CS 547: Foundation of Computer Security

S. Tripathy
IIT Patna

Previous class

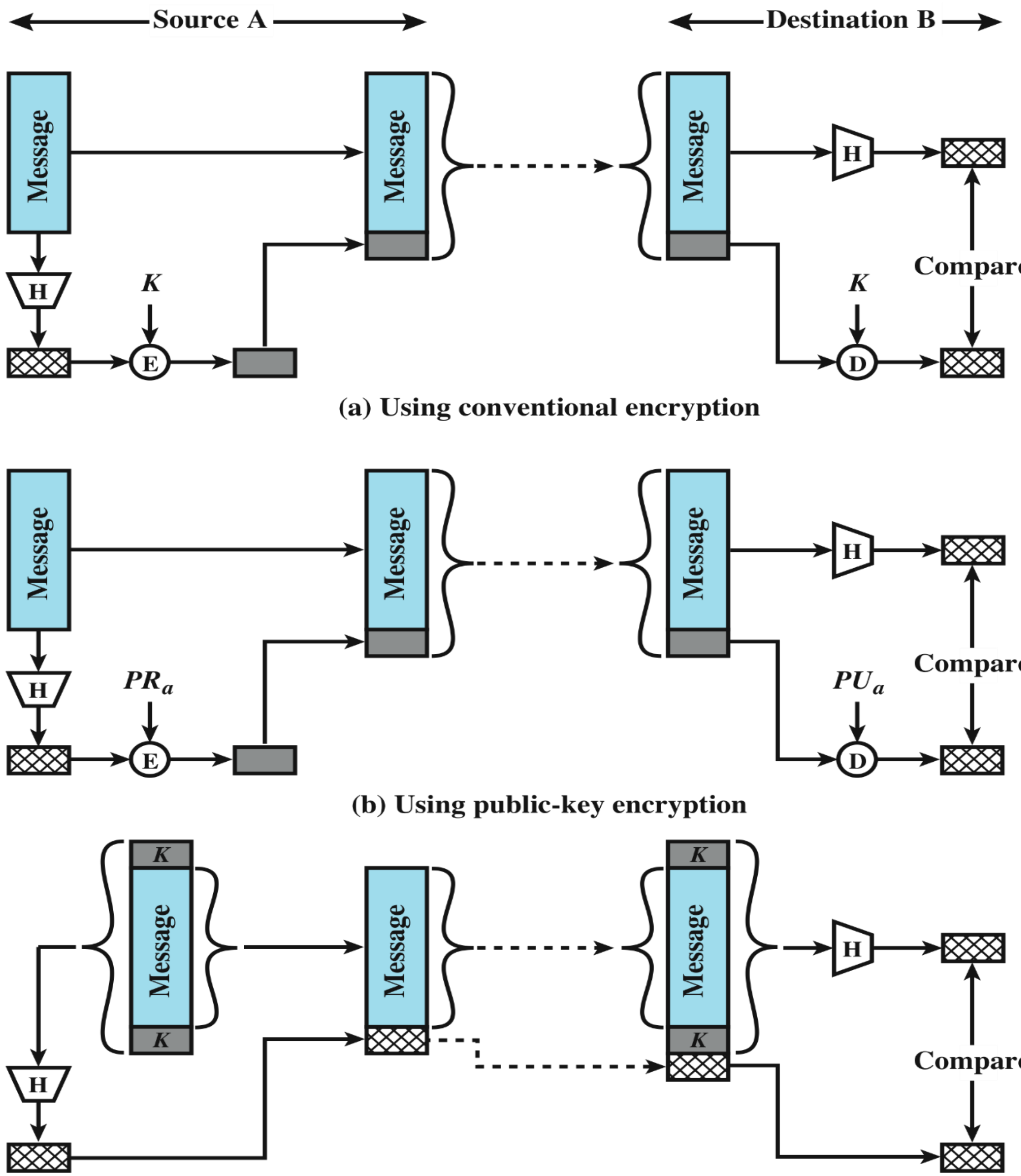
- Crypto Basics
- Cryptographic algorithms
  - important element in security services
- review various types of elements
  - symmetric encryption
  - Hash and MAC

Present class

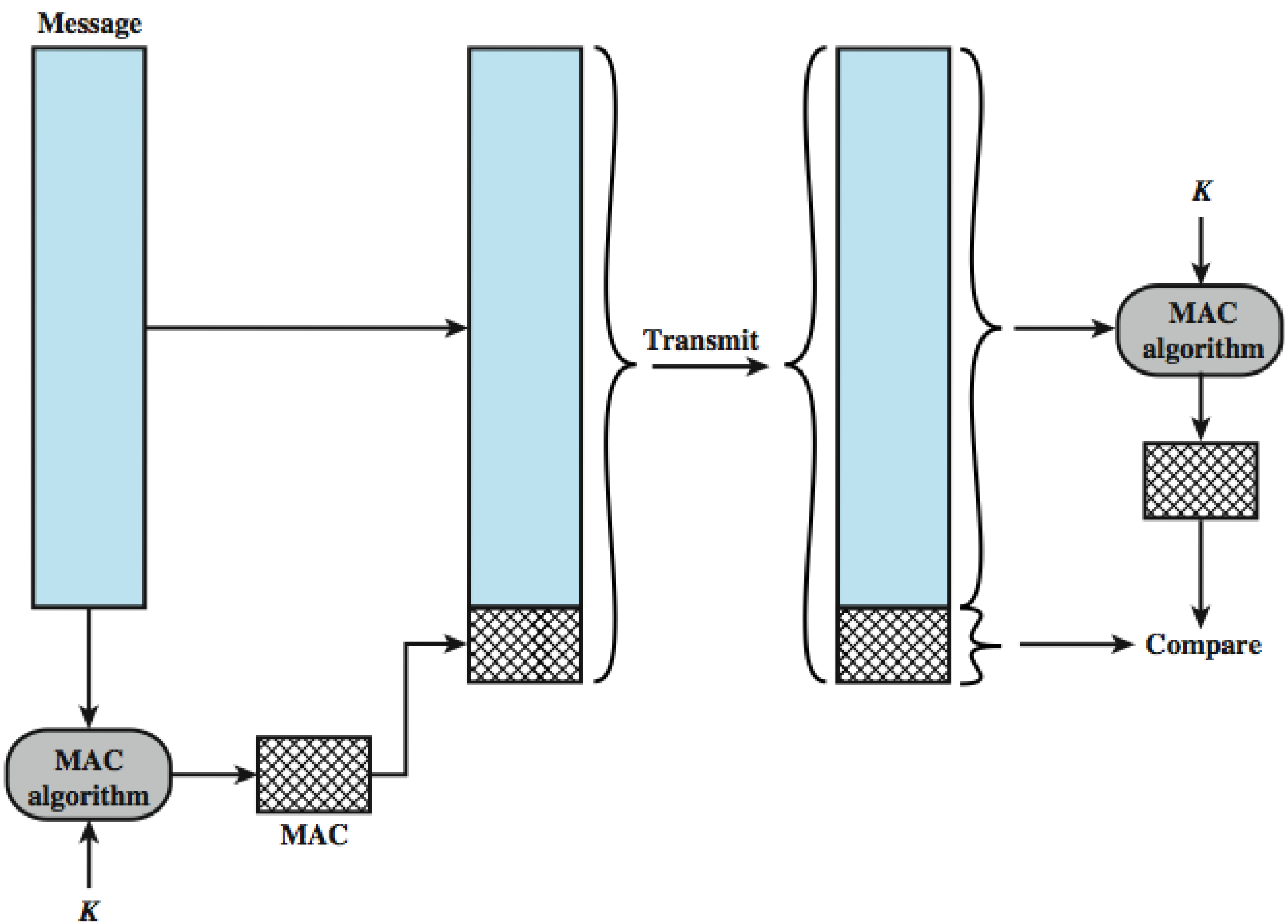
- Crypto Basics

- review various types of elements
  - Public key encryption

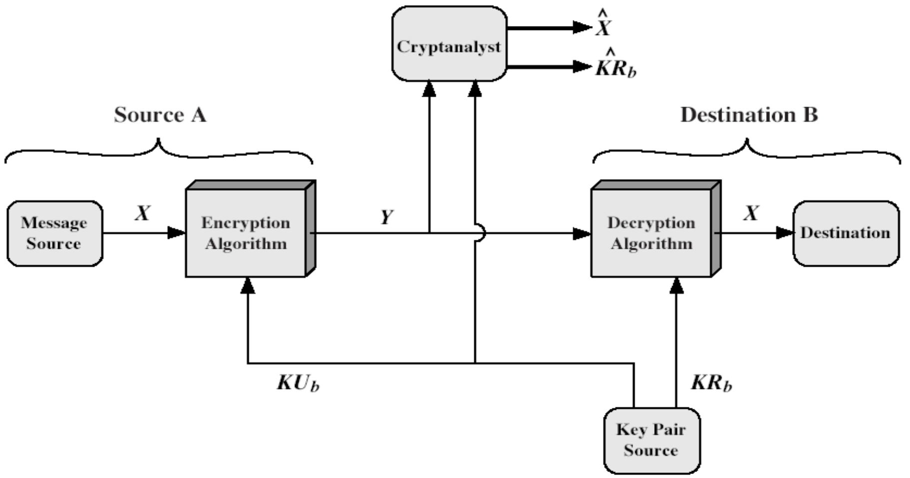
Message Authentication Using a One-Way Hash Function



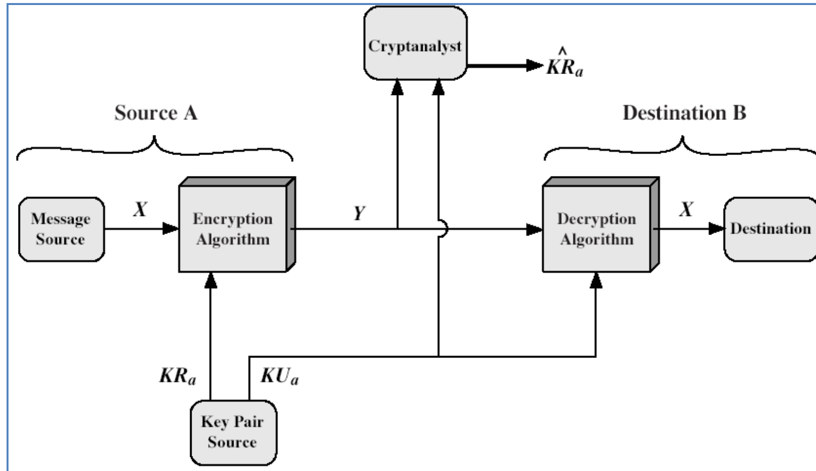
Message Authentication Codes



Public-Key Cryptosystems



Public-Key Cryptosystem: Secrecy



Public-Key Cryptosystem: Authentication

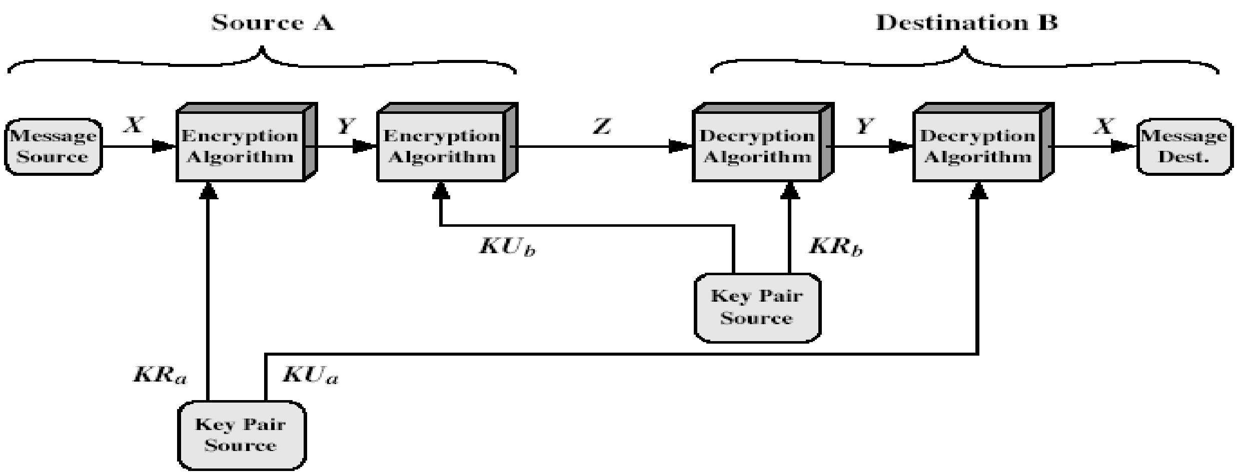


Figure 9.4 Public-Key Cryptosystem: Secrecy and Authentication

Requirements for Public-Key Crypto.

computationally easy to create key pairs

computationally easy for sender knowing public key to encrypt messages

computationally easy for receiver knowing private key to decrypt ciphertext

computationally infeasible for opponent to determine private key from public key

computationally infeasible for opponent to otherwise recover original message

useful if either key can be used for each role

Public key cryptography (RSA)

- RSA:
  - $N=p.q$
  - $\varnothing(N)=(p-1)(q-1)$
  - carefully chosen e & d to be inverses $\bmod \varnothing(N)$
  - hence $e.d=1+k.\varnothing(N)$ for some k
- hence :

$$C^d = (M^e)^d = M^{1+k.\varnothing(N)} = M^1.(M^{k.\varnothing(N)})$$

$$C^d \bmod N = M^1.(1)^k \bmod N = M \bmod N$$

A simple Example

1.Select primes p=11, q=3.

2.n = pq = 11.3 = 33

phi = (p-1)(q-1) = 10.2 = 20

3.Choose e=3

Check gcd(e, phi) = gcd(e, (p-1)(q-1)) = gcd(3, 20) = 1

4.Compute d such that ed ≡ 1 (mod phi)

d = 7

Check: ed-1 = 3.7 - 1 = 20, which is divisible by phi.

5.Public key = (n, e) = (33, 3)

Private key = (n, d) = (33, 7).

Encrypt:

Let us encrypt the message m = 7,

c=m^e mod n=7^3 mod 33 =343 mod 33=13

To check decryption we compute

m'=c^ d mod n=13^7 mod 33 = 7

Digital Signatures

- used for authenticating both source and data integrity
- created by encrypting hash code with private key
- does not provide confidentiality
  - even in the case of complete encryption
  - message is safe from alteration but eavesdropping

RSA Signature Example

- First key gen: $p \leftarrow 7, q \leftarrow 13, n \leftarrow pq = 91, e \leftarrow 5, d \leftarrow 29$
- Thus your public key is (e,n) and your private key is d.
- Say we want to sign the message $m=35$
- , we compute $s = m^d \bmod n$ which is $s \leftarrow 42 \equiv 3529 \bmod n$.
- The message and signature get sent to the other party $(m,s)=(35,42)$
- . Who takes the signature and raises it to the e modulo n, or $42^5 \equiv 35 \bmod n$. Then makes sure that this value is equal to the message that was received, which it is, so the message is valid.

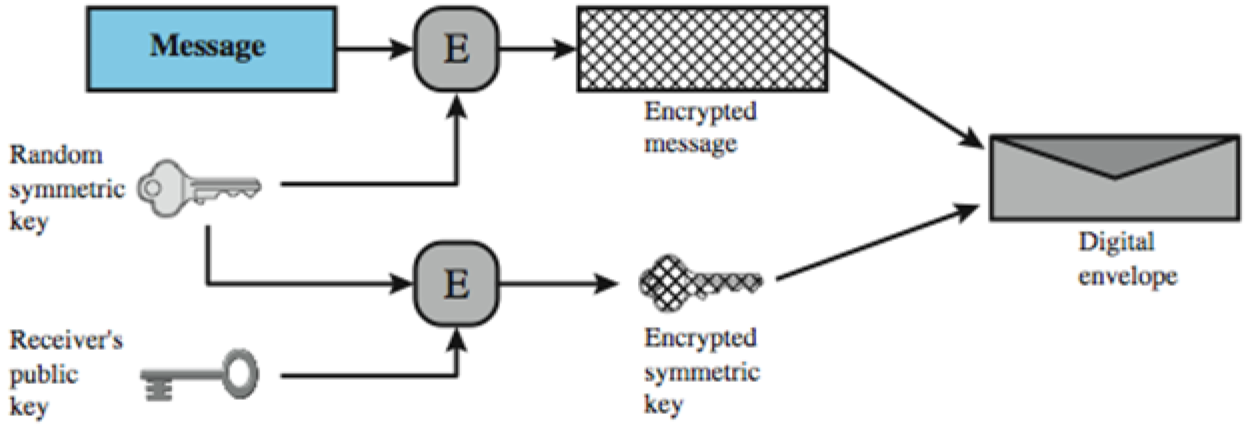
Message Encryption

- Message Encryption
  - Secret key encryption vs. public key encryption
    - Both encryption algorithms can provide confidentiality
    - Secret Key Encryption is more efficient and faster
  - To use secret key encryption
    - Communicating peers must share the same key
    - The key must be protected from access by others

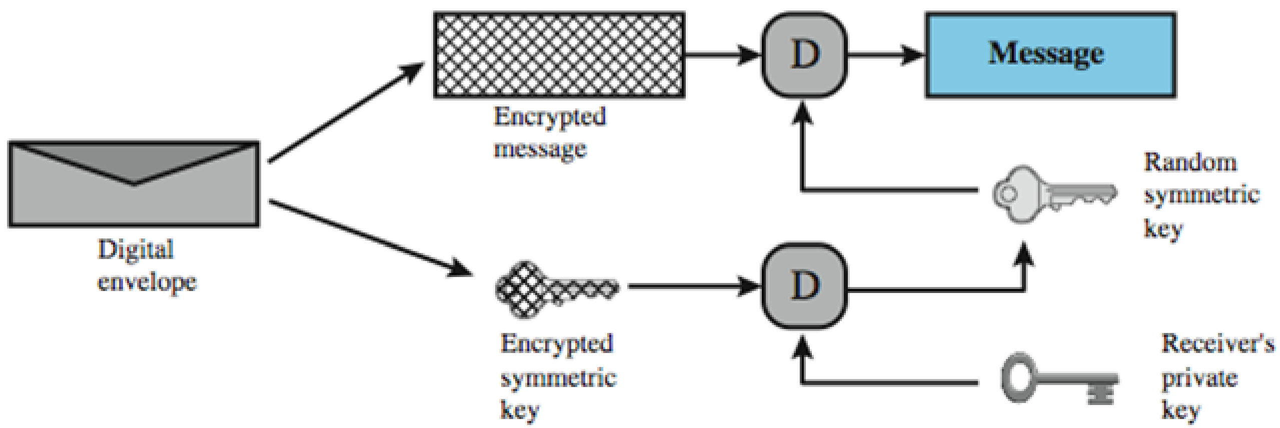
Digital Envelopes

- protects a message without needing to first arrange for sender and receiver to have the same secret key

- equates to the same thing as a sealed envelope containing an unsigned letter

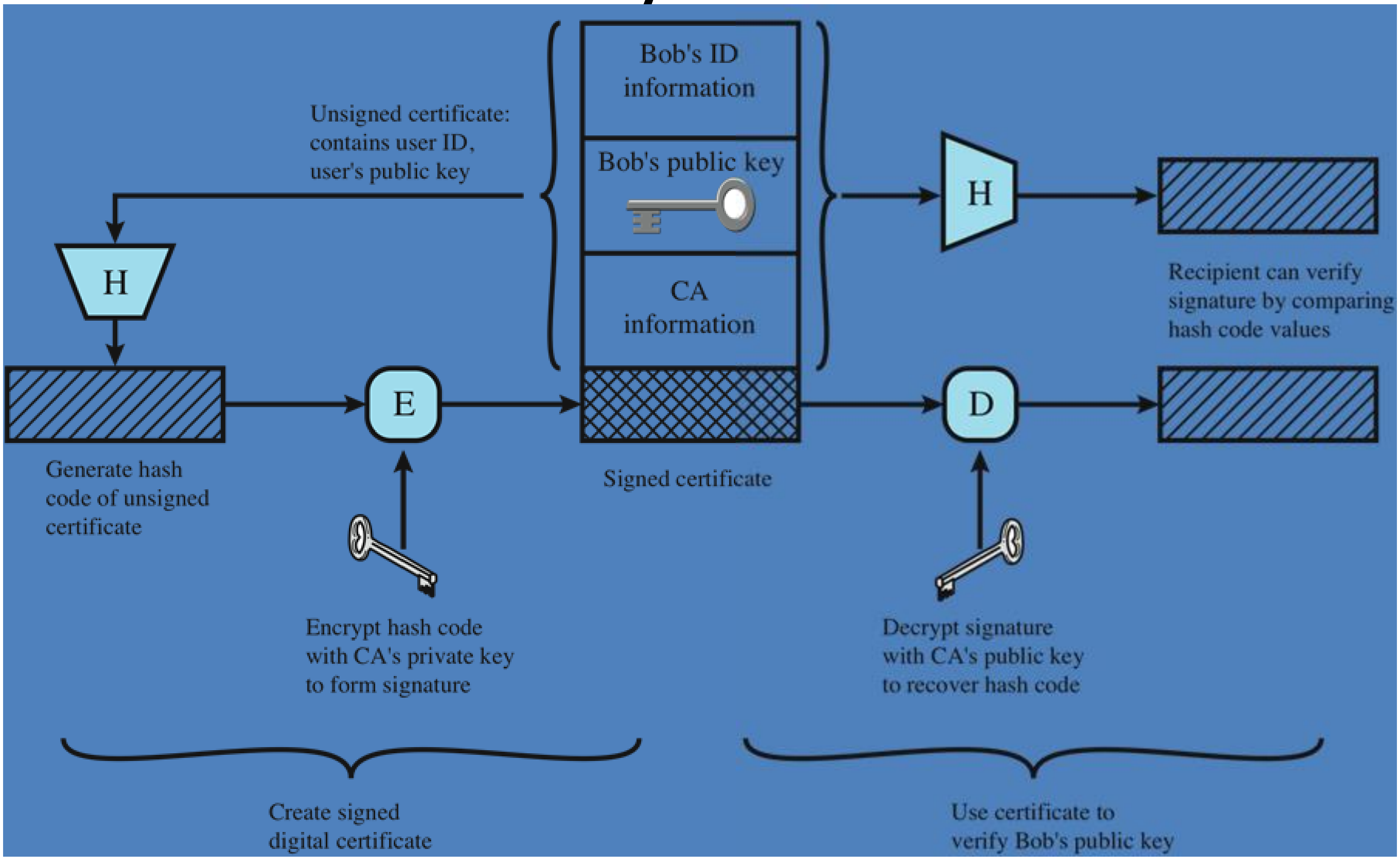


(a) Creation of a digital envelope



(b) Opening a digital envelope

Public Key Certificates



Random Numbers

- **Uses include 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 use algorithms 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

Table 1.6 Relationship Between Security Services and Mechanisms

Service	Mechanism: Encipherment	Digital signature	Access control	Data integrity	Authentication exchange	Traffic padding	Routing control	Notarization
Peer entity authentication	Y	Y			Y			
Data origin authentication	Y	Y						
Access control			Y					
Confidentiality	Y						Y	
Traffic flow confidentiality	Y					Y	Y	
Data integrity	Y	Y		Y				
Non-repudiation		Y		Y				Y
Availability				Y	Y			

Thanks