消息鉴别服务

Message Authentication

Authentication Requirements鉴别需求

- Disclosure 泄密
- Traffic analysis 通信流量分析
- Masquerade 假冒
- Content modification 内容修改
- Sequence modification 顺序修改
- Timing modification 计时修改
- Source repudiation 信息源否认
- Destination repudiation 目的端否认

Message Authentication Mechanism

- message authentication is a service:
  - protecting the integrity of a message (完整性)
  - validating identity of originator （验明正身）
  - non-repudiation of origin (抗抵赖)

- three alternative functions used:
  - message encryption —消息加密
  - message authentication code (MAC)—鉴别码
  - hash function—散列函数

Message Encryption 消息加密

message encryption itself also provides a measure of authentication

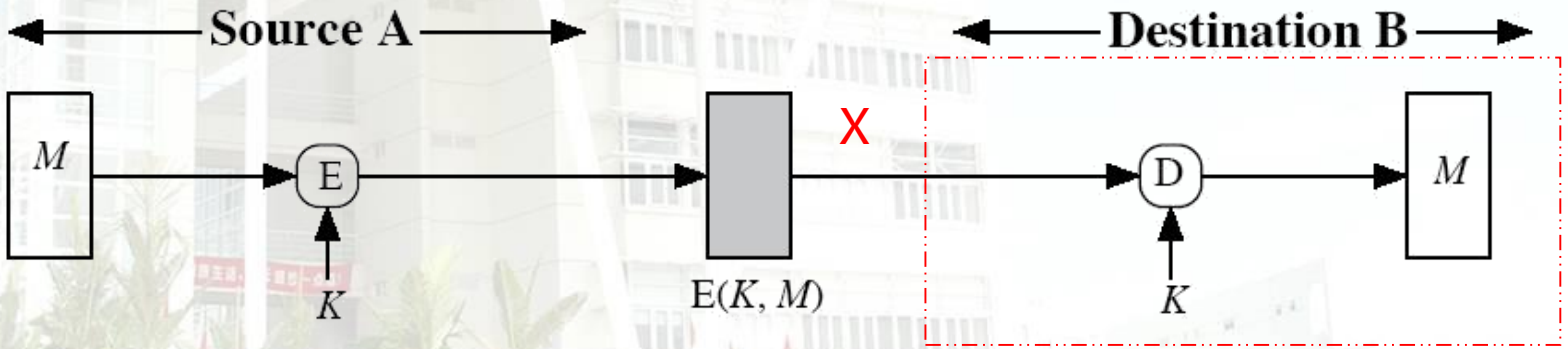
Message Encryption消息加密

symmetric encryption

if symmetric encryption is used then:

- receiver know sender has created it
- only sender and receiver share the key
- content can not be altered

B accepts any input X



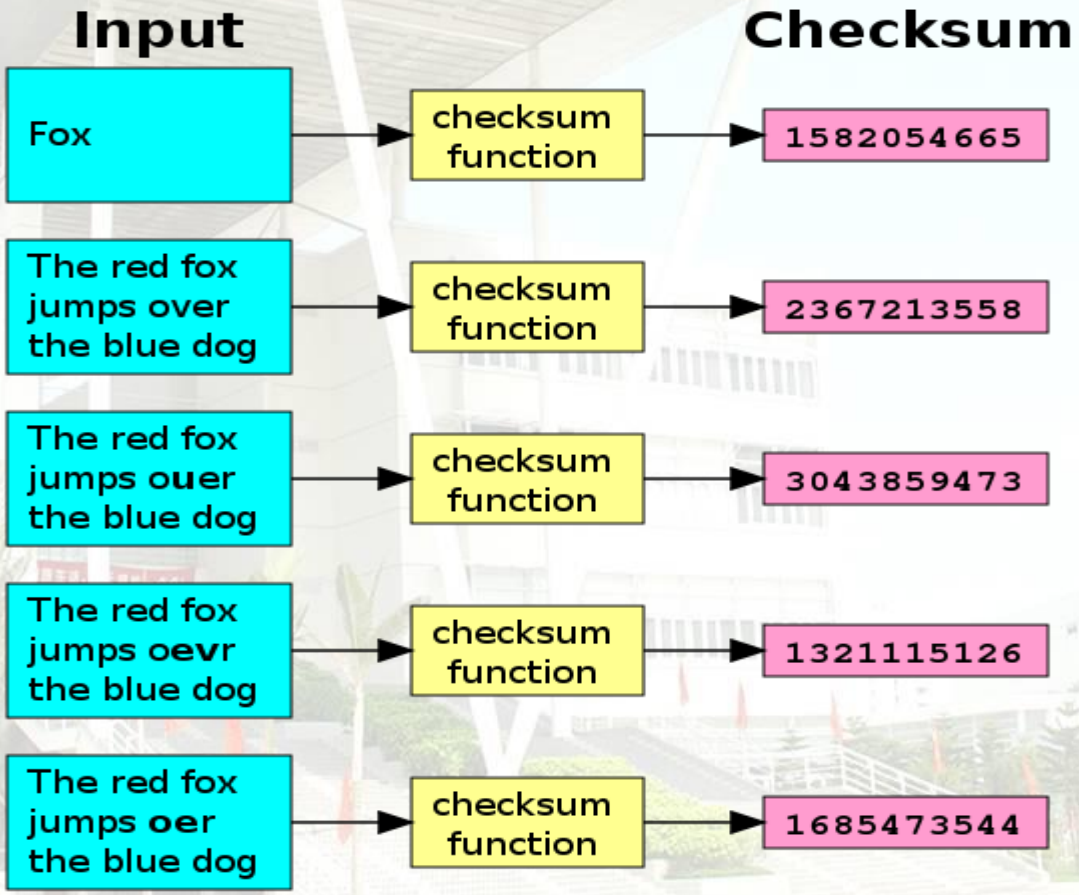
(a) Symmetric encryption: confidentiality and authentication

M 是合法的明文?

由A发出吗?

Checksum 检验和

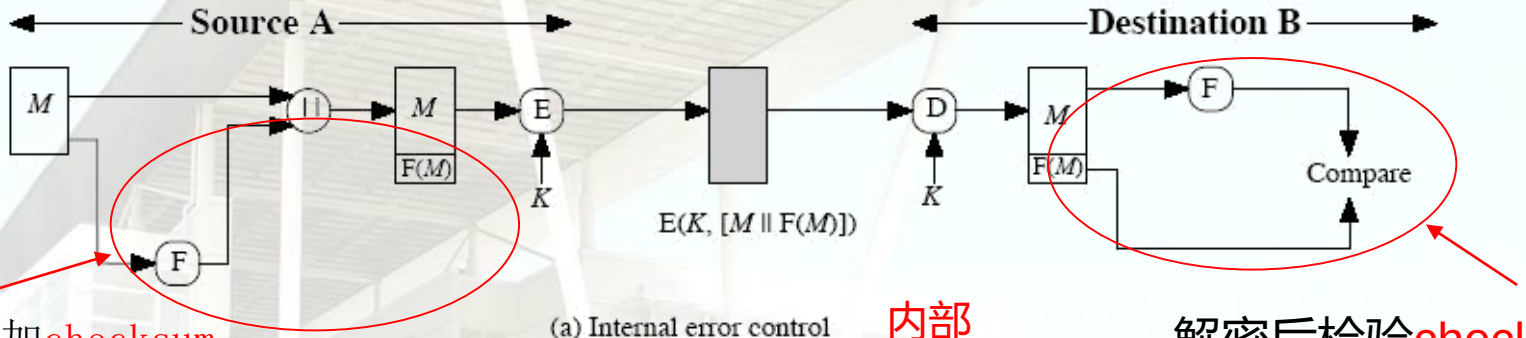
- 保证数据的完整性



Message Encryption消息加密

symmetric encryption

Add a checksum to detect any changes:

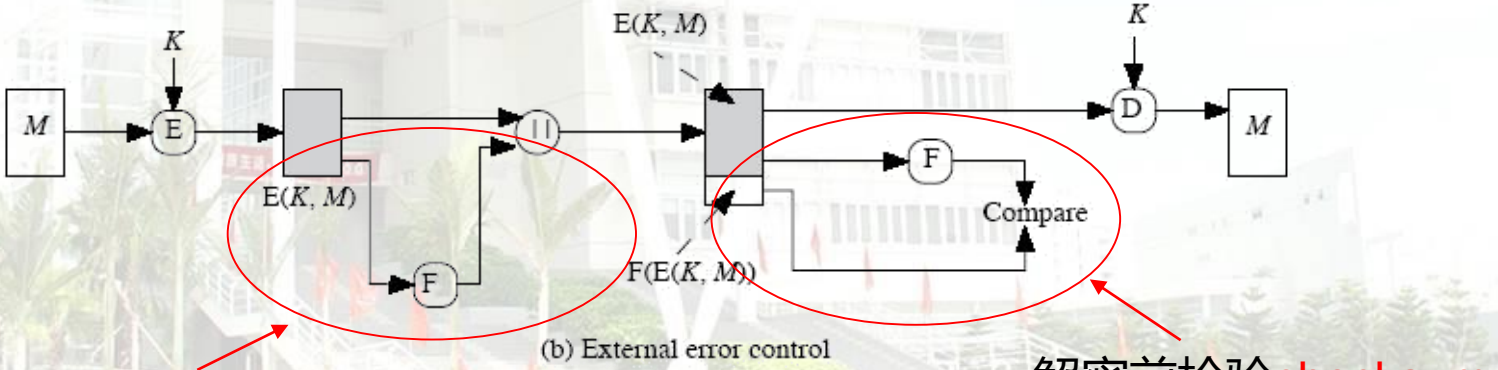


加密前加checksum

or FCS－frame check sequence 帧校验序列

(a) Internal error control 内部

解密后检验checksum



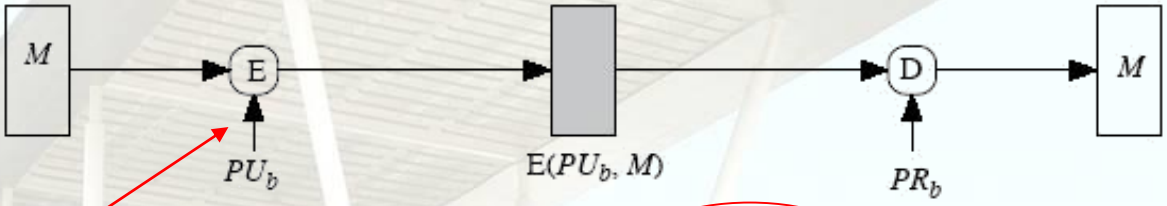
加密后加checksum

(b) External error control 外部

解密前检验checksum

Figure 11.2 Internal and External Error Control

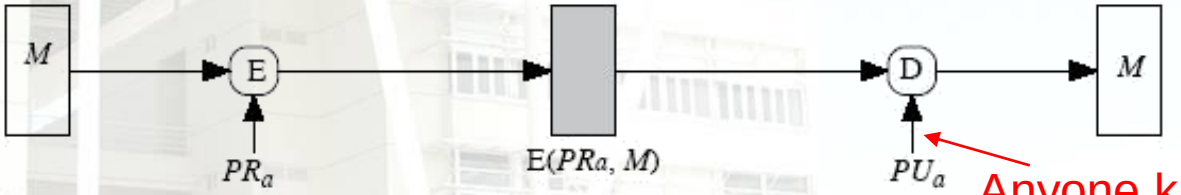
Message Encryption消息加密
public-key encryption



(b) Public-key encryption: confidentiality

Anyone knows public key

No authentication

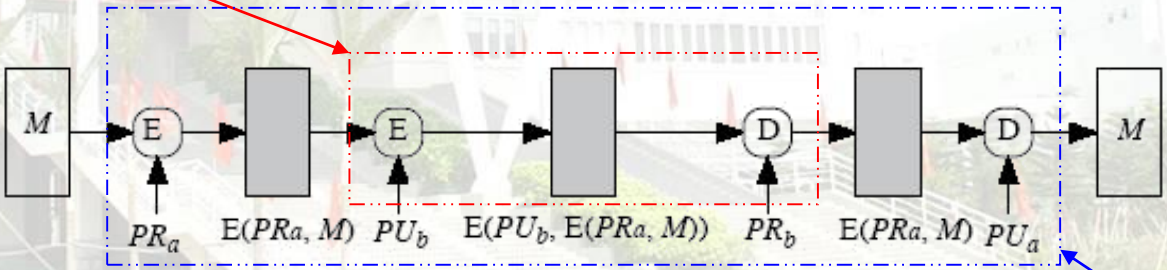


(c) Public-key encryption: authentication and signature

Anyone knows public key

No confidentiality

confidentiality

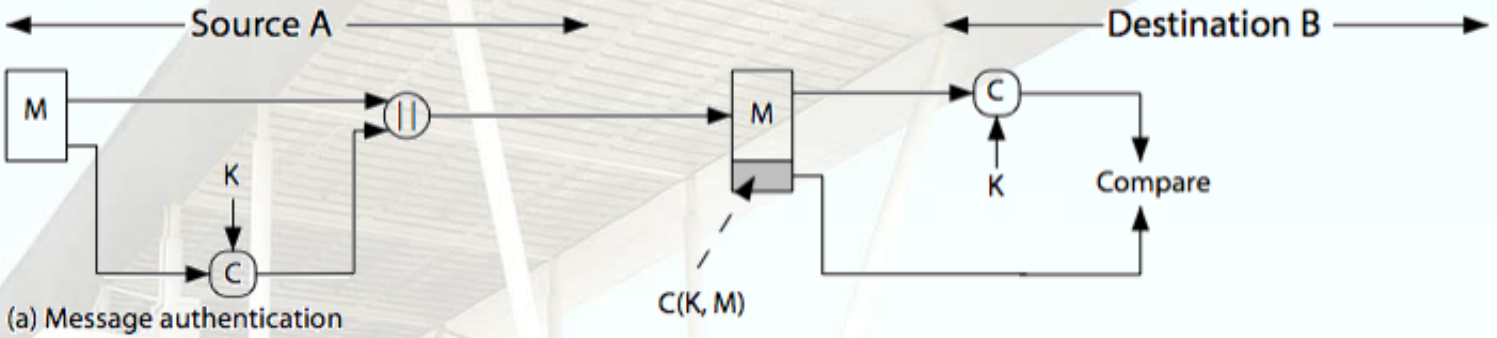


(d) Public-key encryption: confidentiality, authentication, and signature

Four times public-key operation

authentication

消息鉴别一般机制



发送者

接收者

消息

密钥

算法

MAC

消息

附件

消息

算法

密钥

MAC

实际收到附件

如果二者一样，则认为消息是完整的

Message Authentication Code (MAC)

- generated by an algorithm that creates a small fixed-sized block （MAC）
  - depending on both message and key
  - like encryption though need not be reversible

    不要求可逆
- appended to message as a **signature**
- receiver performs computation on message and checks it matches the MAC
- provides assurance that message is unaltered and comes from sender

✓ 如果消息中包含顺序码（如TCP），则接收者可以保证消息的正常顺序

Use of Message Authentication Code

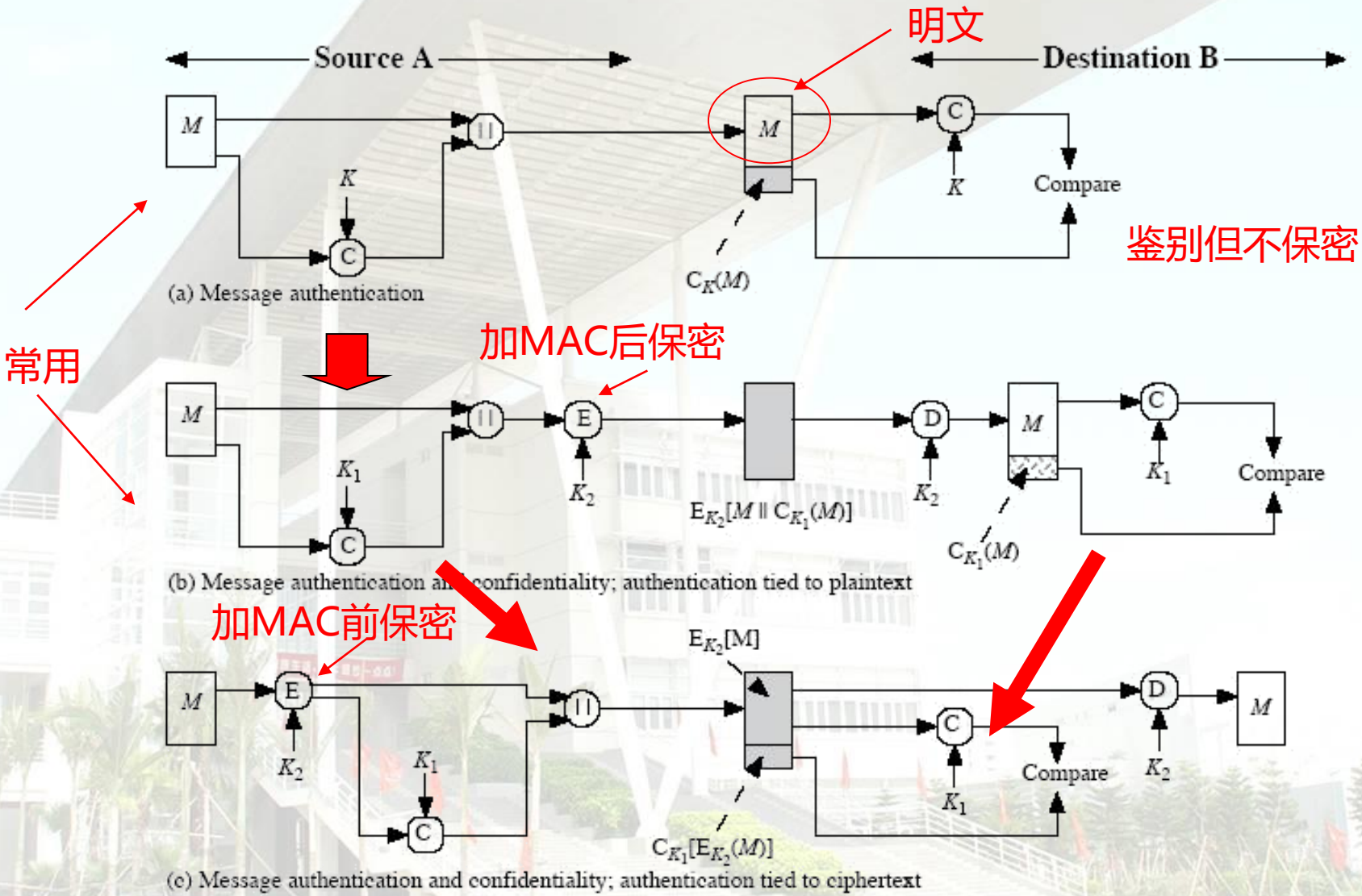


Figure 11.4 Basic Uses of Message Authentication Code (MAC)

Table 11.2 Basic Uses of Message Authentication Code C

$A \rightarrow B: M \| C(K, M)$

- Provides authentication
  - Only A and B share K

(a) Message authentication

$A \rightarrow B: E(K_2, [M \| C(K, M)])$

- Provides authentication
  - Only A and B share K_1
- Provides confidentiality
  - Only A and B share K_2

(b) Message authentication and confidentiality: authentication tied to plaintext

对明文鉴别

$A \rightarrow B: E(K_2, M) \| C(K_1, E(K_2, M))$

- Provides authentication
  - Using K_1
- Provides confidentiality
  - Using K_2

(c) Message authentication and confidentiality: authentication tied to ciphertext

对密文鉴别

MAC and Symmetric Encryption

- Message is broadcast in plaintext with a MAC.

  Only one verifies the MAC-cheaper and more reliable

- For heavy load user, randomly check MAC.

  -selective basis

- verify MAC Only when required

  -Save processor resource

- May not be concern to keep message secret

  -it is important to authenticate messages

Eg. SNMPV3 separates the functions of privacy and authentication

- separates the functions of privacy and authentication

  architectural flexibility

MAC Properties

- MAC is a cryptographic checksum

  $MAC = C_K(M)$

  – condenses a variable-length message M to a fixed-sized authenticator using a secret key K

- many-to-one function 多对一函数

  – potentially many messages have same MAC

Requirements for MACs

taking into account the types of attacks

- ✓ need the MAC to satisfy the following:
  1. knowing a message and MAC, is infeasible to find another message with same MAC
  2. MACs should be uniformly distributed
  3. MAC should depend on all bits of the message

消息摘要Message Digest

- ✓ 把任意长的输入消息串变化成固定长的输出串的一种函数

HASH 函数（散列算法，哈希）

- **功能**
  - 输入为任意长度的消息M; 输出为一个固定长度的值，
  - 函数是单向的
  - 消息M的所有位提供错误检测能力
    消息任何一位或多位的变化将导致该散列值的变化
  - 又称“数字指纹” （Digital finger print)
    数据鉴别码（Data authentication code)
- **作用**

鉴别/数字签名:

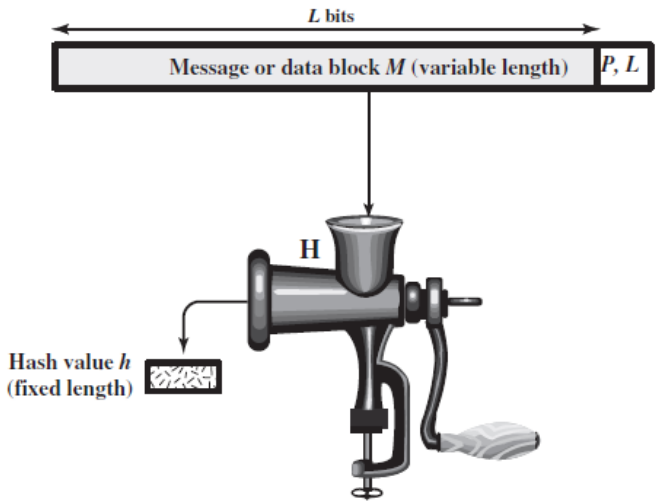
提高数字签名有效性、破坏某些数据结构和

分离保密与签名、鉴别、数据完整性检测和加密

HASH 函数与“碰撞”

- 输入为任意长度的消息;
  输出为一个固定长度值，

- 构造两个不同的消息，将它们映射为同个消息的摘要计算上不可行的



P, L = padding plus length field

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

- ✓ 如果内容不同的明文，通过散列算法得出的结果（信息摘要）相同，就称为发生了“碰撞”
- ✓ 散列算法的用途不是对明文加密，让别人看不懂，而是通过对信息摘要的比对，防止对原文的篡改

  HASH碰撞率要低

碰撞

理论上说，当这种摘要算法被完全攻破时，也就是说可以从摘要恢复出任意原文

注意：是任意原文，因为所有的摘要算法的特点就是存在着一个无穷大的碰撞原文的集合。而真正的原文只是其中一份。

对应这个无穷大的集合来说，这就是可能性无穷小

Requirements for Hash Functions

- can be applied to any sized message M
- produces fixed-length output h
- easy to compute h=H(M) for any message M
- given h is infeasible to find x s.t H(x)=h
  - one-way property **(单向性)**
- given x is infeasible to find y s.t. H(y)=H(x)
  - weak collision resistance （碰撞性1）
- is infeasible to find any x,y s.t. H(y)=H(x)
  - strong collision resistance （碰撞性2）

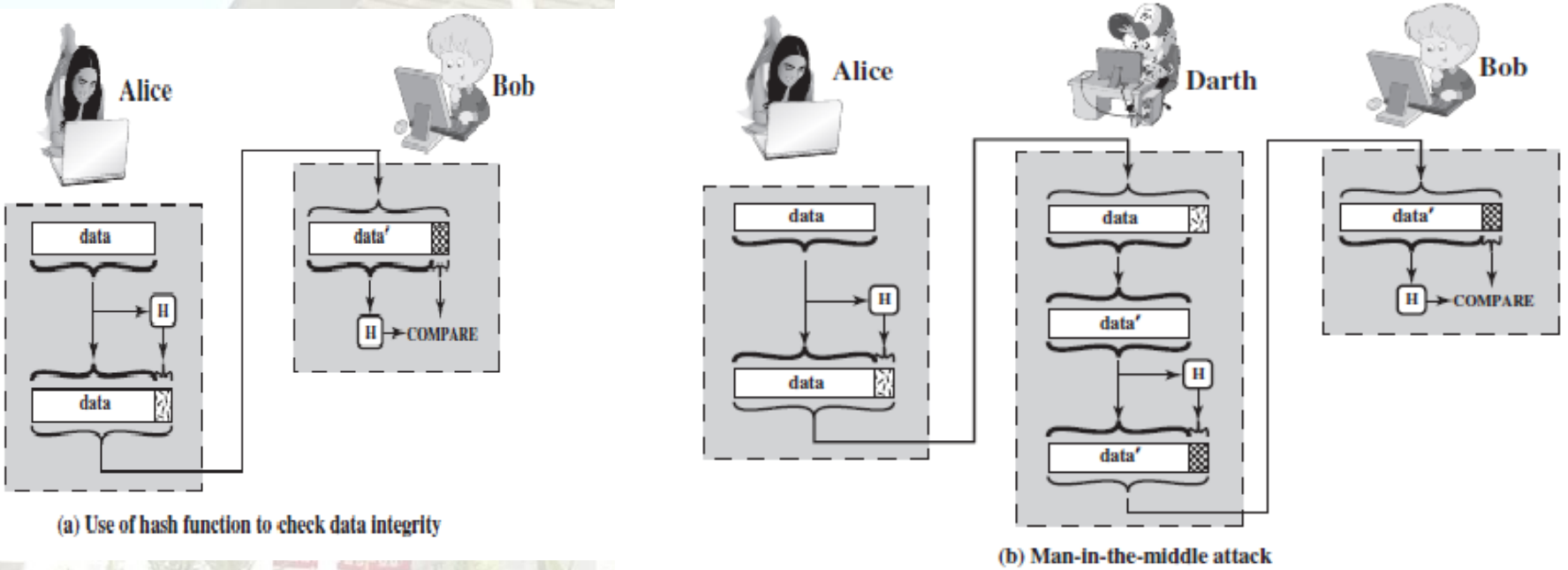
Hash Functions

- condenses arbitrary message to fixed size

  `h = H(M)`

- public and not keyed
  - Note: MAC is keyed
- detect changes to message
- can use in various ways with message
- most often to create a digital signature

数据完整性检测和攻击



(a) Use of hash function to check data integrity

(b) Man-in-the-middle attack

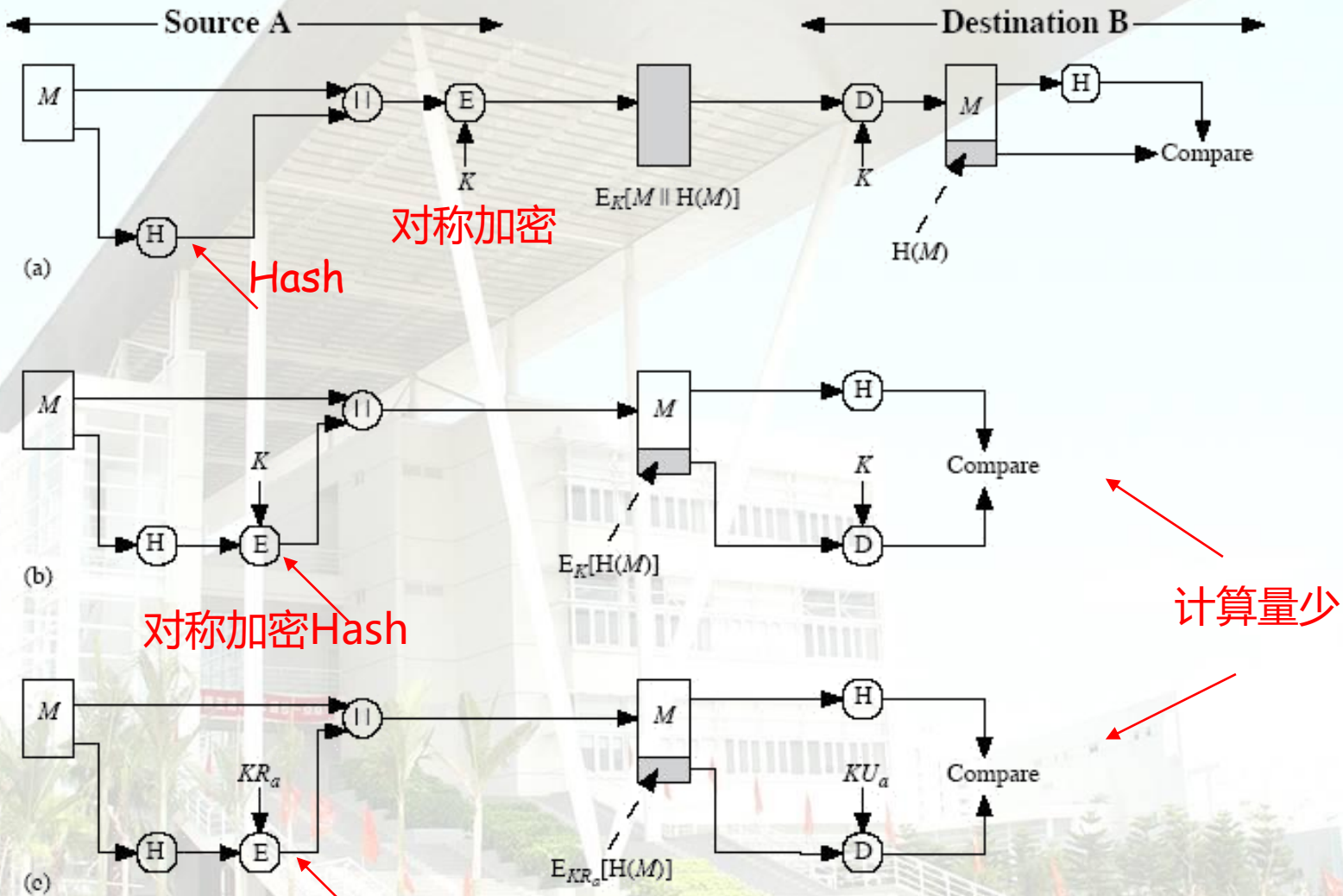
Figure 11.2 Attack Against Hash Function

To prevent this attack, the hash value generated by Alice must be protected.

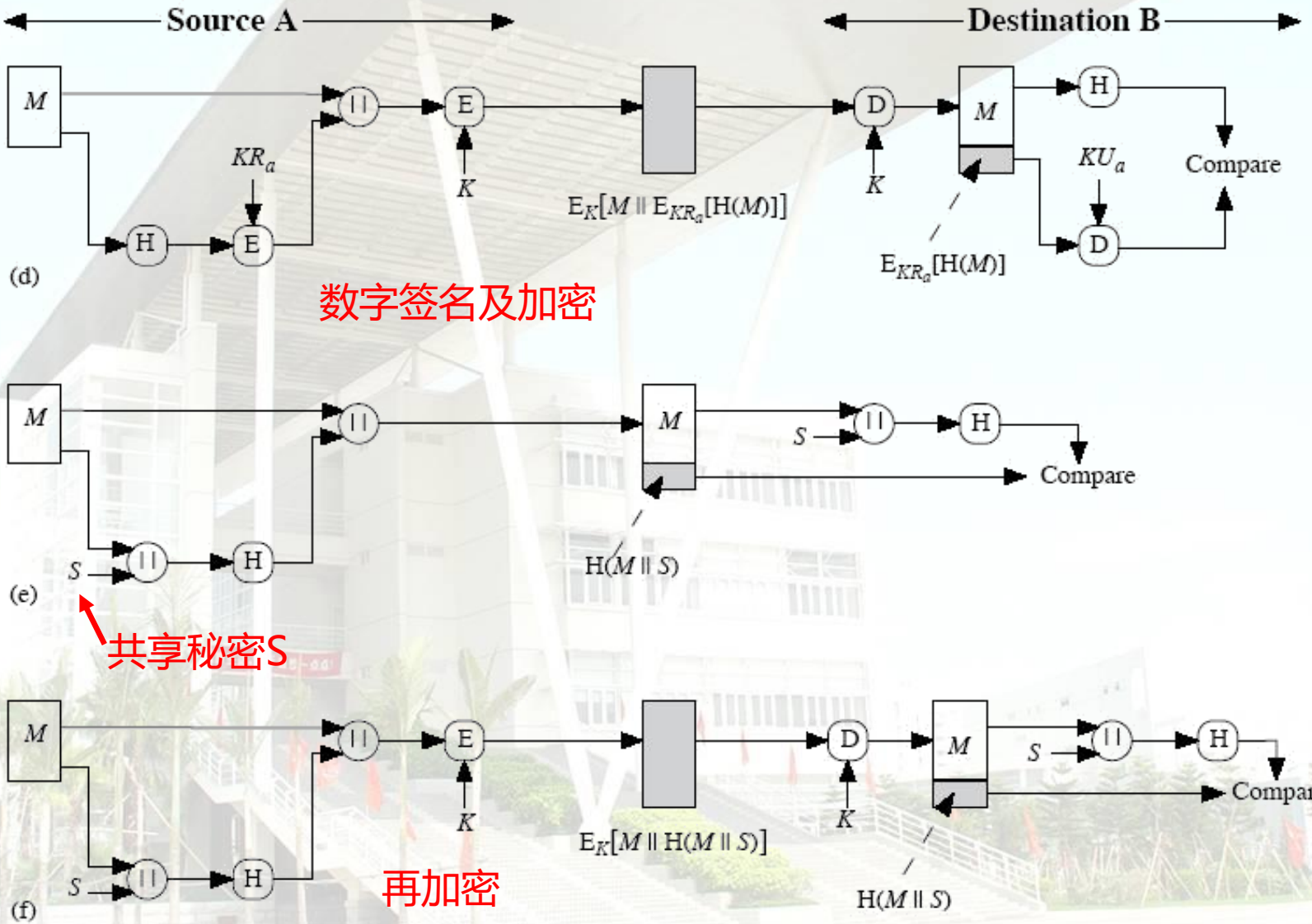
Table 11.3 Basic Uses of Hash Function H

$A \rightarrow B: E(K, [M \| H(M)])$ •Provides confidentiality —Only A and B share K •Provides authentication —H(M) is cryptographically protected	$A \rightarrow B: E(K, [M \| E(PR_a, H(M))])$ •Provides authentication and digital signature •Provides confidentiality —Only A and B share K
(a) Encrypt message plus hash code	(d) Encrypt result of (c) - shared secret key
$A \rightarrow B: M \| E(K, H(M))$ •Provides authentication —H(M) is cryptographically protected	$A \rightarrow B: M \| H(M \| S)$ •Provides authentication —Only A and B share S
(b) Encrypt hash code - shared secret key	(e) Compute hash code of message plus secret value
$A \rightarrow B: M \| E(PR_a, H(M))$ •Provides authentication and digital signature —H(M) is cryptographically protected —Only A could create $E(PR_a, H(M))$	$A \rightarrow B: E(K, [M \| H(M \| S])$ •Provides authentication —Only A and B share S •Provides confidentiality —Only A and B share K
(c) Encrypt hash code - sender's private *key*	(f) Encrypt result of (e)

Basic uses of Hash Functions （1）



Basic uses of Hash Functions （2）



Hash函数分类

需要密钥?

1）一般Hash函数不需要密钥（unkeyed）
MDC (manipulation detection codes) or
MIC (message integrity codes)

2）包含密钥的Hash函数（keyed），用来提供数据源认证和数据完整性校验时 - MAC码

Hash vs MAC Algorithms

- Hash Functions
  - condense arbitrary size message to fixed size
  - by processing message in blocks
  - through some compression function
  - either custom or block cipher based
- Message Authentication Code (MAC)
  - fixed sized authenticator for some message
  - to provide authentication for message
  - by using block cipher mode or hash function

Simple Hash Functions

分为m个分组，每组n位， 完成异或运算

	位1	位2	……	位n
分组1	b_{11}	b_{21}	……	b_{n1}
分组2	b_{12}	b_{22}	……	b_{n2}
	……	…	……	
分组m	b_{1m}	b_{2m}	……	b_{nm}
Hash码	C_1	C_2	……	C_n

$$C_i = b_{i1} \oplus b_{i2} \oplus \cdots \oplus b_{im}$$

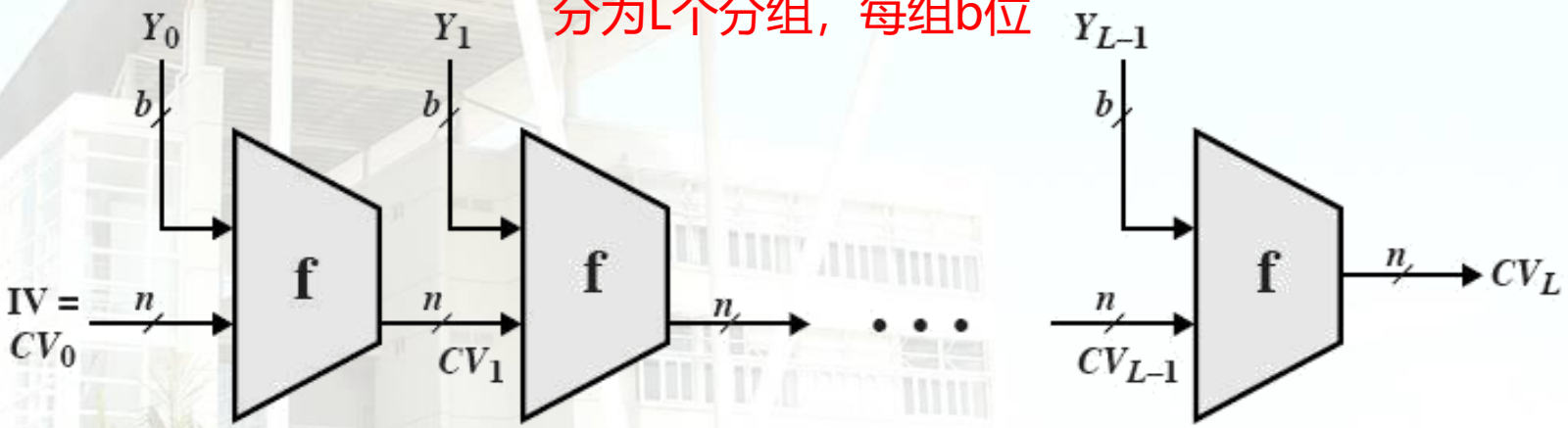
where

C_i = ith bit of the hash code, $1 \le i \le$ n
m = number of n-bit blocks in the input
b_{ij} = ith bit in jth block
$\oplus$ = XOR operation

Hash Algorithm Structure

General Structure of Secure Hash Code

分为L个分组，每组b位



IV = Initial value
CV_i = chaining variable
Y_i = *i*th input block
f = compression algorithm

L = number of input blocks
n = length of hash code
b = length of input block

f11-

Hash小结

- ✓ Hash函数把变长信息映射到定长信息
- ✓ Hash函数不具备可逆性
- ✓ Hash函数速度较快
- ✓ 对Hash函数的密码分析比对称密钥密码更困难
- ✓ Hash函数可用于消息摘要
- ✓ Hash函数可用于数字签名