Assignment -2

Name : Kawar Singh
Roll no : 17085035
Department : Electrical Engineering (B.tech)
Subject : CSE-537
Date Solved : 05/04/2020

(Q) Give a 4-step simple secret key distribution protocol and show that it is insecure against a MiM (Man-in-The-Middle) Attack.

→ A simple secret key distribution protocol is as follows. Suppose users A and B want to communicate

Step 1: A generates a public/private key pair $\{PU_a, PR_a\}$ and transmits a message to user B consisting of PU_a and an identifier ID_A.

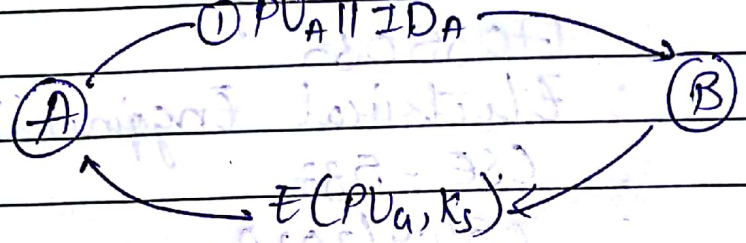
Step-2:- B generates a secret key, K_s and transmits it to A, encrypted with public key of A.

Step-3:- A computes $D(PR_a, E(PU_a, K_s))$ to get the secret key K_s, Now only A can decrypt the message sent by B. So A & B both know the secret key K_s.

Step-4:- A discards PU_a and PR_a and B discard PU_a

This protocol can be illustrated by

following diagram



Man-in-the-Middle (MiM) attack is easily possible suppose there is an adversary E in the middle whose can intercept this communication without being detected.

(1) A generates private/public key $\{PR_a, PU_a\}$ and transmits the message $PU_a \| ID_A$ to B.
(2) Adversary E intercepts this message and creates its own public key $\{PU_e, PR_e\}$ and sends $PU_e \| ID_B$ to B.
(3) B generates a secrete key K_s and transmits $E(PU_e, K_s)$
(4) E intercepts the message and finds $K_s = D(PR_e, E(PU_e, K_s))$
(5) E transmits $E(PU_a, K_s)$ to A.

Now, A & B share the secret key K_s and they are unaware that K_s has also been revealed to E. When A & B will send messages E can discover the messages they were trying to send since he knows the secret key K_s

(28) Suppose that A and B have already exchanged public keys. Show that they can share a secret key using a 5-step protocol that guarantees both confidentiality and authenticity.

→ The 5-step protocol can be described as follows. Suppose A & B want to communicate

(1) Step 1:- A uses B's public key to encrypt a message to B containing an identifier of A (ID_A) and a nonce (N_1), which uniquely identifies the transaction.

(2) Setup-2:- B decrypt the message using its private key and finds the nonce (N_1) and sends message to A encrypted with A's public key containing nonce (N_1) as well as new nonce generate by B (N_2)

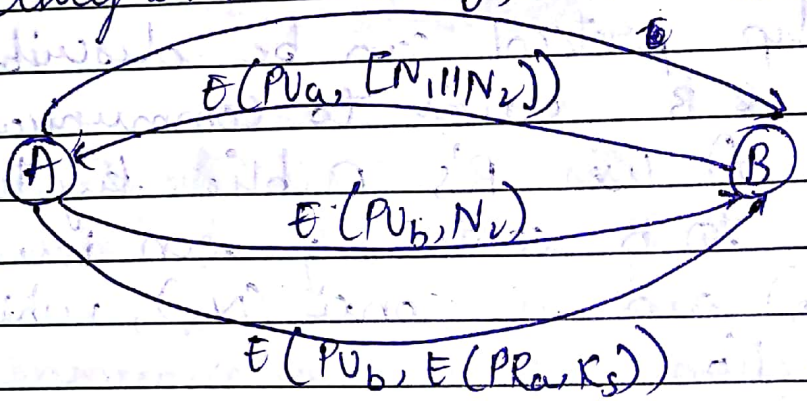
(3) Step-3:- A decrypts the message using its private key to finds N_1 and N_2 which ensures that his correspondent indeed is B. In order to ensure B that its correspondent is A he sends a message to B (using B's public keys) containing the nonce N_2.

(4) Step 4:- A selects a secret key K_s and sends $M = E(PU_b, E(PR_a, K_s))$ to B. This message assures that A is sending the message and only B could decrypt it.

Date:
Page No.

(*) Step-5:- B computes $D(PU_a, D(PR_b, m))$ to recover the secret key

The protocol can be described by following diagram $E(PU_b, [N_1 || ID_A])$



(30) Give a secret key exchange protocol using DH KE. Demonstrate a MiM attack against it.

→ Suppose user A and B want to communicate the necessary public values q and a should be known beforehand.

(1) User A generates a one-time private value X_A, calculates Y_A and sends Y_A to user B.

(2) User B responds by generating a private value X_B, calculates $Y_B = a^{X_B} \bmod q$ and sends Y_B to user A.

(3) User A, B calculates the secret key as

Date:
Page No.

$K_s = (Y_B)^{X_A} \bmod q$ (for A)
$K_s = (Y_A)^{X_B} \bmod q$ (for B)

A man-in-the-middle (MiM) attack is possible suppose Alice and Bob send a message and Darth is the attacker

(1) Darth generates two private values X_{D_1} and X_{D_2} and then computes public key: Y_{D_1} and Y_{D_2}

(2) When Alice transmits Y_{D_1} to Bob Darth intercepts Y_A and transmits Y_{D_1} to Bob. Darth also calculates $K_2 = (Y_B)^{X_{D_2}} \bmod q$

(3) Bob receives Y_{D_1} and calculates $k_1 = (Y_{D_1})^{X_B} \bmod q$. Bob also transmits Y_B to Alice

(4) Darth intercepts Y_B and transmits Y_{D_2} to Alice: Darth calculates $K_1 = (Y_B)^{X_{D_1}}$

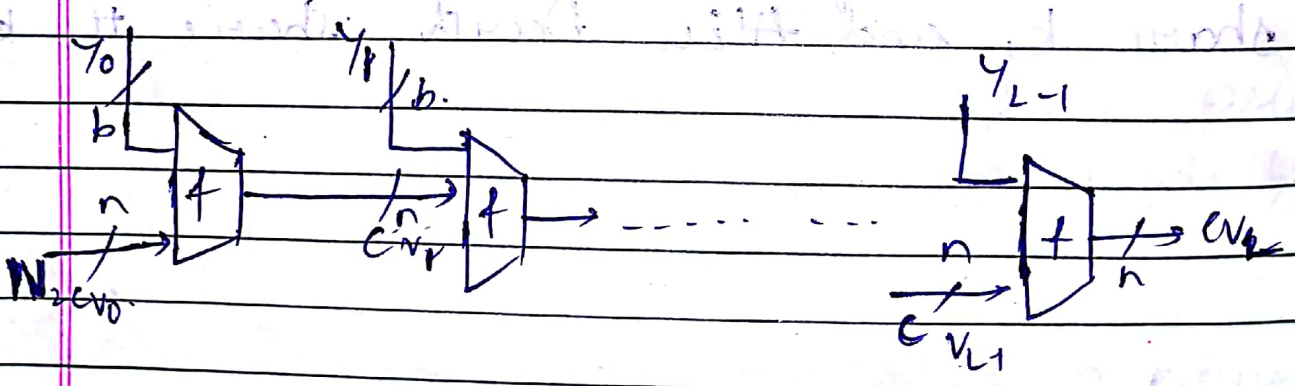
(5) Alice receives Y_{D_2} and calculates $K_2 = (Y_{D_2})^{X_A} \bmod q$. Now Alice and Bob think they share a secret key but in reality Bob and Darth share k_1 and Alice, Darth share the key K_2.

Name - Gaurav Singh
Roll no - 17085035
Dept. - Electrical Engineering (B.Tech)
Subject - CSE-537
Date solved - 20/04/2020

Assignment-3

(1Q) What is an iterated hash function? Give equations and corresponding diagram.

⇒ Many hash functions have a structure called iterated hash function. In iterated hash functions the input message is first partitioned into blocks of equal length. The hash function involves repeated use of a compression function f which takes two input (an n-bit input from the previous step, and a b-bit block) and produces an n-bit block



IV = Initial value
CV_i = Chaining variable
Y_i = i^{th} input block
f = compression function
L = Number of input blocks
n = Length of hash code
b = Length of input block

Date:
Page No.

Equations:

$CV_0 = IV$ = initial n-bit value
$CV_i = f(CV_{i-1}, Y_{i-1})$ $1 \le i \le L$
$H(M) = CV_L$

(8.2) Write the 5 steps followed by the SHA-512 algorithm (no need to discribe the round function).

→ SHA-512 follows the following 5-steps

Step 1:- Append the padding bits. The message is padded so that its length = 896 (mod 1024). Padding is always added, even if the message is already of the desired length. The number of padding bits is in the range of 1 to 1024. The padding consists of a single 1-bit followed by necessary number of 0 bits.

Step-2:- Append length → A block of 128 bits is appended to the message. The outcome of first two steps results the length of the message to be an integer multiple of 1024 bits.

Step-3:- A 512-bits buffer is used to hold intermediate and final results of hash function. The buffer is represented as eight 64-bits registers (a, b, c, d, e, f, g, h) initialized to following

Date:
Page No.

values
a = 6A09E667F3BCC908 b = BB67AE8584CAA738
bc = 3C6EF372FE94F82B d = A54FF53A5F1D36F1
e = 510E527ADE682D1 f = 9B05688C2B3F6C1F
g = 1F83D9ABFB41BD6B h = 5BE0CD19137E2179

The words are obtained by taking the first 64 bits of fractional parts of square roots of first 8 prime numbers

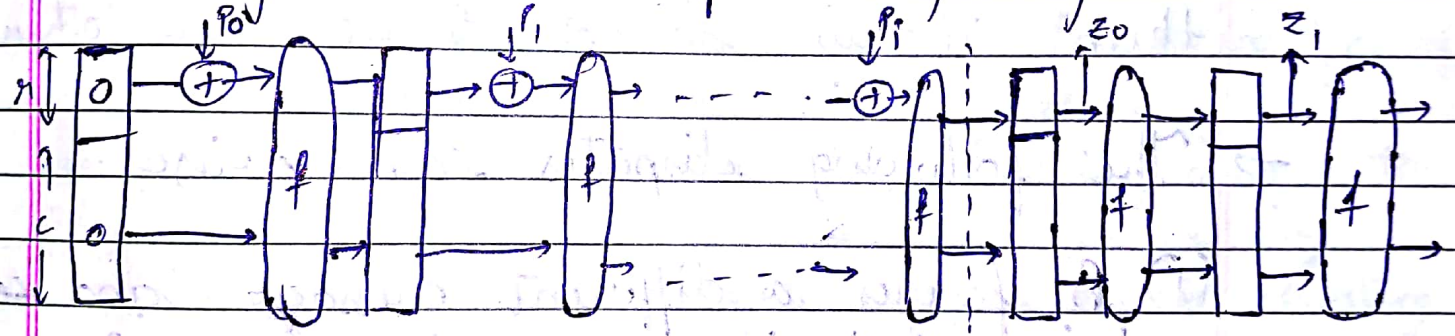
Step-4:- Process the message in 1024 bits blocks Each round takes 512 bit buffer (abcdefghi) and updates content of the buffer, Each round t makes use of a 64-bits value W_0 derived from current 1024-bit block. Each round also mates use of an additive constant K_t.

Step-5:- After all N 1024-bit blocks have processed the output from the N^{th} stage is the 512-bit message digest.

(3Q) How is SHA-3 different from previous versions of the algorithm?

→ In SHA-3 Sponge construction is used in which data is absolved into the sponge and the result is squeezed out. In absorbing phase, message blocks are xored

into a subset of the state, which is then transformed as a whole. In the squeeze phase, output blocks are read from the same subset of the state, alternated with state transformation. The size of the part written and read is called rate, and the part that is untouched by input/output is called capacity. The capacity determines the security of the scheme. The maximum security level is half the capacity



p_i message
z_i hash.

Date:
Page No.

Assignment - 4

Name - Gaurav Singh
Roll.no - 17085035
Department - Electrical Engineering (B.tech)
Subject - CSE-537
Date Solved - 29/04/2020

(1Q) Suppose A sends B a secure message with its authentication code. What possible unresolvable disputes can arise between them if they do not trust each other?

→ The following disputes can arise

(1) B forges a different message and ~~cob~~ claims that it came from A. B would simply have to create a message and append an authentication code using the key that A and B share

(2) A denies sending the message. Since it is possible ~~from~~ B to forge the message, there is no way to prove that John did in fact send the message.

(2Q) (a) State the essential requirements of a digital signature method

→ A digital signature technology must meet

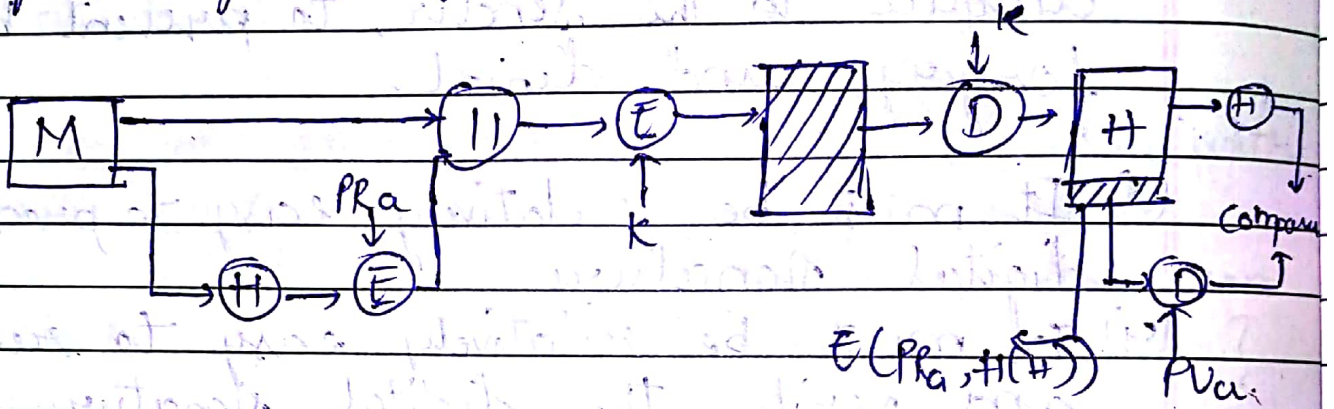
the following requirements.

(1) The signature must be a bit pattern that depends on the message being signed
(2) The signature must use some information unique to the sender, to prevent both forgery and denial.
(3) It must be relatively easy to produce the digital signature
(4) It must be relatively easy to recognize and verify the digital signature.
(5) 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.
(6) It must be practical to retain a copy of the digital signature in storage.

(b) With suitable examples distinguish between direct and arbitrated signature schemes

→ The direct digital signature involves only (source, distination). The distination knows the public key of the source. For example the hash of message M can be calculated and encrypted with sender's private key. It is appended to the original message M and encrypted with the key the

and the receiver ~~scheme~~ is here. At the receiver end the message is decrypted and hash value is calculated to verify the signature. It can illustrated by following diagram.



In arbitrated digital signature every signed message from a sender X to receiver Y goes first to an arbiter A, who subjects the message and its signature to a number of tests to check its origin and content. The message is then dated and sent to Y, with an indication that it has been verified to the satisfaction of the arbiter. The presence of A solves the problem that X might disown the message. For example consider the following arbitrated digital signature scheme:

$$X \rightarrow A: ID_X \| E(PR_X, [ID_X \| E(PU_Y, E(PR_X, M))])$$

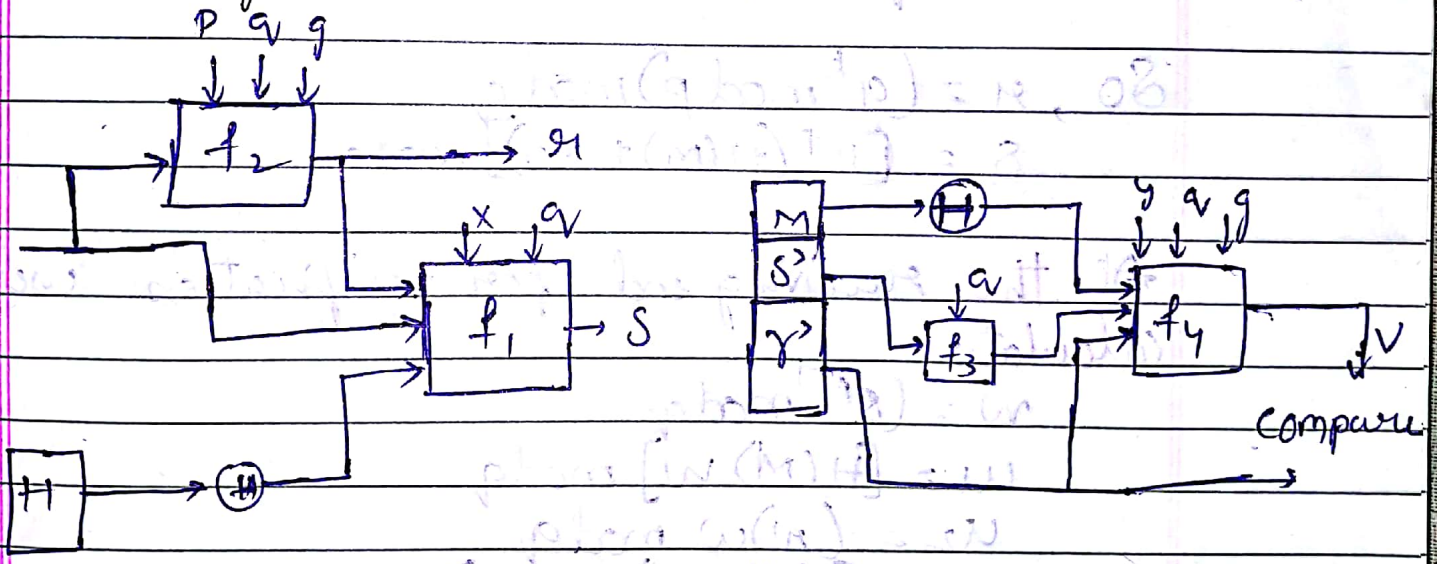
$$A \rightarrow Y: E(PR_a, [ID_X \| E(PU_Y, E(PR_X, M)) \| T])$$

If X wants to send a message to Y he

Sends its ID_x and encrypts the message with his private key and then with Y's public key attaches his ID_x and sends it to A by encrypting with has private key. A verifies X by decrypting with X's public key attaches a time stamp and sends it to Y. In this way Y is assured that message is comming from X and is verified by arbitor A

(39) Using a diagram explain the DSA. What quantities can be precomputed to accelerate the algorithm.

→



p, q, g are global-public key components. Here q is a prime number of 160 bits. A prime number p is selected with a length between 512 and 1024 bits such that q divides (p-1). Finally, g is chosen to be of the form $h^{(p-1)/q}$ where h is an integer between 1 and (p-1) with restriction that g must be greater

than 1. The user selects a private key x between 1 to q-1 and public key $y = g^x \bmod p$. Given y it is computationally infeasible to determine x, which is discrete logarithm of y to the base g mod p. To create a signature user calculates r and s that are functions of
(i) The public key components (p, q, g)
(ii) User's private key (x)
(iii) The hash code of the message H(M)
(iv) An integer k which is generated randomly $0 < k < q$.

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

At the receiving end for verification user calculates
$w = (s')^{-1} \bmod q$
$u_1 = [H(M) w] \bmod q$
$u_2 = (r')w \bmod q$
$v = [(g^{u_1} y^{u_2}) \bmod p] \bmod q$.
Now, if calculated value of v matches r' then the signature is verified.

The calculation of $g^k \bmod p$ is the dominating computation. It doesn't depend on the message. So it can be precomputed similarly k^{-1} is also precomputed.