

some key bits. Ideally, that correlation should be absolute with respect to the key bits, since there is only one key to solve for, but it can be probabilistic with respect to the input and output bits, since there need to be many pairs to test.

As the number of rounds increases, though, the simple correlations disappear. Differential cryptanalysis represents an approach to finding more subtle correlations.

Instead of saying "if this bit is 1 in the input, then that bit will be 0 (or 1) in the output", we say "changing this bit in the input changes (or does not change) that bit in the output".

In fact, however, a complete, pattern of which bits change and do not change in the input and in the output is the subject of differential cryptanalysis. The basic principle of differential cryptanalysis, in its classic form, is this: the cipher being attacked has a *characteristic* if there exists a constant X such that given many pairs of plaintexts A, B , such that $B = A \oplus X$, if a certain statement is true about the key, $E(B, K) = E(A, K) \oplus Y$ for some constant Y will be true with a probability somewhat above that given by random chance.

Linear Cryptanalysis: Linear cryptanalysis, invented by Mitsuru Matsui, is a different, but related technique. Instead of looking for isolated points at which a block cipher behaves like something simpler, it involves trying to create a simpler approximation to the block cipher as a whole.

For a great many plaintext-ciphertext pairs, the key that would produce that pair from the simplified cipher is found, and key bits which tend to be favored are likely to have the value of the corresponding bit of the key for the real cipher. The principle is a bit like the summation of many one-dimensional scans to produce a two-dimensional slice through an object in computer-assisted tomography.

Prob.7 What is Cryptanalysis? Explain stream ciphers and block ciphers with suitable example.

[R.T.U. 2018]

Sol. Cryptanalysis: Cryptanalysis is the art of decrypting or obtaining plain text from hidden messages over an insecure channel. It is also known as code cracking.

Cryptanalysis is the art which is associated with decryption of a cipher text into plain text. An unauthorized person tries to decrypt the message by eavesdropping into the insecure channel. It is also known as code breaking. This person is not bounded by any of the rules. He may use any of the methods in order to get the plain text. In this case, the person is now aware with the proper keys, and thus uses one or many of the cryptanalytic techniques. Some of the techniques are –

B.Tech. (IV Sem.) CS Solved Papers

- Cipher text-only attack – In this case, the attacker has only the cipher text to reach plaintext, and thus he makes guess about the plaintext.
- Known-plaintext attack – In this case, the attacker tries to guess the plaintext by analyzing some part of the cipher text.
- Chosen-plaintext attack – The cryptanalyst can choose plaintexts and obtain their corresponding ciphertexts. The aim is to choose the plaintexts such that the resulting pairs of plaintext and cipher texts makes easy for deducing the encryption key.
- Man in the middle attack – The person will intercept the signals sent by sender and receiver. He will pose to them as the other party and will exchange keys with both of them separately.

Block Cipher: The block cipher uses a deterministic algorithm that conducts operations on fixed-length groupings of bits, or blocks. By using a transformation specified by a symmetric key, a block cipher is able to encrypt bulk data, and is one of the basic components of many cryptographic protocols in use today. The basic idea of a block cipher is to divide text in relatively large blocks, typically 64 or 128 bits long, and encode each block separately. The same encryption key is used for each block and it is the encryption key that determines the order in which substitution, transportation and other mathematical functions are performed on each block. Strong algorithms means that reverse engineering the cipher, or determining which functions were performed on each block, in which order, are virtually impossible.

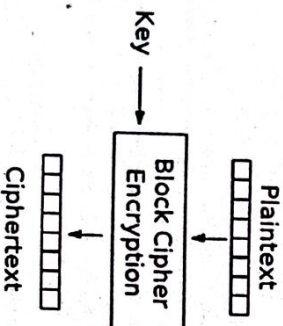


Fig. 1

Stream Cipher: A stream cipher, on the other hand, takes plaintext characters or digits and combines them with a pseudo random cipher digit stream, or key stream. The basic idea of a stream cipher is to divide text into small blocks, one bit or one byte long, and encode each block depending on many previous blocks. Stream ciphers use a different encryption key (a value which must be fed into the algorithm) for each bit or byte, so the same bit or byte produces different cipher

Information Security System

text each time it is encrypted. Some stream ciphers use a key stream generator, which produces a random, or nearly random, stream of bits. The cipher performs a Boolean operation, known as an exclusive OR, between the bits in the key stream and the bits in the plaintext to produce cipher text.

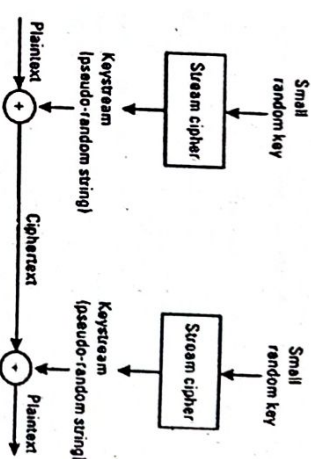


Fig. 2

Prob.8 Encrypt the message "secret message" using Vigenere cipher with the key KLAS (hint: Vigenere cipher uses repetitive key stream to encrypt a message.)

[R.T.U. 2017]

Sol. The Vigenere Cipher: This cipher was long thought to be unbreakable because, unlike the Caesar cipher, there is no simple one-to-one mapping of the plaintext to the cipher alphabet.

The Vigenere cipher is applied by utilizing a series of different Caesar ciphers based on the letters of a particular keyword. The ordinal position of each letter in the keyword is what determines the shift parameter values of the various Caesar ciphers used. The resulting cipher alphabets are then applied in sequence and then repeated to encode the letters in the message.

An example should make the process a bit clearer.

If we use the word KLAS as the keyword, we can see that the shift values are 0, 10, 11 and 19. Note that the letter A produces no shift since its value is 0.

Let's use the Vigenere Cipher to encrypt this message with the keyword KLAS:

Shift 10 (Letter K):

Plain alphabet

: ABCDEFGHIJKLMNOPQR

STUVWXYZ

Cipher alphabet : LMNOPQRSTUVWXYZA

BCDEFGHIJ

Shift 11 (Letter L):

Plain alphabet : ABCDEFGHIJKLMNOPQR

STUVWXYZ

Cipher alphabet : LMNOPQRSTUVWXYZAB

CDEFGHIJK

Shift 0 (Letter A):

Plain alphabet : ABCDEFGHIJKLMNOPQR

STUVWXYZ

Cipher alphabet : ABCDEFGHIJKLMNOPQR

STUVWXYZ

Shift 19 (Letter S):

Plain alphabet : ABCDEFGHIJKLMNOPQR

STUVWXYZ

Cipher alphabet : STUVWXYZABCDEFGHIJ

KLMNOPQR

One trick we can use to facilitate knowing when to apply which cipher alphabet is to repeat the keyword above the encoded message, like so:

Keyword: KLASKLASKLASK

Message: secretmessage

Using these four cipher alphabets in circular sequence will produce the following encrypted message:

CPCIOOE MWCDAYO

Prob.9 Explain the key principle of security with suitable example.

[R.T.U. 2016]

Sol. Key Principle of Security:

1. Confidentiality

(i) Confidentiality is probably the most common aspect of information security. The principle of confidentiality specifies that only the sender and intended recipient should be able to access the contents of a message.

(ii) Confidentiality gets compromised if an unauthorized person is able to access a message. Protection of confidential information is needed. An organization needs to guard against those malicious actions to endanger the confidentiality of its information.

(iii) Example: Banking customers accounts need to be kept secret. Confidentiality not only applies to the transmission of information but also applies to the transmission of information. When we send a piece of the information to be stored in a remote computer or when we retrieve a piece of information from a remote computer we need to conceal it during transmission. Interception causes loss of message confidentiality.

2. Integrity

- (i) Information needs to be changed constantly. Integrity means that changes need to be done only by authorized entities and through authorized mechanisms. When the contents of a message are changed after the sender sends it, before it reaches the intended recipient it is said that integrity of the message is lost.
- (ii) Integrity violation is not necessarily the result of a malicious act; an interruption in the system such as a power surge may also create unwanted changes in some information.
- (iii) Modification causes loss of message integrity.
3. Availability
- (i) The principle of availability states that resources should be available to authorized parties at all times. The information created and stored by an organization needs to be available to authorized entities. Information is useless if it is not available.
- (ii) means it must be accessible to authorized entities. The unavailability of information is just as harmful for an organization as the lack of confidentiality or integrity.
- (iii) Example: The situation can be difficult for a bank if the customer could not access their accounts for transactions.
- (iv) Interruption puts the availability of resources in danger.

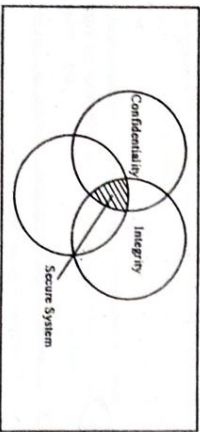


Fig.

- (v) The diagram above explains the balance concept.
- (vi) The right balance of the three goals is needed to build a secure system. If the goals are not balanced then a small hole is created for attackers to nullify the other objectives of security. Having a highly confidential system but low availability then the system is not secure.
- (vii) Example: A system can protect confidentiality and integrity but if the resource is not available the other two goals also are of no use.

Prob.10 What is cryptography? Explain Block and stream ciphers in detail.

[R.T.U. Dec. 2015]

[B.Tech. (IT) Sem.II CS Related Papers]

Sol. Cryptography is a method of storing and transmitting data in a particular form so that only those for whom it is intended can read and process it.

Cryptography includes techniques such as microdots, merging words with images, and other ways to hide information in storage or transit. However, in today's computer-centric world, cryptography is most often associated with scrambling plain text (ordinary text, sometimes referred to as clear text) into cipher text (a process called encryption), then back again (known as decryption).

Cryptography concerns itself with the following four objectives:

1. **Confidentiality:** The information cannot be understood by anyone for whom it was unintended.
 2. **Integrity:** The information cannot be altered in storage or transit between sender and intended receiver without the alteration being detected.
 3. **Non-repudiation:** The creator/sender of the information cannot deny at a later stage his or her intentions in the creation or transmission of the information.
 4. **Authentication:** The sender and receiver can confirm each other's identity and the origin/destination of the information.
- The two most common types of encryption algorithm used in modern cryptography are the block and stream ciphers.
- Block Cipher :** Refer to Prob.7.
- Stream Cipher :** Refer to Prob.7.

Prob.11 Differentiate amongst cryptology, cryptography and cryptanalysis.

[R.T.U. Dec. 2015]

Sol. **Cryptography:** Cryptography is the art of hiding messages by converting them into hidden texts. It is generally done in order to transmit a message over insecure channels. Cryptography converts a plaintext (message to be communicated) to a cipher text message by employing techniques of encryption. The process of obtaining a cipher text from a plaintext is referred to as encryption. This art of cryptography is an ancient art.

Generally, there are three different cryptographic schemes which are used widely -

- Secret key or Symmetric Cryptography – it uses same key for encryption and decryption
 - Public-key or Asymmetric Cryptography – it uses one key for encryption and another for decryption
 - Hash functions – it makes use of a mathematical transformation to encrypt the information in an irreversible manner.
- Cryptanalysis:** Refer to Prob.7.
- Cryptology:** Cryptology is the study of the mathematics behind encryption/decryption.

Information Security System

Cryptography deals with studying the math behind the cipher and way the messages are encrypted or decrypted. Therefore, in short, Cryptology is the mathematical study of cryptography and cryptanalysis.

Prob.12 Write short note on key distribution in symmetric encryption.

[R.T.U. 2015]

Sol. **Symmetric Encryption :** Symmetric encryption is an encryption algorithm where the same key is used for both encryption and decryption. The key must be kept secret, and is shared by the message sender and recipient.

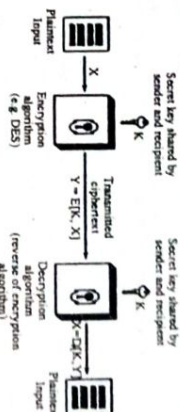


Fig. : Simplified Model of Symmetric Encryption

Symmetric encryption is a two-way algorithm because the mathematical algorithm is reversed when decrypting the message along with using the same secret key. Symmetric encryption is also known as private key encryption or secure key encryption.

Key Distribution in Symmetric Encryption

"Sender and receiver must have obtained copies of the secret key in a secure way and must keep the key secure."

There are two techniques to distribute secret keys :

(i) Key Distribution by Manual Delivery

For two parties A and B:

- A key could be created by A and delivered physically to B (or vice versa).
- A key could be created by the third trusted party C and delivered physically to A and B.

Difficult to use in wide area distributed system, when dynamic connections are needed.

(ii) Key Distribution by Further Techniques

- If A and B have used recently a secret key, one of them could create a new secret key and send it to the partner using old key.

Potential problem: Once the attacker learned one key, he can disclose all keys afterwards.

- There is a third trusted party C connected by encrypted channels with both A and B. Then C creates a key and distributes it among A and B using encrypted channels.

PART-C

Prob.13 What is substitution ciphers. Explain.

[R.T.U. 2018]

Sol. Substitution technique

Caesar Cipher

The scheme was first proposed by Julius Caesar, and is termed, as Caesar Cipher. It was the first example of substitution cipher, in the substitution cipher technique, the characters of a plain text message are replaced by other characters, numbers or symbols. Caesar Cipher is a special case of substitution technique, wherein each alphabet in a message is replaced by an alphabet three places down the line. For instance, using the Caesar Cipher, the plain text ATUL will become cipher text DWXC.

Clearly, the Caesar Cipher is a very weak scheme of hiding plain text messages. All that is required to break the Caesar Cipher is to do the reverse of the Caesar Cipher process—i.e. replace each alphabet in a cipher text message produced by Caesar Cipher with the alphabet that is three places up the line. Thus, to work backwards, take a cipher text produced by Caesar Cipher, and replace each A with X, B with Y, C with Z, D with A, E with B and so on.

Modified Version of Caesar Cipher

Caesar Cipher is good in theory, but not so good in practice. Let us now try and complicate the Caesar Cipher to make an attacker's life difficult. How can we generalize Caesar Cipher a bit more? Let us assume that the cipher text alphabets corresponding to the original plain text alphabets may not necessarily be three places down the order, but instead, can be any places down the order. This can complicate matters a bit.

Thus, we are now saying that an alphabet A in plain text would not necessarily be replaced by D. It can be replaced by any valid alphabet, i.e., by E or by F or by G, and so on. Once the replacement scheme is decided, it would be constant and will be used for all other alphabets in that message. As we know, the english language contains 26 alphabets. Thus, an alphabet A can be replaced by any other alphabet, in the english alphabet set, (i.e. B through Z). Of course, it does not make sense to replace an alphabet by itself (i.e. replacing A with A). Thus, for each alphabet, we have 25 possibilities of replacement.

The major weakness of the Caesar Cipher is its predictability. Once we decide to replace an alphabet in a plain text message with an alphabet that is k positions up or down the order, we replace all other alphabets in the plain text message with the same technique. Thus, the cryptanalyst has to try out a maximum of 25 possible attacks, and she is assured of a success.

Now imagine that rather than using a uniform scheme for all the alphabets in a given plain text message, we decide to use random substitution. This means that in a given plain text message, each A can be replaced by any other alphabet (B through Z), each B can also be replaced by any other random alphabet (A or C through Z), and so on. The crucial difference being, there is no relation between the replacement of B and replacement of A. That is, if we have decided to replace each A with D, we need not necessarily replace each B with E—we can replace each B with any other character.

To put it mathematically, we can now have any permutation or combination of the 26 alphabets, which means $(26 \times 25 \times 24 \times 23 \times \dots \times 2)$ or 4×10^{26} possibilities. This is extremely hard to crack. It might actually take years to try out these many combinations even with the most modern computers.

Homophonic Substitution Cipher

The Homophonic Substitution Cipher is very similar to Mono-alphabetic Cipher. Like a plain substitution cipher technique, we replace one alphabet with another in this scheme. However, the difference between the two techniques is that the replacement alphabet set in case of the simple substitution techniques is fixed (e.g. replace A with D, B with E, etc.), whereas in the case of Homophonic Substitution Cipher, one plain text alphabet can map to more than one cipher text alphabet. For instance, A can be replaced by D, H, P, R, B can be replaced by E, I, Q, S, etc.

Polygram Substitution Cipher

In Polygram Substitution Cipher technique, rather than replacing one plain text alphabet with one cipher text alphabet at a time, a block of alphabets is replaced with another block. For instance, HELLO could be replaced by YUQOW, but HELL could be replaced by a totally different cipher text block TEUI, as shown in Fig. This is true in spite the first four characters of the two blocks of test (HELL) being the same. This shows that, in Polygram Substitution Cipher, the replacement of plain text happens block-by-block, rather than character-by-character.

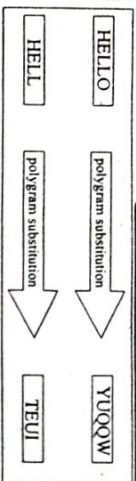


Fig. : Polygram Substitution

Polyalphabetic Substitution Cipher

Leon Battista invented the polyalphabetic Substitution Cipher in 1568. This cipher has been broken many times, and yet it has been used extensively. The Vigenere Cipher and the Beaufort Cipher are examples of Polyalphabetic Substitution Cipher.

This cipher uses multiple one-character keys. Each of the keys encrypts one plain text character. The first key encrypts the first plain text character, the second key encrypts the second plain text character, and so on. After all the keys are used, they are recycled. Thus, if we have 30 one-letter keys, every 30th character in the plain text would be replaced with the same key. This number (in this case, 30) called as the period of the cipher.

Prob.14 Explain the various security policies, attacks, mechanism. [8, R.U. 2017]

Sol. Security Policies : Information security policy is a set of policies issued by an organization to ensure that all information technology users within the domain of the organization or its networks comply with rules and guidelines related to the security of the information stored digitally at any point in the network or within the organization's boundaries of authority.

1. The evolution of computer networks has made the sharing of information ever more prevalent.
2. Every organization needs to protect its data and also control how it should be distributed both within and without the organizational boundaries.
3. A business might employ an information security policy to protect its digital assets and intellectual rights in efforts to prevent theft of industrial secrets and information that could benefit competitors.
4. A typical security policy might be hierarchical and apply differently depending on whom they apply to.

Attacks

Active Attack and Passive Attack : Security attacks can be categorized into two types :

1. Passive threats (Attacks)
2. Active threats (Attacks)

Information Security System

1. Passive Attacks : Passive attacks are in the nature of eavesdropping on, or monitoring of transmissions. The goal of opponents is to obtain information that is being transmitted. Two types of passive attacks are :

- (a) Release of message contents and
- (b) Traffic analysis.

(a) The release of message contents is easily understood. A telephone conversation, an electronic mail message and a transferred file may contain sensitive or confidential information. It is preferred to prevent the opponent from bearing the contents of these transmissions.

(b) The second passive attack, traffic analysis, is more subtle. If we had a way to mask the contents of messages or other information traffic so that opponent, even if they captured the message, could not extract the information from the message. The common techniques for masking contents is encryption. If we had encryption protection in place, an opponent might still be able to observe the pattern of these messages. The opponent could determine the location and identity of communicating hosts and could observe the frequency and length of messages being exchanged.

This information might be useful in guessing the nature of the communication that was taking place. Passive attacks are very difficult to detect because they do not involve any alteration of the data. However it is feasible to prevent the success of these attacks. Thus the emphasis in dealing with passive attacks is on prevention, rather than detection.

2. Active Attacks : This is the second major category of attacks, it involves some modifications of the data stream or the creation of a false stream and can be subdivided into four categories :

- (a) Masquerade
- (b) Replay
- (c) Modification of messages
- (d) Denial of service

(a) A masquerade takes place when one entity pretends to be a different entity. A masquerade attack usually includes one of the other forms of active attack e.g. authentication sequence has taken place, thus enabling an authorized entity with few privileges to obtain extra privileges by impersonating an entity that has those privileges.

(b) Replay involves the passive capture of a data unit and its subsequent retransmission to produce an unauthorized effect.

(c) Modification of messages simply means that some portion of a legitimate message is altered or those messages are delayed or recorded, to produce an unauthorized effect e.g. a message meaning "Allow Prakash Power to read confidential file accounts" is modified to mean "Allow Manish Bhargava to read confidential file accounts."

(d) The denial of service prevents or inhibits the normal use or management of communications facilities. This

attack may have a specific target e.g. the security audit service.

Another form of service denial is the disruption of entire network, either by disabling the network by overloading it with messages or as to degrade performance.

Active Attacks Vs Passive Attacks

Active attacks present the opposite characteristics of passive attacks. Whereas passive attacks are difficult to select, measure are available to prevent their success. On the other hand it is quite difficult to prevent active attacks absolutely because to do so would require complete protection of all communications facilities and paths at all times. The goal is to select the attacks and to recover from only delays or disruption cause by them. It should also contribute to prevention.

Types of passive attacks:

- Release of a message contents : contents of a message are read.
- A message may be carrying sensitive or confidential data.
- Traffic analysis : An intruder makes inferences by observing message patterns.
- Can be done even if messages are encrypted.
- Inferences: location and identity of hosts.

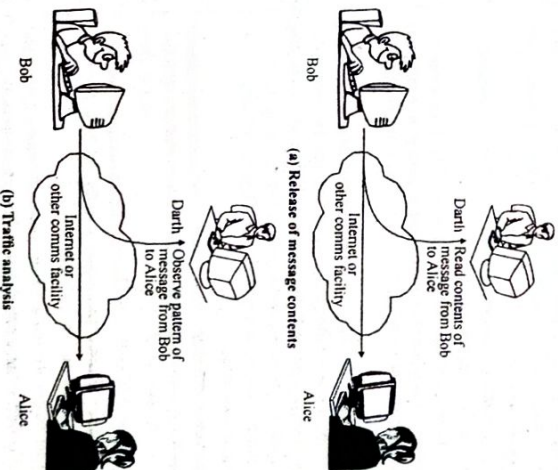


Fig. 1

An active attack is one in which the intruder may transmit messages, replay old messages, modify messages in transit, or delete selected messages from the wire.

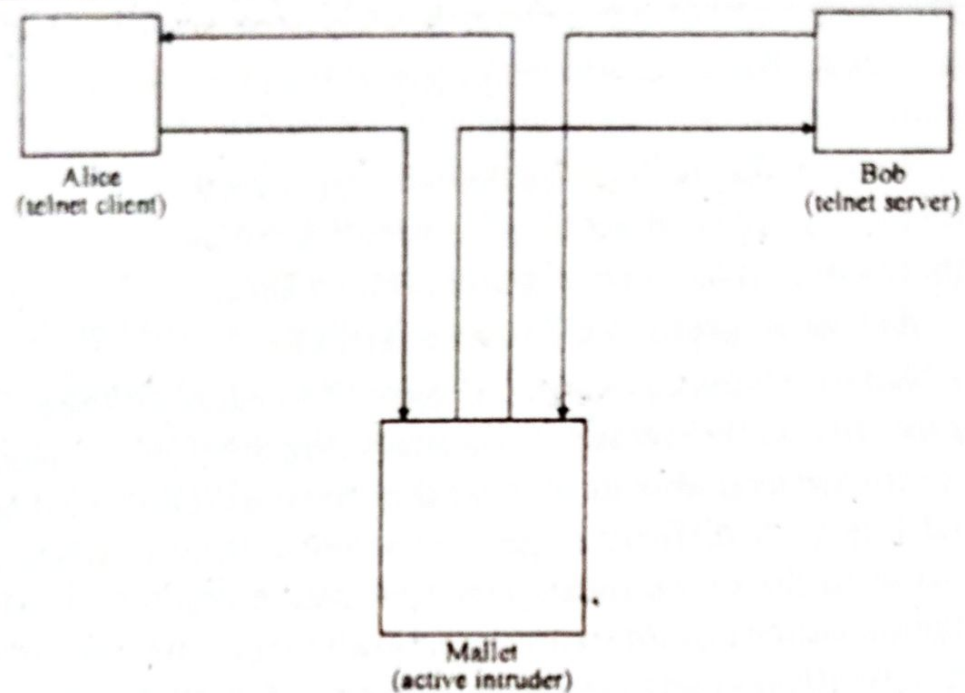


Fig. 2

For an active attack, the attacker needs to gain physical control of a portion of the link and be able to insert and capture transmissions (medium could be telephone twisted pair, coaxial cable, or optical fiber). See Fig. 2 to understand the concept of active attack.

• **Security Mechanism** : Security mechanism can be classified according to the specific protocol implementation. Some of the security mechanism is implemented into the particular protocol layer while others are not specific to particular protocol layer. Security mechanism process called encipherment.

According to security standard X.800 encipherment can be classified into two types :

1. Reversible Encipherment Mechanisms : A reversible encipherment mechanism is simply an encryption algorithm that allows data to be encrypted and subsequently decrypted.

2. Irreversible Encipherment Mechanisms : Irreversible encipherment mechanisms include hash algorithms and message authentication codes, which are used in digital signature and message authentication applications. It does not allow generating the original message from encrypted message.

MODERN BLOCK CIPHERS 2

YEARS QUESTIONS

- Replay attack of packets can be avoided using modes of operation.

PART-B

Prob.5 Explain construction of Balanced function for S-Box. [R.T.U. 2019]

OR
Write short note on Construction of balanced functions. [R.T.U. 2016]

Sol. Construction of Balanced Functions : If $w(g_{n-1}) = w(g_{n-1} \oplus h_{n-1})$, the function is balanced. Therefore, any functions on the subspace V_m can be used to construct balanced functions on V_n ($m < n$). The nonlinearity of function depends on the choice of both $g(x)$ and $h(x)$. Let $g_{n-1}(x)$ and $g_{n-1}(x) \oplus h_{n-1}(x)$ be two functions on V_{n-1} with Hamming weight $w(g)$ and $w(g \oplus h)$ respectively. Suppose the nonlinearity of $g_{n-1}(x)$ and $g_{n-1}(x) \oplus h_{n-1}(x)$ is N_g and $N_{g \oplus h}$ respectively. Then the function

$$f(x) = g_{n-1}(x) \oplus x_n h_{n-1}(x)$$

On V_n has nonlinearity

$$N_f \geq N_g + N_{g \oplus h}$$

Proof : Assume q_g and q_h are two affine functions over V_{n-1} such that $N_g = w(g \oplus q_g)$ and $N_{g \oplus h} = w(g \oplus h \oplus q_g \oplus q_h)$. It is noted that $w(g \oplus q_g)$ and $w(g \oplus h \oplus q_g \oplus q_h)$ are minimal hamming distances for the functions g and $g \oplus h$ with affine functions on V_{n-1} . For function $f \oplus q_g \oplus x_n q_h$ when $x_n = 0$ one has N_g and for $x_n = 1$ there are $N_{g \oplus h}$ is. So the function $f \oplus q_g \oplus x_n q_h$ has Hamming weight $N_g + N_{g \oplus h}$. Therefore for the function $f(x)$, the Hamming distance to an affine function ϕ on V_n always has

$$w(f \oplus \phi) \geq w(f \oplus q_g \oplus x_n q_h).$$

According to the definition of nonlinearity, one has $N_f \geq N_g + N_{g \oplus h}$. Then the equation is proven.

To construct a highly nonlinearity balanced function on V_n , two functions are chosen, $g_{n-1}(x)$ which has high nonlinearity and $h_{n-1}(x)$ which satisfies $w(g) = 2^{n-2} - w(g \oplus h)$, then the function (22) is balanced and has nonlinearity $N_f \geq N_g + N_{g \oplus h}$.

Prob.6 Write short note on propagation and non linearity of S-Box. [R.T.U. 2019]

OR
What do you understand by propagation and nonlinearity? Explain with example. [R.T.U. 2016]

OR
Explain the role of propagation and non-linearity in information theory. [R.T.U. 2018]

Sol. Propagation : Three of the most important criteria for cryptographically strong Boolean functions are the balancedness, the nonlinearity and the propagation criterion.

A Boolean function of n input coordinates is said to satisfy the propagation criterion with respect to a non-zero vector if complementing input coordinates according to the vector results in the output of the function being complemented 50% of the time over all possible input vectors, and to satisfy the propagation criterion of degree k if complementing or less input coordinates results in the output of the function being complemented 50% of the time over all possible input vectors.

Propagation Criterion PC(l) of Order k : Boolean functions satisfy PC(l) when one keeps constant a certain number k of coordinates x_1, \dots, x_n . This property of Boolean functions is called propagation criterion PC(l) of order.

Perfect Nonlinear Boolean Functions : The nonlinear Boolean functions on V_n satisfying PC(l) for all $0 < l \leq n$ are called perfect nonlinear Boolean functions, i.e. the functions satisfy PC(n).

Consider the variables x_1, \dots, x_n to be random uniformly distributed inputs and the values of variables are from GF(2). A function $f(x)$ from V_n to V_m (S-box) is said to be k -th order correlation immune if the probability distribution of the values $f(x)$ is unaltered when at most k of the coordinates x_1, \dots, x_n are kept constant. That function is k -resilient if it is balanced (which means all of the values in V_m occur equally often) and k -th order correlation immune.

Nonlinearity : The nonlinearity of a function

$$f: \{0,1\}^n \rightarrow \{0,1\} \text{ is}$$

$$n(f) = \min_{f \in A_n} w(f \oplus l)$$

where $w(l)$ denotes the Hamming weight of the function.

The nonlinearity of an S-box S is

$$n(S) = \min_{f \in C} n(f)$$

where C is the set of all nontrivial linear combinations of the columns of S .

Prob.7 Write and explain the design criterion of S-Box in detail. [R.T.U. 2019]

OR
Describe substitution box in detail. Why is it so important? [R.T.U. 2018]

OR
What is the purpose of the S-Boxes in DES? What is the criteria to design S-Boxes? [R.T.U. Dec 2015]

Sol. The S-boxes are the non-linear part of DES that makes it difficult to break the algorithm and secure against linear and differential cryptanalysis. The S-boxes provide the "confusion" of data and key values, whilst the permutation P then spreads this as widely as possible, so each S-box output affects as many S-box inputs in the next round as possible, giving "diffusion".

There are 8 S-boxes also known as the substitution boxes, is a table that consist of four rows and 16 columns with 64 entries all together. They take in 6-bits and produce or output 4-bits. That is, the 48-bits into 8 S-boxes will be 6-bits each. However the 6-bits are represented in binary form of say, 010100. The two outer bits (the first and the last bit) represents the row (one of the four rows) and the inner four bits represent the columns (one of the 16 columns). The cell where the row and the column meets represents the value in decimal of the output. This is then converted to binary as the output. From the example 010100, the first and last digits 00 = the row which is the first row (00, 01, 10, 11) and the inner four digits 1010 = the column. All 8 S-boxes will output 4-bits each in similar way and that is 32-bits output is then permuted and further processed in the next round.

Criteria for designing S-boxes : According to "The Design of Rijndael" the design criteria includes:

1. Non-linearity, specifically "the maximum input-output correlation amplitude must be as small as possible" and "the maximum difference propagation probability must be as small as possible". This is to prevent linear and differential cryptanalysis.
2. Algebraic complexity. This was to prevent algebraic attacks.

Prob.8 Explain the importance of one-time initialization process. Describe each steps of AES algorithm. [R.T.U. 2016]

[R.T.U. 2016]

Sol. Importance of One Time Initialization Process: AES algorithm uses the basic techniques of substitution and transposition. The key size and the plain text block decide how many rounds need to be executed. The minimum number of rounds is 10 and maximum number of rounds is 14. The one-time initialization processes expand the 16-byte key to get the actual key block to be used. These initialization process also activate the 16-byte plain text block and XOR the state with key block. Only after these initial one-time initializations further round can commence.

Advanced Encryption Standard (AES) : AES is a block cipher of block size 128 bits. The key length can be 128, 192 or 256 bits, that is, in multiples of 64. AES is a substitution permutation cipher involving n -rounds where n depends on the key length. We take 128 bits block of AES as a 4×4 matrix $[0, 0, \dots, 0, \dots, 0, \dots, 0]$. This matrix is called the "State". The state is filled from the input in columns. For example, the input is 16 bytes $b_0, b_1, b_2, \dots, b_4, b_5, \dots, b_{15}$.

Some operations in the algorithm are performed in columns of the State and some on the row. So that this representation implements a form of columns transposition. The 4 steps of algorithm operate as :

(i) **Byte Substitution :** The 1st step is a simple substitution: $s[i, j]$ becomes $s'[i, j]$ through defined substitution table.

(ii) **Shift Row :** In the 2nd step the rows of s are permuted by left circular shift, 1st i element of row i are shifted around to the end.

(iii) **Mix Columns :** The 3rd step is complex transformation on the column of s under which the 4 elements of each column are multiplied by a polynomial, essentially diffusing each element of the column over all four elements of that column.

(iv) **Add Round Key :** Finally, a key is derived and added to each column.

This sequence is repeated for a number of rounds depending on the key length.

Prob.9 What are the non-linear components used in DES encryption and decryption. [R.T.U. Dec 2015]

[B.Tech. (V7 Sem.) CS Solved Papers]

Sol. In DES, non-linearity is introduced into the encryption so that decryption will be computationally infeasible without the secret key. This is achieved with the use of S-boxes which are basically non-linear substitution tables where either the output is smaller than the input or vice versa.

The S-boxes are the only non-linear operation in DES and are therefore the most important part of its security. They were very carefully designed by the NSA of US Government. The S-boxes accept a 48-bit input and output a 32-bit number.

The input to the S-boxes in 48 bits long arranged into 8, 6 bit blocks (b_1, b_2, \dots, b_6) . There are 8 S-boxes (S_1, S_2, \dots, S_8) each of which accepts one of the 6 bit blocks. The output of each S-box is a four bit number. Each of the S-boxes can be thought of as a 4×16 matrix. Each cell of the matrix is identified by a coordinate pair (i, j) , where $0 \leq i \leq 3$ and $0 \leq j \leq 15$. The value of i is taken as the decimal representation of the first and last bits of the input to each S-box, i.e. $\text{Dec}(b_1 b_2) = i$ and the value of j is taken from the decimal representation of the inner four bits that remains, i.e. $\text{Dec}(b_3 b_4 b_5 b_6) = j$. Each cell within the S-box matrices contains a 4-bit number which is output once that particular cell is selected by the input.

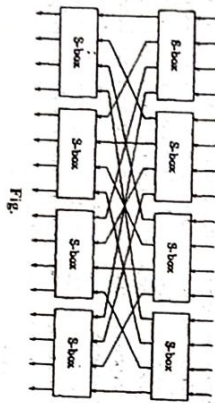


Fig.

Prob.10 Explain all block cipher modes of operation with neat diagram. [R.T.U. 2015]

OR

Explain block cipher modes of operation. [Reg.Um/2006, 2003]

Sol. Block cipher modes are the encryption modes which take as an input a block of data (a group of bits) instead of individual bits. There are various block cipher operation modes.

(i) **ECB - Electronic Codebook Mode :** It is the simplest mode in which plain text is handled 64 bits at a time and each block of plain text is encrypted using the same key. The term codebook is used because for a given key there is a unique ciphertext for every 64 bits block of plain text. For a message longer than 64 bits the procedure is simply to break the message into 64 bits blocks, padding the last block if necessary. Decryption is performed one block at a time, using

[Information Security System]

the same key. ECB method is ideal for short amount of data only, such as an encryption key. Its most significant characteristic is that, if the same 64 bits block of plain text appears more than once in the message, it always produces the same cipher text.

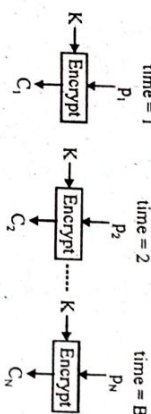


Fig. 1 : ECB Encryption

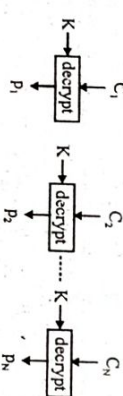


Fig. 2 : ECB Decryption

(ii) **Cipher Feedback Mode :** It is assumed that the unit of transmission is a bit, a common value is $s = 8$. The units of plain text are chained together, so that the ciphertext of any plain text is a function of all the preceding plain text. The plain text is divided into segments of bits. The input to the encryption function is a 64-bit shift register that is initially set to some IV. The left most s bits of O/P are XORed with the first segment of plain text P_1 to produce the first unit of cipher text C_1 , which is transmitted. It is general purpose stream oriented transmission, which also provides authentication.

(iii) **CBC (Cipher Block Chaining) Mode :** It takes 64 bits blocks of input and generates its own 64 bit random number and uses the cipher text C_{i-1} as the next random number r_{i-1} that will be XORed into next plain text.

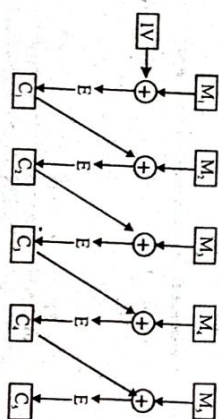


Fig. 3 : CBC Encryption

To avoid having two plain text messages that start the same wind up with the same cipher text in the beginning, CBC selects one random number which gets XORed into the first block of plain text.

ISS-15

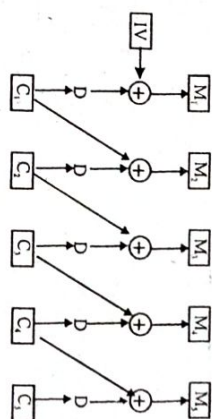


Fig. 4 : CBC Decryption

(iv) **CTR (Counter) Mode :** In this, a one item pad is generated with the data - CTR increments the IV and encrypts the result to get successive blocks of the one item pads. The main advantage is that the cryptography can be precomputed and encryption is simply an XOR and we can decrypt the message starting at any point rather than being forced to start at the beginning.

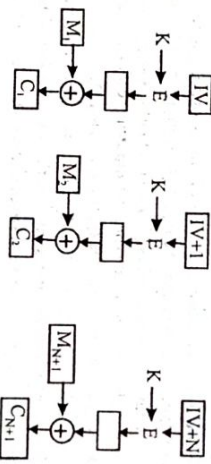


Fig. 5 : CTR Encryption

Decryption is just the reverse of encryption.

Prob.11 What do you understand by the strength of DES?

Sol. Strength of DES : The cryptographic algorithm DES transforms a 64-bit binary value into a unique 64-bit binary value based on a 56-bit variable. If the complete 64-bit input is used (i.e., none of the input bits should be predetermined from block to block) and if the 56-bit variable is randomly chosen, no technique other than trying all possible keys using known input and output for the DES will guarantee finding the chosen key. As there are over 70,000,000,000,000,000 (seventy quadrillion) possible keys of 56-bits, the feasibility of deriving a particular key in this way is extremely unlikely in typical threat environments. Moreover, if the key is changed frequently, the risk of this event is greatly diminished. However, users should be aware that it is theoretically possible to derive the key in fewer trials (with a correspondingly lower probability of success depending on the number of keys tried) and should be cautioned to change the key as often as practical.

The security of the cipher is augmented by the simple structure. For instance, the rate of diffusion is increased by several simple steps in the round: integer multiplication, the quadratic equation, and fixed bit shifting. The data-dependent rotations are improved, because the rotation amounts are determined from the high-order bits in $f(x)$, which in turn are dependent on the register bits. RC6 security has been evaluated to possess an "adequate security margin"; this rating is given with knowledge of theoretical attacks, which were devised out of the multiple evaluations. The AES-specific security evaluations provide sufficient breadth and depth to how RC6 security is affected by the simplicity of the cipher.

Prob.14 Write short note on triple DES.

[R.T.U. 2019]

OR

Explain the DES Algorithm in detail. What is the block size, cipher key size and round key size in DES?

[R.T.U. 2017]

OR

Explain DES with Triple DES with all its steps in detail.

[R.T.U. 2015]

Sol. Triple DES with Three Keys : The block diagram of Triple DES with three keys is shown in fig.1. The plaintext first encrypted with key K_1 , then encrypted with a second key K_2 , and finally with the key K_3 , where K_1 , K_2 and K_3 all are different from each other.

For decryption in Triple DES, ciphertext is first decrypted using key K_3 , then K_2 and finally with key K_1 to obtain the plaintext.

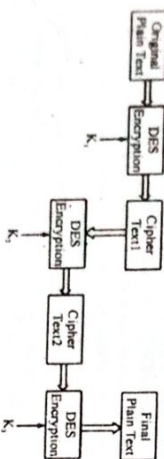


Fig. 1 : Triple DES with Three Keys

Triple DES generally used in e-mail security technique like PGP and S/MIME.

Triple DES with Two Keys : Triple DES with three keys is highly secured but it is difficult to implement in practical situations. Because it requires $56 \times 3 = 168$ bits for key, that is difficult. To overcome this problem Tuchman suggests Triple DES with two keys. The Triple DES with two keys as shown in the fig.2 have the following steps :

1. Encrypt the plaintext with key K_1 to obtain the Ciphertext1.
2. Now decrypt the ciphertext1 using the key K_2 to obtain the ciphertext2.
3. Finally encrypt the ciphertext2 using the key K_1 to obtain the final ciphertext. This process also known as Encrypt-Decrypt-Encrypt (EDE) DES.

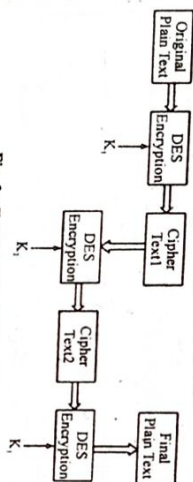


Fig. 2 : Triple DES with Two Keys

For decryption we use the reverse process of encryption.

DES with All Steps

The algorithm involves carrying out combinations, substitutions and permutations between the text to be encrypted and the key, while making sure the operations can be performed in both directions (for decryption). The combination of substitutions and permutations is called a product cipher. The key is ciphered on 64 bits and made of 16 blocks of 4 bits, generally denoted K_1 to K_{16} . Given that only 56 bits are actually used for encrypting, there can be 2^{56} (or 7.2×10^{16}) different keys.

The main parts of the algorithm are as follows :

- Fractioning of the text into 64-bit (8 octet) blocks.
- Initial permutation of blocks.
- Breakdown of the blocks into two parts: left and right, named L and R.
- Permutation and substitution steps repeated 16 times (called rounds).
- Re-joining of the left and right parts then inverse initial permutation.

Information Security System

The main idea behind this algorithm is shown in the figures below:

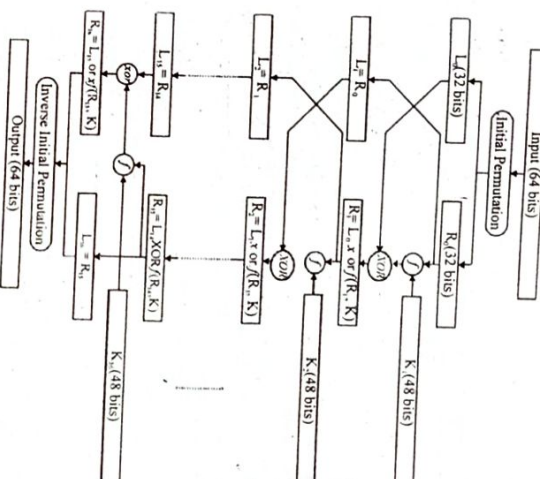


Fig. 3

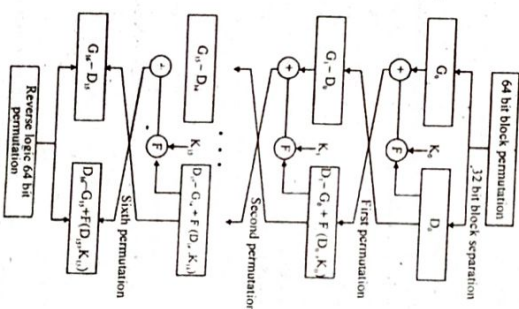


Fig. 4

Steps

Initial Permutation

Firstly, each bit of a block is subject to initial permutation, which can be represented by the following initial permutation (IP) table :

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8

57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

This permutation table shows, when reading the table from left to right then from top to bottom, that the 58th bit of the 64-bit block is in first position, the 50th in second position and so forth.

Division into 32-bit Blocks

Once the initial permutation is completed, the 64-bit block is divided into two 32-bit blocks, respectively denoted L and R (for left and right). The initial status of these two blocks is denoted L_0 and R_0 .

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8

57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

It is interesting to note that L_0 contains all bits having an even position in the initial message, whereas R_0 contains bits with an odd position.

Rounds

The L_n and R_n blocks are subject to a set of repeated transformations called rounds, shown in fig.5, and the details of which are given below :

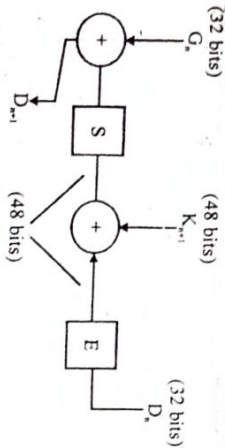


Fig. 5

Expansion Function

The 32 bits of the R_0 block are expanded to 48 bits thanks to a table called an *expansion table* (denoted E), in which the 48 bits are mixed together and 16 of them are duplicated.

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

As such, the last bit of R_0 (that is, the 7th bit of the original block) becomes the first, the first becomes the second, etc.

In addition, the bits 1, 4, 5, 8, 9, 12, 13, 16, 17, 20, 21, 24, 25, 28 and 29 of R_0 (respectively 57, 33, 25, 1, 59, 35, 27, 3, 61, 37, 29, 5, 63, 39, 31 and 7 of the original block) are duplicated and scattered in the table.

Exclusive OR with the Key

The resulting 48-bit table is called R'_0 or $E[R_0]$. The DES algorithm then *exclusive ORs* the first key K_1 with $E[R_0]$. The result of this *exclusive OR* is a 48-bit table we will call R_0 out of convenience (it is not the starting R_0).

Substitution Function

R_0 is then divided into 8, 6-bit blocks, denoted R_{0i} . Each of these blocks is processed by *selection functions* (sometimes called *substitution boxes* or *compression functions*), generally denoted S_i . The first and last bits of each R_{0i} determine (in binary value) the line of the selection function; the other bits (respectively 2, 3, 4 and 5) determine the column. As the selection of the line is based on two bits, there are 4 possibilities (0, 1, 2, 3). As the selection of the column is based on 4 bits, there are 16 possibilities (0 to 15). The selection function "selects" a ciphered value of 4 bits.

Here is the first substitution function, represented by a 4-by-16 table below:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6

Let R_{01} equal 1011110. The first and last bits give 10, that is, 2 in binary value. The bits 2, 3, 4 and 5 give 0111, or 7 in binary value. The result of the selection function is therefore the value located on line no. 2, in column no. 7. It is the value 11, or 111 binary.

Each of the 8, 6-bit blocks is passed through the corresponding selection function, which gives an output of 8 values with 4 bits each. Here are the other selection functions:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5
1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11
2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2
3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2
1	13	7	0	4	9	3	4	6	10	2	8	5	14	12	11
2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14
3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4
1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14
2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8
3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14
1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8
2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0
3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5
1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3
2	9	14	15	2	8	12	3	7	0	4	10	1	13	11	6
3	4	13	2	12	9	5	15	10	11	14	1	7	6	0	8

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6
1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8
2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9
3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3

B.Tech. (VT Sem.) CS Solved Papers

Information Security System

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12
1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9
2	1	7	11	4	1	9	12	14	2	0	6	10	13	15	3
3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6

Each 6-bit block is therefore substituted in a 4-bit block. These bits are combined to form a 32-bit block.

Permutation

The obtained 32-bit block is then subject to a permutation P here is the table:

16	7	20	21	29	12	28	17
1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9
19	13	30	6	22	11	4	25

Exclusive OR

All of these results output from P are subject to an *Exclusive OR* with the starting L_0 (as shown on the fig. 3) to give R_1 , whereas the initial R_0 gives L_1 .

Iteration

All of the previous steps (*rounds*) are repeated 16 times.

Inverse Initial Permutation

At the end of the iterations, the two blocks L_{16} and R_{16} are re-joined, then subject to inverse initial permutation.

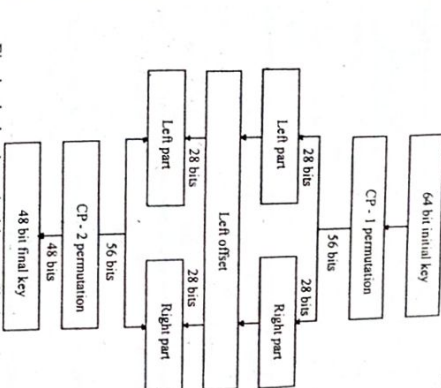
40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

The output result is a 64-bit ciphertext.

Generation of Keys

Given that the DES algorithm presented above is public, security is based on the complexity of encryption keys.

The algorithm shows how to obtain, from a 64-bit key (made of any 64 alphanumeric characters), 8 different 48-bit keys each used in the DES algorithm.



Firstly, the key's parity bits are eliminated so as to obtain a key with a useful length of 56 bits.

The first step is a permutation denoted $PC-1$ whose table is presented below:

57	49	41	33	25	17	9	1	58	50	42	34	26	18
10	2	59	51	43	35	27	19	11	3	60	52	44	36
63	55	47	39	31	23	15	7	62	54	46	38	30	22
14	6	61	53	45	37	29	21	13	5	28	20	12	4

This table may be written in the form of two tables L_1 and R_1 (for left and right) each made of 28 bits:

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36

63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

The result of this first permutation is denoted L_0 and R_0 . These two blocks are then rotated to the left, such that the bits in second position take the first position, those in third position take the second, etc.

The bits in first position move to last position.

The 2, 28-bit blocks are then grouped into one 56-bit block. This passes through a permutation, denoted $PC-2$, giving a 48-bit block as output, representing the key K_1 .

14	17	11	24	1	5	3	28	15	6	21	10
23	19	12	4	26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40	51	45	33	48
44	49	39	56	34	53	46	42	50	36	29	32

Repeating the algorithm makes it possible to give the 16 keys K_1 to K_{16} used in the DES algorithm.

LS	1	2	4	6	8	10	12	14	15	17	19	21	23	25	27	28
----	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----

Prof. 15 What is Shannon's theory of confusion and diffusion? Explain Feistel structure of block ciphers.

OR
Explain Shannon's theory of confusion and diffusion in detail.

OR
What is Feistel block cipher? How it is worked?

OR
Explain the parameters and design choices determines real algorithm of Feistel cipher? Explain Feistel decryption algorithm.

OR
Differentiate the 'confusion' and 'diffusion'. Also explain their significance to make encryption secure.

OR
What do you mean by link to link and end to end encryption? Explain the concept of confusion and diffusion in block cipher.

Sol. Shannon's Theory of Confusion and Diffusion: Shannon suggests two methods for frustrating statistical cryptanalysis: **diffusion** and **confusion**. In Diffusion the statistical structure of the plaintext is dissipated into long range statistics of the ciphertext. This is achieved by having each plaintext digit affect the value of many ciphertext digits, which is equivalent to saying that each ciphertext digit is affected by many plaintext digits. An example of diffusion is to encrypt a message $M = m_1, m_2, m_3, \dots$ of characters with an averaging operation:

$$y_n = \sum_{i=1}^n m_i \pmod{26}$$

adding k successive letters to get a ciphertext letter y_n . One can show that the statistical structure of the plaintext has been dissipated. Thus, the letter frequencies in the ciphertext will be more nearly equal than in the plaintext; the diagram frequencies will also be more nearly equal, and so

on. In a binary block cipher, diffusion can be achieved by repeatedly performing some permutation on the data followed by applying a function to that permutation; the effect is that bits from different positions in the original plaintext contribute to a single bit of ciphertext.

Every block cipher involves a transformation of a block of plaintext into a block of ciphertext, where the transformation depends on the key. The mechanism of diffusion seeks to make the statistical relationship between the plaintext and ciphertext as complex as possible in order to thwart attempts to deduce the key. On the other hand, confusion seeks to make the relationship between the statistics of the ciphertext and the value of the encryption key as complex as possible, again to thwart attempts to discover the key. Thus, even if the attacker can get some handle on the statistics of the ciphertext, the way in which the key was used to produce that ciphertext is so complex as to make it difficult to deduce the key. This is achieved by the use of a complex substitution algorithm. In contrast, a simple linear substitution function would add little confusion.

Encryption can be performed at different communication levels, each with different types of protection and implications. Two general modes of encryption implementation are link encryption and end-to-end encryption. Link encryption encrypts all the data along a specific communication path, as in a satellite link, T3 line, or telephone circuit. Not only is the user information encrypted, but the header, trailers, addresses, and routing data that are part of the packets are also encrypted. The only traffic not encrypted in this technology is the data link control messaging information, which includes instructions and parameters that the different link devices use to synchronize communication methods. Link encryption provides protection against packet sniffers and eavesdroppers. In **end-to-end encryption**, the headers, addresses, routing, and trailer information are not encrypted, enabling attackers to learn more about a captured packet and where it is headed.

Link encryption, which is sometimes called **online encryption**, is usually provided by service providers and is incorporated into network protocols. All of the information is encrypted, and the packets must be decrypted at each hop to the router, or other intermediate device, knows where to send the packet next. The router must decrypt the header portion of the packet, read the routing and address information within the header, and then re-encrypt it and send it on its way. With end-to-end encryption, the packets do not need to be decrypted and then encrypted again at each hop, because the headers and trailers are not encrypted. The devices in between the origin and destination just read the necessary routing information and pass the packets on their way.

End-to-end encryption is usually initiated by the user of the originating computer. It provides more flexibility for the

Information Security System

user to be able to determine whether or not certain messages will get encrypted. It is called "end-to-end encryption" because the message stays encrypted from one end of its journey to the other. Link encryption has to decrypt the packets at every device between the two ends.

Link encryption occurs at the data link and physical layers. Hardware encryption devices interface with the physical layer and encrypt all data that pass through them. Because no part of the data is available to an attacker, the attacker cannot learn basic information about how data flows through the environment. This is referred to as traffic-flow security.

The following list outlines the advantages and disadvantages of end-to-end and link encryption methods.

Advantages of end-to-end encryption include the following:

- It provides more flexibility to the user in choosing what gets encrypted and how.
- Higher granularity of functionality is available because each application or user can choose specific configurations.
- Each hop computer on the network does not need to have a key to decrypt each packet.

Disadvantages of end-to-end encryption include the following:

- Headers, addresses, and routing information are not encrypted, and therefore not protected.
- **Advantages of link encryption include the following:**
- All data are encrypted, including headers, addresses and routing information.
- Users do not need to do anything to initiate it. It works at a lower layer in the OSI model.

Disadvantages of link encryption include the following:

- Key distribution and management are more complex because each hop device must receive a key, and when the keys change, each must be updated.
- Packets are decrypted at each hop; thus, more points of vulnerability exist.

Feistel Block Cipher:

1. **Block Size:** Larger block sizes mean greater security but reduced encryption/decryption speed. A block size of 64 bits is a reasonable tradeoff and has been nearly universal in block cipher design. However, the new AFS uses a 128-bit block size.
2. **Key Size:** Larger key size means greater security but may decrease encryption/decryption speed. Key sizes of 64 bits or less are now widely considered to be inadequate, and 128 bits has become a common size.
3. **Number of Rounds:** The essence of the Feistel cipher is that a single round offers inadequate security but that multiple rounds offer increasing security. A typical size is 16 rounds.

4. Subkey Generation Algorithm: Greater complexity in this algorithm should lead to greater difficulty of cryptanalysis.

5. Round Function: Again, greater complexity generally means greater resistance to cryptanalysis.

Feistel Decryption Algorithm: The process of decryption with a Feistel cipher is essentially the same as the encryption process. The rule is as follows: Use the ciphertext as input to the algorithm, but use the subkeys K_i in reverse order. That is, use K_{16} in the first round, K_{15} in the second round, and so on until K_1 is used in the last round. This is a nice feature because it means we need not implement two different algorithms, one for encryption and one for decryption.

To see that the same algorithm with a reversed key order produces the correct result, consider Fig. which shows the encryption process going up the right-hand side for a 16-round algorithm (the result would be the same for any number of rounds).

For clarity, we use the notation LE_i and RE_i for data travelling through the encryption algorithm and LD_i and RD_i for data travelling through the decryption algorithm. The diagram, indicates that, at every round, the intermediate value of the decryption process is equal to the corresponding value of the encryption process with the two halves of the value swapped. To put this another way, let the output of the i th encryption round be $LE_i || RE_i$ (LE_i concatenated with RE_i). Then the corresponding input to the $(16-i)$ th decryption round is $RE_i || LE_i$, or, equivalently, $RD_{16-i} || LD_{16-i}$.

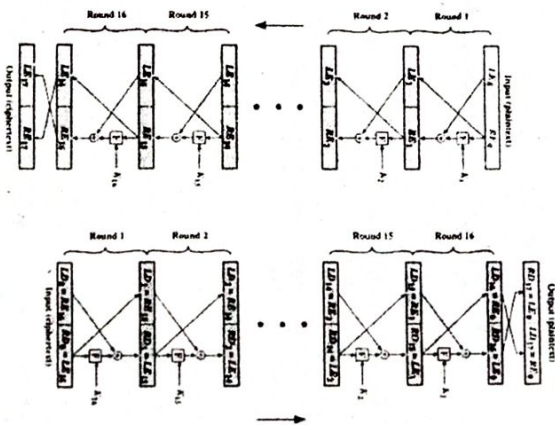


Fig.

PREVIOUS YEARS QUESTIONS

PART-A

Prob.1 Compare RSA and ElGamal schemes.

Sol. Differences between RSA and ElGamal Schemes :

	RSA	ElGamal
1.	More efficient for encryption.	More efficient for decryption.
2.	For a particular security level, lengthy keys are required.	For the same level of security very short keys are required.

Prob.2 What are the advantage of public-key cryptosystems?

Sol. Advantages of Public-key Cryptosystems :

- **Security:** Only the private key needs to be kept a secret.
- **Longevity:** Key pairs may be used without change in most cases over long periods of time-years in some situations.

Prob.3 Write disadvantages of public-key cryptosystems.

Sol. Public-key cryptosystem are slower than their symmetric-key counterparts. For instance, the RSA cryptosystem is roughly a thousand times slower than the DES symmetric-key cryptosystem.

Prob.4 Write the concept of key sizes in public-key cryptosystem.

Sol. Key Sizes : The key sizes in a public-key cryptosystem are significantly larger than the required for a symmetric-key cryptosystem. For instance, the private key in the RSA cryptosystem should be 1024 bits, where as with a symmetric-key cipher, generally 128 bits will suffice. Usually, private keys are ten times larger than secret keys.

Prob.5 An adversary intercepts two encrypted messages $c_1 = 19$ and $c_2 = 32$, encrypted using ElGamal public key $\{p, g, g^d\} = \{53, 2, 8\}$. The adversary comes to know that the first message is $m_1 = 17$ and the message have been encrypted using the same random value. How does he determine the second message m_2 ?

Sol.

$$\begin{aligned}
 m_2 &= c_2 \times (g^{dk})^{-1} \bmod p \\
 &= c_2 \times (c_1 \times m_1^{-1})^1 \bmod p \\
 &= c_2 \times (c_1^{-1} \times m_1) \bmod p \\
 &= 32 \times 19^{-1} \times 17 \bmod 53 \\
 &= 32 \times 14 \times 17 \bmod 53 = 37
 \end{aligned}$$

PART-B

Prob.6 Discuss security analysis of RSA algorithm along with its application. [R.T.U. 2019]

Sol. Security of RSA : The security of the RSA cryptosystem is based on two mathematical problems the problem of factoring large numbers and the RSA problem. Full decryption of an RSA ciphertext is thought to be infeasible on the assumption that both of these problems are hard, i.e., no efficient algorithm exists for solving them. Providing security against partial decryption may require the addition of a secure padding scheme.

The RSA problem is defined as the task of taking E_n roots modulo a composite n , recovering a value PT such that $CT = PT^d \pmod{n}$, where (n, E) is an RSA public key and CT is an RSA cipher text.

Four possible approaches to attacking the RSA algorithm are as follows:

1. Brute Force: This involves trying all possible private keys.
 2. Mathematical Attacks : There are several approaches, all equivalent in effort to factoring the product of two primes.
 3. Timing Attacks : These depend on the running time of the decryption algorithm.
 4. Chosen Ciphertext Attacks : This type of attack exploits properties of the RSA algorithm.
- The defense against the brute force approach is the same for RSA as for other cryptosystems, namely, use a large key space. Thus, the larger the number of bits in D , the better. However, because the calculations involved, both in key generation and in encryption/decryption, are complex, the larger the size of the key, the slower the system will run.
- Applications of RSA Cryptography :** There are many interesting applications of number theory and abstract algebra, especially in computer-related subjects. We shall look closer at one famous application to cryptography.
- 1. The Problem of Secure Communication :** Suppose that two persons want to communicate with each other and they want to protect their communication from being overheard by a third party. This could for example be any of the following situations.
- (a) I want to buy a book at the online bookshop Amazon. To do this, I send them my credit card number through the internet, so that they can deduct the correct amount of money from my bank account. As skilled hacker can easily intercept the communication, get hold of my credit card number, and use it to take all the money in my account.
 - (b) The American CIA agent John MacAgent has infiltrated the North Korean military and wants to send some secret information back to his colleagues in the US. If someone intercepts and understands his message, he will immediately be killed by the North Koreans.
 - (c) Sarah is deeply in love with Othello, but her parents think she is too young to have a boyfriend. So Sarah needs to send secret messages to Othello that her parents cannot understand, even if they manage to find one of the messages.
 - (d) The researchers at the company Amazing Machines Ltd. has come up with an idea for a machine that could produce large amounts of electricity very cheaply. They want to discuss these ideas through email with some leading physicists in Japan, but if someone else manages to intercept the email communication, they could steal the idea, and the company could lose millions of dollars. So how can these problems be solved? How can these people communicate in a safe way? These kinds of problems are investigated in the field of cryptography. The simplest way of solving the problem is to agree on some kind of encoding scheme. For example, Sarah and Othello could agree that in their letters, every A means B, every B means C, every C means D etc. In this case, Othello could for example send a letter with the message:

H KNDUD XNT

and Sarah would be very happy, and write back:

H KNDUD XNT SNM, RYDSDSHD!

If her mother found the piece of paper with these letters, she would not understand, and Sarah could probably convince her that this is just the password for her web mail. There are many other, much more complicated, ways of encoding messages; any such method is called an encoding scheme so that they are not easily readable to others. However, there are two possible problems with all of these methods.

 - (a) Problem 1: If the code is not complicated enough, it could easily be cracked by someone with a computer (perhaps Sarah's mother has computer programming as a hobby, scary!).
 - (b) Problem 2: Suppose that Sarah's father gets hold of the first letter. Then he would be able to understand every subsequent letter. The first problem can perhaps be solved by making the encoding complicated enough, but the second is a major problem! If for example, my computer system agrees with the Amazon website on how to encode the credit card number, and someone gets hold of this information, then they will be able to read my credit card number even if it is encoded. This problem can actually be overcome, by using something called RSA cryptography.

2. Factoring Large Numbers : One of the ideas behind the RSA cryptography is that it is very hard to factor large integers, even if you use a computer. You have learnt how to factor small numbers, but how would you find the prime factorization of an integer with 200 digits? Of course, you could start by checking the primes 2, 3, 5, 7 and so on, to see if any of them divides the large number. However, such a number is so large that you would become old and probably die before you have checked all primes up to n , even if you used a computer. I mentioned earlier that you can find the factorization of the number,

7 4 0 3 7 6 3 4 7 9 5 6 1 7 1 2 8 2 8 0 4 7 9 6 0 9 7 4 2 9
5 7 3 1 4 2 5 9 3 1 8 8 8 8 9 2 3 1 2 8 9 0 8 4 9 3 6 2 3 2 6 3
8 9 7 2 7 6 5 0 3 4 0 2 8 2 6 6 2 7 6 8 9 1 9 9 6 4 1 9 6 2 5 1
1 7 8 4 3 9 5 8 9 4 3 0 5 0 2 1 2 7 5 8 5 3 7 0 1 1 8 9 6 8
0 9 8 2 8 6 7 3 1 7 3 2 7 3 1 0 8 9 3 0 9 0 5 5 2 5 0 5 1 1
6 8 7 7 6 3 2 9 0 7 2 3 9 6 3 8 0 7 8 6 7 1 0 0 8 6 0 9 6 9
6 2 5 3 7 9 3 4 6 5 0 5 6 3 7 9 6 3 5 9 then you would be awarded a sum of US\$ 30,000.

Such a prize exists to encourage research in this area, because it has an enormous impact on the millions of messages and financial transactions that take place every day through the internet and other communication channels.

3. Any Message can be Expressed in Terms of Integers : It is easy to transform any message written with letters to a message written in integers. We could use any of the common methods used in computers, such as ASCII, Unicode and so on. In this course, we will simply write the instead of A, 2, instead of B, and so on, up to 26 instead of Z. We will also write 0 instead of an empty space. So we could send the numbers 8 9 0 20 8 5 18 instead of the message "HI THERE".

Prob-7 Explain Diffie-Hellman key exchange algorithm in detail. [R.T.U. 2019]

OR
What is the man in the middle attack problem in Diffie-Hellman key exchange algorithm? Explain with example. [R.T.U. 2019]

OR
Explain Diffie-Hellman key exchange algorithm in detail what are "Clogging attack" and "Man in the middle attack" on Diffie-Hellman algorithm? [R.T.U. Dec. 2015]

OR
Explain with all Diffie-Hellman key exchange steps in detail. [R.T.U. 2015]

OR
Explain Diffie-Hellman key exchange algorithm in detail. Also discuss "Man in the Middle Attack" problem with suitable example. [R.T.U. 2015]

Sol. The Diffie-Hellman protocol is a method for two computer users to generate a shared private key with which they can then exchange information across an insecure channel. Let the users be named Alice and Bob. First they agree on two prime numbers g and p , where p is large (typically at least 512 bits) and g is a primitive root modulo p . (In practice, it is a good idea to choose p such that $(p-1)/2$ is also prime.) The numbers g and p need not be kept secret from other users.

Now Alice chooses a large random number a as her private key and Bob similarly chooses a large number b . Alice then computes $A = g^a \pmod{p}$, which she sends to Bob and Bob computes $B = g^b \pmod{p}$, which he sends to Alice. Now both Alice and Bob compute their shared key $K = g^{ab} \pmod{p}$, which Alice computes as $K = B^a \pmod{p}$ and Bob computes as $K = A^b \pmod{p} = (g^a)^b \pmod{p}$.

Alice and Bob can now use their shared key K to exchange information without worrying about other users obtaining this information. In order for a potential eavesdropper (Eve) to do so, she would first need to obtain $K = g^{ab} \pmod{p}$ knowing only $g, p, A = g^a \pmod{p}$ and $B = g^b \pmod{p}$.

This can be done by computing a from $A = g^a \pmod{p}$ and p from $B = g^b \pmod{p}$. This is the discrete logarithm problem, which is computationally infeasible for large p . Computing the discrete logarithm of a number modulo p takes roughly the same amount of time as factoring the product of two primes the same size as p , which is what the security of the RSA cryptosystem relies on. Thus, the Diffie-Hellman protocol is roughly as secure as RSA.



Man in the middle attack : The Diffie-Hellman key exchange is vulnerable to a man-in-the-middle attack. In this attack, an opponent Carol intercepts Alice's public value and sends her own public value to Bob. When Bob transmits his public value, Carol substitutes it with her own and sends it to Alice. Carol and Alice thus agree on one shared key. After this exchange, Carol simply decrypts any messages sent out by Alice or Bob, and then reads and possibly modifies them before re-encrypting with the appropriate key and transmitting them to the other party. This vulnerability is present because the Diffie-Hellman key exchange does not authenticate the participants. Possible solutions include the use of digital signatures and other protocol variants.



Clogging attack : The Diffie-Hellman key exchange algorithm is also vulnerable to a clogging attack. Such an attack can be executed as follows:

1. Attacker sends fake Diffie-Hellman messages to a victim from a forged IP address.
2. Victim starts performing modular exponentiations to compute a secret key.
3. If the steps 1 and 2 are repeated again and again, the victim can become blocked with useless work.

Prob-8 Write short note on security of RSA. [R.T.U. 2014]

Sol. Refer to Prob-6.

Prob-9 Write short note on Discrete logarithm. [R.T.U. 2014]

OR
Explain the discrete logarithms problem. [R.T.U. 2017]

OR
Explain Discrete logarithm with example. [R.T.U. 2014, 13, 12]

Sol. Discrete Logarithm : For a finite cyclic group G , containing n elements let b be a generator element of G . Assuming that G is repeated multiple times, then every element g of G can be written in the form $g = b^x$ for some integer x . Furthermore, any two such integers x_1 and x_2 , representing g will be congruent modulo n . We can thus define a function.

Decryption

$$\begin{aligned} M &= c^d \bmod n \\ &= 106^{11} \bmod 143 \\ &= 7 \end{aligned}$$

$$\begin{aligned} \text{(iii) } n &= p \times q \\ &= 17 \times 31 \\ &= 527 \end{aligned}$$

$$\begin{aligned} \phi(n) &= (p-1) \times (q-1) \\ &= 16 \times 30 \\ &= 480 \end{aligned}$$

$$\begin{aligned} \gcd(\phi(n), e) &= \gcd(480, 7) \\ &= 1 \end{aligned}$$

$$d \equiv e^{-1} \pmod{\phi(n)}$$

$$d \times e \bmod \phi(n) = 1$$

$$7d \bmod 480 = 1$$

$$\therefore d = 343$$

So, public key $pr = \{e, n\} = \{7, 527\}$

private key $pr = \{d, n\} = \{343, 527\}$

Encryption

$$\begin{aligned} C &= M^e \bmod n \\ &= 2^7 \bmod 527 \\ &= 128 \end{aligned}$$

Decryption

$$\begin{aligned} M &= c^d \bmod n \\ &= 128^{343} \bmod 527 \\ &= 2 \end{aligned}$$

Prob.14 What do you mean by elliptic curve cryptosystem?

Sol. Elliptic Curve Cryptosystem : Elliptic curve cryptosystems were first proposed independently by Victor Miller and Neal Koblitz in the mid-1980s. At a high level, they are analogs of existing public-key cryptosystems in which modular arithmetic is replaced by operations defined over elliptic curves. The elliptic curve cryptosystems that have appeared in the literature can be classified into two categories according to whether they are analogs to the RSA system or a discrete logarithm based systems.

Just as in all public-key cryptosystem, the security of elliptic curve cryptosystem relies on the underlying hard mathematical problems. It turns out that elliptic curve analogs of the RSA system are mainly of academic interest and offer no practical advantage over the RSA system, since their security is based on the same underlying problem, namely integer factorization. The situation is quite different with elliptic curve variants of discrete logarithm based system. The security of such systems depends on the following hard problem: Given two points G and Y on an elliptic curve such that $Y = kG$ (that is, Y is G added to itself k times), find the integer k . This problem is commonly referred to as the elliptic curve discrete logarithm problem.

Presently, the methods for computing general elliptic curve discrete logarithms are much less efficient than those for factoring or computing conventional discrete logarithms. As a result, shorter key sizes can be used to achieve the same security of conventional public-key cryptosystem, which might lead to better memory requirements and improved performance. One can easily construct elliptic curve encryption, signature, and key agreement schemes by making analogs of ElGamal, DSA, and Diffie-Hellman. These variants appear to offer certain implementation advantages over the original schemes, and they have recently drawn more and more attention from both the academic community and the industry.

Prob.15 Elliptic curve cryptosystem is secure or not. Explain.

Sol. In general, the best attacks on the elliptic curve discrete logarithm problems have been general brute-force method. The current lack of more specific attacks means that shorter key sizes for elliptic cryptosystem appear to give similar security as much larger keys that might be used in cryptosystem based on the discrete logarithm problem and integer factorization. For certain choices of elliptic curves there do exist more efficient attacks. Menezes, Okamoto, and Vanstone have been able to reduce the elliptic curve discrete logarithm problem to the traditional discrete logarithm problem for certain curves, thereby necessitating the same size keys as is used in more traditional public-key systems. However these cases are readily classified and easily avoided.

In 1997, elliptic curve cryptosystem began to receive a lot more attention; by the end of 1999, there were no major developments as to the security of these cryptosystem. The longer this situation continues, the more confidence will grow

that they really do offer as much security as currently appears. However, a sizeable group of very respected researchers have some doubts as to whether this situation will remain unchanged for many years. In particular, there is some evidence that the use of special elliptic curves, sometimes known as Koblitz curve, which provide very fast implementations, might allow new specialized attacks. As a starting point, the basic brute-force attacks can be improved when attacking these curves. While RSA Laboratories believes that continued research into elliptic curve cryptosystem might eventually create the same level of widespread trust as is enjoyed by other public-key techniques (provided there are no upsets), the use of special purpose curves will most likely always be viewed with extreme skepticism.

Prob.16 Explain generalized ElGamal cryptosystem. Write applications of ElGamal cryptosystem.

Sol. Generalized ElGamal Cryptosystem: ElGamal can be generalized to any finite cyclic group where discrete logarithm problem is infeasible. Some other suitable groups for ElGamal cryptosystem are given below.

- Multiplicative group of finite field $GF(p^n)$
- Multiplicative group of the finite field $GF(2^n)$
- Additive elliptic curve group over a finite field $GF(p^n)$ and $GF(2^n)$

We illustrate generalized ElGamal cryptosystem over multiplicative group of finite field $GF(2^n)$ with the following example. This example is based on $GF(2^3)$ over irreducible polynomial $x^3 + x + 1$.

Key Generation : Alice selects multiplicative group over finite field $GF(2^3)$, whose elements are $\{000, 001, 010, 011, 100, 110, 111\}$. The irreducible polynomial for multiplication is $x^3 + x + 1$ (1011). She chooses the generates $g = x$ (010) and the private key $d = 4$. Now,

$$g^d = g^4 = x^2 + x = (110)$$

Therefore, the public key is $\{1011, 0140, 110\}$ which she sends to Bob. Note that instead of prime number p , we specify the irreducible polynomial (1011) in the public key.

Encryption: Bob chooses $k = 5$ to encrypt message $m = (100) = x^2 = 8^2$. He computes ciphertext c and that hint r as indicated below and sends these to Alice.

$$\begin{aligned} (g^d)^k &= g^{20} = g^6 = x^2 + 1 = (101) \\ c &= mg^{dk} \bmod (x^3 + x + 1) \\ &= g^2 \times g^6 = g^8 = x = (010) \\ r &= g^k = g^5 = x^2 + x + 1 = (111) \end{aligned}$$

Decryption : Alice uses her private key to decrypt c as indicated below.

$$r^d = (g^5)^4 = g^{20} = g^6 = x^2 + 1 = (111)$$

$$m = c (r^d)^{-1} \bmod (x^3 + x + 1)$$

$$= g \times g^{-6} = g^{-5} = g^2 = x^2 = (010)$$

Application of ElGamal Cryptosystem : ElGamal finds application where RSA is used.

- It can be used for encrypting small messages. One such typical application, as mentioned in RSA, is exchanging secret symmetric keys between two entities before they commence the exchange of messages encrypted using the symmetric key. ElGamal can be used for encrypting the symmetric key.
- It can be used for generating digital signature which enables authentication of the source.

Prob.17 Explain the concept of public key cryptanalysis.

Sol. Public-Key Cryptanalysis: The tool to break the code is called cryptanalysis and it tells (Kaufman(2002)) how system works and how to find out a secret key. Cryptanalysis is also used to refer to any attempt to circumvent the security of other types of cryptographic algorithms and protocols. Cryptanalysis usually excludes methods of attack that do not primarily target weaknesses in the actual cryptography. These are four types of cryptanalytic attack:

- Cipher text-only:** In this type of cryptanalytic attack, the cryptanalyst access only to a collection of code texts.
- Known-plaintext:** In this type of attack, the attacker has a set of coded text to which he knows to the relevant plaintext.
- Chosen-plaintext:** In this the attacker can obtain the coded text into an arbitrary set of plaintexts of his own choice.
- Related-key attack:** This type of attack is like the above attack the only difference is that the attacker can obtain coded text encrypted with the help of two different keys. These two keys are unknown, but have some relationship in between them.

Prob.18 Write down the requirements and various applications of public key cryptosystem.

Sol. Requirements of Public Key Cryptosystem: The public Key Cryptosystem depends on a cryptographic algorithm based on two related keys. Diffie and Hellman postulated this system without demonstrating that such algorithms exist. However they did lay out the conditions that such algorithms must fulfill.

1. It is computationally easy for a party B to generate a pair (public key KU_b , private key KR_b)
2. It is computationally easy for a sender A, knowing the public key and the message to be encrypted, M , to generate the corresponding ciphertext.

$$C = E_{KU_b}(M)$$

3. It is computationally easy for a receiver B to decrypt the resulting ciphertext using the private key to recover the original message.

$$M = D_{KR_b}(C) = D_{KR_b}[E_{KU_b}(M)]$$

4. It is computationally infeasible for an opponent, knowing the public key, KU_b , to determine the private key, KR_b .
5. It is computationally infeasible for an opponent, knowing the public key, KU_b , and a ciphertext, C , to recover the original message, M .

Applications of Public-key Cryptosystem: With help of two keys i.e. private and public key used in cryptographic algorithm the public-key systems is characterize. There are three types of the usages of public-key cryptosystems: (Kaufman(2002))

1. **Encryption/Decryption:** The sender encrypts a message with the recipient's public key.
2. **Digital Signature:** In cryptographic algorithm, the privacy of the message is achieved with the help of the sender signing with its private key and which applied to the message or some part of the message.
3. **Key Exchange:** Two sides cooperate to exchange a session key. Several different approaches are possible, involving the private key(s) of one or both parties.

PART-C

Prob.19 Explain Public key cryptosystems along with its principles. [R.T.U. 2019]

OR

Write short note on public key? [R.T.U. 2018]

OR

Explain the process of distribution of secret keys using public key cryptosystem. [R.T.U. 2017]

OR

Write short note on principles of public key cryptosystems. [R.T.U. 2015]

OR

Explain the concept of public key cryptography or asymmetric key cryptography with example. Differentiate between them. [R.T.U. 2014]

OR

Explain public key cryptography. Describe encryption and authentication process with diagrams. [R.T.U. 2010, Raj.Univ. 2008]

Sol. Public Key Cryptography : We know that much of the theory of public-key cryptography is based on number theory.

The concept of public key cryptography evolved from an attempt to attack two of the most difficult problem i.e. association with symmetric encryption requires (i) that two communicants already share a key, which has been distributed to them.

(ii) The use of a key distribution center. The overall framework for public key cryptography. The requirements for the encryption/decryption algorithm that is at the heart of the scheme.

Public-key algorithms rely on one key for encryption and a different but related key for decryption.

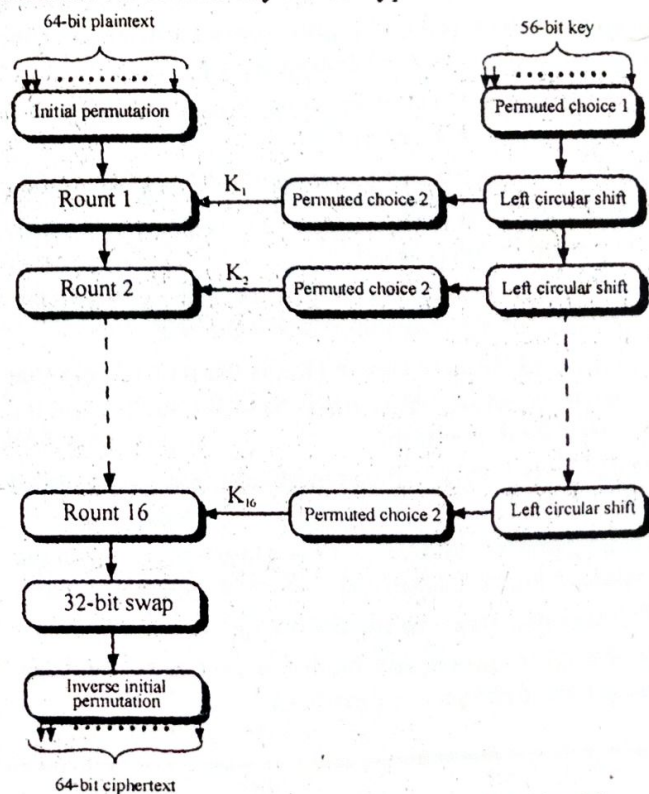


Fig. 1 : General Depiction of DES Encryption Algorithm

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

(a) Initial Permutation (IP)

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

(b) Inverse Initial Permutation (IP⁻¹)

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

(c) Expansion Permutation (E)

16	7	20	21	29	12	28	17
1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9
19	13	30	6	22	11	4	25

(d) Permutation Function (P)

Characteristics : It is computationally infeasible to determine the decryption key given only knowledge of the cryptographic algorithm and encryption key.

Either of the two related keys can be used for encryption, with the other used for decryption.

A public-key encryption scheme has following ingredients:
Plaintext : This is the readable message or data that is fed into the algorithm as input.

Encryption algorithm : The encryption algorithm performs various transformation on the plaintext.

Public and Private key : This is the pair of keys that have been selected so that if one is used for encryption, the other is used for decryption.

Ciphertext : This is the scrambled message produced as output. It depends on the plaintext and the key.

Decryption algorithm : This algorithm accepts the ciphertext and the matching key and produces the original plaintext.

The essential steps are the following :

(1) Each user generates a pair of keys to be used for the encryption and decryption of message.

(2) Each user places one of the two keys in a public register or other accessible file. This is the public key. The companion key is kept private.

If Bob wishes to send a confidential message to Alice, Bob encrypts the message using Alice's public key.

By using this approach, all participants have access to public keys, and private keys are generated locally by each participant and therefore need never be distributed. As long as a system controls its private key, its incoming communication secure.

(a) Encryption

(b) Authentication

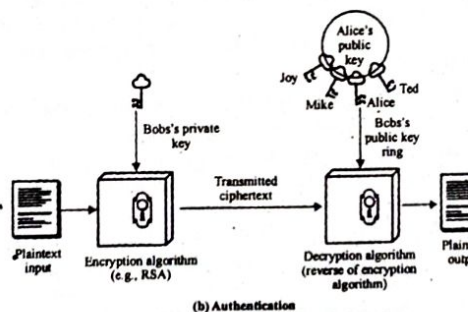
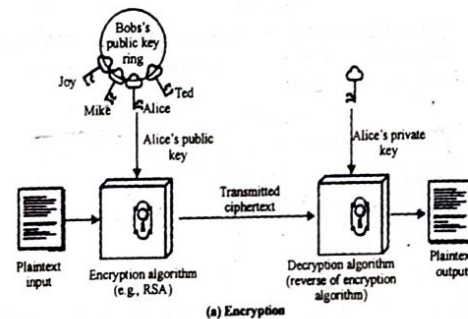


Fig. 2 : Public-Key Cryptography

At any time, a system can change its private key and publish the companion public key to replace its old public key.

To summarize some of the important aspects of symmetric and public key encryption as a secret key; the two keys used for public key encryption are referred to as the public key and the private key.

Difference between public key cryptography and symmetric key cryptography

(1) Symmetric key cryptography uses only one key for encryption and decryption whereas public key cryptography uses two keys for the encryption and decryption. One key is public key known to all users and second is private key known to only one users.

(2) Symmetric key cryptography is simple to implement but public key cryptography is difficult to implement.

(3) Symmetric key cryptography is less secure than public key cryptography.

(4) There is no need of key management of key distribution in symmetric key cryptography but it is required in public key cryptography.

(5) Example of symmetric key algorithm are IDEA, DES and public key algorithm are RSA.

Prob.20 What is the strength of RSA? What are the different kinds of attacks possible against RSA?

[R.T.U. Dec. 2015]

Sol. The asymmetric nature of RSA allows it a sizable advantage over symmetric-key algorithms. The unique private and public keys provided to each user allows them to conduct secure exchanges of information without first needing to devise some way to secretly swap keys. This glaring weakness of secret-key cryptography becomes a crucial strength of RSA.

The RSA is susceptible to the following attacks:

(i) **Higher Susceptibility to Brute Force Attacks:** Keys in RSA, due to their unique nature, are more computationally costly than their counterparts in secret-key cryptography. Asymmetric keys (like those in RSA) must be many times longer compared to keys in secret-key cryptography in order to ensure equivalent security. Keys in asymmetric cryptography are also more vulnerable to brute force attacks than in secret-key cryptography.

(ii) **Known Algorithms for Faster Attacks:** There exist algorithms for public-key cryptography that allow attackers to crack private keys faster than a brute force method would require. The RSA algorithm is indeed susceptible to attacks in less than brute force time.

(iii) **Man-in-the-middle Attack:** In this situation, a malicious third party intercepts a public key on its way

CRYPTOGRAPHIC HASH FUNCTIONS, 4 THEIR APPLICATIONS

PREVIOUS YEARS QUESTIONS

PART-A

Prob.1 Write the requirements of Hash function.

Sol. Hash Function Requirements :

- (i) H can be applied to block of data of any size.
- (ii) H produces a fixed-length output.
- (iii) $H(x)$ is relatively easy to compute for any given x , making both hardware and software implementation practical.

Prob.2 What do you mean by Message Authentication Code?

Sol. Message Authentication Code : It involves the use of a secret key to generate a small block of data, known as message authentication code.

Prob.3 What is Message Padding in SHA?

Sol. Message Padding in SHA : The SHA-1 is called secure because it is computationally infeasible to find a message which corresponds to a given message digest or to find two different messages which produce the same message digest. Any change to a message in transit will, with very high probability, result in a different message digest and the signature will fail to verify.

Prob.4 Explain Message Authentication.

Sol. Message Authentication : A different requirement is to protect against active attack protection against such attacks is known as message authentication. Message authentication is a procedure that allows communicating parties to verify that received messages are authentic. The two important

aspects are to verify that the contents of the message have not been altered and that the source is authentic. We may also wish to verify a message's timeliness and sequence relative to other messages falling between two parties.

Prob.5 What are the characteristics of a good hash function?

Sol. Characteristics of a good hash function : There are four main characteristics of a good hash function:

- (i) The hash value is fully determined by the data being hashed.
- (ii) The hash function uses all the input data.
- (iii) The hash function "uniformly" distributes the data across the entire set of possible hash values.
- (iv) The hash function generates very different hash values for similar strings.

PART-B

Prob.6 (a) Explain symmetric and asymmetric authentication.

(b) Explain authentication protocols for digital signatures. [R.T.U. 2019]

Sol. (a) Definition of Symmetric Encryption : Symmetric encryption is a technique which allows the use of only one key for performing both the encryption and the decryption of the message shared over the internet. It is also known as the conventional method used for encryption.

In symmetric encryption, the plaintext is encrypted and is converted to the ciphertext using a key and an encryption algorithm. While the cipher text is converted back to plain text using the same key that was used for encryption and the

Information Security System
decryption algorithm. Symmetric encryption algorithm executes faster and is less complex hence, they are used for bulk data transmission.



Fig. : Symmetric Encryption

In symmetric encryption, the host that are participating in the communication already have the secret key that is received through the external means. The sender of the message or information will use the key for encrypting the message and the receiver will use the key for decrypting the message. The commonly used symmetric encryption algorithms are DES, 3 DES, AES, RC4.

Definition of Asymmetric Encryption : Asymmetric encryption is an encryption technique that uses a pair of key (private key and public key) for encryption and decryption. Asymmetric encryption uses the public key for the encryption of the message and the private key for the decryption of the message.

The public key is freely available to anyone who is interested in sending the message. The private key is kept secret with the receiver of the message. Any message that is encrypted by the public key and the algorithm is decrypted using the same the algorithm and the matching private key of corresponding public key.

Key Differences between Symmetric and Asymmetric Encryption

1. Symmetric encryption always uses a single key for encryption and decryption of the message. However, in asymmetric encryption, the sender uses the public key for the encryption and private key for decryption.
2. The execution of asymmetric encryption algorithms is slower as compared to the symmetric encryption algorithm. This is because the asymmetric encryption algorithms are more complex and have a high computational burden.
3. The symmetric encryption algorithms that are most commonly used are DES, 3DES, AES and RC4. On the other hand, Diffie-Hellman and RSA are the most common algorithm used for asymmetric encryption.
4. The asymmetric encryption is generally used for exchanging secret keys whereas, the symmetric encryption is used for exchanging a bulk of data.

Sol. (b) Authentication Protocols

- (i) Mutual Authentication Protocol
- (ii) One-way Authentication Protocol

• **Mutual Authentication Protocol**
This protocols enable communicating parties to satisfy themselves mutually about each other's identity and to exchange session keys.

- In this protocol, to prevent compromise of session keys, essential identification and session key information must be communicated in encrypted form.
- This protocol prevent the replay attack (threat of message replay) using timestamps or challenge/response.

• Mutual authentication follows two approaches as Symmetric Encryption approach and Public-key encryption approach.

In symmetric encryption approach :

- (1) $A \rightarrow KDC$
- (2) $KDC \rightarrow A$
- (3) $A \rightarrow B$
- (4) $B \rightarrow A$
- (5) $A \rightarrow B$

In public-key encryption approach :

- (1) $A \rightarrow AS$
- (2) $AS \rightarrow A$
- (3) $A \rightarrow B$
- (4) $B \rightarrow A$

Where, KDC is Key Distribution Center and AS is Authentication Server.

(ii) **One Way Authentication Protocol**

• It also follows two approaches as Symmetric Encryption approach and Public-key encryption approach.

In Symmetric Encryption approach :

- (1) $A \rightarrow KDC$
- (2) $KDC \rightarrow A$
- (3) $A \rightarrow B$

In Public-key encryption approach :

- $A : B \parallel E_{K_A} [H(M)]$

Prob.7 Explain SHA in detail.

OR
Explain the secure Hash Algorithm (SHA) in detail. [R.T.U. 2019, Dec. 2015]

Sol. Secure Hash Algorithm (SHA) : The algorithm takes as input a message with a maximum length of less than 2^{64} bits and produces as output a 160-bit message digest. The input is processed in 512-bit blocks.

The processing consists of the following steps :

Step 1 : Append padding bits : The message is padded so that its length is congruent to 448 modulo 512 (length = 448 mod 512). That is the length of the padded message is 64 bits less than multiple of 512-bits.

Step 2 : Append length : A block of 64 bits is appended to the message. This block is treated as an unsigned 64-bit integer and contains the length of the original message. The inclusion of a length value makes more difficult a kind of attack known as a padding attack.

Step 3 : Initialize MD buffer : A 160-bit buffer is used to hold intermediate and final results of the hash function.

The buffer can be represented as five 32-bit registers (A, B, C, D, E). These registers are initialized to the following 32-bit integers.

A = 67452301
B = EFCDA89
C = 98BADCFE
D = 10325476
E = C3D2E1F0

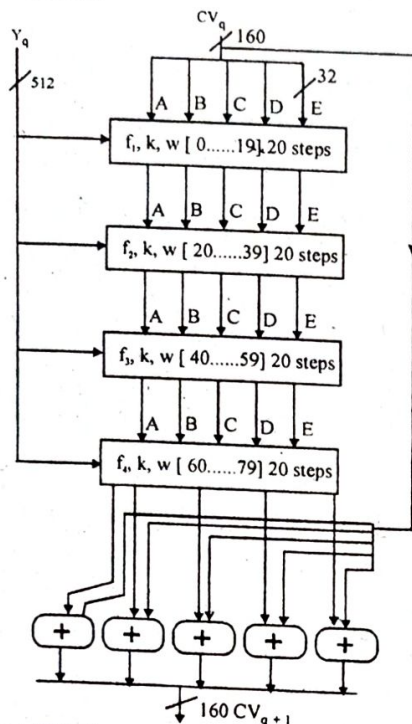


Fig. : SHA-1 processing of a single 512 bit block

Step 4 : Process message in 512-bit (16-word) blocks : The heart of the algorithm is a module known as a compression function that consists of four rounds of processing of 20 step each. Each bit uses a different primitive logical function, which we refer to as F_1 , F_2 , F_3 , and F_4 .

Step 5 : Output : After all 512-bit blocks have been processed, the output from the L^{th} stage is the 160-bit message digest.

The SHA-1 algorithm has the property that every bit of the hash code is a function of every bit of the input.

SHA uses a 160-bit encryption key. It is cryptographically stronger and recommended when security needs are higher.

The Algorithm

1. Begin by converting the message to a unique representation of the message that is a multiple of 512 bits in length, without loss of information about its exact original length in bits, as follows: append a 1 to the message. Then add as many zeroes as necessary to reach the target length, which is the next possible length that is 64 bits less than a whole multiple of 512 bits.

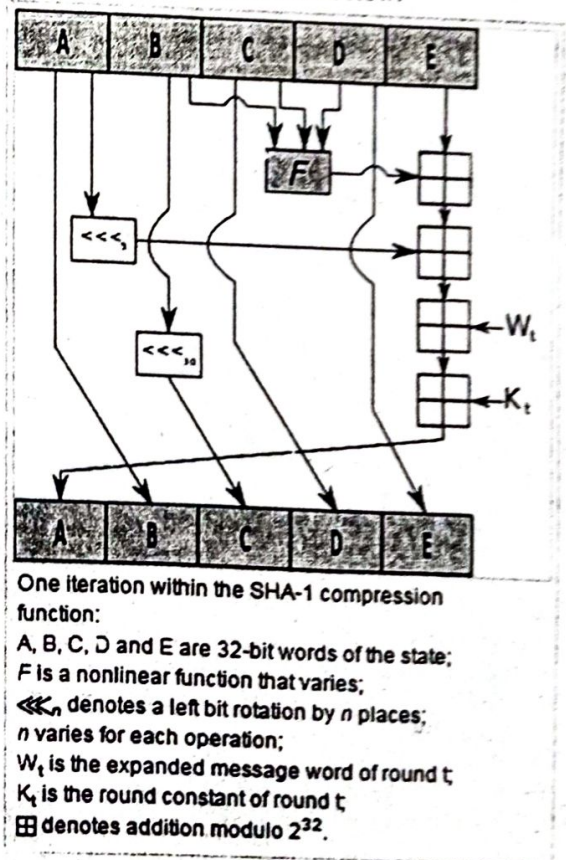
Finally, as a 64-bit binary number, append the original length of the message in bits.

2. Expand each block of 512, when it becomes time to use it, into a source of 80 32-bit subkeys as follows: the first 16 subkeys are the block itself. All remaining subkeys are generated as follows: subkey N is the exclusive OR of subkeys $N-3$, $N-8$, $N-14$, and $N-16$, subjected to a circular left shift of one place.
3. Starting from the 160-bit block value (in hexadecimal) 67452301 EFCDA89 98BADCFE 10325476 C3D2E1F0 as input for the processing of the first 512-bit block of the modified message, for each message block, do the following:

- (a) Encipher the starting value using the 80 subkeys for the current message block.
- (b) Add each of the 32-bit pieces of the ciphertext result to the starting value, modulo 2^{32} and use that result as the starting value for handling the next message block.

The starting value created at the end of the handling. The last block is the hash value, which is 160 bits long.

One iteration (i.e. processing of one message block) in the above function is illustrated below:



One iteration within the SHA-1 compression function:

A, B, C, D and E are 32-bit words of the state;
F is a nonlinear function that varies;
 $\ll n$ denotes a left bit rotation by n places;
 n varies for each operation;
 W_t is the expanded message word of round t ;
 K_t is the round constant of round t ;
 \boxplus denotes addition modulo 2^{32} .

Fig.

The Secure Hash Algorithm (SHA) was developed by the National Institute of Standards and Technology (NIST) and published as a Federal Information Processing Standard (FIPS PUB 180) in 1993; a revised version was issued as FIPS PUB 180-1 in 1995 and is generally referred to as SHA-1.

Prob.13 Write short note on digital signature.

[R.T.U. 2015]

Sol. Digital Signature : The most important development from the work on public-key cryptography is the digital signature. The digital signature provides a set of security capabilities that would be difficult to implement in any other way.

Importance : Message authentication protects two parties who exchange messages from any third party. However, it does not protect the two parties against each other. Several forms of dispute between the two are possible.

For example : Suppose that John sends an authenticated message to Mary, using one of the schemes of (figure) consider the following disputes that could arise :

(1) Mary may forge a different message and claim that it came from John. Mary would simply have to create a message and append an authentication code using the key that John and Mary share.

(2) John can deny sending the message. Because it is possible for Mary to forge a message there is no way to prove that John did in fact send the message. Both scenarios are of legitimate concern. Here is an example of the first scenario; an electronic funds transfer takes place and the receiver receives an increase in the amount of funds transferred and claims that the larger amount has arrived from the sender. An example of the second scenario is that an electronic mail message contains instructions to a stock broker for a transaction that subsequently turns out badly. The sender pretends that the message was never sent.

In situations where there is no complete trust between sender and receiver something more than authentication is needed. The most attractive solution to this problem is the digital signature. The digital signature is analogous to the handwritten signature. It must have the following properties :

(1) It must verify the author and the date and time of the signature.

(2) It must authenticate the contents at the time of the signature.

(3) It must be verifiable by third parties, to resolve disputes, thus the digital signature function includes the authentication function.

On the basis of these properties, we can formulate the following requirements for a digital signature :

(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.

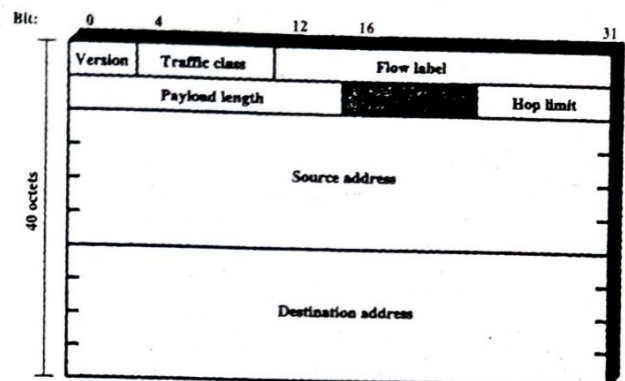


Fig. : IPv6 header

(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 exit 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.

A secure hash function, embedded in a scheme such as that of figure satisfies these requirements.

A variety of approaches has been proposed for the digital signature function.

Prob.14 Write short note on ElGamal signatures and undeniable signatures.

[R.T.U. 2015]

Sol. The ElGamal Signature Scheme

The ElGamal signature scheme is a randomized signature mechanism. It generates digital signatures with appendix on binary message of arbitrary length, and requires a hash function $h: \{0, 1\}^* \rightarrow Z_p$ where p is a large prime number. The DSA is a variant of the ElGamal signature mechanism.

(1) The ElGamal digital signature scheme generates a hint r and signature s . Selective forgery is very difficult in ElGamal digital signature scheme. Private key can be attacked, if the signer repeats the use of same random number for generating digital signature.

The fig. 2 depicts the exact working of hash based message authentication code :

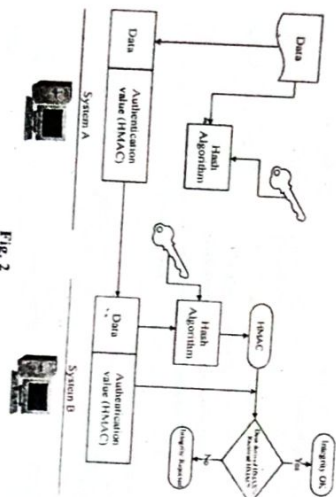


Fig. 2

Message Authentication Code (MAC) : This authentication technique involves the use of a secret key to generate a small fixed-size block of data, known as a cryptographic checksum or MAC that is appended to a message. This technique assumes that two communicating parties, say A and B, share a common secret key K. When A has a message to send to B, it calculates the MAC as a function of the message and the key:

$$MAC = C(K, M),$$

Where

M = Input message

C = MAC function

K = Shared secret key

MAC = Message authentication code

Hash Function : Refer to Prob. 11.

Prob.16 Explain the concept of hash function based on cipher block chaining.

Sol. Hash Functions Based On Cipher Block Chaining:

A number of proposals have been made for hash functions based on using a cipher block chaining technique but without using the secret key. One of the first such proposals was that of Rivin. Divide a message M into fixed-size blocks M_1, M_2, \dots, M_n and use a symmetric encryption system such as DES to compute the hash code G as

$$H_0 = \text{initial value}$$

$$H_i = E(M_i, H_{i-1})$$

$$G = H_n$$

This is similar to the CBC technique, but in this case, there is no secret key. As with any hash code, this scheme is subject to the birthday attack and if the encryption algorithm

is DES and only a 64-bit hash code is produced, then the system is vulnerable.

Furthermore, another version of the birthday attack can be used even if the opponent has access to only one message and its valid signature and cannot obtain multiple signatures. Here is the scenario. We assume that the opponent intercepts a message with a signature in the form of an encrypted hash code and that the unencrypted hash code is m bits long.

1. Use the algorithm defined at the beginning of this subsection to calculate the unencrypted hash code G .
2. Construct any desired message in the form Q_1, Q_2, \dots, Q_{N-2} .
3. Compute $H_i = E(Q_i, H_{i-1})$ for $i = 1, \dots, (N-2)$.
4. Generate $2^{m/2}$ random blocks, for each block X , compute $E(X, H_{N-2})$. Generate an additional $2^{m/2}$ random blocks; for each block Y , compute $D(Y, G)$, where D is the decryption function corresponding to E.
5. Based on the birthday paradox, with high probability there will be an X and Y.
6. Form the message $Q_1, Q_2, \dots, Q_{N-2}, X, Y$. This message has the hash code G and therefore can be used with the intercepted encrypted signature.

This form of attack is known as a meet-in-the-middle-attack. A number of researchers have proposed refinements intended to strengthen the basic block chaining approach. For example, Davies and Price describe the variation:

$$H_i = E(M_i, H_{i-1}) \oplus H_{i-1}$$

Another variation, proposed is

$$H_i = E(H_{i-1}, M_i) \oplus M_i$$

However, both of these schemes have been shown to be vulnerable to a variety of attacks. More generally, it can be shown that some form of birthday attack will succeed against any hash scheme involving the use of cipher block chaining without a secret key, provided that either the resulting hash code is small enough (e.g., 64 bits or less) or that a larger hash code can be decomposed into independent subcodes.

Thus, attention has been directed at finding other approaches to hashing. Many of these have also been shown to have weaknesses.

Prob.17 What do you mean by simple hash function?

Sol. Simple hash functions:

- For a hash function, the input is viewed as a sequence of n-bit blocks. The input is processed one block at a time in an iterative fashion to produce an n-bit hash function.

Information Security System

- One of the simplest hash functions is the bit-by-bit exclusive-OR of every block. This can be expressed as follows:

$$C_i = b_1 \oplus b_2 \oplus b_3 \oplus \dots \oplus b_m$$

where,

$$C_i = i^{\text{th}} \text{ bit of the hash code, } 1 \leq i \leq n$$

$$m = \text{number of n-bit blocks in the input}$$

$$b_j = j^{\text{th}} \text{ bit in } j^{\text{th}} \text{ block}$$

$$\oplus = \text{XOR operation}$$

- A simple way to improve matters is to perform a one bit circular shift or rotation, on the hash value after each block is processed. The procedure is as follows:
- 1. Initially set the n-bit hash value to zero.
- 2. Process each successive n-bit block of data as follows:
 - a. Rotate the current hash value to the left by one bit.
 - b. XOR the block into the hash value.

Fig shows two types of hash functions.

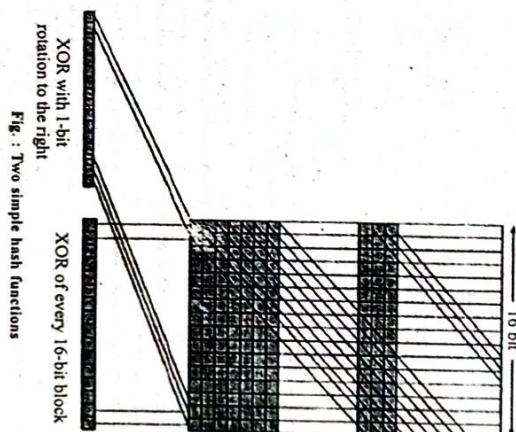


Fig. : Two simple hash functions

Prob.18 Write short note on security of hash function and MAC.

Sol. Security of Hash Function and MAC :

1. Brute-force attack
2. Cryptanalysis

1. Brute - Force attack:

(i) Hash function

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

Desirable properties:

- (a) **One-way:** For any given code h , it is computationally infeasible to find x such that $H(x) = h$.
- (b) **Weak collision resistance:** For any given block x , it is computationally infeasible to find $y \neq x$ with $H(y) = H(x)$.
- (c) **Strong collision resistance:** It is computationally infeasible to find any pair (x, y) such that $H(x) = H(y)$.

For a hash code of length n , the level of effort required, as we have seen is proportional to the following:

One way	2^n
Weak collision resistance	2^n
Strong collision resistance	$2^{n/2}$

(ii) Message Authentication Code (MAC) :

- Given one or more text MAC pair $(x, C(K, x))$ it is computationally infeasible to compute any text MAC pair $(x, C(K, x))$ for any new input $x \neq x_i$.
- The attacker would like to come up with the valid MAC code for a given message x .
- There are two lines of attack possible. Attack the key space and attack the MAC value.
- If an attacker can determine the MAC key then it is possible to generate a valid MAC value for any input x .
- An attacker can also work on the MAC value without attempting to recover the key. Here, the objective is to generate a valid MAC value for a given message or to find a message that matches a given MAC value.
- The level of effort for brute-force attack on a MAC algorithm can be expressed as $\min(2^k, 2^n)$.

2. Cryptanalysis:

Hash function:

- The hash algorithm involves repeated use of a compression function (f), that takes two inputs and produces an n-bit output.
- Cryptanalysis of hash functions focuses on the internal structure of f and is based on attempts to find efficient techniques for producing collisions for a single execution of f.

Prob.19 Write short note on Schnorr signature scheme.

Sol. The Schnorr Signature Scheme: The Schnorr signature scheme is a well-known variant of the ElGamal signature scheme. This signature also requires a hash function

$h: \{0, 1\} \rightarrow Z_q$. In this scheme, to sign a binary message m of any length, the user A ,

- chooses at random secret integer $k_A \in Z_q$.
- computes $r_A = h(g^{k_A} \bmod p, m)$ and $S_A = k_A - x_A \cdot r_A \bmod p$.

The pair (r_A, S_A) is the signature of the user A for the message m . To verify the signature, the recipient B checks the equality

$$r_A = h(g^{S_A} y^{r_A} \bmod p, m).$$

Schnorr Digital Signature Scheme-Signing and

Verifying: In this signature scheme, Alice's public key is (e_1, e_2, p, q) ; her private key (d) .

M : Message

r : Random secret

I : Concatenation

S_1, S_2 : Signatures

(d) : Alice's private key

$h(\dots)$: Hash algorithm

V : Verification

(e_1, e_2, p, q) : Alice's public key

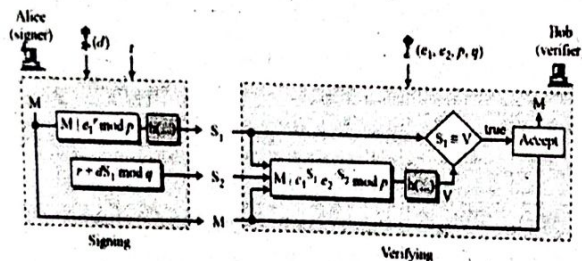


Fig.

In the signing process, two functions create two signatures; in the verifying process, the output of one function is compared to the first signature for verification.

The important point is that the scheme uses two modules: p and q .

Functions 1 and 3 use p ; function 2 uses q .

Signing

1. Alice chooses a random number r .
2. Alice calculates $S_1 = h(M \parallel e_1 \bmod p)$.
3. Alice calculates $S_2 = r + d \times S_1 \bmod q$.
4. Alice sends M , S_1 , and S_2 .

Verifying Message:

1. Bob calculates $V = h(M \parallel e_1^{S_2} \cdot e_2^{-S_1} \bmod p)$.
2. If S_1 is congruent to V module p , the message is accepted; otherwise rejected.

Prob.20 Explain the NIST digital signature algorithm.

Sol. The National Institute of Standards and Technology has revised the digital signature standard (DSS), designed to secure the identity of an electronic document signer.

number of collisions will result, cutting down on the efficiency of the hash table.

Rule 4: In real world applications, many data sets contain very similar data elements. We would like these data elements to still be distributable over a hash table.

Prob.25 What is message authentication code (MAC)? Explain types of MAC. [R.T.U. 2015]

Sol. Message Authentication Code (MAC) : Refer to Prob.15.

Types of MAC

(1) **The CBC-MAC function :** CBC-MAC means cipher block chaining message authentication code. This is a method of turning a block cipher into a MAC. The key k is used as the block cipher key. The idea behind CBC-MAC is to encrypt the message m using CBC mode and then throw away all but except the last block of cipher text. For a message $P_1, P_2, P_3 \dots P_k$, the MAC is computed as :

$$\begin{aligned} H_0 &: - IV \\ H_i &: = E_k(P_i \oplus H_{i-1}) \\ MAC &: = H_k \end{aligned}$$

Using CBC - MAC is bit tricky, but it is generally considered secure if the underlying cipher is secure. There is a number of different collision attacks on CBC-MAC that effectively limit the security to half the length of the block size [12]. Here is a simple collision attack. Let m be a CBC - MAC function. If we know that :

$m[a] = m[b]$; then we also know that
 $m[a \parallel c] = m[b \parallel c]$
 $\{ m[a \parallel c] = m = \text{name of function}$
 $a = \text{message}$
 $c = \text{key} \}$

This is due to structure of CBC - MAC. We can illustrate this with a simple case : C consists of a single block. We have :

$$\begin{aligned} m[a \parallel c] &= E_k(c \oplus m(a)) \\ m[b \parallel c] &= E_k(c \oplus m(b)) \end{aligned}$$

These two must be equal because $m(a) = m(b)$.

The attack proceeds in two stages, in the first stage the attacker collects the MAC values of a large number of messages until a collision occurs. This provides the a and b for which $m(a) = m(b)$. If the attacker can now get the sender to authenticate $a \parallel c$, he can replace the message with $b \parallel c$ without changing the MAC values. The receiver will check the MAC and accept the bogus message $b \parallel c$.

This is not a trivial attack, as it does not work with ideal MAC function. Finding collision is not the problem that can be done for an ideal MAC function in exactly the same way.

But once we have 2 messages a and b for which $m(a) = m(b)$ we cannot use them to force a MAC on a new message whereas we can do that with CBC - MAC.

We have to be careful with how we use CBC - MAC. We cannot just CBC - MAC the message itself, as that leads to simple attacks, instead, we should do the following :

- Construct a string s from the connections of l and m , where l is the length of m encoded in a fixed length format.
- Pad s until the length is a multiple of the block size.
- Apply CBC-MAC to the padded string s .
- Output the last cipher text block, or part of that block.

The advantage of CBC - MAC is that it uses the same type of computations as the block cipher encryption modes, encryption and MAC are the only two functions which are ever applied to the bulk data. So these are two speed critical areas. Having them use the same primitive functions make efficient implementations easier, especially in hardware. But at the same time CBC - MAC is difficult to use correctly.

(2) **HMAC :** If we use an HMAC function instead of just a plain hashing algorithm, a symmetric key would be concatenated with her message. The result of this process would be put through a hashing algorithm, and the result would be a MAC value. This MAC value is then appended to her message and sent to scott. If Bruce were to intercept this message and modify it, he would not have the necessary symmetric key to create the MAC value that scott will attempt to generate.

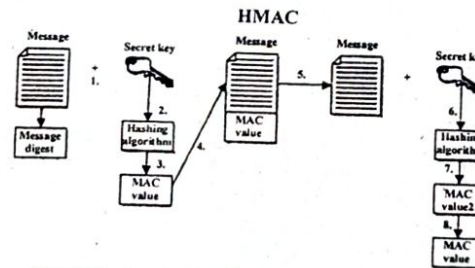


Fig. : The steps involved in using a hashing algorithm and HMAC function

The fig. shows the steps of an HMAC:

- The sender concatenates a symmetric key with the message.
- The result is put through a hashing algorithm.
- A MAC value is generated.
- The MAC value is appended to the message.
- The sender sends the message to the receiver. (Just the message with the attached MAC value. The sender does not send the symmetric key with the message.)
- The receiver concatenates a symmetric key with the message.

- The receiver puts the results through a hashing algorithm and generates her own MAC value.
- The receiver compares the two MAC values. If they are the same, the message has not been modified.

Prob.26 What is the digital signature? How authentication is accomplish using digital signature? [R.T.U. 2013]

Sol. Digital Signature : Refer to Prob.22.

Encryption and Decryption of Digital Signature : The components that a digital signature comprise of :

- Your public key :** This is the part that any one can get a copy of and is part of the verification system.
- Your name and e-mail address :** This is necessary for contact information purposes and to enable the viewer to identify the details.
- Expiration date of the public key :** This part of the signature is used to set a shelf life and to ensure that in the event of prolonged abuse of a signature eventually the signature is reset.
- Name of the company :** This section identifies the company that the signature belongs too.
- Serial number of the Digital ID :** This part is a unique number that is bundled to the signature for tracking and extra identification of reasons.
- Digital signature of the CA (Certification Authority) :** This is a signature that is issued by the authority that issues the certificates.

User A is depicted in fig. 1 and has two keys a public key, this key is available to the public for download, and a private key, this key is not available to the public. All keys are used to lock the information in an encrypted mode. The same keys are required to decrypt the data.

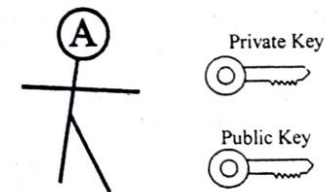


Fig. 1

Another user can encrypt the data using users A's public key. User A will use the private key to decrypt the message. Without user A's private key the data can not be decrypted. Figure 2 depicts the encryption method and decryption method and which keys are used.

Digital signature can be used to make document e-mails and other data private. The important thing is to choose a high encryption mechanism ensuring that any one attempting to decrypt the data would find it unviable to attempt decryption.

AND DISTRIBUTION

5

ARS QUESTIONS

PART-B

Prob.5 Explain Lamport's Hash in detail. [R.T.U. 2019]

OR

Explain Lamport's Hash Protocol. [R.T.U. 2018]

OR

Explain the Lamport's Hash protocol in detail. [R.T.U. 2016]

OR

Write short note on Lamport's Hash. [R.T.U. Dec. 2015]

Sol. Lamport's Hash : Lamport's hash implements a one-time password protecting against eavesdropping and password file theft.

The server stores for each user

1. the user name
2. an integer n
3. the n -fold hash of the password : $hash^n$ (password).

If the user wants to log on, the user types in the password. Her machine sends a request to the server, which answers with a prompt for n . The user's machine calculates $hash^{n-1}$ (password) and sends this to the server. The server calculates $hash(hash^{n-1}(\text{password})) = hash^n(\text{password})$. If this value matches the one on file, then the login is successful. The server replaces $hash^n(\text{password})$ with $hash^{n-1}(\text{password})$ and decrements n . When n reaches 0, the password needs to be reset.

The scheme can be improved by using a salting. The salt is stored at the server. The server then sends back both the salt and n . Salting as usual weakens the dictionary attack.

Lamport's hash is limited by the initial value for n . If n is very large, then the calculation of the initial value $hash^n(\text{password})$ is too involved, if n is small, then the scheme needs to be reset soon.

ISS.82

Lamport's hash is vulnerable to a man in the middle attack, called the *small n attack*. The attacker impersonates the server. When the client tries to authenticate, the man in the middle queries with a small n . The client answers with $hash^n$ (password). The man in the middle can calculate $hash^m$ (password) for any $m > n$.

Prob.6 Write short note on Encrypted key exchange. [R.T.U. 2019, 2018]

OR

Explain the Encrypted Key Exchange (KEK) protocol in detail.

[Note: Read (KEK) as (EKE)]

[R.T.U. 2016]

Sol. Encrypted Key Exchange (EKE): Encrypted key exchange is a protocol, or a set of rules, that allows two parties sharing a common password to communicate over an insecure network without exposing that password. The protocol was originally developed by Steven Bellovin and Michael Merritt of AT and T Bell Laboratories, who produced a seminal work on the subject.

Asymmetric and Symmetric Encryption: Encrypted key exchange involves a combination of asymmetric, or public key encryption and symmetric, or secret key, encryption. Public key encryption uses a pair of related keys - values which must be fed into a mathematical formula, or algorithm to decode an encrypted message -- one of which is known by all parties and one of which is kept private, or secret. Secret key encryption, on the other hand, uses a single secret key to encrypt and decrypt messages.

Key Derivation: In encrypted key exchange a secret key, or password, is derived from one party's public key and another party's private key. The shared secret key is then used to encrypt subsequent communication between the parties, who may have no prior knowledge of each other using a symmetric key cipher. The public and private key pairs can be generated again and again, each time the protocol is run to maintain security.

Effectiveness: Unlike classical cryptographic protocols, encrypted key exchange provides protection against active attacks, in which an attacker attempts to guess the password and more sophisticated types of online attacks, known as dictionary attacks. In a dictionary attack, an attacker tries all possible combinations of secret keys in a small set of values known as the dictionary to try to break the security of an encryption scheme by brute force.

Limitations: In its original form, the encrypted key exchange protocol required that both parties stored the shared password in an unencrypted form known as cleartext. Furthermore, the original encrypted key exchange protocol protected passwords being sent over a network, but required a trusted, third-party key distributed center. These limitations presented problems,

B.Tech. (VI Sem.) CS Solved Papers

particularly if a user logged onto a computer that didn't rely on a secure key server for authentication and led to the development of what is known as enhanced encrypted key exchange. Enhanced encrypted key exchange does not require passwords to be stored in unencrypted form; local users' passwords are protected by the file protection mechanisms of the local operating system, remote users' passwords are protected by the protocol itself and no third party is involved.

Prob.7 Describe how master secret is created from pre-master secret in SSL. [R.T.U. 2017]

Sol. The pre-master key is the value you directly obtain from the key exchange (e.g. $gab(modp)gab(modp)$) if using Diffie-Hellman). Its length varies depending on the algorithm and the parameters used during the key exchange. To make things simpler, we would want a fixed-length value to derive the keys for any cipher suite we would want to use. This is the reason behind a pre master secret. The fixed-length value we'll call master secret.

Handshaking work point in SSL:

Authentication and Pre-Master Secret

Client authenticates the server certificate. (e.g. Common Name / Date / Issuer) Client (depending on the cipher) creates the pre-master secret for the session, Encrypts with the server's public key and sends the encrypted pre-master secret to the server.

Decryption and Master Secret

Server uses its private key to decrypt the pre-master secret. Both Server and Client perform steps to generate the master secret with the agreed cipher.

Prob.8 What is the risk of using short - length keys in SSL? What type of attack can an intruder try if the keys are short? [R.T.U. 2017]

Sol. Key length defines the upper-bound on an algorithm's security (i.e., a logarithmic measure of the fastest known attack against an algorithm, relative to the key length), since the security of all algorithms can be violated by brute force attacks. Ideally, key length would coincide with the lower-bound on an algorithm's security. Indeed, most symmetric-key algorithms are designed to have security equal to their key length.

Keys are used to control the operation of a cipher so that only the correct key can convert encrypted text (ciphertext) to plaintext. Many ciphers are actually based on publicly known algorithms or are open source and so it is only the difficulty of obtaining the key that determines security of the system, provided that there is no analytic attack (i.e., a 'structural weakness' in the algorithms or protocols used),

algorithm of choice is the Data Encryption Standard (DES).

In Kerberos-speak, the KDC issues various types of tickets. Understanding these tickets is critical to understanding Kerberos. A ticket contains the keys and other information required to access network resource. One special ticket that the KDC issues is the all-important ticket-granting ticket or TGT. A TGT, which is issued when a user initially logs into the system, acts as the user's credentials. The TGT is then used to obtain (ordinary) tickets that enable access to network resources. The use of TGTs is crucial to the statelessness of Kerberos.

Each TGT contains a session key, the user ID of the user to whom the TGT is issued, and an expiration time. For simplicity, we'll ignore the expiration time, but it's worth noting that TGTs don't last forever. Every TGT is encrypted with the key K_{KDC} . Recall that only the KDC knows the key K_{KDC} . As a result, a TGT can only be read by the KDC.

Why does the KDC encrypt a user's TGT with a key that only the KDC knows and then send the result to the user? The alternative would be for the KDC to maintain a database of which users are logged in, their session keys etc. That is, the TGT would have to maintain state. In effect, TGTs provides a simple effective and secure way to distribute this database to the users. Then when, say, Alice presents her TGT to the KDC, the KDC can decrypt it and verify it, it remembers everything it needs to know about Alice.

Prob.12 Explain symmetric key distribution using symmetric encryption.

Sol. Symmetric Key Distribution Using Symmetric Encryption: For symmetric encryption to work, the two parties to an exchange must share the same key and that key must be protected from access by others. Furthermore, frequent key changes are usually desirable to limit the amount of data compromised if an attacker learns the key. Therefore, the strength of any cryptographic system rests with the key distribution technique, a term that refers to the means of delivering a key to two parties that wish to exchange data without allowing others to see the key.

Key distribution can be achieved in a number of ways. For two parties A and B, there are the following options:

1. A key could be selected by A and physically delivered to B.
2. A third party could select the key and physically deliver it to A and B.
3. If A and B have previously and recently used a key, one party could transmit the new key to the other, using the old key to encrypt the new key.
4. If A and B each have an encrypted connection to a third party C, C could deliver a key on the encrypted links to A and B.

Options 1 and 2 call for manual delivery of a key. For link encryption, this is a reasonable requirement, because each link encryption device is only going to be exchanging data with its partner on the other end of the link. However, for end-to-end encryption over a network, manual delivery is awkward. In a distributed system, any given host or terminal may need to engage in exchanges with many other hosts and terminals over time. Thus, each device needs a number of keys supplied dynamically. The problem is especially difficult in a wide-area distributed system.

Option 3 is a possibility for either link encryption or end-to-end encryption, but if an attacker ever succeeds in gaining access to one key, then all subsequent keys are revealed. Even if frequent changes are made to the link encryption keys, these should be done manually. To provide keys for end-to-end encryption, option 4 is preferable.

For option 4, two kinds of keys are used:
Session Key: When two end systems (hosts, terminals, etc.) wish to communicate, they establish a logical connection (e.g., virtual circuit). For the duration of that logical connection, called a session, all user data are encrypted with a one-time session key. At the conclusion of the session the session key is destroyed.
Permanent key: A permanent key is a key used between entities for the purpose of distributing session keys.

A necessary element of option 4 is a key distribution center (KDC). The KDC determines which systems are allowed to communicate with each other. When permission is granted for two systems to establish a connection, the key distribution center provides a one-time session key to that connection.

In general terms, the operation of a KDC proceeds as follows:

1. When host A wishes to set up a connection to host B, it transmits a connection-request packet to the KDC. The communication between A and the KDC is encrypted using a master key shared only by A and the KDC.
2. If the KDC approves the connection request, it generates a unique one-time session key. It encrypts the session key using the permanent key it shares with A and delivers the encrypted session key to A. Similarly, it encrypts the session key using the permanent key it shares with B and delivers the encrypted session key to B.
3. A and B can now set up a logical connection and exchange messages and data, all encrypted using the temporary session key.

The automated key distribution approach provides the flexibility and dynamic characteristics needed to allow a number of users to access a number of servers and for the

servers to exchange data with each other. The most widely used application that implements this approach is Kerberos, described in the next section.

Prob.13 Explain remote user authentication using asymmetric encryption.

Sol. Remote User Authentication Using Asymmetric Encryption

One approach to the use of public-key encryption for the purpose of session-key distribution, a protocol assumes that each of the two parties knows the current public key of the other.

1. A \rightarrow AS : $ID_A \parallel ID_B \parallel PU_A \parallel TU \parallel EPR_{K_A}(ID_B)$
2. AS \rightarrow A : $EPR_{K_A}(ID_B \parallel PU_A \parallel TU \parallel EPR_{K_A}(ID_B))$
3. A \rightarrow B : $ID_A \parallel ID_B \parallel PU_A \parallel TU \parallel EPR_{K_A}(ID_B)$

In this case, the central system is referred to as an authentication server (AS), provides public-key certificates. The session key is chosen and encrypted by A. The timestamps protect against replays. This protocol requires the synchronization of clocks.

1. A \rightarrow KDC : $ID_A \parallel ID_B$
2. KDC \rightarrow A : $EPR_{K_{KDC}}(ID_A \parallel ID_B)$
3. A \rightarrow B : $ID_A \parallel ID_B \parallel PU_A \parallel TU \parallel EPR_{K_{KDC}}(ID_A \parallel ID_B)$
4. B \rightarrow KDC : $ID_A \parallel ID_B \parallel PU_A \parallel TU \parallel EPR_{K_{KDC}}(ID_A \parallel ID_B)$
5. KDC \rightarrow B : $EPR_{K_{KDC}}(ID_A \parallel ID_B)$
6. B \rightarrow A : $EPR_{K_{KDC}}(ID_A \parallel ID_B)$
7. A \rightarrow B : $EPR_{K_{KDC}}(ID_A \parallel ID_B)$

Step 1: A informs the KDC of its intention to establish a secure connection with B.

Step 2: The KDC sends copy of B's public-key certificate to A.

Step 3: Using B's public key, A informs B of its desire to communicate and sends a nonce N_A .

Step 4: B asks the KDC for A's public-key certificate and requests a session key. B includes A's nonce so that the KDC can stamp the session key with that nonce. The nonce is protected using the KDC's public key.

Step 5: The KDC sends a copy of A's public-key certificate to B, plus the information (N_A, K_{AB}, ID_A) .

Step 6: The information (N_A, K_{AB}, ID_A) is forwarded to A along with nonce N_B generated by B. All the message is encrypted using A's public key.

Step 7: A retrieves the session key K_{AB} , uses it to encrypt N_B and sends it to B. This last message assures B about A's knowledge of the session key.

One-Way Authentication:

- Public-key encryption approaches including encryption of the entire message for confidentiality, authentication, or both. These approaches require that either the sender know the recipient's public key (confidentiality), the recipient knows the sender's public key (authentication), or both (confidentiality plus authentication).
- If confidentiality is the primary concern, $A \rightarrow B : EPR_{K_A}(M)$

In this case, the message is encrypted with a one-time secret key. A also encrypts this one-time key with B's public key. Only B will be able to use the corresponding private key to recover the one-time key and then use that key to decrypt the message.

If authentication is the primary concern, $A \rightarrow B : M \parallel EPR_{K_A}(HM)$

This technique allows another kind of fraud. If Bob composes a message to his boss, Alice that contains an idea that will save the company money. He appends his digital signature and sends it into the e-mail system. Suppose Max has heard of Bob's idea and gains access to the mail queue before delivery. He finds Bob's signature, removes Bob's signature, appends his own signature and delivers it to Alice. Max gets credit for Bob's idea.

To prevent this fraud, both the message and signature can be encrypted with the recipient's public key:
 $A \rightarrow B : EPR_{K_B}(M \parallel EPR_{K_B}(HM))$

Prob.14 What is HTTPS? Write its advantages.

Sol. HTTPS: HTTPS stands for Hyper Text Transfer Protocol: Secure. It is a protocol for securing the communication between two systems e.g. the browser and the web server.

The following figure illustrates the difference between communication over http and https.

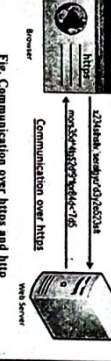
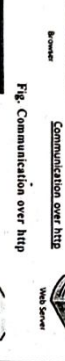


Fig. Communication over http and https

As you can see in the above figure, http transfers data between the browser and the web server in the hypertext format, whereas https transfers data in the encrypted format. Thus, https prevents hackers from reading and modifying the data during the transfer between the browser and the web server. Even if hackers manage to intercept the communication, they will not be able to use it because the message is encrypted.

HTTPS established an encrypted link between the browser and the web server using the Secure Socket Layer (SSL) or Transport Layer Security (TLS) protocols. TLS is the new version of SSL.

Advantages of https:

- Secure Communication: https makes a secure connection by establishing an encrypted link between the browser and the server or any two systems.
- Data Integrity: https provides data integrity by encrypting the data and so, even if hackers manage to intercept the data, they cannot read or modify it.
- Privacy and Security: https protects the privacy and security of website users by preventing hackers to passively listen to communication between the browser and the server.
- Faster Performance: https increases the speed of data transfer compared to http by encrypting and reducing the size of the data.
- SEO: Use of https increases SEO ranking. In Google Chrome, Google shows the Not Secure label in the browser if user's data is collected over http.
- Future: https represents the future of the web by making internet safe for users and website owners.

Part-C

Prob.15 Explain the architecture of IP security in detail.

Sol. Draw the & P security authentication header and explain the functions of each field.

OR
[R.TU. 2019]

OR
[R.TU. 2019]

OR
[R.TU. 2019]

OR
[R.TU. 2019]

OR
[R.TU. 2019]

OR
[R.TU. 2019]

OR
[R.TU. 2019]

OR
[R.TU. 2019]

OR
[R.TU. 2019]

OR
[R.TU. 2019]

IP datagrams and provides protection against replay attacks.

- Encapsulating Security Payloads (ESP) provide confidentiality, data-origin authentication, connectionless integrity, an anti-replay service (a form of partial sequence integrity), and limited traffic-flow confidentiality.

- Security Associations (SA) provide the bundle of algorithms and data that provide the parameters necessary for AH and/or ESP operations. The Internet Security Association and Key Management Protocol (ISAKMP) provides a framework for authentication and key exchange, with actual authenticated keying material provided either by manual configuration with pre-shared keys, Internet Key Exchange (IKE) and IKEv2, Kerberized Internet Negotiation of Keys (KINK), or IPSECKEY DNS records.

Authentication Header: Authentication Header (AH) is a member of the IPsec protocol suite. AH guarantees connectionless integrity and data origin authentication of IP packets. Further, it can optionally protect against replay attacks by using the sliding window technique and discarding old packets (see below).

- In IPv4, the AH protects the IP payload and all header fields of an IP datagram except for mutable fields (i.e., those that might be altered in transit), and also IP options such as the IP Security Option (RFC 1108). Mutable (and therefore unauthenticated) IPv4 header fields are DSCP/TOS, ECN, Flags, Fragment Offset, TTL and Header Checksum.

- In IPv6, the AH protects most of the IPv6 base header, AH itself, non-mutable extension headers after the AH, and the IP payload. Protection for the IPv6 header excludes the mutable fields: DSCP, ECN, Flow Label, and Hop Limit.

AH operates directly on top of IP, using IP protocol number 51.

The following AH packet diagram shows how an AH packet is constructed and interpreted.

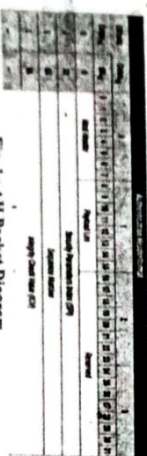


Fig. 1: AH Packet Diagram
Next Header (8 bits): Type of the next header, indicating what upper-layer protocol was protected. The value is taken from the list of IP protocol numbers.

Reserved (24 bits): The length of this Authentication Header in 4-octet units, minus 2. For example, an AH value

of 4 equals $3 \times (32\text{-bit fixed-length AH fields}) + 3 \times (32\text{-bit ICV fields}) - 2$, and thus an AH value of 4 means 24 octets. Although the size is measured in 4-octet units, the length of this header needs to be a multiple of 8 octets if carried in an IPv6 packet. This restriction does not apply to an Authentication Header carried in an IPv4 packet.

Reserved (16 bits): Reserved for future use (all zeros until then).

Security Parameters Index (32 bits): Arbitrary value which is used (together with the destination IP address) to identify the security association of the receiving party.

Sequence Number (32 bits): A monotonically increasing sequence number (incremented by 1 for every packet sent) to prevent replay attacks. When replay detection is enabled, sequence numbers are never reused, because a new security association must be renegotiated before an attempt to increment the sequence number beyond its maximum value.

Integrity Check Value (multiple of 32 bits): Variable length check value. It may contain padding to align the field to an 8-octet boundary for IPv6, or a 4-octet boundary for IPv4.

Encapsulating Security Payload: Encapsulating Security Payload (ESP) is a member of the IPsec protocol suite. In IPsec it provides origin authenticity, integrity and confidentiality protection of packets. ESP also supports encryption-only and authentication-only configurations, but using encryption without authentication is strongly discouraged because it is insecure unlike Authentication Header (AH). ESP in transport mode does not provide integrity and authentication for the entire IP packet. However, in Tunnel Mode, where the entire original IP packet is encapsulated with a new packet header added, ESP protection is afforded to the whole inner IP packet (including the inner header) while the outer header (including any outer IPv4 options or IPv6 extension headers) remains unprotected. ESP operates directly on top of IP, using IP protocol number 50.

The following ESP packet diagram shows how an ESP packet is constructed and interpreted.

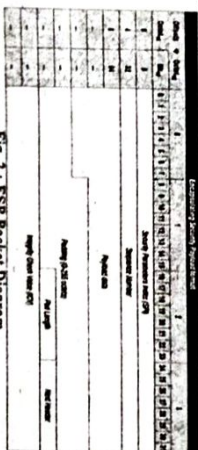


Fig. 2: ESP Packet Diagram
Security Parameters Index (32 bits): Arbitrary value used (together with the destination IP address) to identify the security association of the receiving party.

Sequence Number (32 bits): A monotonically increasing sequence number (incremented by 1 for every packet sent) to protect against replay attacks. There is a separate counter kept for every security association.

Payload Data (variable): The protected contents of the original IP packet, including any data used to protect the contents (e.g., an Initialization Vector for the cryptographic algorithm). The type of content that was protected is indicated by the Next Header field.

Padding (0-255 octets): Padding for encryption, to extend the payload data to a size that fits the encryption's cipher block size, and to align the next field.

Pad Length (8 bits): Size of the padding (in octets). Next Header (8 bits): Type of the next header. The value is taken from the list of IP protocol numbers.

Integrity Check Value (multiple of 32 bits): Variable length check value. It may contain padding to align the field to an 8-octet boundary for IPv6, or a 4-octet boundary for IPv4.

Security association: The IP security architecture uses the concept of a security association as the basis for building the security functions into IP. A security association is simply the bundle of algorithms and parameters (such as keys) that is being used to encrypt and authenticate a particular flow in one direction. Therefore, in normal bi-directional traffic, the flows are secured by a pair of security associations.

Security associations are established using the Internet Security Association and Key Management Protocol (ISAKMP). ISAKMP is implemented by manual configuration with pre-shared secrets, Internet Key Exchange (IKE) and IKEv2, Kerberized Internet Negotiation of Keys (KINK), and the use of IPSECKEY DNS records. RFC 5386 defines Better-Than-Nothing Security (BTNS) as an unauthenticated mode of IPsec using an extended IKE protocol.

In order to decide what protection is to be provided for an outgoing packet, IPsec uses the Security Parameter Index (SPI), an index to the security association database (SADB), along with the destination address in a packet header, which together uniquely identify a security association for that packet. A similar procedure is performed for an incoming packet, where IPsec gathers decryption and verification keys from the security association database.

For multicast, a security association is provided for the group, and is duplicated across all authorized receivers of the group. There may be more than one security association for a group, using different SPIs, thereby allowing multiple levels and sets of security within a group. Indeed, each sender can have multiple security associations, allowing authentication, since a receiver can only know that someone knowing the keys sent the data. Note that the relevant

standard does not describe how the association is chosen and duplicated across the group; it is assumed that a responsible party will have made the choice.

Prob.16 Explain encapsulation security payload in transport and Tunnel mode with multiple security association in detail.
[R.T.U. 2019]

OR

Write short note on Transport and Tunnel mode.
[R.T.U. Dec. 2015]

Sol. Transport Mode

Transport mode is the default mode for IPsec, and it is used for end-to-end communications (for example, for communications between a client and a server). When transport mode is used, IPsec encrypts only the IP payload. Transport mode provides the protection of an IP payload through an AH or ESP header. Typical IP payloads are TCP segments (containing a TCP header and TCP segment data), a UDP message (containing a UDP header and UDP message data), or an ICMP message (containing an ICMP header and ICMP message data).

Authentication Header transport mode

Authentication Header (AH) provides authentication, integrity, and anti-replay protection for the entire packet (both the IP header and the data payload carried in the packet). It does not provide confidentiality, which means that it does not encrypt the data. The data is readable, but protected from modification. AH uses keyed hash algorithms to sign the packet for integrity. For more information, see Data integrity with hash functions.

For example, Alice on Computer A sends data to Bob on Computer B. The IP header, the AH header, and the data are protected with integrity. This means Bob can be certain it was really Alice who sent the data and that the data was unmodified.

Integrity and authentication are provided by the placement of the AH header between the IP header and the IP payload, as shown in the following illustration.

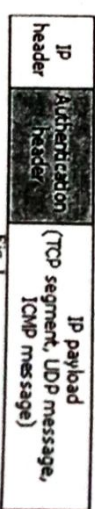


Fig. 1
AH is identified in the IP header with an IP protocol ID of 51. AH can be used alone or combined with the Encapsulating Security Payload (ESP) protocol.

The AH header contains the following fields:

- **Next Header:** Identifies the IP payload by using the IP protocol ID. For example, a value of 6 represents TCP.
- **Length:** Indicates the length of the AH header.

Information Security System

2. A rule for the inbound traffic for the tunnel. The rule for the inbound traffic is configured with a filter list that describes the traffic to be received through the tunnel and a tunnel endpoint of a local IP address (the computer or router on the local side of the tunnel). Additionally, filter actions, authentication methods, and other settings need to be specified for each rule.

For conceptual information about IPSec policy tunnel settings, see Tunnel endpoint. For information about configuring an IPSec tunnel, see Specify an IPSec tunnel. For information about how tunneling is used for virtual private networking, see Virtual private networking with IPSec.

Prob.17 Write short note on X.509.

[R.T.U. 2019]

OR

What is X.509 certificate? Differentiate between X.509 client certificate and a normal SSL certificate.

[R.T.U. Dec. 2015]

Explain the format of X.509 authentication certificate.

[R.T.U. 2018]

OR

Explain the process of distribution of secret keys using X.509 certificate.

[R.T.U. 2017]

Sol. An X.509 certificate is a digital certificate that uses the widely accepted international X.509 Public Key Infrastructure (PKI) standard to verify that a public key belongs to the user, computer or service identity contained within the certificate.

In the X.509 system, an organization wanting a signed certificate requests one via a Certificate Signing Request, (CSR).

To do this, they first generate a key pair, keeping the private key secret and using it to sign the CSR. This contains information identifying the applicant and the applicant's public key that is used to verify the signature of the CSR - and the Distinguished Name (DN), the fully qualified domain name that the certificate is for. The CSR may be accompanied by other credentials or proofs of identity required by the certificate authority.

The certification authority issues a certificate binding a public key to a particular distinguished name.

An organization's trusted root certificates can be distributed to all employees so that they can use the company PKI system. Browsers such as Internet Explorer, Firefox, Opera, Safari and Chrome come with a predetermined set of root certificates pre-installed, so SSL certificates from larger vendors will work instantly; in effect the browsers' developers determine which CAs are trusted third parties for the browsers' users.

X.509 also includes standards for Certificate Revocation List (CRL) implementations, an often neglected aspect of PKI

systems. The IETF-approved way of checking a certificate's validity is the Online Certificate Status Protocol (OCSP). Firefox 3 enables OCSP checking by default along with versions of Windows including Vista and later.

Structure of Certificate:

The structure foreseen by the standards is expressed in a formal language, Abstract Syntax Notation One (ASN.1).

The structure of an X.509 v3 digital certificate is as follows:

- Certificate
- Version Number
- Serial Number
- Signature Algorithm ID
- Issuer Name
- Validity Period
- Not Before
- Not After
- Subject Name
- Subject Public Key Info
- Public Key Algorithm
- Subject Public Key
- Issuer Unique Identifier (optional)
- Subject Unique Identifier (optional)
- Extensions (optional)
- Certificate Signature Algorithm
- Certificate Signature

Each extension has its own ID, expressed as object identifier, which is a set of values, together with either a critical or non-critical indication. A certificate-using system must reject the certificate if it encounters a critical extension that it does not recognize, or a critical extension that contains information that it cannot process. A non-critical extension MAY be ignored if it is not recognized, but MUST be processed if it is recognized.

Difference between X.509 Client Certificate and Normal SSL Certificate :

As mentioned earlier, an X.509 certificate is a digital certificate that uses the widely accepted international X.509 public key infrastructure (PKI) standard to verify that a public key belongs to the user, computer or service identity contained within the certificate.

An X.509 certificate contains information about the identity to which a certificate is issued and the identity that issued it. Standard information in an X.509 certificate includes:

1. Version – which X.509 version applies to the certificate (which indicates what data the certificate must include)
2. Serial number – the identity creating the certificate must assign it a serial number that distinguishes it from other certificates
3. Algorithm information – the algorithm used by the issuer to sign the certificate

4. Issuer distinguished name – the name of the entity issuing the certificate (usually a certificate authority)
5. Validity period of the certificate – start/end date and time
6. Subject distinguished name – the name of the identity the certificate is issued to
7. Subject public key information – the public key associated with the identity
8. Extensions (optional)

When an X.509 certificate is used for user/client authentication, it is known as X.509 Client Certificate.

On the other hand when an X.509 certificate is used for a server authentication through SSL protocol, it is known as SSL certificate.

SSL Certificates are small data files that digitally bind a cryptographic key to an organization's details. When installed on a web server, it activates the padlock and the https protocol and allows secure connections from a web server to a browser. Typically, SSL is used to secure credit card transactions, data transfer and logins, and more recently it is becoming the norm when securing browsing of social media sites.

SSL Certificates bind together:

1. A domain name, server name or hostname.
2. An organizational identity (i.e. company name) and location.

An organization needs to install the SSL Certificate onto its web server to initiate a secure session with browsers. Once a secure connection is established, all web traffic between the web server and the web browser will be secure.

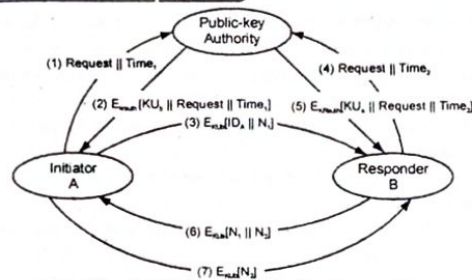


Fig. : Public Key Distribution Scenario

Prob.21 Write short note on distribution of secret keys using public key cryptosystems. [R.T.U. 2015, 2013]
OR

Explain the method of distributing secret keys. Using public-key encryption, what types of attacks can take place in the scheme? How to address them? [Raj. Univ. 2005, 2001]

Sol. Simple Secret Key Distribution : This protocol is vulnerable to an active attack. If an opponent E, has control of the intervening communication channel, then E can compromise the communication in the following sessions without being detected.

(1) Generates a public/private-key pair $\{K_V, K_P\}$ and transmits a message intended for B consisting of K_V and an identifier of A, ID_A .

(2) E intercepts the message, creates its own public/private-key pair $\{K_V, K_P\}$ and transmits K_V and ID_A to B.

(3) B generates a secret key, K_S and transmit $E_{K_V}[K_S]$.

(4) E intercepts the message, and learns K_S by computing $D_{K_P}[E_{K_V}[K_S]]$.

(5) E transmits $E_{K_V}[K_S]$ to A.

The result is that both A and B know K_S and are unaware that K_S has also been revealed to E. A and B can now exchange messages using K_S . E no longer actively interferes with communications channel but simply eavesdrops. Knowing K_S , E can decrypt all messages and both A and B are unaware of the problem. Thus this simple protocol is only useful in an environment where the only threat is eavesdropping.

Distribution of public-key : Several techniques have been proposed for the distribution of public-key. Virtually all these proposals can be grouped into the following general schemes :

- (1) Public announcement.
- (2) Public available directory.
- (3) Public-key authority.
- (4) Public-key certificates.

Anyone can forge such a public announcement. That is, some user could pretend to be user A and send a public-key to another participant or broadcast such a public-key. Until such time as user A discovers the forgery and alerts other participants. The forger is able to read all encrypted messages intended for A and can use the forged keys for authentication.

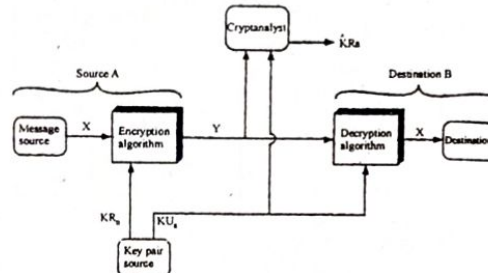


Fig. 1 : Public-Key Cryptosystem : Authentication

This scheme is clearly more secure than individual public announcements but still has vulnerabilities.

If an opponent succeeds in obtaining or computing the private-key of the directory authority, the opponent could authoritatively pass out counterfeit public-keys and subsequently impersonate any participant and eavesdrop on messages sent to any participant.

A total of seven messages are required. However, the initial four messages need be used only infrequently because both A and B can save the other's public-key for future use, a technique known as **caching**. Periodically, a user should request fresh copies of the public-keys of its correspondents to ensure currency.

In this content, the compromise of a private-key is comparable to the loss of a credit card. The owner cancels the credit card number but is at risk until all possible communications are aware that the old credit card is absolute. Thus, the time stamp serves as something like an expiration date. If a certificate is sufficiently old it is assumed to be expired.

Distribution of session keys by public-key encryption could degrade overall system performance because of the relatively high computational load of public-key encryption and decryption. With a three level hierarchy public-key encryption is used only occasionally to update the master key between a user and the KDC.

Public Key Certificate : With conventional encryption, a fundamental requirement for two parties to communicate securely is that they share a secret key. Suppose Bob wants to create a messaging application that will enable him to exchange e-mail securely with anyone who has access to the internet or to some other network that the two of them

share suppose Bob wants to do this using conventional encryption. With conventional encryption, Bob and his correspondent, say, Alice, must come up with a way to share a unique secret key that no one else knows. How are they going to do that? If Alice is in the next room from Bob, Bob could generate a key and write it down on a piece of paper or store it on a diskette and hand it to Alice. But if Alice is on the other side of the continent or the world, what can Bob do?

He could encrypt this key using conventional encryption and e-mail it to Alice, but this means that Bob and Alice must share a secret key to encrypt this new secret key. Furthermore Bob and everyone else who uses this new e-mail packages faces the same problem with every potential correspondent. Each pair of correspondents must share a unique secret key. One approach is the use of Diffie-Hellman key exchange. This approach is indeed widely used. However, it suffers, the drawback that, in its simplest form Diffie-Hellman provides no authentication of the two communicating partners.

A powerful alternative is the use of public-key certificates. When Bob wishes to communicate with Alice, Bob do the following :

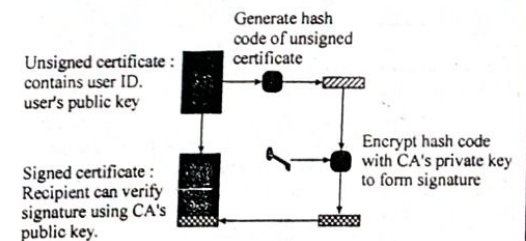


Fig. 2 : Public-Key Certificate Use

- (1) Prepare a message.
- (2) Encrypt that message using conventional encryption with a one-time conventional session key.
- (3) Encrypt the session key using public-key encryption with Alice's public-key.
- (4) Attach the encrypted session key to the message and send it to Alice.

Only Alice is capable of decrypting the session key and therefore of recovering the original message. If Bob obtained Alice's public-key by means of Alice's public-key certificate, then Bob is assured that it is the key.

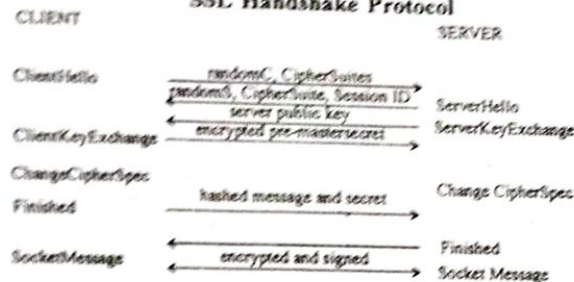
Prob.22 Write short note on SSL. [R.T.U. 2015]
OR

Why SSL is important? Explain the working of SSL using diagram, what are the difference between SSL, SET and TLS. [R.T.U. 2014]

(ii) Record Layer protocol

- Fragmentation, compression/decompression
- Encryption/decryption of message records

SSL Handshake Protocol



Prob.23 Write detailed description of public key infrastructure.

Sol. Public Key Infrastructure: A public key infrastructure, or PKI, is the sum total of everything required to securely use public keys in the real world. It's surprisingly difficult and involved to assemble all of the necessary pieces of a PKI into a working whole. For a discussion of some of the risks inherent in PKI.

A digital certificate (or public key certificate or, for short, certificate) contains a user's name along with the user's public key and this is signed by a certificate authority, or CA. For example, Alice's certificate contains

$M = ("Alice", \text{Alice's public key})$ and $S = \{M\}_{CA}$

To verify this certificate, Bob would compute $\{S\}_{CA}$ and verify that this matches M .

The CA acts as a trusted third party, or TTP. By signing the certificate, the CA is vouching for the fact it gave the corresponding private key to Alice. That is, the CA created a public and private key pair and it put the public key in Alice's certificate. Then the CA signed the certificate (using its private key) and it gave the private key to Alice. If you trust the CA, you believe that it actually gave the private key to Alice, and not to anyone else.

A subtle but important point here is that the CA is not vouching for the identity of the holder of the certificate. Certificates act as public keys and, consequently, they are public knowledge. So, for example, Trudy could send Alice's public key to Bob and claim to be Alice. Bob must not fall for this trick.

When Bob receives a certificate, he must verify the signature. If the certificate is signed by a CA that Bob trusts, then he uses that CA's public key for verification. On the other hand, if Bob does not trust the CA, then the certificate is useless to him. Anyone can create a certificate and claim to be anyone else. Bob must trust the CA and verify the signature before he can assume the certificate is valid.

To finish beating this dead horse, after verifying the signature, Bob trusts that Alice has the corresponding private key. It's critical that Bob does not assume anything more than this. For example, Bob learns nothing about the sender of the certificate – certificates are public information, so anyone could have sent it to Bob.

In addition to the required public key, a certificate could contain just about any other information that is deemed useful to the participants. However, the more information, the more likely the certificate will become invalid. For example, it might be tempting for a corporation to include the employee's department and phone number in a certificate. But then the inevitable reorganization will invalidate the certificate.

If a CA makes a mistake, the consequences can be dire. For example, VeriSign once issued a signed certificate for Microsoft to someone else, that is, VeriSign gave the private key to someone other than Microsoft. That "someone else" could then have acted (electronically, that is) as Microsoft. This particular error was quickly detected, and the certificate was revoked, apparently before any damage was done.

This raises an important PKI issue, namely, certificate revocation. Certificates are usually issued with an expiration date. But if a private key is compromised, or it is discovered that a certificate was issued in error, the certificate must be revoked immediately. Most PKI schemes require periodic distribution of certificate revocation lists, or CRLs, which are supposed to be used to filter out compromised certificates. In some situations, this could place a significant burden on users, which could lead to mistakes and security flaws.

To summarize, any PKI must deal with the following issues:

- Key generation and management
- Certificate authorities (CAs)
- Certificate revocation

Next, we'll briefly discuss a few of the many PKI trust models that are used today. Perhaps the most obvious trust model is the monopoly model, where one universally trusted organization is the CA for the known universe. This approach is naturally favored by whoever happens to be the biggest commercial CA at the time (currently, VeriSign). Some have suggested that the government should play the role of the monopoly CA. However, believe it or not many people don't trust the government.

One major drawback to the monopoly model is that it creates a big target for attack. If the monopoly CA is ever compromised, the entire PKI system fails. And if you don't trust the CA, then the system is useless for you.

The oligarchy model is one step away from the monopoly model. In this model, there are multiple trusted CAs. In fact, this is the approach that is used today – a web browser might be configured with 80 or more CA certificates. A security-conscious user such as Alice is free to decide which of the CAs she is willing to trust and which she is not. On the

other hand, a more typical user like Bob will trust whatever CAs are configured in the default settings on his browser.

At the opposite extreme from the monopoly model is the anarchy model. In this model, anyone can be a CA, and it's up to the users to decide which CAs they want to trust. In fact, this approach is used in PGP, where it goes by the name "web of trust".

The anarchy model can place a significant burden on users. For example, suppose you receive a certificate signed by Frank and you don't know Frank, but you do trust Bob and Bob says Alice is trustworthy and Alice vouches for Frank. Should you trust Frank? This is clearly beyond the patience of the average user, who is likely to simply trust everybody or nobody so as to avoid headaches like this.

There are many other PKI trust models, most of which try to provide reasonable flexibility while putting a minimal burden on end users. The fact that there is no generally agreed upon trust model is itself one of the major problems with PKI.

Prob.24 Write short note on followings:

- Kerberos login
- Kerberos ticket
- Kerberos security

Sol.(i) Kerberized Login: To understand Kerberos, let's first consider how a "Kerberized" login works, that is, we'll examine the steps that occur when Alice logs in to a system where Kerberos is used for authentication. As on most systems, Alice first enters her username and password. In Kerberos, Alice's computer then derives the key K_A from Alice's password, where K_A is the key that Alice and the KDC share. Alice's computer uses K_A to obtain Alice's TGT from the KDC. Alice can then use her TGT (i.e., her credentials) to securely access network resources. Once Alice has logged in, all of the security is automatic and takes place behind the scenes, without any additional involvement by Alice.

A Kerberized login is illustrated in figure, where the following notation is used.

- The key K_A is derived as $K_A = h(\text{Alice's password})$
- The KDC creates the session key S_A
- Alice's computer uses K_A to obtain S_A and the TGT; then Alice's computer forgets K_A
- $\text{TGT} = E("Alice", S_A, K_{KDC})$

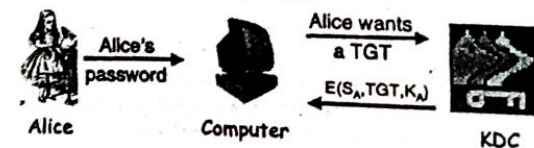


Fig. : Kerberized Login

the entire transparency reliance (ii) Ker TGT, it resource Bob. T along w timesta verifies Bob." securely acquisi the foll

I and use is the se



Alice

then s illustra authen

K_B to c The ke integri Bob.



Al Con

Kerbe sent. becom times synch