Assignment No.

Title: DSA signature

Aim: Write a program to produce a DSA signature using parameter tuple <p, q, g>, long term key pair and a message digest.

Objective:

- Understand the concept of DSA Signature and message digest.

Theory:

DSA is a United States Federal Government standard for digital signatures. It was proposed by the National Institute of Standards and Technology (NIST) in August 1991 for use in their Digital Signature Standard (DSS), specified in FIPS 186 in 1993. The first part of the DSA algorithm is the public key and private key generation, which can be described as:

Choose a prime number q, which is called the prime divisor.

- Choose another primer number p, such that p-1 mod q = 0. p is called the prime modulus.
- Choose an integer g, such that 1 < g < p, g**q mod p = 1 and g = h**((p–1)/q) mod p. q is also called g's multiplicative order modulo p.
- Choose an integer, such that 0 < x < q.
- Compute y as g**x mod p.
- Package the public key as {p,q,g,y}.
- Package the private key as {p,q,g,x}.

The second part of the DSA algorithm is the signature generation and signature verification, which can be described as:

To generate a message signature, the sender can follow these steps:

- Generate the message digest h, using a hash algorithm like SHA1.
- Generate a random number k, such that 0 < k < q.
- Compute r as (g**k mod p) mod q. If r = 0, select a different k.
- Compute i, such that k*i mod q = 1. i is called the modular multiplicative inverse of k modulo q.
- Compute s = i*(h+r*x) mod q. If s = 0, select a different k.

- Package the digital signature as {r,s}.

To verify a message signature, the receiver of the message and the digital signature can follow these steps:

- Generate the message digest h, using the same hash algorithm.
- Compute w, such that s*w mod q = 1. w is called the modular multiplicative inverse of s modulo q.
- Compute u1 = h*w mod q.
- Compute u2 = r*w mod q.
- Compute v = (((g**u1)*(y**u2)) mod p) mod q.
- If v == r, the digital signature is valid.

A digital signature is represented in a computer as a string of binary digits. The signature is computer using a set of rules and parameters (algorithm) such that the identity of the person signing the document as well as the originality of the data can be verified. The signature is generated by the use of a private key. A private key is known only to the user. The signature is verified makes use of a public key which corresponds to (but not the same, i.e. mathematically infeasible to deduct private key from public) the private key. With every user having a public/private key pair, this is an example of public-key cryptography. Public keys, which are known by everyone, can be used to verify the signature of a user. The private key, which is never shared, is used in signature generation, which can only be done by the user.

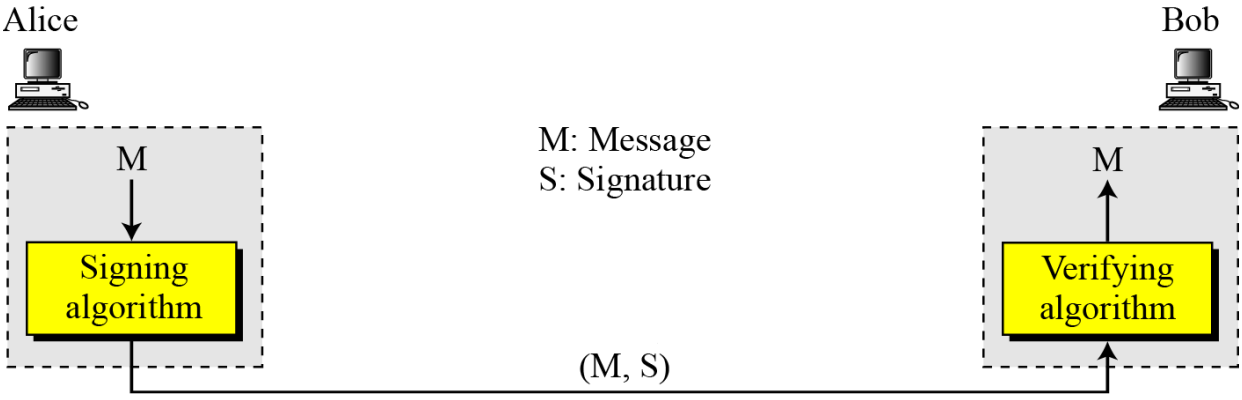

Fig: Digital signature Process

The sender uses a signing algorithm to sign the message. The message and the signature are sent to the receiver. The receiver receives the message and the signature and applies the verifying

algorithm to the combination. If the result is true, the message is accepted; otherwise, it is rejected.

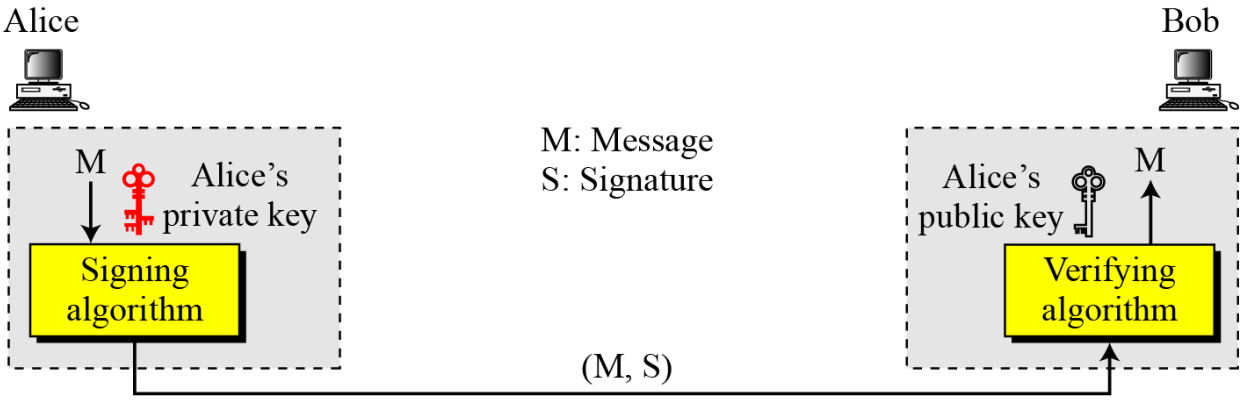

Fig: Adding key to the digital signature process

A digital signature needs a public-key system. The signer signs with her private key; the verifier verifies with the signer's public key.

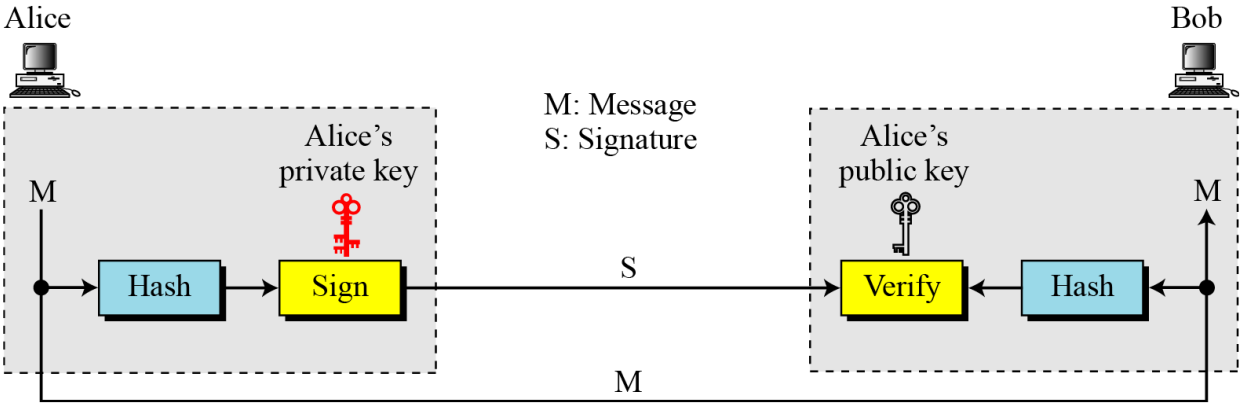

Fig: Signing the digest

A digital signature can directly provide the three services for message confidentiality

1. Message Integrity: The integrity of the message is preserved even if we sign the whole message because we cannot get the same signature if the message is changed.

2. Nonrepudiation: - Nonrepudiation can be provided using a trusted party.

3. Confidentiality:- A digital signature does not provide privacy. If there is a need for privacy, another layer of encryption/decryption must be applied.

Conclusion: Thus we have studied and implemented DSA Signature.