*k2)+
k1*r1 = r1(k3*(1+k1*k2)+k1)
We have r1|a

If there exists a number c<=a,
c | a and c | b

we have c|a➔c | k3*b+r
Since c|b, we have c|r

we have c|b➔c |(r1+r*k2)
Since c|r, we have c|r1



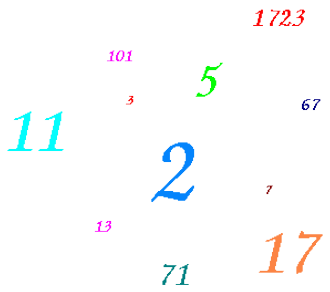
Euclid's algorithm

```
EuclidGCD(a,b)
   Assume a and b are nonnegative integers
   if (b == 0)
       gcd(a,b) = a;                  // stopping condition.
   else
       gcd(a,b) = gcd(b, a% b)        // recursive step (%=mod)
```

Examples

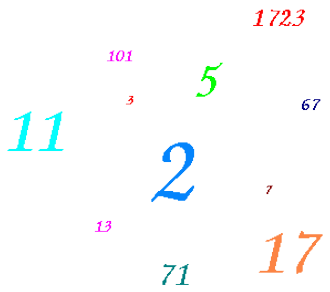
1. Let a = 54, b = 30
   gcd(54,30) = gcd(30,54 % 30) = gcd(30,24)
   gcd(30,24) = gcd(24,30 % 24) = gcd(24,6)
   gcd(24,6) = gcd(6,24 % 6) = gcd(6,0)
   gcd(6,0) = 6 // stop: gcd(54,30) = 6

2. Let a = 45, b = 16
   gcd(45,16) = gcd(16,45 % 16) = gcd(16,13)
   gcd(16,13) = gcd(13,16 % 13) = gcd(13,3)
   gcd(13,3) = gcd(3,13 % 3) = gcd(3,1)
   gcd(3,1) = gcd(1,3 % 1) = gcd(1,0)
   gcd(1,0) = 1 // stop: gcd(45,16) = 1



The gcd(a,b) as a Linear Combination of a &b

- ax+ by = "linear combination" of a and b
  - 12x + 20y = {…, -12,-8,-4,0,4,8,12, … }
- The minimum positive linear combination of a & b = gcd(a,b)
- Proof in two steps:
  - If d = min(ax+by) and d > 0, then d | a, d | b
  - d is the greatest divisor.



d | a, d | b

By contradiction

Let $S = \{z = ax + by \mid z > 0\}$

Let $d = \min\{S\} = ax_1 + by_1$

Let $a = qd + r$. $0 <= r < d$

$r = a - qd = a - q(ax_1 + by_1)$

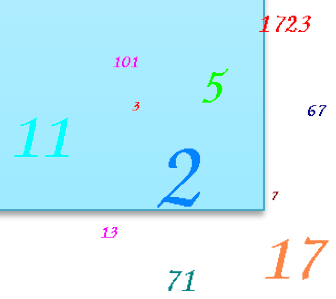
$r = a(1 - qx_1) + (-qy_1)b$

If $r > 0$, $r \in S$

But $r < d$, which is a contradiction, because $d = \min\{S\}$

$\therefore r = 0 \quad \Rightarrow d \mid a$

$\therefore$ Similarly, we have $d|b$

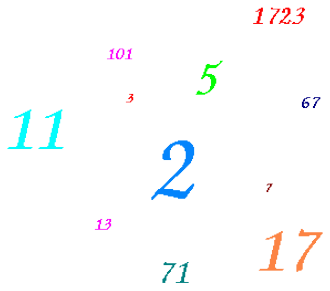


Extended Euclidean Algorithm

Given two integers a and b, we often need to find other two integers, u and v, such that

$$u \times a + v \times b = \gcd(a, b)$$

The extended Euclidean algorithm can calculate the gcd (a, b) and at the same time calculate the value of a and b.



Extended Euclidean Algorithm

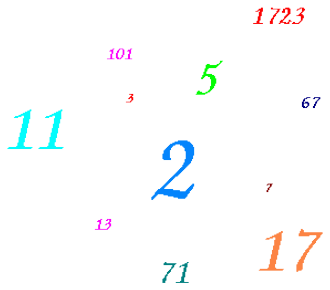
Dividend	Divisor	Quotient	Remainder
a=60	= b=13	× 4	+ 8
b=13	= 8	× 1	+ 5
8	= 5	× 1	+ 3
5	= 3	× 1	+ 2
3	= 2	× 1	+ 1

1 = 3 - 2× 1
1 = 3 – (5 - 3×1)× 1 = 3×2 - 5×1
1 = (8 – 5×1)×2 – 5×1 = 8 × 2 - 5×3
1 = 8 × 2 – (13 – 8 × 1) ×3 = 8 × 5 – 13 × 3
1 = (60 - 13×4) × 5 –13 × 3 = 60×5 – 13×23

GCD(a=60,b=13)

GCD(60,13)=1

1 = 60×5 +13×(-23)
U=5
V=-23



Modular Arithmetic

- $a \equiv b \pmod{n}$
  - n is the modulus
  - a is "congruent" to b, modulo n
  - a - b is divisible by n $n|(a-b)$
  - a % n = b % n $7\%12 = 19\%12=7$
- $a \equiv b \pmod{n}$, $c \equiv d \pmod{n}$
  - Addition

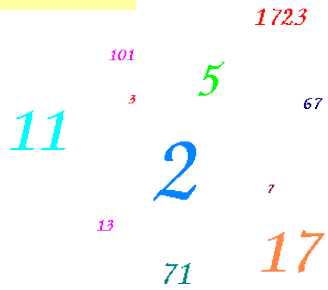
    $a + c \equiv b + d \pmod{n}$
  - Multiplication

    $ac \equiv bd \pmod{n}$

a - b = jn

c - d = kn

a + c - (b + d) = (j + k) n



Modular Arithmetic (Cont.)

- Power
  - $a \equiv b \pmod{n} \Rightarrow a^k \equiv b^k \pmod{n}$

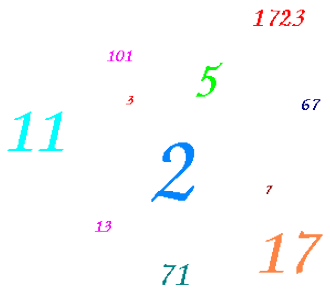
If $a^k \equiv b^k \pmod{n}$,

According to multiplication rule

$a \cdot a^k \equiv b \cdot b^k \pmod{n}$,

$\therefore\ a^{k+1} \equiv b^{k+1} \pmod{n}$

- Going n times around the clock
  - $a + kn \equiv b \pmod{n}$



Modular Arithmetic (Cont.)

- If a, b have no common factors, there exists a^i such that $a.a^i \equiv 1 \pmod b$
  - a^i is called the "multiplicative inverse"
  - We can calculate a^i with extended Euclidean algorithm

Dividend	Divisor	Quotient	Remainder
a=60	= b=13	× 4	+ 8
b=13	= 8	× 1	+ 5
8	= 5	× 1	+ 3
5	= 3	× 1	+ 2
3	= 2	× 1	+ 1

For example:

$13 \times ? \equiv 1 \bmod 60$

$13 \times 37 = 481 = 8 \times 60 + 1 \bmod 60 = 1 \bmod 60$

$1 = 3 - 2 \times 1$

$1 = 3 - (5 - 3 \times 1) \times 1 = 3 \times 2 - 5 \times 1$

$1 = (8 - 5 \times 1) \times 2 - 5 \times 1 = 8 \times 2 - 5 \times 3$

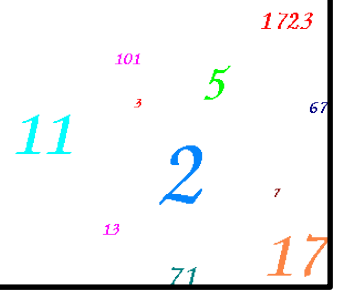
$1 = 8 \times 2 - (13 - 8 \times 1) \times 3 = 8 \times 5 - 13 \times 3$

$1 = (60 - 13 \times 4) \times 5 - 13 \times 3 = \underline{60} \times 5 - \underline{13} \times 23$

$1 = 60 \times 5 - 13 \times 23 \bmod 60 = -13 \times 23 \bmod 60$

$1 = 13 \times (-23) = 13 \times 37 \bmod 60$

Since 37+23 = 60 mod 60, 37 = -23 mod 60

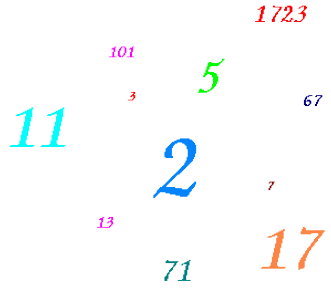


Exercises

- 1. If $p|10a - b$, $p|10c - d$, then $p|ad - bc$
- 2. if n is odd, then $3|2^n + 1$
- 3. $k = 0,1,2,\ldots$ for $n \in Z$, we have $2n + 1|1^{2k+1} + 2^{2k+1} + \cdots + (2n)^{2k+1}$
- 4. if $m - p|mn + pq$ then $m - p|mq + np$
- 5. if $x \equiv 1\ mod\ m^k$ then $x^m \equiv 1\ mod\ m^{k+1}$

Euler Totient Function ø(n)

- when doing arithmetic modulo n
- **complete set of residues** is: 0..n-1
- **reduced set of residues** is those numbers (residues) which are relatively prime to n
  - E.g., for n=10,
  - complete set of residues is {0,1,2,3,4,5,6,7,8,9}
  - reduced set of residues is {1,3,7,9}
- number of elements in reduced set of residues is called the **Euler Totient Function ø(n)**



Euler Totient Function ø(n)

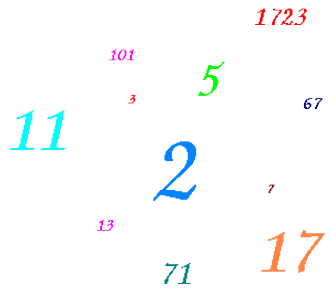
- to compute ϕ(n) need to count number of residues to be excluded
- in general need prime factorization, but
  - for p (p prime) ϕ(p) = p-1
  - for p.q (p,q prime)

    ϕ(pq) = ϕ(p) x ϕ(q)=(p-1)x(q-1)
- eg.

  ϕ(37) = 36, ϕ(11) = 10

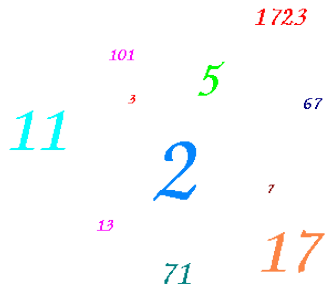
  ϕ(21) = (3–1)x(7–1) = 2x6 = 12,

  ϕ(10) = (2-1) x(5–1)=1x4=4 {1,3,7,9}



Fermat's Theorem

- $a^{p-1} = 1 \pmod p$
  - where p is prime and gcd(a,p)=1
- also known as Fermat's Little Theorem
- also $a^p = a \pmod p$
- useful in public key and primality testing
- $\phi(p)=p-1$



Euler's Theorem

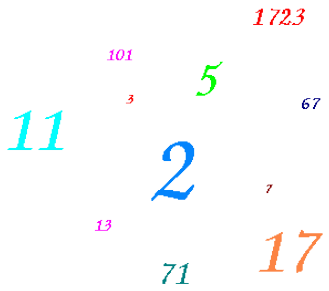
- a generalisation of Fermat's Theorem
- $a^{\phi(n)} = 1 \pmod{n}$
  - for any a,n where gcd(a,n)=1
- eg.

  a=3;n=10; $\phi(10)$=4;

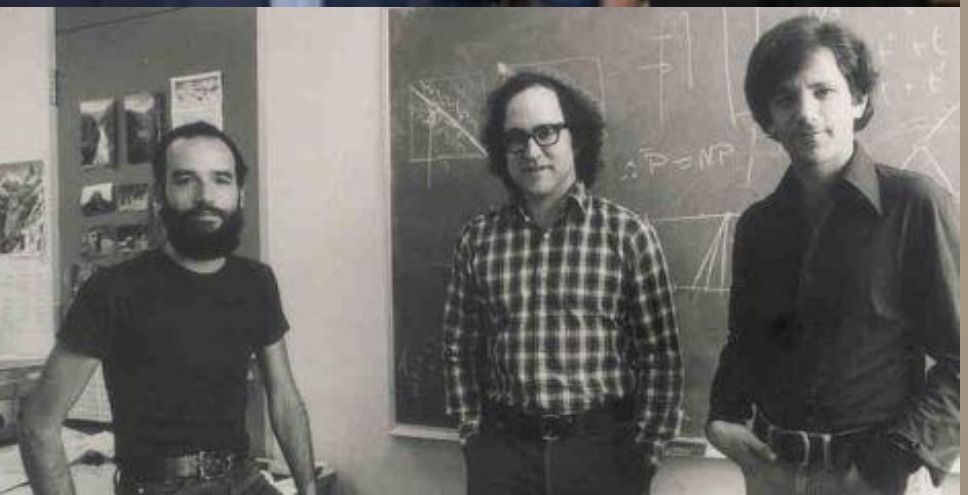
  hence $3^4 = 81 = 1 \bmod 10$

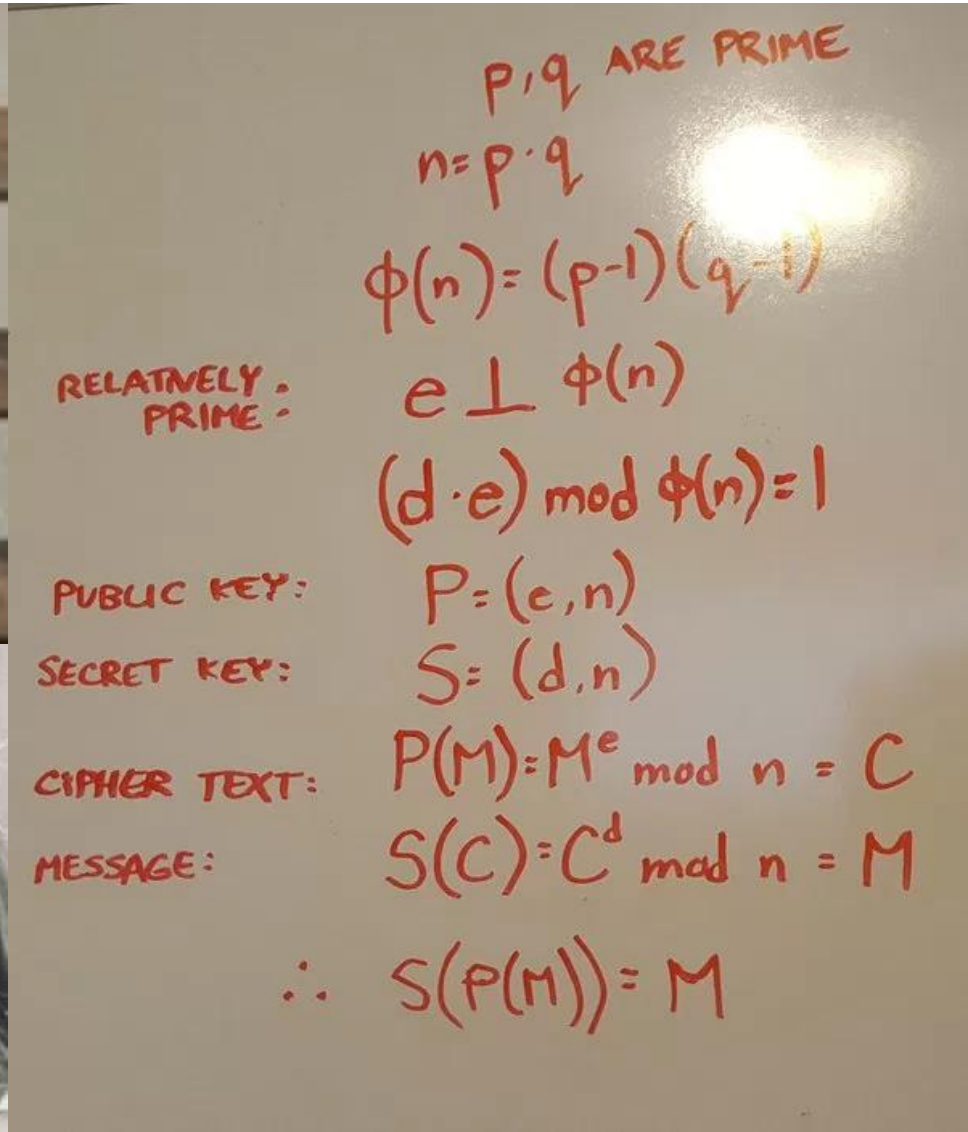
  a=2;n=11; $\phi(11)$=10;

  hence $2^{10} = 1024 = 1 \bmod 11$



RSA Public Key Cryptosystem






Thank
you

University of New Brunswick, Faculty of Computer Science

CS4355/6355: Topic 2 – Additional Note

1 Group Problems

Check whether the following sets can form group under the given operation?

- Case 1: the set of real numbers R, for the operation $a \circ b = 2(a + b)$

  Answer: Cannot. Because for the given operation $a \circ b = 2(a + b)$, there is no identity. Suppose x is the identity, we have $x \circ 0 = 2(x + 0) = 2x = 0$, thus $x = 0$. However, for 1, $1 \circ 0 = 2(1 + 0) = 2 \neq 1$, which is contradictory to $x = 0$.

- Case 2: $G = \{1, -1\}$, for the ordinary multiplication operation.

  Answer: Can.

×	1	-1
1	1	-1
-1	-1	1

- Case 3: Non-Zero Real Number Set R^*, for operation $a \circ b = 2ab$.

  Answer: Can. It is easy to see Associativity is satisfied; 1/2 is the identity of R^*, for any $a \in R^*$, $\frac{1}{4a}$ is its inverse.

- Case 4: Let $G = \{(a, b) | a, b$ are real numbers and $a \neq 0\}$, for the operation $(a, b) \circ (c, d) = (ac, ad + b)$.

  Answer: Can. Check the followings:

  – G is a non-empty set

– Closure: for any (a, b), (c, d) in G, where $a \neq 0, c \neq 0$, we have $(ac, ad + b)$ are still real numbers and $ac \neq 0$, thus $(a, b) \circ (c, d) = (ac, ad + b)$ is still in G.

– Associativity: (e, f) in G, we have

$$[(a, b) \circ (c, d)] \circ (e, f) = (ac, ad + b) \circ (e, f) = (ace, acf + ad + b)$$

$$(a, b) \circ [(c, d) \circ (e, f)] = (a, b) \circ (ce, cf + d) = (ace, acf + ad + b)$$

– Existence of Identity: $(1, 0)$ in G, and $(1, 0) \circ (a, b) = (a, b)$, i.e., $(1, 0)$ is the left identity. (it is easy to see $(1, 0)$ is the right identity)

– Existence of Inverse: (a, b) in G, we have $(1/a, -b/a)$ in G and $(1/a, -b/a) \circ (a, b) = (1, 0)$ (it is easy to see $(1/a, -b/a)$ is the right identity)

– Therefore, it is a group. But it is not a commutative group, for example

$$(3, 6) = (1, 2) \circ (3, 4) \neq (3, 4) \circ (1, 2) = (3, 10)$$

University of New Brunswick, Faculty of Computer Science

CS4355/6355: Topic 3 – Additional Note

1 Group Problems

1. Let G be a group. Please prove G is an abelian group if and only if for any elements $a, b \in G$, the condition $(ab)^2 = a^2b^2$ is true.

   Proof. (1) If G is an abelian group, then for any elements $a, b \in G$, we have $(ab)^2 = (ab)(ab) = a^2b^2$.

   (2) For any elements $a, b \in G$, we have $(ab)^2 = a^2b^2$, that is, $abab = aabb$. Both sides left-multiplies a^{-1}, right-multiplies b^{-1}, we have

   $$a^{-1}ababb^{-1} = a^{-1}aabbb^{-1} \Rightarrow ebae = eabe \Rightarrow ba = ab$$

   As a result, it is an abelian group.

2. Let G be a group, and a, b, c are any three elements in G. Please prove the equation $xaxba = xbc$ has *one and only one* solution in G.

   Proof.

   $$xaxba = xbc \Rightarrow x^{-1}xaxba = x^{-1}xbc \Rightarrow axba = bc \Rightarrow a^{-1}axba = a^{-1}bc \Rightarrow xba = a^{-1}bc$$

   $$\Rightarrow xbaa^{-1} = a^{-1}bca^{-1} \Rightarrow xb = a^{-1}bca^{-1} \Rightarrow xbb^{-1} = a^{-1}bca^{-1}b^{-1}$$

   $$\Rightarrow x = a^{-1}bca^{-1}b^{-1}$$

   Therefore, it is easy to see $x = a^{-1}bca^{-1}b^{-1}$ is one solution for the equation $xaxba = xbc$.

Assume y is another solution of the equation

$$xaxba = xbc \tag{1.1}$$

i.e., we have

$$yayba = ybc \tag{1.2}$$

From Eq.(1.1), we have

$$axbac^{-1}b^{-1} = e$$

From Eq.(1.2), we have

$$aybac^{-1}b^{-1} = e$$

As a result, $x = y$. That is, the equation $xaxba = xbc$ has *one and only one* solution in G.

3. Let G be a group, please prove the elements within each case have the same order.
   - Case 1: a and a^{-1}.

     Proof. Assume $a^n = e$, we have

     $$(a^{-1})^n = a^{-n} = (a^n)^{-1} = e^{-1} = e$$

     that is, $(a^{-1})^n = e$. On the other hand, assume $(a^{-1})^n = e$, we have

     $$a^n(a^{-1})^n = a^n a^{-n} = e.$$

     we have $a^n = e$. Therefore, $|a| = |a^{-1}|$. (Note $|a|$ denotes the order of a.)

     Proof 2. We always have $aa^{-1} = e$, we have $aaa^{-1}a^{-1} = aea^{-1} = e$. Continue it, we have

     $$a^n(a^{-1})^n = e$$

     if $a^n = e$, we have $(a^{-1})^n = e$, and vice verse.
   - Case 2: a and cac^{-1} for any $c \in G$.

     Proof. Assume $a^n = e$, we have $ca^nc^{-1} = cec^{-1} = e$. Then,

     $$\underbrace{cac^{-1}cac^{-1}\cdots cac^{-1}}_{n} = ca^nc^{-1} = e$$

     We have $(cac^{-1})^n = e$.

     On the other hand, if $(cac^{-1})^n = e$, we have

     $$e = (cac^{-1})^n = \underbrace{cac^{-1}cac^{-1}\cdots cac^{-1}}_{n} = ca^nc^{-1}$$

     Then, $c^{-1}ca^nc^{-1}c = c^{-1}ec \Rightarrow a^n = e$. Therefore, $|a| = |cac^{-1}|$.

- Case 3: ab and ba.

  Proof. Assume $(ab)^n = e$, that is,

  $$(ab)^n = \underbrace{(ab)(ab)\cdots(ab)}_{n} = e \Rightarrow a^{-1}\underbrace{(ab)(ab)\cdots(ab)}_{n}b^{-1} = a^{-1}eb^{-1}$$

  $$\Rightarrow \underbrace{(ba)(ba)\cdots(ba)}_{n-1} = a^{-1}eb^{-1} \Rightarrow \underbrace{(ba)(ba)\cdots(ba)(ba)}_{n} = a^{-1}eb^{-1}(ba) = e$$

  Therefore, $(ba)^n = e$, and vice versa. Therefore, $|ab| = |ba|$.

  Proof 2. Use the result of Case 2. Because

  $$ab = a(ba)a^{-1}$$

  from the result of Case 2, we have $|ab| = |ba|$.

- Case 4: abc, bca, cab.

  Proof. Use the result of Case 2. Because

  $$bca = a^{-1}(abc)a, \quad cab = c(abc)c^{-1}$$

  from the result of Case 2, we have $|abc| = |bca| = |cab|$.

4. Let G be a group, and an element $a \in G$ has the order n. Please prove $a^s = a^t \Leftrightarrow n|(s-t)$.

   Proof. Because $a^s = a^t$, we have

   $$a^s = a^t \Rightarrow a^s(a^{-t}) = a^t a^{-t} = e \Rightarrow a^{s-t} = e$$

   Therefore, $n|(s-t)$.

   On the other side, $n|(s-t) \Rightarrow (s-t) = n \cdot k$ for some k. Then, $a^{s-t} = a^{n\cdot k} = e$.

   $$a^{s-t} = e \Rightarrow a^{s-t}a^t = ea^t \Rightarrow a^s = a^t$$

2 Ring Problem

1. Let R be a ring with identity (denoted as 1). Prove R is also a ring with the identity under the operations $a \oplus b = a + b - 1$, $a \circ b = a + b - ab$.

   Proof.

   Under $\oplus$, R is a group. We easily check it is non-empty, closure, identity $=1$, and a's inverse is $2 - a$.

   Regarding Associativity,

   $$(a \oplus b) \oplus c = a \oplus (b \oplus c) = a + b + c - 2$$

Under $\circ$, Associativity

$$(a \circ b) \circ c = a \circ (b \circ c) = a + b + c - ab - ac - bc + abc$$

Also,

$$a \circ (b \oplus c) = (a \circ b) \oplus (a \circ c) = 2a + b + c - 1 - ab - ac$$

Similarly,

$$(a \oplus b) \circ c = (a \circ c) \oplus (b \circ c)$$

As a result, R for the operations $(\oplus, \circ)$ is a ring.

University of New Brunswick, Faculty of Computer Science

CS4355/6355: Topic 3 – Additional Note

1 Number Theory Problems

1. Let n be an integer than 1. Prove that 2^n is the sum of two odd consecutive integers.

   Proof. For the problem, the relation $2^n = (2k-1) + (2k+1)$ implies $k = 2^{n-2}$ and we obtain $2^n = (2^{n-1} - 1) + (2^{n-1} + 1)$.

   □

2. Let n be an integer than 1. Prove that 3^n is the sum of three consecutive integers.

   Proof. For this problem, the relation $3^n = (s-1) + s + (s+1)$ implies $s = 3^{n-1}$ and we obtain the representation $3^n = (3^{n-1} - 1) + 3^{n-1} + (3^{n-1} + 1)$.

   □

3. Prove that if x, y, z are integers such that $x^2 + y^2 = z^2$, then $xyz \equiv 0 \bmod 30$.

   Proof.

   - First, all three of x, y, z cannot be odd, since odd + odd = even. So xyz is even, i.e., $2|(xyz)$.
   - Second, $1^2 \equiv 2^2 \equiv 1 \bmod 3$, all perfect squares are 0 or 1 mod3. However, $x^2 + y^2 \equiv z^2 \bmod 3$ is not solved by making each of x^2, y^2, z^2 be 1 mod 3. Thus, one is 0 mod 3, and so xyz is divisible by 3, i.e., $3|(xyz)$

- Third, we have $1^2 \equiv 4^2 \equiv 1 \bmod 5$, and $2^2 \equiv 3^2 \equiv -1 \bmod 5$. So $x^2 + y^2 = z^2 \bmod 5$ can look like:

$$left\ side = \begin{cases} case1: & 1+1 = 2 \bmod 5 \\ case2: & 1+(-1) = 0 \bmod 5 \\ case3: & (-1)+1 = 0 \bmod 5 \\ case4: & (-1)+(-1) = -2 = 3 \bmod 5 \end{cases}$$

$$right\ side = \begin{cases} case1: & 1 \bmod 5 \\ case2: & -1 \bmod 5 \end{cases}$$

If none of x, y, z is 0 mod 5, the left side is NOT equal to the right side. Therefore, one of x, y, z is 0 mod 5, and xyz is divisible by 5, i.e., $5|(xyz)$.

Finally, because $2|(xyz)$, $3|(xyz)$, and $5|(xyz)$, we have $2 \cdot 3 \cdot 5|(xyz)$, i.e., $xyz \equiv 0 \bmod 30$.

□

University of New Brunswick, Faculty of Computer Science

CS4355/6355: Topic 3 – Additional Note

1 Number Theory Problems

1. If $p|10a - b$, $p|10c - d$, then $p|ad - bc$.

   Proof. From $p|10a - b$, we know $p \cdot k_1 = 10a - b$ for some k_1. Then,

   $$p \cdot k_1 \cdot c = (10a - b) \cdot c$$

   Let $k_2 = k_1 \cdot c$, we have $p \cdot k_2 = (10a - b) \cdot c = 10ac - bc$.
   Similarly, we have $p \cdot k_4 = (10c - d) \cdot a = 10ca - ad$ for some k_4. Then,

   $$p \cdot k_2 - p \cdot k_4 = 10ac - bc - 10ac + ad \Rightarrow p(k_2 - k_4) = ad - bc$$

   Therefore,

   $$p|ad - bc$$

   □

2. If n is odd, then $3|2^n + 1$

   Proof. Since $2 + 1 \equiv 0 \bmod 3$, we have $2 \equiv -1 \bmod 3$. Then,

   $$2^n \equiv (-1)^n \bmod 3$$

   Because n is odd, we have

   $$2^n - (-1)^n \equiv 0 \bmod 3 \Rightarrow 2^n + 1 \equiv 0 \bmod 3 \Rightarrow 3|2^n + 1$$

   □

3. $k = 0, 1, 2, \cdots$, for $n \in \mathbb{Z}$, we have $2n+1 | 1^{2k+1} + 2^{2k+1} + \cdots + (2n-1)^{2k+1} + 2n^{2k+1}$.

   Proof. For each $i = 1, 2, \cdots, n$, we have

   $$i + (2n+1) - i \equiv 2n + 1 \equiv 0 \bmod 2n+1$$

   $$i \equiv -((2n+1) - i) \bmod 2n+1 \Rightarrow i^{2k+1} \equiv [-((2n+1) - i)]^{2k+1} \bmod 2n+1$$

   Since $2k+1$ is odd, we have

   $$i^{2k+1} + ((2n+1) - i)^{2k+1} \equiv 0 \bmod 2n+1$$

   $$\sum_{i=1}^{n} [i^{2k+1} + ((2n+1) - i)^{2k+1}] \equiv 0 \bmod 2n+1$$

   Therefore,

   $$2n+1 | 1^{2k+1} + 2^{2k+1} + \cdots + (2n-1)^{2k+1} + 2n^{2k+1}$$

   □

4. If $m - p | mn + pq$, then $m - p | mq + np$

   Proof. Since

   $$(m-p) | (m-p)(n-q) \Rightarrow (m-p) | mn + pq - (mq + np)$$

   Because $m - p | mn + pq$, we have $m - p | mq + np$. □

5. If $x \equiv 1 \bmod m^k$, then $x^m \equiv 1 \bmod m^{k+1}$.

   Proof. Since $x \equiv 1 \bmod m^k$, we have $x = 1 + k \cdot m^k = 1 + (k \cdot m^{k-1}) \cdot m$, thus

   $$m^k | x - 1, \qquad x \equiv 1 \bmod m$$

   From $x \equiv 1 \bmod m$, we have $x^i \equiv 1^i \bmod m$ for $i = 0, 1, \cdots m-1$. Then,

   $$\sum_{i=0}^{m-1} x^i \equiv \sum_{i=0}^{m-1} 1^i \bmod m$$

   $$1 + x + x^2 + \cdots + x^{m-1} \equiv m \bmod m \Rightarrow 1 + x + x^2 + \cdots + x^{m-1} \equiv 0 \bmod m$$

   Then,

   $$m | (1 + x + x^2 + \cdots + x^{m-1})$$

   Finally, we have

   $$m^k \cdot m | (x-1)(1 + x + x^2 + \cdots + x^{m-1}) \Rightarrow m^{k+1} | x^m - 1 \Rightarrow x^m \equiv 1 \bmod m^{k+1}.$$

   □

University of New Brunswick, Faculty of Computer Science

CS4355/6355: Topic 3 – Additional Note

1 Number Theory Problems

1. Let n be a positive integer. Prove that $3^{2^n}+1$ is divisible by 2, but not by 4.

   Proof. **Method 1.** Clearly, 3^{2^n} is odd and $3^{2^n}+1$ is even. Note that $3^{2^n} = (3^2)^{2^{n-1}} = 9^{2^{n-1}} = (8+1)^{2^{n-1}}$. Recall the **Binomial theorem**

   $$(x+y)^m = x^m + \binom{m}{1}x^{m-1}y + \binom{m}{2}x^{m-2}y^2 + \cdots + \binom{m}{m-1}xy^{m-1} + y^m$$

   Setting $x = 8, y = 1$, and $m = 2^{n-1}$ in the above equation, we see that each summand besides the last (that is, $y^m = 1$) is a multiple of 8 (which is a multiple of 4). Hence the remainder of 3^{2^n} on dividing by 4 is equal to 1, and the remainder of $3^{2^n}+1$ on dividing by 4 is equal to 2. □

   Proof. **Method 2.** We have $3+1 \equiv 0 \bmod 4$, that is, $3 \equiv -1 \bmod 4$. Then, $3^{2^n} \equiv (-1)^{2^n} \bmod 4$, we have $3^{2^n} \equiv 1 \bmod 4$. As a result, $3^{2^n}+1 \equiv 2 \bmod 4$, the proof is completed.

   □

2. Let p be a prime number. Then $x^2 \equiv 1 \bmod p$ if and only if $x \equiv \pm 1 \bmod p$.

   Proof. If $x \equiv \pm 1 \bmod p$, we have $x^2 \equiv 1 \bmod p$. Conversely, if $x^2 \equiv 1 \bmod p$, then p divides $x^2 - 1 = (x-1)(x+1)$, and so p must divide $x-1$ or $x+1$. □

3. If p is prime, then $(p-1)! \equiv -1 \bmod p$.

Proof. If $p = 2$, $(p-1)! \equiv -1 \bmod p$ is true, since $1! \equiv -1 \bmod 2$.

If $p = 3$, $(p-1)! \equiv -1 \bmod p$ is also true, since $2! \equiv -1 \bmod 3$.

If p is prime ≥ 5, we know $(Z_p^*, *)$ is a group, where $Z_p^* = \{1, 2, 3, \cdots, p-1\}$ has total $p-1$ elements. Based on the Group theory, each element $a \in Z_p^*$ has its inverse $a^{-1} \in Z_p^*$ such that $a * a^{-1} \equiv 1 \bmod p$. Based on the Question 2 above, we know $a = a^{-1}$ if and only $a = 1$ or $a = p-1$. Therefore, we can partition the $p-3$ numbers in the set $\{2, 3, \cdots, p-2\}$ into $(p-3)/2$ pairs of integers $\{a_i, a_i^{-1}\}$ such that $a_i * {a_i}^{-1} \equiv 1 \bmod p$ for $i = 1, 2, \cdots, (p-3)/2$. Then,

$$(p-1)! \equiv 1 \cdot 2 \cdot 3 \cdots (p-2)(p-1) \equiv (p-1) \prod_{i=1}^{(p-3)/2} a_i a_i^{-1} \equiv p-1 \equiv -1 \bmod p.$$

□

4. Let $p \geq 7$ be a prime. Prove that the number

$$\underbrace{11\cdots1}_{p-1 \quad 1's}$$

is divisible by p.

Proof. We have

$$\underbrace{11\cdots1}_{p-1 \quad 1's} = \frac{10^{p-1}-1}{9}$$

and the conclusion follows from Fermat's Little Theorem. (Note also that $\gcd(10, p) = 1$.) □

CS 6355/4355: Cryptanalysis and Database Security

Topic 4: RSA Public Key Encryption

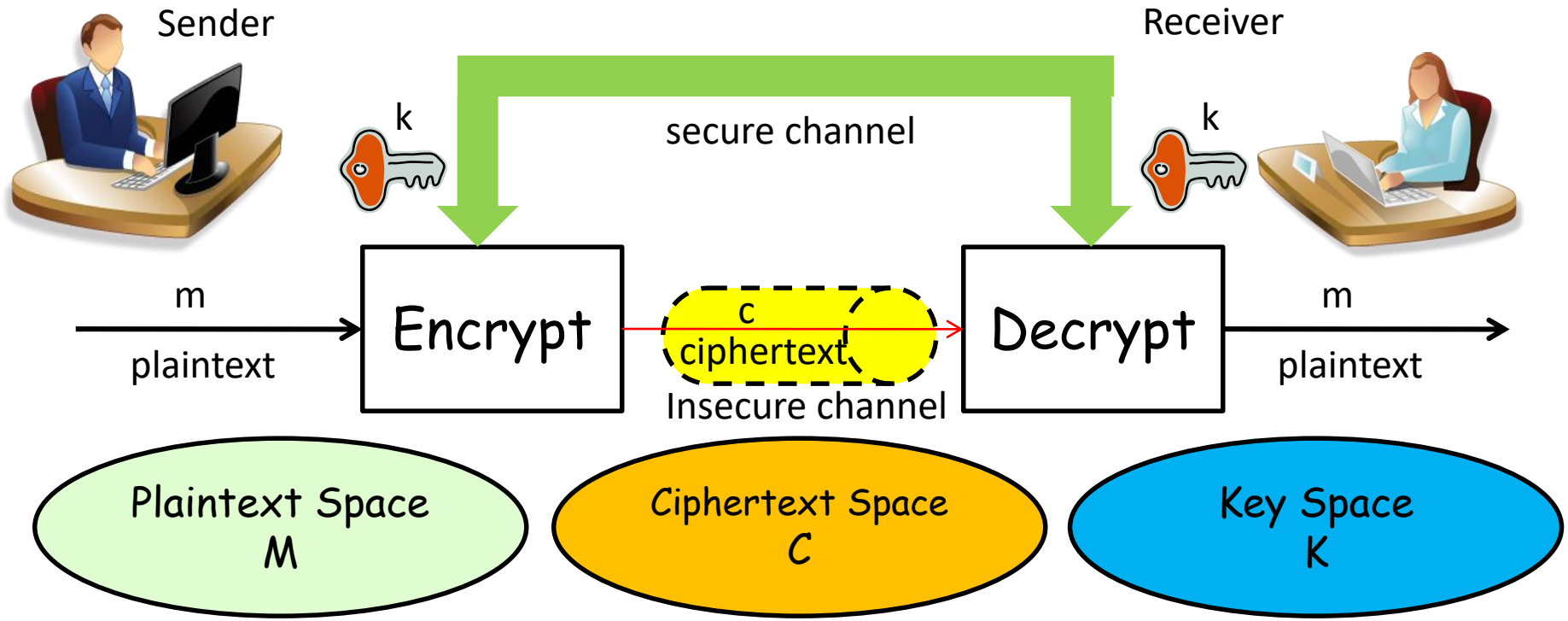
Lecturer: Rongxing LU

Email: RLU1@unb.ca Office: GE 114

Website: http://www.cs.unb.ca/~rlu1/

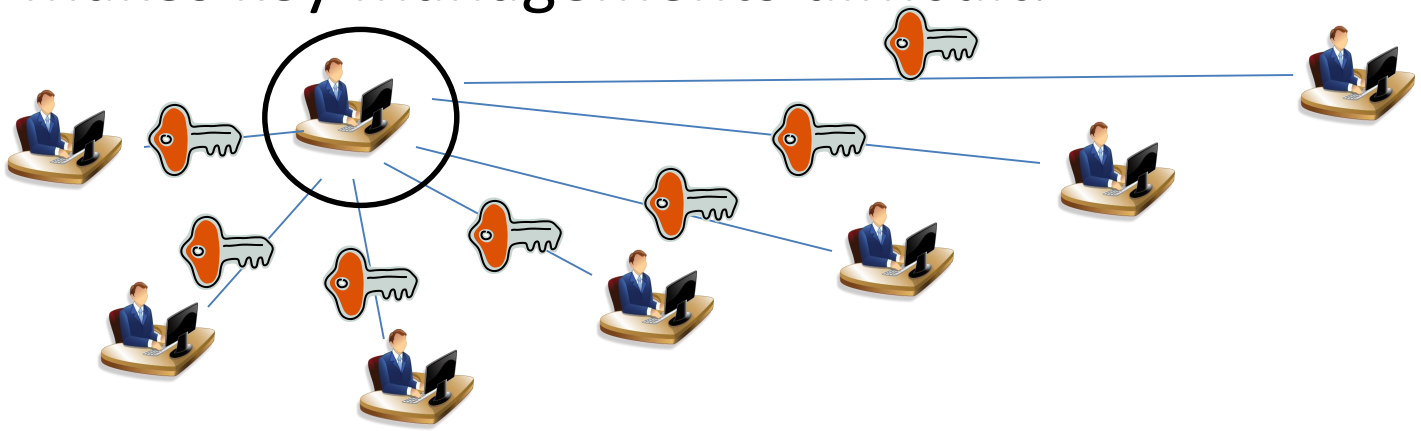
Faculty of Computer Science, University of New Brunswick

Review of Symmetric Cryptography

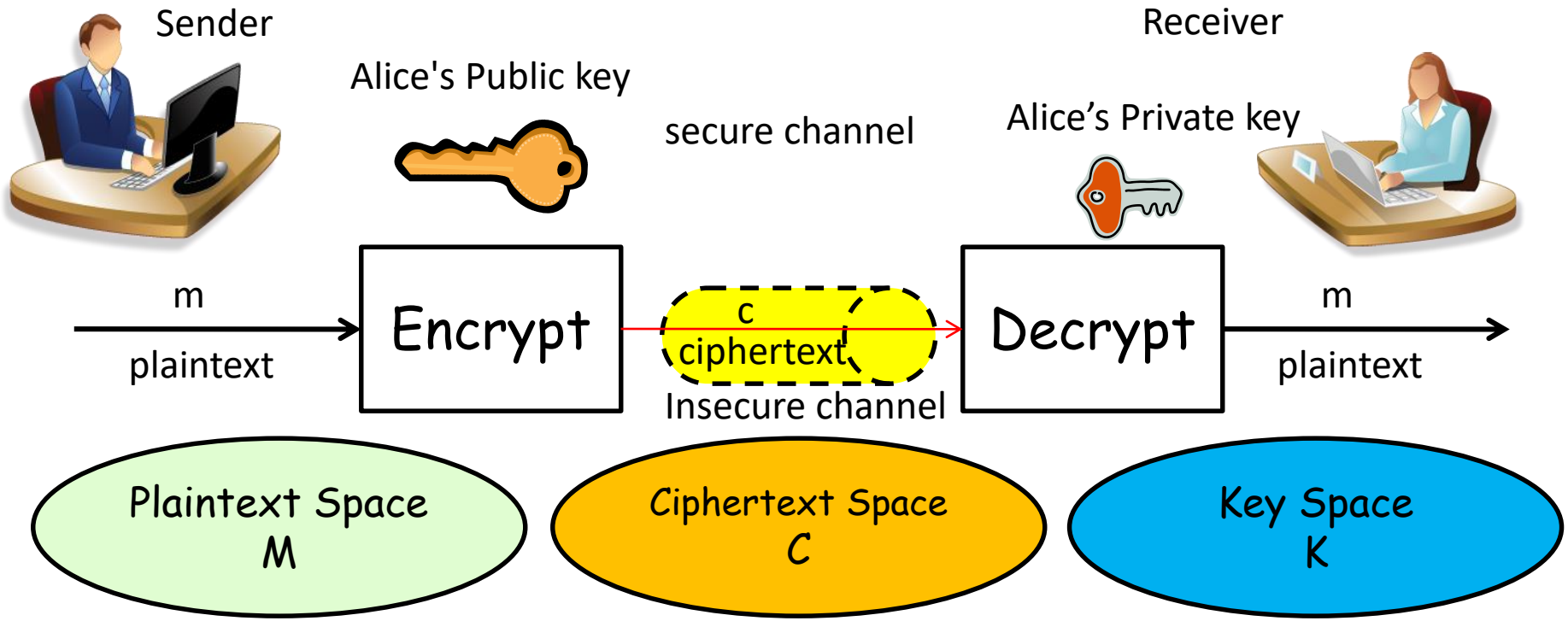


Disadvantages of Symmetric Cryptography

- The sender and receiver must share the same secret key. Key distribution is a must.
- If 1000 people want to communicate (two and two, in all possible ways), each must keep 999 secret keys, and the system requires a total of (999*1000)/2=499500 secret keys.
- This makes key managements difficult.

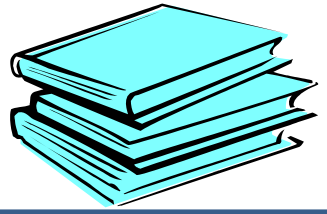


Asymmetric Cryptography



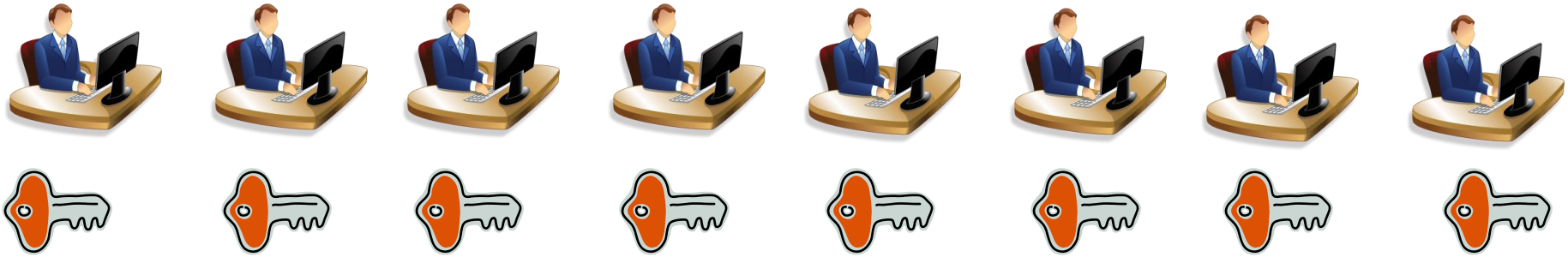
Asymmetric Cryptography

=>Public key Cryptography



Public key Directory

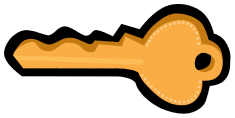
1. Simplify the key management
2. Make the digital signature possible



Public Key Cryptography

- **Public Key/Asymmetric** cryptography involves the use of **two** keys:

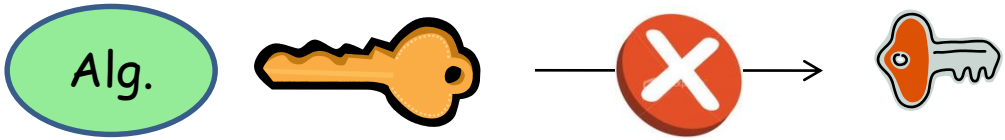
  - **public-key**, which may be known by anybody, and can be used to **encrypt messages**, and **verify signatures**



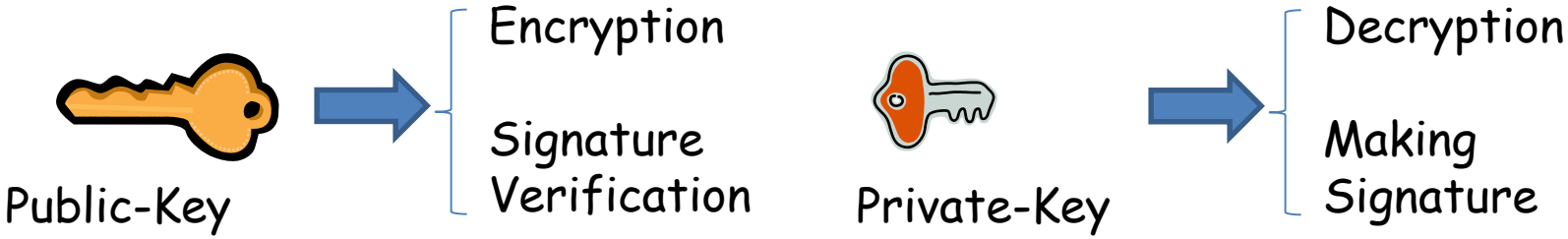
  - **private-key**, known only to the recipient, used to **decrypt messages**, and **sign** (create) **signatures**

Characteristics of Public Key

- Public-Key algorithms rely on two keys where:
  - ➢ it is computationally infeasible to find decryption key knowing only algorithm & encryption key

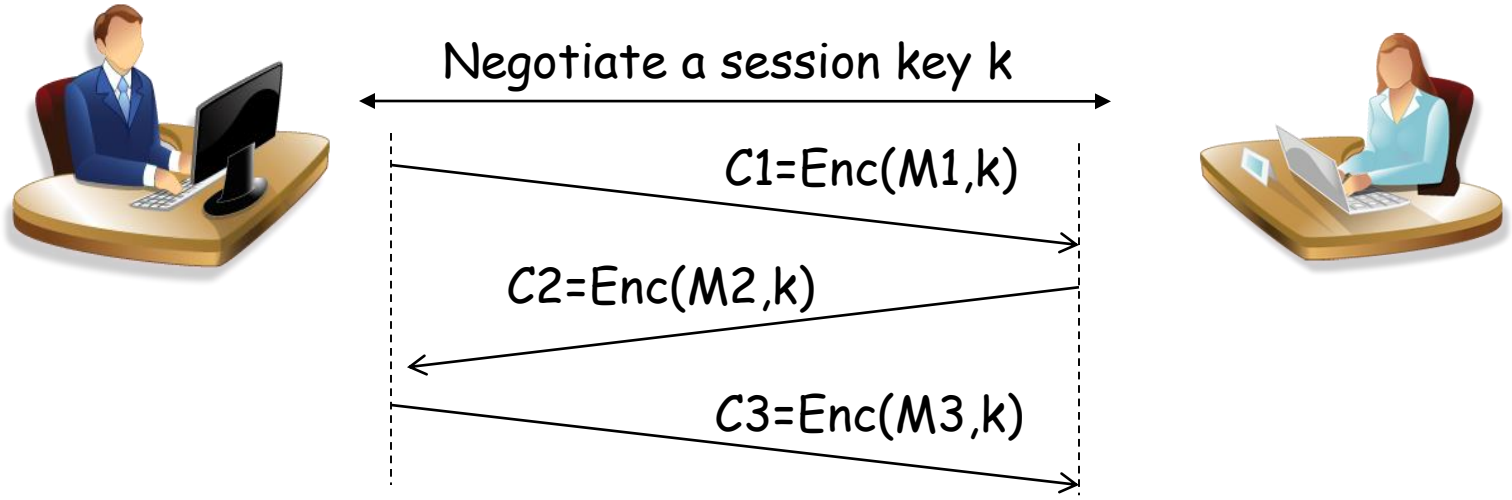


  - ➢ it is computationally easy to en/decrypt messages when the relevant (en/decrypt) key is known
  - ➢ either of the two related keys can be used for encryption, with the other used for decryption (for some algorithms)



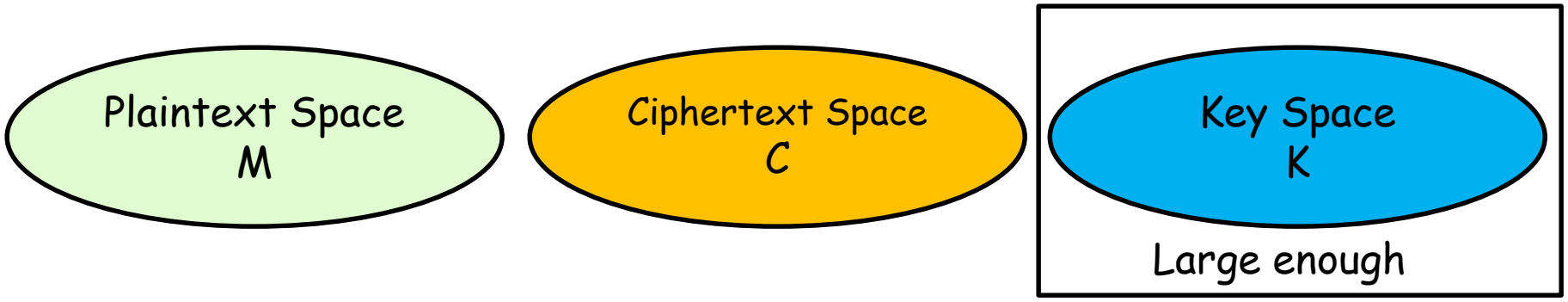
Applications of Public Key Cryptography

- Public-Key algorithms
  - ➢encryption/decryption (provide confidentiality)
  - ➢digital signatures (provide authentication)
  - ➢key exchange (of session keys)



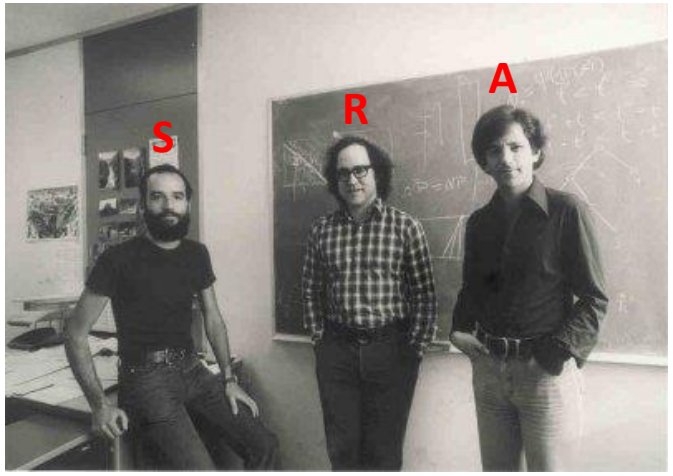
Security of Public Key Algorithms

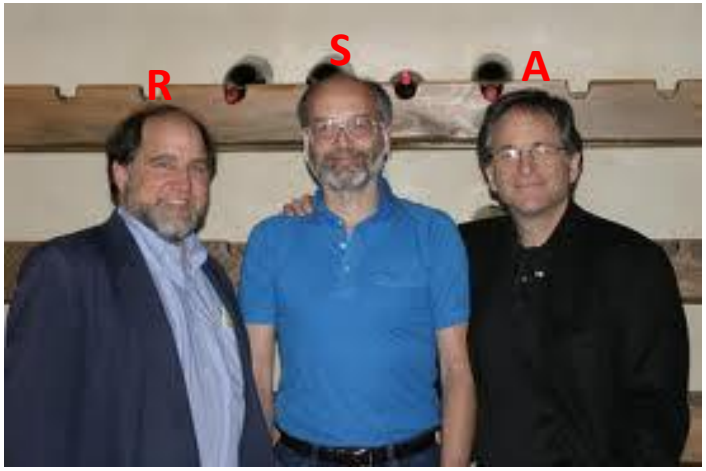
- keys used are too large (>512bits, 1024 bits, ...)
- security relies on the known hard problem, it is hard enough to be impractical to break
- requires the use of very large numbers
- hence is slow compared to private key schemes



Introduction of RSA

- Mathematic Background of RSA
- RSA cryptosystem
- RSA cryptanalysis





P & Q PRIME
$N = PQ$
$ED \equiv 1 \bmod (P-1)(Q-1)$
$C = M^E \bmod N$
$M = C^D \bmod N$
RSA Algorithm

https://www.youtube.com/watch?v=f_8s451zYYE

Fermat's Little Theorem

- $a^{p-1} = 1 \pmod p$
  - where p is prime and gcd(a,p)=1

Proof

- List all elements < p, e.g., a set A={1,2,3,…, p-1}
- Choose any element a < p, and multiple a with all elements in the set A, i.e., a set B = {a*1 mod p, a*2 mod p, a*3 mod p, …., a*(p-1) mod p}
- Suppose two element (r, s) in A such that r≠s, but a*r = a*s mod p. In this case, we have a(r - s)= 0 mod p. Because gcd(a, p)=1, we have (r - s)= 0 mod p, i.e., r=s, which contradicts with r≠s. Therefore, the set B also includes p-1 elements, which are less than p.

Fermat's Little Theorem

- $a^{p-1} = 1 \pmod p$
  - where p is prime and gcd(a,p)=1

Proof

- Now, since both the set A= {1,2,3,…, p-1} and the set B = {a*1 mod p, a*2 mod p, a*3 mod p, …., a*(p-1) mod p} contains p-1 elements less than p, we can multiple all elements in A and B as follows,

  $1*2*3*\ldots*(p-1) = \underline{a*1} * \underline{a*2} * \underline{a*3} * \ldots * \underline{a*(p-1)} \bmod p$
- $1*2*3*\ldots*(p-1) = 1*2*3*\ldots*(p-1) * \underline{a^{p-1}} \bmod p$
- $1 = a^{p-1} \bmod p$ ■

Exercises

$4^6 \bmod 7 = ?$

$19^{22} \bmod 23 = ?$

$89^{100} \bmod 101 = ?$

$4^6 \bmod 7 = 1$

$19^{22} \bmod 23 = 1$

$89^{100} \bmod 101 = 1$

$4^{37} \bmod 7 = ?$

$19^{45} \bmod 23 = ?$

$89^{1001} \bmod 101 = ?$

$$4^{37} \bmod 7 = 4^{6\times 6+1} \bmod 7 = (4^6)^6 \cdot 4 \bmod 7 = 1^6 \cdot 4 \bmod 7 = 4$$

$$19^{45} \bmod 23 = 19^{22\times 2+1} \bmod 23 = (19^{22})^2 \cdot 19 \bmod 23 = 19$$

$$89^{1001} \bmod 101 = 89^{100\times 10+1} \bmod 101 = (89^{100})^{10} \cdot 89 \bmod 101 = 89$$

For a large prime p, and any element a, $\gcd(a, p) = 1$,

we have $a^{p-1} \bmod p = 1, a^{(p-1)\cdot k+1} \bmod p = a, a^p \bmod p = a$

Euler's Theorem

- If n and a are coprime positive integers, i.e., gcd(a, n)=1, then $a^{\varphi(n)} = 1 \pmod n$, where φ(n) is Euler's totient function.
  - for p (p prime) $\varphi(p) = p-1$
  - for p.q (p,q prime)

    $\varphi(pq) = \varphi(p) \times \varphi(q) = (p-1) \times (q-1)$

  - eg.

    $\varphi(37) = 36$, $\varphi(11) = 10$

    $\varphi(21) = (3-1) \times (7-1) = 2 \times 6 = 12$,

    $\varphi(10) = (2-1) \times (5-1) = 1 \times 4 = 4$ {1,3,7,9}

Euler's Theorem

- φ(n) denotes the number of positive integers less than n which are relatively prime to n, where n=pq
- Example to show why

$$\varphi(pq) = \varphi(p) \times \varphi(q) = (p-1) \times (q-1)$$

- p=7, q=5, n=pq=7*5=35

0	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	32	33	34

gcd(a,35)=1

a= 1,2,3,4,6,8,9,11,12,13,16,17,18,19,22,23, 24,26,27,29,31,32,33,34

p=7, 7*x, x={1,2,3,4}, (q-1) elements gcd(7*x,35)=7
q=5, 5*y, y={1,2,3,4,5,6}, (p-1) elements gcd(5*y,35)=5

φ(n)=p*q-1- (q-1) -(p-1)=(p-1)(q-1) φ(35)=35-1- 4-6=24

Euler's Theorem

Proof of $a^{\varphi(n)} = 1 \pmod n$, where $n=pq$

- First, $\varphi(n)$ denotes the number of positive integers less than n which are relatively prime to n
- Let $A=\{r_1, r_2, \ldots r_{\varphi(n)}\}$ be the set of $\varphi(n)$ integers less than n and relatively prime to n, where any $r_i \neq r_j$.
- For any element a in A, gcd(a, n)=1, then, $a*r_1 \bmod n$, $a*r_2 \bmod n$, ... $a*r_{\varphi(n)} \bmod n$ are distinct, otherwise if $r_i*a = r_j*a \pmod n$ then $r_i = r_j \pmod m$ (Contradiction).
- Since each r_i*a is relatively prime to n and there are only $\varphi(n)$ distinct numbers relatively prime to n, therefore, the set $B=\{a*r_1 \bmod n, a*r_2 \bmod n, \ldots a*r_{\varphi(n)} \bmod n\}$ is identical to the set $A=\{r_1, r_2, \ldots r_{\varphi(n)}\}$.

Euler's Theorem

Proof of $a^{\varphi(n)} = 1 \pmod n$, where $n=pq$

- Since the set $B=\{a*r_1 \bmod n,\ a*r_2 \bmod n, \ldots a*r_{\varphi(n)} \bmod n\}$ is identical to the set $A=\{r_1, r_2, \ldots r_{\varphi(n)}\}$.

$$r_1 * r_2 * \ldots * r_{\varphi(n)} = \underline{a*r_1} * \underline{a*r_2} * \ldots * \underline{a*r_{\varphi(n)}} \bmod n$$

- $r_1 * r_2 * \ldots * r_{\varphi(n)} = r_1 * r_2 * \ldots * r_{\varphi(n)} * \underline{a^{\varphi(n)}} \bmod n$
- $1 = \underline{a^{\varphi(n)}} \bmod n$ ■

Exercise

$$25^{120} \bmod 143 = ?$$

$$19^{60} \bmod 77 = ?$$

$$25^{120} \bmod 143 = 25^{(11-1)(13-1)} \bmod 11\times 13 = 1$$

$$19^{60} \bmod 77 = 19^{(7-1)(11-1)} \bmod 7\times 11 = 1$$

Exercises

$$25^{1201} \bmod 143 = ?$$

$$19^{481} \bmod 77 = ?$$

$$25^{1201} \bmod 143 = 25^{120\times 10+1} \bmod 11\times 13 = (25^{120})^{10} \cdot 25 \bmod 11\times 13 = 25$$

$$19^{481} \bmod 77 = 19^{60\times 8+1} \bmod 7\times 11 = (19^{60})^{8} \cdot 19 \bmod 7\times 11 = 19$$

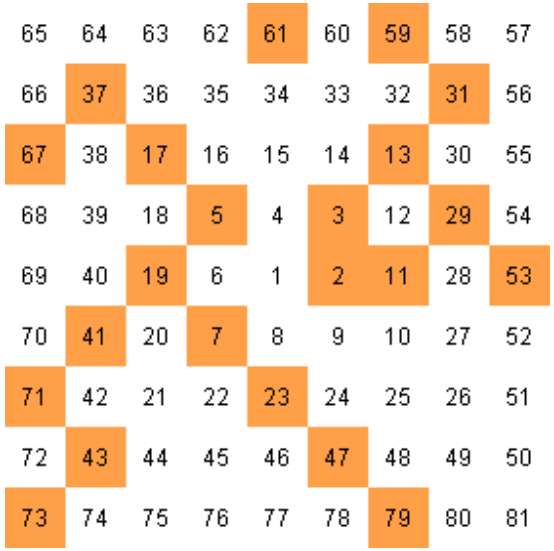
For a large number $n = pq$, where p, q are primes,

and any element $a, \gcd(a, n) = 1,$

we have $a^{\varphi(n)} \bmod n = 1, a^{\varphi(n)\cdot k+1} \bmod n = a, a^{\varphi(n)+1} \bmod n = a$

Testing for Primality

- For many cryptographic algorithms including RSA, it is necessary to select one or more very large prime numbers at random. Thus we are faced with the task of determining whether a given large number is prime. There is no simple yet efficient means of accomplishing this task.



Testing for Primality

Proof: "There are Infinitely Many Primes"

- Assume that the primes are finite, and we can list them as $L=\{p_1, p_2, \ldots, p_r\}$
- Let p be any common multiple of these primes plus one, i.e., $p= p_1 * p_2 * \ldots * p_r+1$. Then, p is either a prime or not.
- If p is a prime, then p is a prime that was not in L.
- If p is not a prime, then p is divisible by some prime, call p_0
- Here, p_0 can not be any of $p_1, p_2, \ldots, p_r$, otherwise p_0 would divide 1, which is impossible. So this prime p_0 is some prime that was not in list $L=\{p_1, p_2, \ldots, p_r\}$. In other words, the $L=\{p_1, p_2, \ldots, p_r\}$ was incomplete, and there are infinite many primes. ■

Testing for Primality

Naive Try

```
boolean isPrime(integer n)
        if ( n < 2 ) return false
        for(i = 2 to n -1)
           if( i |n )  // "i divides n"
              return false
        return true
```

Question: What is the running time of this algorithm?

Answer: Assuming divisibility testing is a basic operation – so O (1) (this is an invalid assumption)– then above primality testing algorithm is O (n).

Testing for Primality

Second Try

- Don't try number bigger than n/2.
- After trying 2, don't try any other even numbers, because know n is known odd if 2ł n by trying 2.
- In general, try only smaller prime numbers.
- In fact, only need to try to divide by prime numbers no larger than √n

```
boolean isPrime(integer n)
        if ( n < 2 ) return false
        for  each prime number i no larger than √n
          if( i |n )     // "i divides n"
             return false
        return true
```

Testing for Primality

Theorem:
If n is a composite, then its smallest prime factor is $\leq \sqrt{n}$

Proof. (By contradiction).
Suppose the smallest prime factor is $> \sqrt{n}$ .Then we can decompose $n = p*q*x$ where p and q are primes $> \sqrt{n}$ and x is some integer. Therefore

$$n = p*q*x > \sqrt{n}* \sqrt{n}*x=nx,$$

which implies that $n>n$.
Therefore, it shows the supposition (the smallest prime factor is $> \sqrt{n}$) was false.
As a result, the theorem is proved.

```
boolean isPrime(integer n)
        if ( n < 2 ) return false
        for  each prime number i no larger than √n
          if( i |n )          // "i divides n"
            return false
        return true
```

Running time < O (√n)

Testing for Primality

Examples. Test if 139 and 143 are primes.

List all primes up to $\sqrt{n}$ and check if they divide the numbers.

{2,3,5,7,11}, since $13^2 = 169 > 143$

	139	143
2	2 ł 139	2 ł 143
3	3 ł 139	3 ł 143
5	5 ł 139	5 ł 143
7	7 ł 139	7 ł 143
11	11 ł 139	11 | 143

11*13=143

Therefore, 139 is a prime number, but 143 is composite.

Testing for Primality

Direct application of Fermat's Little Theorem for testing primality of n $a^{p-1} = 1 \pmod{p}$

If we want to test if p is prime, then we can pick random a's in the interval and see if the equality holds. If the equality does not hold for a value of a, then p is composite. If the equality does hold for many values of a, then we can say that p is probably prime.

Inputs: n: a value to test for primality;
k: a parameter that determines the number of times to test for primality
Output: composite if n is composite, otherwise probably prime
repeat k times:
pick a randomly in the range (1, n – 1]
if $a^{n-1} \neq 1 \pmod{n}$, then return composite
return probably prime

Testing for Primality

Example.

For n = 221, we test whether it is a prime.

1. We randomly pick $1 \le a < 221$, say a = 38. Check

$$a^{n-1} = 38^{220} = 1 \bmod 221$$

2. From the first test, 221 could be a prime.
3. We take another a =26 for test

$$a^{n-1} = 26^{220} = 169 \neq 1 \bmod 221$$

4. Therefore, 221 is not a prime.
   (221=17*13)

repeat k times

Testing for Primality

Miller-Rabin algorithm

Observation: Given an odd integer n > 2, n - 1 can be expressed as $n - 1 = 2^k q$, with $k > 0$, and q odd.

Fact. If n is a prime number, for some a < n, gcd(a, n)=1, we have $a^{n-1} = 1 \bmod n$. Then,

$$a^{n-1} - 1 = 0 \bmod n \Rightarrow a^{2^k q} - 1 = 0 \bmod n$$

$$a^{2^k q} - 1 = (a^{2^{k-1} q} - 1)(a^{2^{k-1} q} + 1) = 0 \bmod n$$

And we have $a^{2^{k-1} q} - 1 = 0 \bmod n$ $\quad$ $a^{2^{k-1} q} + 1 = 0 \bmod n$

$$a^{2^{k-1} q} = (a^q)^{2^{k-1}} = 1 \bmod n$$

$$a^q = 1 \bmod n$$

$$a^{2^{k-1} q} = -1 = n - 1 \bmod n$$

Testing for Primality

Therefore, if n is a prime number, we have

$$a^q = 1 \bmod n \qquad a^{2^{k-1}q} = -1 = n-1 \bmod n$$

But if $a^q = 1 \bmod n \qquad a^{2^{k-1}q} = -1 = n-1 \bmod n$

we can say n is probably prime.

If $a^q \neq 1 \bmod n \qquad a^{2^{k-1}q} \neq -1 = n-1 \bmod n$

we can confirm n is NOT a prime number

MILLER-RABIN(n) Repeat for Primality Testing

1. find integers $k > 0$, q odd, so that $n - 1 = 2^k q$
2. select a random integer a, $1 < a < n - 1$
3. **if** $a^q \bmod n \equiv 1$ **then return** "probably prime"
4. **for** $j = 0$ **to** $k - 1$ **do**
5. **if** $a^{2^j q} \bmod n \equiv n-1$ **then return** "probably prime"
6. **return** "composite"

Testing for Primality

Determine If n = 221 is prime. We can write

$$n - 1 = 220 = 2^2 \cdot 55$$

We have k = 2 and q = 55. Randomly select a number a such that a < n, e.g., 174. Then,

$$a^{2^0 \cdot q} \bmod n = 174^{55} \bmod 221 = 47 \neq 1, n-1$$

$$a^{2^1 \cdot q} \bmod n = 174^{110} \bmod 221 = 220 = n-1$$

Since $220 \equiv -1 \bmod n$, 221 is probably prime. We try another random a, this time choosing a=137:

$$a^{2^0 \cdot q} \bmod n = 137^{55} \bmod 221 = 88 \neq 1, n-1$$

$$a^{2^1 \cdot q} \bmod n = 137^{110} \bmod 221 = 205 \neq n-1$$

Hence 221 is not a prime number. Actually, 221=13*17

repeat k times

Integer Factorization Problem

- Input:
  - N: odd composite integer with at least two distinct prime factors. For example, N=pq.
- Output:
  - Primes p and q.
- Examples.
  - 21=7×3, 77=7×11, 143 =11×13, 483=21×23
  - 11797055816053=3554179×3319207
  - 6962917758703=1836059×3792317

Integer Factorization Problem

- Examples.
  - 21=7×3, 77=7×11, 143 =11×13, 483=21×23
  - 11797055816053=3554179×3319207
  - 6962917758703=1836059×3792317
  - Given p, q, to compute N=pq (easy)
  - Given N=pq, to compute p or q
    - if N is small, easy
    - else if N is large, difficult

The RSA Challenge Numbers

- N=pq
  - RSA-576
  - Decimal Digits: 174
    - N=188198812920607963838697239461650439807163563379417382700763356422988859715234665485319060606504743045317388011303396716199692321205734031879550656996221305168759307650257059

The RSA Challenge Numbers

- RSA-576
  - On December 3, 2003, a team of researchers reported a successful factorization of the RSA-576
  - Decimal Digits: 174
    - P=3980750864240649373971255005503864911990643623425267084063851895759463889572617685833 17
    - q=4727721461074353025362230719730482246329146953020971164598521711305207112563635903975 27

The RSA Challenge Numbers

- RSA-896 (Not factored)
  - Decimal Digits: 270
    - N=412023436986659543855531365332575948179811699844327982845455626433876445565248426198098870423161841879261420247188869492560931776375033421130982397485150944909106910269861031862704114880866970564902903653658867433731720813104105190864254793282601391257624033946373269391

RSA Cryptosystem

- p and q are large primes.
- $n = p.q$
- $\varphi(n) = (p - 1)(q - 1)$
- e is an integer, $1 < e < \varphi(n)$, $gcd(e, \varphi(n)) = 1$
- d is an integer, $1 < d < \varphi$, $e.d \equiv 1 \pmod{\varphi}$
  (d is inverse of e in modulo $\varphi(n)$, calculated by the extended Euclidean algorithm)
- (n, e) is public key
- (n, d) is private key

RSA Cryptosystem (Cont.)

- Encryption : $c=m^e \bmod n$
- Decryption : $m=c^d \bmod n$
- m is plaintext, c is ciphertext.

$$m = c^d \bmod n = m^{ed} \bmod n$$
$$= m^{1+k\varphi(n)} \bmod n = m \cdot (m^{\varphi(n)})^k \bmod n$$
$$= m \cdot 1 \bmod n = m$$

RSA Example

$$PK : (n, e)$$
$$SK : d$$

$$m = 15$$

$$C = m^e \bmod n = 15^{13} \bmod 77 = 64$$

$$C$$

$$p = 7, q = 11, n = pq = 7 \times 11 = 77$$
$$\phi(n) = (p-1)(q-1) = 6 \times 10 = 60$$
$$e = 13, d = ?$$
$$\because \gcd(60,13) = 1$$

$$m = C^d \bmod n = 64^{37} \bmod 77 = 15$$
$$= 15^{13 \times 37} \bmod 77 = 15^{1+8 \times 60} \bmod 77 = 15$$

Dividend	Divisor	Quotient	Remainder
ϕ(n) =60	= e=13	× 4	+ 8
e=13	= 8	× 1	+ 5
8	= 5	× 1	+ 3
5	= 3	× 1	+ 2
3	= 2	× 1	+ 1

1 = 3 - 2× 1

1 = 3 – (5 - 3×1)× 1 = 3×2 - 5×1

1 = (8 – 5×1)×2 – 5×1 = 8 × 2 - 5×3

1 = 8 × 2 – (13 – 8 × 1) ×3 = 8 × 5 – 13 × 3

1 = (60 - 13×4) × 5 –13 × 3 = 60×5 – 13×23

1 = 60×5 – 13×23 mod 60 = – 13×23 mod 60

1 = 13×(-23) = 13 × 37 mod 60

$d = 37$

Since 37+23 = 60 mod 60, 37 = -23 mod 60

Security of RSA Cryptosystem

- Since n and e are in public key, in order to recover $m=c^d \bmod n$ from c, we need to know the private key d.
- While in order to compute d, we need to factorize $n=p.q$ to compute $\varphi(n)=(p-1)(q-1)$.
- Therefore, the security of RSA cryptosystem is dependent upon the hardness of factor large integer.
- Currently, 1024 bit-RSA is commonly used.
- For other applications, like military application, 4096 bit-RSA is prefered.

The difference value |p-q| should be large

Why

- If n (>0) is an odd integer n and written as n=x*y. Then, n can be also written as $u^2 - v^2$, where u=(x+y)/2, v=(x-y)/2. If x>y, both u and v are >0.
- If n=x*y, and x, y are close, then v is a small number, and u is a little larger than $\sqrt{n}$.
- In this case, we can guess u by trying ($\sqrt{n}+1$, $\sqrt{n}+2$, $\sqrt{n}+3$, …) to satisfy $u^2-n=v^2$.
- Since u is a little larger than $\sqrt{n}$, we can quickly determine u, and then v.
- Finally, we can factor n=x*y as x=u+v, y=u-v.

Example

- If p=401, q=409, then n=p*q=164009
- We can calculate $\sqrt{n}$=404.98≈405.
- Since $405^2-n= 4^2$, (405^2=164025)
- We can determine v=4, and then u=405.
- Finally x=u+v=409, y=u-v=401

- Therefore, the difference value |p-q| should be large, but p, q should be of the same length. |p| ≈|q|.

The modulus n=p*q can not be common in the system

Why

- Suppose two entities share the common modulus n=p*q, but have different public-private key pairs (e1,d1) and (e2, d2) with gcd(e1,e2)=1. If the same message m is encrypted with $c1=m^{e1} \bmod n$, $c2=m^{e2} \bmod n$, then the message m can be recovered without knowing the private keys.
- Since gcd(e1,e2)=1, we can use the extended Euclidean algorithm to calculate r, s such that r*e1+s*e2=1.
- Then, $c1^r * c2^s = m^{r*e1+s*e2} = m \bmod n$, we can get m
- If r<0, we should calculate $(c1^{-1})^{-r} * c2^s = m \bmod n$

Example

- If p=7, q=11, n=p*q=77, φ(n)=(p-1)(q-1)=60.
- (e1=13, d1=37), (e2=17, d2=53) extended Euclidean algorithm
- For m=15, $c1=m^{e1} \bmod n=15^{13} \bmod 77=64$, $c2=m^{e2} \bmod n=15^{17} \bmod 77=71$
- Since gcd(e1,e2)=1, we have 4*13 + (-3)*17 = 1 extended Euclidean algorithm
- Because s=-3, we compute $(c2)^{-1} \bmod 77=-13 \bmod 77=64$ extended Euclidean algorithm −13*71 + 12*77 = 1
- $c1^r*(c2^{-1})^{-s}=64^4*64^3 \bmod 77= 15$

common modulus attack

Example (2)

- If p=23, q=29, n=p*q=667, φ(n)=(p-1)(q-1)=616.
- (e1=13, d1=273), (e2=41, d2=601) extended Euclidean algorithm
- For m=20, $c1=m^{e1} \bmod n=20^{13} \bmod 667=451$, $c2=m^{e2} \bmod n=20^{41} \bmod 667=132$
- Since gcd(e1,e2)=1, we have 19*13 + (-6)*41 = 1 extended Euclidean algorithm
- Because s=-6, we compute $(c2)^{-1} \bmod 667=-96 \bmod 667=571$ extended Euclidean algorithm -96*132 + 19*667 = 1
- $c1^r*(c2^{-1})^{-s}=451^{19}*571^{6} \bmod 667=$ 20

common modulus attack

The padding is required in RSA

$PK : (n, e)$

$SK : d$

$m = \text{"hello world"}$

$C = m^e \bmod n$ C C'

$R = r^e \bmod n$

$C' = C \cdot R \bmod n$

$C'^d \bmod n = m \cdot r \bmod n$

Since m is meaningful, If the recovered is m*r, which becomes meaningless. Then, the receiver can detect it.

However, if m is a session key (a random string), the receiver cannot identify the recovered m*r is correct or not.

The padding is required (Cont.)

m

$$C = (m \| \underbrace{H(m)}_{l-bits})^e \bmod n$$

C

$PK : (n, e)$

$SK : d$

C'

$$C' = C \cdot R \bmod n$$

$$C'^d \bmod n = a \| \underbrace{b}_{l-bits}$$

check $H(a) = b$

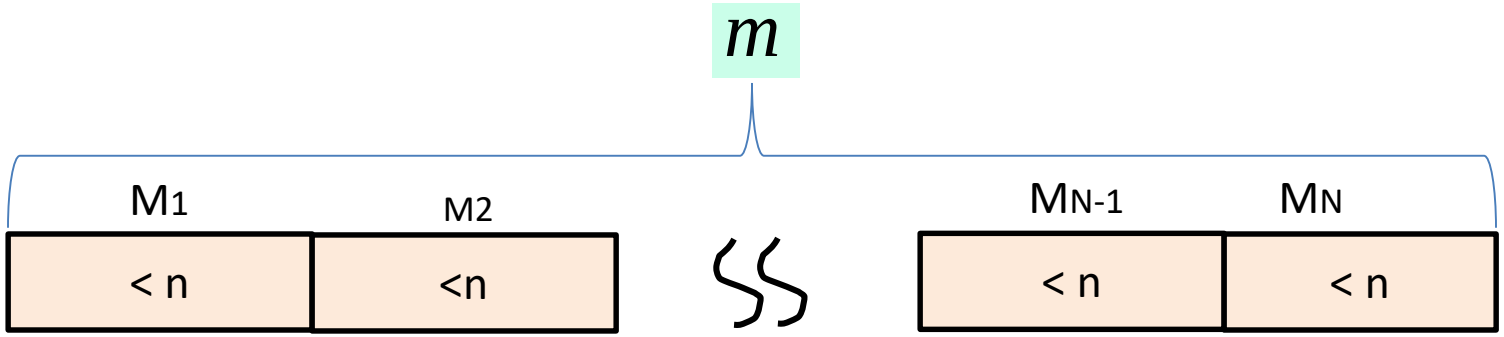
With the padding, the receiver can detect whether the message has Modified during the transmission.

How to encrypt a long message

m

$PK : (n, e)$

$SK : d$



- If m is large than n, then m should be first divided into m=m1m2m3…, each mi < n.
- However, the efficiency is big problem.

How to encrypt a long message (Cont.)

Encryption algorithm	Comparison1	Comparison 2
Symmetric Encryption Algorithm, e.g., AES	fast	Need sharing a secret key in advance
RSA Algorithm	Slow, usually 100-1000 times slower than AES, (usually used for encrypting small message)	Knowing the receiver's public key, do not need sharing a secret key in advance

no shared key

$\mathrm{pk}\ (n, e)$

$\mathrm{sk}\ d$

Discussion 1: how to send a short text message m (<1k) secretly and efficiently?
Discussion 2: how to send a file M of large size secretly and efficiently?

$$c = m^e \bmod n$$

$$\xrightarrow{c}$$

$$m = c^d \bmod n$$

choose a key k

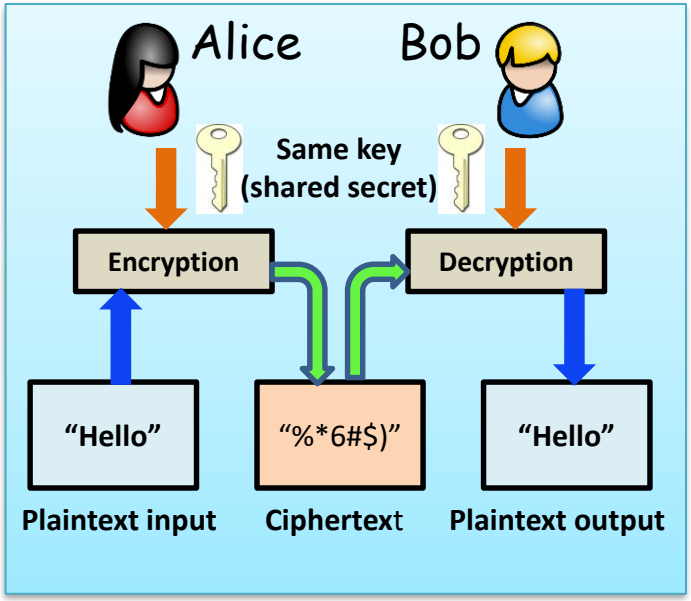
$$c_1 = k^e \bmod n$$

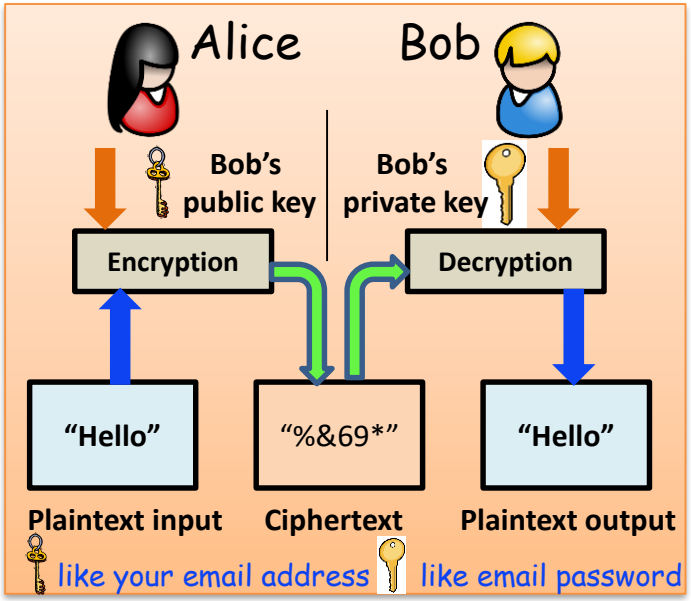
$$C_2 \leftarrow \mathrm{AES}(M, k) \xrightarrow{(c_1, C_2)}$$

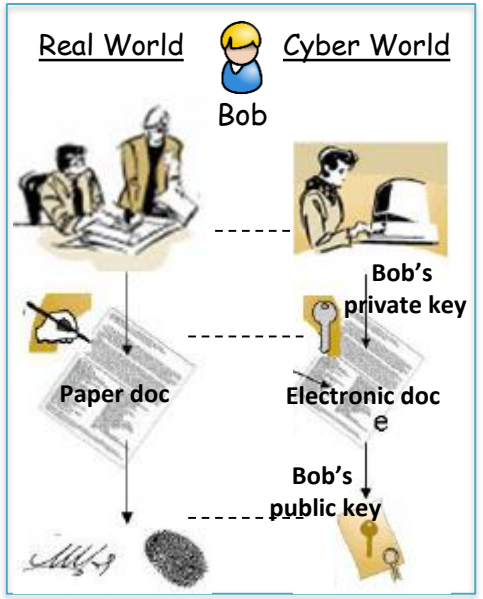
$$k = c_1^{\,d} \bmod n$$

$$M \leftarrow \mathrm{AES}(C_2, k)$$

Comparison Symmetric Encryption and Public Key Encryption







Symmetric Encryption (AES, DES, …)

- Fast
- Need share the same key
- Achieve Confidentiality

Public key Encryption (RSA

- Slow (due to exponential operation)
- Do not need share the same key
- Achieve Confidentiality

Digital Signature

+ Use private key to sign
+ Use public key to verify
+ Achieve authentication, data integrity, non repudiation


Thank
you

University of New Brunswick, Faculty of Computer Science

CS4355/6355: Topic 4 – Additional Note

1 The cycling attack

The cycling attack was one of the first attacks on RSA [1]. As the name of this attack suggests, the way this attack works is by repeatedly encrypting the ciphertext. When an attacker gets $c \equiv m^e \bmod n$, he will encrypt the ciphertext with the public key and this will lead him to, eventually getting an encryption which will be the original ciphertext. That is, after l encryptions, he will have

$$c^{e^l} \equiv c \equiv m^e \bmod n$$

so he will know that the previous encryption is the original plaintext, that is,

$$c^{e^{l-1}} \equiv m \bmod n$$

The value l is called the recovery exponent for the plaintext m. Suppose a plaintext m is encrypted with the public key (e, n), the recovery exponent of m divides $\phi(\phi(n))$. Because $e \in Z^*_{\phi(n)}$, we have $e^{\phi(\phi(n))} \equiv 1 \bmod n$. If $ord(e) = l$, we have $l|\phi(\phi(n))$. We need to choose e with a larger l.

2 Pollard's ρ algorithm

Pollard's ρ algorithm, described by Pollard in 1975 [2], is to find a small factor p of a given integer N. The simplified version of this algorithm is described as follows.
Algorithm: Pollard's ρ Algorithm: Given a composite $N = pq$:

1. set $a = 2, b = 2$.
2. Define the modular polynomial $f(x) = (x^2 + c) \bmod N$, with $c \neq 0, -2$
3. For $i = 1, 2, \cdots$ do:
   a) Compute $a = f(a), b = f(f(b))$.
   b) Compute $d = (a - b, N)$.
   c) If $1 < d < N$, then return d with success.
   d) If $d = N$, then terminate the algorithm with failure.

The function f is used to create two pseudo random sequences on $\mathbb{Z}_N$. The reason for this is that, picking randomly two numbers $x, y \in \mathbb{Z}_N$, there is a probability of 0.5 that after $1.777\sqrt{p}$ tries, one will be congruent modulo p. If they are $a \neq b$, then $(a - b, N)$ yields a factor of N [3]. Concretely,

$$a, b \in \mathbb{Z}_N \to a' = a \bmod p, b' = b \bmod p \to a', b' \in \mathbb{Z}_p^*$$

After $1.777\sqrt{p}$ tries, we may have $a' = b' \bmod p$, which also shows $a = b \bmod p$. Then, we have $p|(a - b)$, and $\gcd(a - b, N) = p$.
The runtime of the algorithm is $O(\sqrt{p})$, where p is N's smallest prime factor [4]. This means that against an RSA modulus N with balanced primes the runtime of the algorithm is $O(N^{1/4})$, making its an inefficient method.

Example.
Let $N = 8051$ and $f(x) = (x^2 + 1) \bmod 8051$, then, from the initial values $a = 2, b = 2$, we have

- when $i = 1$, $a = 5$, $b = 26$, then $\gcd(|x - y|, 8051) = 1$
- when $i = 2$, $a = 26$, $b = 7474$, then $\gcd(|x - y|, 8051) = 1$
- when $i = 3$, $a = 677$, $b = 871$, then $\gcd(|x - y|, 8051) = 97$
- when $i = 4$, $a = 7474$, $b = 1481$, then $\gcd(|x - y|, 8051) = 1$

3 Pollard's $p - 1$ algorithm

Let $n = pq$, where p, q are large primes. If $q|(p - 1)$, where q is also a large prime, then p is a strong prime. Otherwise, p is strong, and n can be factored by Pollard's $p - 1$ algorithm. For example, $p - 1 = 2p_1p_2p_3 \cdot p_k$ only includes small prime factors, where $p_0 = 2$. If all p_i, $i = 1, 2, \cdots, k$, $p_i < B$, where B is an integer, we will know that

$$p_0p_1p_2p_3 \cdot p_k|B! \Rightarrow (p - 1)|B! \Rightarrow B! = (p - 1) \cdot \alpha$$

Algorithm: Pollard's ρ Algorithm: Given a composite $n = pq$:

1. set $a = 2$.

2. For $i = 1, 2, \cdots, B$ do:

   a) Compute $a \equiv a^i \bmod n$.

   $d = gcd(a - 1, n)$;

   if $(1 < d < n)$

   * return $p = d$;

   else

   * return failure.

After running the algorithm, we know $a \equiv 2^{B!} \bmod n$. That is, $a = 2^{B!} + kn = 2^{B!} + kpq = 2^{B!} + k'p$, where $k' = kq$. Then,

$$a \equiv 2^{B!} \bmod p$$

Based on the Fermat's Little Theorem, we know

$$2^{p-1} \equiv 1 \bmod p \Rightarrow (2^{p-1})^{\alpha} \equiv 1^{\alpha} \bmod p \Rightarrow 2^{B!} \equiv 1 \bmod p \Rightarrow a \equiv 1 \bmod p$$

Then,

$$p|(a-1) \Rightarrow a - 1 = p \cdot k'' \Rightarrow \gcd(a - 1, n) = p$$

Example. Suppose $n = 15770708441$. If we set $B = 180$, then from the above algorithm, we can find that $a = 1162221425$ and d is computed to be 135979. In fact, the complete factorization of n into primes is

$$15770708441 = 135979 \times 115979$$

In this example, the factorization is successful because $135979 - 1 = 135978$ has only "small" prime factors:

$$135978 = 2 \times 3 \times 131 \times 173$$

Therefore, by taking $B \geq 173$, it will be the case that $135978|B!$, as desired.

References

[1] G. J. Simmons and M. J. Norris, Preliminary Comments on the MIT Public-key Cryptosystem, Cryptologia (1977).

[2] J. M. Pollard, A Monte Carlo Method for Factorization, BIT Numerical Mathematics (1975).

[3] H. Riesel, Prime Numbers and Computer Methods for Factorization, Birkhauser, 1994.

[4] Abdullah Darwish, Imad Khaled Salah, and Saleh Oqeili, Mathematical At- tacks on RSA Cryptosystem, Journal of Computer Science (2006).