Modern Cryptography
and Its Applications

10 Digital Signatures

Ch13 in textbook

Yanwei Yu

E-mail: ywyu@ustc.edu.cn

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

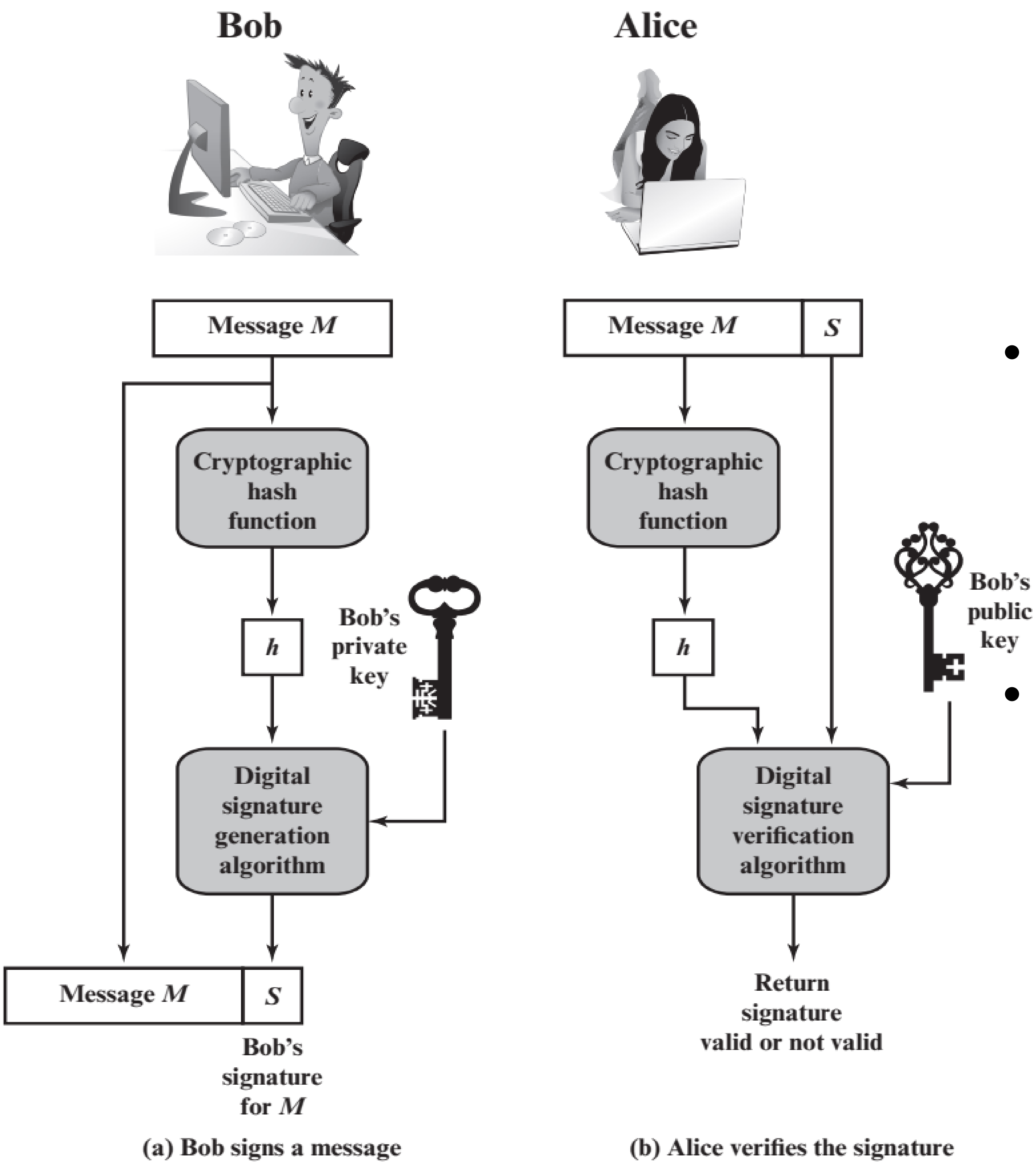


Figure 13.1 Simplified Depiction of Essential Elements of Digital Signature Process

- **no one else could have created a signature that could be verified for this message with Bob's public key.**
- **It is impossible to alter the message without access to Bob's private key,**

Outline

- **Digital Signatures**
- **Elgamal Digital Signature Scheme**
- **Schnorr Digital Signature Scheme**
- **NIST Digital Signature Algorithm**
- **Elliptic Curve Digital Signature Algorithm**
- **RSA-PSS Digital Signature Algorithm**

Outline

- Digital Signatures
- Elgamal Digital Signature Scheme
- Schnorr Digital Signature Scheme
- NIST Digital Signature Algorithm
- Elliptic Curve Digital Signature Algorithm
- RSA-PSS Digital Signature Algorithm

Digital Signatures

- **Consider the following disputes that could arise:**
  - **Mary may forge a different message and claim that it came from John. Mary would simply have to create a message and append an authentication code using the key that John and Mary share.**
  - **John can deny sending the message. Because it is possible for Mary to forge a message, there is no way to prove that John did in fact send the message.**

Digital Signatures

- Message authentication
  - protects two parties who exchange messages from any third party
  - does not protect the two parties against each other: One party may forge a different message and claim that it came from the other party.
  - does not address issues of deny sending the message
- digital signatures provide the ability to:
  - verify author, date & time of signature
  - authenticate message contents
  - be verified by third parties to resolve disputes

Digital Signature Properties

- must depend on the message signed
- must use information unique to sender
  - to prevent both forgery and denial
- must be relatively easy to produce
- must be relatively easy to recognize & verify
- be computationally infeasible to forge
  - forge new message for existing digital signature
  - forge digital signature for given message
- be practical to save digital signature in storage

Direct Digital Signatures

- involve only sender & receiver
- assumed receiver has sender's public-key
- digital signature made by sender signing entire message or hash with private-key
- can verify signature using sender's public-key

- important that sign first then encrypt message & signature
- security depends on sender's private-key

Arbitrated(仲裁) Digital Signatures

- involves use of arbiter A
  - validates any signed message
  - then dated and sent to recipient
- requires suitable level of trust in arbiter
- can be implemented with either private or public-key algorithms
- arbiter may or may not see message

Signature

(1) X ⟶ A: $M||E(K_{xa}, [ID_X||H(M)])$

(2) A ⟶ Y: $E(K_{ay}, [ID_X||M||E(K_{xa}, [ID_X||H(M)])||T])$

(a) Conventional Encryption, Arbiter Sees Message

(1) X ⟶ A: $ID_X||E(K_{xy}, M)||E(K_{xa}, [ID_X||H(E(K_{xy}, M))])$

(2) A ⟶ Y: $E(K_{ay},[ID_X||E(K_{xy}, M)])||E(K_{xa}, [ID_X||H(E(K_{xy}, M))||T])$

(b) Conventional Encryption, Arbiter Does Not See Message

(1) X ⟶ A: $ID_X||E(PR_x, [ID_X||E(PU_y, E(PR_x, M))])$

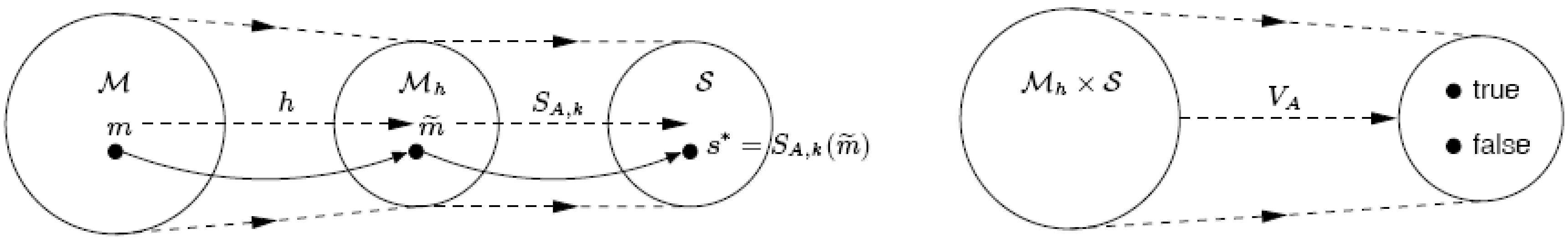
(2) A ⟶ Y: $E(PR_a, [ID_X||E(PU_y, E(PR_x, M))||T])$

(c) Public-Key Encryption, Arbiter Does Not See Message

X = sender
Y = recipient
A = Arbiter
M = message
T = timestamp

Optional: Classification of Digital Signatures

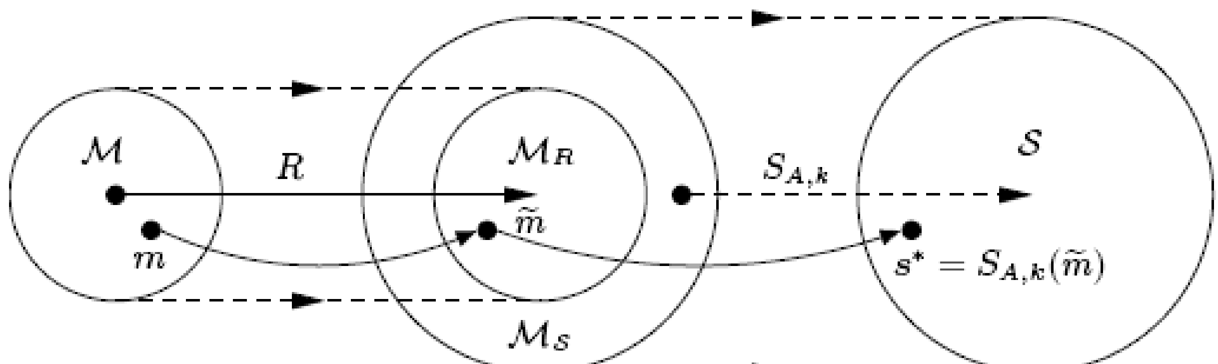
- Digital signature with appendix



(a) The signing process

(b) The verification process

- Digital signature with message recovery



- **Four main process:**
  - **Select global public domain parameters.**
  - **Generate public & private key pair**
  - **Sign the message**
  - **Verify the signature**

Outline

- Digital Signatures
- Elgamal Digital Signature Scheme
- Schnorr Digital Signature Scheme
- NIST Digital Signature Algorithm
- Elliptic Curve Digital Signature Algorithm
- RSA-PSS Digital Signature Algorithm

Select global public domain parameters.

Before proceeding, we need a result from number theory. Recall from Chapter 2 that for a prime number q, if α is a primitive root of q, then

$$\alpha, \alpha^2, \ldots, \alpha^{q-1}$$

are distinct (mod q). It can be shown that, if α is a primitive root of q, then

1. For any integer m, $\alpha^m \equiv 1 \pmod{q}$ if and only if $m \equiv 0 \pmod{q-1}$.
2. For any integers, i, j, $\alpha^i \equiv \alpha^j \pmod{q}$ if and only if $i \equiv j \pmod{q-1}$.

Key Generation

1. Generate a random integer X_A, such that $1 < X_A < q - 1$.
2. Compute $Y_A = \alpha^{X_A} \bmod q$.
3. A's private key is X_A; A's pubic key is $\{q, \alpha, Y_A\}$.

Sign the message

1. Choose a random integer K such that $1 \le K \le q - 1$ and $\gcd(K, q - 1) = 1$. That is, K is relatively prime to $q - 1$.
2. Compute $S_1 = \alpha^K \bmod q$. Note that this is the same as the computation of C_1 for Elgamal encryption.
3. Compute $K^{-1} \bmod (q - 1)$. That is, compute the inverse of K modulo $q - 1$.
4. Compute $S_2 = K^{-1}(m - X_A S_1) \bmod (q - 1)$.
5. The signature consists of the pair (S_1, S_2).

Any user B can verify the signature as follows.

1. Compute $V_1 = \alpha^m \bmod q$.
2. Compute $V_2 = (Y_A)^{S_1}(S_1)^{S_2} \bmod q$.

Example

For example, let us start with the prime field GF(19); that is, $q = 19$. It has primitive roots {2, 3, 10, 13, 14, 15}, as shown in Table 2.7. We choose $\alpha = 10$.

Alice generates a key pair as follows:

1. Alice chooses $X_A = 16$.
2. Then $Y_A = \alpha^{X_A} \bmod q = \alpha^{16} \bmod 19 = 4$.
3. Alice's private key is 16; Alice's pubic key is $\{q, \alpha, Y_A\} = \{19, 10, 4\}$.

Suppose Alice wants to sign a message with hash value $m = 14$.

1. Alice chooses $K = 5$, which is relatively prime to $q - 1 = 18$.
2. $S_1 = \alpha^K \bmod q = 10^5 \bmod 19 = 3$ (see Table 2.7).
3. $K^{-1} \bmod (q - 1) = 5^{-1} \bmod 18 = 11$.
4. $S_2 = K^{-1}(m - X_A S_1) \bmod (q - 1) = 11(14 - (16)(3)) \bmod 18 = -374 \bmod 18 = 4$.

Bob can verify the signature as follows.

1. $V_1 = \alpha^m \bmod q = 10^{14} \bmod 19 = 16$.
2. $V_2 = (Y_A)^{S_1}(S_1)^{S_2} \bmod q = (4^3)(3^4) \bmod 19 = 5184 \bmod 19 = 16$.

Thus, the signature is valid because $V_1 = V_2$.

- **why verification is feasible?**

The signature is valid if $V_1 = V_2$. Let us demonstrate that this is so. Assume that the equality is true. Then we have

$\alpha^m \bmod q = (Y_A)^{S_1}(S_1)^{S_2} \bmod q$	assume $V_1 = V_2$
$\alpha^m \bmod q = \alpha^{X_A S_1}\alpha^{KS_2} \bmod q$	substituting for Y_A and S_1
$\alpha^{m - X_A S_1} \bmod q = \alpha^{KS_2} \bmod q$	rearranging terms
$m - X_A S_1 \equiv KS_2 \bmod (q-1)$	property of primitive roots
$m - X_A S_1 \equiv KK^{-1}(m - X_A S_1) \bmod (q-1)$	substituting for S_2

Outline

- Digital Signatures
- Elgamal Digital Signature Scheme
- Schnorr Digital Signature Scheme
- NIST Digital Signature Algorithm
- Elliptic Curve Digital Signature Algorithm
- RSA-PSS Digital Signature Algorithm

Review: The Powers of an Integer, Modulo prime P

a	a^2	a^3	a^4	a^5	a^6	a^7	a^8	a^9	a^{10}	a^{11}	a^{12}	a^{13}	a^{14}	a^{15}	a^{16}	a^{17}	a^{18}
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	4	8	16	13	7	14	9	18	17	15	11	3	6	12	5	10	1
3	9	8	5	15	7	2	6	18	16	10	11	14	4	12	17	13	1
4	16	7	9	17	11	6	5	1	4	16	7	9	17	11	6	5	1
5	6	11	17	9	7	16	4	1	5	6	11	17	9	7	16	4	1
6	17	7	4	5	11	9	16	1	6	17	7	4	5	11	9	16	1
7	11	1	7	11	1	7	11	1	7	11	1	7	11	1	7	11	1
8	7	18	11	12	1	8	7	18	11	12	1	8	7	18	11	12	1
9	5	7	6	16	11	4	17	1	9	5	7	6	16	11	4	17	1
10	5	12	6	3	11	15	17	18	9	14	7	13	16	8	4	2	1
11	7	1	11	7	1	11	7	1	11	7	1	11	7	1	11	7	1
12	11	18	7	8	1	12	11	18	7	8	1	12	11	18	7	8	1
13	17	12	4	14	11	10	16	18	6	2	7	15	5	8	9	3	1
14	6	8	17	10	7	3	4	18	5	13	11	2	9	12	16	15	1
15	16	12	9	2	11	13	5	18	4	3	7	10	17	8	6	14	1
16	9	11	5	4	7	17	6	1	16	9	11	5	4	7	17	6	1
17	4	11	16	6	7	5	9	1	17	4	11	16	6	7	5	9	1
18	1	18	1	18	1	18	1	18	1	18	1	18	1	18	1	18	1

Table 8.3 Powers of Integers, Modulo 19

Key Generation

1. Choose primes p and q, such that q is a prime factor of $p - 1$.
2. Choose an integer a, such that $a^q = 1 \bmod p$. The values a, p, and q comprise a global public key that can be common to a group of users.
3. Choose a random integer s with $0 < s < q$. This is the user's private key.
4. Calculate $v = a^{-s} \bmod p$. This is the user's public key.

Sign the message

1. Choose a random integer r with $0 < r < q$ and compute $x = a^r \bmod p$. This computation is a preprocessing stage independent of the message M to be signed.
2. Concatenate the message with x and hash the result to compute the value e:

$$e = \mathrm{H}(M \| x)$$

3. Compute $y = (r + se) \bmod q$. The signature consists of the pair (e, y).

Any other user can verify the signature as follows.

1. Compute $x' = a^y v^e \bmod p$.
2. Verify that $e = \mathrm{H}(M \| x')$.

- **why verification is feasible?**

To see that the verification works, observe that

$$x' \equiv a^y v^e \equiv a^y a^{-se} \equiv a^{y-se} \equiv a^r \equiv x \pmod{p}$$

Hence, $\mathrm{H}(M \| x') = \mathrm{H}(M \| x)$.

Outline

- Digital Signatures
- Elgamal Digital Signature Scheme
- Schnorr Digital Signature Scheme
- NIST Digital Signature Algorithm
- Elliptic Curve Digital Signature Algorithm
- RSA-PSS Digital Signature Algorithm

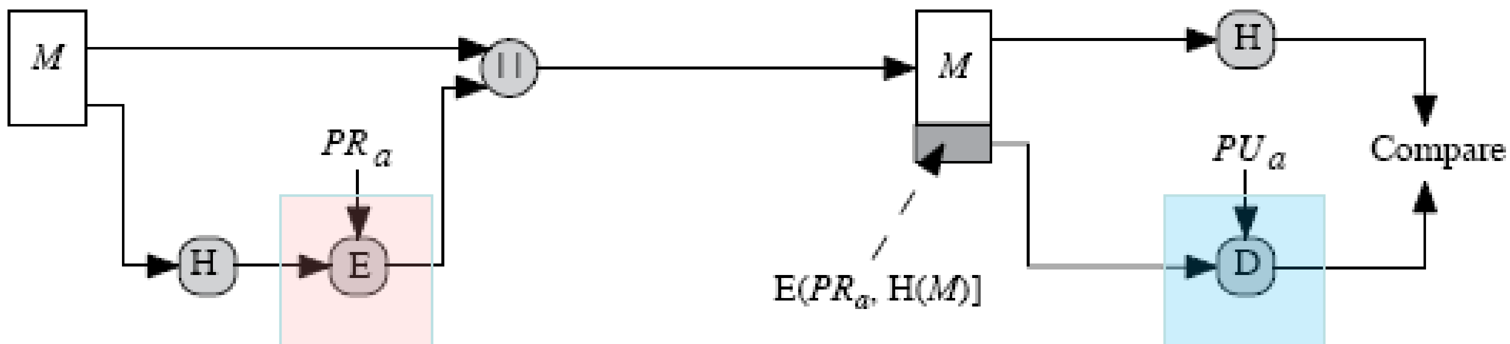
Digital Signature Standard (DSS)

- US Govt approved signature scheme
- designed by NIST & NSA in early 90's
- published as FIPS-186 in 1991
- revised in 1993, 1996 & then 2000
- uses the SHA hash algorithm
- DSS is the standard, DSA is the algorithm
- FIPS 186-2 (2000) includes alternative RSA & elliptic curve signature variants

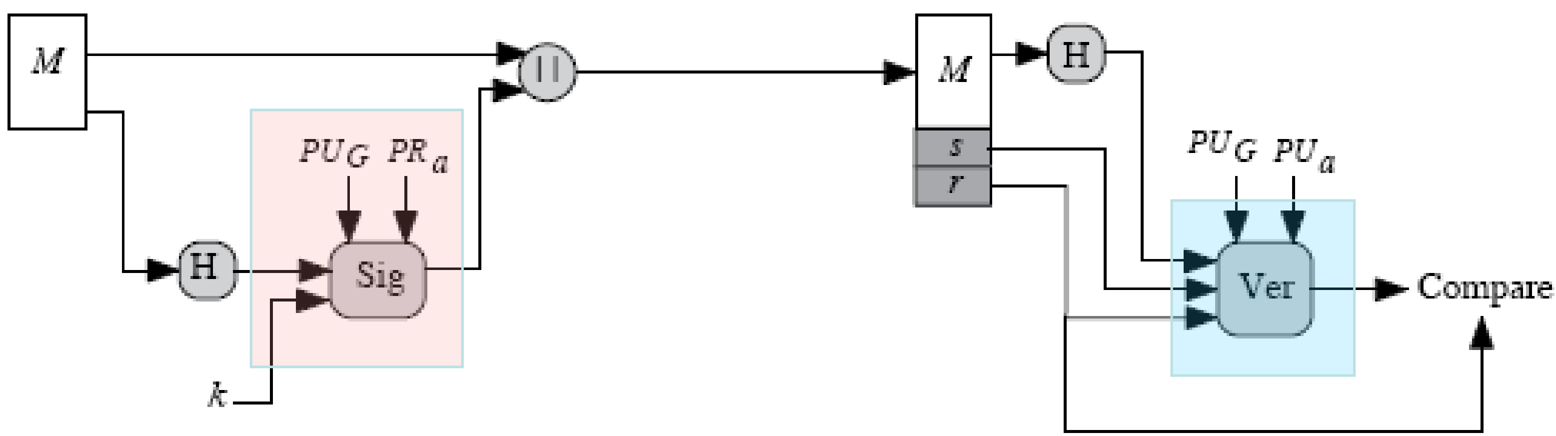
Digital Signature Algorithm (DSA)

- creates a 320 bit signature
- with 512-1024 bit security
- smaller and faster than RSA
- a digital signature scheme only
- security depends on difficulty of computing discrete logarithms
- variant of ElGamal & Schnorr schemes

DSS vs. RSA Signature Scheme



(a) RSA Approach



(b) DSS Approach

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 L 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 $= h(\mathrm{p} - 1)/\mathrm{q}$ is an exponent mod 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

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

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

Signature = (r, s)

Verifying

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

$u_1 = [H(M')w] \bmod q$

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

$v = [(g^{u1}y^{u2}) \bmod p] \bmod q$

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 (DSA)

DSA Key Generation

- have shared global public domain values (p,q,g):
  - choose q, a 160-bit prime
  - choose a large prime $p = 2^L$
    - where L= 512 to 1024 bits and is a multiple of 64
    - and q is a prime factor of $(p-1)$
  - choose $g = h^{(p-1)/q}$
    - where $h<p-1$, $h^{(p-1)/q} \pmod{p} > 1$
- users choose private & compute public key:
  - choose $x<q$ (private key)
  - compute $y = g^x \pmod{p}$ (public key)

DSA Signature Creation

- to sign a message `M` the sender:
  - generates a random signature key `k`, `k<q`
  - nb. `k` must be random, be destroyed after use, and never be reused
- then computes signature pair:

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

$$s = (k^{-1}.H(M) + x.r) \pmod q$$

- sends signature `(r,s)` with message `M`

DSA Signature Verification

- having received M & signature `(r,s)`
- to verify a signature, recipient computes:

$$w = s^{-1} \pmod q$$

$$u1 = (H(M).w) \pmod q$$

$$u2 = (r.w) \pmod q$$

$$v = (g^{u1}.y^{u2} \pmod p) \pmod q$$

- if `v=r` then signature is verified

- **why verification is feasible?**

$$s \equiv k^{-1}(\mathrm{H}(m) + x \cdot r) \mod q$$

$$\Rightarrow ks = (\mathrm{H}(m) + x \cdot r) \mod q$$

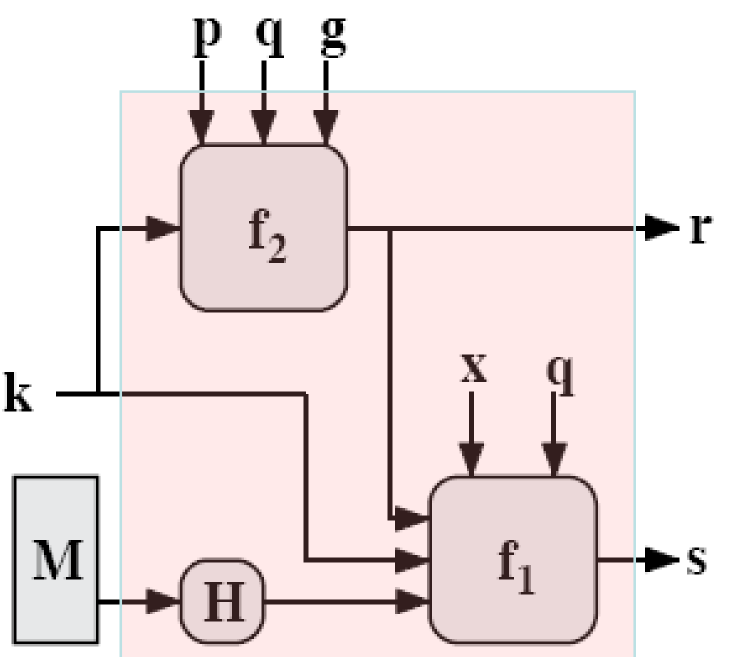
$$\Rightarrow \mathrm{k} = s^{-1}(\mathrm{H}(m) + x \cdot r) \mod q$$

$$\Rightarrow v = [(g^{\mathrm{u1}}.\mathrm{y}^{\mathrm{u2}}) \text{ mod p] mod q}$$

$$= [(g^{\mathrm{H(M)/s}}.(g^{x})^{\mathrm{r/s}}) \text{ mod p] mod q}$$

$$= [(g^{\mathrm{H(M)/s}}.(g^{x})^{\mathrm{r/s}}) \text{ mod p] mod q}$$

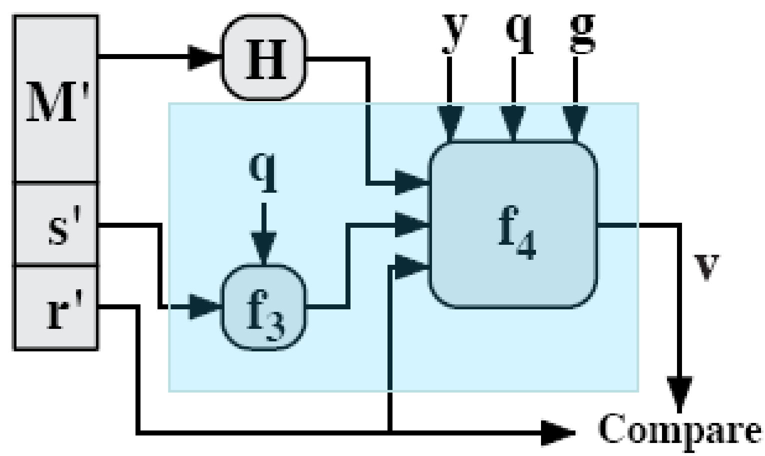
$$= (g^{\mathrm{k}} \text{ mod p) mod q = r}$$



$$s = f_1(H(M), k, \mathbf{x}, r, q) = (k^{-1}\,(H(M) + \mathbf{x}r)) \bmod q$$

$$r = f_2(k, p, q, g) = (g^k \bmod p) \bmod q$$

(a) Signing



$$w = f_3(s', q) = (s')^{-1} \bmod q$$

$$v = f_4(y, q, g, H(M'), w, r')$$

$$= ((g^{(H(M')w) \bmod q}\, y^{r'w \bmod q}) \bmod p) \bmod q$$

(b) Verifying

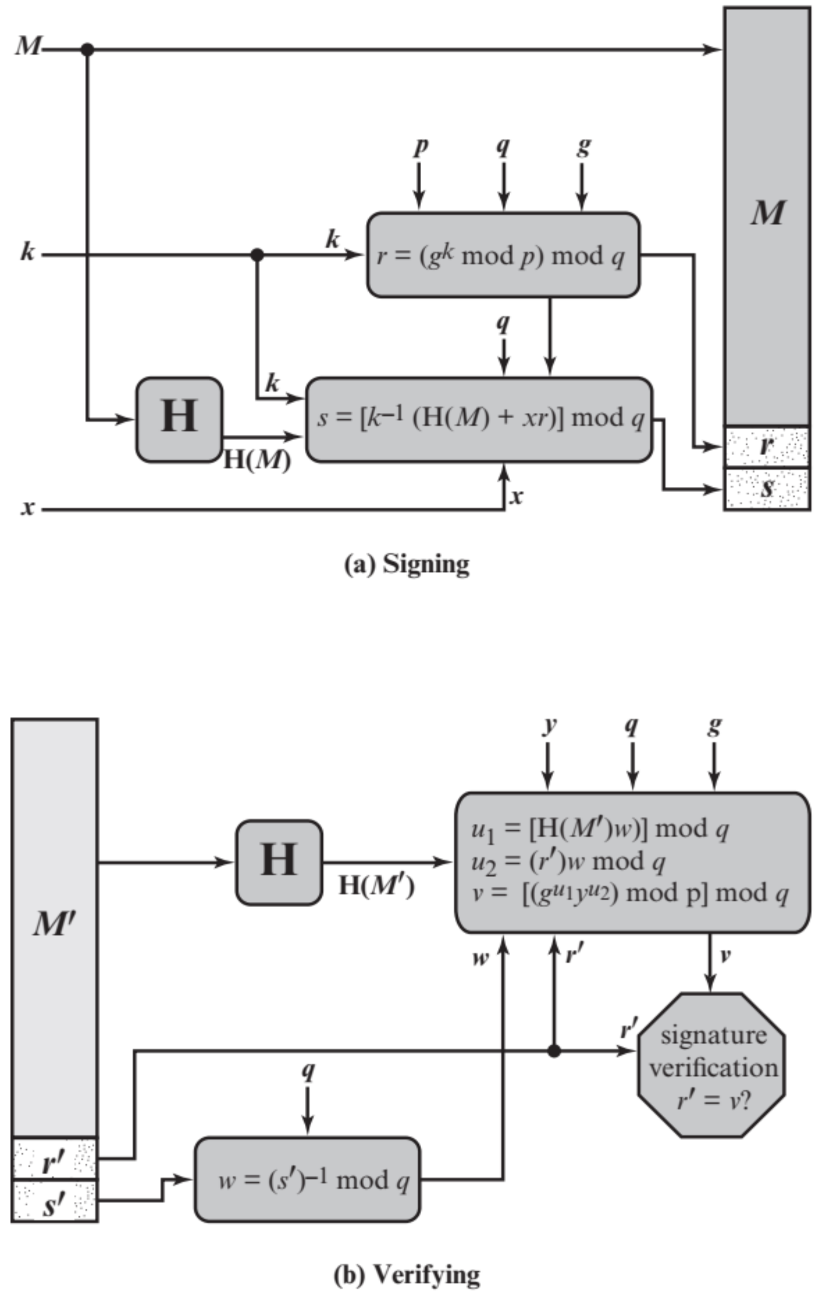


Figure 13.4 DSA Signing and Verifying

Outline

- Digital Signatures
- Elgamal Digital Signature Scheme
- Schnorr Digital Signature Scheme
- NIST Digital Signature Algorithm
- Elliptic Curve Digital Signature Algorithm
- RSA-PSS Digital Signature Algorithm

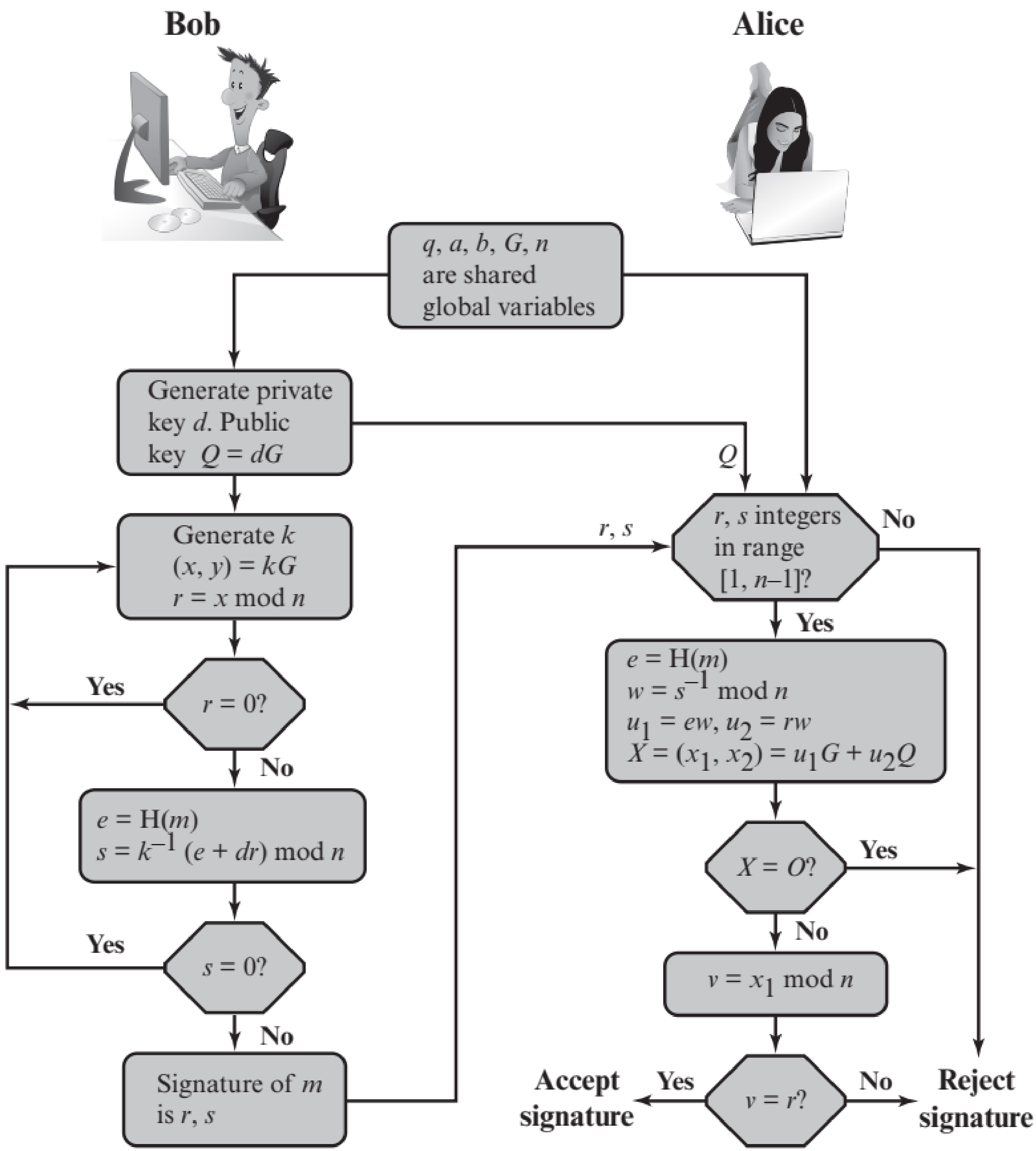


Figure 13.5 ECDSA Signing and Verifying

Outline

- Digital Signatures
- Elgamal Digital Signature Scheme
- Schnorr Digital Signature Scheme
- NIST Digital Signature Algorithm
- Elliptic Curve Digital Signature Algorithm
- RSA-PSS Digital Signature Algorithm

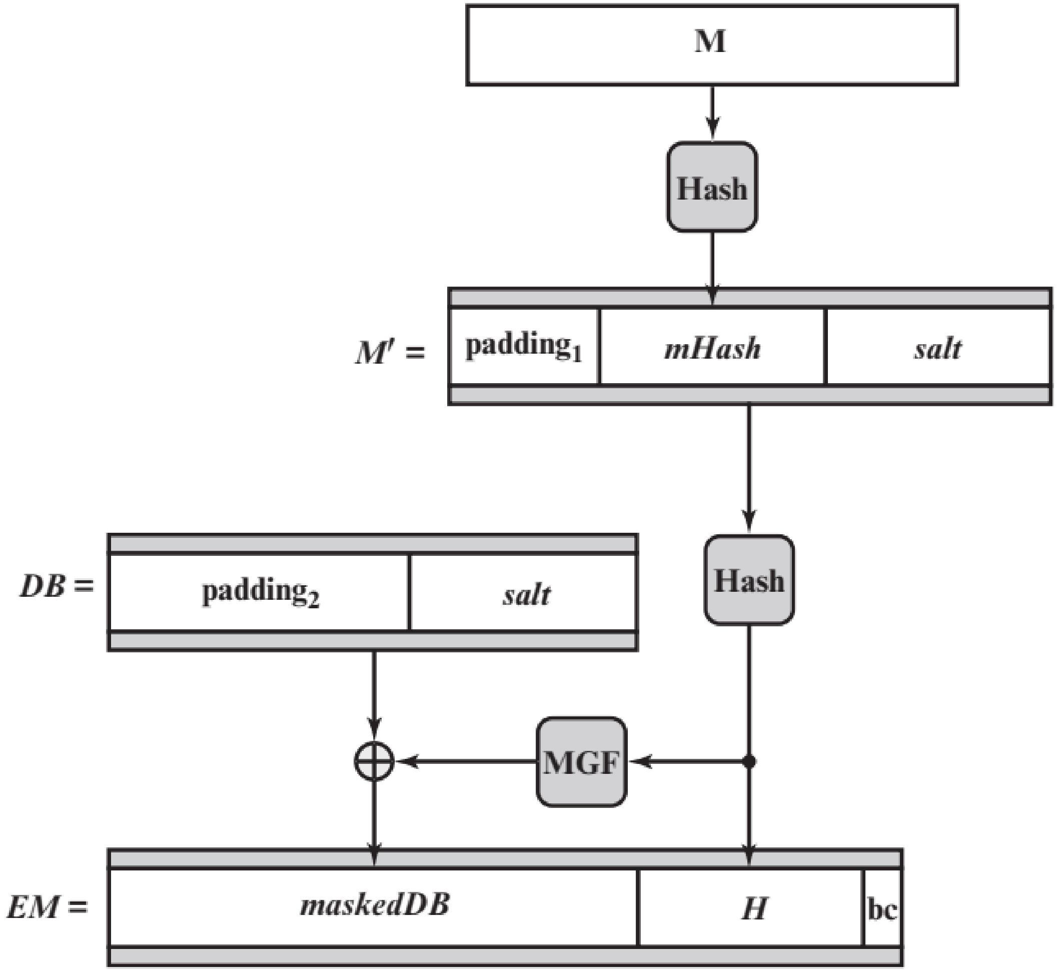


Figure 13.6 RSA-PSS Encoding

$padding_2$	hexadecimal string of 00 octets with a length ($emLen - sLen - hLen - 2$) octets, followed by the hexadecimal octet with value 01.
salt	a pseudorandom number.
bc	the hexadecimal value BC.

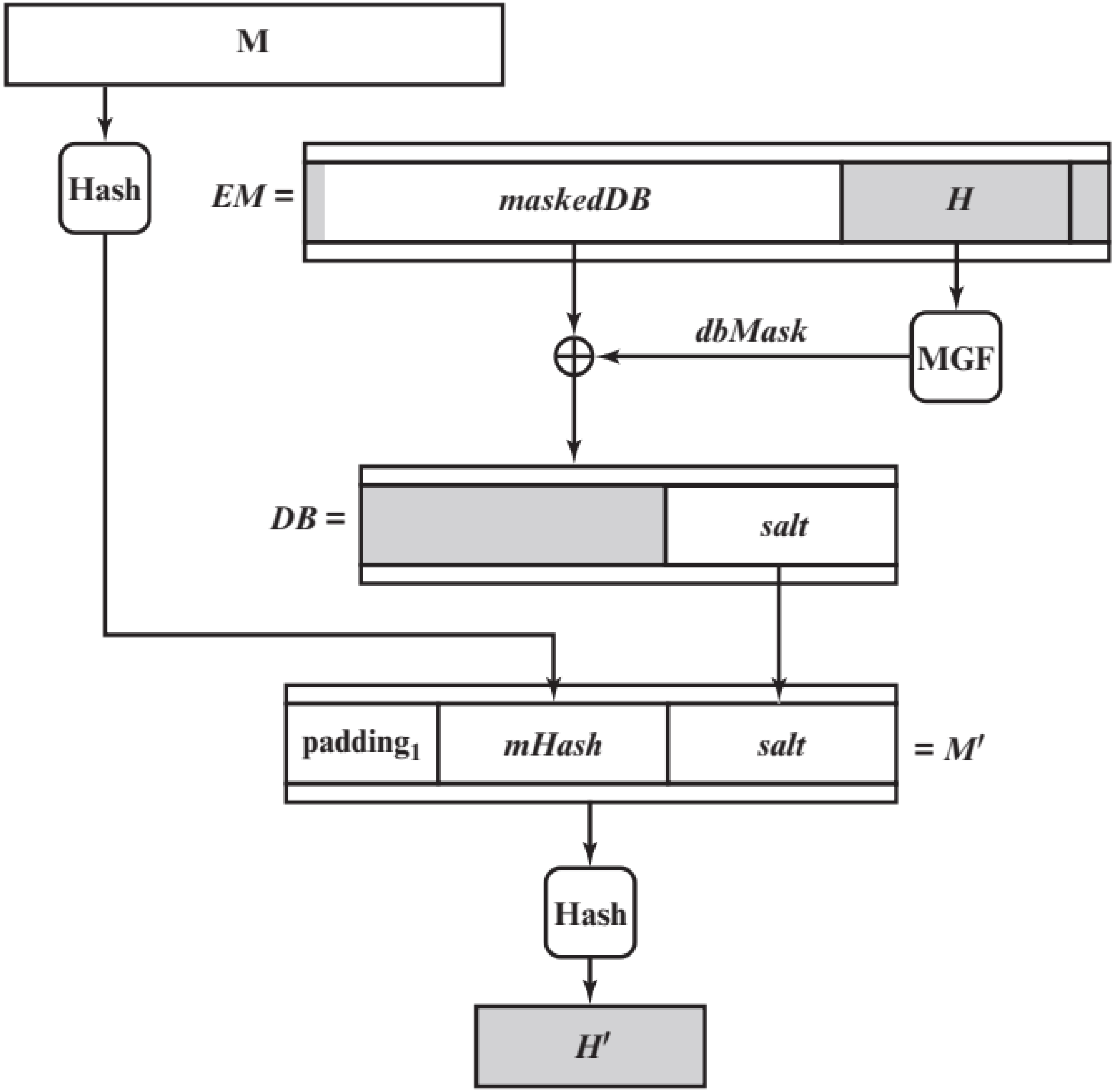


Figure 13.7 RSA-PSS EM Verification

Summary

- have discussed:
  - digital signatures: definition, objective, properties, direct vs. Arbitrated Digital Signatures, classification
  - RSA related signature standard
    - ISO/IEC 9796 formatting
    - PKCS #1 formatting
  - Digital Signature Standard (DSS)

Key Terms

digital signature Digital Signature Algorithm (DSA) direct digital signature	Elgamal digital signature Elliptic Curve Digital Signature Algorithm (ECDSA)	Schnorr digital signature timestamp

Review Questions

- **13.1 List two disputes that can arise in the context of message authentication.**
- **13.2 What are the properties a digital signature should have?**
- **13.5 In what order should the signature function and the confidentiality function be applied to a message, and why?**
- **13.6 What are some threats associated with a direct digital signature scheme?**

Thanks!

- **PKCS #1**
  - **involves RSA encryption and signature scheme**
  - **use a digital signature scheme with appendix.**
  - **Involves three procedure**
    - **(i) data formatting**
    - **(ii) Signature process**
    - **(iii) Verification process**