Mathematics in Information Security

UNIT 4

Elgamal’s Encryption

ElGamal Encryption is a public-key cryptosystem. It uses asymmetric key encryption to communicate between two parties and encrypt the message. This cryptosystem is based on the difficulty of finding discrete logarithms in a cyclic group.

Public Key: This is used by anyone who wants to send a message securely.
Private Key: This is kept secret and is used by the recipient to decrypt the message.

Elgamal’s Encryption

1. Generating Keys:

- First, a large prime number is chosen.
- Then, a number is selected that can be used to generate other numbers based on the prime number (this is called the generator).
- The recipient (who will receive the encrypted message) picks a secret private key and calculates a public key from it. The public key is shared with everyone, but the private key is kept secret.

Elgamal’s Encryption

Encrypting the Message:

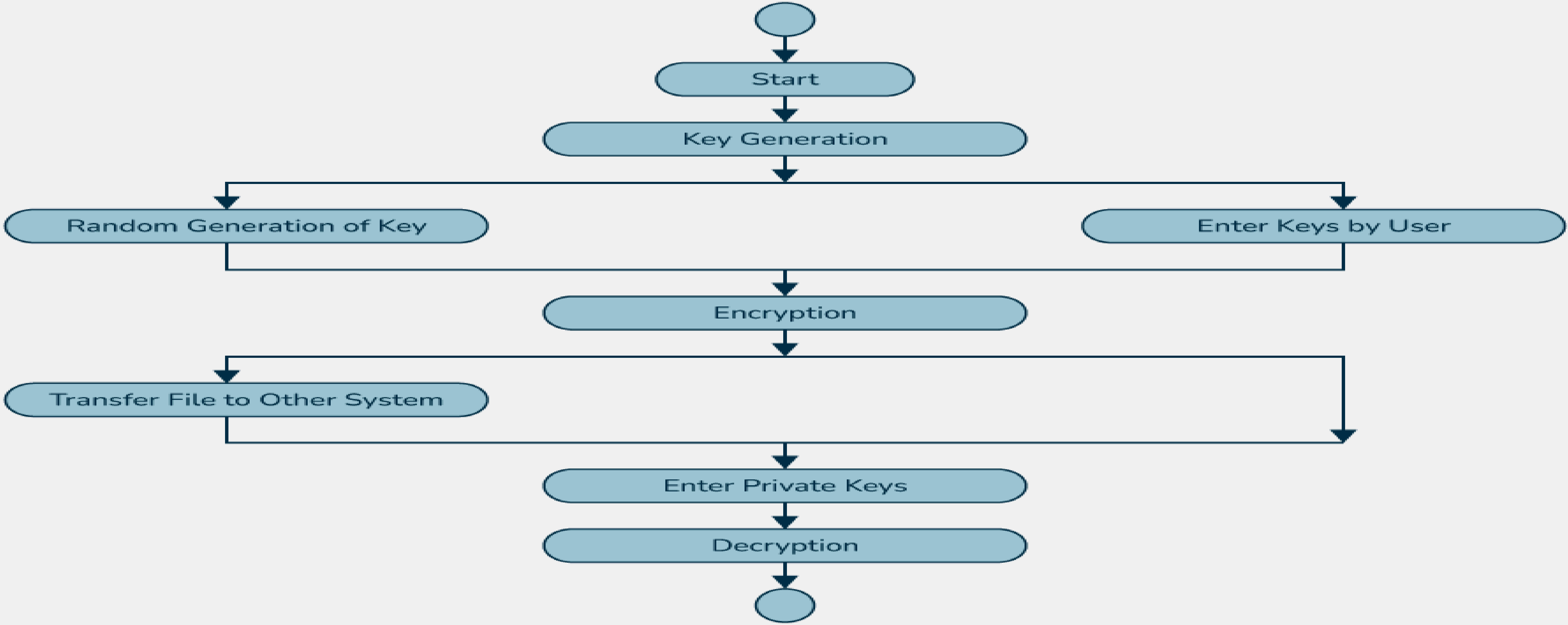
- When someone wants to send a secret message to the recipient, they use the recipient’s public key.
- They pick a random number each time they send a message to ensure that each message is encrypted in a unique way.
- Using this random number and the recipient's public key, they create two pieces of information (called ciphertext) that represent the encrypted message.

- C1 and C2 are both necessary to decrypt the message. Each part contains important information:
- C1 contains a piece that ensures the encryption is different for each message (since it involves the random k).
- C2 contains the actual message m in a transformed form, mixed with the public key and the random number.

Elgamal’s Encryption

Decrypting the Message:

- The recipient, who knows their private key, can use it to decrypt the ciphertext.



ElGamal Signature Scheme

The ElGamal Signature Scheme is a method used in cryptography to ensure that a message is both authentic and has not been altered. It relies on public-key cryptography, where two keys are used: a public key (which is shared with others) and a private key (which is kept secret by the owner).

ElGamal Signature Scheme

Key Generation:

First, a large prime number and a special number called a generator are chosen.
The private key is randomly selected, and then the public key is calculated from it using the prime and generator. The public key is shared with others, while the private key remains secret.

ElGamal Signature Scheme

Signing:

When the owner wants to sign a message, they generate a random number.
This random number is used to create part of the signature (called r).
Another part of the signature (called s) is calculated using the message itself, the private key, and the random number.
The result is a pair of numbers (r and s) that together form the signature.

Verification:

When someone else wants to check that the signature is valid, they use the public key and the message to perform a mathematical check.
If the check passes, they know that the signature is authentic, meaning the message was signed by the rightful owner and hasn’t been tampered with. If it fails, the signature is invalid.

Digital Signature Algorithm

Digital Signature is a verification method. It does not provide confidential communication. If you want to achieve confidentiality, both the message and the signature must be encrypted using either a secret key or a public key cryptosystem. This additional layer of security can be incorporated into a basic digital signature scheme.

Digital Signature Algorithm

Key Pair:

- Private Key: The person who wants to sign a message keeps a private key secret.
- Public Key: They also have a public key, which is shared with others to verify the signature.

Applications of DSA:

1. Authentication: DSA is commonly used in authentication systems to verify the identity of the sender.
2. Digital Signatures: DSA is widely used in creating digital signatures in a variety of security protocols such as SSL/TLS, digital certificates, and secure email.
3. Software and Firmware Integrity: Ensures that software has not been tampered with by verifying signatures on downloaded files or updates.

Digital Signature Algorithm

Signing the Message:

- When a message needs to be signed, it’s first turned into a fixed-size "hash" (a unique number that represents the message).
- The signer then creates a unique signature using their private key and a random number. This signature is sent along with the message.

Digital Signature Algorithm

Verifying the Signature:

- The person who receives the signed message uses the sender’s public key to verify the signature.
- They first check if the message hash matches the signature using some mathematical steps.
- If it matches, the signature is valid, meaning the message was not tampered with and came from the rightful sender.

Key Exchange and Mutual Authentication

Key exchange in cryptography is the process where two parties securely share a secret key over an insecure channel. This key can then be used for encrypting and decrypting messages between them.

1. Diffie-Hellman Key Exchange :

Idea: Two people can create a secret key together, without ever actually sending the secret key itself over the internet.

How it works: They agree on some public numbers, then each person generates a private number. They share their "public" numbers with each other and use the other person’s public number and their own private number to create the same secret key. No one else can figure out the key easily.

Key Exchange and Mutual Authentication

2. Elliptic Curve Diffie-Hellman (ECDH)

Idea: This is a faster and more secure version of Diffie-Hellman, using elliptic curve math.

How it works: Similar to Diffie-Hellman, but it uses smaller numbers for the same level of security. This makes it more efficient, especially for mobile and web applications.

3. RSA Key Exchange

Idea: One person has a public key and a private key. They send the secret key encrypted with the public key. The recipient can only decrypt it using their private key.

How it works: One person sends an encrypted secret key to the other person using their public key. The other person decrypts it with their private key.

Key Exchange and Mutual Authentication

4. Quantum Key Exchange (QKD) :

Idea: Uses the principles of quantum mechanics to ensure that no one can secretly listen in on the communication.

How it works: When someone tries to eavesdrop on the key exchange, the quantum system is disturbed, so both parties know someone is listening.

Key Exchange and Mutual Authentication

Mutual authentication in cryptography refers to a process where both parties in a communication—usually a client and a server—authenticate each other to ensure that both are who they claim to be. This is a critical aspect of secure communication, as it prevents unauthorized access and mitigates risks such as man-in-the-middle attacks.

Categories :

Client Authentication: The client proves its identity to the server. This is usually done by providing credentials (such as a username and password or a digital certificate) that the server can verify.

Server Authentication: The server also proves its identity to the client. This can be done by presenting a certificate, typically issued by a trusted certificate authority (CA), to prove that the server is legitimate.

Key Exchange and Mutual Authentication

Mutual Authentication Works:

- Initial Request: The client connects to the server and requests a connection.
- Server Identification: The server sends its digital certificate to the client. The client verifies the certificate using the public key of the certificate authority (CA).
- Client Identification: The server may then ask the client to authenticate, which could involve presenting a digital certificate or other credentials.
- Mutual Trust: After both sides authenticate each other, they proceed with the communication, often establishing an encrypted channel, like a TLS (Transport Layer Security) session.

Entity Authentication

Entity authentication in cryptography refers to the process of verifying the identity of a user, device, or system in a communication or transaction. This ensures that the entity (such as a person or system) involved in a cryptographic exchange is who they claim to be. It plays a crucial role in maintaining the integrity and security of data exchanges.

It typically involves two steps:

- Identification: One entity (the prover) presents a credential to prove its identity.
- Authentication: The other entity (the verifier) checks that the presented credential matches the expected one.

Entity Authentication

Types of Entity Authentication :

There are two main types of entity authentication mechanisms:

- One-way authentication: Only one entity is verified (e.g., a user logging into a system).
- Mutual authentication: Both parties (client and server) authenticate each other.

Station-to-Station (STS)

The Station-to-Station (STS) protocol is a cryptographic protocol designed for securely establishing a shared secret between two parties over an insecure channel.

It is based on Diffie-Hellman key exchange, but it also includes additional steps to authenticate the parties and ensure the integrity of the key exchange.

Station-to-Station (STS)

Basic Steps of the Station-to-Station Protocol:

Party A and Party B initiate the exchange:

Both parties generate their Diffie-Hellman parameters, which involve selecting private keys and corresponding public keys.

Exchange of Public Keys:

Party A sends its Diffie-Hellman public key along with its signature on the public key, verifying that it is indeed Party A's key.
Party B similarly sends its Diffie-Hellman public key and a signed message.

Station-to-Station (STS)

Verification:

Each party checks the received public key and verifies that it has been signed by the other party, ensuring authenticity and preventing a man-in-the-middle attack.

Key Computation:

Each party computes the shared secret using the received Diffie-Hellman public key and its own private key. This results in a shared secret that both parties can use for encryption.

Station-to-Station (STS)

Final Authentication and Message Exchange:

Both parties authenticate each other by verifying the signature of the other party's public key, ensuring that no one has tampered with the exchange.

Shared Secret Established:

Once the keys are authenticated, both parties can securely use the shared secret for symmetric encryption (e.g., using AES) to communicate confidentially.

Station-to-Station (STS)

Benefits of the Station-to-Station Protocol:

Resistance to Man-in-the-Middle Attacks:
The use of public key signatures ensures that the identities of both parties are verified.

Efficient Key Exchange:
It allows secure key exchange without needing a trusted third party.

Secure Communication:
After the exchange, both parties can communicate securely using the shared secret key.

Completion of Unit-4

Your Questions

THANKS