Message Authentication and Hash Functions

5.4 Authentication Basics

- Authentication is a key feature in multi-user system
- It divide up resources with capabilities between many users
- It restricts user's access to resources.
- Typical authentication mechanism – passwords.

Syllabus Topic : Authentication Requirement

5.4.1 Authentication Requirements

In the context of communications across a network, the following attacks can be identified:

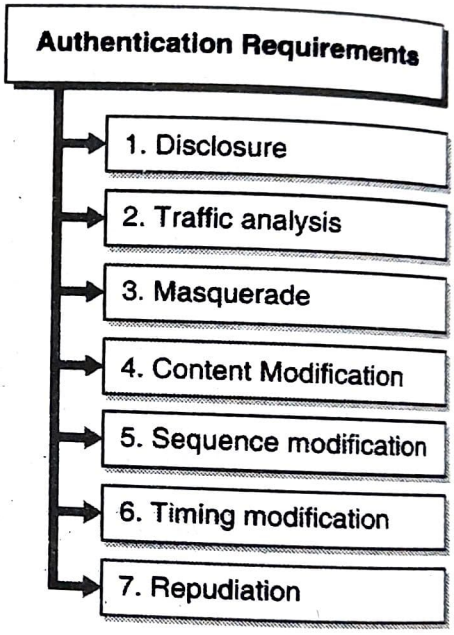

Fig. 5.4.1

1. Disclosure

Release of message contents to any person or process not possessing the appropriate cryptographic key.

2. Traffic Analysis

Discovery of the pattern of traffic between parties. In a connection oriented application, the frequency and duration of connections could be determined. In either a connection-oriented or connectionless environment, the number and length of messages between parties could be determined.

3. Masquerade

- Insertion of messages into the network from a fraudulent source.
- This includes the creation of messages by an opponent that are purported to come from an authorized entity. Also included are fraudulent acknowledgments of message receipt or non receipt by someone other than the message recipient.

4. Content Modification

Changes to the contents of a message, including insertion, deletion, transposition, or modification.

5. **Sequence Modification**

Any modification to a sequence of messages between parties, including insertion, deletion, and reordering.

6. **Timing Modification**

Delay or replay of messages. In a connection-orientated application, an entire session or sequence of messages could be a replay of some previous valid session, or individual messages in the sequence could be delayed or replayed.

7. **Repudiation**

Denial of receipt of message by destination or denial of transmission of message by source.

Syllabus Topic : Authentication Functions

5.4.2 Message Authentication Functions

- Message authentication is a process to make sure that messages are arrived from the alleged source and not altered while transmitting.
- Correctness and sequencing are also verified by message authentication.
- A digital signature, an authentication mechanism which also contains methods to counter denial either by source or by destination.
- Message authentication mechanisms have two levels.
- At lower level, some function generates a value known as an authenticator which is used to authenticate a message.
- Then this function is used as primitive in higher level authentication protocol that facilitates a receiver to authenticate the message.
- It also provides various functions used to generate an authenticator.
- These functions may be grouped into following three classes :
  1. **Message Encryption** : The cipher text of the message operates as its authenticator.
  2. **Message Authentication Code (MAC)** : A public function of the message and a fixed length value produced by a secret key operates as the authenticator.
  3. **Hash Functions** : A public function that correlates a message of any length into a fixed length hash value, which operates as the authenticator.

Syllabus Topic : Message Authentication Codes

5.4.3 Message Authentication Codes (MAC)

- MAC algorithm is a symmetric key cryptographic technique
- Message authentication is supported by MAC algorithm.
- A symmetric key K is shared by sender and receiver, for setting up a MAC process.
- Basically, to ensure message authentication, a message is sent along with an encrypted checksum generated by a MAC.
- MAC authentication process is illustrated in the following Fig.5.4.2.

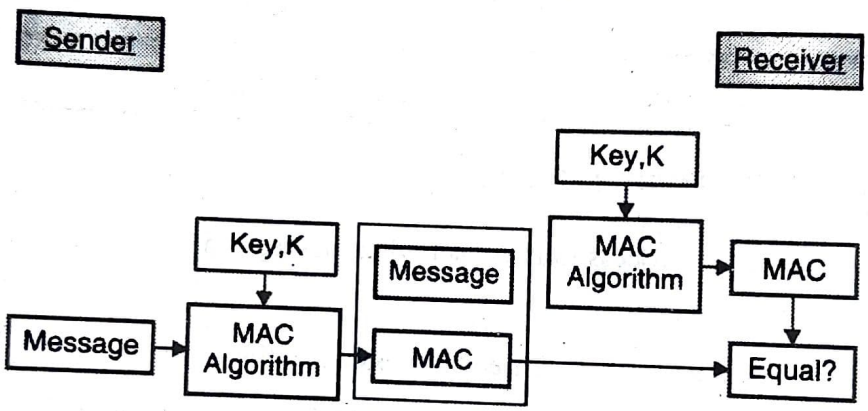

Fig. 5.4.2

- A MAC value is generated by a sender who uses commonly used MAC algorithms and by inputting some message and secret key K.
- Like hash, MAC function also reduces random generated long input into a fixed size output.
- For compression MAC uses secret key and this is mainly the difference between hash and MAC.
- Along with the MAC sender sends the message.
- Here we have to assure that the message communication happens in authenticated environment.
- Message needs to be encrypted in case of confidentiality.
- Once the message and the MAC is received by the receiver, he supplies the shared key K and received message into MAC algorithm and regenerates the MAC value.
- Now, the receiver verifies uniformity of recently computed MAC with the MAC obtained from the sender.
- If the computed MAC match with the MAC obtained from the sender, then the message is accepted by the receiver and he assures that the message has been transmitted by the intended sender.

- If it does not match, the receiver cannot verify whether the message has been changed or not and also whether the origin of the message is reliable or not.
- A receiver assumes that the message is not authentic.

5.4.4 Limitations of MAC

Mainly there are two limitations of MAC, both as a result of its symmetric kind of operation

- Formation of Shared Secret.
  - For legitimate pre-decided users who knows shared key, it provides message authentication.
  - It needs formation of shared secret before the use of MAC.
- **Lack of ability to Provide Non-Repudiation**
  - Non Repudiation ensures that either of the parties involved in communication do not deny sending or receiving of data.
  - Non repudiation service is not provided by MAC technique.
  - MAC cannot give assurance of whether the message was really sent by the sender in case of disagreement over message origination by the sender and receiver.
  - Even if third party can not compute the MAC, sender might refuse having sent the message and claimed falsity of the message by receiver as it is not possible to verify parties involved in computing MAC.

5.4.5 Requirements for Message Authentication Codes (MAC)

- A Message Authentication Code (MAC) is also known as a cryptographic checksum.
- MAC is obtained by a function C of the form:

  T = MAC(K, M)

  Where,
  - M is message of variable length
  - A secret key K is shared only by sender and receiver
  - Fixed length authenticator, MAC(K, M) called a tag.
- If key size k is greater than MAC size n; k>n then
  - For known value of T1 and M1 with T1 = MAC(K, M), for each value of i, cryptanalyst can perform Ti = MAC(Ki, M1).
  - Total of 2^k tags will be generated.
  - But only $2^n < 2^k$ different tag values.
- The correct tag will be created by a number of keys.

- Average number of keys that produce match : $2^{(k-n)}$
- To search for a match k = m * n, m rounds are needed.
- Only one round is required to find a match, if k <= n.
- If the other party notices the value of M and MAC(K, M), it should be practically impossible to the opponent to construct a message M' such that
- MAC(K, M') = MAC(K, M)
- MAC(K, M) should be uniformly disseminated in such a way that for randomly selected messages M' and M, the probability that MAC(K, M) = MAC(K, M') is $2^{(-n)}$ where n is the number of bits in the tag.
- Let M' equivalent to some known transformation on M. i.e M' = f(M). For example, f may consist of inverting one or more specific bits.

  Pr [MAC(K, M) = MAC(K, M')] = $2^{(-n)}$

Syllabus Topic : Hash Function

5.5 Hash Function

- Storing data in an array with the aim of performing operations like sorting, inserting, deleting and searching speedily is called as hashing.
- Unique key is needed to perform such operations with hashing.
- It finds the correct location of a record by comparing the records also searching for the location of element in the array.
- '**Hash function**' is the function that returns the position of the record in array and '**hash table**' is the array used to store records.
- It is very difficult to find out an appropriate hash function and its execution algorithm though it is essential for better hash table performance.
- A basic requirement is that the function must provide an equal distribution of hash values.
- Unequal distribution rises the number of collisions and efforts for undertaking them.

5.5.1 Uses of Hash Tables

(1) Compilers use hash tables for symbol storage.

(2) Hash tables are used in disk based data indexing in Databases.

(3) Hash tables are used in high speed routing.

(4) Hash tables are used in many algorithms for speedily processing of data.

(5) Hash tables are used in various dynamic languages like Python, JavaScript, Perl to implement objects.

(6) The Linux Kernel uses hash tables to manage memory pages and buffers.

5.5.2 Advantages of Hash Tables

(1) Synchronization

(2) More efficient than arrays, search trees or table lookup structure.

(3) Speedily access the data.

5.5.3 Need of Hashing

- Many application like search engines, web pages, social networking web sites deals with large amount of data.
- To search for a particular value from this massive amount of data we can use countless look ups. But they are time consuming.
- Data structures like arrays, linked list may not be efficient enough to handle sufficient searches.
- So efficient search techniques like hashing is used to minimize such comparisons and make the process fast.
- In hashing the searching depends upon the location of the record and not the location with respect to other keys.

5.5.4 Hash Table Example

- The simple example of hash table is considered as an array of records.
- This example has 10 records (0 – 9).
- Each record has a key associated with it.
- To insert a new record, key has to be converted into an array index which is considered to be a hash value of the key.
- We need to add 3 element in hash table :- 1236, 1207, 1200
- We need to apply the division method where we have to divide the elements by 10 and use a remainder as an index.

  1236 mod 10 = 6

  1207 mod 10 = 7

  1200 mod 10 = 0

0	1	2	3	4	5	6	7	8	9
1200						1236	1207		

- While using the hash table, some places contain the valid records whereas other remains empty.
- If we want to insert a new record 1232, it will occupy index 2 of an array.

  1232 mod 10 = 2

0	1	2	3	4	5	6	7	8	9
1200		1202				1236	1207		

- When two records produces same hash key, then **collision** occurs.
- To insert a record 1216, in hash table 6^{th} index is needed.

  1216 mod 10 = 6
- But 6^{th} place is already occupied, which is called collision.
- So when collision occurs, the record needs to be inserted into the next empty cell in hash table.

0	1	2	3	4	5	6	7	8	9
1200						1236	1207	1216	

5.6 The Birthday Attack

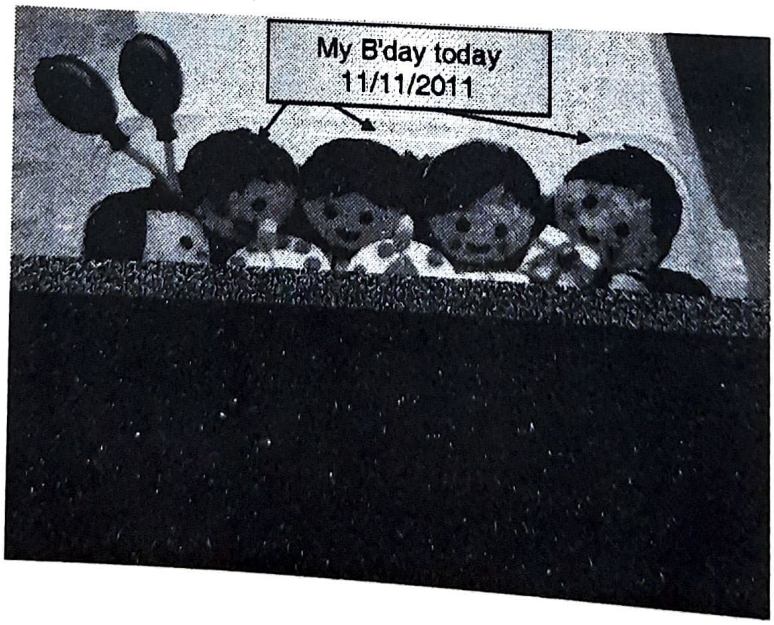

Fig. 5.6.1

What's the chance of two people in a group having the same birthday?

With only 23 people, the probability is greater than 50 %!

- Birthday attacks use brute force techniques to attempt to solve cryptographic hash function problem.
- It is a cryptanalytic technique.
- It is used to find collisions in a hash function.
- The surprising result when the probability of two or more people in a group of 23 shares the similar birthday is more than 0.5. Hence the name birthday attack.
- Some function when provided with a random input, returns any of k equally likely values.
- So by continuously executing the same function again for various inputs, we presume to get the same output after 1.2 $\sqrt{k}$ evaluations.
- Substitute k with 365.
- The source, S is ready to sign the message by adding the suitable m-bit hash code and by using A's private key encrypting the same hash code.
- The destination, D produces $2^{m/2}$ alterations on the message, which expresses the similar meaning.
- D generates the same number of messages, which are supposed to be the variations of the fake message to be replaced for the real one.
- Both the messages are compared to obtain a pair of messages that generates the same hash code.
- The probability of success is more than 0.5.
- If there is no match, then more valid and fake messages are produced until match is found.
- D sends valid variations to S for signature which can be attached with the fake variations to be transmitted to D.
- As both variations have the identical hash code, they will generate the same hash code. Even then D assures success while not knowing the encryption key.

5.7 Hash Functions from Cryptosystems

- In cryptography, cryptographic hash functions play a vital role.
- In network security, Cryptographic hash function are used to provide :
  - Messages authenticity security
  - Verification of data integrity which inhibits data alteration from being hidden.
  - Digital signature and time stamping method
- In all network security applications, hash functions occur and are very useful.

- A hash function is a numerical function that translates a numerical input value to a different compressed numerical value.
- The hash function output is of fixed length but input is of random length.
- Values received by a hash function are called as message digester.
- The following image describes hash function

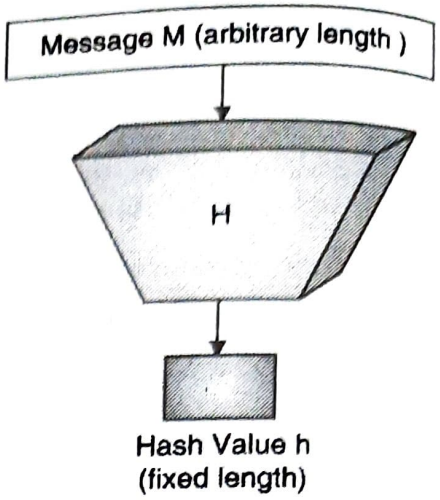

Fig. 5.7.1

5.7.1 Features of Hash Functions

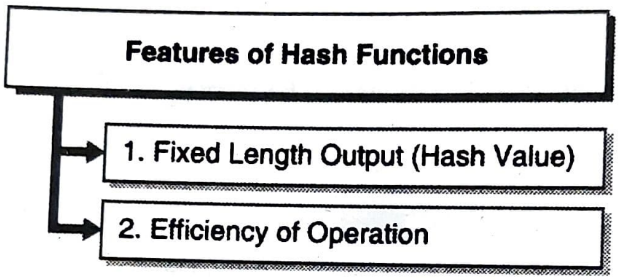

Fig. 5.7.2

1. Fixed Length Output (Hash Value)

- Hashing of data means converting arbitrary length data into a fixed length by using hash function.
- Basically, hash value is much lesser than the input data. So sometimes hash functions are also called as compression function.
- Hash functions are also called as Hash digest means it is also to represent larger data smaller.
- n-bit hash function means hash function with n bit output.
- Generally, hash function produces values between 160 and 512 bits.

2. Efficiency of Operation

- Ideally, for any input x with any hash function, calculation of h(x) is a faster operation.
- Mathematically, calculation of hash functions are done much rapidly than symmetric encryption

- A function that accepts some text of random length as input and converts it into fixed length output such as a message digest, a digital fingerprint, a hash value or a checksum, is called as a hash function.
- A hash function defines it function as

  $f : d \Rightarrow r$

  where,

  d = {0, 1} is a domain which denotes that the domain elements comprises binary string of random length.

  $r = \{0, 1\}^n$ is a range for some n > = 1, which denotes that range elements comprises fixed length binary string.

  f is a function that accepts random size message M and produces result of n size fixed length hash value.
- A hash function is also called as compression function as values of domain d are finite.

5.7.2 A Cryptographic Hash Function h has the Following Security Properties

1. Input of h should be a random size block of data.
2. Fixed length output, independent of input, should be produced by h.
3. Working of h should be like a random function though it's effectively reproducible and deterministic.
4. For a message M, computation of its digest H is very easy that means in polynomial time O(n) h can be computed where n is the input message length, which makes implementation of hardware and software is feasible and practical.
5. For a message digest H, computing the value of M is difficult; h(M) = H. It is known as pre-image or one-way resistance property. It means one should be incapable to recover the original text from its hash value.
6. For message M1, it is mathematically infeasible to compute another message M2 ≠ M1 with H(M1) = H(M2). It is known as second pre-image or weak collision resistance property.
7. It is mathematically infeasible to compute random pair of different messages (M1, M2) such that H(M1) = H(M2). This is known as the strong collision resistance property.
8. The security of the hash function not only initiates for maintaining hash function secret but also for its ability to create collision free one way hash values.
9. Hash functions used with symmetric or asymmetric key is known as Message Authentication Code (MAC).

Syllabus Topic : Security of Hash Functions and MACs

5.8 Security of Hash Functions and MACs

Attacks on hash functions and MACs are categorized into two parts when used with symmetric and public key encryption.

- Brute force attacks
- Cryptanalysis

5.8.1 Brute Force Attacks

The workings of brute-force attacks are different for hash functions and MACs.

Hash Functions

- For brute-force attacks, the strength of a hash function is fully dependent on the length of hash code generated by algorithm
- The following are three properties are required for hash functions :-
  - **One-way :** For a certain code h, it is mathematically impossible to compute x such that $H(x) = h$. The efforts required for a hash code of length n is 2^n
  - **Weak collision resistance :** For a certain block x, it mathematically impossible to compute y, x with $H(y) = H(x)$. The efforts required for a hash code of length n is 2^n
  - **Strong collision resistance :** For a certain pair (x, y) such that $H(x) = H(y)$, it is mathematically impossible to compute such pair. The efforts required for a hash code of length n is $2^{n/2}$

Message Authentication Codes

- A brute force attack on MAC is very difficult task as it needs known message – MAC pairs.
- The following steps shows the attack on hash code
  - n – bit hash code $h = H(x)$, where x is a fixed message.
  - A brute force technique for finding a collision is to select an arbitrary bit string y and verify if $H(y) = H(x)$.
  - The attacker can perform this repetitively off line.
- The following property of a MAC algorithm is required :-
  - Computation resistance :- It is mathematically impossible to calculate some text – MAC pair $(x, C_k(x))$, for the new input $x \neq x_i$, even though the multiple text – MAC pairs $(x_i, C_K[x_i])$,
  - It means, for a given message x, the attacker may generate the valid MAC code.

- There are two possibilities of attack
  - Attack the key space
  - Attack the MAC values

5.8.2 Cryptanalysis

Cryptanalytic attacks performed on hash MAC algorithms and hash functions obtained to develop some property of the algorithm to make some attack rather than an extensive search.

Syllabus Topic : Secure Hash Algorithm

5.9 Secure Hash Algorithm

- Secure Hashing Algorithm known as SHA, is a group of cryptographic functions intended to maintain data security.
- It functions by converting the data using a hash function which uses an algorithm that contains modular additions, compression functions and bitwise operations.
- Then the hash functions generated a fixed size string which is similar to original string.
- These algorithms are proposed to be one way functions i.e. it is virtually impractical to convert them into original form once they are altered into their corresponding hash values.
- Algorithms like SHA – 1, SHA – 2, SHA – 5 are mainly developed with gradually stronger encryption with respect to hacker attacks.
- A basic application of SHA is encrypting passwords because rather than keeping track of actual password, the server side only has to maintain specific user's hash value.
- It is useful in case an attacker attacks on the database, as they will only get the hashed function and not the real passwords, thus if the input is considered to be the hashed value as a password, it will be transformed into different string by the hash function and then it deny the access.
- Also, SHA reveals the avalanche effect, where alteration of very limited encrypted letters affects a big change in output; or extremely dissimilar strings produce identical hash values.
- This result affects on hash values for not sharing any information about the input string like its original length.
- Moreover, SHAs are also used to expose the altering of data by attackers, in case, if text file is altered to some extent and it is hardly noticed, then the updated file's hash value and original file's hash value will be dissimilar, and thus tampering of message will be perceptible.

5.9.1 Features of SHA

- It is impossible to find the original message, provided its message digests.
- It is impossible to obtain two messages generating the identical message digest.

5.9.2 Working of SHA

Step 1 : Padding

End of the original message is appended with padding in such a way that the message length is 64 bits smaller than the multiple of 512

Step 2 : Append Length

Calculation of the message length eliminating padding length is appended as 64 bit block to the end of padding.

Step 3 : Split the input into 512 bit blocks

The input is spilt into blocks of length 512 bits each.

Step 4 : Initialize chaining variables

Initialization of A to E, chaining variables and their hexadecimal values are

A : 01 23 45 67

B : 89 ab cd ef

C : fe dc ba 98

D : 76 54 32 10

E : C3 D2 e1 f0

Step 5: process block

To process the blocks the following steps are considered :

1. Five chaining variables are copied into five corresponding variables a, b, c, d, e. Single register is considered for keeping temporary, intermediary and final result.
2. Separating the current 512 bit block into 16 sub block each of size 32 bit.
3. SHA supports 4 rounds with 20 iterations each. So total 80 iterations and in each round processing of all 16 sub blocks takes place.

Syllabus Topic : HMAC

5.10 HMAC (Hash-based Message Authentication Code)

- In cryptography, Hash based Message Authentication Code is a particular type of Message Authentication Code (MAC) that consists of a secret cryptographic key and a hash function.
- HMAC provides a private key each for client and the server which is only known to that particular client and server.
- For each request of the server, client generates a unique HMAC by hashing the requested data with the help of private keys and dispatching it as a part of request.
- It is used to verify message authentication and data integrity simultaneously with MAC.
- The algorithm offers better protection against extensive length attacks as the message and key are hashed in different steps.
- Instead of encrypting the message HMAC sends it along with HMAC hash.
- Again the parties with secret key will themselves hash the message and if it is authorized then the calculated and received hashes will match.
- Once the request is received by the server and regenerated its unique HMAC, comparison of two HMAC takes place. If they are equal, the client is trustworthy and executed its request.

5.10.1 Working of HMAC

Consider the following variables :

MD = Message digest or hash function which is used (eg. MD5, SHA-1, etc)

M = Input message whose MAC is to be computed

L = Number of blocks in the message

b = Number of bits in each block

K = Shared symmetric key which is to be used for HMAC

Ipad = A string 00110110 repeated b/8 times

Opad = A string 01011010 repeated b/8 times

Step 1:

Make the key length K equal to b

There are three possibilities existed depending on the key length K

Key length K less than b

- The size of key length K is increased so that its size and total number of bits b in the initial message block are equal. To make this, left of K are appended with 0 bits.
- Example :- if the original key length K = 150 bits and b = 256 bits then append 106 bits to the left all with a value 0.

Key Length K equal to b

None of the action will be performed and it will proceed towards step 2 .

Key length K greater than b

- Reduce the key size K to make its size and total numbers of bits b in the initial message blocks are equal.
- The value of K is passed to the message digest algorithm (H) particularly selected for this case of HMAC, which produces key K, reduced so that key length K and value of b are equal.

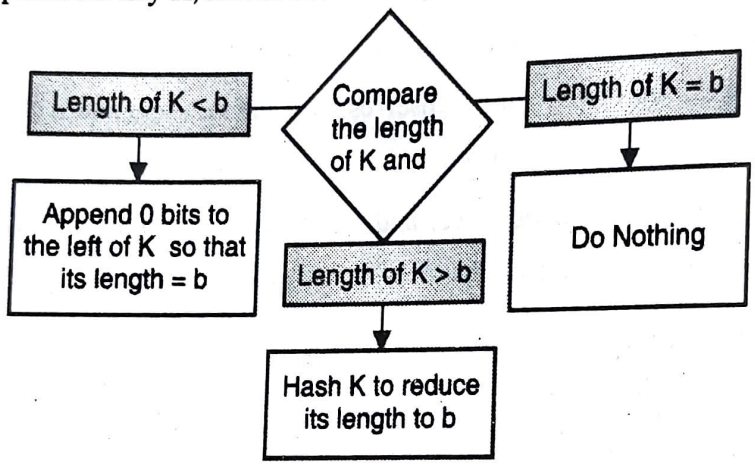

Fig. 5.10.1

Step 2 : XOR operation between K and ipad to get the value of S1

XOR operation is performed between key K generated in step 1 and ipad to get the value of variable S1.

Step 3 : S1 is appended by message M

End of S1 is appended by original message (M) generated in step 2.

Step 4 : Message digest algorithm

Message digest algorithm e.g. SHA- 1, MD5, etc. is selected and applied on the step 3 output which operation is further known as H.

Step 5 : XOR operation between key length K and opad to know the value of S2

XOR operation is performed between key K generated in Step1 and opad to get the value of a variable S2.

Step 6 : S2 is appended by message digest H

End of s2 produced in step 5 is appended by message digest H generated in step 5.

Step 7 : Message digest algorithm

Message digest algorithm e.g. SHA- 1, MD5, etc. is selected and applied on the step 6 output which is further consider to be final MAC.

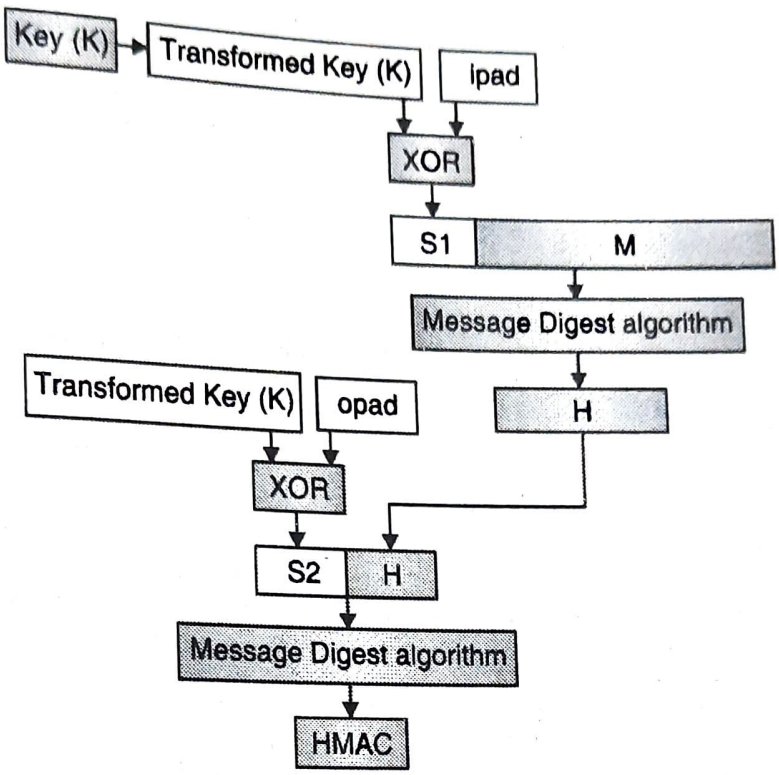

Fig. 5.10.2

Review Questions

Q. 1 Explain the concept of key management in cryptography.

Q. 2 Explain in detail Diffie-Hellman key exchange algorithm with example.

Q. 3 Describe the term MAC.

Q. 4 What are the requirements of Message Authentication Code (MAC)?

Q. 5 What is the need of hashing?

Q. 6 Explain birthday attack problem in detail.

Q. 7 Explain the working of SHA.

Q. 8 Write a note on HMAC.

Q. 9 Explain the working of HMAC.

❑❑❑

CHAPTER 6

Unit II

Digital Signatures and Authentication

Syllabus

Digital Signatures and Authentication: Digital Signatures, Authentication Protocols, Digital Signature Standard.

Syllabus Topic : Digital Signatures

6.1 Digital Signatures

- Digital signatures, encryption and message authentication are all considered to be very useful in implementation of modern cryptography.
- A signature is a known technique used for public key cryptography based non – repudiation.
- Digital signature is a mathematical technique used to produce authenticity of digital documents.
- Digital signature assures recipients :
  - Authentication - The message is sent by the known sender
  - Non – repudiation – The sender cannot deny that he has not sent the message
  - Integrity – While transmitting the message it has not altered
- Integrity of the message and the original source of the document are ensured by attaching code like signature.
- Digital signatures are known to be public key primitives of message authentication.
- Even, through this technique person / entity get connected to the digital data.
- Receiver of the message or any third party can independently verify this binding.
- Digital signature is a cryptographic value computed from secret key and data and is known only to the signer.

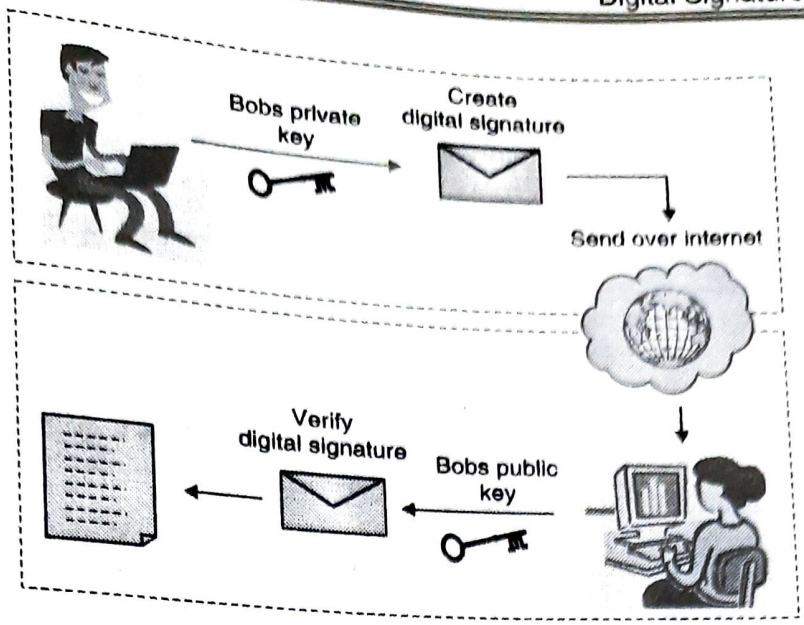

Fig. 6.1.1

- Digital signature gives assurance of proof of origin, identity and an electronic document status. Also it acknowledges informed authorities and signatory endorsement. So most of the organizations are using digital signatures as their authenticity stamps.

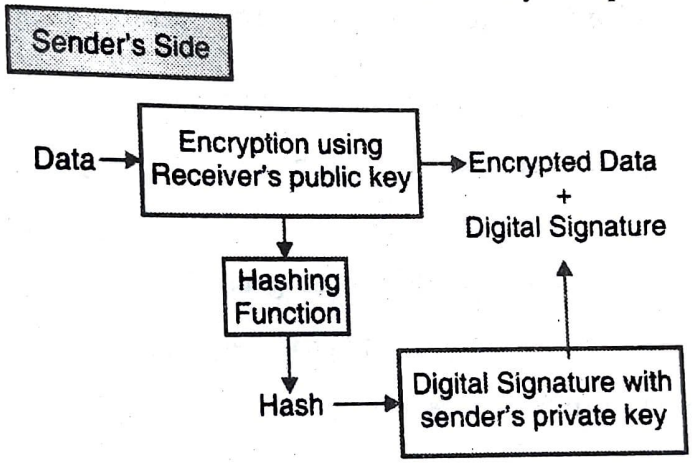

Fig. 6.1.2 : Encryption with digital signature

6.1.1 Properties of Digital Signature

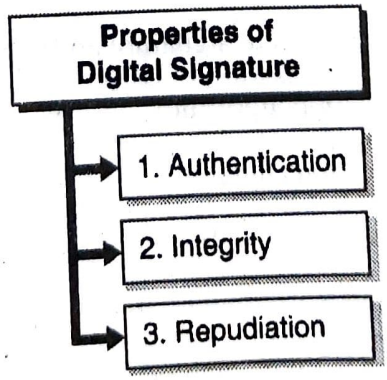

Fig. 6.1.3

The digital signature is applied for communication for the following reasons :

1. **Authentication**
   - Digital signatures verifies the source of messages.
   - When each user has their own digital signature secret key, a valid signature proves that a particular user has only sent that message.
2. **Integrity**
   - The sender and receiver of a message are sure about during transmission the message has not altered that is originality of the message.
   - Even the contents of encrypted message can be altered some times. But when the document is digitally signed, any alteration in the message after signature will cause in invalidate the signature.
3. **Repudiation**

   After signing on the documents, the entity cannot deny having signed it.

6.1.2 Processes Involved in Digital Signature

- Every party involved in the process should have a public and private key pair.
- Basically, Different key pairs are used for the process of verifying / signing and encryption /decryption.
- Signature key is that is private key is used for signing and the verification key is the public key.
- Hash function data is provided to the signer and produces hash of data.
- The signature algorithm is then fed by the signature key and hash value.
- It generates the digital signature on that particular hash.
- Data is appended by the signature.
- Both Signature and data are sent to the verifier.
- Verification algorithm is fed by the digital signature and the verification key by the verifier.
- Output is generated by the verification algorithm.
- To produce the hash value, verifier executes same hash function on the data which is received.
- Output of verification algorithm and this hash value are compared for verification.
- The validity of the digital signature is verified by the verifier which is based on comparison result.
- As signer generates digital signature by its own private key, he cannot in future deny about signing the document in future.

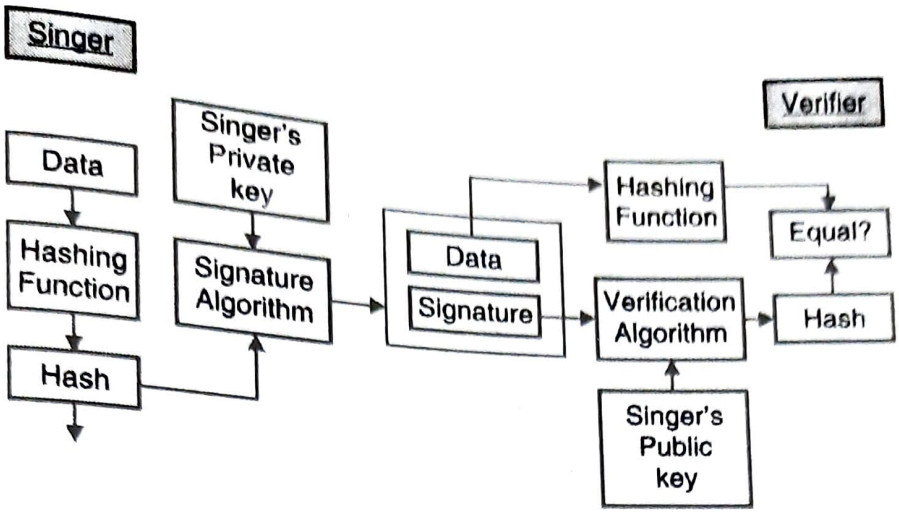

Fig. 6.1.4

6.1.3 Requirements for Digital Signature

- The signature should be in bit pattern format which depends on message being signed.
- The signature should use at least few unique information details to prevent attacks and forgeries.
- The signature should be easily produced.
- Signature should be easily recognized and verified.
- It must be sensible to preserve a copy of the digital signature in storage.
- It should 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.

6.2 Attacks and Forgeries

The following listed are the types of attacks where, **V** (Victim) denotes the user whose signature method is being attacked, and **A** (Attacker) denotes the attacker.

- **Key only attack** : A only knows V's public key.
- **Known message attack** : Particular messages and their corresponding signatures are accessible by **A** .
- **Generic chosen message attack** : A chooses a list of messages before trying to break V's signature pattern, independent of V's public key. A then obtains from V valid signatures for selected messages.
- **Directed chosen message attack** : It is similar to generic chosen message attack; except that prior to seeing the signature A selects set of messages to be signed when it knows V's public key

- **Adaptive chosen message attack** : A can use V as predictor means V can demand for signatures of messages depends on message signature pairs received earlier.
- The following listed are the types of forgeries where, V can perform with a non-negligible probability.
- **Total break** : A determines V's private key.
- **Universal forgery** : A finds an efficient signing algorithm that provides the same way of constructing signatures on arbitrary messages.
- **Selective forgery** : A forges a signature for a particular message chosen by A.
- **Existential forgery** : A forges a signature for at least one message. A has no control over the message.

Syllabus Topic : Authentication Protocols

6.3 Authentication Protocols

- A type of cryptographic protocol or communication protocol specially designed for transmitting the authentication data between two parties is called as an authentication protocol.
- The connecting entity (client) is authenticated by the receiving entity and the connecting entity (server) is authenticated by the authentication protocol by giving the needed information for syntax and authentication.
- It is the most important layer of security needed for protected communication within the networks.

6.3.1 Purpose of Authentication Protocols

- The need for protecting from unauthorized access of data arises due to accessibility of large amount of reliable information over the network.
- It is easy to steal someone's identity.
- To check the authenticity of origin of data, special verification methods need to be invented.
- The function of authentication protocol is to identify the correct series of steps required for authentication execution.
- It has to fulfil the main protocol principles :-
  - Two or more parties should be involved in protocol.
  - Each party must know the details of the protocol in advance.
  - The protocol should be followed by all the parties who are included.
  - A protocol should to be unambiguous, it means every step have to be described precisely.

- A protocol must be entirely perfect, it means for each possible situation specified action should be included.

6.3.2 Password-based Authentication using Authentication Protocols

- Alice (client) and Bob (server) are both agreed on using the same protocol.
- Bob is having Alice's password which is kept in a database for comparison.
- Alice sends her password in a packet satisfying the protocol rules to the bob.
- Bob verifies the password which is received with the stored password in his database.
- Then based on the result he sends "Authentication successful" or "Authentication fail" packet.
- The above is an example of common authentication protocol exposed to multiple threats like replay attack, brute force attacks, eavesdropping or man in the middle attack.
- To become strong against such kind of attacks, most of the authentication protocols are designed to be more complicated.

The following is the description of various authentication protocols :

1. **CHAP (Challenge Handshake Authentication Protocol)**
   - It is a three way handshake protocol which is considered more secure than PAP and SPAP because it doesn't transmit the password in clear text.
   - The server sends a challenge to the client which must decrypt it and return the correct response. This allows the server to verify the user's credentials without sending them across an insecure link.

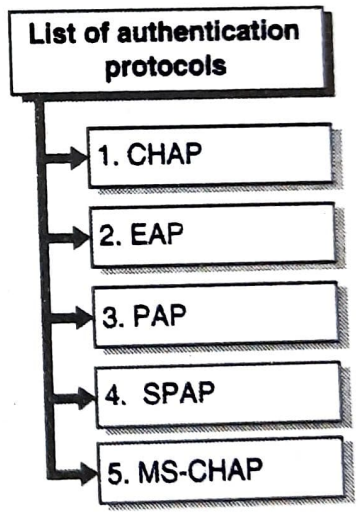

Fig. 6.3.1

2. **EAP (Extensible Authentication Protocol)**
   - It is operated between server and dial in client to know the choice of authentication protocol to be used. It is used for various security means like certificates.
   - It is also used in devices like biometric readers and smart card readers as it is designed to work with various types of security.
3. **PAP (Password Authentication Protocol**
   - It is a common and basic type of authentication protocol. It is configured to use with PPP. It is a two way handshake protocol.

- It transfers details of all authentications in the form of clear text without encryption. This protocol is exposed to hackers due to the said reason.
- Client and server are also cannot be able to authenticate each other.

4. SPAP (Shiva Password Authentication Protocol)

- It is more secure than PAP but it is rarely used.
- It is used for communicating with remote access devices made by Shiva. It is used for backward compatibility.

5. MS-CHAP (Microsoft Challenge Handshake Authentication Protocol)

- It is the extension of CHAP by Microsoft. It is used for integrated Windows authentication. Two versions of MS-CHAP are :- v1, v2.
- Though MS-CHAP v2 is more secure than MS-CHAP v1, it is not supported by all systems.

Syllabus Topic : Digital Signature Standard

6.4 Digital Signature Standard (DSS)

- The U. S. National Security Agency developed Digital Signature Standard (DSS).
- It is a collection of standards and processes for producing a digital signature used for authentication of electronic documents.
- It is Specified as Federal Information Processing Standard 186 by the National Institute of Standards and Technology (NIST) in 1994, the Digital Signature Standard has become the U.S. government standard for authenticating electronic documents.
- The Digital Signature Standard is intended to be used in electronic funds transfer, software distribution, electronic mail, data storage and applications which require high data integrity assurance. The Digital Signature Standard can be implemented in software, hardware or firmware.
- Digital Signature Algorithm (DSA) is used for the implementation of DSS.
- DSA is a pair of large numbers which are calculated as stated by the specific algorithm which enables the verification of parties involved in communication and data integrity within parameters.
- Digital signatures are verified and generated by DSA.
- Digital signatures are produced in combination with private key and verification takes place by corresponding public key.
- Each party should have their private and public key.

A signature can only be produced by an authenticated person having their private keys and the signature verification can be done by anyone possessing the public key.

Difference between signature and encryption in DSS is operation on digital signature is reversible whereas it is irreversible with encryption.

DSS are incapable of key exchange or key distribution.

Security of DSS mainly depends on privacy of the private key of the party.

DSS ensures about the authentication of the digital signature and security of digital signatures carrying electronic documents.

DSS also ensures non repudiation of signatures and offers all necessary protections for imposter prevention.

DSS also ensures tracking of digital signed documents.

Review Questions

Q. 1 Explain the concepts of digital signature.

Q. 2 What are the properties a digital signature should have ?

Q. 3 What requirements should a digital signature scheme satisfy ?

Q. 4 Explain the term authentication protocol in cryptography.

Q. 5 List out and describe various authentication protocols.

Q. 6 What do you mean by Digital Signature Standard?

❑❑❑