UNIT-3

• BUTTER FLOW ATTACK

In computer security and programming, a butter overflow is an anomaly where a program, while writing data to a buffer, overruns the buffer's boundary and overwrites adjacent memory locations.

Buffers are areas of memory set aside to hold data, often while moving it from one section of program to another, or between programs.

A Buffer overflow occurs when data written to a buffer also corrupts data values in memory addresses adjacent to destination buffer due to insufficient bounds checking. This can occur when copying data from one buffer to another without first checking that the data fits within the destination buffer.

Heap Based – It is difficult to execute and least common of the two.
└ Attack an application by flooding the memory space reserved for a program.

Stack Based – More common among attackers.
└ It exploits application and programs by using stack memory space used to stored user input.

These can result in erratic program behaviour, including memory access errors, incorrect results, a crash or a breach of system security. Thus, Buffer flow attacks are the basis of many software vulnerabilities and can be maliciously exploited.

• DISTRIBUTED DENIAL OF SERVICES

DDOS attacks make computer system inaccessible by flooding servers, networks or even end user systems with useless traffic so that legitmate users can no longer gain access to those resources.

Denial of service (DoS) attack is an attempt to prevent legitmate users of a service from using that service.

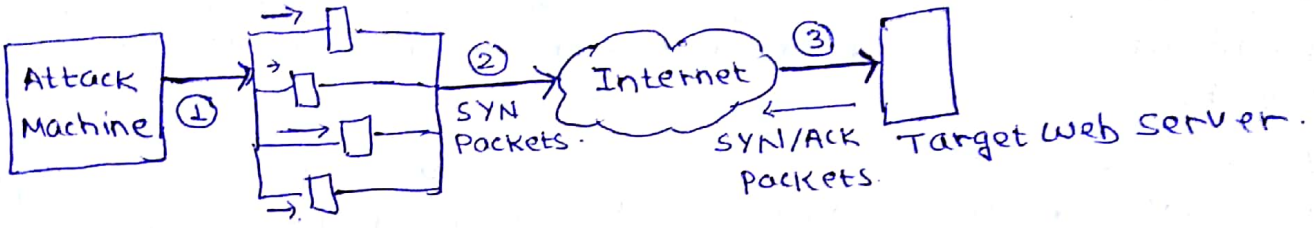
In DDos attack, an attacker is able to recruit a number of hosts throughout the Internet to simultaneously or in a coordinated fashion launch an attack upon the target by sending useless packets to the target.

- ## TYPES OF DDOS

A DDoS attack attempts to consume the target's resources so that it cannot provide service.

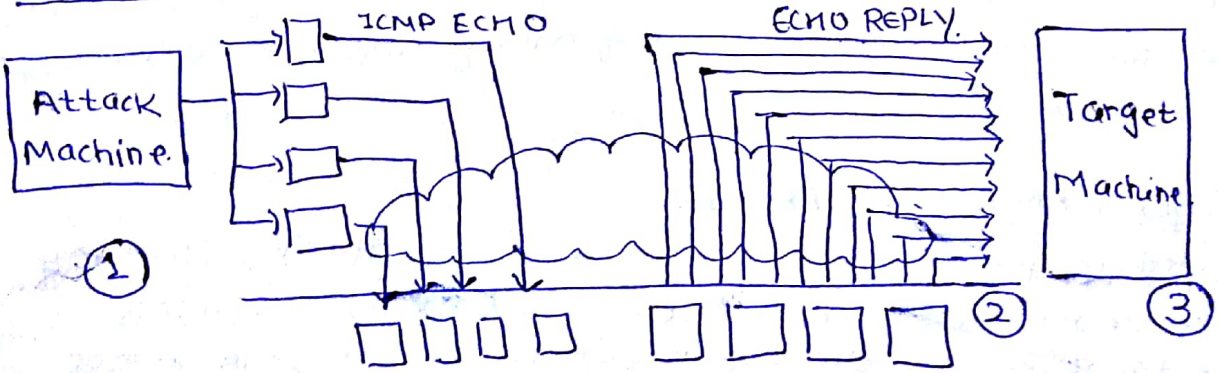
1. Internal host resource attack.
2. Attack that consumes data transmission resources

(1) INTERNAL RESOURCE ATTACK (SYN Flood Attack)



1. The attacker takes control of the multiple hosts over the Internet, instructing them to contact the target web server
2. Slave hosts begin sending TCP/IP SYN (synchronize/initialization) packets with erroneous return IP address information, to target.
3. Each SYN packet is a request to open TCP connection. For each such packet, the web server responds with a SYN/ACK packet The web server becomes bogged down as more traffic floods in. It results that legitimate connections are denied while the victim machine is waiting to complete bogus connections.

(2) ATTACK THAT CONSUMES DATA TRANSMISSION RESOURCES

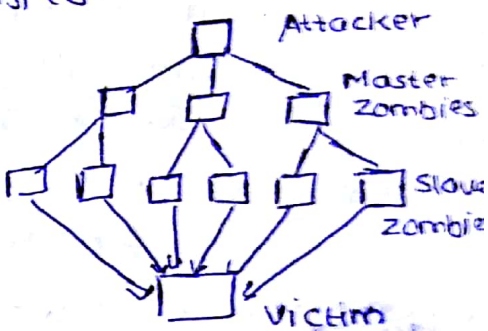


1. The attacker takes control of multiple hosts over internet, instructing them to send ICMP ECHO packets with target's spoofed IP address to group of hosts that acts as reflectors.
2. Nodes at bounce site receive multiple spoofed requests and respond by sending echo reply packets to target site.

3. The target's router is flooded with packets from bounce site, leaving no data transmission capacity for legitimate traffic.

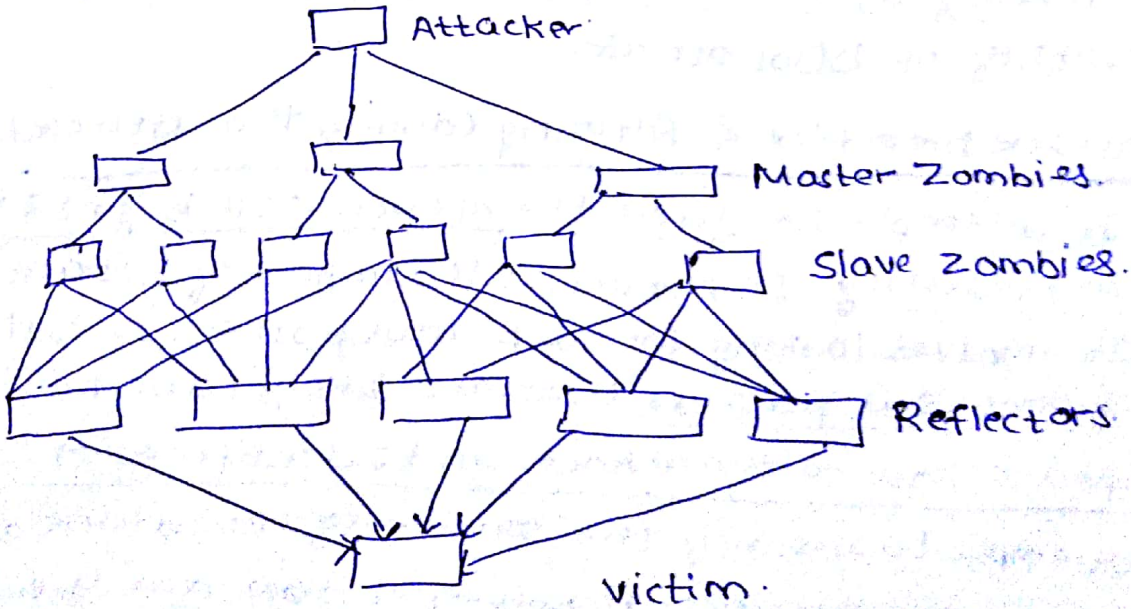
- **DIRECT DDoS**

The attacker implants zombie software on a no. of sites distributed throughout the internet
Direct DDoS involves two level of zombie machines: ~~master~~ master zombies and slave zombies.
The attacker coordinates and triggers the master zombies, which in turn coordinate and trigger the slave zombies. The two level of zombies makes it more difficult to trace the attack back to its source.



- **REFLECTOR DDOS**

It adds another layer of machines known as reflectors.
The slave zombies constructs packets requiring a response.
These packets are sent to uninfected machines i,e reflectors.
The reflectors respond with packets directed at the target machine.
A Reflector DDoS attack can easily involve more machines and more traffic than a direct DDoS attack and hence can be more damaging.



To construct an Attacking Network, the attacker first scans out a number of vulnerable machine and infects them. Then, the Zombie software that is installed in infected machine repeats the same scanning process, until a large distributed network of infected machines is created.

Scanning strategies:

- Random - Randomly hits any machine in network producing high volume of Internet traffic.
- Hit-List - Attacker compiles a list of vulnerable machines; This can be a slow process. Once the list is compiled attacker begins infecting machines on list. Each machine on the list is given a portion of list to scan.
- Topological - Info contained on an infected victim is used to find more hosts to scan.
- Local Subnet - Hosts looks for target in its own local network. The host uses the subnet address to find other hosts in network.

• DDOS COUNTERMEASURES

(1) Attack Prevention & Preemption (Before Attack)

It enables victim to endure attack attempts without denying service to legitimate clients.

It includes enforcing policies for resource consumption and providing backup resources available on demand.

It modify systems and protocols on Internet to reduce the possiblity of DDos attacks.

(2) Attack Detection & Filtering (during the attack)

It attempts to detect the attack as it begins & respond immediately. It minimises the impact of attack on target. It involves looking for suspicious pattern of behaviour & filters out packets that are likely to be part of attack.

(3) Attack source traceback and Identification (After attack)

It attempts to identify the source of the attack as a step to prevent future attacks. However, it does not yield results fast enough, if at all, to mitigate an ongoing task.

- **SUBSTITUTION BOXES**

S Box is a basic component of symmetric key algorithm which performs substitution. In block cipher, they are typically used to depict relationship between the key and ciphertext.

S Box takes m no. of input bits and transforms them into n no. of output bits where n is not necessarily equal to m.

An mxn s-box can be implemented as lookup table with 2^m words of n bits each.

• HASH FUNCTIONS

A hash function is a function which takes an input (or message) and returns a fixed sized alphanumeric string known as a hash value. A hash value h is generated by function H as:

$$h = H(M)$$

where h = hash value (message digest / checksum)
M = variable-length message
H(M) = Hash function

The hash value is appended to the message at the source when the message is assumed to be correct. The receiver authenticates the message by recomputing the hash value. It acts as a 'fingerprint' of a file, message or other block of data.

A hash function must have following properties:

1. H can be applied to a block of data of any size.
2. H produces fixed-length output.
3. H(x) is relatively easy to compute for any given x.
4. One way Property - For any given value h, it is computationally infeasible to find x such that H(x) = h.
5. Weak collision Resistance - For any given block x, it is computationally infeasible to find $y \neq x$ such that H(x) = H(y).
6. Strong collision Resistance - It is computationally infeasible to find any pair (x, y) such that H(x) = H(y)

i.e Two different messages cannot have same hash values.

The input (message, file etc) is viewed as a sequence of n-bit blocks. The I/P is processed one block at a time in an iterative fashion to produce n-bit hash function.

One of the simplest hash function is the bit-by-bit exclusive OR (XOR) of every block.

$$C_i = b_{i1} \oplus b_{i2} \oplus \cdots \oplus b_{im}$$

C_i = ith bit of the hash code.
m = no. of n bit blocks in I/P.
b_{ij} = ith bit in jth block.
$\oplus$ = XOR operation.

- SHA-512 Logic
- SECURITY OF HASH FUNCTIONS

(1) BRUTE FORCE ATTACK

The strength of hash function against brute-force attacks depends solely on the length of the hash code produced by the algorithm.

For a hash code of length n, the level of efforts required is:

One Way	2^n
Weak Collision Resistance	2^n
Strong Collision Resistance	$2^{n/2}$

(2) CRYPTANAYSIS

Cryptanalysis attacks on hash function algo to exploit some property of algorithm to perform some attack other than exhaustive search.

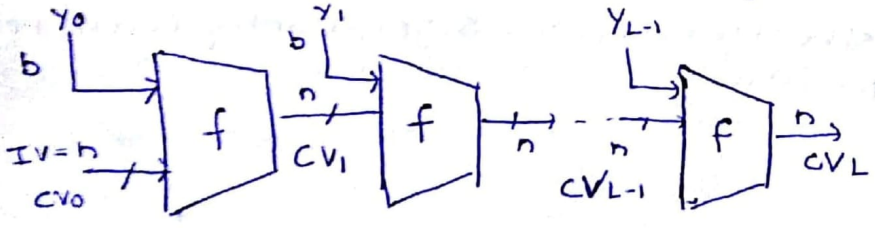
Secure Hash Function Structure

→ Hash function takes an I/P message & partitions it into L fixed sized block of b bits each. (If needed, final block is padded to b bits)

→ Hash algo involves repeated use of compression function f, that takes 2 inputs & produces n bit O/P
- n bit I/P from previous step, chaining variable
- n bit block

→ At the start of hashing, the chaining variable has a fixed value that is specified as a part of algorithm.

→ The final value of chaining variable is the hash value



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

Cryptanalysis of hash functions focuses on the internal structure of f and is based on attempts to find efficient techniques for producing calisions for single execution of f. Once that is done, attacker saves the value of fixed IV.

- **SECURE HASH ALGORITHM**

The secure hash algorithms are family of cryptographic hash functions developed by National Institute of standards and Technology (NIST). It comprises of 4 SHA algorithms i,e SHA-0, SHA-1, SHA-2 and SHA-3.

- SHA-0 → It is a 160 bit hash function, was published by NIST in 1993. It had few weeknesses and did not become very popular and was replaced by SHA-1.

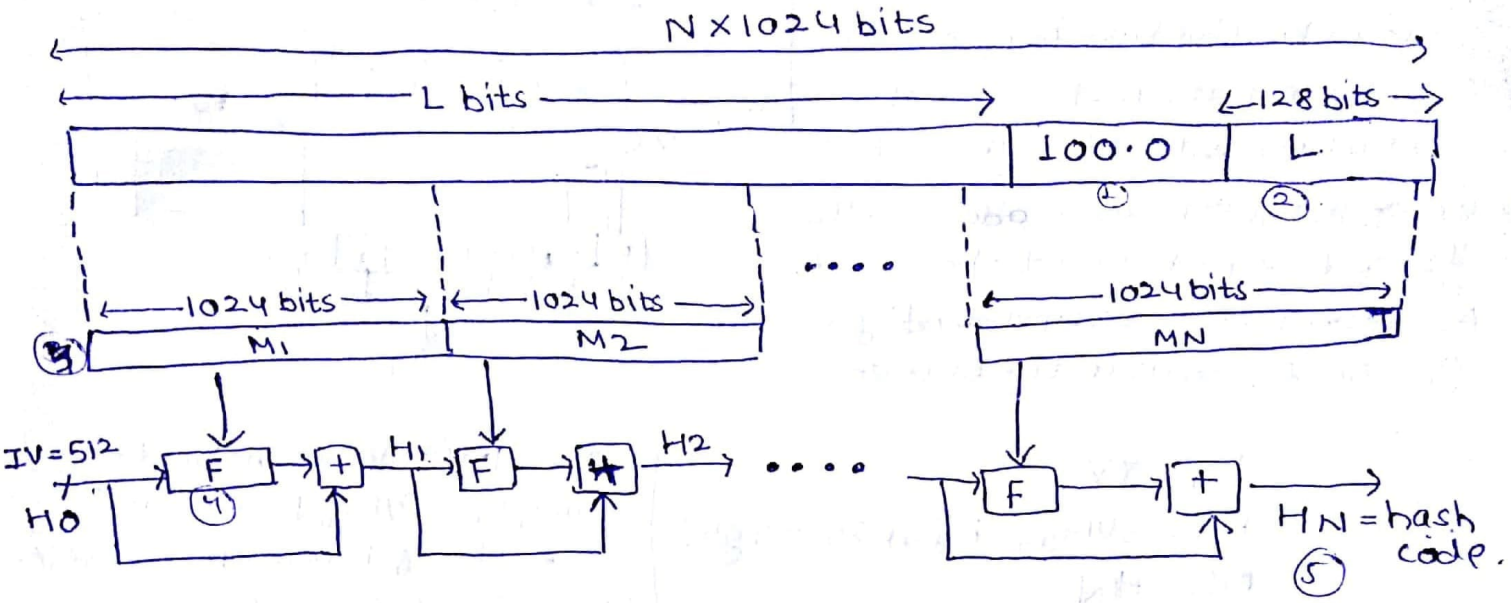
- SHA-1 → It is a 160 bit hash function and is employed in several widely used applications and protocols including secure Socket Layer (SSL) security. weaknesses were discovered in SHA-1 and it is no longer used after 2010.

- SHA-2 → It has 4 variants - SHA-224, SHA-256, SHA-384, SHA-512 depending up on no. of bits in their hash value. It is a strong hash function and uses basic design of SHA-1. No successful attacks have yet been reported on SHA-2 hash function.

- SHA-3 → A hash function known as **Keccak** is chosen by NSIT as the new SHA-3 standard. Keccak offers many benefits such as efficient performance and good resistance for attacks.
It supports same hash lengths as SHA-2 and its internal structure differs significantly from rest of the SHA-family.

• SHA-512 Logic

This algorithm takes an I/P a message with a max. length of less than 2^{128} bits and produces O/P a 512-bit message digest. The input is processed in 1024-bit blocks.



Step 1 : Append padding bits

The message is padded so that its length is congruent to 896 module 1024. Padding is always added, no. of padding bits is in the range of 1 to 1024. Single 1 bit is followed by necessary no. of 0 bits.

Step 2 : Append length.

A block of 128 bits is appended to the message This block is treated as an unsigned 128 bit integer and contains length of message before the padding.

Step 3 : Initialize hash buffer.

A 512-bit buffer is used to hold intermediate and final results of hash function. The buffer can be represented as 8 64-bit registers.

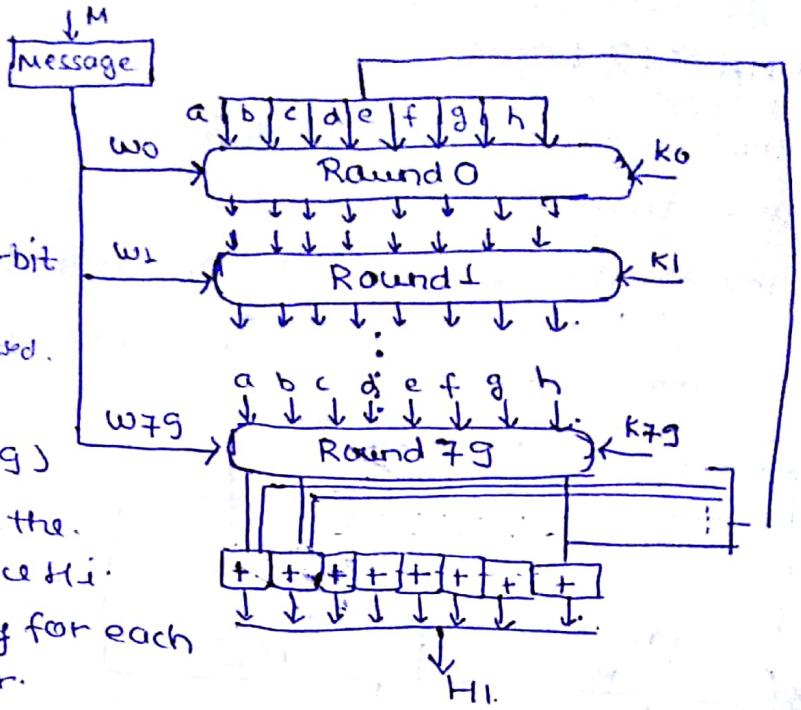
Step 4 : Processing message in 1024 bits blocks.

The module F consists of 80 rounds. and works as follow ⇒

Step 5 : Output.

After all N 1024-bit blocks have been processed, the O/P from the Nth stage is 512-bit message digest.

1. Each round takes I/P the 512-bit buffer value abcdefgh and updates the content of buffer.
2. Each round t makes use of 64-bit value W_t derivered from the current 1024 bit block being processed.
3. Each round also makes use of an additive constant K_t $(0 \le t \le 79)$
4. O/P of the 80th round is added to the I/P of the 1st round to produce H_i. Addition is done independently for each of the 8 words in the buffer.



$H_0 = IV$

$H_i = SUM_{64}(H_{i-1}, abcdefgh_i)$

$MD = H_N$.

IV → intial value of buffer.
$abcdefgh_i$ → O/P of last round
N → No of blocks in message.
MD → Final digest value.

- **Message Digest (MD)**

The MD family comprises of hash functions MD2, MD4, MD5 and MD6.

MD5 digests have been widely used to provide assurance about integrity of transferred file (e.g checksum for the files).

It produces 128-bit hash value.

• KERBEROS

Kerberos is an authentication service designed to use in distributed environment. It makes use of trusted third-party authentication service that enables clients and servers to establish authenticated communication.

The problem that Kerberos addresses is:
Assume an open distributed environment in which clients wish to access services on servers that are distributed throughout the network.
The servers should be able to restrict access to authorized users and should be able to authenticate requests for services.
So, rather than building authentication protocols at each server, Kerberos provides a centralized authentication server whose function is to authenticate users to servers and servers to users. It relies exclusively on symmetric encryption, making no use of public-key encryption.

Requirements of Kerberos

1. **Secure**: A network eavesdropper should not be able to obtain necessary information. Kerberos should be strong enough that no weak link can be found.
2. **Reliable**: Kerberos should be highly reliable, as are the services that rely on Kerberos for access control needs. Kerberos for supported services. It should employ a distributed server architecture with one system to back up another.
3. **Transparent**: User should not be aware that authentication is taking place, beyond the requirement to enter password.
4. **Scalable**: The system should be capable of supporting large no. of clients and servers.

Page 409 → Kerberos Relam.

• PRETTY GOOD PRIVACY (PGP)

PGP is an open-source freely available software package for e-mail security. It provides

1. **Authentication** – through the use of digital signature
2. **Confidentiality** – through the use of symmetric block encryption
3. **Compression** – Using the ZIP algorithm
4. **Email compatibility** – using the radix-64 encoding scheme
5. **Segmentation & reassembly** – to accomodate long e-mails.

PGP is used widely because

1. It is freely available and runs on a variety of platforms.
2. It is based on algorithms that are considered extremely secure.
3. It has a wide range of applicability
4. It is now on an Internet standards track.

(1) AUTHENTICATION

PGP provides digital signature service. The sequence is as follow

1. Sender creates a message.
2. SHA-1 generates 160 bit hash code of the message.
3. Hash code is encrypted with RSA using sender's private key.
4. Receiver uses RSA with sender's public key to decrypt and recover the hash code.
5. Receiver generates a new hash code for the message and compares with the decrypted one. If the two match, the message is accepted as authentic.

Because of RSA – receiver is assured that authenticated machine can generate digital signature.

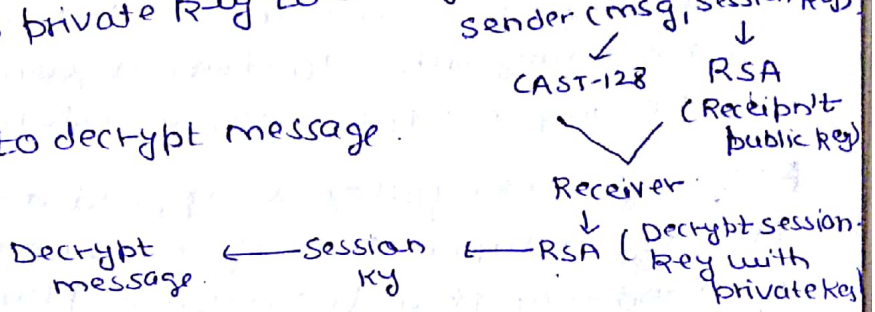
Because of SHA-1 – receiver is assured that no one else could generate a new message with same hash code.

Detached Signature – It may be stored and transmitted separately from the message it signs.

(2) CONFIDENTIALITY

It is provided by encrypting messages to be transmitted or stored locally as files. Encryption algorithm CAST-128 is used.

1. The sender generates a message and a random 128 bit no which is used as a session key for this message only.
2. Message is encrypted using CAST-128 with the session key.
3. session key is encrypted with RSA using receipnt's public key.
4. Receiver uses RSA with its private key to decrypt & recover the session key
5. The Session key is used to decrypt message.



AUTHENTICATION + CONFIDENTIALITY

Signature is generated.
↓
Message + Signature encrypted using CAST-128
↓
Session key is encrypted using RSA.

Sender signs the message with its own private key.
Encrypts the message with session key.
Encrypts the session key with receipnt's public key.

AUTHENTICATION

Sender (message).
↓
SHA-1 → 160 bit hash code.
↓
Hash code is encrypted using RSA sender's private key.
↓
Receiver uses sender's public key to decrypt hash code.
↓
Matches hash code.

CONFIDENTIALITY

Sender (message, session key).
↓
Message encrypted using session key. CAST-128
↓
Session key is encrypted with RSA using Receipnt's public key.
↓
Receiver uses RSA with its private key to decrypt session key
↓
Session key is used to decrypt message.

(3) COMPRESSION

PGP compresses the message after applying the signature but before encryption. It saves space for both e-mail transmission and for file storage.

Signature is created before compression because:

1. One can store only the uncompressed message together with the signature for the future verification.
   If signature is done after compression, then it would be necessary either to store a compressed message or to recompress the message when verification required.

Message encryption is applied after compression to strengthen the cryptographic security because compressed message has less redundancy than the orginal plaintext and hence cryptanalysis is difficult.

(4) COMPATIBILITY

The resulting encrypted blocks consists of a stream of 8 bit octet
Many electronic mail systems only permit the use of blocks consisting of ASCII text.
To accomdate this restriction, PGP provides service of converting the 8-bit binary stream to a stream of ASCII characters.

Radix-64 conversion technique is used for this conversion.
3 octets of binary data is mapped into 4 ASCII characters.

1. Signature is created using hash code of the uncompressed plaintext
2. Plain text + signature is compressed.
3. compressed text + signature is encrypted and added with public-key-encrypted symmetric encryption key.
4. Finally, the entire block is converted to radix-64 format.

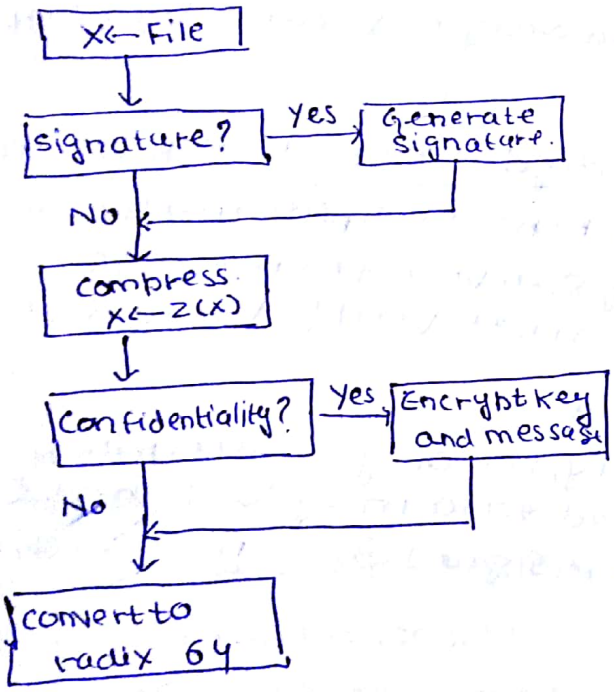
1. On reception, the incoming block is converted back from radix-64 format to binary.
2. Recover the session key and decrypts the message.
3. Decompress the message.
4. Recovers the transmitted hash code and matches with its own calculated hash code.

⑤ SEGMENTATION AND REASSEMBLY

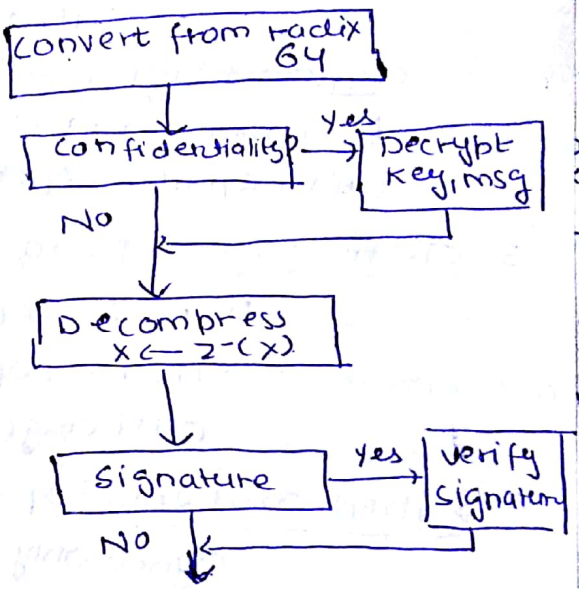
Email facilities often are restricted to maximum message length (e.g max. length of 50,000 octets). Any message longer than that must be broken up into smaller segments each of which is mailed separately.

PGP automatically subdivides a message that is too large into segments that are small enough to send via e-mail.
segmentation is done after all other processes, including the radix 64-conversion.

At receiving end, PGP must strip off all e-mail headers and reassemble the entire orginal block.



Generatic Transmission Diagram



Generic Reception Diagram

- **S/MIME** (Secure/Multipurpose Internet Mail Extension)

It is a security enhancement to MIME Internet e-mail format standard.

In terms of general functionality, S/MIME is very similar to PGP. Both offers ability to sign and encrypt messages.

FUNCTIONS OF S/MIME

1. Enveloped Data - It consists of encrypted content and the encryption keys for one or more receipts.
2. Signed Data - Digital signature is formed by taking message digest of the content to be signed & then encrypting with signer's private key.

Content + signature are encoded using base64 encoding. A signed data message can only be viewed by recipient with S/MIME capability.

3. Clear-signed Data - only digital signature is encoded using base64 and hence receipts without S/MIME capability can view the message, although they cannot verify the signature.
4. Signed and enveloped Data

Signed only and encrypted only entities may be nested so that encrypted data may be signed & signed data / clear signed data may be encrypted

• SIDE CHANNEL ATTACK

It is a attack based on the information gained from the physical implementation of cryptosystem, rather than brute force or theoretical weaknesses in algorithm

Side-channel attacks monitor power consumption and electromagnetic emissions while a device is performing cryptographic operations.

These attacks are relatively simple and inexpensive to execute.

Some side-channel attacks require technical knowledge of the internal operation of the system on which cryptography is implemented.

- Cache attack - attacker monitors cache accesses made by victim
- Timing attack - attacks based on how much time various computations take.
- Power monitoring attack - use varying power consumption by hardware.
- Electromagnetic attack - based on leaked electromagnetic radiation, which can provide plaintexts and other information
- Acoustic attack - exploits sound produced during computation.

• Countermeasures

1. Eliminate / reduce the release of information.
2. Eliminate relationship b/w the leaked information & secret data. ie make the leaked info unrelated to the secret data through some form of randomization of the ciphertext that transforms the data in a way that can be undone after the cryptographic operation is completed.