11464: INFORMATION SYSTEMS SECURITY

Chapter 2: Cryptography Tool

Cryptography Tool

By

Mustafa Al-Fayoumi

Outline

- Confidentiality with Symmetric Encryption
  - Symmetric Encryption
  - Symmetric Block Encryption Algorithms
  - Stream Ciphers
- Message Authentication and Hash Functions
  - Authentication Using Symmetric Encryption
  - Message Authentication without Message Encryption
  - Secure Hash Functions
  - Other Applications of Hash Functions
- Public-key cryptosystems
  - Public-Key Encryption Structure
  - Applications for Public-Key Cryptosystems
  - Requirements for Public-Key Cryptography
  - Asymmetric Encryption Algorithms
- Digital Signature and Key Management
  - Digital Signature
  - Public-Key Certificates
  - Symmetric Key Exchange Using Public-Key Encryption
  - Digital Envelopes

Introduction of Cryptography

- What is cryptography and cryptology?
- The main components of a crypto system.
- Problems solved by cryptography.
- Basic concepts: symmetric cryptography, asymmetric cryptography, digital signatures.
- Types of algorithms and related concepts.

Cryptography and Cryptology

- **Enciphering** or **Encryption:** transformation of intelligible, understandable information into unintelligible form to disguise its meaning and intent from intruders. (Process of converting from plaintext to ciphertext)
- **Deciphering or Decryption:** The inverse transformation of encrypted information into intelligible form. (Restoring the plaintext from the ciphertext)
- Both encryption and decryption are based on keys. It should be difficult or impossible to decrypt a message without knowing the key.
- **Cryptography**: Study of encryption principles/methods (encryption + decryption).
- **Cryptanalysis (codebreaking) :** analyzing encrypted information with the intent of recovering the original plain information, without knowing the key. (study of principles/ methods of deciphering ciphertext *without* knowing key).
- **Cryptology:** Field of both cryptography and cryptanalysis.

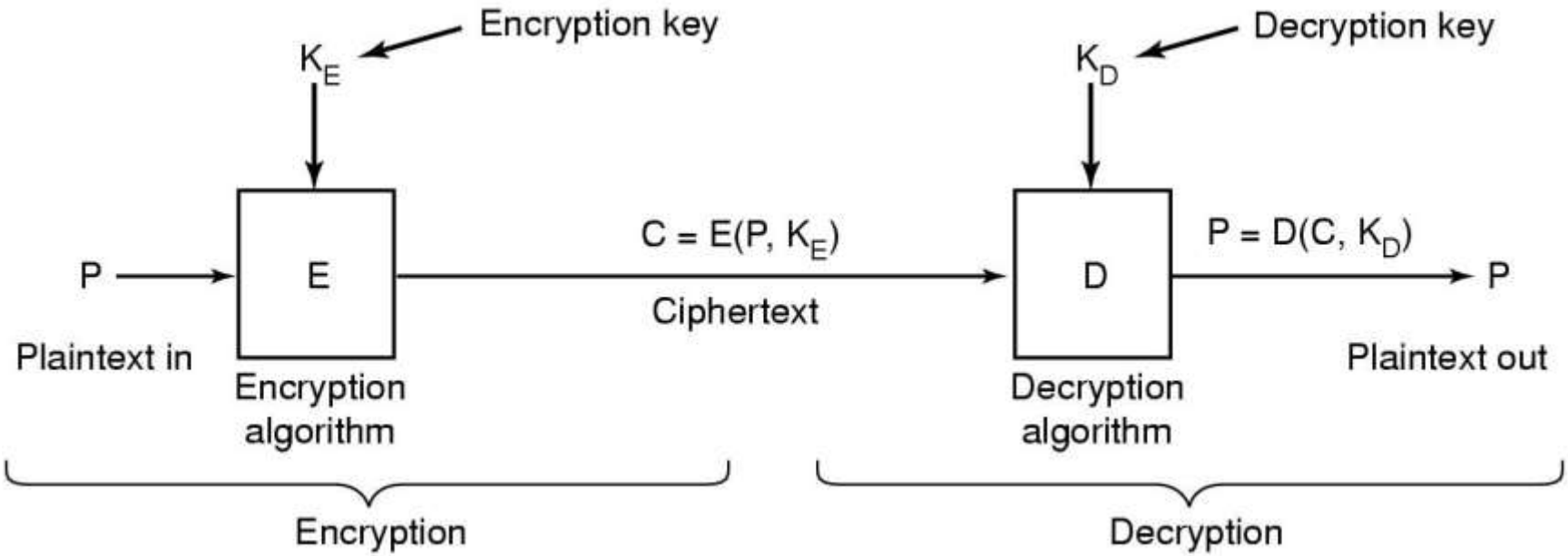
The major components of a crypto system (the model)

- **Plain text:** the original message before encryption.
- **Encryption Algorithm:** the algorithm used to transform the plaintext into unintelligible form (the cipher text).
- **The cipher text:** the encrypted text.
- **Encryption key:** the encryption process is always based on a key.
- **Decryption Algorithm:** used to transforms cipher text back to plaintext.
- **The Decryption key:** the key used in the decryption process.

All algorithms must be public; only the keys are secret.

The Encryption and Decryption Process

- The encryption model



Intruders and Cryptanalysis

- It is assumed that there is an intruder who listens to all communications and he may copy or delete any message
  - An active intruder modifies some messages and re-inserts them
  - A passive intruder just listens
- To decrypt a message without having a key, an intruder practices the art of cryptanalysis

What Does Cryptography Solve?

- Cryptographic algorithms important element in security services
- Various types of algorithms
  - Symmetric encryption
  - Public-key (asymmetric) encryption
  - Secure hash functions
  - Digital signatures and key management

What Does Cryptography Solve?

Cryptographic provides mechanisms for:

- **Confidentiality**
  - Ensure that nobody can get knowledge of what you transfer even if listening to the whole conversation
- **Integrity**
  - Ensure that message has not been modified during the transmission
- **Authenticity**
  - You can verify that you are talking to the entity you think you are talking to
- **Identity**
  - You can verify who is the specific individual behind that entity
- **Non-repudiation**
  - The individual behind that asset cannot deny being associated with it

Cryptographic Systems

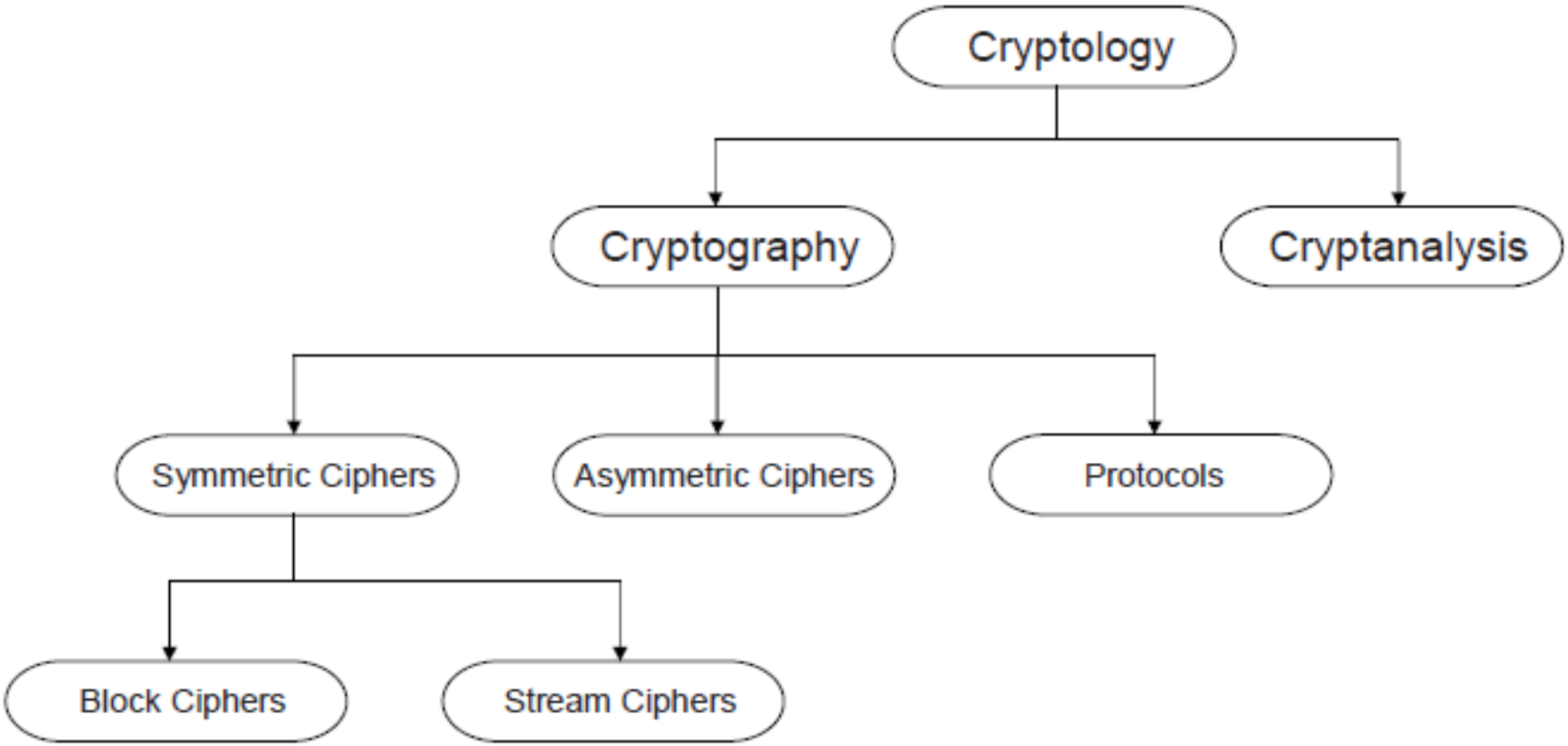
Cryptography can characterize by:

- **Type of encryption operations used**
  - Substitution / Transposition / Product
- **Number of keys used**
  - Single-key Or Secret / Two-key Or Public
- **Way in which plaintext is processed**
  - Block / Stream

Cryptography Classification

The old Encryption and Decryption techniques before the implementation of computer systems are called Classical techniques, while those invented and implemented for the computer systems are called modern techniques. However, cryptography systems (whether classical or modern) are generally classified along three independent dimensions:

Classification the field of Cryptography



Cryptography Classification

1. Type of operations used for transforming plaintext to ciphertext. All encryption algorithm are based on general principle:

a) **Substitution:**

b) **Transposition**

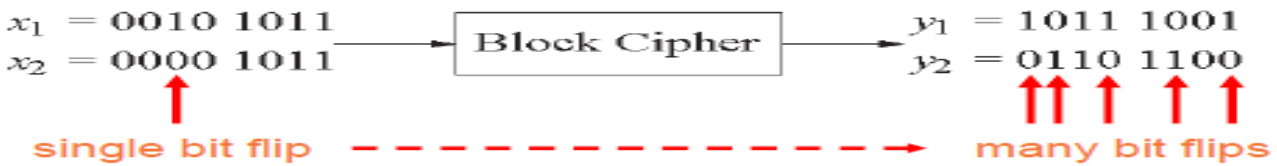
c) **Bit manipulation**

All types aim to fulfill Shanon's Concept of Security: "Diffusion and Confusion"

Shannon's principle of Confusion and diffusion

- **Claude Shannon**: There are two primitive operations with which strong encryption algorithms can be built:
- **Confusion**
  - means that each binary digit (bit) of the ciphertext should depend on several parts of the key.
    - The relationship between the ciphertext and the key is obscured
    - Today, a common element for achieving confusion is substitution, which is found in both AES and DES.
- **Diffusion**
  - means that if we change a single bit of the plaintext, then (statistically) one bit out of two of the ciphertext should change, and similarly, if we change one bit of the ciphertext, then approximately one half of the plaintext bits should change.
    - The influence of one plaintext symbol (bit) is spread over many cipher text symbols (bits)
    - **changing of one bit of plaintext results** *on average* in the **change of half the output bits.**
    - A simple diffusion element is the bit permutation, which is frequently used within DES.

Example:



Cryptography Classification

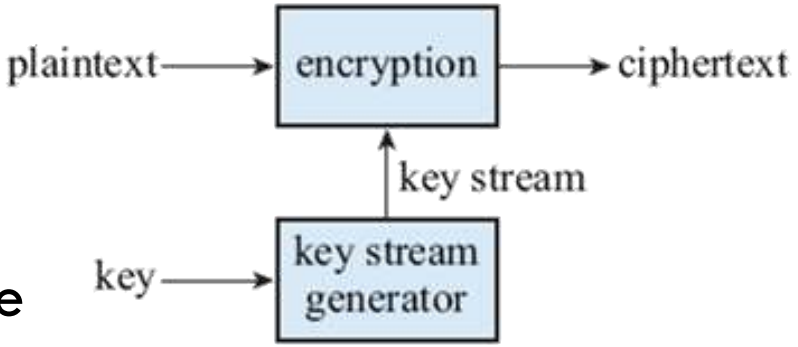
2. Cryptography is classified according to the number of keys used:

- **Symmetric key cryptography (Private-Key cryptography)**
  - The same key to encrypt and decrypt. Like DES Data Encryption Standard.
- **Asymmetric key cryptography (Public-Key Cryptography)**
  - Two mathematically related keys are used, one is the public key to encrypt and the other is the private key to decrypt. Like RSA or Al Gamal, DSA.

Cryptography Classification

3. Processing Way, in which the plaintext is processed:

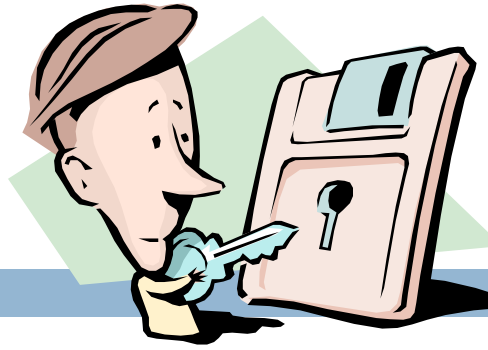
- **Block Cipher:**
  - breaks the plaintext into equal-sized blocks, usually of 64 or 128 bits, and encrypts each block separately to produce a ciphertext output exactly equal in length to the input.
  - A block cipher processes the input one block of elements at a time, producing an output block for each input block.
- **Stream Cipher**
  - The input elements are processed individually, producing an output as one element at a time, too.
  - processes the input elements continuously, producing output one element at a time, as it goes along.



- For applications that require encryption/decryption of a stream of data, such as over a data communications channel or a browser/Web link, a stream cipher might be the better alternative.

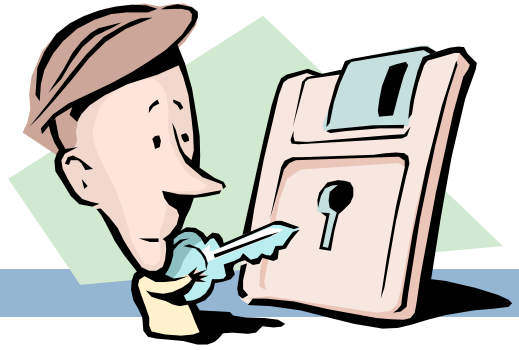
- For applications that deal with blocks of data, such as file transfer, e-mail, and database, block ciphers may be more appropriate. However, either type of cipher can be used in virtually any application.

Symmetric Encryption



- The universal technique for providing confidentiality for transmitted or stored data
- Also referred to as conventional encryption or single-key encryption
- Was the only type of encryption in use prior to the development of public-key encryption in the 1970s
- Remains by far the most widely used of the two types of encryption
  - **Data Encryption Standard (DES)**
  - **Advanced Encryption Standard (AES)**
- sender and recipient share a common key

Symmetric Encryption



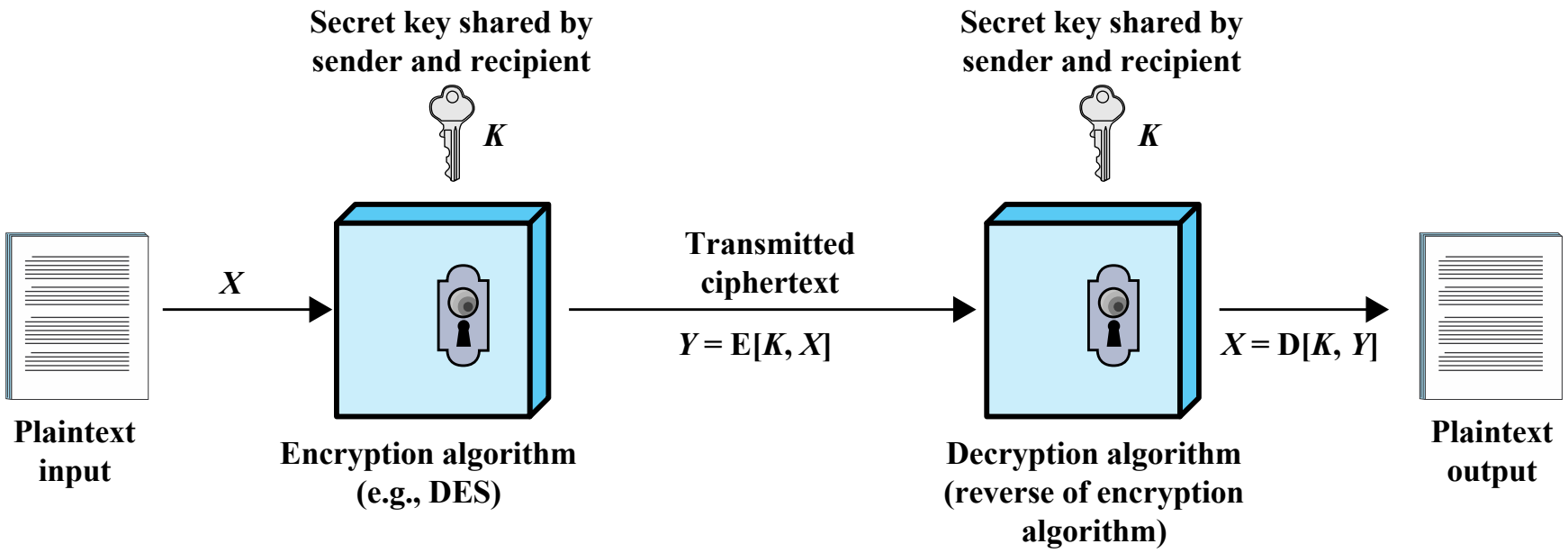
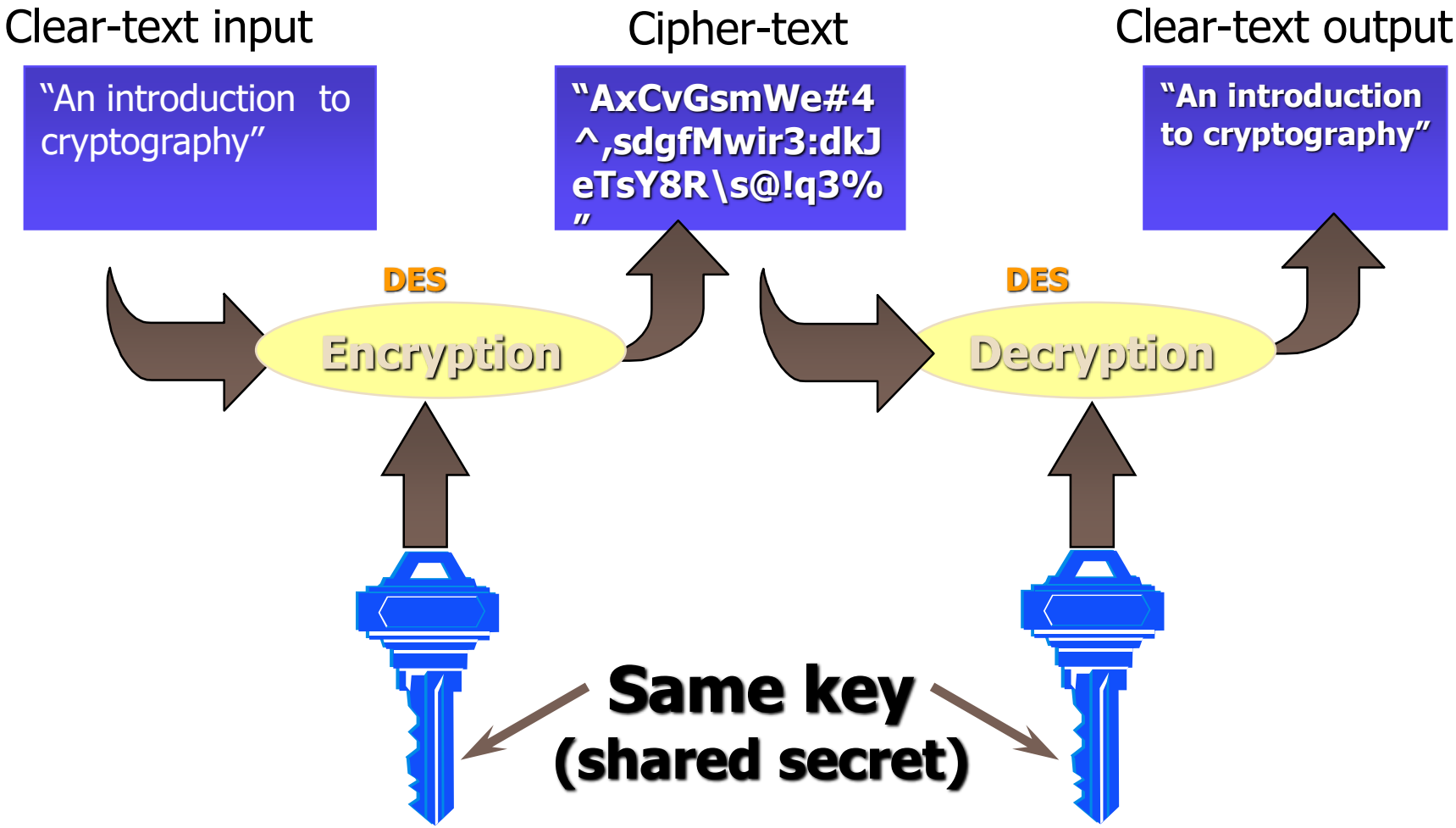


Figure 2.1 Simplified Model of Symmetric Encryption

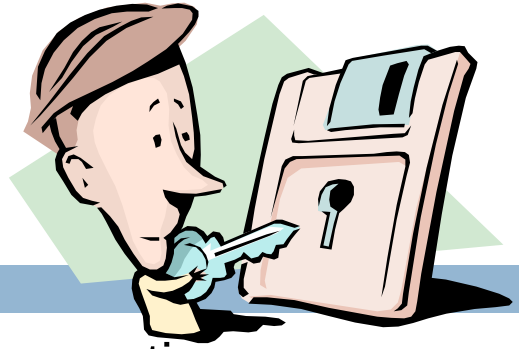
Confidentiality With Symmetric Encryption



Symmetric Encryption

- In a symmetric encryption system, both the sender and receiver must possess the same key.
- The sender encrypts the message using the key and the receiver decrypts the cipher-text message using the same secret key.
- The word "symmetric" here means that the same key is used for encryption and decryption.

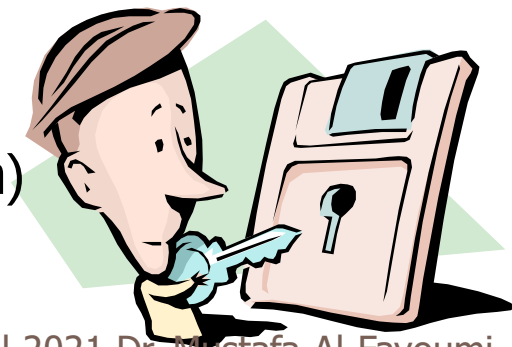
Requirements



- Two requirements for secure use of symmetric encryption:
  - **A strong encryption algorithm**: we would like the algorithm to be such that an opponent who knows the algorithm and has access to one or more ciphertexts would be unable to decipher the ciphertext or figure out the key. This requirement is usually stated in a stronger form:
    - The opponent should be unable to decrypt ciphertext or discover the key even if he or she is in possession of a number of ciphertexts together with the plaintext that produced each ciphertext.
  - **A secret key known only to sender / receiver:** Sender and receiver must have obtained copies of the secret key in a secure fashion and must Keep the key secure
- Assume encryption algorithm is known
- Implies a secure channel to distribute key

Examples of Symmetric Cipher

- The two most important symmetric encryption algorithms which are block encryption algorithms:
  - **DES** (Data Encryption Standard)
  - **AES** (Advanced Encryption Standard)
- Other Exapmples:
  - Twofish
  - Serpent
  - Blowfish
  - CAST5
  - RC4
  - Triple DES
  - IDEA (International Data Encryption Algorithm)



Attacking Symmetric Encryption

There are two general approaches to attacking a symmetric encryption scheme

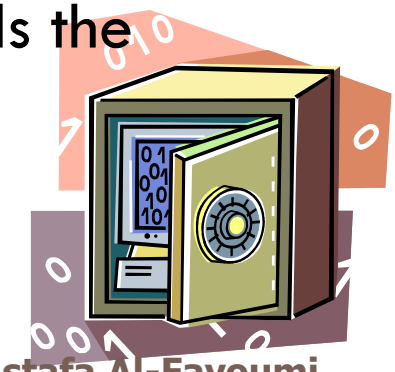
- cryptanalysis
  - Rely on
    - Nature of the algorithm
    - Some knowledge of plaintext characteristics
    - Some sample plaintext-ciphertext pairs
  - Exploits characteristics of algorithm to deduce specific plaintext or key
    - If successful all future and past messages encrypted with that key are compromised
- brute-force attack
  - Try all possible keys on some ciphertext until get an intelligible translation into plaintext

Table 2.1 Types of Attacks on Encrypted Messages

Type of Attack	Known to 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 • 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 key
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 key

Encryption Scheme Security

- 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
- 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



Brute-Force Attack

- Imagine that you have a bunch of keys and you know that one of them (but not which one) will unlock the door to a room you wish to enter. What would you do to unlock the door?

  - The obvious thing to do is to try every key in the lock in turn.
    - If you are lucky, the first one you try will open the door.
    - If you are unlucky it may be the last one.
  - **A similar method to this can be used to break a cipher using a known algorithm!**

Brute-Force Attack

- **Example:** if you have a ciphertext message that you know has been encrypted using the simple Caesar cipher which will be described later, ***how many keys would you need to try before you could be certain of finding the right one?***
  - **The answer is 26, since there are 26 possible keys that could be used with this algorithm.**

- This method of trying all possible combinations in a key space is known as **a brute force attack.**
- The number of possible key combinations for a particular algorithm is known as its **key space.**

Brute-Force Attack

- **The time taken to break a cipher** by this method alone is **directly proportional to the key space.**
  - **For example:** The Caesar cipher has a very small key space and so can be broken very quickly.
- **A brute force attack can be applied to transposition ciphers as well as substitution ciphers:**
  - **Substitution cipher:** test every key in the key space.
  - **Transposition cipher:** test every permutation of the possible transpositions.

How long would it take a Brute force attack to break a cipher?

Brute-Force Attack

- **Exercise 2:** How many different arrangements would be possible using the seven letters of the word '*article*'?
- **Sol:**
  - Each letter in the word 'article' appears only once.
  - Taking one letter at a time, the first can appear in any of the seven positions; the second in any of the 6 remaining positions; the third in any of the five remaining positions; and so on.
  - This gives a total possible number of combinations of

$$7! = 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1 = 5040$$

Brute-Force Attack

- From **Exercise 2**, working through all possible 5040 permutations using a pencil and paper would take quite a long time to do.
- **However, a computer would be able to yield the correct answer in a fraction of a second!**
- **Example:** A computer can perform one thousand billion calculations every second, how long would it take to try all the possible arrangements of the word 'article'?
- **Sol.:** The computer speed is: $1000 \times 1\ 000\ 000\ 000 = 1 \times 10^{12}$ calculations per second
  - So to perform $5040 \approx 5 \times 10^3$ calculations would take roughly:

$$\frac{5 \times 10^3}{1 \times 10^{12}} = 5 \times 10^{-9} \text{ Seconds} = 5 \text{ nanoseconds.}$$

Brute-Force Attack

- **Homework1:** Using a computer that can perform 10^{12} calculations a second, roughly how long would it take to try all possible permutations of:

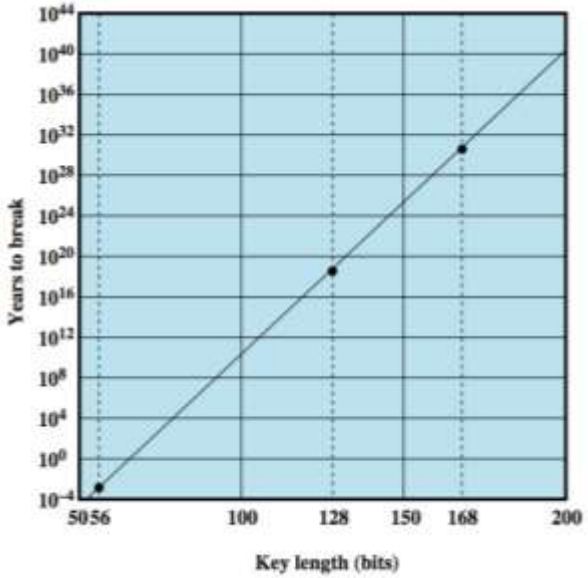
  (a) 10 different letters

  (b) 15 different letters

  (c) 20 different letters.

Exhaustive Key Search

Key Size (bits)	Number of Alternative Keys	Time Required at 1 Decryption/μs	Time Required at 10^6 Decryptions/μs
32	$2^{32} = 4.3 \times 10^9$	$2^{31}\ \mu s = 35.8$ minutes	2.15 milliseconds
56	$2^{56} = 7.2 \times 10^{16}$	$2^{55}\ \mu s = 1142$ years	10.01 hours
128	$2^{128} = 3.4 \times 10^{38}$	$2^{127}\ \mu s = 5.4 \times 10^{24}$ years	5.4×10^{18} years
168	$2^{168} = 3.7 \times 10^{50}$	$2^{167}\ \mu s = 5.9 \times 10^{36}$ years	5.9×10^{30} years
26 characters (permutation)	$26! = 4 \times 10^{26}$	$2 \times 10^{26}\ \mu s = 6.4 \times 10^{12}$ years	6.4×10^6 years



Brute-Force Attack- Calculations

- **Example:** Using the previous table to compute the time would it take a Brute force attack to break a cipher- draw your attention the calculations will refer to average case.
- According to the text book, the author take into his account the following rule: On average half of all possible keys that must be tried to achieve success brute-force attack?

Brute-Force Attack- Calculations

- **Sol.:** Case 1: the key size (bits) =32 bits and assume the time required for computer would take 1µs to perform a single decryption, (i.e 1 Microsecond (µs) =10^{-6} second)
  - The computer speed is: it would perform 1 key per microsecond (i.e 10^6 keys (calculations) per second) - 1 key per microsecond
  - The key space is $2^{32} = 4.3 \times 10^9$
  - In the case of average we will take half of all possible keys (key space) = $4.3 \times 10^9 / 2 = 2.15 \times 10^9$ keys = 2^{31} keys.
  - calculations would take roughly:

$$\frac{2.15 \times 10^9}{1} = 2.15 \times 10^9 = 2147483648 \text{ µs}$$

$$2147483648 * \frac{1}{10^6\ second} = 2147.483648 \text{ Seconds}$$

$$\frac{2147.483648}{60\ minutes} = 35.8 \text{ minutes.}$$

Brute-Force Attack- Calculations

- **Sol.:** Case 1: the key size (bits) =56 bits and the time required for computer is 10^6 decryption (calculations) per μs (1 μ =10^{-6} second)
  - The time required computer speed is: it will perform 1 million keys per microsecond (i.e 10^6 keys (calculations) per microsecond)
  - The key space is $2^{56} = 7.2 \times 10^{16}$
  - In the case of average we will take half of all possible keys (key space) $= 7.2 \times 10^{16}/2 = 3.6 \times 10^{16}$ keys $= 2^{55}$ keys.
  - calculations would take roughly:

$$\frac{3.6 \times 10^{16}}{1 \times 10^{6}} = 36028797019\mu s$$

$$36028797019 * \frac{1}{10^6\ second} = 36028.7970\ S$$

$$= \frac{36028.7970}{60\ minutes} = 600.5 minutes. = \frac{600.5}{60\ hours} = 10.8\ hours$$

Homework 1

- Calculate all items in the pervious table in term of:
  - On average case
  - On worst case
  - Repeat all calculations using your computer (speed)
- Compare between the above results

a) If a key is supposed to be eight letters word "mustache", how many different attempts by brute force attack to break the cipher text provided that each letter in the word "mustache" appears only once, show your answer in details?

- The first letter can appear in any of the 8 positions; the second in any of the 7 remaining positions; the third in any of the 6 remaining positions; and so on.
- This gives a total possible number of combinations of
- $8! = 8 \times 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1 = 40320$

b) Consider the previous case in part (a) but each letter has the probability to appear more than once, how many different attempts by brute force attack are needed to break the key? (5 marks)

- The first position can be any of the 8 letters; the second position can be any of the 8 letters as well, and so on. This gives a total possible number of combinations of
- 8^8 = 16777216

c) a fast computer was used to break the previous word in part (b), the speed of processor was 2 MIPS (million instructions per second), if each attempt needs 60 instructions, what is the time needed to break the word. (Consider the worst case, i.e. the last attempt is the successful one), express your answer in second and minutes?

- Total number of instructions = 16777216x 60 = 1006632960 instructions
- Time = 1006632960 / (2x10^6) = 503.3 s *(2.5 marks)*
- Time = 8.4 m.

Symmetric Block Encryption Algorithms

	DES	Triple DES	AES
Plaintext block size (bits)	64	64	128
Ciphertext block size (bits)	64	64	128
Key size (bits)	56	112 or 168	128, 192, or 256

DES = Data Encryption Standard
AES = Advanced Encryption Standard

DES

- Data Encryption Standard (DES) is the most widely used encryption scheme
  - Referred to as the Data Encryption Algorithm (DEA)
  - uses 64 bit plaintext block and 56 bit key to produce a 64 bit ciphertext block
  - Strength concerns:
    - Concerns about algorithm itself : DES is the most studied encryption algorithm in existence
  - use of 56-bit key:
    - With a key length of 56 bits, there are 2^{56} possible keys, which is approximately $7.2 * 10^{16}$ keys
    - Electronic Frontier Foundation (EFF) announced in July 1998 that it had broken a DES encryption

DES and Triple-DES

Key size (bits)	Cipher	Number of Alternative Keys	Time Required at 10^9 decryptions/s	Time Required at 10^{13} decryptions/s
56	DES	$2^{56} \approx 7.2 \times 10^{16}$	2^{55} ns = 1.125 years	1 hour
128	AES	$2^{128} \approx 3.4 \times 10^{38}$	2^{127} ns = 5.3×10^{21} years	5.3×10^{17} years
168	Triple DES	$2^{168} \approx 3.7 \times 10^{50}$	2^{167} ns = 5.8×10^{33} years	5.8×10^{29} years
192	AES	$2^{192} \approx 6.3 \times 10^{57}$	2^{191} ns = 9.8×10^{40} years	9.8×10^{36} years
256	AES	$2^{256} \approx 1.2 \times 10^{77}$	2^{255} ns = 1.8×10^{60} years	1.8×10^{56} years

- Average Time Required for Exhaustive Key Search

Triple-DES

- Repeats basic DES algorithm three times using either two or three unique keys, for a key size of 112 or 168 bits
- First standardized for use in financial applications in ANSI standard X9.17 in 1985
- Attractions:
  - 168-bit key length overcomes the vulnerability to brute-force attack of DES
  - Underlying encryption algorithm is the same as in DES
- Drawbacks:
  - Algorithm is sluggish in software(much more secure but also much slower)
  - Uses a 64-bit block size - For reasons of both efficiency and security, a larger block size is desirable.

Advanced Encryption Standard (AES)

- needed a better replacement for DES
- NIST called for proposals in 1977
  - Should have a security strength equal to or better than 3DES
  - significantly improved efficiency,
  - symmetric block cipher with a block length of 128 bits
  - support for key lengths of 128, 192, and 256 bits
- Evaluation criteria included security, computational efficiency, memory requirements, hardware and software suitability, and flexibility.
- selected Rijndael in Nov 2001
- now widely available commercially

Practical Security Issues

- Typically symmetric encryption is applied to a unit of data larger than a single 64-bit or 128-bit block
- Electronic codebook (ECB) mode is the simplest approach to multiple-block encryption
  - Each block of plaintext is encrypted using the same key
  - Cryptanalysts may be able to exploit regularities in the plaintext
- Modes of operation
  - Alternative techniques developed to increase the security of symmetric block encryption for large sequences like: **Cipher-block chaining (CBC).**
  - Overcomes the weaknesses of ECB

Practical Security Issues

- There are two basic approaches to block encryption:
  - One is to encrypt each block independently of any other,
  - The other is to encrypt each block so that its output ciphertext is dependent on the output of the previous block.
- Electronic Codebook (ECB): An independent encryption approach

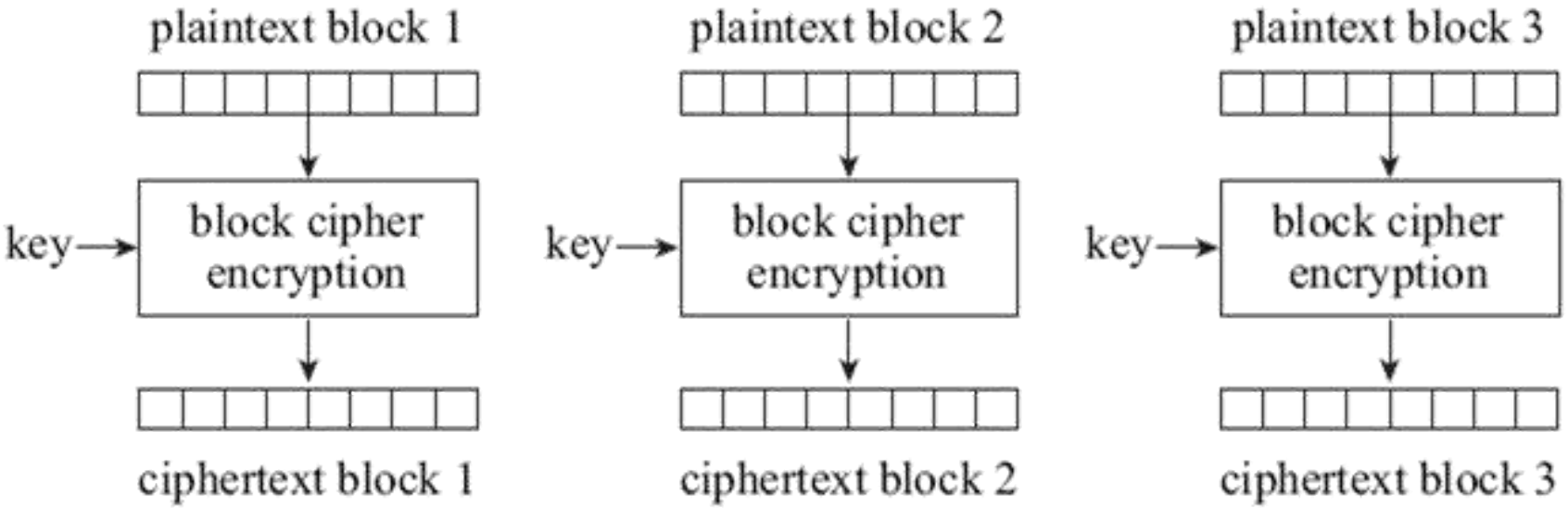


Figure 5.19 Electronic codebook (ECB) mode (adapted from Wikipedia, 2009)

Practical Security Issues

- **Electronic Codebook (ECB):**
- The same key will be used for each block.
- The encryption of each block is completely independent from the other blocks.
- **Drawbacks of ECB:**
- **Two similar blocks of plaintext will result in similar blocks of ciphertext**
- Since the position of the ciphertext blocks remains fixed relative to the plaintext blocks **this introduces a vulnerability.**
- **ECB is not practical when data involves long repetitive strings of 1s and 0s,** such as a picture data.

Practical Security Issues

- **Cipher-block chaining (CBC):** A dependent encryption approach

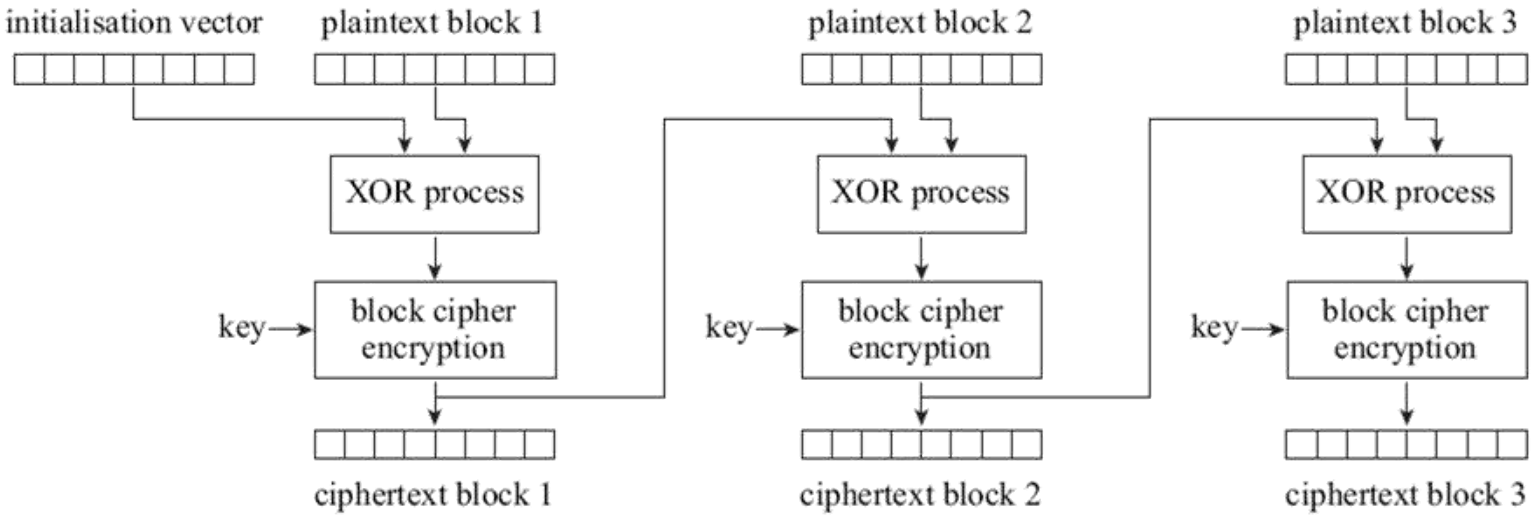


Figure 5.21 Cipher-block chaining (CBC) mode

Practical Security Issues

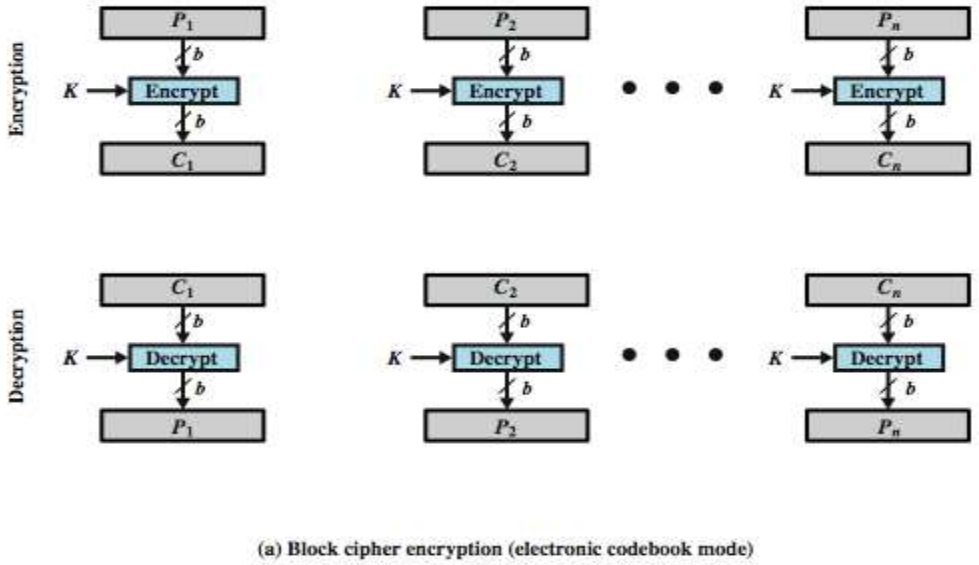
- **Cipher-block chaining (CBC):**
- An XOR process is used to combine the ciphertext output from one block with the plaintext input of the following block.
- Every ciphertext block is dependent on the ciphertext output from the preceding block as well as its own plaintext input
- An encryption of identical input blocks will produce different results.
- CBC mode requires an additional extra input, known as an **initialisation vector** (IV), to the first block.
- The **initialisation vector (IV)** is a pseudo-random binary sequence that is used in the XOR process for the first block only.

What is the ciphertext of the previous penguin picture when encrypted with a CBC technique?

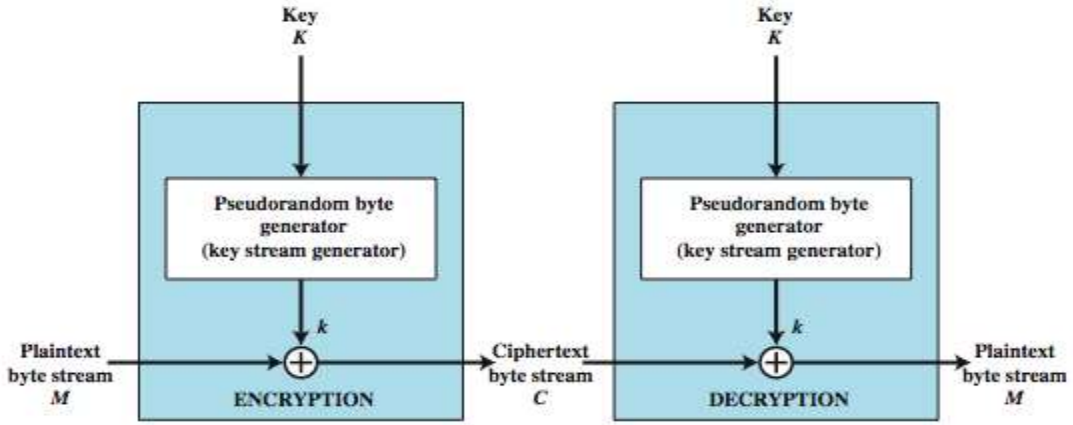
Practical Security Issues

- **Drawbacks of CBC:**
- A single encryption error in one block is cascaded through to the following blocks.
- The decryption relies on knowledge of the previous block.
  - Any error or loss of data in the previous block prevents that block and all following blocks from being decrypted!

Block verses Stream Ciphers



(a) Block cipher encryption (electronic codebook mode)



(b) Stream encryption

Block verses Stream Ciphers

Block Cipher

- **A block cipher** processes the input one block of elements at a time, producing an output block for each input block.
  - Processes the input one block of elements at a time
  - Produces an output block for each input block
  - Can reuse keys
  - More common

Stream Cipher

- Unlike Block Ciphers, **Stream ciphers** operate on very small segments of data – usually at the bit level
  - Processes the input elements continuously
  - Produces output one element at a time
  - Primary advantage is that they are almost always faster and use far less code
  - Encrypts plaintext one byte at a time
  - Pseudorandom stream is one that is unpredictable without knowledge of the input key

Block verses Stream Ciphers

- In stream cipher a key is input to a pseudorandom bit generator that produces a stream of 8-bit numbers that are apparently random.
- A pseudorandom stream is one that is unpredictable without knowledge of the input key and which has an actually random character.
- The output of the generator, called a **keystream**, is combined one byte at a time with the plaintext stream using the bitwise exclusive-OR (XOR) operation.
- For applications that require encryption/decryption of a stream of data, such as over a data communications channel or a browser/Web link, a stream cipher might be the better alternative.
- For applications that deal with blocks of data, such as file transfer, e-mail, and database, block ciphers may be more appropriate.

Message Authentication Code

Message Authentication

- Protects against active attacks (**falsification** of data and transactions)
- Verifies received message is authentic
  - Contents unaltered (**data authentication**)
  - From authentic source (the source is authentic)
  - Timely and in correct sequence (it has not been artificially **delayed** and replayed)
    - The MAC is appended to the message at the source at a time when the message is assumed or known to be correct. The receiver authenticates that message by recomputing the MAC.
    - if the message includes an error-detection code and a sequence number, the receiver is assured that no alterations have been made and that sequencing is proper. If the message also includes a timestamp, the receiver is assured that the message has not been delayed beyond that normally expected for network transit.

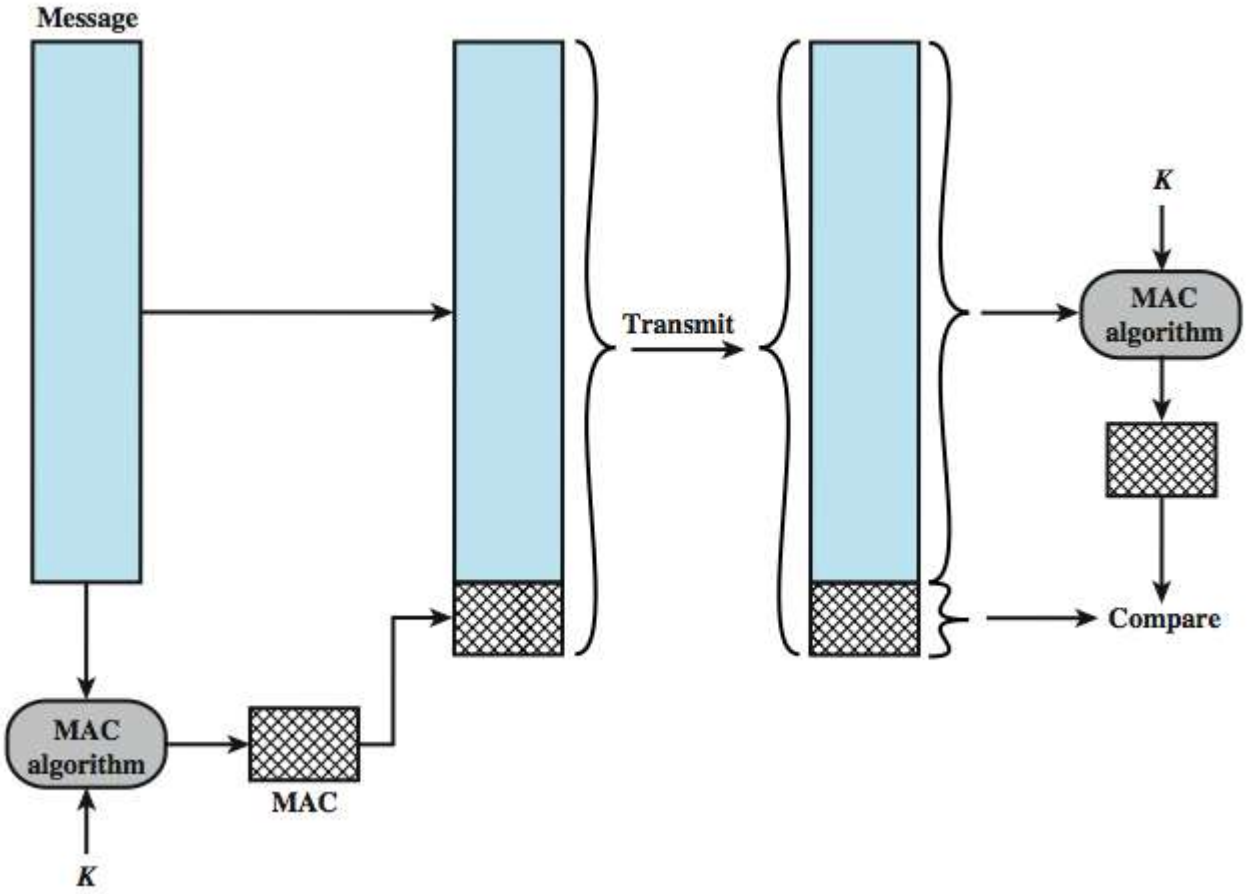
Message Authentication

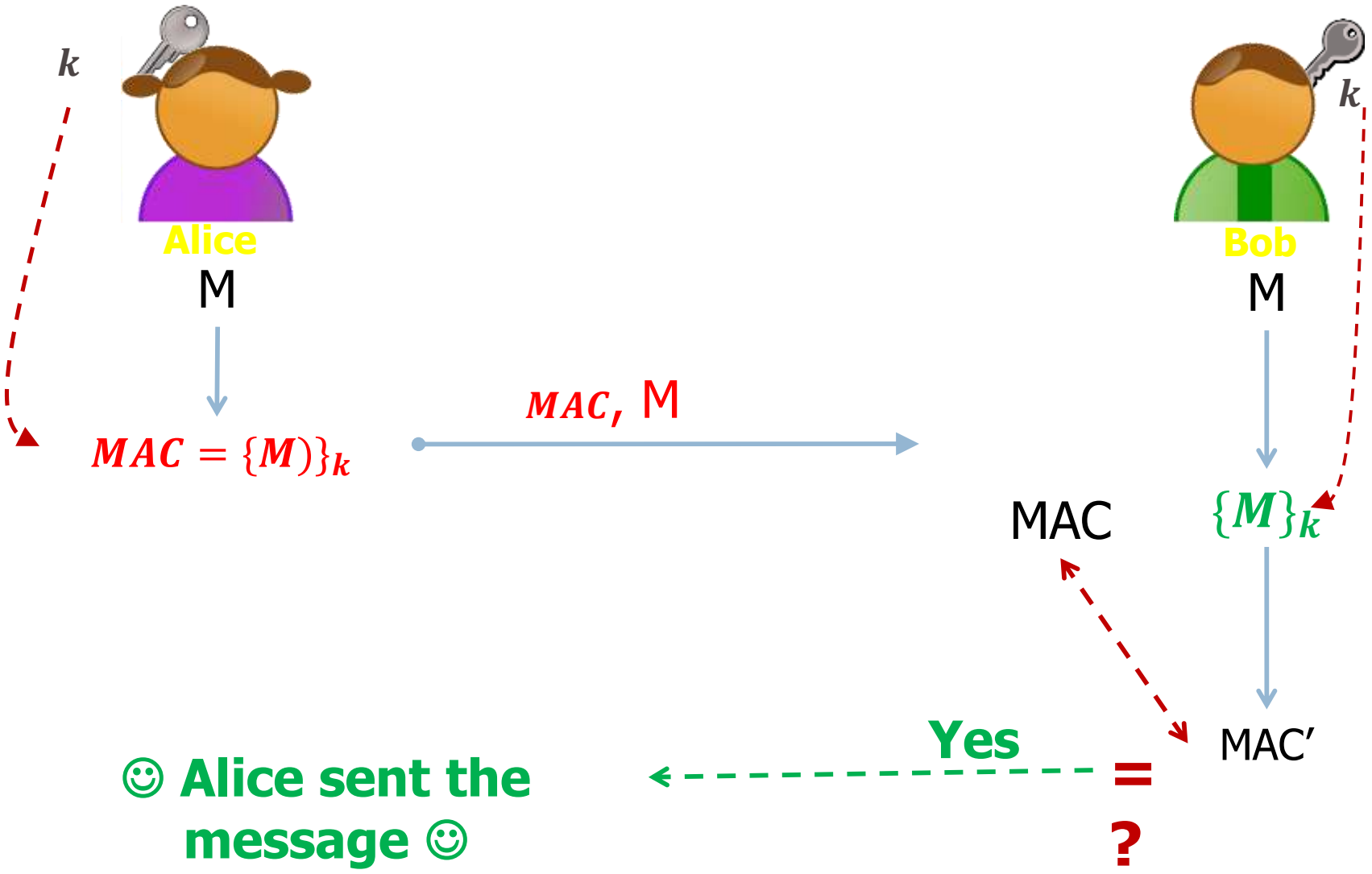
- can use conventional encryption
  - only sender & receiver have key needed (shared key)
    - Symmetric encryption used in message-authentication code (MAC) authentication algorithm
    - A MAC, also known as a cryptographic checksum, is generated by a function C of the form
    - MAC = C(K, M)
      - where M is a variable-length message, K is a secret key shared only by sender and receiver, and C(K, M) is the fixed-length authenticator.
    - If we assume that only the sender and receiver share a key (which is as it should be), then only the genuine sender would be able to encrypt a message successfully for the other participant, provided the receiver can recognize a valid message.

Message Authentication

- or separate authentication mechanisms
  - message authentication is provided as a separate function (authentication tag) from message encryption.
  - append authentication tag to cleartext message

Message Authentication Codes

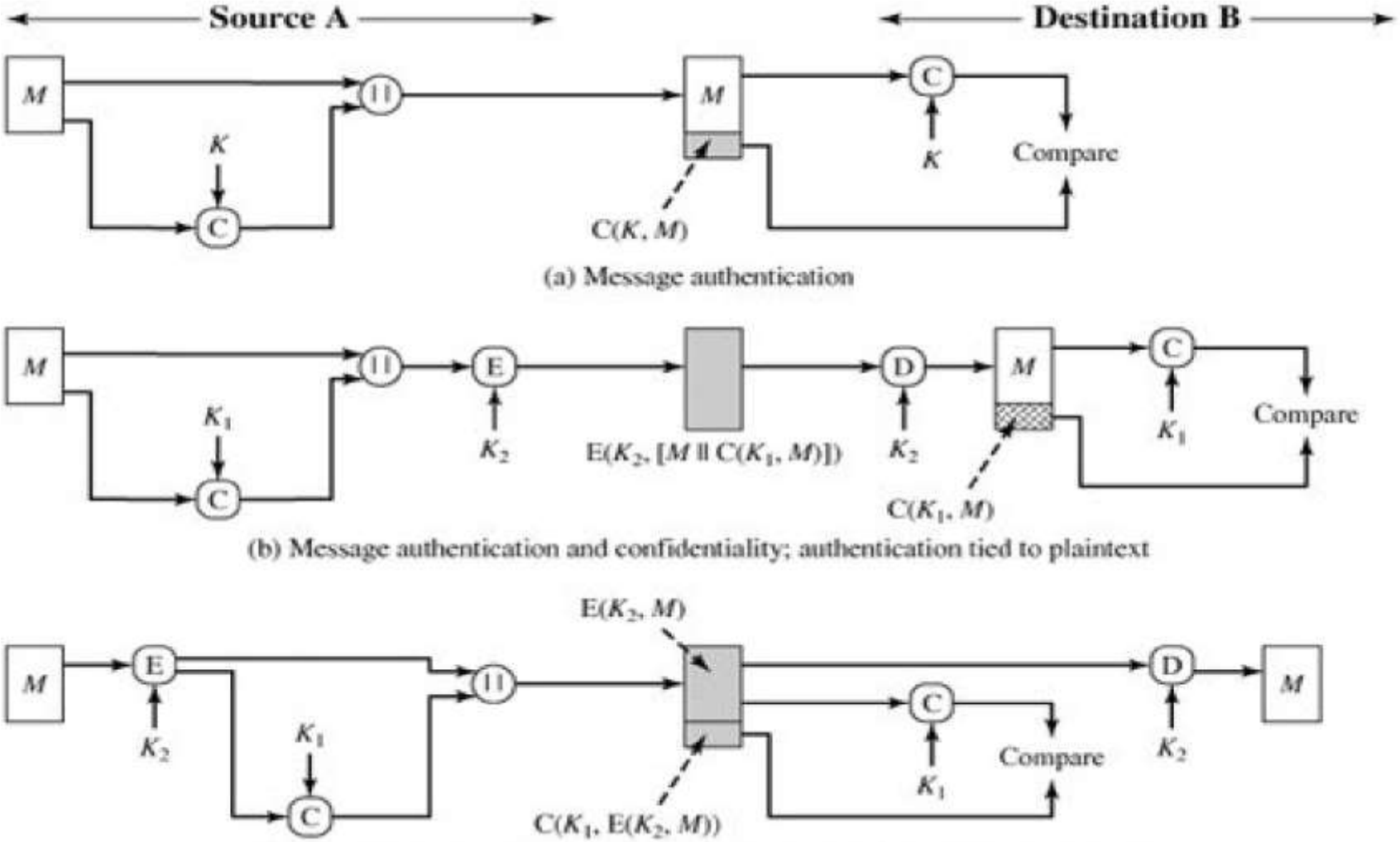



k
Alice
M
MAC = {M}k
MAC, M
Bob
M
k
{M}k
MAC
MAC'
Yes
=
?
☺ Alice sent the message ☺

Message Authentication

- Thus, in assessing the security of a MAC function, we need to consider the types of attacks that may be mounted against it. With that in mind, let us state the requirements for the function. Assume that an opponent knows the MAC function C but does not know *K*. Then the MAC function should satisfy the following requirements:
  1. If an opponent observes *M* and C(*K*, *M*), it should be computationally infeasible for the opponent to construct a message *M* ' such that C(*K*, *M*') = C(*K*, *M*).
  2. C(*K*, *M*) should be uniformly distributed in the sense that for randomly chosen messages *M* and *M*', the probability that C(*K*, *M*) = C(*K*, *M*') is 2^n, where *n* is the number of bits in the MAC.
  3. Let *M*' be equal to some known transformation on *M*. That is, *M*' = f(*M*). For example, f may involve inverting one or more specific bits. In that case, Pr [C(*K*, *M*) = C(*K*, *M*')] = 2^n.

Message Authentication



Cryptographic Hash Functions

Hash Functions

- A hash function H accepts a variable-length block of data M as input and produces a fixed-size hash value
  - $h = H(M)$
  - Principal object is data integrity
- Cryptographic hash function
  - An algorithm for which it is computationally infeasible to find either:

    (a) a data object that maps to a pre-specified hash result (the **one-way property**) For any given code h, it is computationally infeasible to find x such that H(x)= h. A hash function with this property is referred to as one-way or preimage resistant.

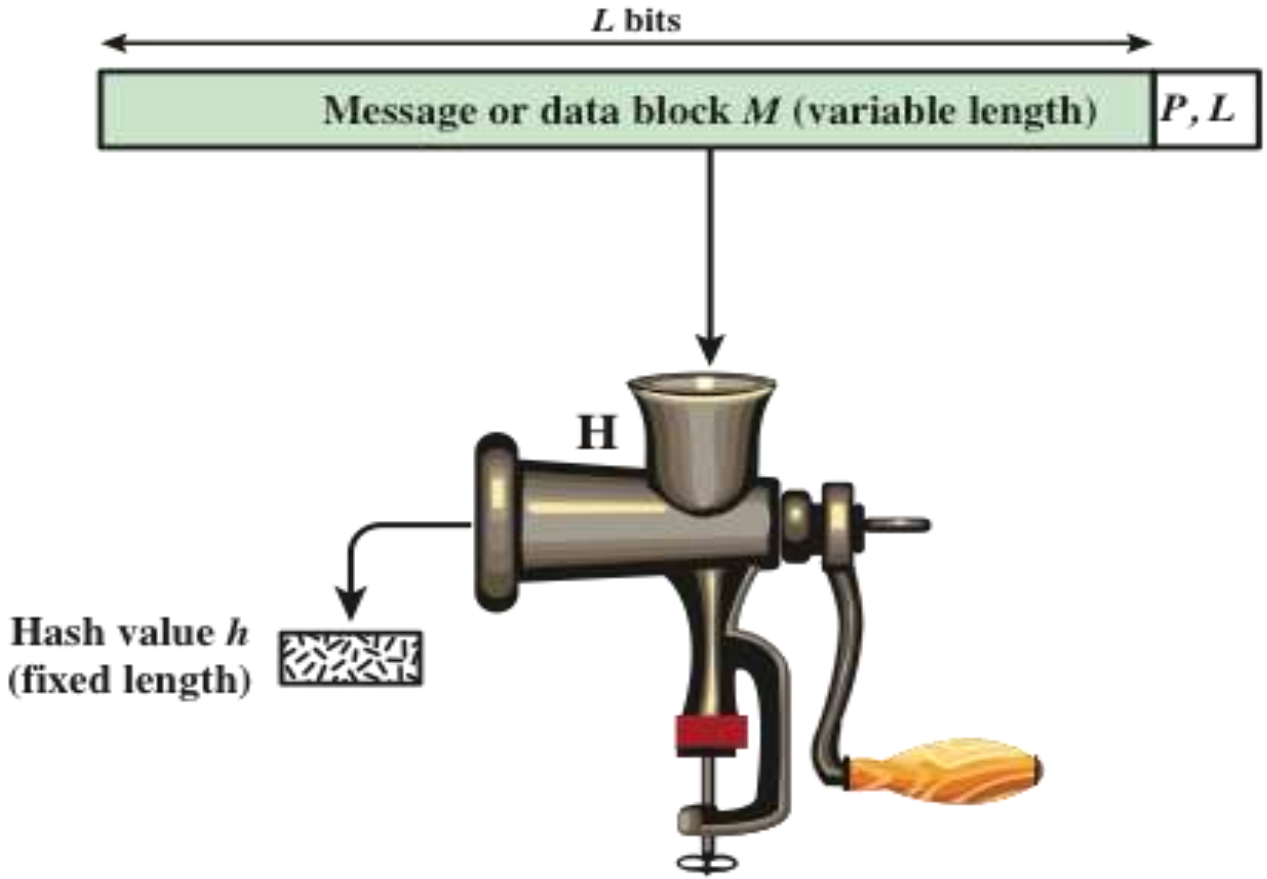
    (b) two data objects that map to the same hash result (the **collision-free property**).
    - For any given block x, it is computationally infeasible to find y ≠ x with H(y)= H(x). A hash function with this property is referred to as second preimage resistant. This is sometimes referred to as weak collision resistant
    - It is computationally infeasible to find any pair (x, y) such that H(x)=H(y). A hash function with this property is referred to as collision resistant. This is sometimes referred to as strong collision resistant.

Hash Functions

- The collision free property guarantees that it is impossible to find an alternative message with the same hash value as a given message.
- This prevents forgery when an encrypted hash code is used (Figures 2.5a and b).
- If this property were not true, an attacker would be capable of the following sequence:
  - First, observe or intercept a message plus its encrypted hash code;
  - second, generate an unencrypted hash code from the message;
  - third, generate an alternate message with the same hash code.

Hash Functions

- The one-way function property: It is easy to generate a code given a message, but virtually impossible to generate a message given a code. This property is important if the authentication technique involves the use of a secret value. (Figures 2.5c)
- The secret value itself is not sent; however, if the hash function is not one way, an attacker can easily discover the secret value: If the attacker can observe or intercept a transmission, the attacker obtains the message *M and the hash code* MD_M $H(S_{AB} \,||\, M)$. *The attacker* then inverts the hash function to obtain $S_{AB} \,||\, M$ $H^{-1}(MD_M)$. *Because the attacker now* has both *M and* $S_{AB} \,||\, M$, *it is a trivial matter to recover* S_{AB}.



P, L = padding plus length field

Figure 11.1 Cryptographic Hash Function; $h = H(M)$

Fig (a): if it is assumed that only the sender and receiver share the encryption key, then authenticity is assured

Message Auth

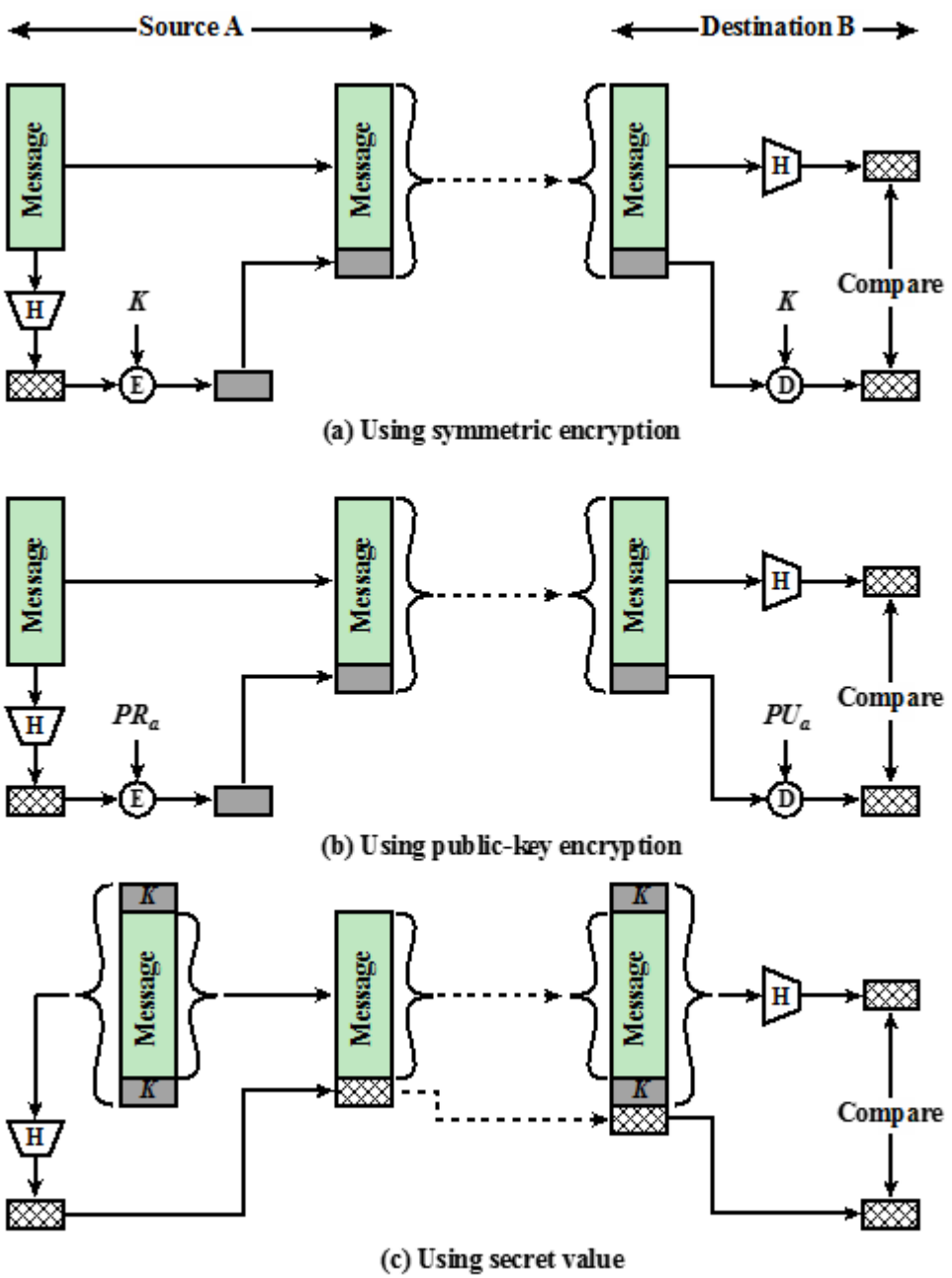


Figure 2.5 Message Authentication Using a One-Way Hash Function.

Hash Functions

- two attack approaches
  - cryptanalysis
    - exploit logical weakness in alg
  - brute-force attack
    - trial many inputs
    - strength proportional to size of hash code ($2^{n/2}$)
- SHA most widely used hash algorithm
  - SHA-1 gives 160-bit hash
  - more recent SHA-256, SHA-384, SHA-512 provide improved size and security

Public-Key Cryptosystems

Public-Key Cryptosystems

- The concept of asymmetric-key cryptography evolved from an attempt to solve two of the most difficult problems associated with conventional encryption.

Key distribution

- **How to have secure communications in general without having to trust a KDC with your key**

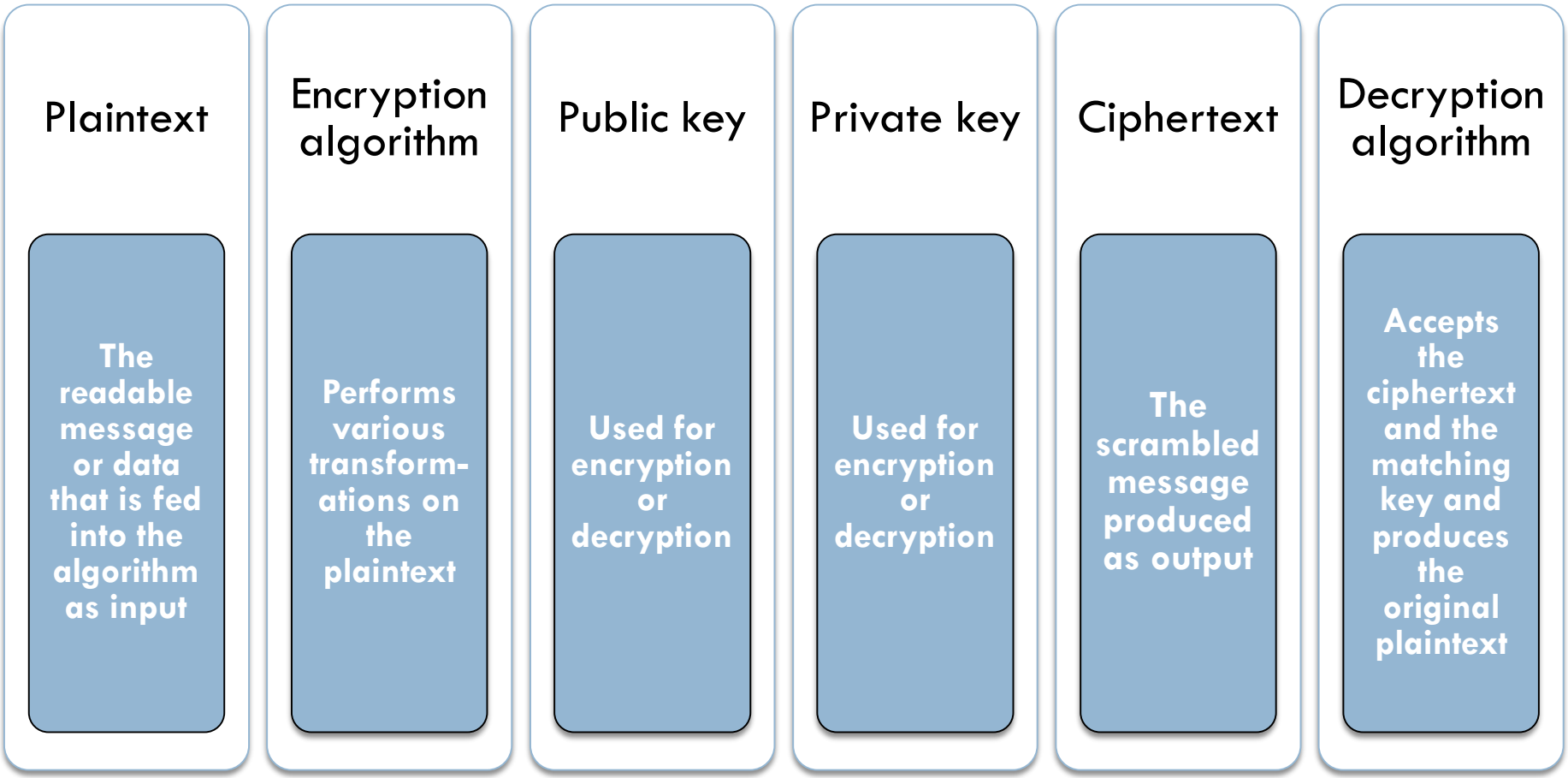
Digital signatures

- **How to verify that a message comes intact from the claimed sender**

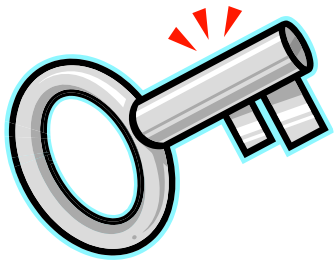
- Whitfield Diffie and Martin Hellman from Stanford University achieved a breakthrough in 1976 by coming up with a method that addressed both problems and was radically different from all previous approaches to cryptography

Public-Key Cryptosystems

- A public-key encryption scheme has six ingredients:



Public-Key Cryptography



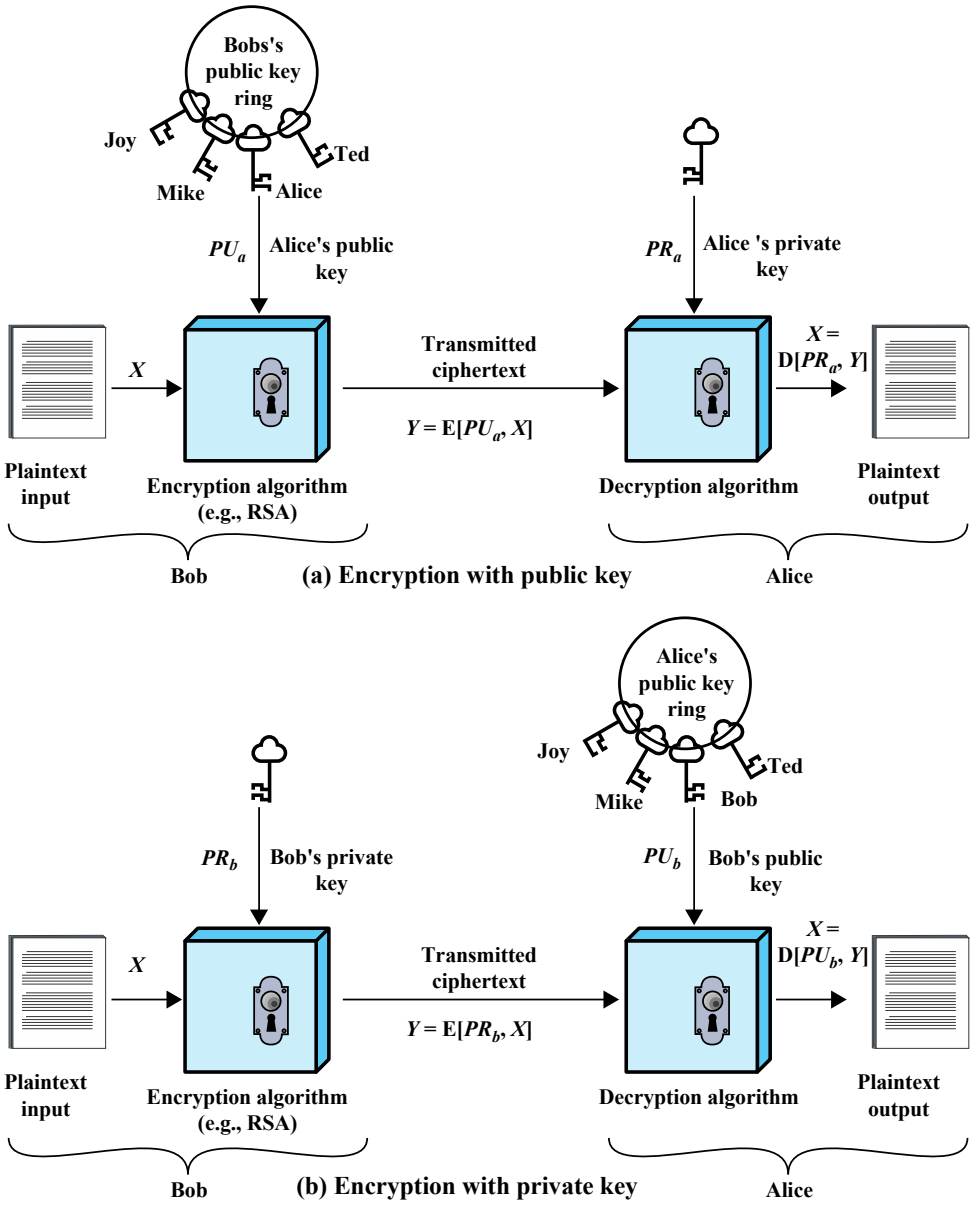
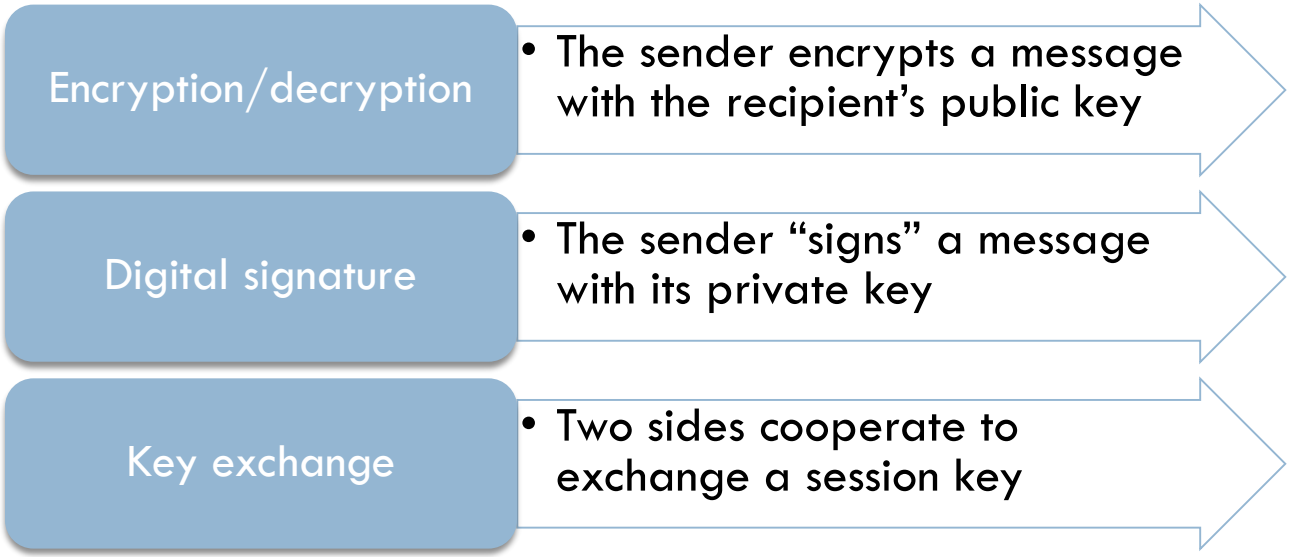


Figure 9.1 Public-Key Cryptography

Applications for Public-Key Cryptosystems

- Public-key cryptosystems can be classified into three categories:

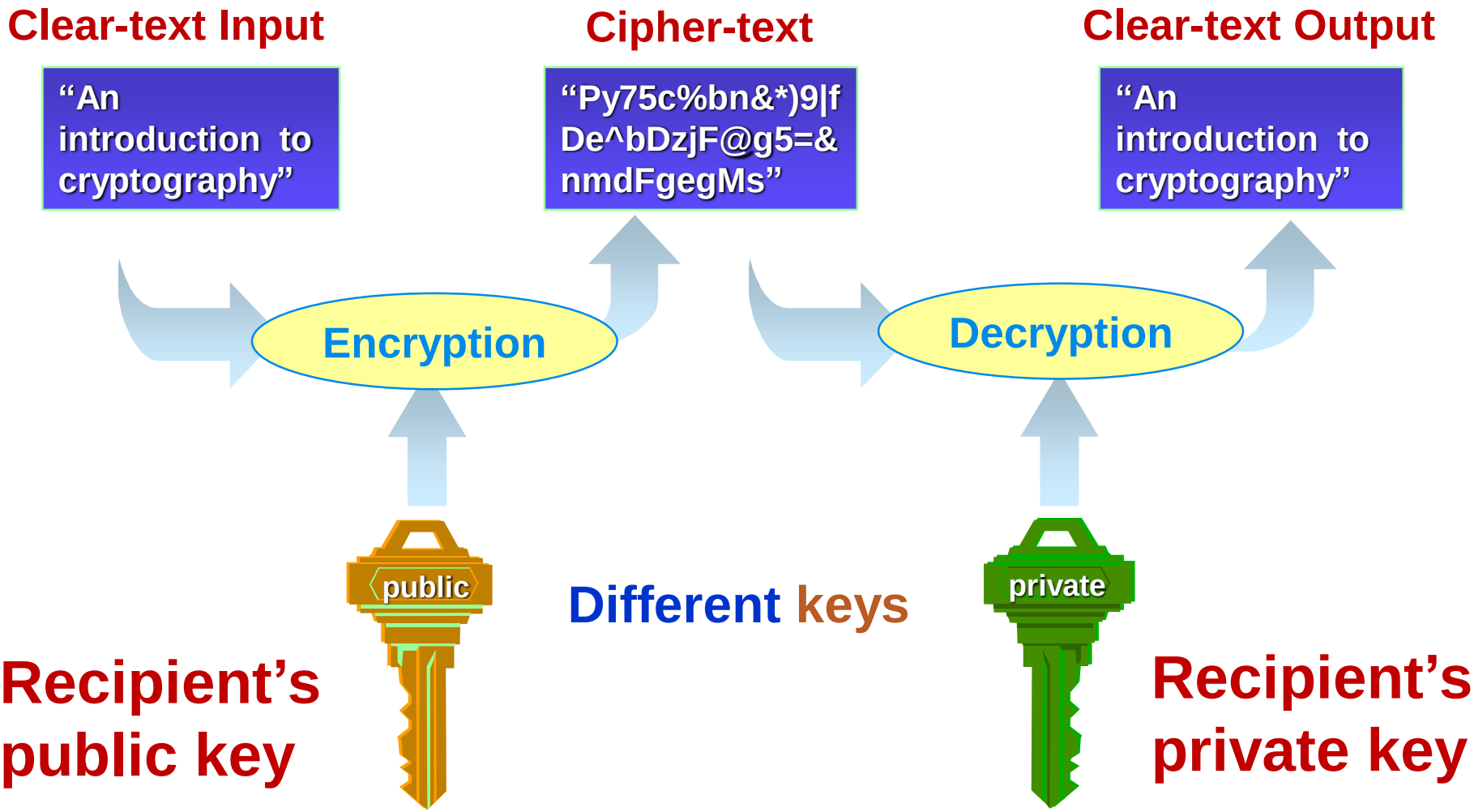


- Some algorithms are suitable for all three applications, whereas others can be used only for one or two

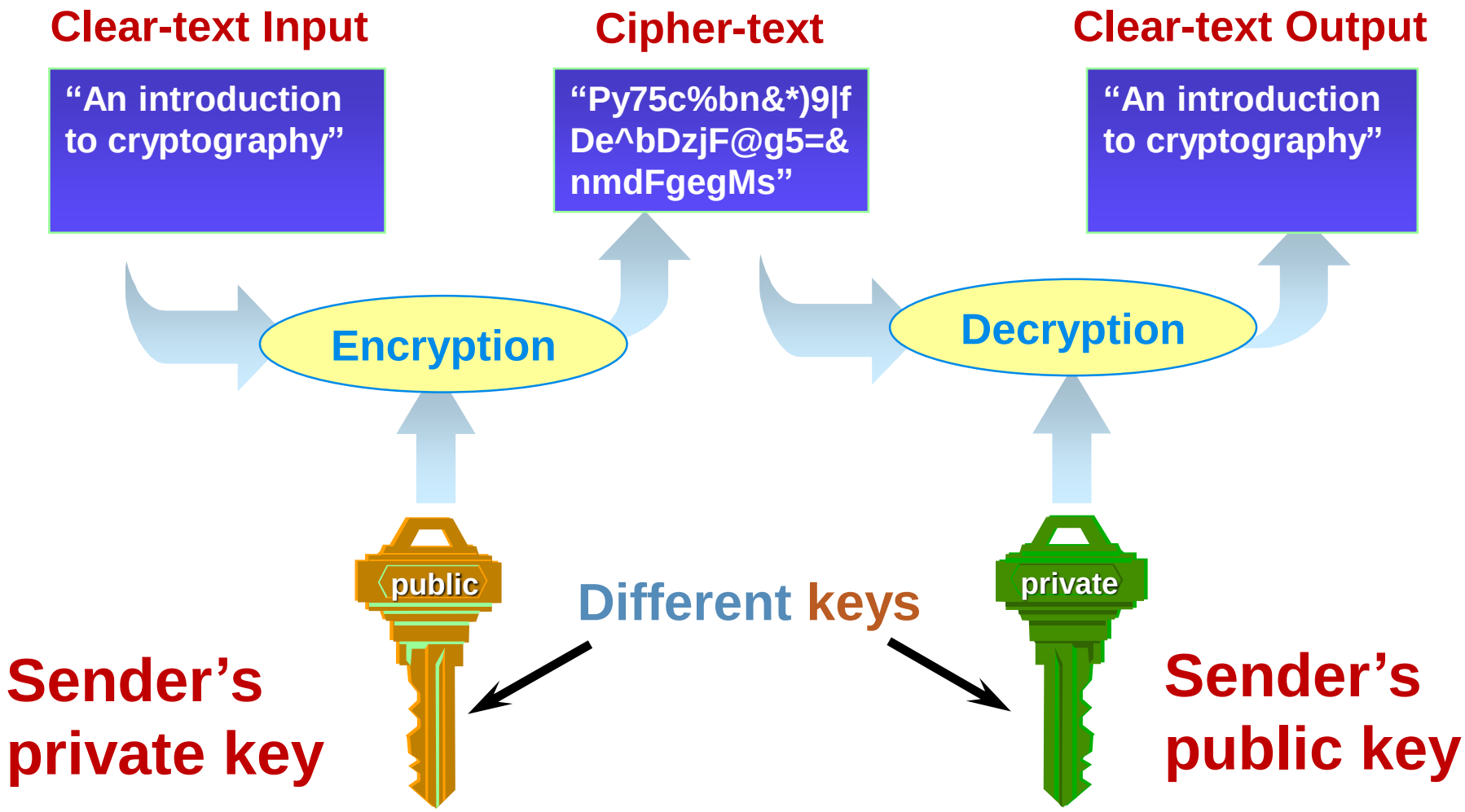
Applications for Public-Key Cryptosystems

Algorithm	Encryption/Decryption	Digital Signature	Key Exchange
RSA	Yes	Yes	Yes
Elliptic Curve	Yes	Yes	Yes
Diffie-Hellman	No	No	Yes
DSS	No	Yes	No

Example: Confidentiality



Example: Authenticity



Public-Key Cryptosystem: Authentication and Secrecy

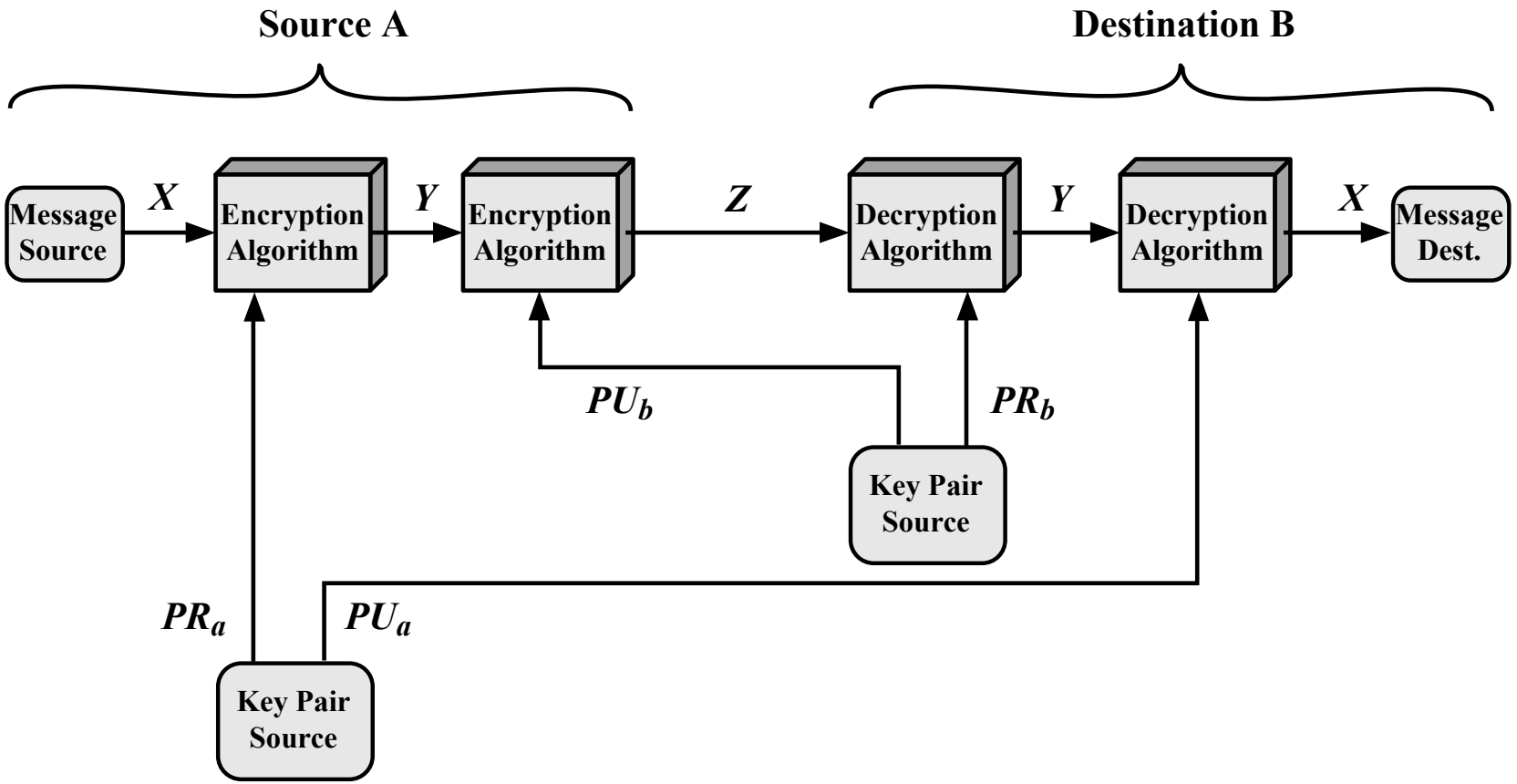


Figure 9.4 Public-Key Cryptosystem: Authentication and Secrecy

Asymmetric Digital Signature

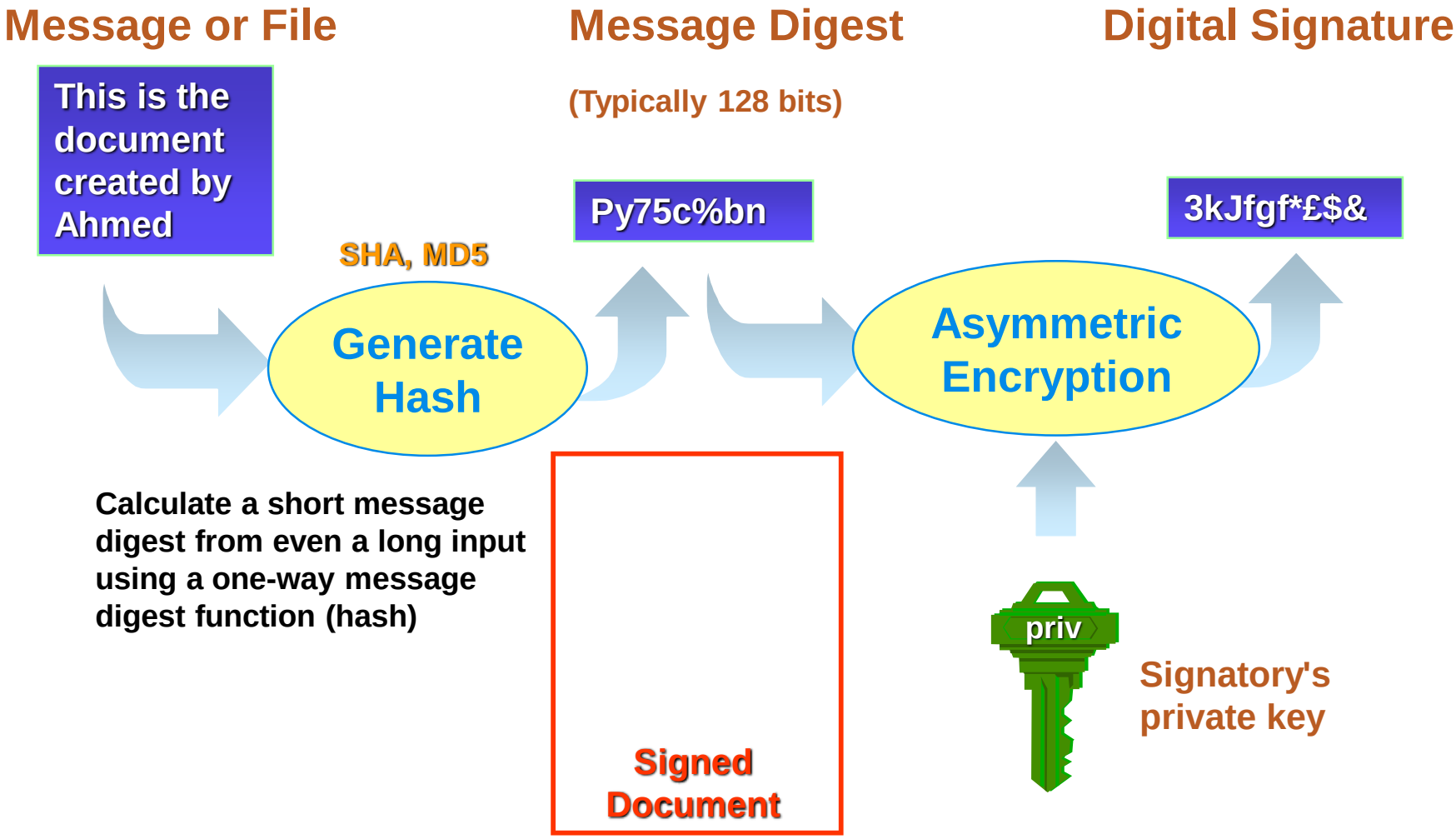
- Used for authenticating both source and data integrity
- A digital signature is "data appended to, or a cryptographic transformation of, a data unit, that allows a recipient of the data unit to prove the source, and integrity of the data unit and protect against forgery"
- Digital signatures make public key cryptography a most practical tool in real-life applications, being the most reliable method for authentication, data integrity and non-repudiation.
- Does not provide confidentiality
  - Even in the case of complete encryption
  - Message is safe from alteration but not eavesdropping

Asymmetric Digital Signature Model

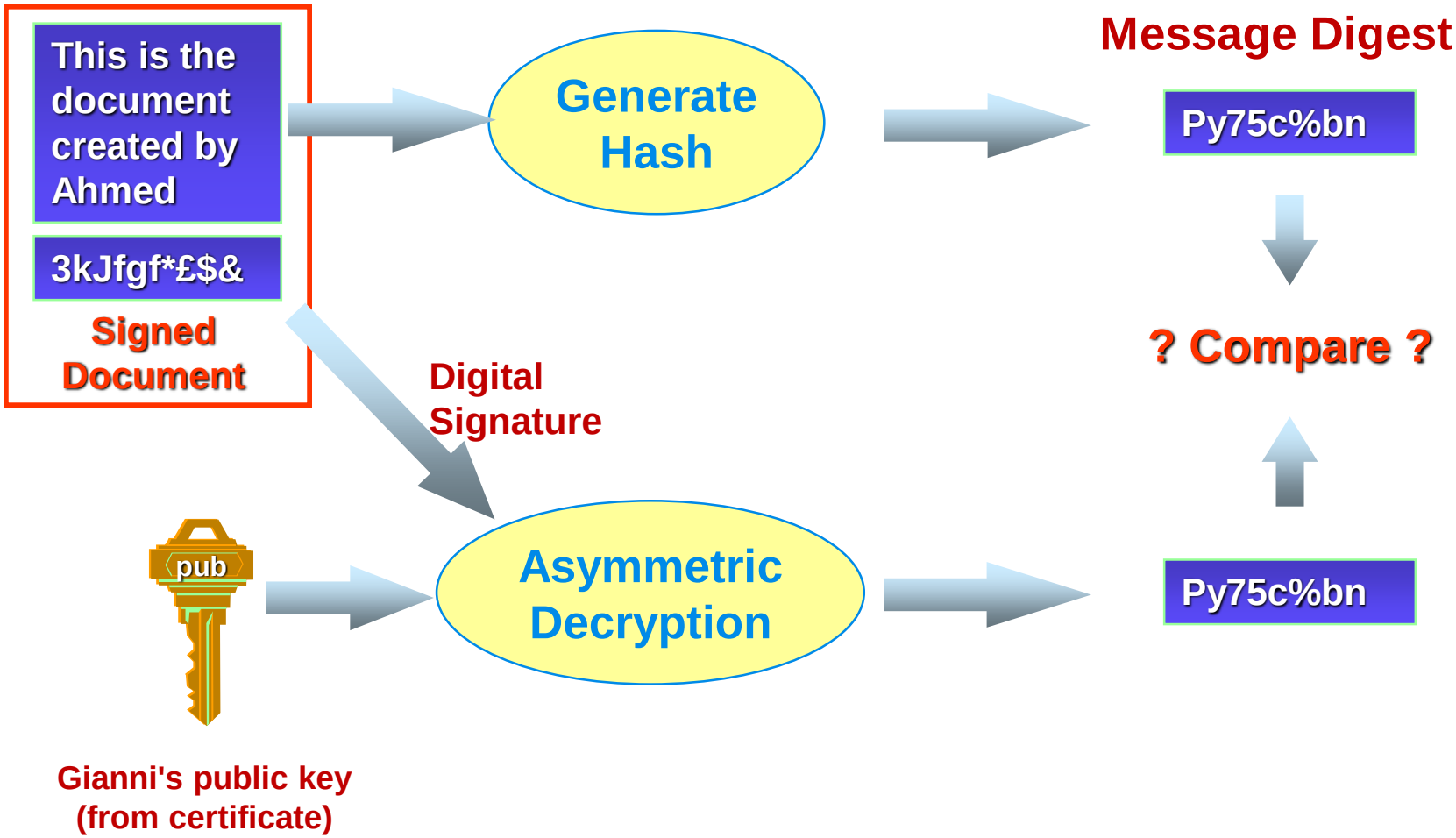
- A digital signature depends on two fundamental assumptions:
  - first, the private key is secure and only the owner of the key has access to it,
  - and second, the only way to produce a digital signature is to use the private key.
- The first assumption has no technical answer except that keys must be protected. But the second assumption can be examined from a mathematical point of view.

K_{PB}

Creating a Digital Signature



Verifying a Digital Signature



Properties of Digital Signature

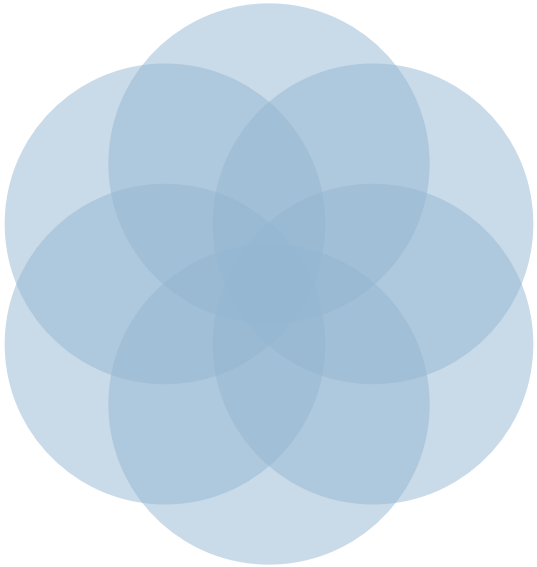
- It must have the following properties.
  - The signature must be verifiable by third parties, to resolve disputes.
  - It must be possible to verify the author, the date and time of the signature.
  - It must be possible to authenticate the contents at the time of the signature.

Public-Key Requirements

Computationally easy to create key pairs

Useful if either key can be used for each role

Computationally easy for sender knowing public key to encrypt messages



Computationally infeasible for opponent to otherwise recover original message

Computationally easy for receiver knowing private key to decrypt ciphertext

Computationally infeasible for opponent to determine private key from public key

Public-Key Requirements

- Conditions that these algorithms must fulfill:
  - It is computationally easy for a party B to generate a pair (public-key PU_b, private key PR_b)
  - It is computationally easy for a sender A, knowing the public key and the message to be encrypted, to generate the corresponding ciphertext $C = E(PU_b, M)$
  - It is computationally easy for the receiver B to decrypt the resulting ciphertext using the private key to recover the original message $M = D(PR_b, C) = D[PR_b, E(PU_b, M)]$
  - It is computationally infeasible for an adversary, knowing the public key, to determine the private key (PR_b)
  - It is computationally infeasible for an adversary, knowing the public key PU_b and a ciphertext,C, to recover the original message (M)
  - The two keys can be applied in either order $M = D[PU_b, E(PR_b, M)] = D[PR_b, E(PU_b, M)]$

Public-Key Requirements

- Need a trap-door one-way function
  - A one-way function is one that maps a domain into a range such that every function value has a unique inverse, with the condition that the calculation of the function is easy, whereas the calculation of the inverse is infeasible
    - Y = f(X) easy
    - X = f^{-1}(Y) infeasible
- A trap-door one-way function is a family of invertible functions f_k, such that
  - Y = f_k(X) easy, if k and X are known
  - X = f_k^{-1}(Y) easy, if k and Y are known
  - X = f_k^{-1}(Y) infeasible, if Y known but k not known
- A practical public-key scheme depends on a suitable trap-door one-way function

Public-Key Cryptanalysis

- A public-key encryption scheme is vulnerable to a brute-force attack
  - Countermeasure: use large keys
  - Key size must be small enough for practical encryption and decryption
  - Key sizes that have been proposed result in encryption/decryption speeds that are too slow for general-purpose use
  - Public-key encryption is currently confined to key management and signature applications
- Another form of attack is to find some way to compute the private key given the public key
  - To date it has not been mathematically proven that this form of attack is infeasible for a particular public-key algorithm

Asymmetric Encryption Algorithms

RSA (Rivest, Shamir, Adleman)	Developed in 1977	Most widely accepted and implemented approach to public-key encryption	Block cipher in which the plaintext and ciphertext are integers between 0 and *n*-1 for some *n*.
Diffie-Hellman key exchange algorithm	Enables two users to securely reach agreement about a shared secret that can be used as a secret key for subsequent symmetric encryption of messages	Limited to the exchange of the keys	
Digital Signature Standard (DSS)	Provides only a digital signature function with SHA-1	Cannot be used for encryption or key exchange	
Elliptic curve cryptography (ECC)	Security like RSA, but with much smaller keys		

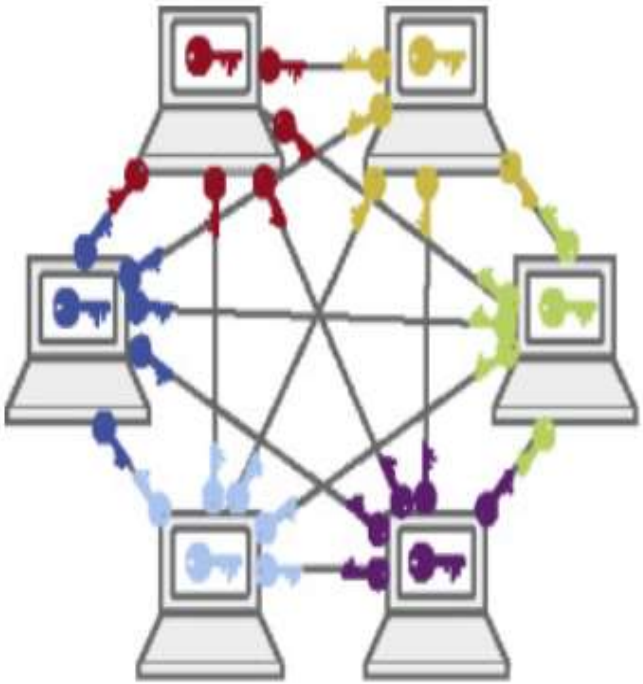
Symmetric-key
vs.
Asymmetric-key cryptography

Conventional and Public-Key Encryption

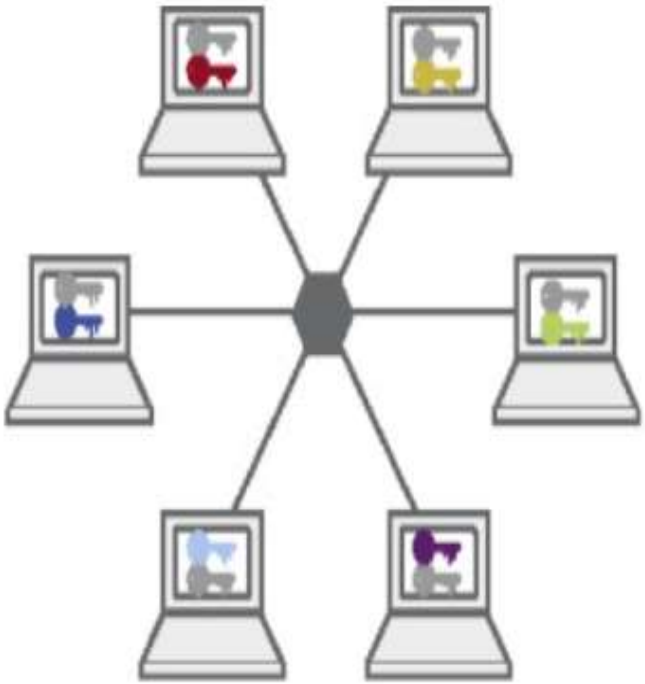
Conventional Encryption	Public-Key Encryption
Needed to Work: 1. The same algorithm with the same key is used for encryption and decryption. 2. The sender and receiver must share the algorithm and the key. *Needed for Security:* 1. The key must be kept secret. 2. It must be impossible or at least impractical to decipher a message if the key is kept secret. 3. Knowledge of the algorithm plus samples of ciphertext must be insufficient to determine the key.	*Needed to Work:* 1. One algorithm is used for encryption and a related algorithm for decryption with a pair of keys, one for encryption and one for decryption. 2. The sender and receiver must each have one of the matched pair of keys (not the same one). *Needed for Security:* 1. One of the two keys must be kept secret. 2. It must be impossible or at least impractical to decipher a message if one of the keys is kept secret. 3. Knowledge of the algorithm plus one of the keys plus samples of ciphertext must be insufficient to determine the other key.

Disadvantages of symmetric-key

- In a two-party communication, the key must remain secret at both ends. In order to use a secure channel, it requires prior communication of the key between sender and receiver before any ciphertext is transmitted. In practice, this may be very difficult to achieve, because there is no security channel in wireless communication systems.
- In a large network, there are many key pairs to be managed. It requires a large amount of keys. For a cryptosystem with n users, since each user has to possess n-1 keys, the required total number of keys are n(n-1)/2. Thus, when the number of users increases, the risk of revealing the secret information was drastically increased.



SYMMETRIC



ASYMMETRIC

Advantages of symmetric-key

- Have high rates of data throughput .
- Keys for symmetric-key ciphers are relatively short.

Advantages of Asymmetric-key

- The concept of asymmetric-key cryptography evolved from an attempt to solve two of the most difficult problems associated with conventional encryption.
  - Key distribution.
  - Many public-key schemes yield relatively efficient digital signature mechanisms
- Only the private key must be kept secret.
- Depending on the mode of usage, a private key/public key pair may remain unchanged for considerable periods of time.

Disadvantages of Asymmetric-key

- Slower than the best known symmetric-key schemes.
- Key sizes are typically much larger.

Public-key Certificates

What if you do not have a website to publish your key?

What if someone who is not on your list of contacts wants to send you a private message?

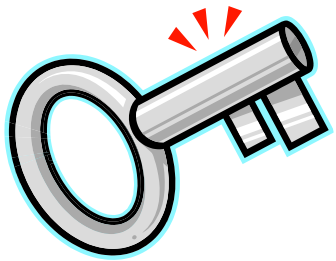
What if you want to change your key because it is become compromised in some way?

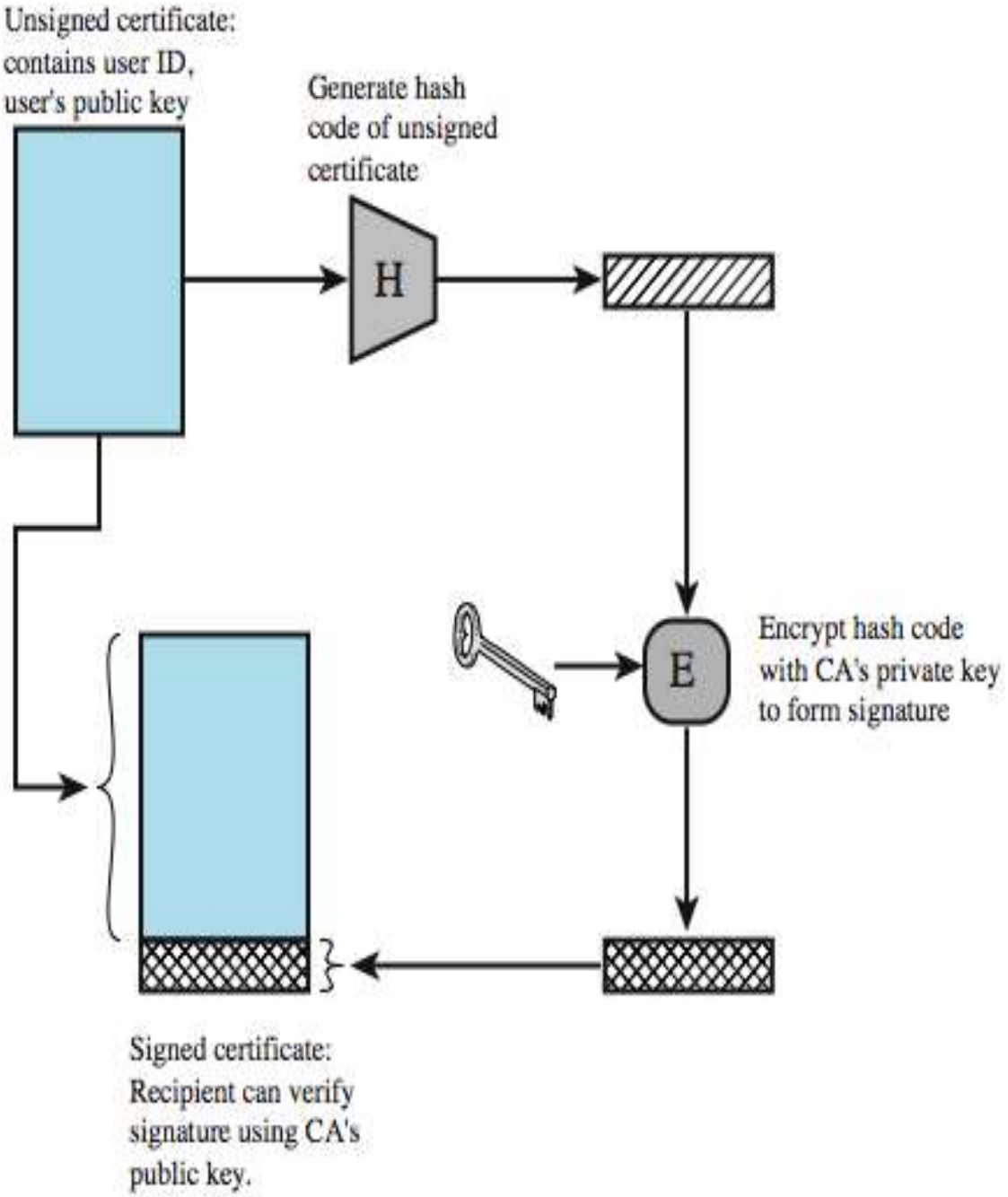
What if someone else published a key and claimed that it was yours?

Public-key Certificates

- **A Certification Authority** (CA) acts as a trusted third party with the role of issuing **digital certificates** that bind individuals' identities to their public keys.
  - ***Digital certificates*** *are analogous to passports, and Certification Authorities are like Passport Authorities.*
- **There is a hierarchy of Certification Authorities,** the most trusted and influential being the **Root Certification Authorities**
  - Examples of Root Certification Authorities: VeriSign; Thawte
- Some are able to issue keys as well as digital certificates; all must maintain a register of revoked certificates.

Public Key Certificates





Public-key Certificates

- **A digital certificate will typically include:**
  - **A copy of the public key**
  - **Information about the owner of the key:** the owner's name, etc.
  - **Information about the digital certificate:** a serial number, expiry date, etc.
  - **Information about the CA itself:** CA name, its own digital signature, etc.
- **A digital certificate needs to conform to a recognised standard**
  - The standard specifies what information should be included and how that information should appear.
    - The X.509 standard published by the ITU is the most commonly used.

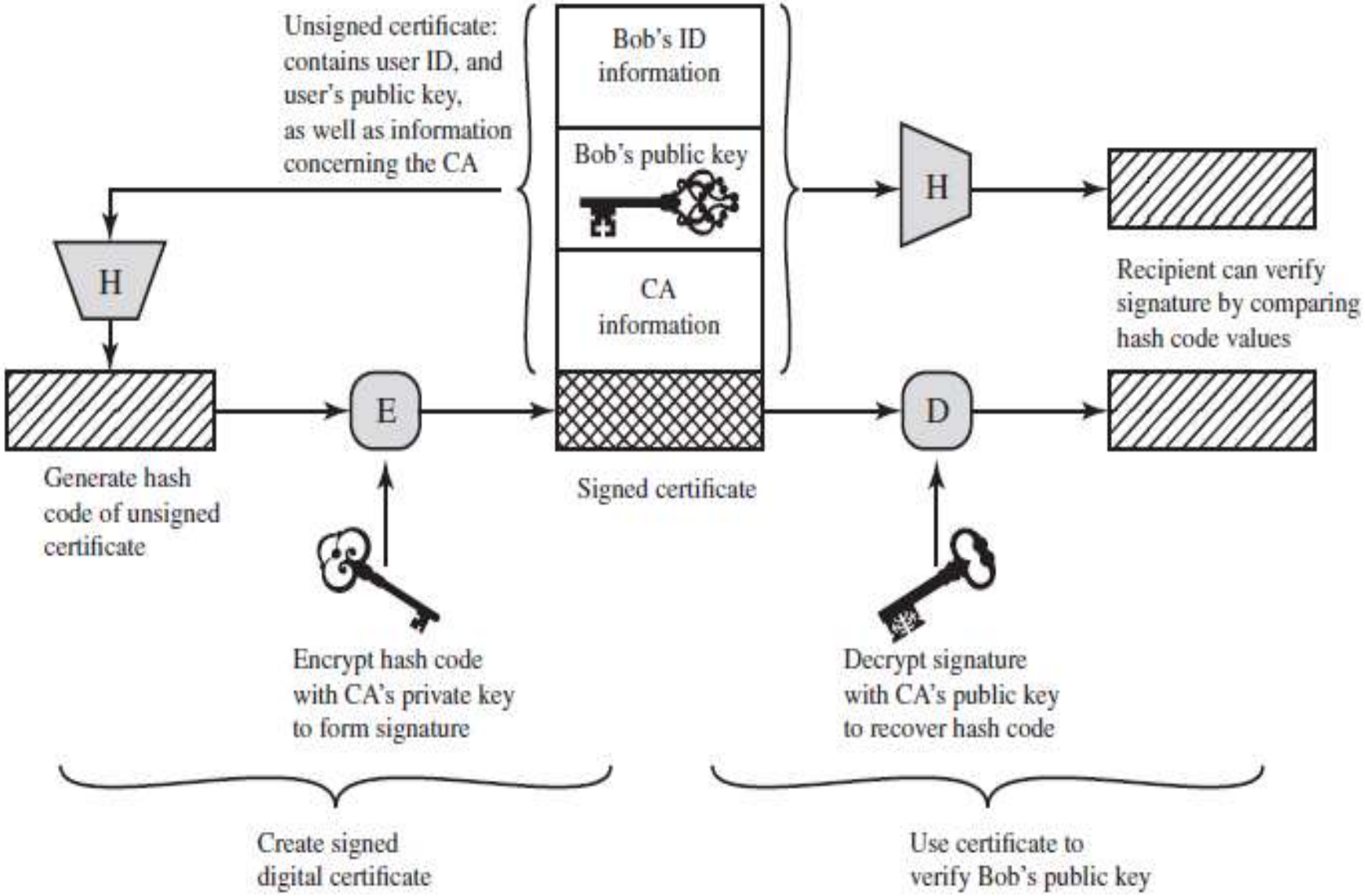
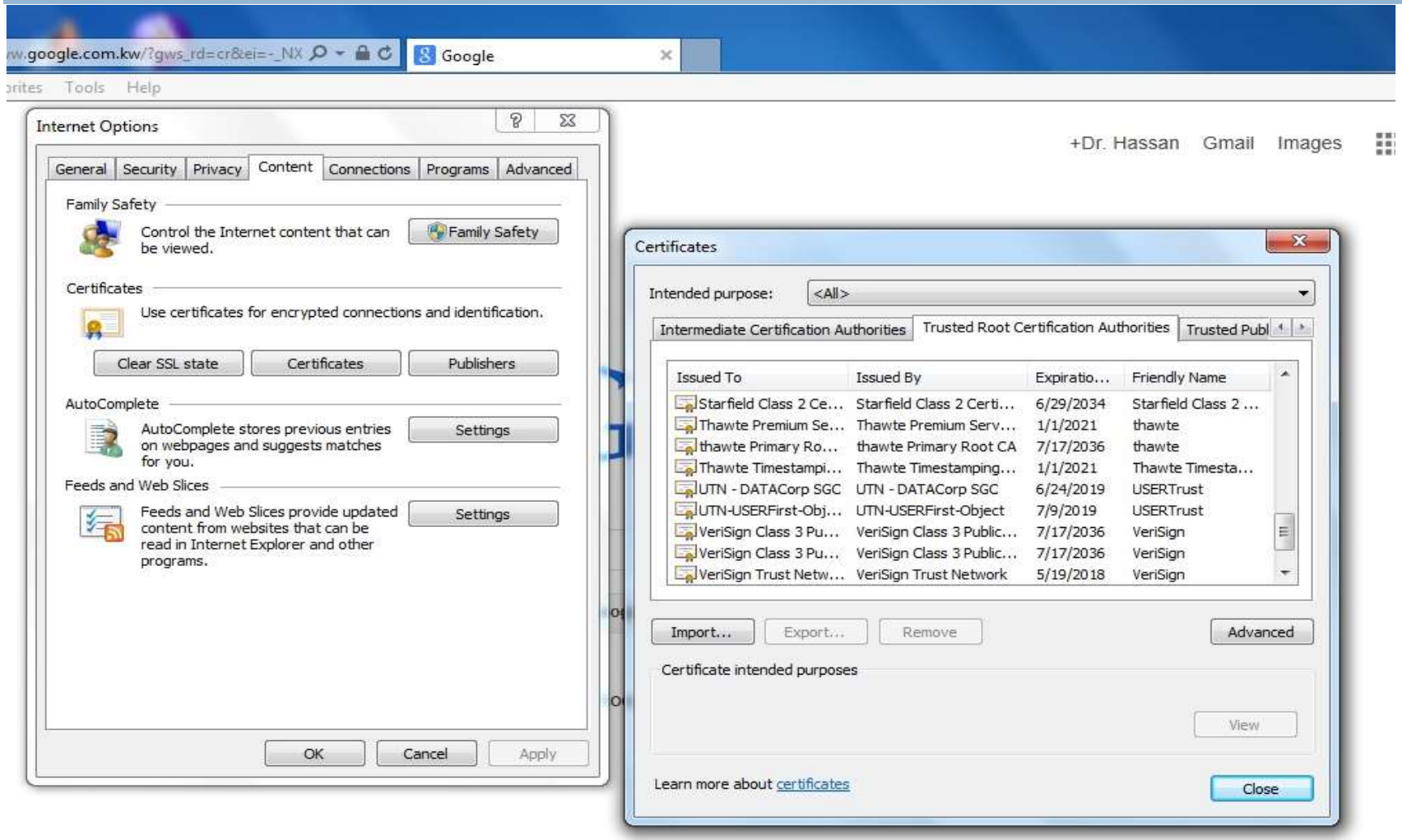


Figure 2.7 **Public-Key Certificate Use**

Public-key Certificates



Public-key Certificates

- The key steps can be summarized as follows:
  1. User software (client) creates a pair of keys: one public and one private.
  2. Client prepares an unsigned certificate that includes the user ID and user's public key.
  3. User provides the unsigned certificate to a CA in some secure manner. This might require a face-to-face meeting, the use of registered e-mail, or happen via a web form with e-mail verification.
  4. CA creates a signature as follows:
     A. CA uses a hash function to calculate the hash code of the unsigned certificate. A hash function is one that maps a variable-length data block or message into a fixed-length value called a hash code, such as SHA family.
     B. CA encrypts the hash code with the CA's private key.
  5. CA attaches the signature to the unsigned certificate to create a signed certificate.
  6. CA returns the signed certificate to client.
  7. Client may provide the signed certificate to any other user.

Public-key Certificates

8. **Any user may verify that the certificate is valid as follows:**
   A. User calculates the hash code of certificate (not including signature).
   B. User decrypts the signature using CA's known public key.
   C. User compares the results of (a) and (b). If there is a match, the certificate is valid.

- One scheme has become universally accepted for formatting public-key certificates: the X.509 standard. X.509 certificates are used in most network security applications, including IP Security (IPsec), Transport Layer Security (TLS), Secure Shell (SSH), and Secure/Multipurpose Internet Mail Extension (S/MIME).

Symmetric Key Exchange Using Public-key Encryption

- **Drawbacks of a Public key systems:**
  - The keys are long in order to provide the required resistance to cryptanalysis
  - This imposes a processing overhead (processing is more complicated)
  - The time needed for encryption/decryption increases (compared with symmetric systems).
- **In practice, It is rare, to use an asymmetric key system for the encryption of message data.**
  - Usually a short symmetric key, known as **sessions key** is used by both parties (for instance, Alice and Bob) to encrypt their secret messages
  - **Asymmetric systems are commonly used for key exchange**

Symmetric Key Exchange Using Public-key Encryption

- **Using an asymmetric systems for key exchange:**
  - One of the parties, for instance, Alice generates the **session key- temporary key**
  - Alice uses the second party's public key, for instance, Bob's public key is to encrypt the session key
  - Alice sends the encrypted session key to Bob
  - Bob decrypts the encrypted session key using his own private key
  - Both parties (Alice and Bob) share the same symmetric session key
  - The session key is used for encrypting their messages.

Symmetric Key Exchange Using Public-key Encryption

- **Advantages of a Session Key:**
  - A session key is short.
  - A session key imposes a much lower processing overhead than public key systems.
- **Drawbacks of a session key:**
  - Low resistance to attack
  - Session keys usually stay in service for a relatively short time (sometimes only a single transaction) before being discarded.
    - if they are compromised their short service life limits the number of messages that are at risk.

Symmetric Key Exchange Using Public-key Encryption

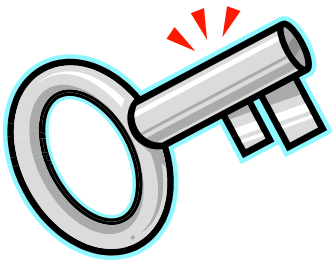
- One approach is the use of Diffie-Hellman key exchange. This approach is indeed widely used.
- However, it suffers the drawback that, in its simplest form, Diffie-Hellman provides no authentication of the two communicating partners.
- There are variations to Diffie-Hellman that overcome this problem. Also, there are protocols using other public-key algorithms that achieve the same objective

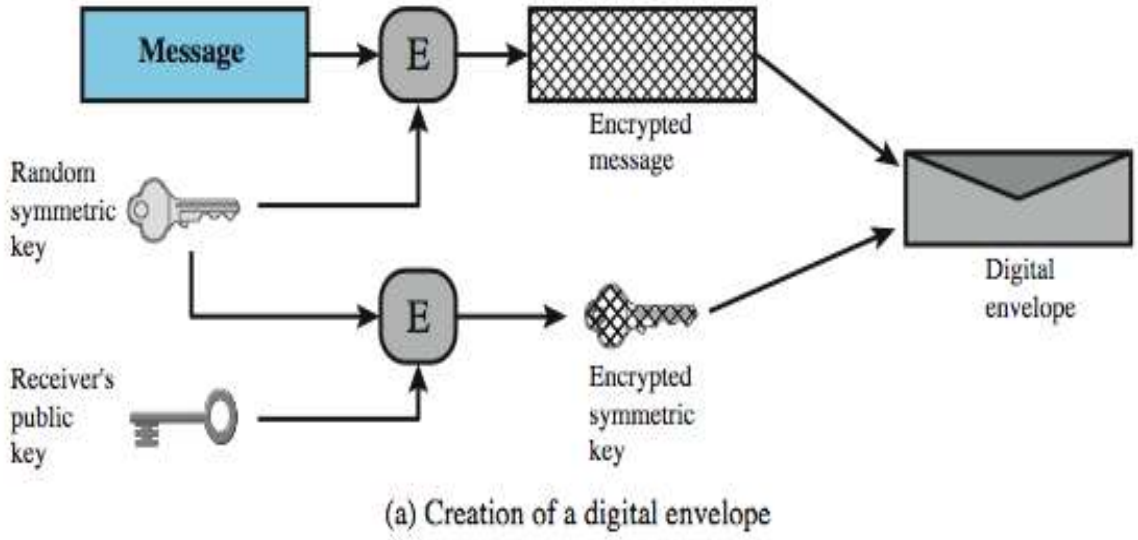
Digital Envelopes

- Another application in which public-key encryption is used to protect a symmetric key is the digital envelope
  - Protects a message without needing to first arrange for sender and receiver to have the same secret key
- The technique is referred to as a digital envelope, which is the equivalent of a sealed envelope containing an unsigned letter.

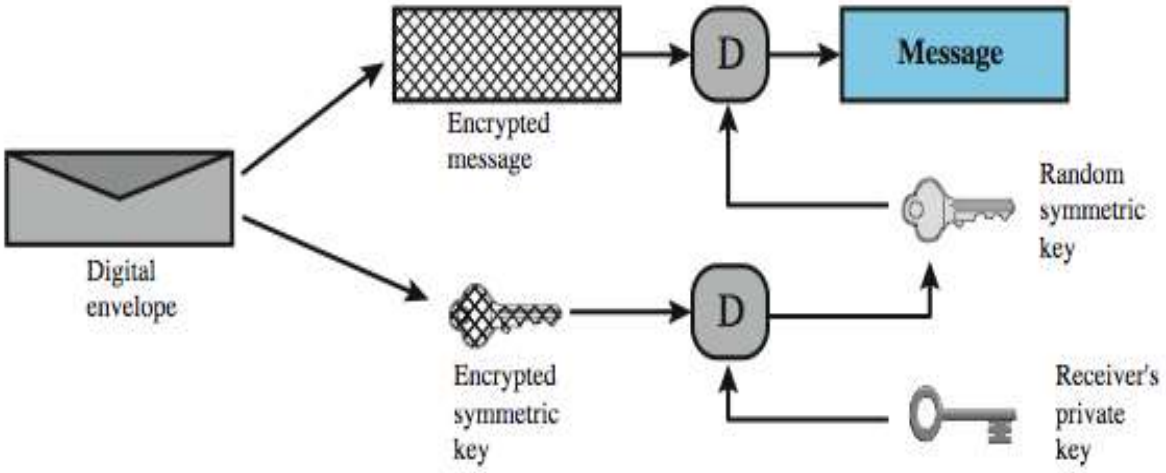
Digital Envelopes

Another application of public key alg





(a) Creation of a digital envelope



(b) Opening a digital envelope

Summary

- introduced cryptographic algorithms
- symmetric encryption algorithms for confidentiality
- message authentication & hash functions
- public-key encryption
- digital signatures and key management
- random numbers