

Computer Security: Principles and Practice

Fourth Edition

By: William Stallings and Lawrie Brown

Chapter 2

Cryptographic Tools

Symmetric Encryption

- The universal technique for providing confidentiality for transmitted or stored data
- Also referred to as:
 - conventional encryption or
 - single-key encryption or
 - secret key encryption
- Two requirements for secure use:
 - Need a **strong encryption** algorithm
 - Sender and receiver must have obtained copies of the **secret key in a secure fashion** and must keep the key secure

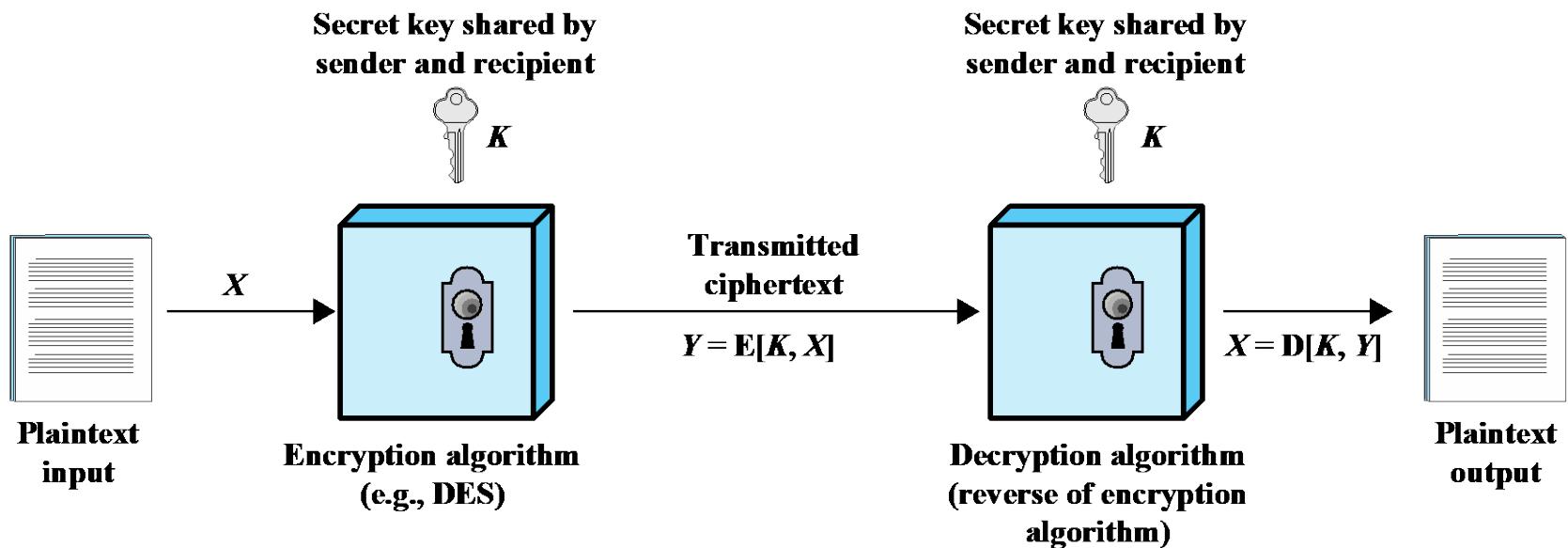


Figure 2.1 Simplified Model of Symmetric Encryption

Symmetric Encryption

- Secret key cryptography between Alice and Bob
 - Alice creates a secret key K and sends it to Bob
 - Alice and Bob now both have the same symmetric key
 - Alice generates a message M
 - She encrypts the message with K and forwards it to Bob
 - Bob uses the secret key K to decrypt the message, and gets the message M



Problem?

$(M)_K$



Secret
key K

Secret
key K

Symmetric Encryption

- Mike can intercept the key K!!!
- And the encrypted message!!! $(M)_K$
- And is then also able to decrypt the message!!!



Secret
key K

Secret
key K

Symmetric Encryption

- Considerations
 - Performance
 - Fast
 - Typical 128 to 256 bit key
 - Key distribution problem
 - Simple architecture

Attacking Symmetric Encryption

Cryptanalytic Attacks

- Rely on:
 - Nature of the algorithm
 - Some knowledge of the general characteristics of the plaintext
 - Some sample plaintext-ciphertext pairs
- Exploits the characteristics of the algorithm to attempt to deduce a specific plaintext or the key being used
 - If successful all future and past messages encrypted with that key are compromised

Brute-Force Attacks

- Try all possible keys on some ciphertext until an intelligible translation into plaintext is obtained
 - On average half of all possible keys must be tried to achieve success

Table 2.1

| | 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

Comparison of Three Popular Symmetric
Encryption Algorithms

Data Encryption Standard (DES)

- 
- Until recently (2008) was the most widely used encryption scheme
 - FIPS PUB 46
 - Referred to as the Data Encryption Algorithm (DEA) or Data Encryption Standard (DES)
 - Uses 64 bit plaintext block and 56 bit key to produce a 64 bit ciphertext block
 - Strength concerns:
 - Concerns about the algorithm itself
 - DES is the most studied encryption algorithm in existence
 - Concerns about the use of a 56-bit key
 - The speed of commercial off-the-shelf processors makes this key length woefully inadequate

Table 2.2

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

Average Time Required for Exhaustive Key Search

Triple DES (3DES)

- Repeats basic DES algorithm three times using either two or three unique keys
- 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
 - Uses a 64-bit block size

Advanced Encryption Standard (AES)

Needed a replacement for 3DES

3DES was not reasonable for long term use

NIST called for proposals for a new AES in 1997

Should have a security strength equal to or better than 3DES

Significantly improved efficiency

Symmetric block cipher

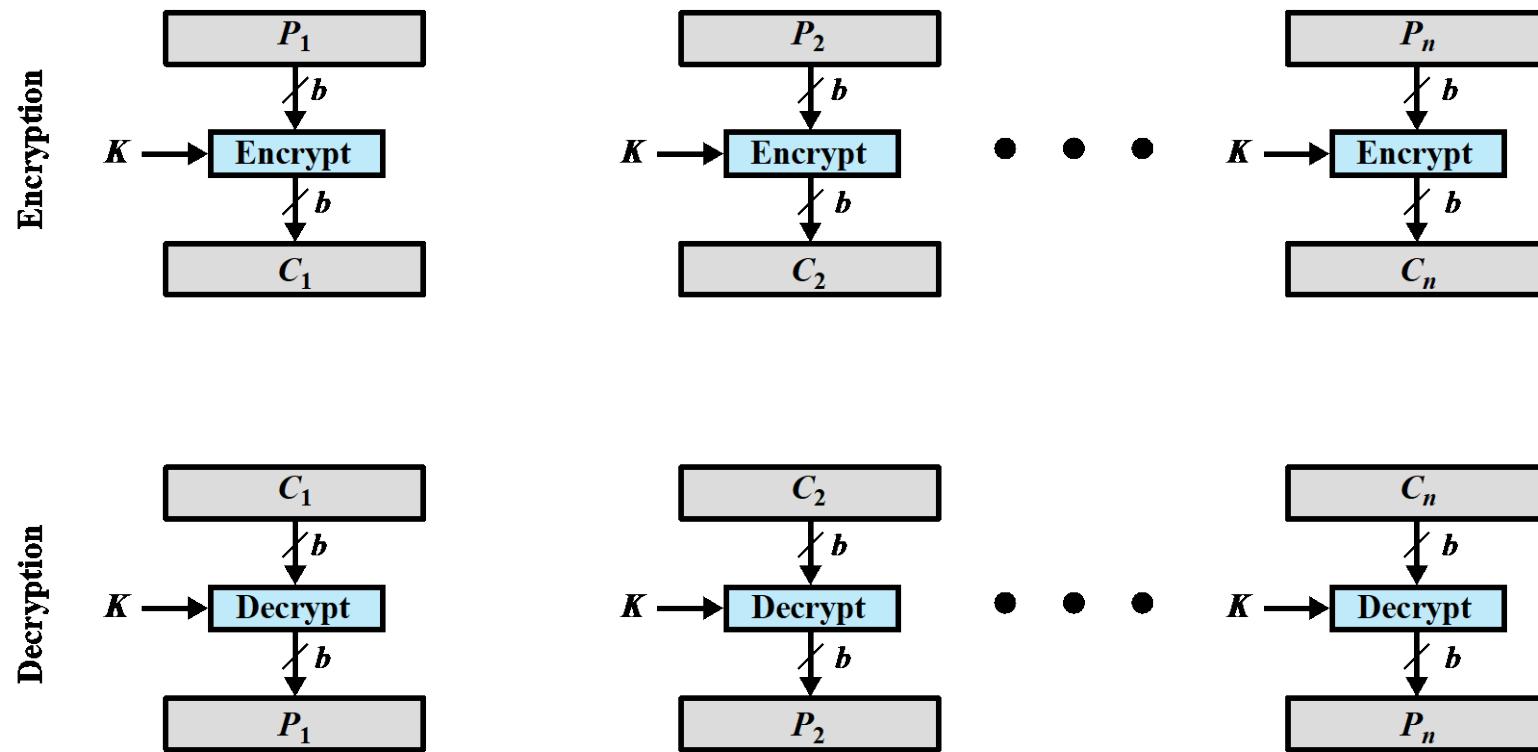
128 bit data and 128/192/256 bit keys

Selected Rijndael in November 2001

Published as FIPS 197

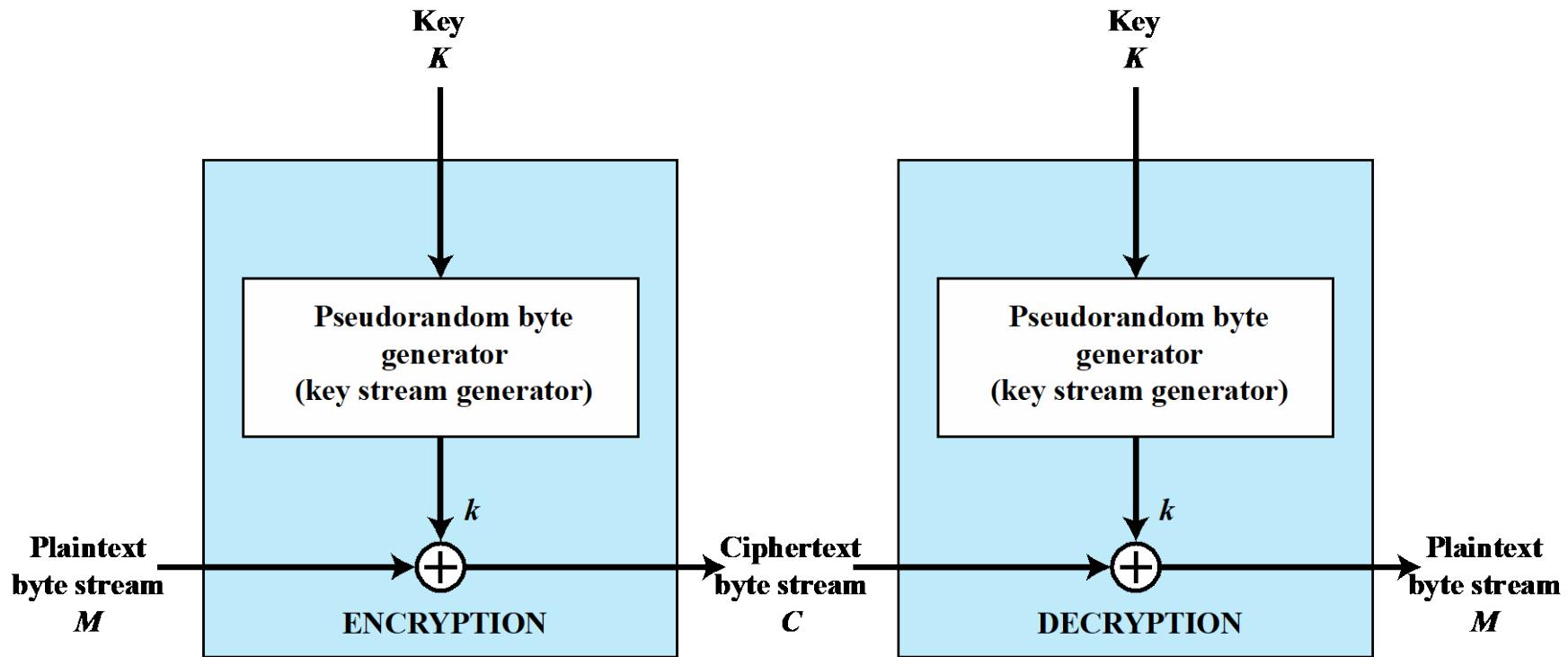
Practical Security Issues

- 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
 - Not secure for large messages, because cryptanalysts may be able to exploit regularities in the plaintext
- Solved problem with: Modes of operation
 - Alternative techniques developed to increase the security of symmetric block encryption for large sequences
 - Overcomes the weaknesses of ECB (CH20, not for this course)



(a) Block cipher encryption (electronic codebook mode)

Figure 2.2 Types of Symmetric Encryption



(b) Stream encryption

Figure 2.2 Types of Symmetric Encryption

Block & Stream Ciphers

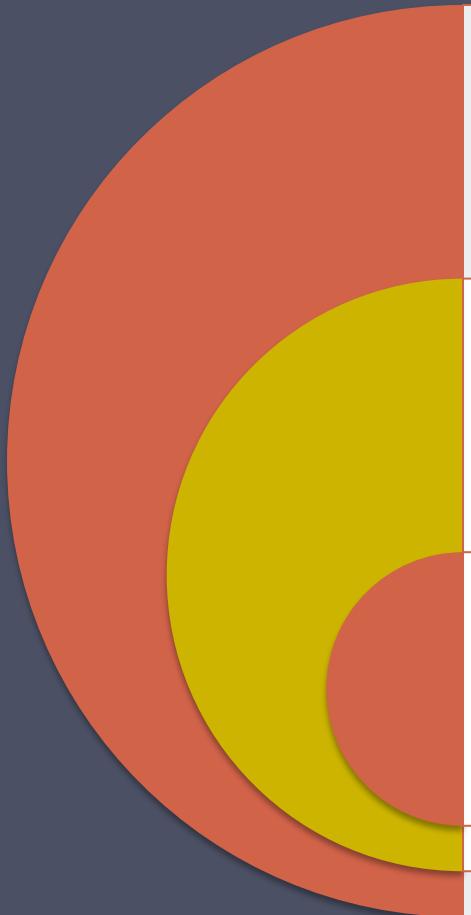
Block Cipher

- 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

- Processes the input elements continuously
- Encrypts plaintext one byte at a time
- Primary advantage is that they are almost always faster and use far less code
- Pseudorandom stream is one that is unpredictable without knowledge of the input key
- Examples: measuring atmospheric noise, thermal noise, cosmic background radiation, radioactive decay over short time

Message Authentication



Protects against
active attacks

Verifies received
message is
authentic

Can use
conventional
encryption

- Contents not been altered
- Produces a MAC (hash)
- From authentic source
- Timely and in correct sequence

- Only if confidentiality is required
- Only sender and receiver share a key

Message Authentication Without Confidentiality

- Message encryption by itself does not provide a secure form of message authentication
- It is possible to combine authentication and confidentiality in a single algorithm by encrypting a message plus its authentication tag
- Typically message authentication is provided as a separate function from message encryption (message confidentiality)

Message Authentication Without Confidentiality

- Situations in which message authentication without confidentiality may be preferable include:
 - There are a number of applications in which the same message is broadcast to a number of destinations
 - An exchange in which one side has a heavy load and cannot afford the time to decrypt all incoming messages
 - Authentication of a computer program in plaintext is an attractive service
- Thus, there is a place for both authentication and encryption in meeting security requirements

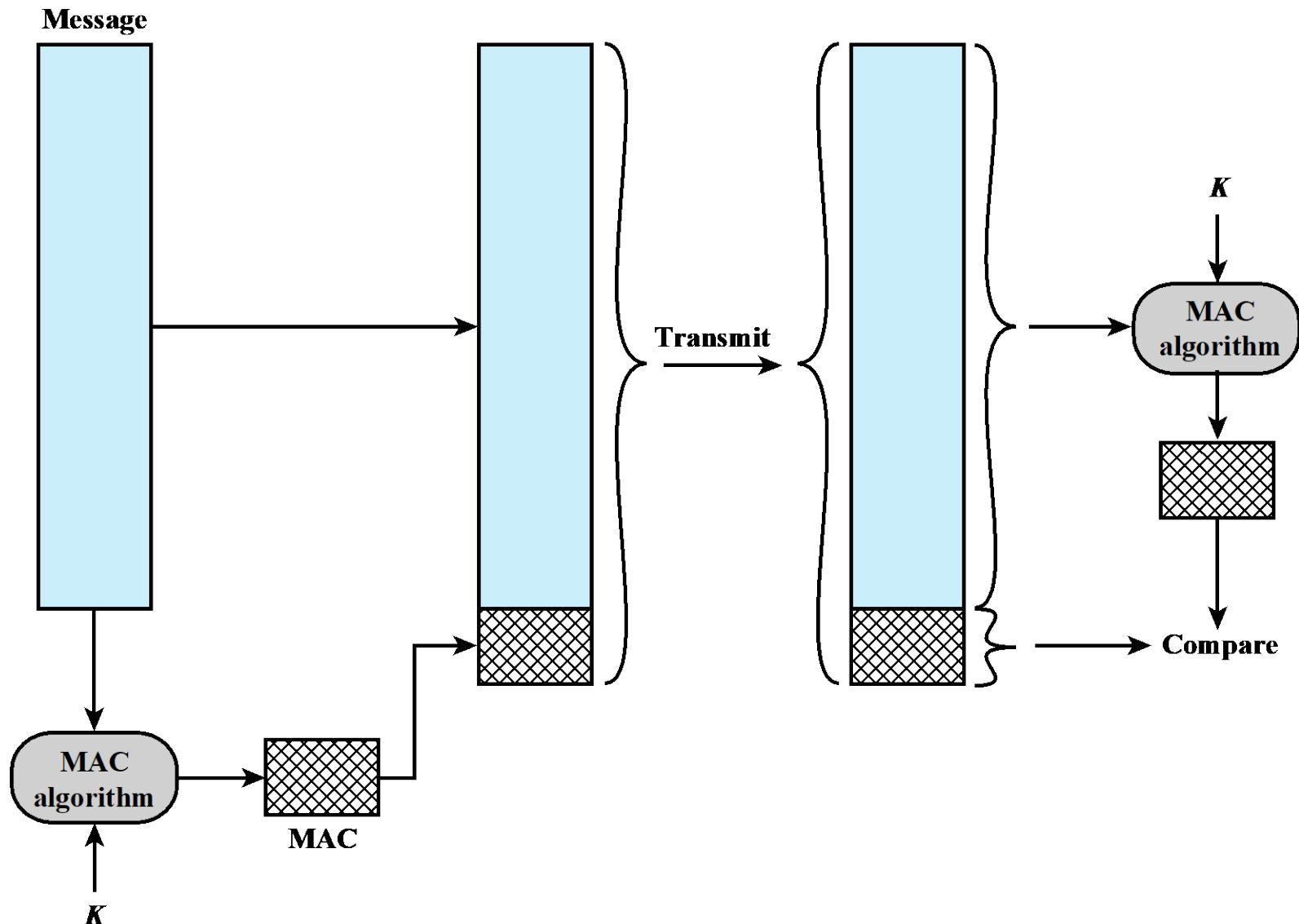


Figure 2.3 Message Authentication Using a Message Authentication Code (MAC).

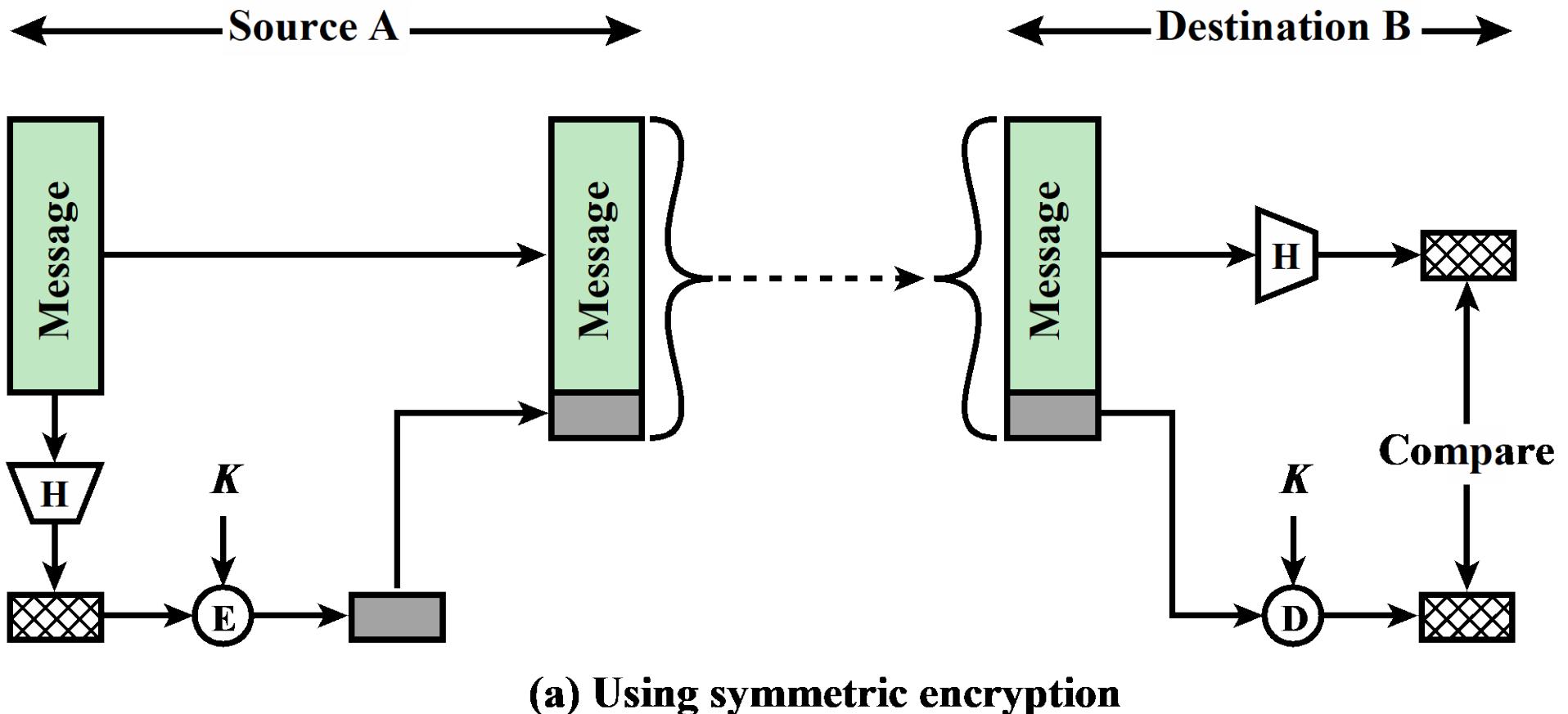


Figure 2.5 Message Authentication Using a One-Way Hash Function.

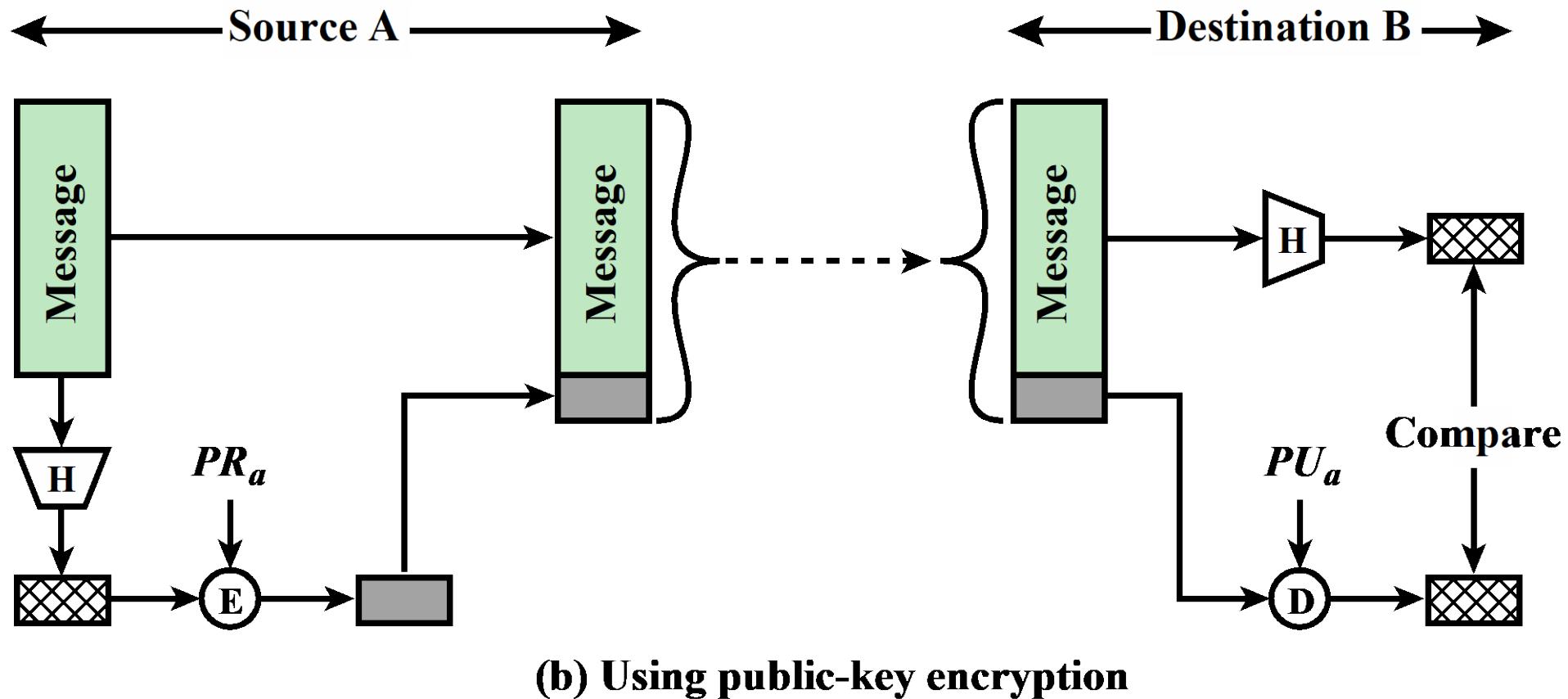
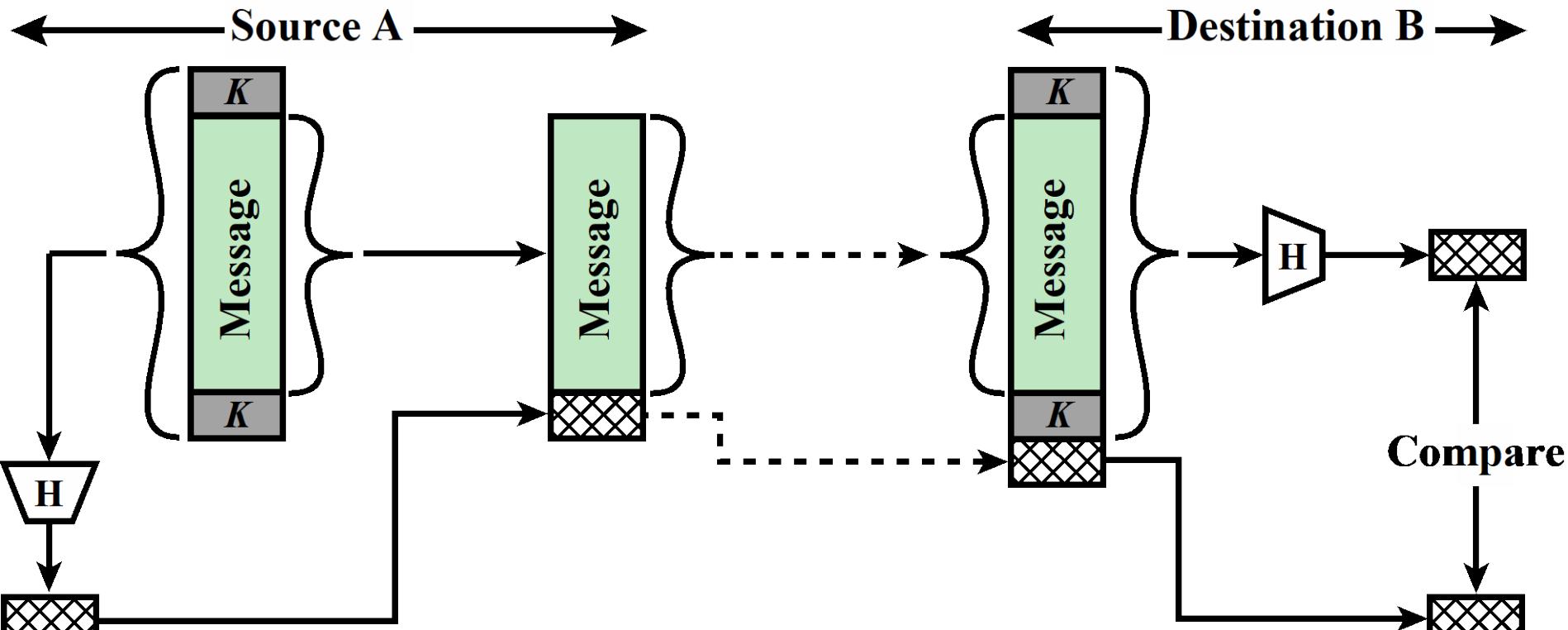


Figure 2.5 Message Authentication Using a One-Way Hash Function.



(c) Using secret value

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

To be useful for message authentication, a hash function H must have the following properties:

Can be applied to a block of data of any size

Produces a fixed-length output

$H(x)$ is relatively easy to compute for any given x

One-way or pre-image resistant

- Computationally infeasible to find x such that $H(x) = h$ (to derive plaintext from hash)

Second pre-image resistant

Computationally infeasible to find $y \neq x$ such that $H(y) = H(x)$ (i.e. to have the same hash for two different messages)

Collision resistant or strong collision resistance

- Computationally infeasible to find any pair (x,y) such that $H(x) = H(y)$ (to derive different messages from the same hash)

Security of Hash Functions

SHA most widely used hash algorithm

There are two approaches to attacking a secure hash function:

Additional secure hash function applications:

Cryptanalysis

- Exploit logical weaknesses in the algorithm

Passwords

- Hash of a password is stored by an operating system

Brute-force attack

- Strength of hash function depends solely on the length of the hash code produced by the algorithm

Intrusion detection

- Store $H(F)$ for each file on a system and secure the hash values
- Digital Forensics: forensic sound copies

Public-Key Encryption Structure

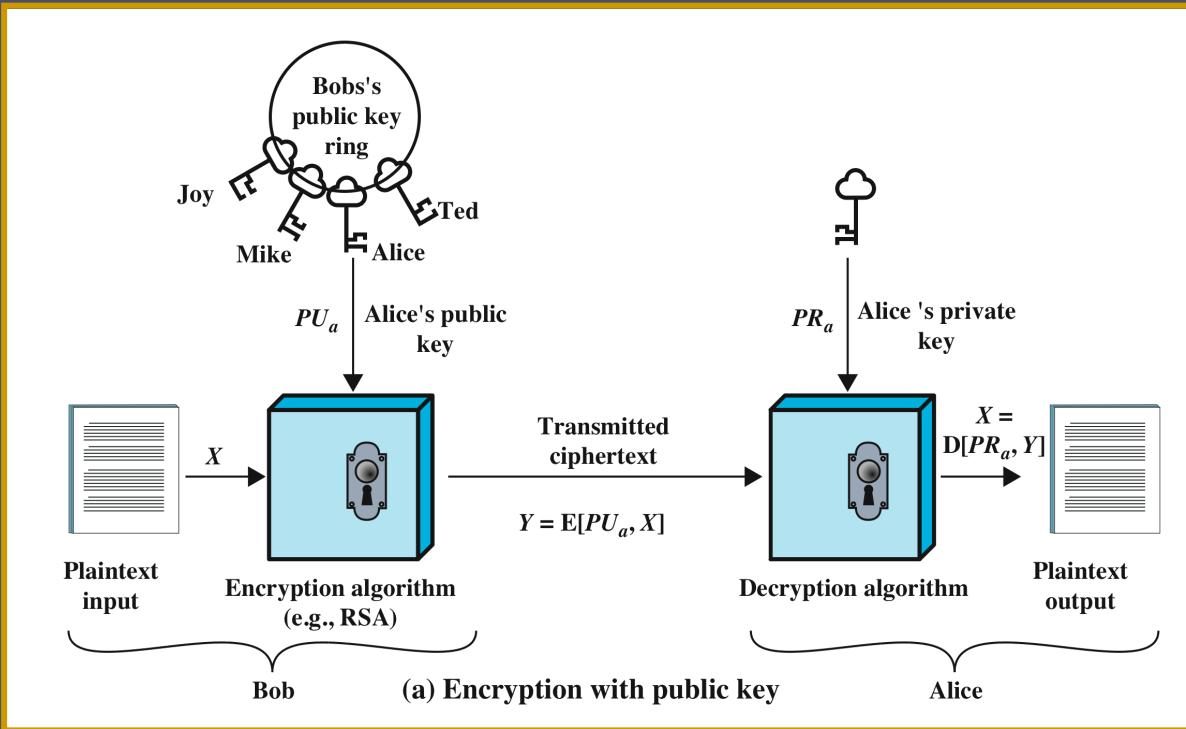
Publicly proposed by Diffie and Hellman in 1976

Based on mathematical functions (strong maths)

Asymmetric

- Uses two separate keys
- Public key and private key
- Public key is made public for others to use

Can solve the symmetric key distribution problem



● Plaintext

- Readable message or data that is fed into the algorithm as input

● Encryption algorithm

- Performs transformations on the plaintext

● Public and private key

- Pair of keys, one for encryption, one for decryption

● Ciphertext

- Scrambled message produced as output

● Decryption key

- Produces the original plaintext

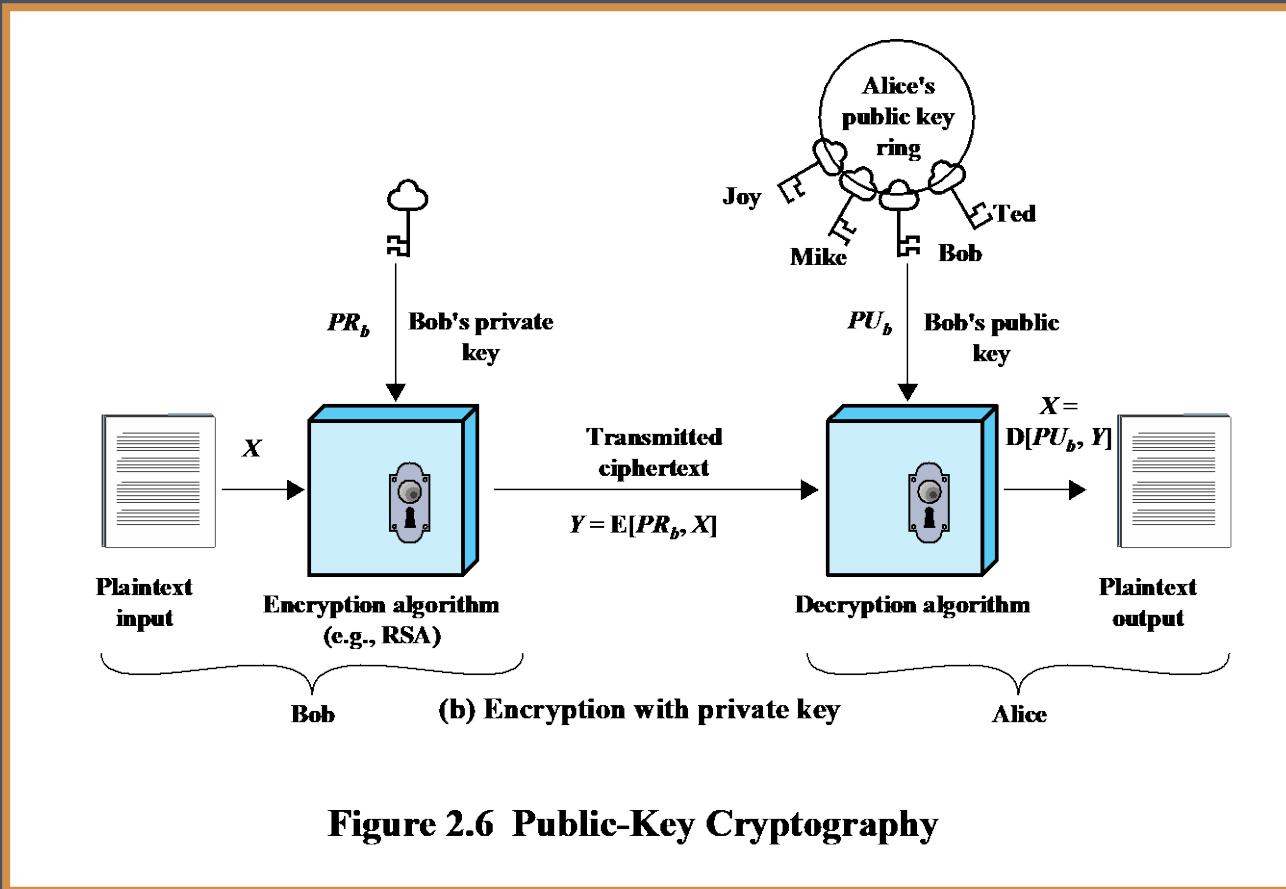


Figure 2.6 Public-Key Cryptography

- User encrypts data using his or her own private key
- Anyone who knows the corresponding public key will be able to decrypt the message

Public key (asymmetric) encryption



A



Private



Public

$(M)_{\text{PubB}}$ →

B



Private



Public

Public key (asymmetric) encryption

- Considerations
 - Performance
 - Slow
 - Typical 2048 to 4096 bit key
 - Can solve key distribution problem!
 - Complex architecture

Solving the key distribution problem with Public Key encryption



A



Private



Public



Secret

B



Secret



Private



Public

Table 2.3

Applications for Public-Key Cryptosystems

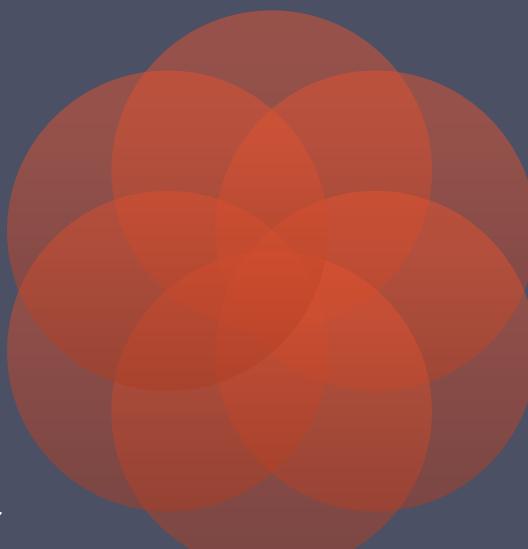
| Algorithm | Digital Signature | Symmetric Key Distribution | Encryption of Secret Keys |
|------------------|--------------------------|-----------------------------------|----------------------------------|
| RSA | Yes | Yes | Yes |
| Diffie-Hellman | No | Yes | No |
| DSS | Yes | No | No |
| Elliptic Curve | Yes | Yes | Yes |

Requirements for Public-Key Cryptosystems

Computationally
easy to create
key pairs

Useful if either key
can be used for
each role

Computationally
infeasible for
opponent to
otherwise recover
original message



Computationally
infeasible for
opponent to
determine private
key from public key

Computationally
easy for sender
knowing public
key to encrypt
messages

Computationally
easy for receiver
knowing private
key to decrypt
ciphertext

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)

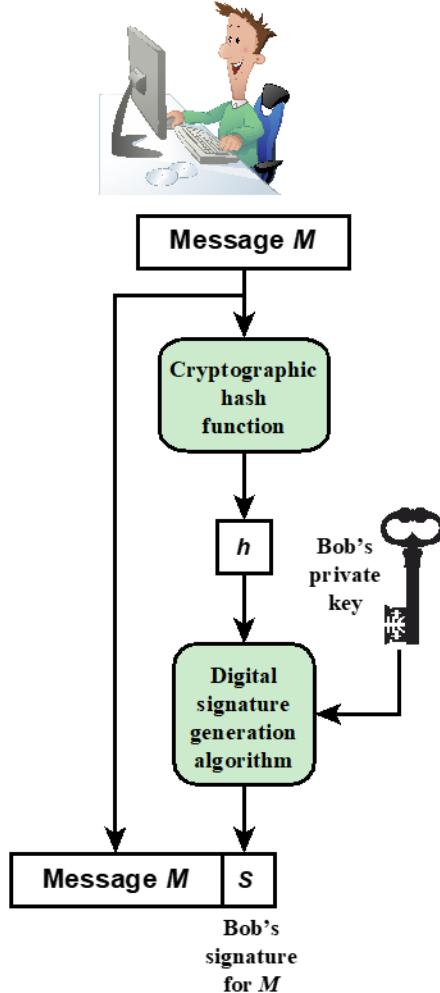
Security like RSA, but with much smaller keys

Digital Signatures

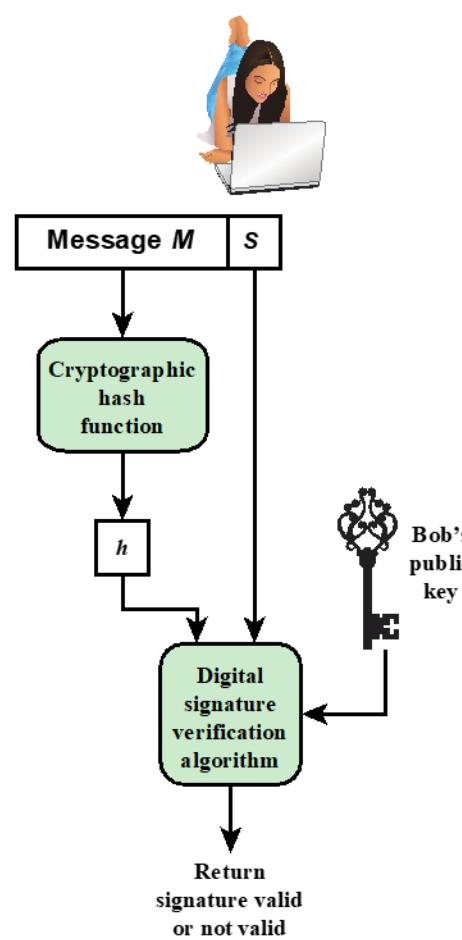
- NIST FIPS PUB 186-4 defines a digital signature as:

"The result of a cryptographic transformation of data that, when properly implemented, provides a mechanism for verifying origin authentication, data integrity and signatory non-repudiation."
- Thus, a digital signature is a data-dependent bit pattern, generated by an agent as a function of a file, message, or other form of data block
- FIPS 186-4 specifies the use of one of three digital signature algorithms:
 - Digital Signature Algorithm (DSA)
 - RSA Digital Signature Algorithm
 - Elliptic Curve Digital Signature Algorithm (ECDSA)

Bob



Alice

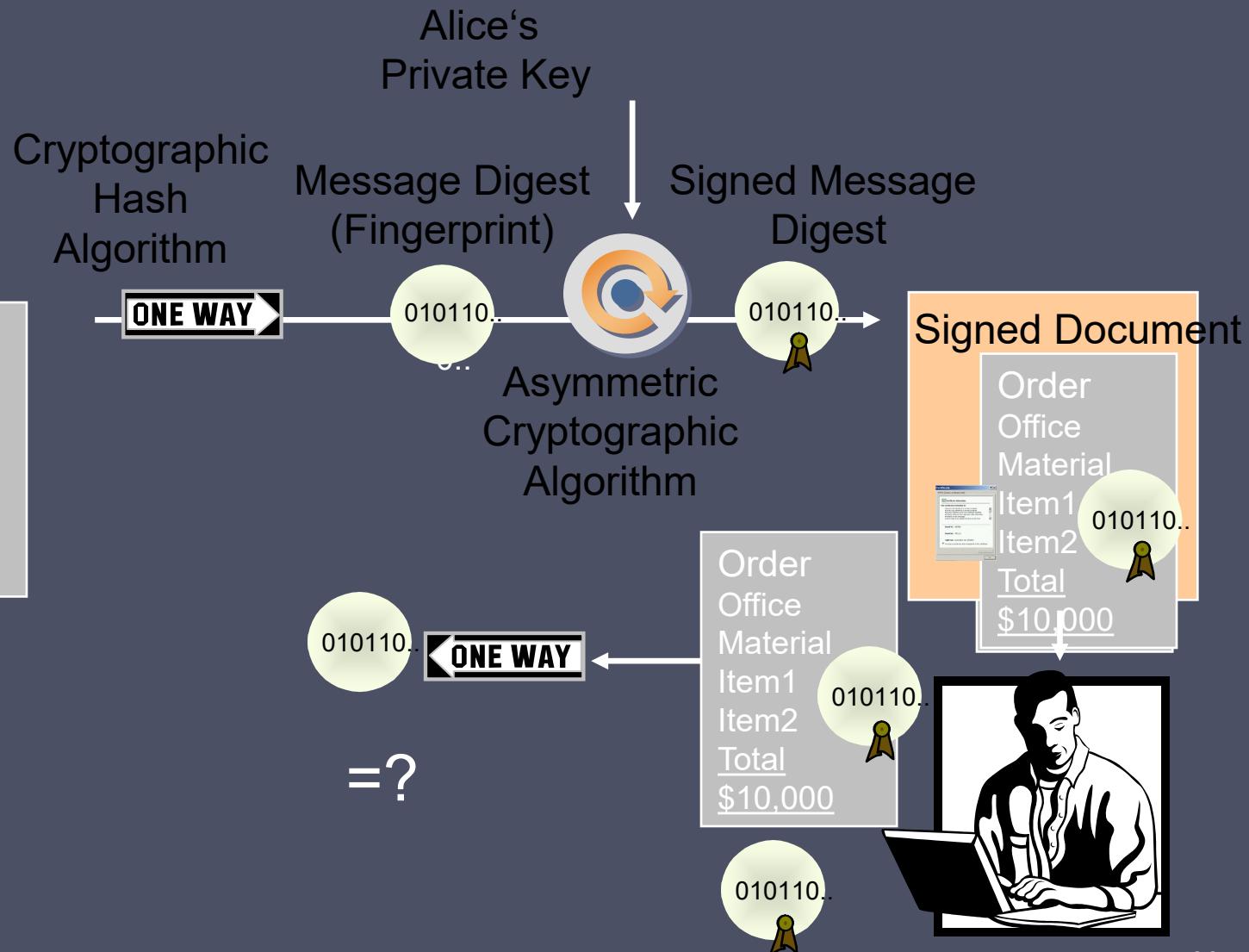


(a) Bob signs a message

(b) Alice verifies the signature

Figure 2.7 Simplified Depiction of Essential Elements of Digital Signature Process

With an animation...



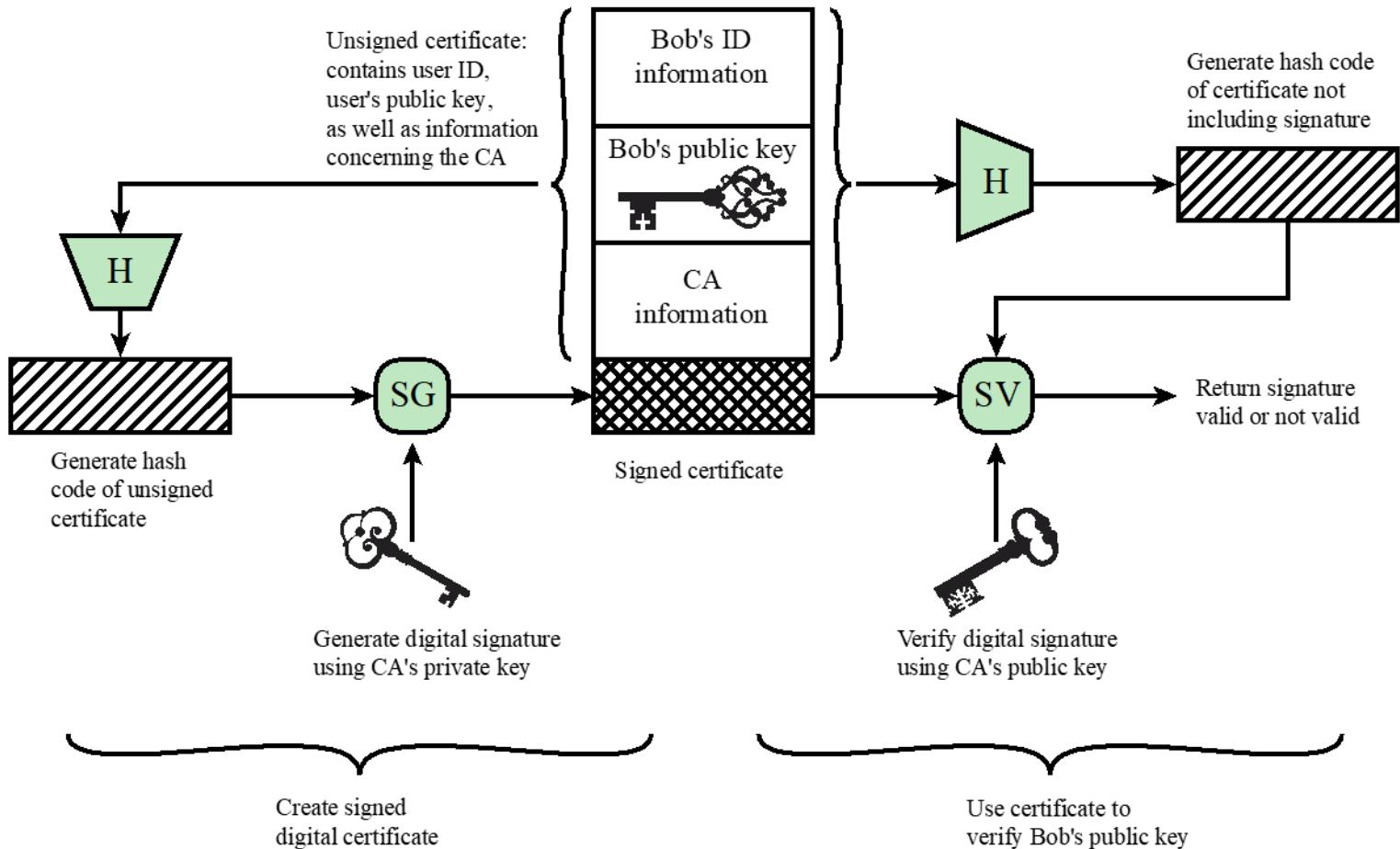
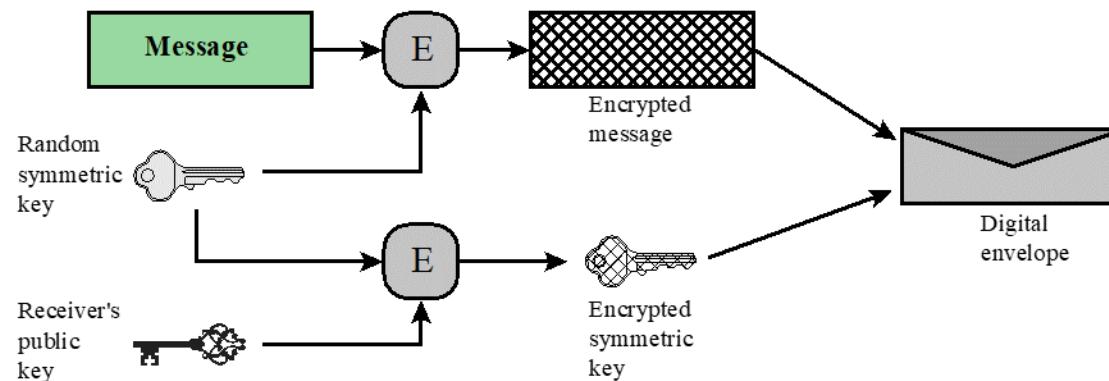
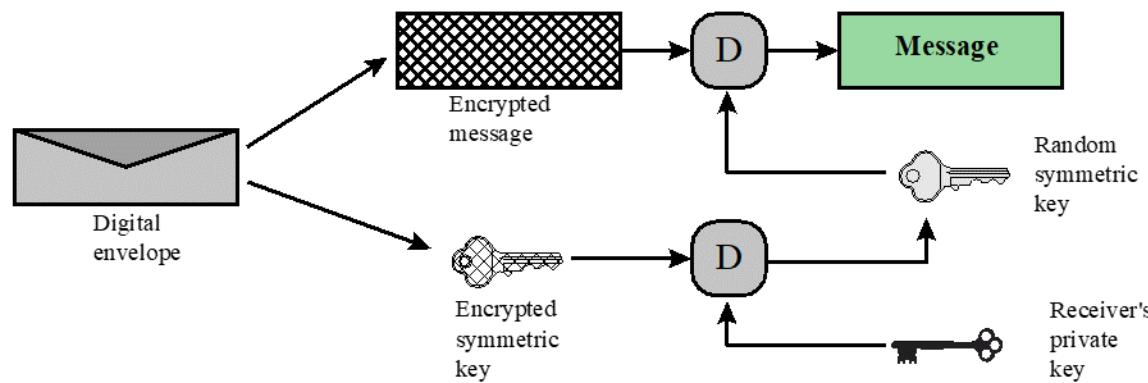


Figure 2.8 Public-Key Certificate Use



(a) Creation of a digital envelope



(b) Opening a digital envelope

Figure 2.9 Digital Envelopes

Random Numbers

**Uses include the
generation of:**

- Keys for public-key algorithms
- Stream key for symmetric stream cipher
- Symmetric key for use as a temporary session key or in creating a digital envelope
- Handshaking to prevent replay attacks
- Session key

Random Number Requirements

Randomness

- Criteria:

- Uniform distribution
 - Frequency of occurrence of each of the numbers should be approximately the same
- Independence
 - No one value in the sequence can be inferred from the others

Unpredictability

- Each number is statistically independent of other numbers in the sequence
- Opponent should not be able to predict future elements of the sequence on the basis of earlier elements

Random versus Pseudorandom

Cryptographic applications typically make use of algorithmic techniques for random number generation

- Algorithms are deterministic and therefore produce sequences of numbers that are not statistically random

Pseudorandom numbers are:

- Sequences produced that satisfy statistical randomness tests
- Likely to be predictable

True random number generator (TRNG):

- Uses a nondeterministic source to produce randomness
- Most operate by measuring unpredictable natural processes
 - e.g. radiation, gas discharge, leaky capacitors
- Increasingly provided on modern processors

Encryption of Stored Data

Common to encrypt transmitted data

Much less common for stored data

There is often little protection beyond domain authentication and operating system access controls

Data are archived for indefinite periods

Even though erased, until disk sectors are reused data are recoverable

Approaches to encrypt stored data:

Use a commercially available encryption package

Back-end appliance

Library based tape encryption

Background laptop/PC data encryption

Summary

- 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
- Random and pseudorandom numbers
 - The use of random numbers
 - Random versus pseudorandom
- Public-key encryption
 - Structure
 - Applications for public-key cryptosystems
 - Requirements for public-key cryptography
 - Asymmetric encryption algorithms
- Digital signatures and key management
 - Digital signature
 - Public-key certificates
 - Symmetric key exchange using public-key encryption
 - Digital envelopes
- Practical Application: Encryption of Stored Data

Practical 1

- Complete Problem 2.2 in your textbook.
- This problem uses a real-world example of a symmetric cipher, from an old U.S. Special Forces manual (public domain). The document, filename *Special Forces.pdf*, is available on the website.
 - a. Using the two keys (memory words) *cryptographic* and *network security*, encrypt the following message:

Be at the third pillar from the left outside the lyceum theatre tonight at seven. If you are distrustful bring two friends.

Make reasonable assumptions about how to treat redundant letters and excess letters in the memory words and how to treat spaces and punctuation. Indicate what your assumptions are. Note: The message is from the Sherlock Holmes novel *The Sign of Four*.

- b. Decrypt the ciphertext. Show your work.
- c. Comment on when it would be appropriate to use this technique and what its advantages are.

Practical 1

- Supporting files will be uploaded on the course website.
- Submit your work in PDF online on the course website, under the practical submission link.
- Be warned about plagiarism!
- Submission deadline: Monday 12 August 2024, 9:00am
- Absolutely no late submissions will be allowed. No excuse of slow connection, no (or ran out of) data, load shedding etc.
- No email submissions will be accepted, and certainly no late submissions after the deadline!⁴⁷