DIGITAL SIGNATURE:-

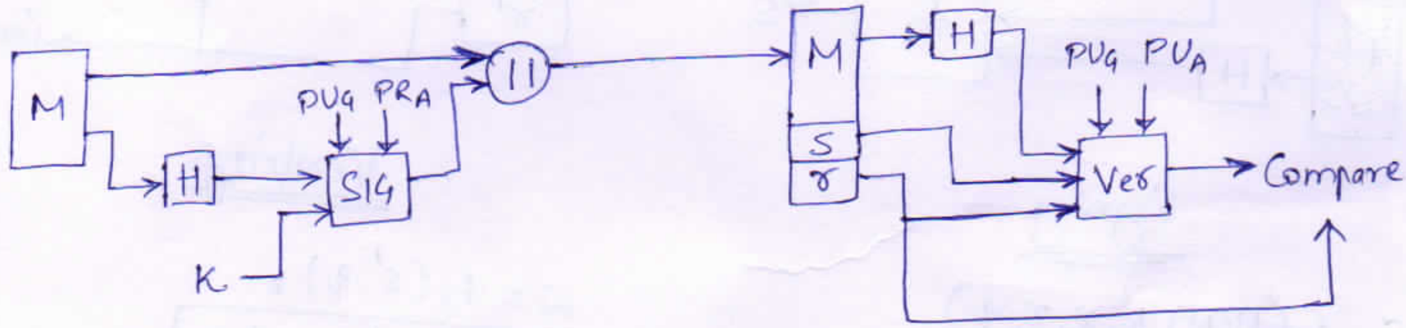
→ It is an authentication mechanism that enables the creator of a message to attach code that acts as a signature. The signature is formed by taking the hash of the message and encrypting the message with the creator's ~~public~~ private key.

→ The signature guarantees the source and integrity of the message.

~~DSS APR~~

DSS APPROACH:- (DIGITAL SIGNATURE STANDARD)

DSS uses an algorithm that is designed to provide only the digital signature function.



Algorithm :-

① Global Public key Components

p prime number, with a length between 512 and 1024 bits such that q divides $(p-1)$.

q A 160-bit prime number q.

g selected to be of the form $h^{(p-1)/q} \bmod p$, where h is an integer between 1 and $(p-1)$

② User's Private key

x random integer with $0 < x < q$

③ User's Public key

$y = g^x \bmod p$

④ User's Per-message Secret Number

K random integer with $0 < K < q$

⑤ Signing

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

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

Signature $= (r, s)$

⑥ Verifying

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

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

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

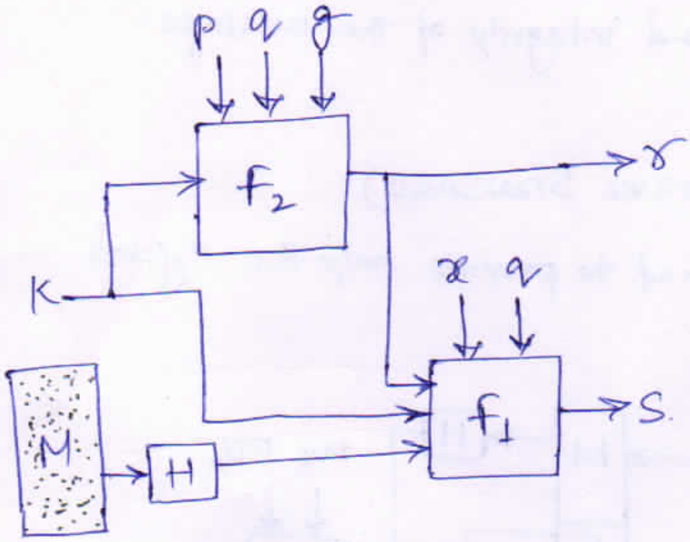
$v = [(g^{u_1} y^{u_2}) \bmod p] \bmod q$

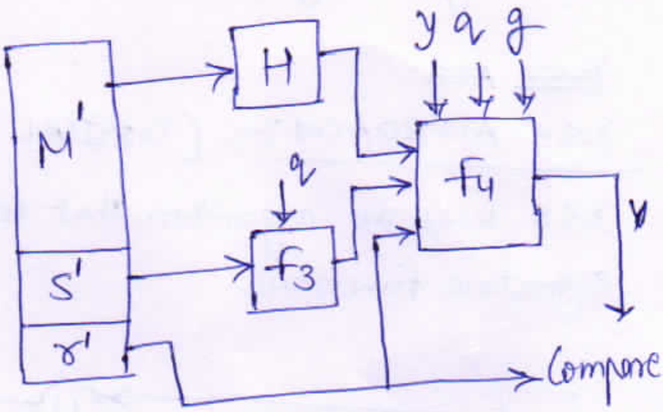
Test : $v = r'$

M = message to be signed.

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

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





Signing

Verifying

$$s = f_1(H(M), k, x, r, q)$$

$$\boxed{s = (k^{-1}(H(M) + x \cdot r)) \bmod q}$$

$$r = f_2(k, p, q, g)$$

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

$$w = f_3(s', q)$$

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

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

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

ZERO KNOWLEDGE PROTOCOL:-

I CAN'T TELL YOU MY SECRET, BUT I CAN PROVE TO YOU THAT I KNOW THE SECRET

→ It is a method by which one party (the prover) can prove to another party (the verifier) that a given statement is true, without conveying any information apart from the fact that the statement is true.

Properties :-

i) Completness :- The verifier will always accepts a proof from the prover, given that they both follows the correct protocol.

ii) Soundness :- The verifier will not accept any "incorrect" proof from the prover, given that the verifier follows the correct protocol.

iii) Zero-Knowledge :- During the whole "proving" process, the verifier will learn nothing about the Prover's secret, nor will be able to prove that secret to any other party.

PROVER :- He knows some kind of secret but he don't want to share it with anyone, not even the verifier.

VERIFIER :- He verify whether (Prover) knows the secret or not.

CHALLENGE-RESPONSE AUTHENTICATION :-

→ It is a family of protocols in which one party presents a question ("challenge") and another party must provide a valid answer ("Response") to be authenticated.

→ Challenge-response protocol is password authentication, where the challenge is asking for the password and the valid response is the correct password.

SIMPLE AUTHENTICATION SEQUENCE :-

i) Server sends a unique value SC to client.

ii) Client generates unique challenge value CC

iii) Client computes Cr

$Cr = hash(cc + sc + secret)$

SC = Server generated Challenge

CC = Client gen. challenge

Cr = Client ~~gen.~~ response

Sr = Server response.

iv) Client sends **Cr** and **CC** to the server.

v) Server calculates the expected value of **Cr** and ensures the client responded correctly.

vi) Server computes **Sr** = hash (SC + CC + secret)

vii) Server sends **Sr**

viii) Client calculates the expected value of **Sr** and ensures the server responded correctly.

TECHNIQUES FOR C-R AUTHENTICATION :-

1). Using a Symmetric-key cipher
2). Using keyed-Hash functions
3) Using an Asymmetric-key cipher
4). Using Digital Signature

SIDE CHANNEL ATTACKS :- It is any attack based on information gained from the physical implementation of a cryptosystem. for ex. timing information, power consumption, electromagnetic leaks or even sound can provide an extra source of information, which can be exploited to break the system. RELY ON the relationship between information emitted (leaked) through a side channel and the secret data.

→ General class of S-C attacks :-

i) Timing attack - Based on measuring how much time various computations take to perform.

ii) Power-monitoring attack - Make use of varying power consumption by the hardware during computation.

iii) Differential fault analysis - In which secrets are discovered by introducing faults in a computation.

iv) Acoustic cryptanalysis - Attacks that exploits sound produced during a computation.

v) Row Hammer - In which off-limits memory can be changed by accessing adjacent memory.

→ COUNTERMEASURES :- i) Eliminate or reduce the release of secret information.
ii) Eliminate the relationship between the leaked information and the secret data.