Chapter3

Public-Key Cryptography and Message Authentication

Contents

Authentication Requirements

- In the context of communications across a network, the following attacks can be identified:
  - Disclosure
  - Traffic analysis
  - *Masquerade*
  - *Content modification*
  - *Sequence modification*
  - *Timing modification*
  - Source repudiation
  - Destination repudiation

Authentication

- Requirements - must be able to verify that:
  1. Message came from apparent source or author,
  2. Contents have not been altered,
  3. Sometimes, it was sent at a certain time or sequence.
- Protection against active attack (falsification of data and transactions)

Approaches to Message Authentication

- Authentication Using Conventional Encryption
  - Only the sender and receiver should share a key
- Message Authentication without Message Encryption
  - An authentication tag is generated and appended to each message
  - Approaches
    - Message Authentication Code
    - One-way Hash Function

Message Authentication Code

- Calculate the MAC as a function of the message and the key.
  - **$MAC_M = F(K_{AB}, M)$**
- MAC algorithms
  - DES: The last number of bits of ciphertext are used as the code. A 16- or 32-bit code is typical.
- MAC V.S. Encryption
  - The authentication algorithm need not be reversible.

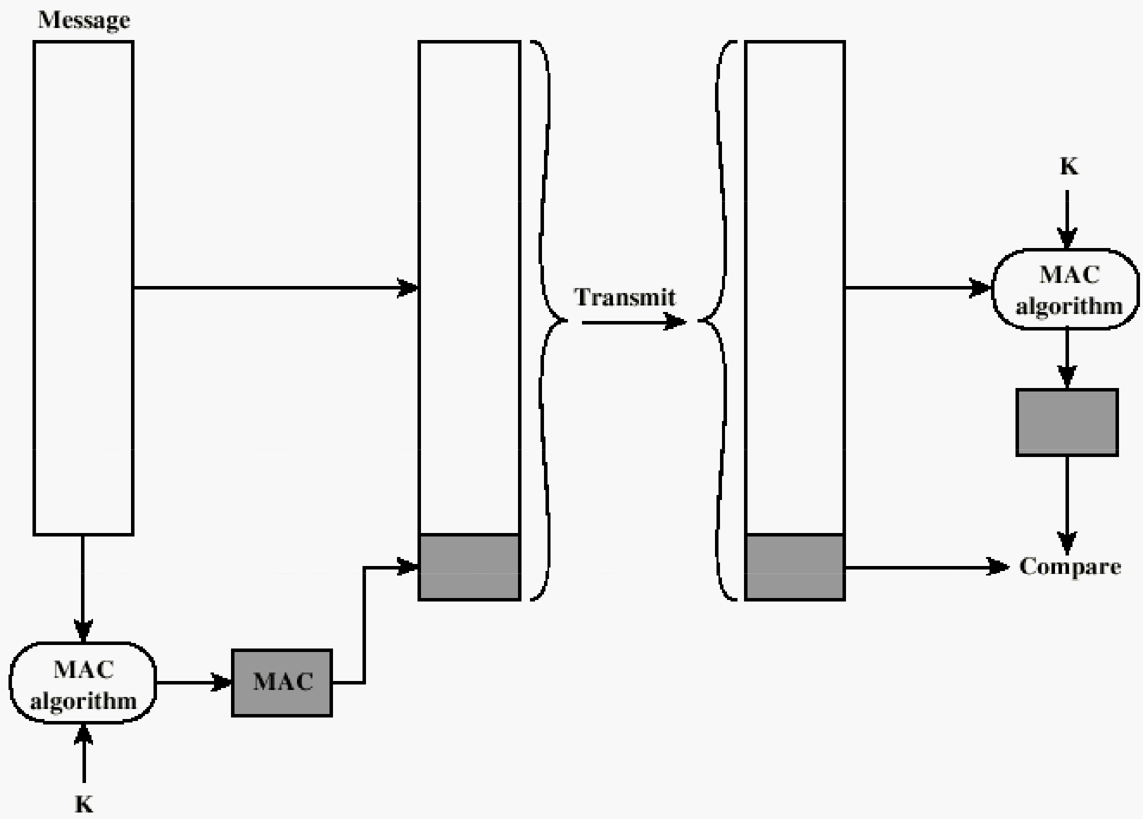
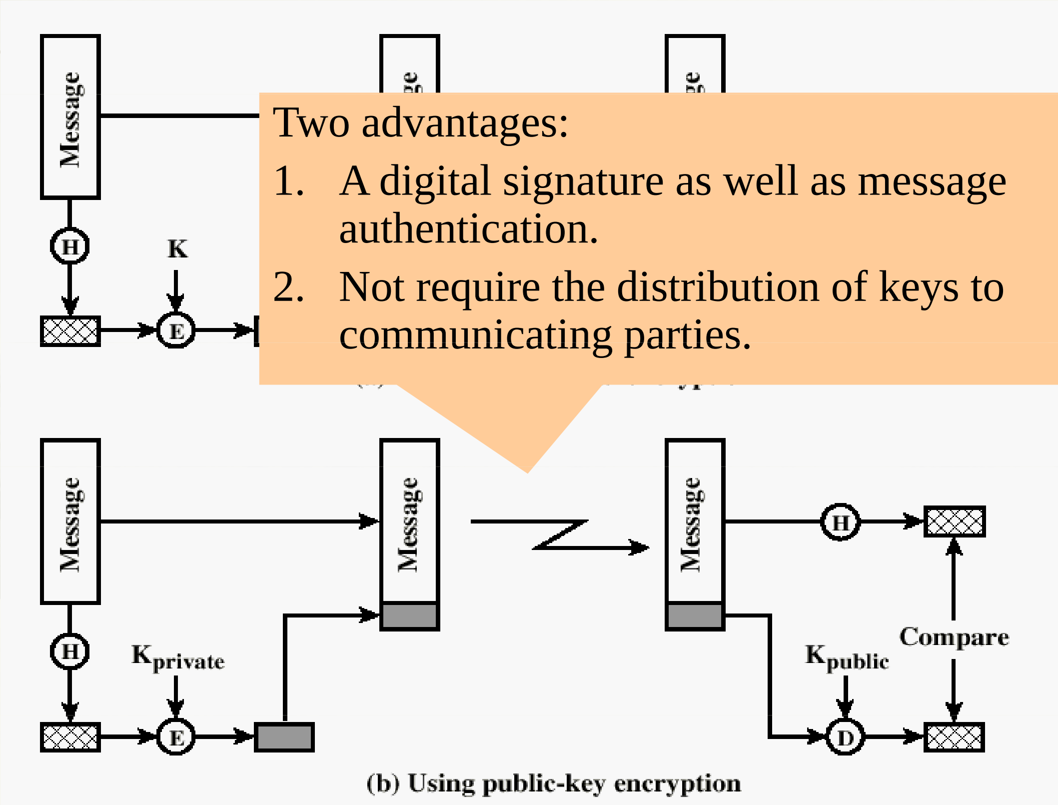


Figure 3.1 Message Authentication Using a Message Authentication Code (MAC)

One-Way Hash Function

- Hash function V.S. MAC
  - Common features
    - Accept a variable-size message M as input,
    - Produce a fixed-size message digest H(M) as output.
  - Differences
    - A hash function does not also take a secret key as input.
- Message authentication using a one-way hash function.
  - Using conventional encryption
  - Using public-key encryption
  - Using secret value

One-way HASH function

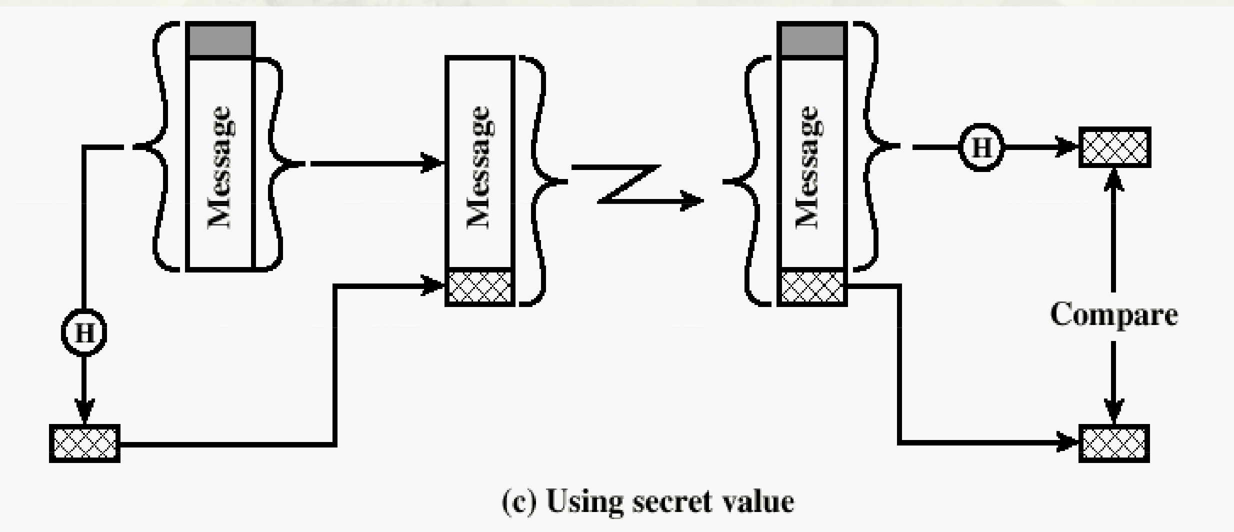


Two advantages:

1. A digital signature as well as message authentication.
2. Not require the distribution of keys to communicating parties.

One-way HASH function

- Secret value is added before the hash and removed before transmission.
  - $MD_M=H(S_{AB}||M)$



Secure HASH Functions

- Purpose of the HASH function is to produce a ”**fingerprint**.

Requirement	Description
Variable input size	H can be applied to a block of data of any size.
Fixed output size	H produces a fixed-length output.
Efficiency	H(*x*) is relatively easy to compute for any given *x*, making both hardware and software implementations practical.
Preimage resistant (one-way property)	For any given hash value *h*, it is computationally infeasible to find *y* such that H(*y*) = *h*.
Second preimage resistant (weak collision resistant)	For any given block *x*, it is computationally infeasible to find *y* ! *x* with H(*y*) = H(*x*).
Collision resistant (strong collision resistant)	It is computationally infeasible to find any pair (*x*, *y*) such that H(*x*) = H(*y*).
Pseudorandomness	Output of H meets standard tests for pseudorandomness

Simple Hash Functions

	bit 1	bit 2	• • •	bit *n*
block 1	b_{11}	b_{21}		b_{n1}
block 2	b_{12}	b_{22}		b_{n2}
	⋮	⋮	⋮	⋮
block *m*	b_{1m}	b_{2m}		b_{nm}
hash code	C_1	C_2		C_n

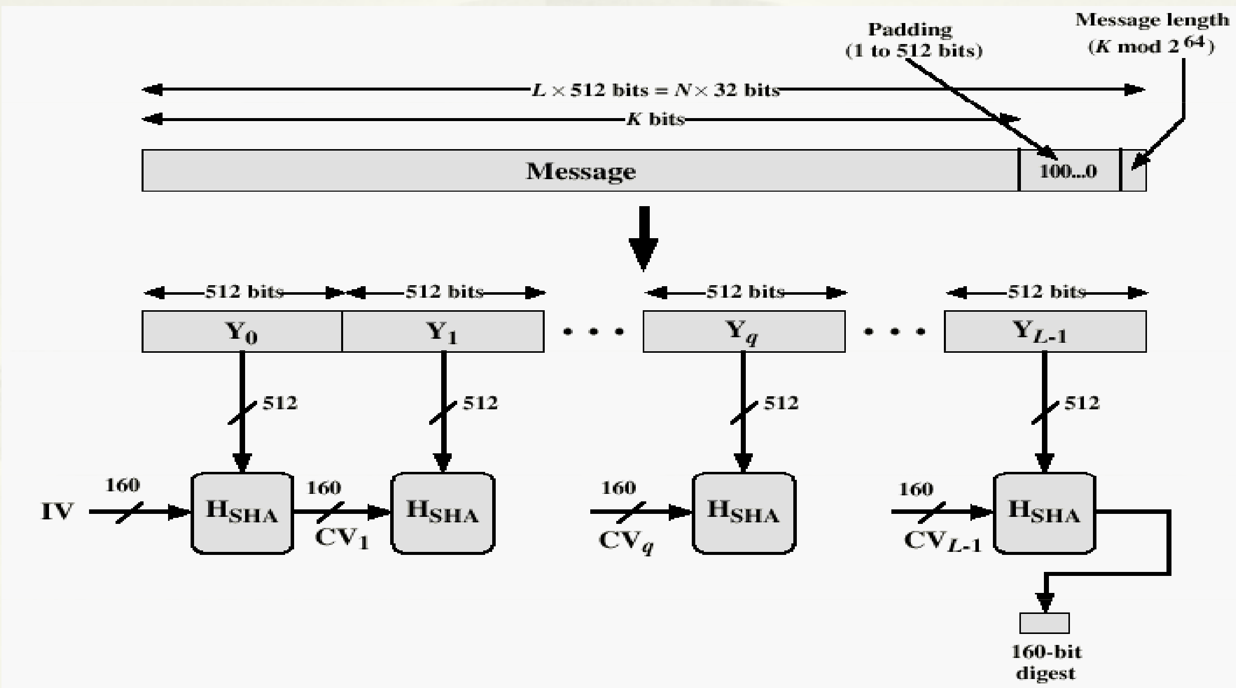
Figure 3.3 Simple Hash Function Using Bitwise XOR

- $C_i = b_{i1} \oplus b_{i2} \oplus \ldots \oplus b_{im}$
- One-bit circular shift on the hash value after each block is processed would improve

The SHA Secure Hash Function

- Secure Hash Algorithm (SHA) was developed by the NIST and published as FIPS PUB 180 in 1993; a revised version (SHA-1) was issued as FIPS PUB 180-1 in 1995.
  - Message: Maximum length of a message < 2^{64} bits
  - INPUT: 512-bit blocks
  - OUTPUT: a 160-bit message digest
- SHA-1
  - Step 1: Append padding bits.
  - Step 2: Append length.
  - Step 3: Initialize MD buffer
  - Step 4: Process message in 512-bit (16-word) blocks
  - Step 5: Output.

Message Digest Generation Using SHA-1



SHA-1 Processing of single 512-Bit Block

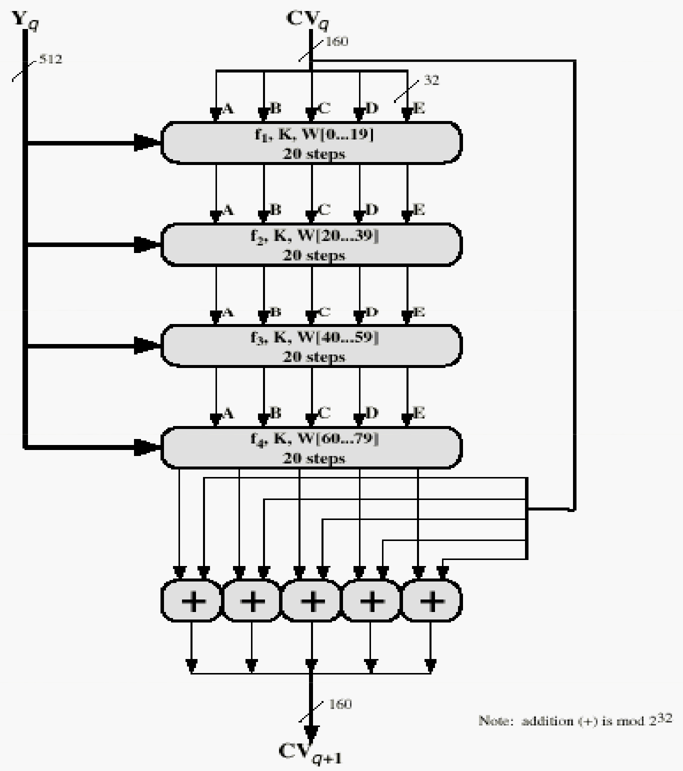


Figure 3.5 SHA-1 Processing of a Single 512-bit Block

Other Secure HASH functions

	SHA-1	MD5	RIPEMD-160
Digest length	160 bits	128 bits	160 bits
Basic unit of processing	512 bits	512 bits	512 bits
Number of steps	80 (4 rounds of 20)	64 (4 rounds of 16)	160 (5 paired rounds of 16)
Maximum message size	2^{64}-1 bits	∞	∞

Revised Secure Hash Standard

- Recent 2005 results on security of SHA-1 have raised concerns on its use in future applications
- NIST issued revision FIPS 180-2 in 2002
- adds 3 additional versions of SHA
  - SHA-256, SHA-384, SHA-512
- designed for compatibility with increased security provided by the AES cipher
- structure & detail is similar to SHA-1
- hence analysis should be similar
- but security levels are rather higher

SHA Versions

	SHA-1	SHA-224	SHA-256	SHA-384	SHA-512
Message digest size	160	224	256	384	512
Message size	$< 2^{64}$	$< 2^{64}$	$< 2^{64}$	$< 2^{128}$	$< 2^{128}$
Block size	512	512	512	1024	1024
Word size	32	32	32	64	64
Number of steps	80	64	64	80	80

SHA-512 Overview

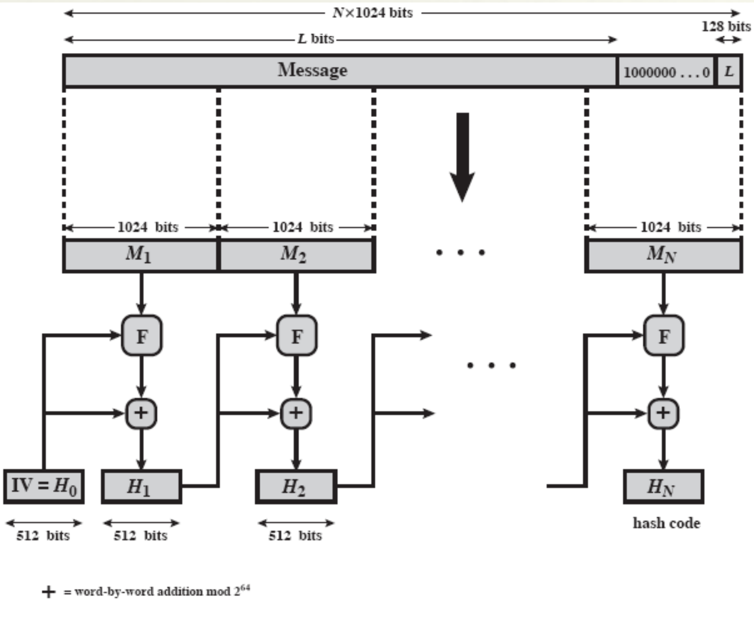


Figure 3.4 Message Digest Generation Using SHA-512

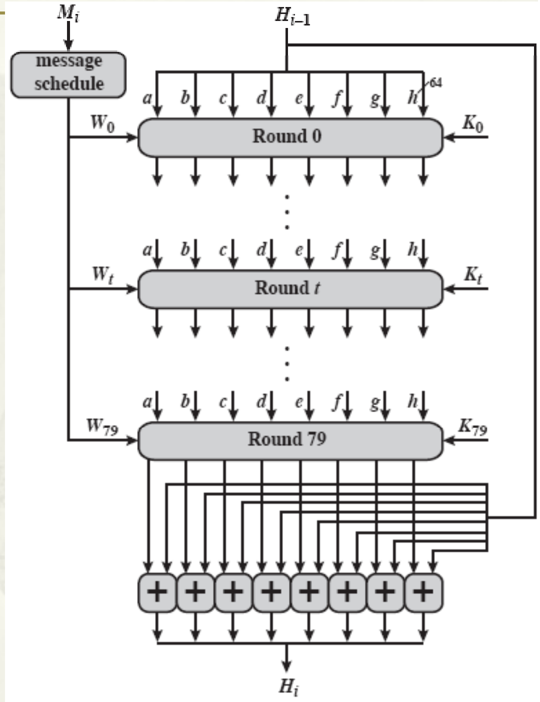


Figure 3.5 SHA-512 Processing of a Single 1024-Bit Block

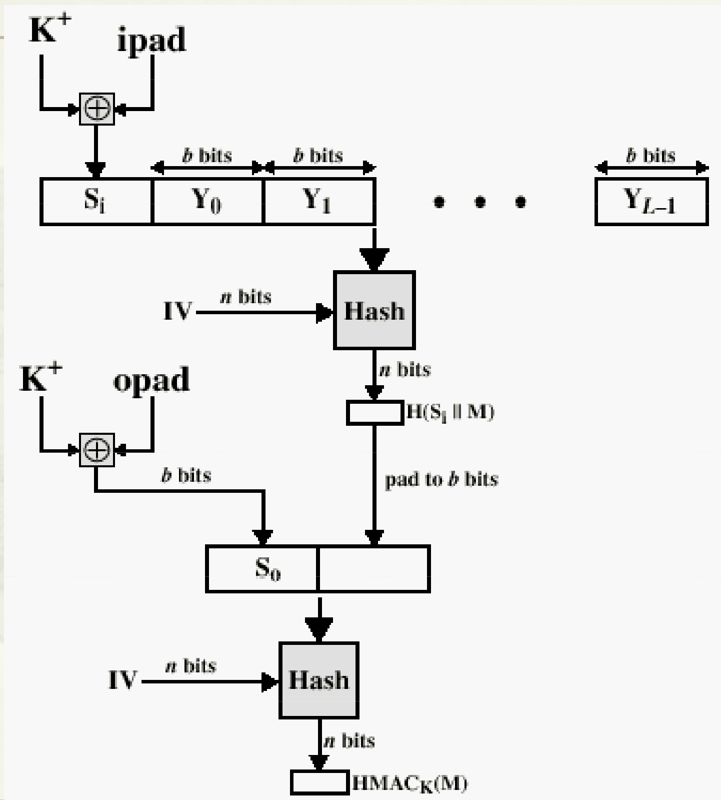
SHA-512 Compression Function

- heart of the algorithm
- processing message in 1024-bit blocks
- consists of 80 rounds
  - updating a 512-bit buffer
  - using a 64-bit value Wt derived from the current message block
  - and a round constant based on cube root of first 80 prime numbers

HMAC

- Use a MAC derived from a cryptographic hash code, such as SHA-1.
- Motivations
  - Cryptographic hash functions executes faster in software than encryptoin algorithms such as DES
  - Library code for cryptographic hash functions is widely available
  - No export restrictions from the US
- A hash function such as SHA-1 was not designed for use as a MAC and cannot be used directly for that purpose because it does not rely on a secret key.

HMAC Structure



MACs Based on Block Ciphers

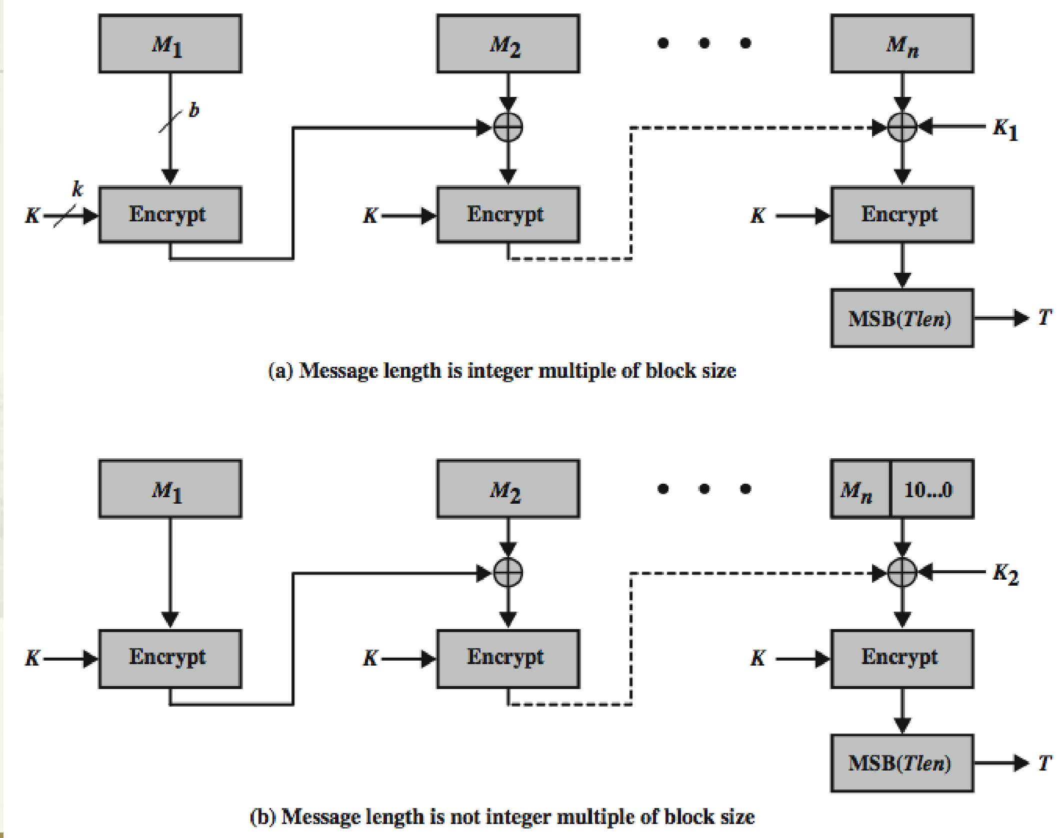
- Cipher-based Message Authentication Code (CMAC)
- Counter with Cipher Block Chaining-Message Authentication Code (CCM)

CMAC

- The CMAC mode of operation is for use with AES and 3-DES.
- It is specified in NIST Special Publication 800-38B.

$C_1 = E(K, M_1)$
$C_2 = E(K, [M_2 \oplus C_1])$
$C_3 = E(K, [M_3 \oplus C_2])$
$\vdots$
$C_n = E(K, [M_n \oplus C_{n-1} \oplus K_1])$
$T = MSB_{Tlen}(C_n)$

CMAC Overview



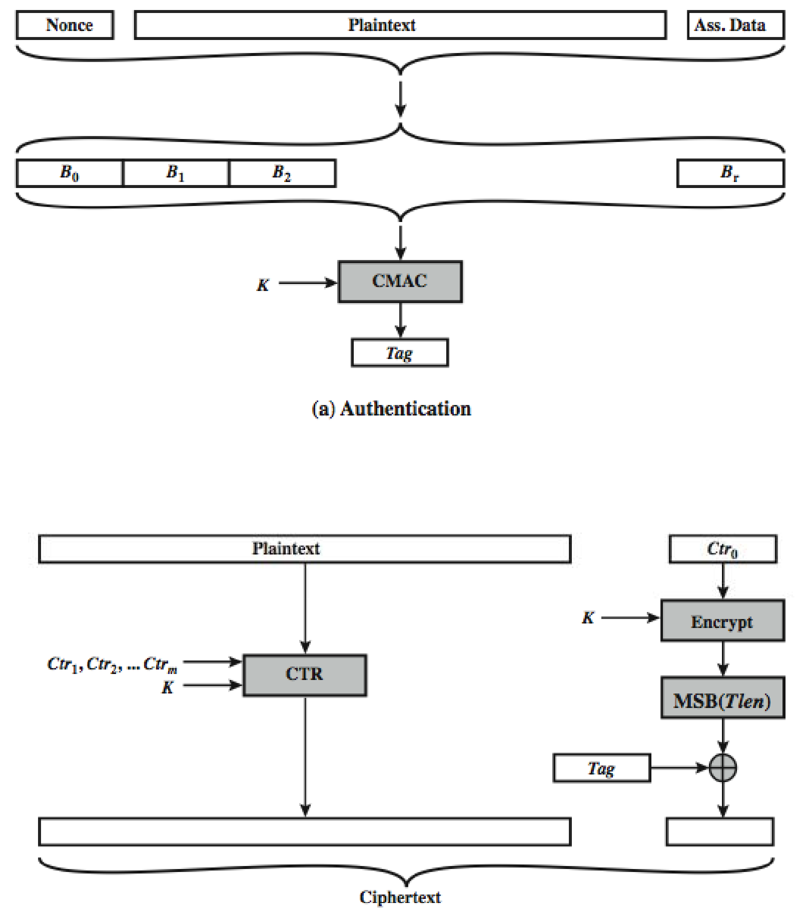
(a) Message length is integer multiple of block size

(b) Message length is not integer multiple of block size

Counter with Cipher Block Chaining-Message Authentication Code (CCM)

- The CCM mode of operation is referred to as **an authenticated encryption mode.**
- Algorithmic ingredients
  - AES encryption algorithm
  - CTR mode of operation
  - CMAC authentication algorithm
- A single key is used for both encryption & MAC algorithms.

CCM Operation

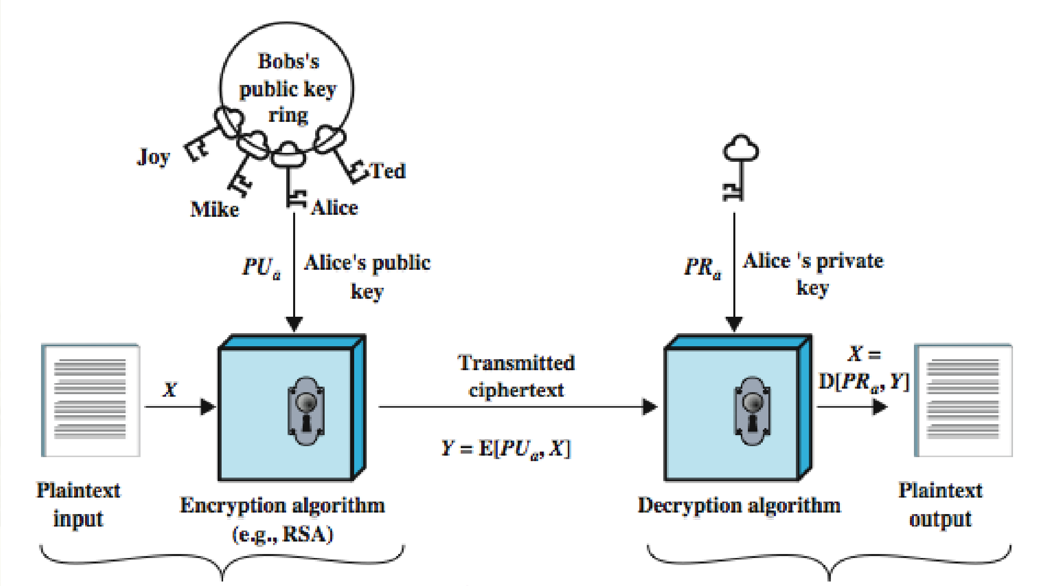


(a) Authentication
(b) Encryption

Public-Key Cryptography Principles

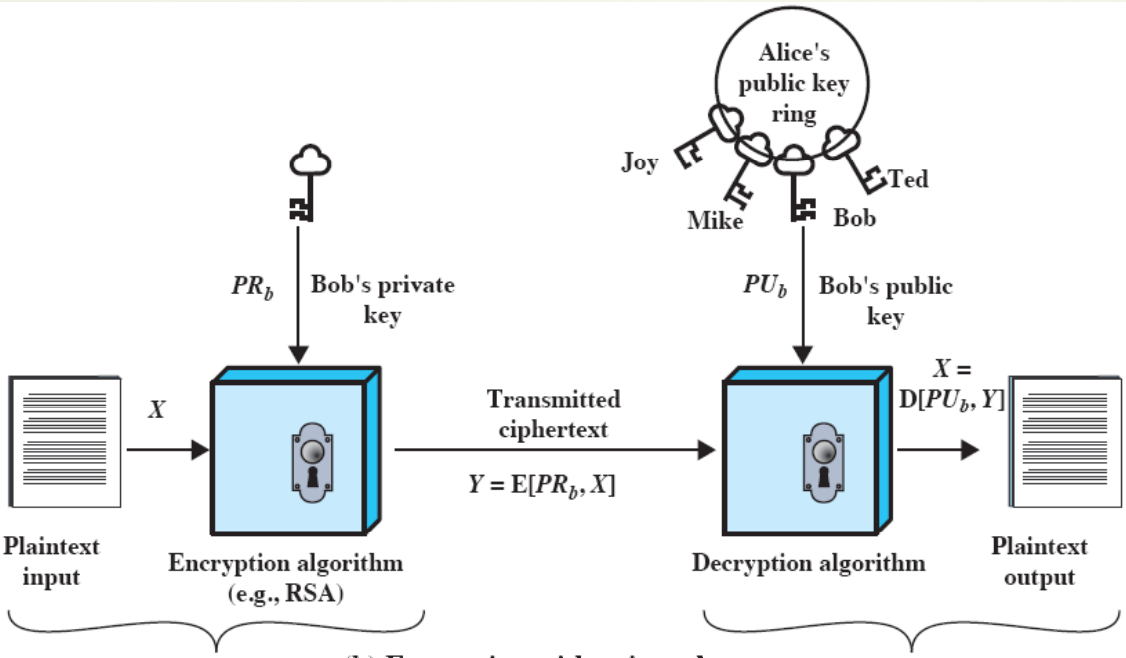
- The use of two keys has consequences in:
  - Key distribution, confidentiality and authentication.
- The scheme has six ingredients (see Figure 3.9)
  - Plaintext
  - Encryption algorithm
  - Public and private key
  - Ciphertext
  - Decryption algorithm

Encryption using Public-Key system



(a) Encryption with public key

Authentication using Public-Key System



(b) Encryption with private key

Public-Key Cryptosystem: Secrecy and Authentication

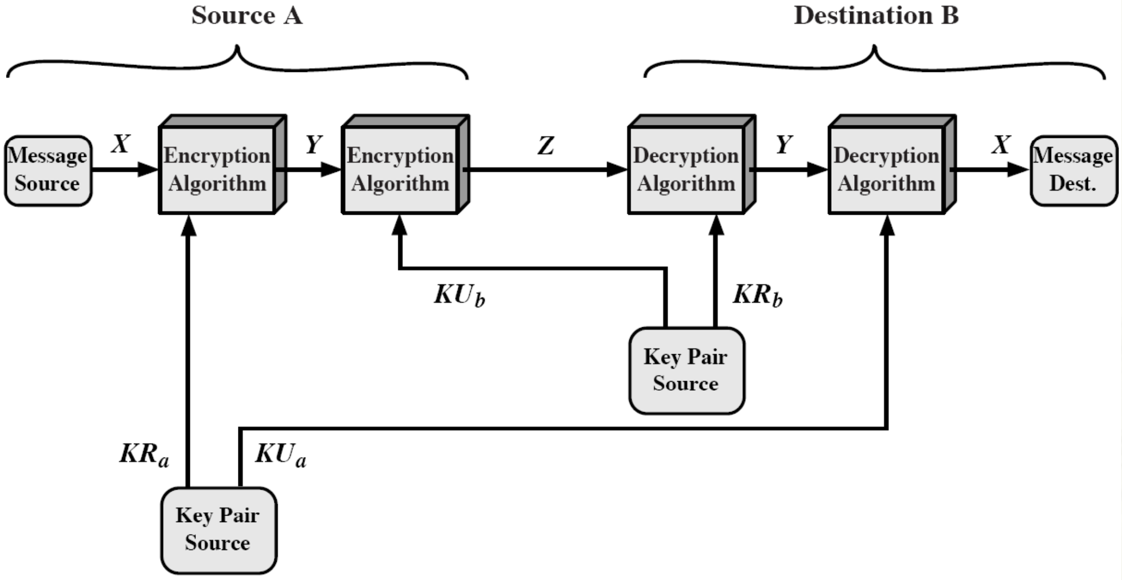


Figure 9.4 Public-Key Cryptosystem: Secrecy and Authentication

Applications for Public-Key Cryptosystems

- Three categories:
  - **Encryption/decryption:** The sender encrypts a message with the recipient's public key.
  - **Digital signature:** The sender "signs" a message with its private key.
  - **Key echange:** Two sides cooperate two exhange a session key.

Requirements for Public-Key Cryptography

1. Computationally easy for a party B to generate a pair (public key KU_b, private key KR_b)
2. Easy for sender to generate ciphertext:

$$C = E_{KUb}(M)$$

3. Easy for the receiver to decrypt ciphertect using private key:

$$M = D_{KRb}(C) = D_{KRb}[E_{KUb}(M)]$$

Requirements for Public-Key Cryptography

4. Computationally infeasible to determine private key (KR_b) knowing public key (KU_b)
5. Computationally infeasible to recover message M, knowing KU_b and ciphertext C
6. Either of the two keys can be used for encryption, with the other used for decryption:

$$M = D_{KRb}[E_{KUb}(M)] = D_{KUb}[E_{KRb}(M)]$$

Public-Key Cryptographic Algorithms

- RSA and Diffie-Hellman
- **RSA** - Ron Rives, Adi Shamir and Len Adleman at MIT, in 1977.
  - RSA is a block cipher
    - Plaintext and ciphertext are integers between *0* and *n-1* for some *n*.
  - The most widely implemented
- **Diffie-Hellman**
  - Exchange a secret key securely
  - Compute discrete logarithms

The RSA Algorithm – Key Generation

1.	Select *p,q*	*p* and *q* both prime
2.	Calculate *n* = *p* x *q*	
3.	Calculate $\Phi(n) = (p-1)(q-1)$	
4.	Select integer e	$\gcd(\Phi(n), e) = 1; 1 < e < \Phi(n)$
5.	Calculate *d*	$d = e^{-1} \bmod \Phi(n)$
6.	Public Key	KU = {e,n}
7.	Private key	KR = {d,n}

Example of RSA Algorithm

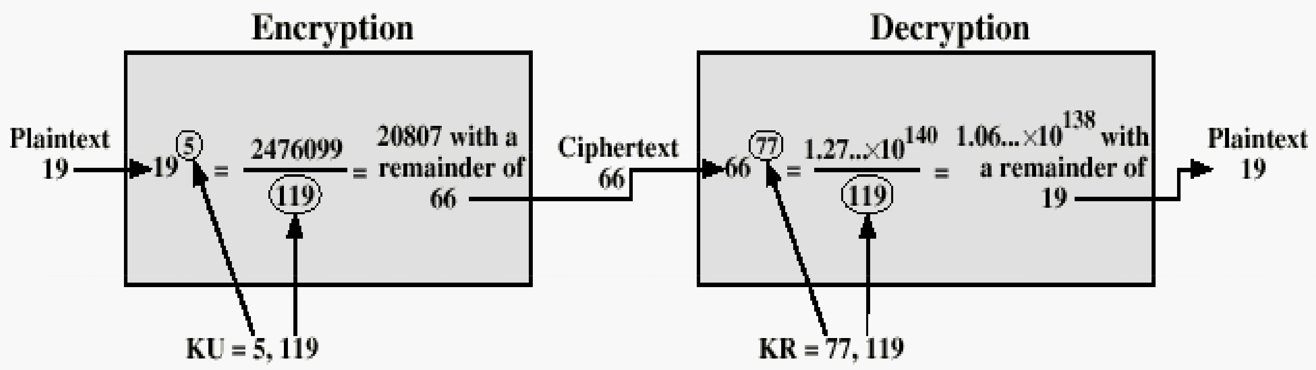


Figure 3.9 Example of RSA Algorithm

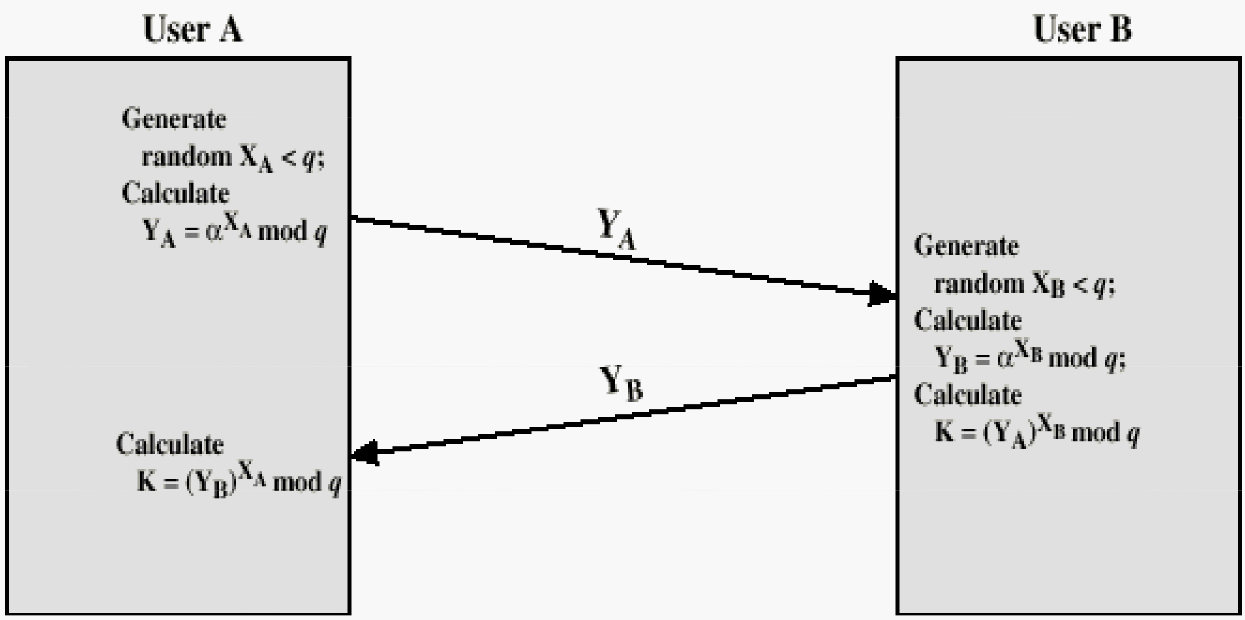
The RSA Algorithm - Encryption

- Plaintext: $M<n$
- Ciphertext: $C = M^e \pmod{n}$

The RSA Algorithm - Decryption

- Ciphertext: C
- Plaintext: $M = C^d \pmod{n}$

Diffie-Hellman Key Echange

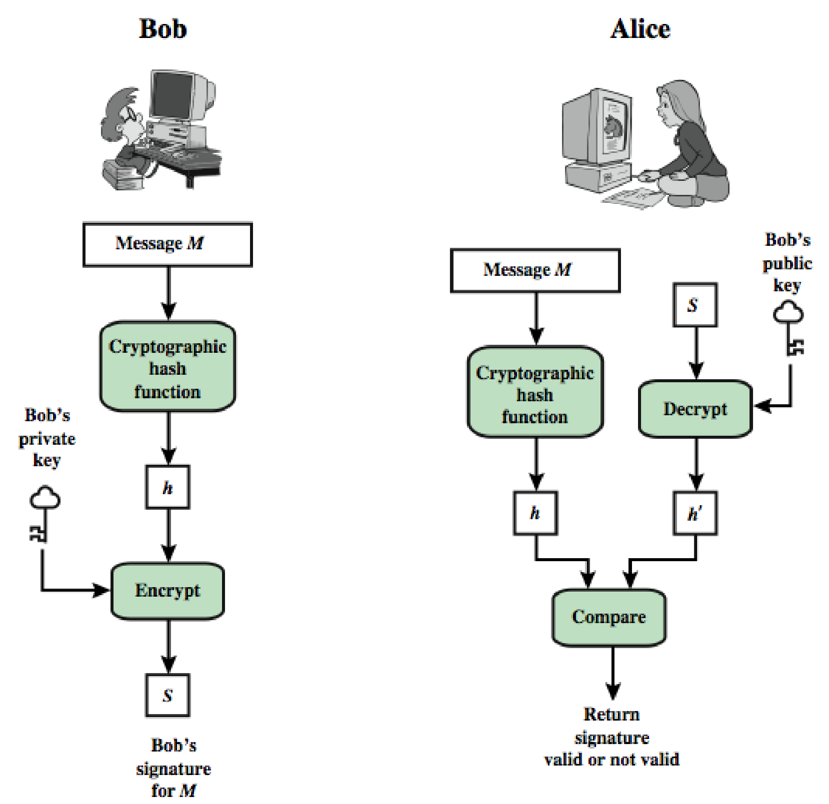


Other Public-Key Cryptographic Algorithms

- Digital Signature Standard (DSS)
  - Makes use of the SHA-1
  - Not for encryption or key echange
- Elliptic-Curve Cryptography (ECC)
  - Good for smaller bit size
  - Low confidence level, compared with RSA
  - Very complex

Digital Signature

- The message is authenticated both in terms of source and in terms of data integrity.



Key Management

- One of the major roles of public-key encryption is to address the problem of key distribution.
  - The distribution of public keys
  - The use of public-key encryption to distribute secret keys
- The point of public-key encryption is that the public key is public.
- A major weakness in public-key encryption
  - Anyone can forge such a public announcement.
- Solution: ***The public-key certificate***
  - A certificate consists of a public key plus a User ID of the key owner, with the whole block signed by a trusted third party, that is a certificate authority (CA).
  - Anyone needing this user's public key can obtain the certificate and verify that it is valid by way of the attached trusted signature.
  - Format of public-key certificates: ***X.509 standard***

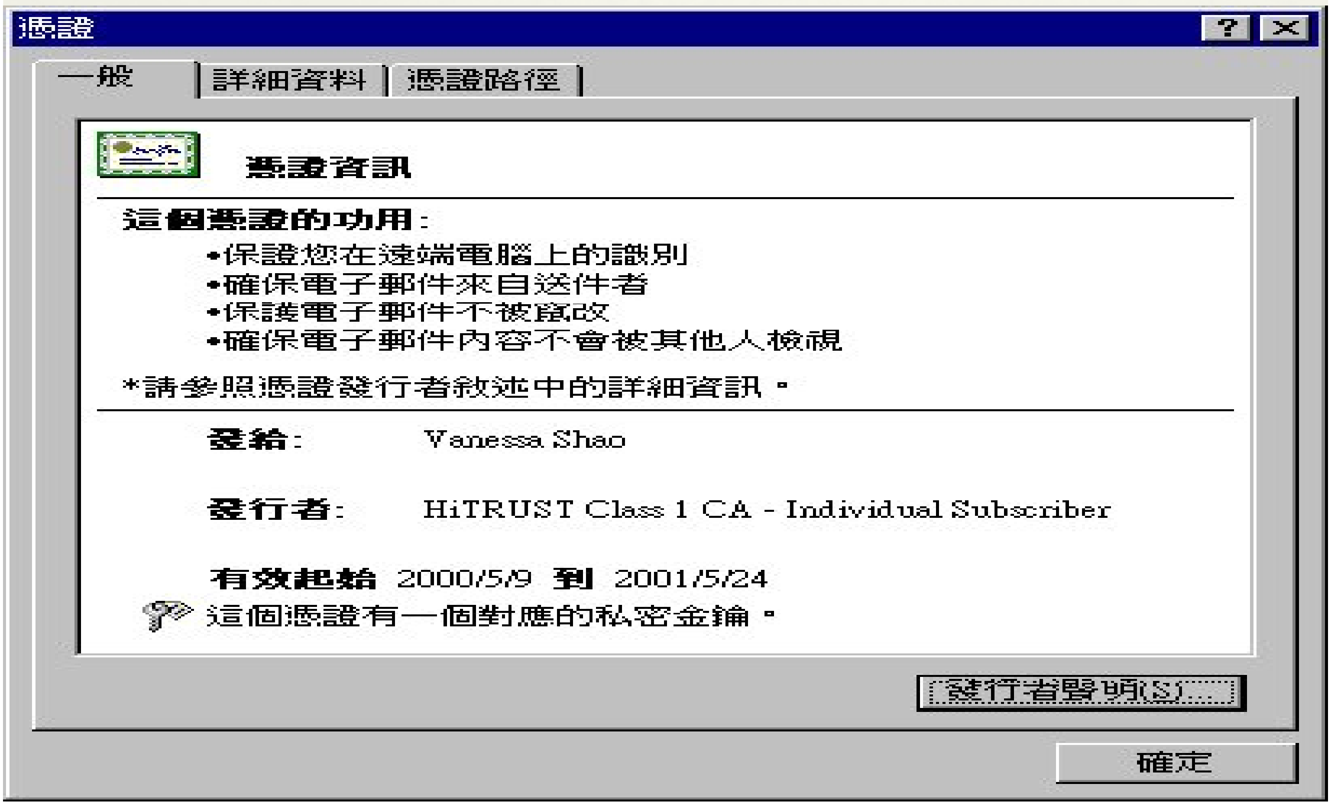
數位憑證

- 記載某人(或某個體)公開金鑰資訊的數位化證書，也稱為「公開金鑰憑證」
  - 記載某人或某一個體的唯一識別、公開金鑰資訊，並附有憑證機構的數位簽章
- 檢驗數位簽章的必要工具
- 憑證機構的簽章使唯一識別和公開金鑰資訊無法分離
  - 確認公開金鑰的真偽與有效性，以達成身分確認
- 大多數的數位憑證皆依循 ITU-T X.509 標準 (ISO/IEC 9594-8) 之規範

X.509 數位憑證的資料項目

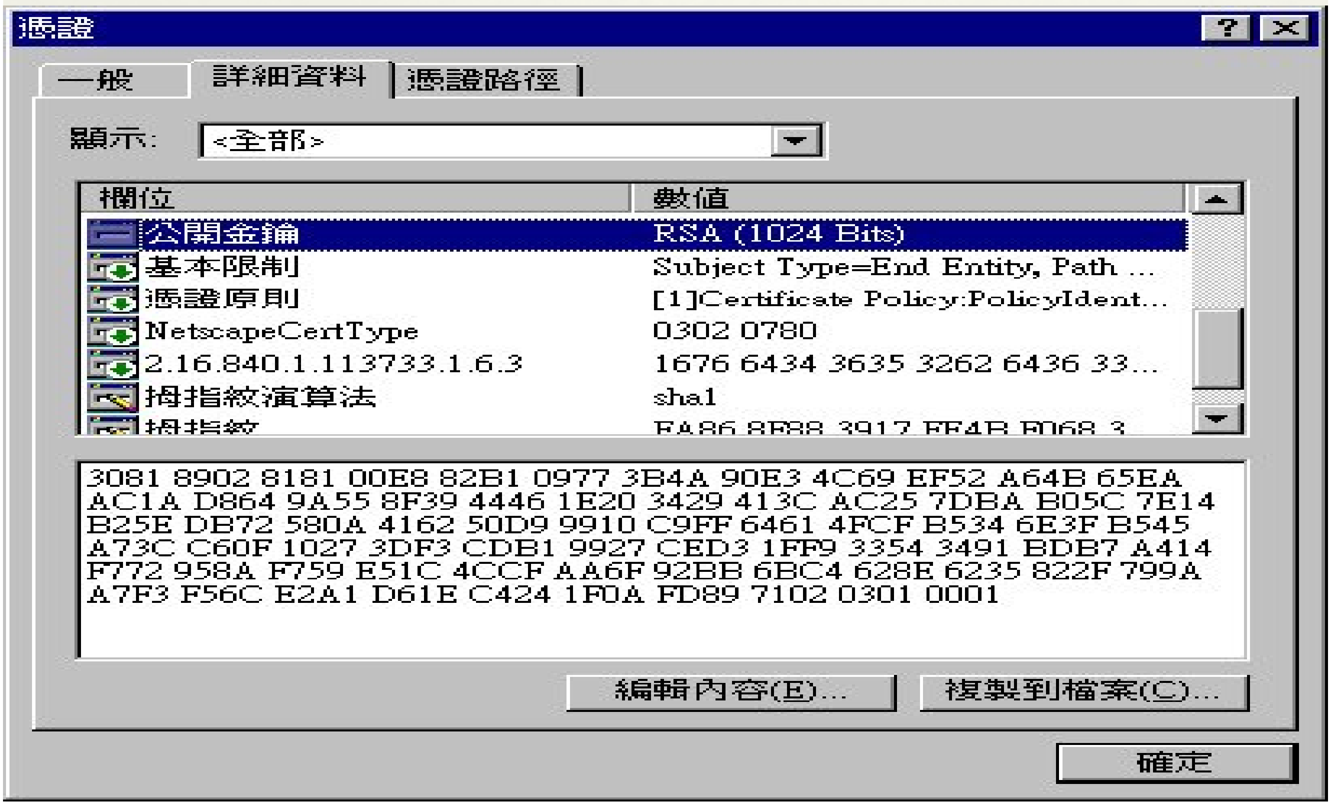
分類	資料項目
主體	唯一識別、公開金鑰
核發機構	唯一識別、數位簽章
有效期間	起始日期、終止日期
行政管理資訊	序號、版本
擴充資訊	各擴充欄位

數位憑證 *(cont.)*



數位憑證 (cont.)

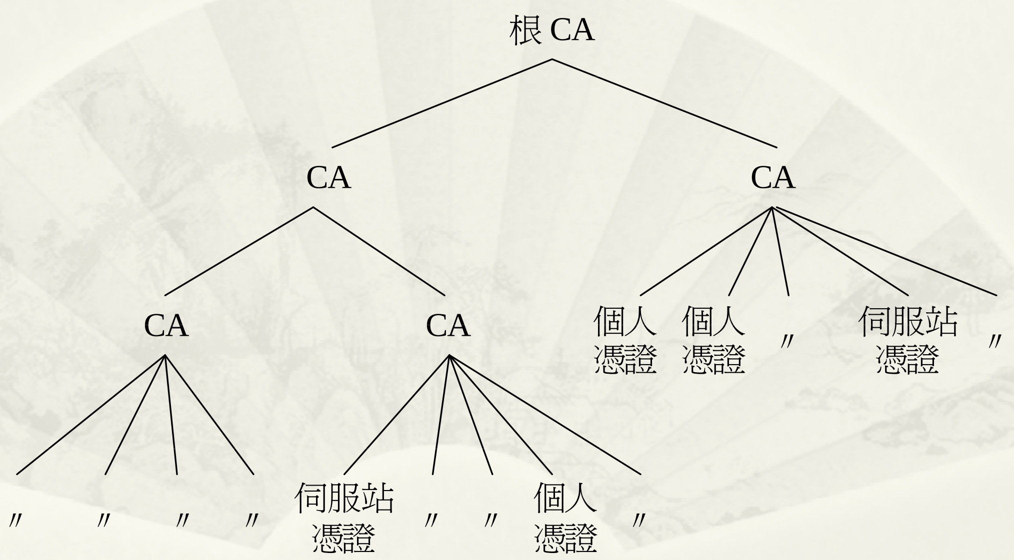
數位憑證 (cont.)



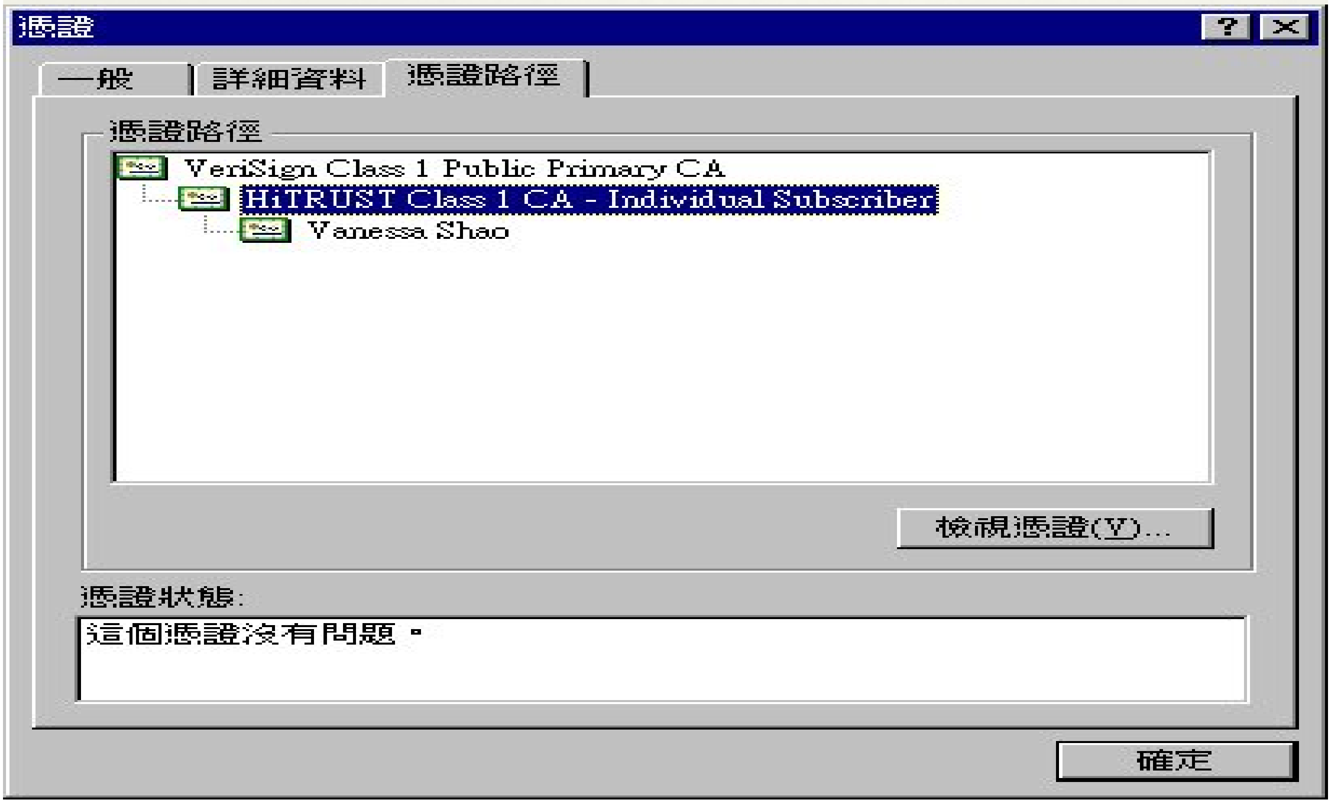
憑證機構 (Certification Authority)

- 簽署並管理公開金鑰憑證的機構 (CA)
- CA 的功能與服務項目
  - 簽發數位憑證
  - 管理憑證的行政工作
    - 決定憑證的有效期間
    - 憑證註銷清單 (CRL) 的維護
  - CA 金鑰管理
- 扮演 CA 角色的機構
  - 公司、學校、城市、提供憑證服務的專業公司

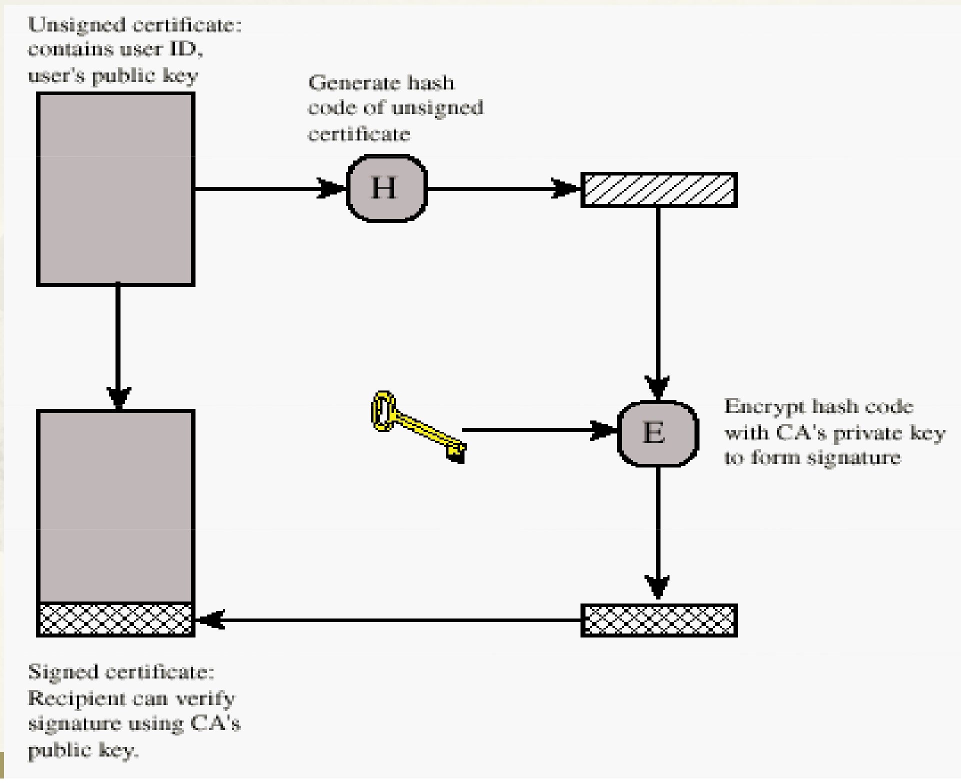
CA 關係圖



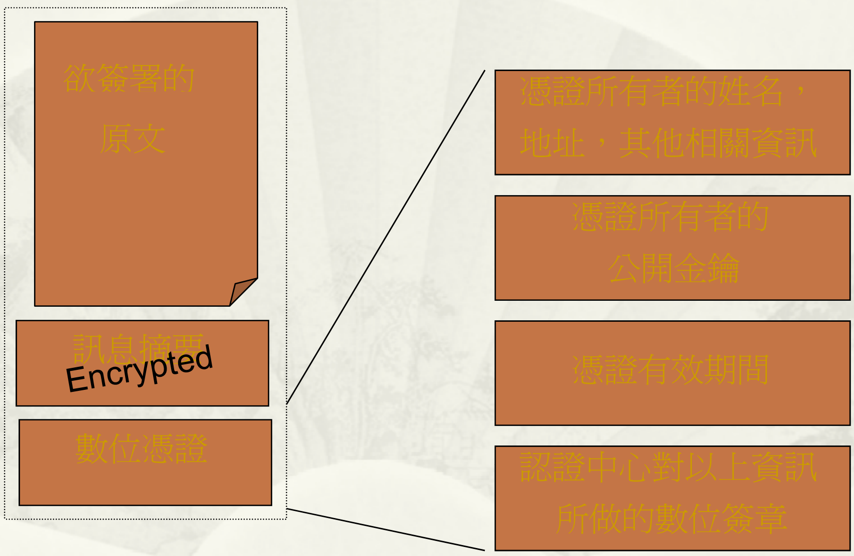
數位憑證 *(cont.)*



Key Management
Public-Key Certificate Use



數位憑證的使用



使用數位憑證以驗證簽章