Cryptography and Network Security

Eighth Edition

by William Stallings

Chapter 13

Digital Signatures

What is Digital Signature

- The most important development from the work on public-key cryptography is the digital signature.
- The digital signature provides a set of security capabilities that would be difficult to implement in any other way.

See the next slide for depiction of this Fig.

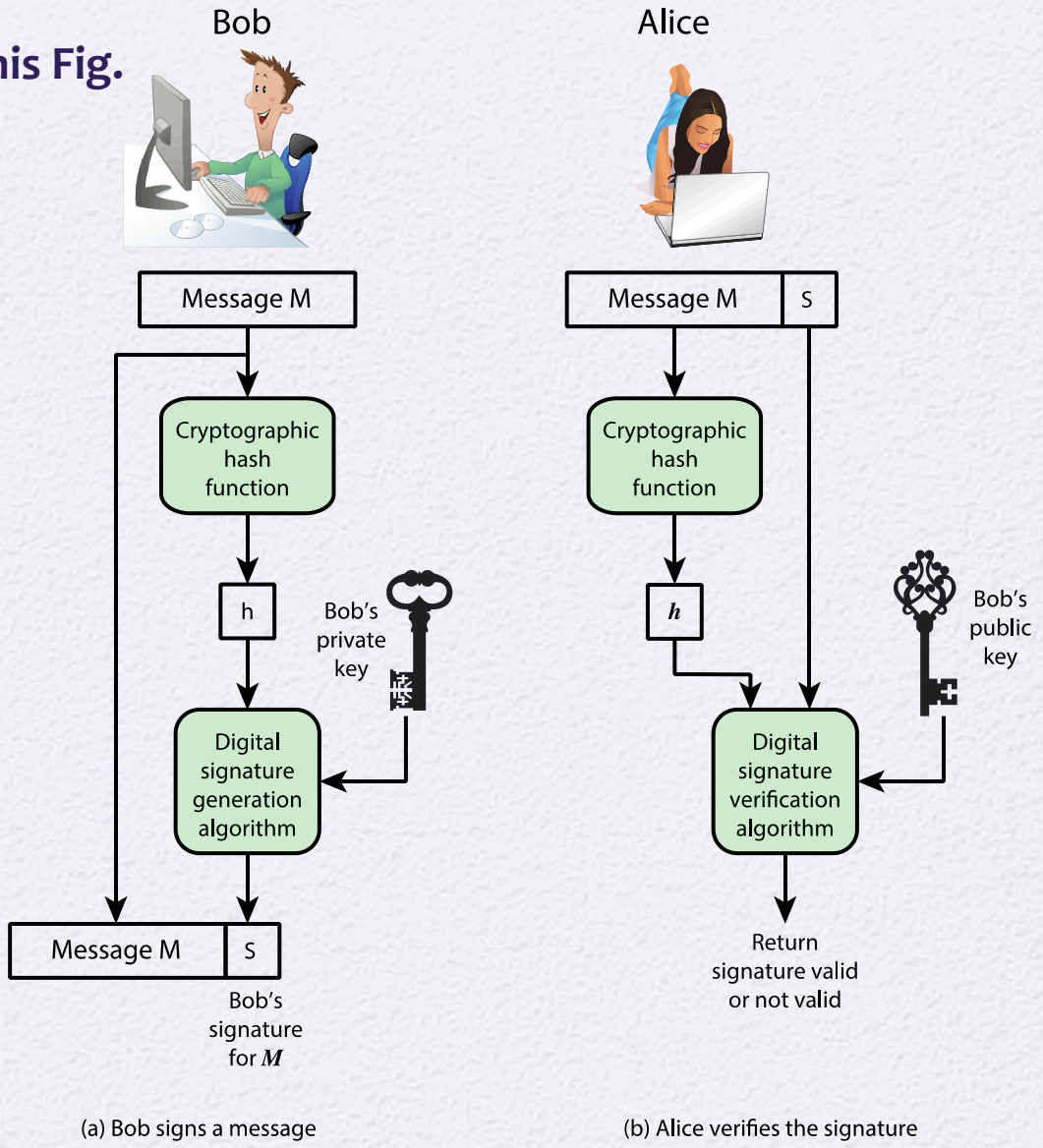


Figure 13.1 Simplified Depiction of Essential Elements of Digital Signature Process

- Figure 13.1 is a generic model of the process of constructing and using digital signatures.
- Suppose that Bob wants to send a message to Alice. For this purpose, Bob uses a secure hash function, such as SHA-512, to generate a hash value for the message.
- A digital signature generation algorithm takes this hash value, together with Bob's private key, and produces a short block that functions as a digital signature

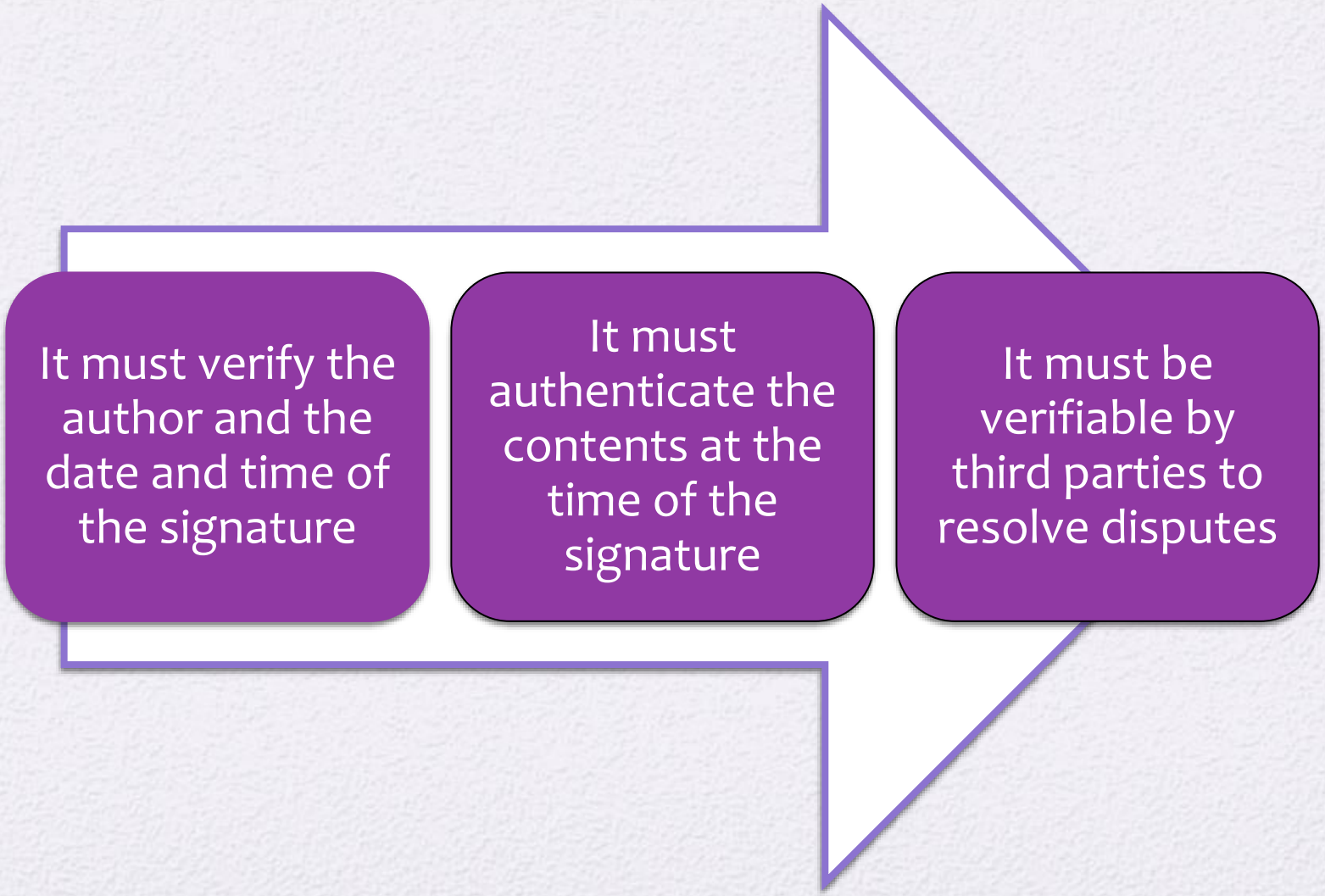
Digital Signature Properties

In situations where there is not complete trust between sender and receiver, something more than authentication is needed. The most attractive solution to this problem is the digital signature. The digital signature must have the following properties:

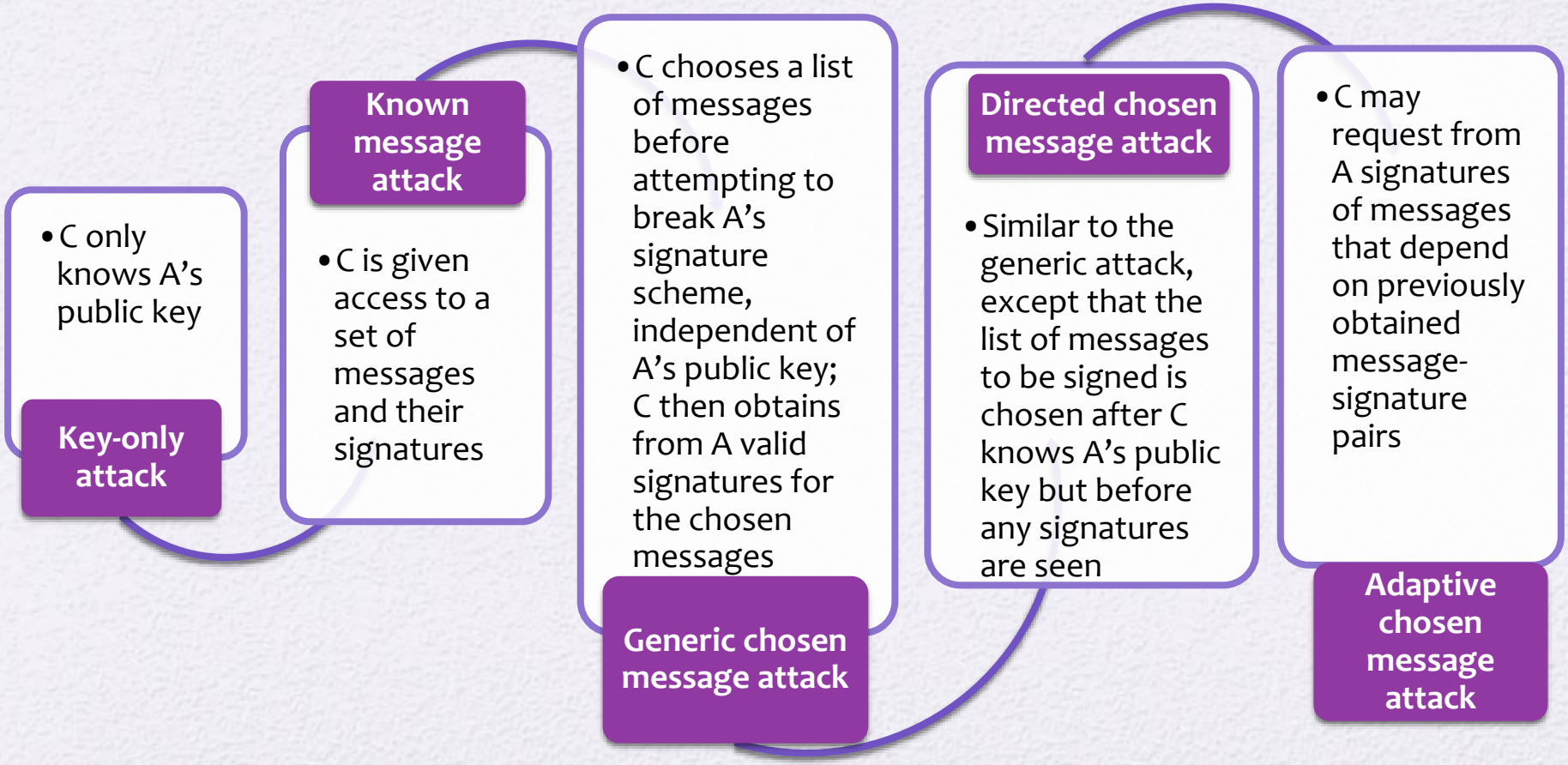
- It must verify the author and the date and time of the signature.
- It must authenticate the contents at the time of the signature.
- It must be verifiable by third parties to resolve disputes.

Thus, the digital signature function includes the authentication function.

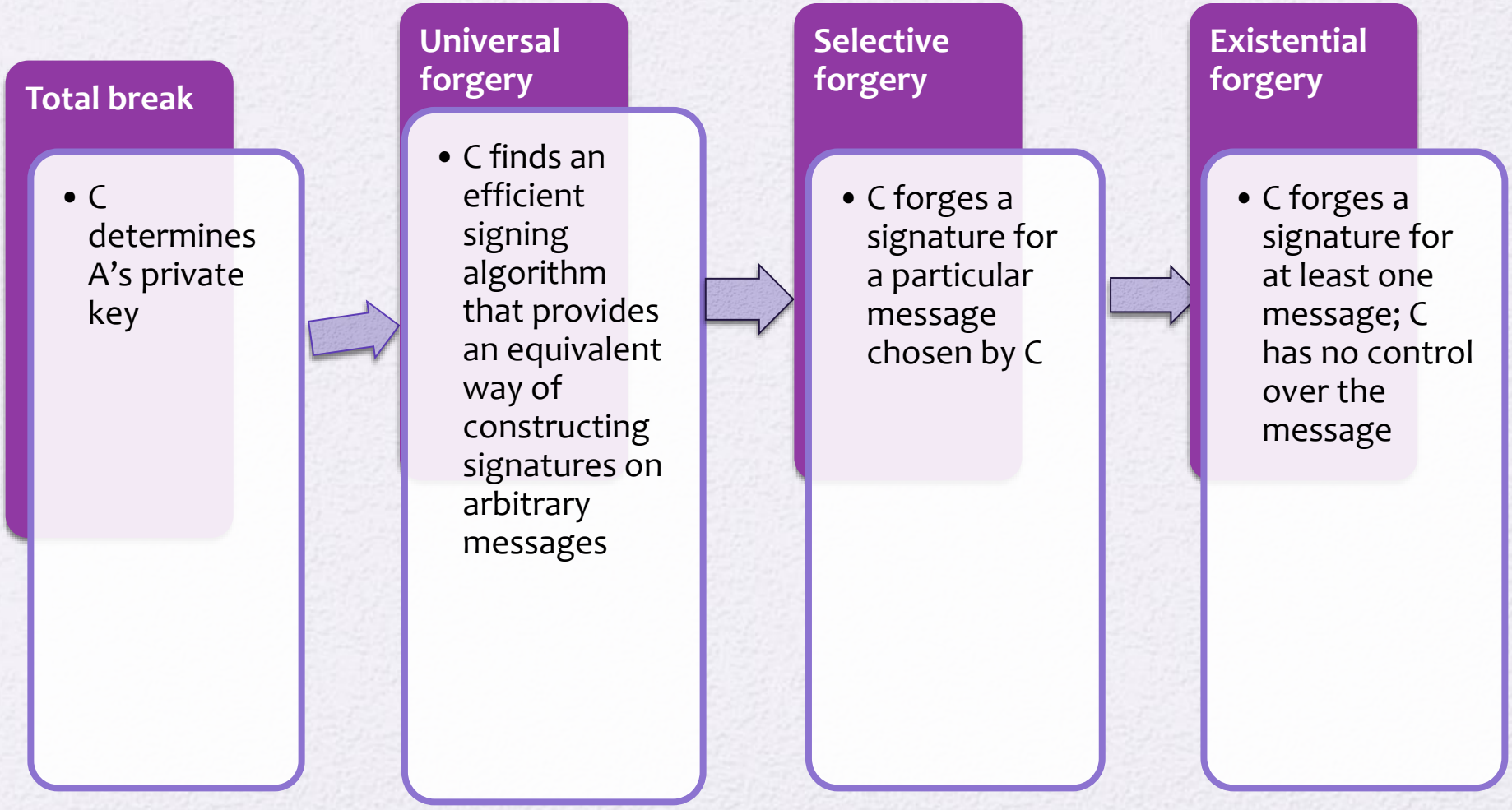
Digital Signature Properties



Attacks



Forgeries



Digital Signature Requirements

- The signature must be a bit pattern that depends on the message being signed
- The signature must use some information unique to the sender to prevent both forgery and denial
- It must be relatively easy to produce the digital signature
- It must be relatively easy to recognize and verify the digital signature
- It must be computationally infeasible to forge a digital signature, either by constructing a new message for an existing digital signature or by constructing a fraudulent digital signature for a given message
- It must be practical to retain a copy of the digital signature in storage

Direct Digital Signature

Refers to a digital signature scheme that involves only the communicating parties

- It is assumed that the destination knows the public key of the source

Confidentiality can be provided by encrypting the entire message plus signature with a shared secret key

- It is important to perform the signature function first and then an outer confidentiality function
- In case of dispute some third party must view the message and its signature

The validity of the scheme depends on the security of the sender's private key

- If a sender later wishes to deny sending a particular message, the sender can claim that the private key was lost or stolen and that someone else forged his or her signature
- One way to thwart or at least weaken this ploy is to require every signed message to include a timestamp and to require prompt reporting of compromised keys to a central authority

ElGamal Digital Signature

- Scheme involves the use of the private key for encryption and the public key for decryption (its security based difficulty of computing discrete logarithms, as in D-H).
- Global elements are a prime number *q* and a, which is a primitive root of *q*
- Use private key for encryption (signing)
- Uses public key for decryption (verification)
- Each user generates their key
  - Chooses a secret key (number): $1 < x_A < q-1$
  - Compute their public key: $y_A = a^{xA} \bmod q$

ElGamal Digital Signature

- Alice signs a message M to Bob by computing
  - the hash $m = H(M)$, $0 <= m <= (q-1)$
  - chose random integer K with $1 <= K <= (q-1)$ and $\gcd(K, q-1) = 1$
  - compute temporary key: $S_1 = a^k \bmod q$
  - compute K^{-1} the inverse of $K \bmod (q-1)$
  - compute the value: $S_2 = K^{-1}(m - x_A S_1) \bmod (q-1)$
  - signature is: (S_1, S_2)
- any user B can verify the signature by computing
  - $V_1 = a^m \bmod q$
  - $V_2 = y_A^{S_1} S_1^{S_2} \bmod q$
  - signature is valid if $V_1 = V_2$

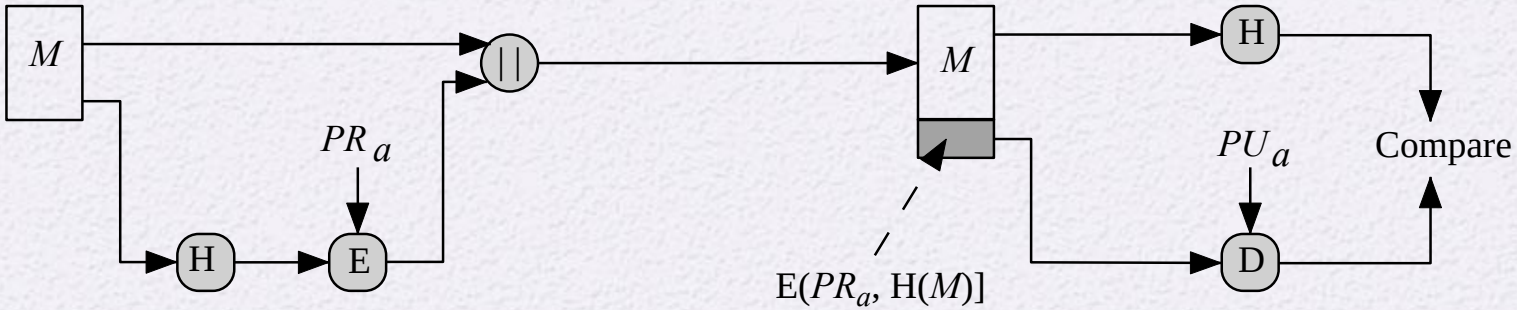
ElGamal Signature Example

- use field GF(19) $q=19$ and $a=10$
- Alice computes her key:
  - A chooses $x_A=16$ & computes $y_A=10^{16} \bmod 19 = 4$
- Alice signs message with hash $m=14$ as $(3,4)$:
  - choosing random $K=5$ which has $\gcd(18,5)=1$
  - computing $S_1 = 10^5 \bmod 19 = 3$
  - finding $K^{-1} \bmod (q-1) = 5^{-1} \bmod 18 = 11$
  - computing $S_2 = 11(14-16.3) \bmod 18 = 4$
- any user B can verify the signature by computing
  - $V_1 = 10^{14} \bmod 19 = 16$
  - $V_2 = 4^3.3^4 = 5184 = 16 \bmod 19$

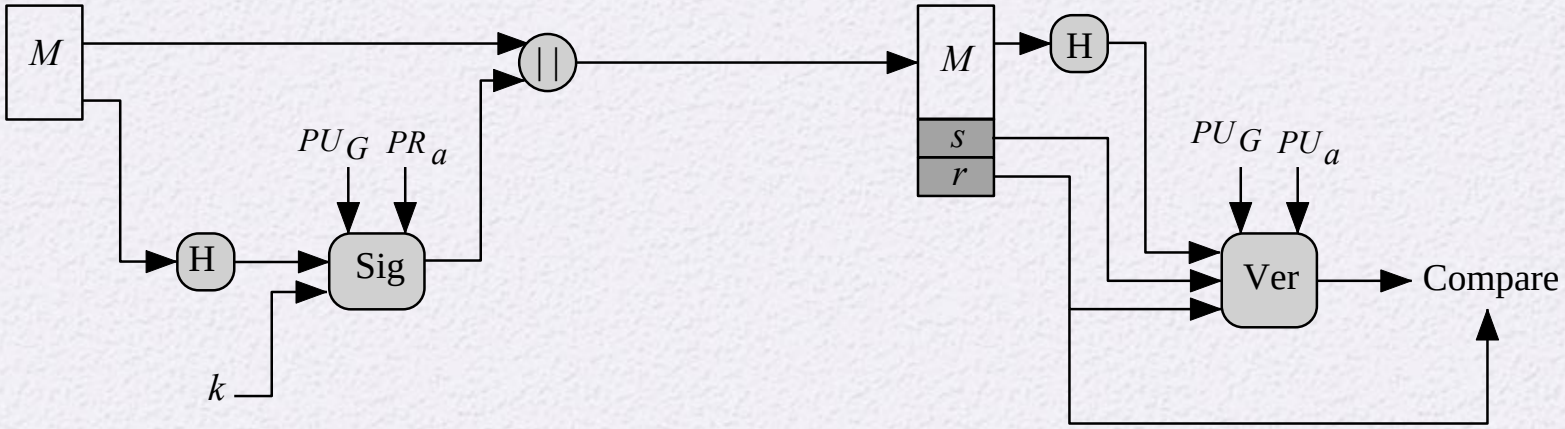
NIST Digital Signature Algorithm

- Published by NIST as Federal Information Processing Standard FIPS 186
- Makes use of the Secure Hash Algorithm (SHA)
- The latest version, FIPS 186-3, also incorporates digital signature algorithms based on RSA and on elliptic curve cryptography

See the next slide for depiction of this Fig.



(a) RSA Approach



(b) DSA Approach

Figure 13.2 Two Approaches to Digital Signatures

Two Approaches to Digital Signatures

Figure 13.2 contrasts the DSA approach for generating digital signatures to that used with RSA.

- In the RSA approach, the message to be signed is input to a hash function that produces a secure hash code of fixed length. This hash code is then encrypted using the sender's private key to form the signature. Both the message and the signature are then transmitted.

Two Approaches to Digital Signatures

- The recipient takes the message and produces a hash code. The recipient also decrypts the signature using the sender's public key. If the calculated hash code matches the decrypted signature, the signature is accepted as valid.
- Because only the sender knows the private key, only the sender could have produced a valid signature.

Two Approaches to Digital Signatures

- The DSA approach also makes use of a hash function.
- The hash code is provided as input to a signature function along with a random number k generated for this particular signature.
- The signature function also depends on the sender's private key (PRa) and a set of parameters known to a group of communicating principals. We can consider this set to constitute a global public key (PUG).
- The result is a signature consisting of two components, labeled s and r .

Two Approaches to Digital Signatures

- At the receiving end, the hash code of the incoming message is generated. This plus the signature is input to a verification function.
- The verification function also depends on the global public key as well as the sender's public key (PUa), which is paired with the sender's private key.
- The output of the verification function is a value that is equal to the signature component r if the signature is valid.
- The signature function is such that only the sender, with knowledge of the private key, could have produced the valid signature.

DSS

Global Public Key Components

p prime number where $2^{L-1} < p < 2^L$ for $512 \le L \le 1024$ and L a multiple of 64 i.e., bit length of between 512 and 1024 bits in increments of 64 bits

q prime divisor of $(p-1)$, where $2^{N-1} < q < 2^N$ i.e., bit length of N bits

g $= h^{(p-1)/q} \bmod p$ where h is any integer with $1 < h < (p-1)$ such that $h^{(p-1)/q} \bmod p > 1$

User's Private Key

x random or pseudorandom integer with $0 < x < q$

User's Public Key

y $= g^x \bmod p$

User's Per-Message Secret Number

k = random or pseudorandom integer with $0 < k < q$

Signing

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

$s = \left[k^{-1}(\mathrm{H}(M) + xr)\right] \bmod q$

Signature = (r, s)

Verifying

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

$u_1 = \left[\mathrm{H}(M')w\right] \bmod q$

$u_2 = (r')w \bmod q$

$v = \left[\left(g^{u_1} y^{u_2}\right) \bmod p\right] \bmod q$

TEST: $v = r'$

M = message to be signed
H(M) = hash of M using SHA-1
M', r', s' = received versions of M, r, s

Figure 13.3 The Digital Signature Algorithm (DSS)

DSA Example

- Global Public-Key Components: p= 283, q=47 and g=60, where g= $h^{(p-1)/q}$ mod p, h is selected 1 < h < (p-1), h=5)
- i.e., g= $5^{(283-1)/47}$ mod 283=60
- Alice computes her key:
  - A chooses x=24

    (x is the private key where 1 < x <47)
  - A computes the public key y=60^{24} mod 283=158

  - Thus, pub. key is(283,47,60,158)
  - priv. key is (283,47,60,24)

DSA Example

- Alice signs the message m with hash $H(m)=41$ as **(19,30):**
  - choosing random $k=15$ where, $0 < k < 47$
  - computing **r** $= (60^{15} \bmod 283) \bmod 47 =$ **19**
  - finding $k^{-1} \bmod q = 15^{-1} \bmod 47 = 22$
  - computing **s**$= 22.(41+24.19) \bmod 47 = 30$
- any user B can verify the signature by computing
  - $w = s^{-1} \bmod q = 30^{-1} \bmod 47 = 11$
  - $u1 = (H(m).w) \bmod q = 41.11 \bmod 47 = 28$
  - $u2 = r.w \bmod q = 19.11 \bmod 47 = 21$
  - **v**$= (g^{u1}. y^{u2} \bmod p) \bmod q$
    $= (60^{28}. 158^{21} \bmod 283) \bmod 47 =$ **19**
  - since **v** $=$ **r**, then the signature is valid