

# T-79.4501

## Cryptography and Data Security

Lecture 9:

- Principles of authentication
- Digital signatures
- DSS

Stallings: Ch. 11.1-2; 13.1; 13.3

# Principles of message authentication

Attacks against message security:

- Disclosure
- Traffic analysis
- Masquerade (impersonate); this is what a man-in-the-middle does
- Content modification
- Sequence modification
- Timing modification; replay
- Source repudiation
- Destination repudiation

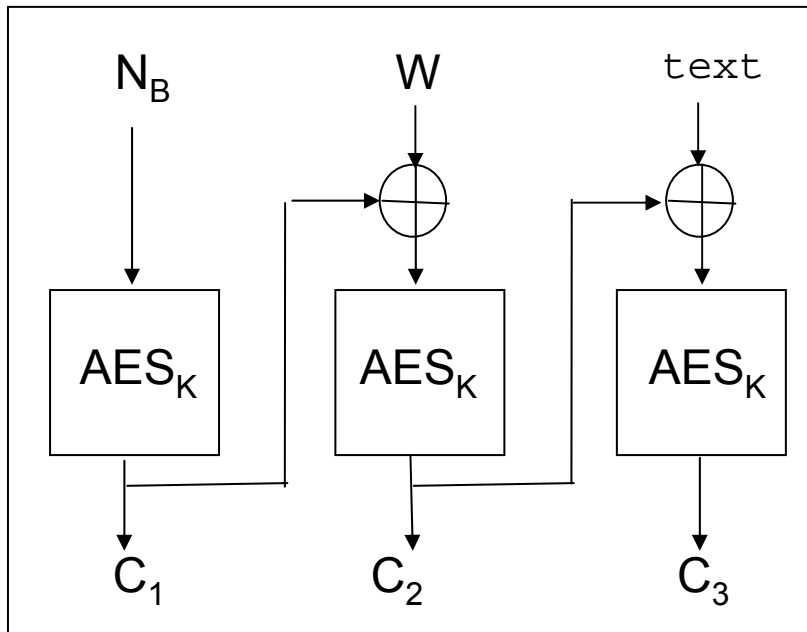
Message authentication can be used to prevent from these attacks.

# Authentication functions

- Authentication functions are cryptographic primitives which are used by message authentication protocols between two parties, sender and receiver. Sender attaches to the message an authenticator. Receiver uses the authenticator to verify authenticity of the message.
- Authentication functions:
  - Message encryption (with integrity protection)
  - Message authentication code (MAC function)
  - Hash function
  - Digital signature
- Note. Message encryption even with a block cipher does not provide message integrity, see next slide.

# Non-integrity of CBC encryption

- Bob wants to verify the liveness of Alice's love and receive a fresh new key
- Alice's message  $M = W \parallel \text{"I love you"}$ , where  $W$  is a 128-bit key
- Encryption is CBC with 128-bit block cipher (AES)
- $N_B$  is a 128-bit value;  $(C_1, C_2, C_3) = E_K(N_B, M)$



```
text = 49 20 6c 6f 76 65 20 79 6f 75
      Δ = 00 00 04 0e 02 00 00 00 00 00
text' = 49 20 68 61 74 65 20 79 6f 75
```

- Malice changes the second ciphertext block to  $C_2' = C_2 \oplus \Delta$
- After decryption Bob reads  $M' = W' \parallel \text{"I hate you"}$  where  $W'$  is a random 128-bit value

# Message Authentication Protocols

Messages are sent from Alice to Bob:

Authenticity requirements:

1. Bob can verify that Alice sent the message
2. Bob can verify that the contents of the message is as it was when Alice sent it.
3. Bob can prove to Carol that Alice sent the message
4. Bob can prove to Carol what the message contents was when Alice sent it.
5. Alice cannot deny that she sent the message.

Requirements 1 and 2 can be fulfilled using protocols based on symmetric key authentication functions.

Requirements 3-5 can be fulfilled only using protocols based on asymmetric (public key) cryptosystems: Digital Signatures

# Authentication Notions:

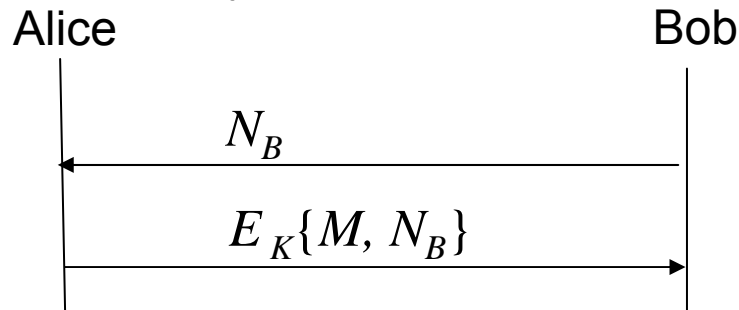
## Data authentication

- Data (data-origin, message) authentication involves
  - Communications, receiver and transmitter
  - identifying the source of the data
  - freshness of a message (non-replay)
- Successful validation by the receiver establishes
  - the identity of the message transmitter
  - liveness (at some point) of the message transmitter
  - integrity of the data subsequent to being transmitted

# Basic Authentication Techniques: Message Freshness and Principal Liveness

## Challenge-Response Mechanism

- Alice and Bob share a key  $K$  of an encryption algorithm.
- Alice has a message  $M$ , she wants to transmit to Bob.
- Bob wants verify the freshness of  $M$  and liveness of Alice



- It is necessary that the algorithm  $E_K$  offers data-integrity (see page 4). If confidentiality is not needed then better to use a message authentication algorithm.

# Authentication Notions:

## Entity authentication

- Entity Authentication is
  - is a lively correspondence by a principal with a second principal
  - Aims to corroborate the identity of the second entity
  - Entity authentication is very rarely the only goal of the security protocol
  - Entity authentication may be performed by other than cryptographic means (e.g., manual authentication)



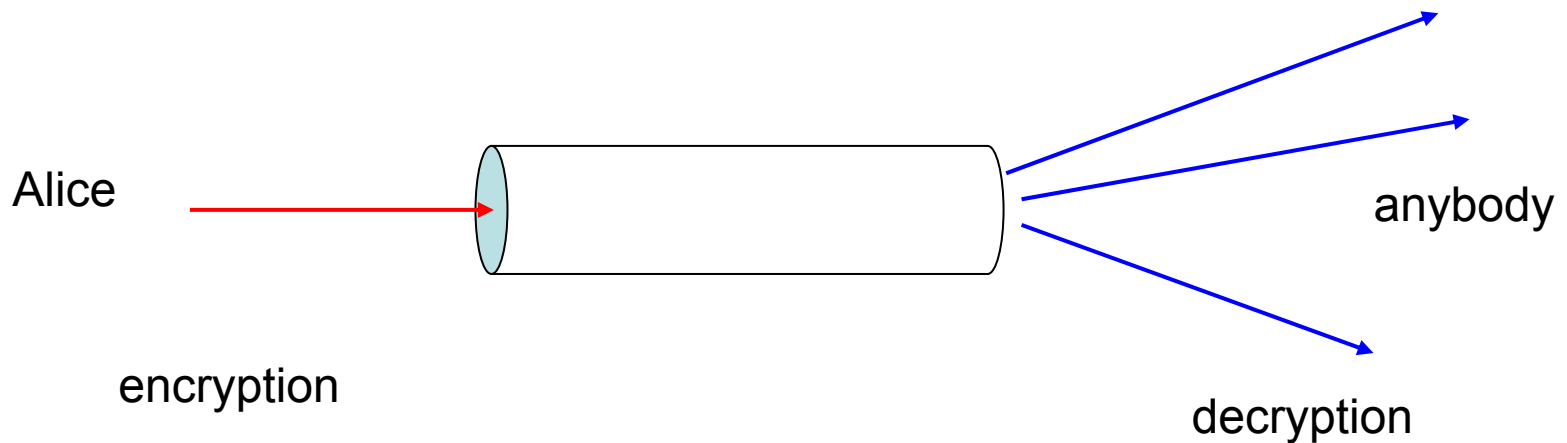
# Authentication Notions: Authenticated Key Agreement

- Authenticated Key Agreement involves
  - Establishment of a cryptographic key between two entities
  - Data authentication of the established cryptographic key
- Security
  - Authentication protocol is flawed if a principal concludes a normal run of the protocol while the intended other principal would have a different conclusion.
  - A flaw in a protocol does not necessarily imply a flaw in the cryptographic algorithms used in the protocol.
  - Important to validate the suitability of a cryptographic algorithm for the protocol. Here theoretical models and security proofs are useful.

# Authentication function based on asymmetric cryptography

Encryption operation is private

Decryption is a public operation



Alice's key for a public key cryptosystem is a pair:  
 $(K_{\text{pub}}, K_{\text{priv}})$  where  $K_{\text{pub}}$  is public and  $K_{\text{priv}}$  cannot be used by anybody else than Alice.

# Digital Signature

## Two types

- *Digital signature with message recovery*: the entire message is encrypted using the private key; before encryption some verifiable redundancy must be added to the message. The message authenticator is the entire ciphertext.
- *Digital signature with appendix*: First a hash code is computed from the message. Then the hash code is encrypted using private key. The encrypted hash code is the authenticator, which is appended to the cleartext message.

# The RSA Digital Signature (with appendix)

- Key derivation: the same as in RSA encryption:  
 $n = pq$ ,  $p$ ,  $q$  two different primes,  $e$  public exponent,  $d$  private exponent,  $ed \bmod \phi(n) = 1$
- RSA authenticator generation function: given data  $D$  the authenticator  $S$  of  $D$  is computed as  $S = D^d \bmod n$
- RSA verification function: given  $S$ , the RSA verification function is computed as  $S^e \bmod n$
- Hash function: any hash function allowed
- EMSA-PKCS1-v1\_5 Formatting of  $D$  is specified in PKCS#1 (octet string):

$$D = 0x00 \parallel 0x01 \parallel \{\text{at least eight times } 0xFF\} \parallel 0x00 \parallel T ,$$

where  $T$  is the ASN.1 encoding of the hash type and the hash code of the message.  $\parallel$  denotes concatenation of octet strings. The number of all-one octets  $0xFF$  in the middle is chosen to adjust the length of  $D$  at most equal to the length of the modulus  $n$ .

# The Digital Signature Algorithm DSA

- FIPS 186-2 (2000)
- DSA is a digital signature with appendix
- The complete specification defines:
  - The asymmetric cryptosystem: Key derivation, private key operation (for signature creation), public key operation (for signature verification)
  - Prime number generation
  - The hash function
  - Pseudo-random number generator

# The DSA public key cryptosystem

## Global public key components

$p$  is a 1024-bit prime

$q$  a prime divisor of  $p - 1$ , where  $q$  is a 160-bit number

$g = h^{(p-1)/q} \bmod p$ , where  $h$  is any integer such that  $1 < h < p - 1$  and  $h^{(p-1)/q} \bmod p \neq 1$ .

(Then the order of the group  $\langle g \rangle$  generated by  $g$  in  $Z_p^*$  is equal to  $q$ .)

## User's private key

$x$  random or pseudo-random integer with  $0 < x < q$

## User's public key

$y = g^x \bmod p$

# DSA: Signature generation

- Message  $M$
- Hash code  $H = \text{SHA-1}(M)$  taken as integer
- per-message randomizer  $k$  :  $k$  is a secret random or pseudorandom integer  $0 < k < q$
- The first part  $r$  of the signature :

$$r = (g^k \bmod p) \bmod q$$

- The second part  $s$  of the signature:

$$s = k^{-1} \cdot (H + r \cdot x) \bmod q$$

Private key  
used here!

- The signed message is  $M, (r, s)$ , where  $(r, s)$  is the authenticator appended to the message  $M$

# DSA: Signature verification

Verifier has  $p, q, g$  and  $y$ .

Verifier receives:  $M', (r', s')$  and computes:

$$H' = \text{SHA-1}(M')$$


$$w = (s')^{-1} \bmod q$$

$$u_1 = w \cdot H' \bmod q$$

$$u_2 = w \cdot r' \bmod q$$

$$v = g^{u_1} y^{u_2} \bmod p$$

Public key  
used here!



and checks if  $v \equiv r' \pmod{q}$ .



# Why DSA verification works

$$\begin{aligned}v &= g^{u_1} y^{u_2} \bmod p = g^{w \cdot H} y^{w \cdot r} \bmod p \\ &= g^{w \cdot H} g^{w \cdot a \cdot r} \bmod p = g^{w \cdot H} g^{w \cdot a \cdot r} \bmod p \\ &= g^{s^{-1} \cdot (H + a \cdot r)} \bmod p = g^k \bmod p\end{aligned}$$

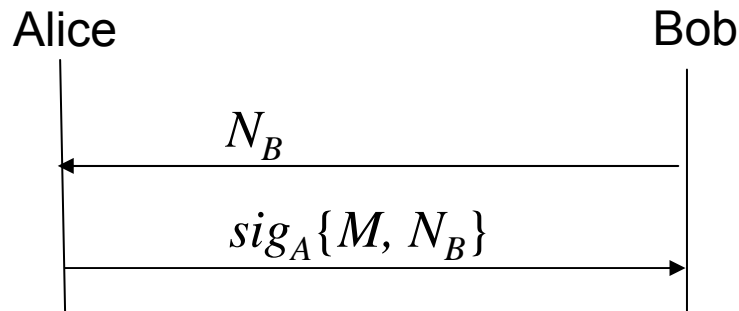
Hence  $v \equiv r \pmod{q}$ .

If verification is successful, then (with large probability)

- The received values are correct
- The entity with knowledge of the secret key has generated the signature.

# Challenge-Response Mechanism using digital signature

- Alice uses digital signature mechanism, Bob has Alice's public key.
- Alice has a message  $M$ , she wants to transmit to Bob.
- Bob wants verify the freshness of  $M$  and liveness of Alice



- Here  $sig_A\{D\}$  means the signed message (= message + authenticator) signed by Alice.
- Alice's free choice of  $M$  is important. Also,  $N_B$  shall never be taken to have some other meaning as the random challenge. Otherwise Bob can compute it as a hash of some contract beneficial to him.