5. DIGITAL SIGNATURE ALGORITHM

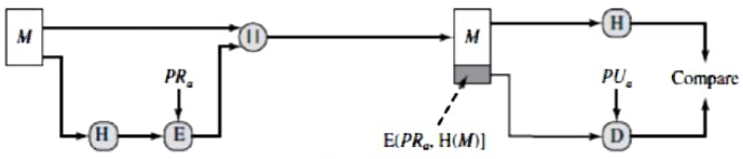
- **Explain Digital Signature Standard. (May/June'14)**
- **Give the details of digital signature algorithm. (May/June'07)**
- **With a neat sketch, explain signing and verifying functions of DSA. (May/June'12)**

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).
- The DSA makes use of the Secure Hash Algorithm (SHA).
- The DSA was originally proposed in 1991 and revised in 1993, 1996 and then 2000 an expanded version of the standard was issued as FIPS 186-2, subsequently updated to FIPS 186-3 in 2009.
- The DSA uses an algorithm that is designed to provide only the digital signature function. Unlike RSA, it cannot be used for encryption or key exchange. Nevertheless, it is a public-key technique.

The 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.

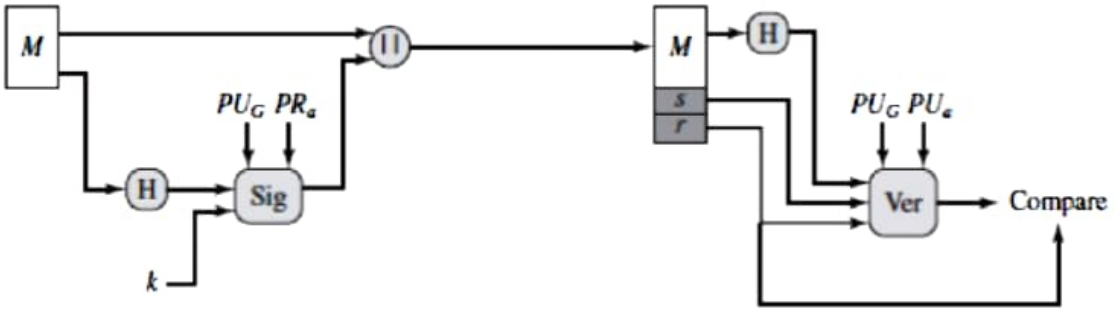


(a) RSA approach

The DSA Approach:

- 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 (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.
- 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 (PU_a), 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.



(b) DSA approach

The Digital Signature Algorithm

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

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

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

Signature $= (r, s)$

Verifying

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

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

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

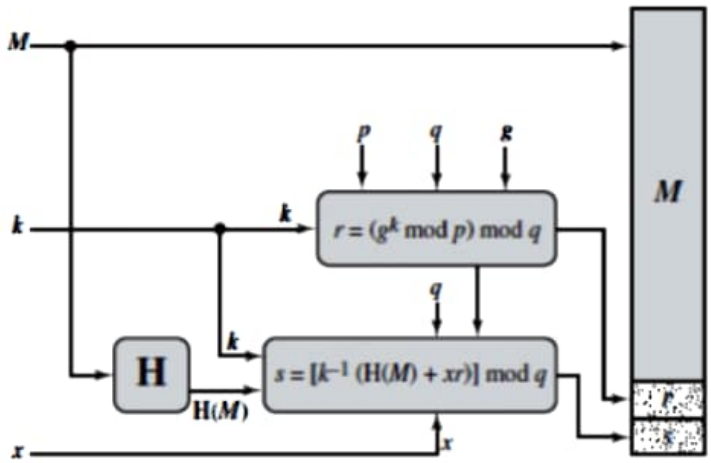
$v = [(g^{u1} y^{u2}) \bmod p] \bmod q$

TEST: $v = r'$

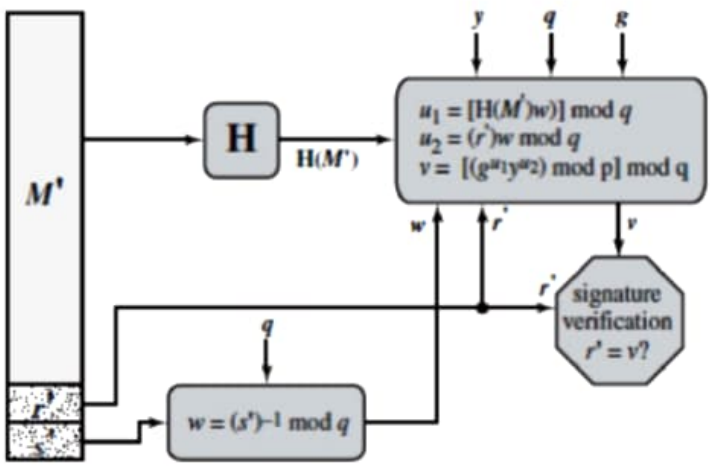
M = message to be signed

$H(M)$ = hash of M using SHA-1

M', r', s' = received versions of M, r, s



(a) Signing



(b) Verifying

Figure: DSA Signing and Verifying