

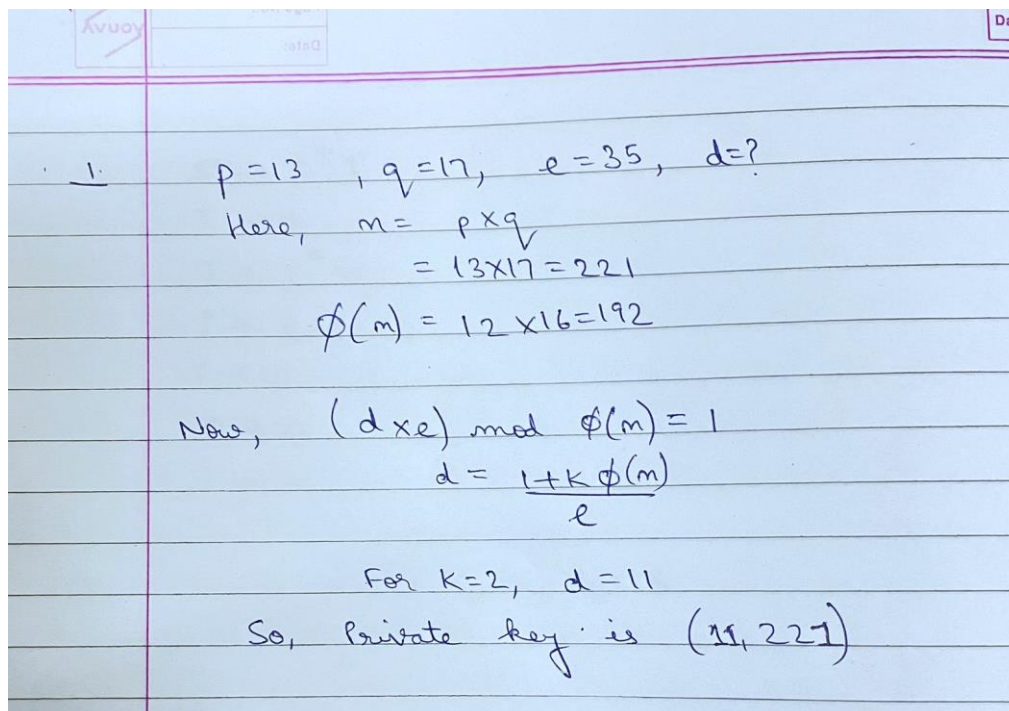
SUBJECT: INFORMATION SECURITY Theory

NAME: DHAIRYA SHAH

ROLL NO.: 21BCP010

Assignment No: 2

1. In an RSA cryptosystem, a particular A uses two prime numbers $p = 13$ and $q = 17$ to generate her public and private keys. If the public key of A is 35. Then the private key of A is?



Handwritten solution for Question 1:

1. $p = 13$, $q = 17$, $e = 35$, $d = ?$

Here, $n = p \times q$
 $= 13 \times 17 = 221$

$\phi(n) = 12 \times 16 = 192$

Now, $(d \times e) \bmod \phi(n) = 1$
 $d = \frac{1 + k\phi(n)}{e}$

For $k = 2$, $d = 11$

So, Private key is $(11, 221)$

2. Using $p=3$, $q=13$, $d=7$ and $e=3$ in the RSA algorithm, what is the value of cipher text for a plain text 5?

Given: $p=3$, $q=13$, $d=7$ and $e=3$ To find: c , for $m=5$

$$n = p \times q = 3 \times 13 = 39$$

$$c = m^e \bmod n$$

$$c = 5^3 \bmod 39$$

$$c = 8$$

Therefore, for plain text 5, cipher text is 8.

3. Suppose that two parties A and B wish to set up a common secret key (D-H key) between themselves using the Diffie Hellman key exchange technique. They agree on 7 as the modulus and 3 as the primitive root. Party A chooses 2 and party B chooses 5 as their respective secrets. What is their shared D-H key?

3: $m=7, a=3, x_a=2, x_b=5$

D-H key = ?

Public key: $y_a = a^{x_a} \bmod m$
 $= 3^2 \bmod 7$

$$y_a = 2$$

$$y_b = a^{x_b} \bmod m$$
$$= 3^5 \bmod 7$$

$$y_b = 5$$

Diffie-Hellman key:

For A: $(y_b)^{x_a} \bmod m$
 $= 5^2 \bmod 7$
 $= 4$

For B: $(y_a)^{x_b} \bmod m$
 $= 2^5 \bmod 7$
 $= 4$

Value of shared D-H key is 4

4. In a Diffie-Hellman Key Exchange, Alice and Bob have chosen prime value $q = 17$ and primitive root $= 5$. If Alice's secret key is 4 and Bob's secret key is 6, what is the secret key they exchanged?

4. $m = 17, a = 5, x_a = 4, x_b = 6$
D-H key = ?

Public key:

$$y_a = a^{x_a} \bmod m$$

$$= 5^4 \bmod 17$$

$$= \boxed{13}$$

$$y_b = a^{x_b} \bmod m$$

$$= 5^6 \bmod 17$$

$$= \boxed{2}$$

Diffie-Hellman key:

For Alice $= (y_b)^{x_a} \bmod m$

$$= 2^4 \bmod 17$$

$$= \boxed{16}$$

For Bob $= (y_a)^{x_b} \bmod m$

$$= 13^6 \bmod 17$$

$$= \boxed{16}$$

Value of shared D-H key is $\boxed{16}$

5. What is trapdoor one-way function?

A trapdoor one-way function is a one-way function with an additional requirement. Informally, a one-way function might be described as a function for which evaluation in one direction is straightforward, while computation in the reverse direction is far more difficult. Such a function becomes a trapdoor one-way function when we add the requirement that computation in the reverse direction becomes straightforward when some additional (trapdoor) information is revealed.

A trapdoor one-way function is a function f with domain X and range (codomain) Y where $f(x)$ is 'easy' to compute for all $x \in X$ but for 'virtually all' elements $y \in Y$ it is 'computationally infeasible' to find an x such that $f(x) = y$. Yet, given certain trapdoor information z , it is easy to describe an 'efficient' function g with domain Y and range X such that $g(z)(y) = x$ and $f(x) = y$.

6. Explain knapsack cryptosystem.

Knapsack is an asymmetric-key cryptosystem which requires two keys for communication: public key and private key. In knapsack public key is used only for encryption and private key is used only for decryption.

For example, if you have a knapsack that weighs 23 that has been made from the weights of the super-increasing series {1, 2, 4, 9, 20, 38} then it does not contain the weight 38 (as $38 > 23$)

but it does contain the weight 20; leaving 3;

which does not contain the weight 9 still leaving 3;

which does not contain the weight 4 still leaving 3;

which contains the weight 2, leaving 1; which contains the weight 1.

The binary code is therefore 110010.

It is much harder to decrypt a non-super-increasing knapsack problem. Give a friend a non-super-increasing knapsack and a total and see why this is the case.

7. Name 7 categories of attacks on RSA. Explain any five in detail

Possible approaches to attacking RSA are:

1. Brute Force Attack: An attacker tries all possible private keys to decrypt RSA-encrypted data, which is computationally infeasible given sufficiently large key sizes.
2. Factorization Attacks: These attacks aim to factorize the public modulus, thereby discovering the private key. Common methods include Pollard's Rho algorithm and General Number Field Sieve (GNFS).
3. Timing Attacks: Attackers analyse the time taken for RSA operations, potentially revealing sensitive information about the private key, mainly in physical implementation settings.
4. Chosen-Ciphertext Attacks (CCA): Attackers can interact with the encryption oracle to receive ciphertexts of chosen plaintexts, potentially discovering the private key. Padding oracle attacks are a type of CCA.
5. Man-in-the-Middle Attacks: Attackers intercept and manipulate RSA-encrypted communications between two parties, potentially revealing sensitive information or impersonating one of the parties.
6. Side-Channel Attacks:
7. Meet-in-the-Middle Attacks

8. Discuss the security issues in a) cipher feedback mode b) output feedback mode

a) Error Propagation: CFB mode can be sensitive to errors in the ciphertext. If an error occurs in one block of ciphertext, it can propagate and affect the decryption of subsequent blocks. This means that a single bit error can cause the entire decryption process to go awry. To mitigate this, error-detection and error-correction mechanisms may be needed.

Initialization Vector (IV) Security: The IV is a crucial parameter in CFB mode. It must be unpredictable and unique for each encryption operation. If an attacker can predict or control the IV, they may be able to compromise the security of the encryption. Reusing an IV can lead to serious vulnerabilities.

Block Size Limitation: CFB mode is tied to the block size of the underlying block cipher. If the block size is small, CFB mode may not be suitable for some applications because it cannot process data efficiently. For example, AES in CFB mode with a 128-bit block size cannot encrypt data longer than 128 bits in a single operation.

b) Reuse of the Initialization Vector (IV): The IV used in OFB mode must be unique for each encryption operation. Reusing an IV can lead to serious vulnerabilities, as it allows an attacker to make predictions about the keystream. Predictable keystreams can lead to plaintext recovery or other cryptographic attacks.

Error Propagation: OFB mode can be sensitive to errors in the ciphertext. If an error occurs in one block of ciphertext, it can propagate and affect the decryption of subsequent blocks. This means that a single bit error can cause the entire decryption process to go awry. To mitigate this, error-detection and error-correction mechanisms may be needed.

Lack of Data Integrity: OFB mode provides confidentiality by encrypting plaintext, but it does not inherently provide data integrity. An attacker could potentially tamper with the ciphertext, leading to decryption of modified or unauthorized data. To ensure both confidentiality and integrity, additional mechanisms like message authentication codes (MACs) or authenticated encryption modes should be used.

9. Explain why there is no need for ciphertext stealing in CFB, OFB, and CTR modes.

Ciphertext stealing (CTS) enables a block-cipher to process variable-length messages without expanding with padding. CTR does not rely on CTS because it, being a stream cipher, is already variable-length.

CTR does not use CTS because CTS implies a dependency on the previous block, with the initial pseudo-block being the initialization vector.

In CTR mode, the encryption of the counter produces a stream of pseudo-random bits. Encryption is XORing plaintext with that stream of bits, yielding ciphertext (and vice versa for decryption). We do not need ciphertext stealing because we can produce the stream of bits from IV alone: it has no dependency on plaintext; and we can truncate that stream of bits so that its length matches the plaintext.

The same reasoning works for OFB mode. For CFB, the stream of bits is plaintext-dependent, but for the last block it does not depend on the last plaintext block, thus we do not need ciphertext stealing either.

Ciphertext stealing is useful for CBC and ECB modes, where ciphertext is not the XOR of plaintext and some pseudo-random stream, but rather is an output of the block cipher.

10. A) What is the need of S-box? Explain two types of S-boxes.

In cryptography, an S-box (substitution-box) is a basic component of symmetric key algorithms which performs substitution. In block ciphers, they are typically used to obscure the relationship between the key and the ciphertext, thus ensuring Shannon's property of confusion. Mathematically, an S-box is a nonlinear[1] vectorial Boolean function.[2]

In general, an S-box takes some number of input bits, m , and transforms them into some number of output bits, n , where n is not necessarily equal to m . [3] An $m \times n$ S-box can be implemented as a lookup table with 2^m words of n bits each. Fixed tables are normally used, as in the Data Encryption Standard (DES), but in some ciphers the tables are generated dynamically from the key (e.g. the Blowfish and the Twofish encryption algorithms).

An S-Box, being the lone non-linear part of a block cipher, is very useful for enhancing the security of the plaintext by creating confusion in the ciphertext. The non-linearity provided by an S-Box offers defense against linear cryptanalysis [10]. Block ciphers use two types of S-Boxes: Static and dynamic.

Static s-box is defined as one particular same s-box will be used in each round of the block cipher

dynamic s-box is defined as different s-boxes will be used in each round depending on the way it been generated.

B) What is the need of D-box? How many types of D-boxes can be used in modern block ciphers?

In the context of modern block ciphers, the term "D-box" typically refers to a diffusion layer or diffusion permutation layer. The diffusion layer plays a crucial role in ensuring that changes to the input bits (e.g., plaintext, key) propagate widely and unpredictably throughout the block of data. This layer is important for achieving confusion and diffusion, two fundamental properties of secure encryption algorithms. The diffusion layer is responsible for making small changes to the input data result in significant changes in the output data, which enhances security.

The diffusion layer in block ciphers can be implemented in various ways, and there are different types of D-boxes that can be used in modern block ciphers. Here are some common types of diffusion layers:

1. Substitution-Permutation Network (SPN): SPN is a widely used structure in modern block ciphers. It typically consists of alternating layers of substitution boxes (S-boxes) and permutation layers (P-boxes). The S-boxes introduce confusion by performing non-linear substitutions, while the P-boxes introduce diffusion by rearranging bits.

2. Feistel Network: A Feistel network splits the input into two halves and processes them independently through multiple rounds, providing confusion and diffusion through repeated application of a function and bitwise XOR operations.

3. Linear Transformations: Some block ciphers use linear transformations, such as matrix multiplications or linear mixing layers, as their diffusion layer. These transformations have the property of spreading changes throughout the data but might lack non-linearity, so S-boxes are often used alongside them.

4. Bit Permutations: Bit permutations involve rearranging the bits in a specific pattern. This can be used as a diffusion layer in combination with other components like S-boxes to provide security.

The choice of the diffusion layer type and its design can significantly impact the security and performance of a block cipher. The selection depends on the specific requirements of the cipher and the cryptographic properties desired, including resistance to differential and linear cryptanalysis, avalanche effect, and diffusion properties.

It's important to note that modern block ciphers typically combine these different diffusion techniques, such as S-boxes, P-boxes, Feistel networks, and linear transformations, to create complex and secure diffusion layers. The exact structure and design of the D-box or diffusion layer vary from one cipher to another, and they are often the result of careful analysis and design to meet the desired security goals.

11. Name any 10 components used in modern block ciphers.

Substitution Boxes (S-boxes)

Permutation Boxes (P-boxes)

Feistel Network

Linear Transformations

Key Schedule

Avalanche Effect

Confusion and Diffusion Layers

Round Function

Subkey Generation

Inverse Functions

12. Differentiate between the two classes of product cipher.

Characteristic	Feistel Ciphers	Non-Feistel (SPN) Ciphers
Structure	Data block divided into two halves, processed sequentially.	Entire data block processed simultaneously.
Round Operations	One half undergoes substitution and permutation, combined with the other half.	All bits or bytes processed in parallel with substitution and permutation operations.
Key Handling	Round keys generated from the main key, often used in reverse order for decryption.	Round keys generated from the main key and applied to the entire data block in each round.

Characteristic	Feistel Ciphers	Non-Feistel (SPN) Ciphers
Confusion and Diffusion	Confusion achieved through round-specific key mixing and substitutions. Diffusion through permutations and mixing of data between halves.	Confusion through substitution operations (e.g., S-boxes) and diffusion through permutation operations (e.g., bitwise transpositions).
Examples	Data Encryption Standard (DES), Triple-DES, etc.	Advanced Encryption Standard (AES), etc.

13. Distinguish between synchronous and asynchronous stream ciphers.

Synchronous ciphers have the advantage that key stream can be pre-computed before plaintext or ciphertext is provided, often with parallelism. They have the disadvantage of ciphertext malleability (a known change in ciphertext produces a known change in plaintext) and so will need to be combined with some form of message authentication. If a transmission is only received in part, it can be difficult to recover the fragment as the exact position in the key stream needs to be identified (synchronisation). Synchronous ciphers are also deterministic constructions with a finite number of possible states and so will eventually cycle. Care must be taken in their construction/initialisation to make sure that short cycles are unlikely. For most stream ciphers (e.g. those combining key and plaintext by XOR) chosen ciphertext is equivalent to chosen plaintext in the information provided about the cipher which simplifies the security modelling. With no ciphertext dependency two different plaintexts encrypted with the same key are added to the same keystream which leads to significant weakness.

Asynchronous ciphers are largely serial in the encryption process (though not necessarily the decryption process) and the key stream can only be computed once the plaintext/ciphertext is provided. However, changing the ciphertext changes subsequent key stream and so there is considerably less malleability and a degree of message authentication. They also allow easy synchronisation at any point in transmission as the receiver can infer key stream after an initial run up of cipher bits (self-synchronisation). Asynchronous ciphers can always be provided with plaintext that prevents cycling and so no special cycle analysis is required. However chosen ciphertext has an effect on keystream and so CPA and CCA must be considered separately. Distinct plaintexts will lead to distinct keystreams, reducing the danger of repeated keys.

14. Name any two block ciphers influenced by DES:

Triple DES (3DES): Triple DES, also known as DESede, is a symmetric key block cipher that was designed to provide increased security over the original DES algorithm. It applies the Data Encryption Standard

algorithm three times (hence the name "triple") to each data block, using two or three different keys. 3DES was introduced to address the security limitations of DES, primarily its small key size.

DESX: DESX is another block cipher that is an improvement on DES. It adds an XOR operation with a public 64-bit value to the output of DES, effectively enhancing the security of the original DES algorithm. DESX was designed to mitigate some of the vulnerabilities of the standard DES encryption method.

15. Comment on the weaknesses in DES due to

a) Design of S-box

- In S-box 4, the last three output bits can be changed in the same method as the first output bit by integrating some of the input bits.
- Two particularly chosen inputs to an S-box array can generate the same output.
- It is possible to acquire the same output in an individual round by converting bits in only three neighbouring S-boxes.

b) Design of D-box

- The goals of the initial and final permutations is not clear.
- In the expansion permutation (inside the function), the first and fourth bits of every 4-bit series are done again.

c) Key size

The most serious weakness of DES is in its key size (56 bits). It can do a brute-force attack on a given ciphertext block, the adversary required to test 256 keys. There are four out of 256 possible keys are known as weak keys. A weak key is the one that, after parity drop operation includes either of all 0s, all 1s, or half 0s and half 1s. The round keys produced from some weak keys are the same and have the similar pattern as the cipher key. For instance, the sixteen round keys generated from the first key is all create of 0s; the one from the second is create of half 0s and half 1s. The reason is that the key generation algorithm first breaks the cipher key into two halves. Shifting or permutation of a block does not modify the block if it is creates of all 0s or all 1s.

16. Explain the steps in 1 round of AES with example.

AES Round Transformation:

1. SubBytes (Substitution): In this step, each byte of the 128-bit state is replaced with a corresponding byte from the S-box, a fixed 16x16 substitution table. This introduces non-linearity into the data. For example, if we have a state:

53 CA B7 76

45 38 15 AD

40 9A 0F D5

8E C2 32 6A

After SubBytes, it might become:

```
7C 84 78 25
63 4C D1 90
53 5C 22 82
A5 D3 85 E6
```

2. ShiftRows: In this step, the bytes in each row of the state are shifted left by a varying number of positions. The first row is not shifted, the second row is shifted by one position to the left, the third row by two positions, and the fourth row by three positions. This step provides diffusion and ensures that data in each row becomes spread out. For example:

```
7C 84 78 25
4C D1 90 63
22 82 53 5C
A5 D3 85 E6
```

3. MixColumns: In this step, the columns of the state are mixed by applying a linear transformation. Each byte in a column is multiplied by a fixed polynomial modulo an irreducible polynomial (in the finite field $GF(2^8)$). This operation provides further diffusion. For example:

```
D4 E0 B8 1E
BF B4 41 27
5D 52 11 98
30 AE F1 E5
```

4. AddRoundKey: In this step, a round key is XORed with the state. The round key is derived from the main encryption key and is unique for each round. It serves to introduce the secrecy of the encryption key into the state. For example, if the round key is:

```
32 88 31 E0
43 5A 31 37
F6 30 98 07
A8 8D A2 34
```

After AddRoundKey:

```
E8 68 49 2F
FF EE 00 54
AB 82 09 9F
20 59 27 D1
```

These four steps (SubBytes, ShiftRows, MixColumns, AddRoundKey) constitute one round in AES. AES operates for multiple rounds (10, 12, or 14 rounds depending on the key size) to achieve its encryption or decryption process. Each round provides additional diffusion, substitution, and key mixing to ensure the security and robustness of the encryption.

17. List the criteria defined by NIST for AES.

- **Security:** The encryption algorithm should provide a high level of security against various types of cryptanalysis, including differential and linear cryptanalysis. It should be resistant to known attacks and have a wide security margin.
- **Efficiency:** The algorithm should be computationally efficient and not overly resource-intensive. It should perform well on a wide range of computing platforms, including hardware and software implementations.
- **Simplicity:** The algorithm should have a clear and understandable design, making it easy for cryptographers to analyze and verify its security properties. A complex design could hide vulnerabilities.
- **Flexibility:** The algorithm should be flexible enough to support different key lengths and block sizes. NIST wanted a cipher that could be adapted to various security needs.
- **Wide Acceptance:** The algorithm should be well-received and widely accepted by the cryptographic community, industry, and government agencies.
- **Licensing and Intellectual Property:** The algorithm should be available for use without restrictions, such as licensing fees or patent constraints. NIST aimed for an encryption standard with broad availability.
- **Adoption of Public Algorithms:** NIST preferred algorithms that were publicly known and had undergone extensive analysis by the cryptographic community. Proprietary or secret algorithms were not considered.
- **Mathematical Soundness:** The algorithm should be based on sound mathematical principles and concepts, ensuring that its security properties are well-founded.
- **International Acceptance:** NIST sought an encryption standard that would be accepted and used internationally, not limited to the United States.

18. Find the inverse of 550 in GF(1759) using extended Euclidean Theorem.

Q	A	B	R	T1	T2	T
3	1759	550	109	0	1	-3
5	550	109	5	1	-3	16
21	109	5	4	-3	16	-339
1	5	4	1	16	-339	355
4	4	1	0	339	355	-1081
	1	0		<u>355</u>	-953	

Therefore, inverse of 550 in GF(1759) is 355.

19. Prove the secret exchange of key proposed by Diffie Hellman.

User A selects a random private integer $X_A < q$ and computes public integer

$$Y_A = \alpha^{X_A} \bmod q.$$

Similarly, user B independently selects a random private integer $X_B < q$ and computes public integer

$$Y_B = \alpha^{X_B} \bmod q.$$

Each side keeps the X value private and makes the Y value available publicly to the other side. User A computes the key as

$$K = (Y_B)^{X_A} \bmod q \text{ and}$$

User B computes the key as

$$K = (Y_A)^{X_B} \bmod q.$$

These two calculations produce identical results:

$$\begin{aligned} K &= (Y_A)^{X_B} \bmod q \\ &= (\alpha^{X_A} \bmod q)^{X_B} \bmod q \\ &= (\alpha^{X_A})^{X_B} \bmod q \\ &= \alpha^{X_B X_A} \bmod q \\ &= (\alpha^{X_B})^{X_A} \bmod q \text{ (by the rules of modular arithmetic)} \\ &= (\alpha^{X_B} \bmod q)^{X_A} \bmod q \\ &= (Y_B)^{X_A} \bmod q \end{aligned}$$

The result is that the two sides have exchanged a secret value.


20. A) Explain with an example how meet in the middle attack is possible in Diffie Hellman key exchange.

A meet-in-the-middle attack is a cryptanalysis technique used to break cryptographic schemes, like the Diffie-Hellman key exchange, by finding a common key through a two-stage process.

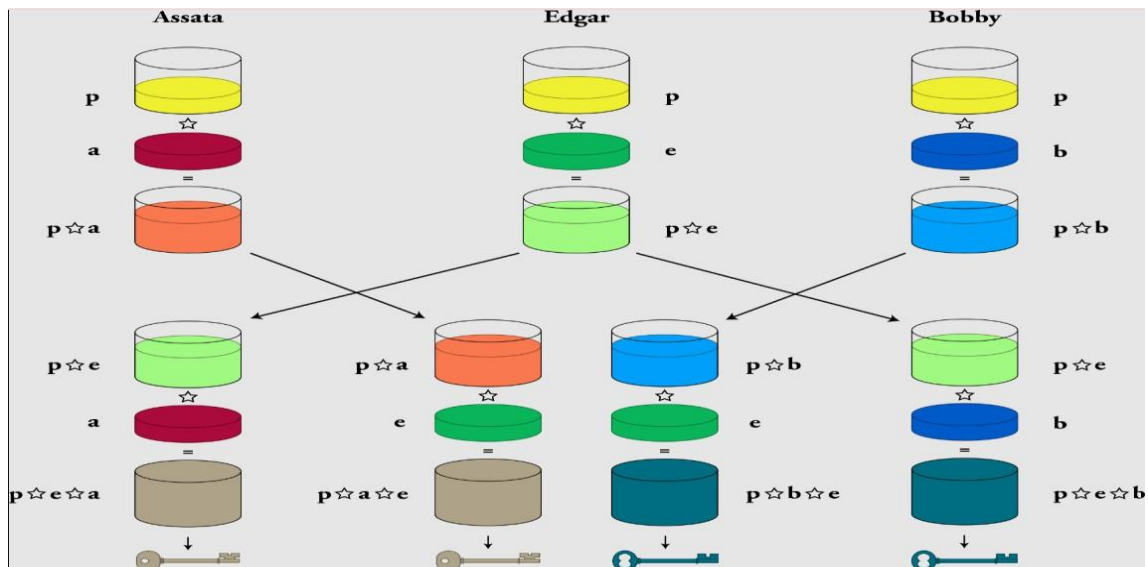
Assumptions:

- Alice and Bob want to securely exchange a secret key using the Diffie-Hellman key exchange.
- An eavesdropper, intercepts the exchanged values over an insecure channel.

B) Prove meet in the middle attack in Diffie Hellman key exchange.

Alice	Attacker	Bob
<div></div> <div>X= 3</div> <div>$A_a=7^3\text{mod}11=2$</div> <div><div><div><div></div></div></div><div><div><div></div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><div></div></div></div> <div><div><</div></div>		

Proof:



21. Describe pseudorandom number generation based on RSA.

Pseudorandom number generation uses the properties of the RSA algorithm to generate random-looking numbers that are computationally difficult to predict. Here's a high-level overview of how pseudorandom number generation based on RSA works:

1. Key Generation:

The process begins with the generation of RSA key pairs, which consist of a public key and a private key. The public key is typically used for encryption or pseudorandom number generation, while the private key is kept secret.

2. Public Key Components:

The public key consists of two primary components:

- Modulus (n): A product of two large prime numbers, which is used for various cryptographic operations.
- Public Exponent (e): A small, fixed value that is used in the encryption and pseudo random number generation processes.

3. Seed Value:

To generate pseudorandom numbers, you need an initial seed value (or message) that you want to transform into a pseudorandom sequence. This seed should be a relatively short and random value.

4. Transformation Process:

The RSA pseudorandom number generation process involves raising the seed to the power of the public exponent (e) and then taking the modulo of the result with the RSA modulus (n).

22. Illustrate ElGamal cryptographic system.

ElGamal encryption uses asymmetric key encryption for communicating between two parties and encrypting the message. It is based on the Diffie–Hellman key exchange. This cryptosystem is based on the difficulty of finding discrete logarithm in a cyclic group that is even if we know g_a and g_k , it is extremely difficult to compute g_{ak} . ElGamal is generally used to encrypt only the symmetric key (Not the plaintext). This is because asymmetric cryptosystems like ElGamal are usually slower than symmetric ones.

Let prime number $p=23$ and generator $g=7$

Choose $x=9$ and $y= g^x \bmod p= 7^9 \bmod 23=15$

Public key: $\{23, 7, 15\}$

Private key= 9 Encryption for $m=20$

Choose random $k=3$

$C_1= 7^k \bmod 23=21$

$C_2=20 \times 15^k \bmod 23=20 \times 17 \bmod 23=18$

Send $(C_1, C_2)=(21, 18)$ as a ciphertext

Decryption

$M=18 \times 21^{-9} \bmod 23 = 20$

23. On the elliptic curve over the real numbers $y^2 = x^3 - (17/12)x + 1$, let $P=(0,1)$ and $Q=(1.5,1.5)$. Find $P+Q$ and $2P$.

23. $y^2 = x^3 - \frac{17}{12}x + 1$

Let $P = (0, 1)$

$Q = (1.5, 1.5)$

Let coordinates of $P+Q$ on EC be x_3, y_3

$$x_3 = \left(\frac{y_2 - y_1}{x_2 - x_1} \right)^2 - x_1 - x_2$$

$$x_3 = \left(\frac{1.5 - 1}{1.5 - 0} \right)^2 - 0 - 1.5$$

$$x_3 = -\frac{25}{18}$$

$$\begin{aligned} y_3 &= y_1 + \left(\frac{y_2 - y_1}{x_2 - x_1} \right) (x_1 - x_3) \\ &= -1 + \left(\frac{1.5 - 1}{1.5 - 0} \right) \left(0 - \left(-\frac{25}{18} \right) \right) \\ &= -1 + \frac{1}{3} \left(\frac{25}{18} \right) \end{aligned}$$

$$= -\frac{29}{54}$$

So, $P+Q$ is $\left(-\frac{25}{18}, -\frac{29}{54} \right)$

Now,

Let co-ordinates of 2P be x_3, y_3

$$x_3 = \left(\frac{3x_1^2 + a}{2y_1} \right)^2 - 2x_1$$

$$x_3 = \left(\frac{3(0)^2 + 11}{2(1)} \right)^2 - 2(0)$$

$$x_3 = \frac{343}{516}$$

$$y_3 = -y_1 + \left(\frac{3x_1^2 + a}{2y_1} \right)(x_1 - x_3)$$

$$= -1 + \left(\frac{3(0)^2 + 11}{2(1)} \right) \left(0 - \frac{343}{516} \right)$$

$$= -1 + \left(\frac{11}{2} \right) \left(-\frac{343}{516} \right)$$

$$= \frac{-13,824 - 5831}{13,824}$$

$$= \frac{19,655}{13,824}$$

$$\text{So, } 2P = \left(\frac{343}{516}, \frac{19,655}{13,824} \right)$$

24. Solve for the elliptic curve encryption/ decryption. The cryptosystem parameters are E11(1,6) and G=(2,7). B's private key is $nB=7$.

a) Find B's public key PB

- b) A wishes to encrypt the message $P_m = (10, 9)$ and chooses the random key $k=3$. Determine the ciphertext C_m .
- c) Show the calculation by which B recovers P_m from C_m

24' $P_B = 3(2, 7)$

$$m_2 = \frac{3x_1^2 + a}{2y_1}$$

$$= \frac{3(2)^2 + 1}{2 \times 7}$$

$$= \frac{13}{14} = 2 \times (3+1) = \boxed{8}$$

$$x_2 = m_2^2 - x_1 \cdot x_1$$

$$= 8^2 - 2 \times 2 = 60$$

$$\equiv \boxed{5}$$

$$y_2 = m_2(x_1 - x_2) - y_1$$

$$= 8(2 - 5) - 7 = 31$$

$$\equiv \boxed{2}$$

$$P_B = (2, 7) + (5, 2)$$

$$m_3 = \frac{y_2 - y_1}{x_2 - x_1} = \frac{2 - 7}{5 - 2} = \frac{-5}{3}$$

$$= -5 \times (3+1)$$

$$\equiv \boxed{2}$$

$$x_3 = m_3^2 - x_1 - x_3$$

$$= (2)^2 - 2 - 5$$

$$= -3$$

$$\equiv \boxed{8}$$

$$y_3 = m_3(x_1 - x_3) - y_1$$

$$= 2(2 - 8) - 7$$

$$= -19$$

$$\equiv \boxed{13}$$

$$PB = 3(2, 7) = (8, 3)$$

$$b - KQ = 4(2, 7) = 2(5, 2)$$

$$m = \frac{3(5)^2 + 1}{2 \times 2} = 16(4-1) = 10 \times 3 \equiv 8$$

$$x = 8^2 - 2 \times 5$$

$$= 54 \equiv 10$$

$$y = 8 \times (5 - 10) - 2 = -42 \equiv 12$$

$$KQ = (10, 2)$$

$$KPB = 4(8, 3)$$

$$m = \frac{3(8)^2 + 1}{2 \times 3} = 6(6-1) = 1$$

$$x = 1^2 - 8 - 8 = -15 \equiv 7$$

$$y = 1(8-7) - 3 = -2 \equiv 9$$

$$KPB = 2(7, 9)$$

$$m = \frac{3(9)^2 + 1}{2 \times 3} = 5(7+1) = 40 \equiv 7$$

$$x = 7^2 - 7 - 7 = 35 \equiv 2$$

$$y = 7(7-2) - 9 = 26 \equiv 4$$

$$KPB = (2, 4)$$

$$P_m + KPB = (10, 9) + (2, 4)$$

$$m = \frac{4-9}{2-10} = \frac{5}{8} \equiv 6(3+1) \equiv 2$$

$$x = 2^2 - 10 - 2 = -8 \equiv 3$$

$$y = 2(10-3) - 9 \equiv 5$$

Date: _____

$$P_m + K_{PB} = (3, 5)$$

$$C_m = \{(10, 2), (3, 5)\}$$

$$Z - P_m = \{P_m + K_{PB}\} - m B(K_b)$$

$$= (3, 5) - 3(10, 2)$$

$$m = \frac{3(10)^2 + 1}{2 \times 2} = \frac{4}{4} = 1$$

$$x = 1^2 - 10 - 12 = -19 \equiv 3$$

$$y = 1(10 - 3) - 2 = 5$$

$$2(10, 2) = (3, 5)$$

$$m = \frac{5 - 2}{3 - 10} = \frac{3}{-7} = -3 \times 3 = 9$$

$$x = 9^2 - 10 - 3 = 68 \equiv 2$$

$$y = 9(10 - 2) - 2 = (2, -4) \equiv (2, 7)$$

$$P_m = (3, 5) + (2, 7)$$

$$m = \frac{7 - 5}{2 - 3} = \frac{2}{-1} = -2 \equiv 9$$

$$x_m = 9^2 - 3 - 2 = 16 \equiv 10$$

$$y = 9(3 - 10) - 5 = 68 \equiv 9$$

$$P_m = (10, 9)$$

25. You want to secretly send a message to your friend using public key cryptography.

Which one would you prefer: RSA or ECC? Justify your choice.

Here, efficiency can be prioritized considering it a personal and informal use case. In my resource-constrained environment, ECC is a preferable choice due to its smaller key sizes and faster operations.

The first reason being that ECC provides the same level of security as RSA but with significantly smaller key sizes. This makes ECC more attractive in resource-constrained environments, such as mobile devices and IoT devices. ECC is known for its efficiency. It requires shorter key lengths to achieve the same level of security as RSA. ECC operations, such as encryption, decryption, and key exchange, are faster and require fewer computational resources. This makes ECC more suitable for applications with performance constraints such as the given scenario. Smaller key sizes simplify key generation, storage, and transmission.

26. a) Identify the security service(s) offered by the models described in

i. FIGURE 1

ii. FIGURE 2

iii. FIGURE 3

b) Give suggestions to improve the cryptography model described in FIGURE 3 so that it is resistant to release of message content attack.

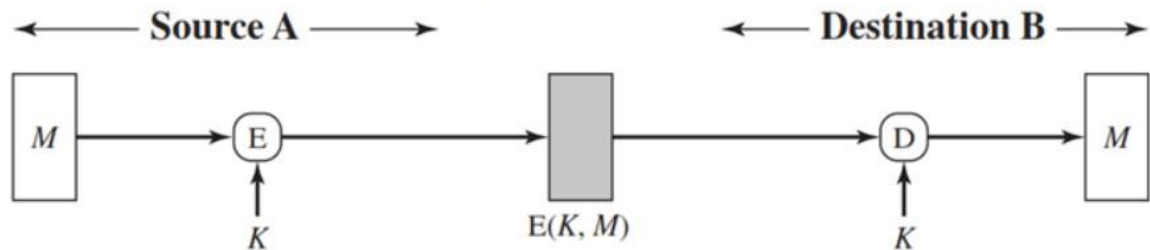


FIGURE 1

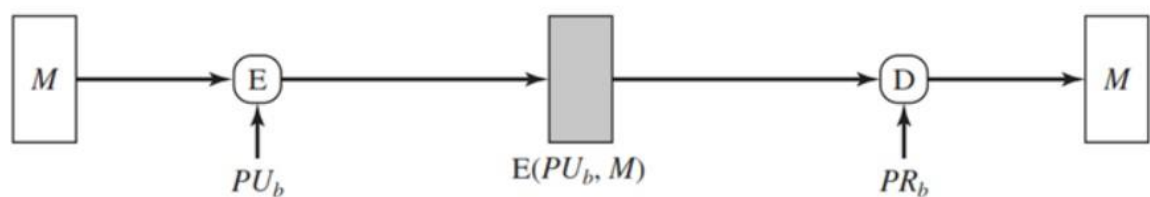


FIGURE 2

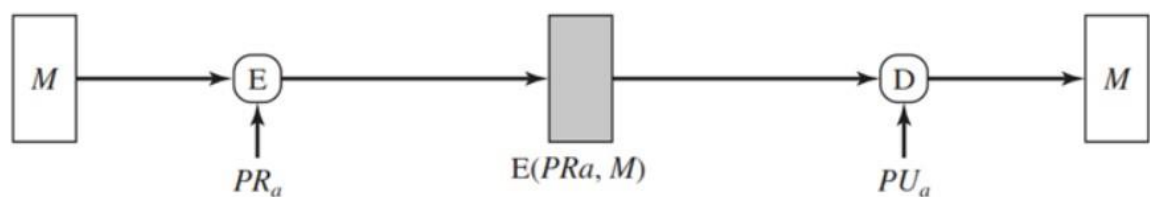


FIGURE 3

1) FIGURE 1

The figure shows symmetric encryption. Security services offered are:

- Confidentiality- Symmetric encryption secures data by encrypting it with a shared key, ensuring that only authorized parties with the key can decrypt and access the information, preserving confidentiality.
- Authentication- While not an inherent feature of symmetric encryption, authentication is often added through mechanisms like digital signatures or

message authentication codes (MACs) to confirm the identity of communication partners, preventing impersonation and tampering.

2) FIGURE 2

The figure shows public-key encryption. Security services offered are:

- Confidentiality- Public-key encryption safeguards data by allowing encryption with the recipient's public key, ensuring that only the corresponding private key holder can decrypt the information, maintaining confidentiality.

3) FIGURE 3

The figure shows public-key encryption. Security services offered are:

- Authentication- Public-key encryption enables authentication by allowing the recipient to verify the sender's identity using a digital signature. The sender signs the message with their private key, and the recipient uses the sender's public key to verify the signature. This process confirms the authenticity of the sender, ensuring secure communication.

Signature- Public-key encryption facilitates digital signatures, which provide a means to sign messages using the sender's private key. These signatures can be verified by others using the sender's public key, ensuring the integrity of the message and the authenticity of the se