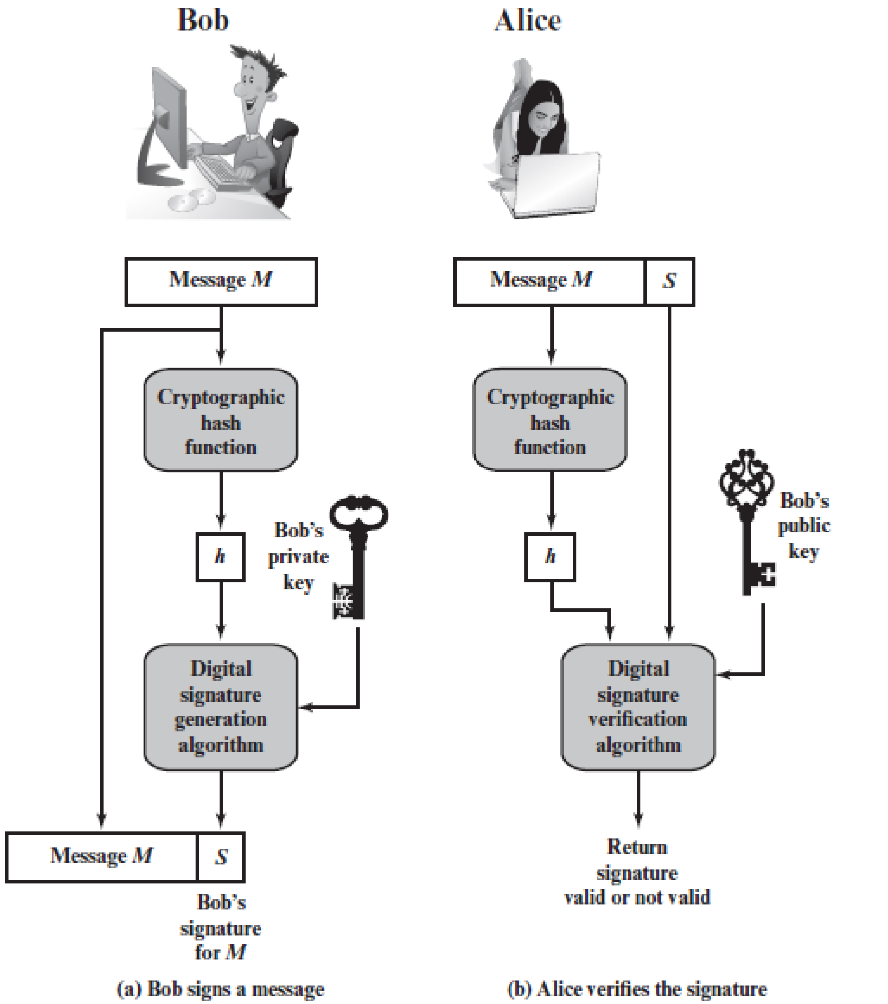
Data Security and Privacy
DSE 3258

L13 –Digital Signature

Digital Signature- Introduction


Bob
Alice
Message M
Cryptographic hash function
h
Bob's private key
Digital signature generation algorithm
Message M
S
Bob's signature for M
Message M
S
Cryptographic hash function
h
Bob's public key
Digital signature verification algorithm
Return signature valid or not valid

(a) Bob signs a message

(b) Alice verifies the signature

Figure 13.1 Simplified Depiction of Essential Elements of Digital Signature Process

Properties

- Message authentication protects two parties who exchange messages from any third party. However, it does not protect the two parties against each other. Several forms of dispute between the two parties are possible.

- For example, suppose that John sends an authenticated message to Mary. Consider the following disputes that could arise.
- **1.** 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.
- **2.** 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.

- In situations where there is not complete trust between sender and receiver, something more than authentication is needed.
- 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

DSA – attacks and forgeries

Note:
A – user
C - attacker

- **Types of attacks.**
- **Key-only attack:** C only knows A's public key.
- **Known message attack:** C is given access to a set of messages and their signatures.
- **Generic chosen message attack:** C chooses a list of messages before attempting to breaks A's signature scheme, independent of A's public key. C then obtains from A valid signatures for the chosen messages. The attack is generic, because it does not depend on A's public key; the same attack is used against everyone.
- **Directed chosen message attack:** Similar to the generic attack, except that the list of messages to be signed is chosen after C knows A's public key but before any signatures are seen.
- **Adaptive chosen message attack:** C is allowed to use A as an "oracle." This means that C may request from A signatures of messages that depend on previously obtained message-signature pairs.

DSA – attacks and forgeries

- **Successful attacks:**

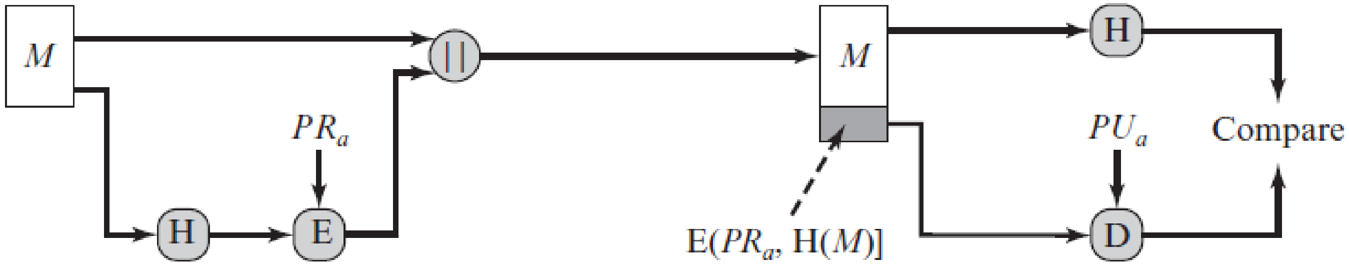
Note:
A – user
C - attacker

- **Total break:** C determines A’s private key.
- **Universal forgery:** C finds an efficient signing algorithm that provides an equivalent way of constructing signatures on arbitrary messages.
- **Selective forgery:** C forges a signature for a particular message chosen by C.
- **Existential forgery:** C forges a signature for at least one message. C has no control over the message. Consequently, this forgery may only be a minor nuisance to A.

Digital Signature Requirements

- The signature must be a bit pattern that depends on the message being signed.
- The signature must use some information only known 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.

Digital Signature using RSA

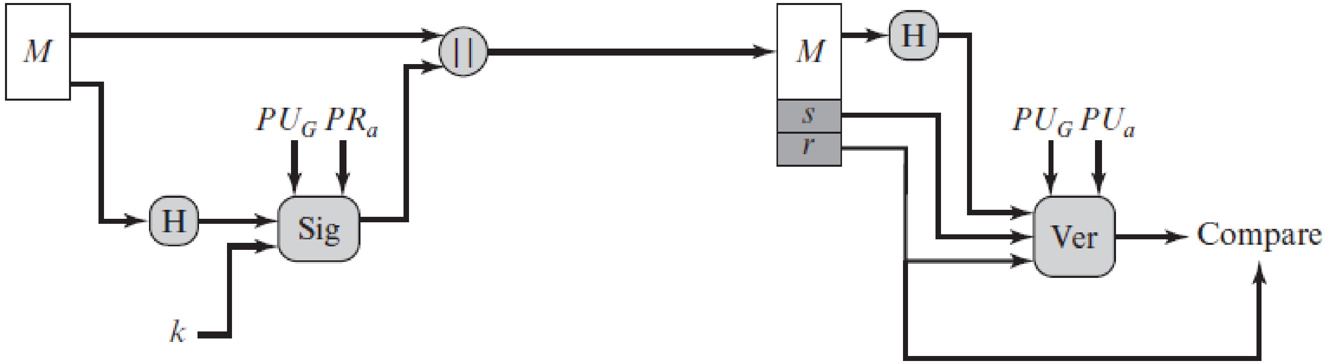


(a) RSA approach

- 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.
- 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.

NIST Digital Signature Algorithm

- The National Institute of Standards and Technology (NIST) has published Federal Information Processing Standard FIPS 186, known as the *Digital Signature Algorithm (DSA).*
- For hashing - Secure Hash Algorithm (SHA)
- DSA is used only for authentication not for confidentiality i.e. only signatures can be generated not to encrypt the message.



(b) DSA approach

NIST DSA (contd..)

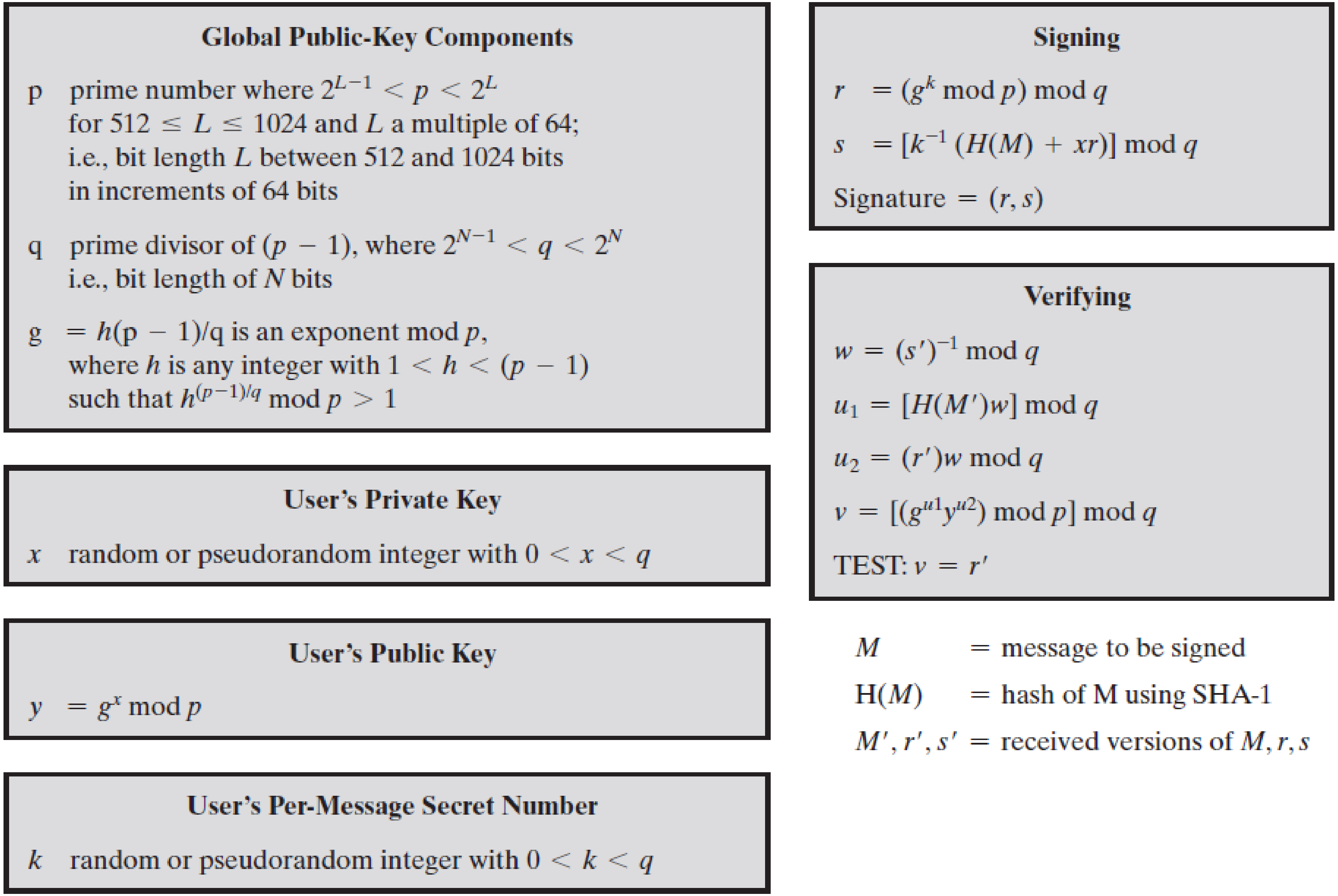
At sender end

- 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 (PR_a) and a set of parameters known to a group of communicating principals.
- We can consider this set to constitute a global public key (PU_G).
- The result is a signature consisting of two components, labeled *s* and *r*.

NIST DSA (contd..)

At receiver end

- The hash code of the incoming message is generated. The hash code and the signature are inputs 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.



Global Public-Key Components

p prime number where $2^{L-1} < p < 2^{L}$
for $512 \leq L \leq 1024$ 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)