| **Course Name:** | **Information Security (116U01L602)** | **Semester:** | **VI** |
| --- | --- | --- | --- |
| **Date of Performance:** | **13/03/2025** | **DIV/ Batch No:** | **C - 3** |
| **Student Name:** | **Romil Lodaya** | **Roll No:** | **16010122096** |

| **Title: Email security using PGP implementation (Pretty Good Privacy).** |
| --- |

| **Objectives:** |
| --- |
| * Understand the fundamentals of **Pretty Good Privacy (PGP)** in securing email communication. * Learn how to **encrypt, decrypt, sign, and verify** emails using PGP. * Compare PGP with other **email security mechanisms**. * Implement **key generation, key exchange, and encryption** for secure email communication. |

| **Expected Outcome of Experiment:** |
| --- |
| **CO4:** Illustrate and Compare network security mechanisms |

| **Books/ Journals/ Websites referred:** |
| --- |
| * **"Cryptography and Network Security"** – William Stallings * <https://youtu.be/xtiWwvHL7p0> * <https://youtu.be/_GxpeZa-uZ8> |

| **Pre Lab/ Prior Concepts:** |
| --- |
| * Basics of **cryptography** (encryption, decryption, digital signatures). * Understanding **public-key and private-key cryptography**. * Familiarity with **email protocols** (SMTP, IMAP, POP3). * Basic knowledge of **command-line tools** or GUI-based PGP clients (e.g., GnuPG, Kleopatra). |

| **New Concepts to be learned:** |
| --- |
| * **PGP key pair generation** (public & private keys). * **Key management and distribution** (key servers, fingerprint verification). * **Email encryption & decryption** using PGP. * **Digital signatures** for authentication and integrity. * **Comparison of PGP vs. S/MIME vs. TLS** in email security. |

| **Abstract:** |
| --- |
| Pretty Good Privacy (PGP) is a widely used encryption protocol for **securing email communication** through **cryptographic authentication and privacy mechanisms**. This experiment demonstrates the **generation, encryption, decryption, signing, and verification** of emails using PGP. We explore **key exchange, message confidentiality, and integrity verification** in real-world email security implementations. The study also **compares PGP with other security mechanisms**, highlighting its strengths and limitations in **modern network security**. |

| **Related Theory:** |
| --- |
| **Pretty Good Privacy (PGP)** is an encryption method using a combination of **symmetric and asymmetric cryptography** to secure email communication.   * **Key Generation:** PGP generates a **public-private key pair**. The **public key** is shared, while the **private key** is kept secret. * **Encryption:** Emails are encrypted using the **receiver’s public key**, ensuring **only they can decrypt it** with their private key. * **Digital Signatures:** PGP supports **message signing**, allowing the recipient to verify the sender’s identity and ensure message integrity. * **Key Management:** Public keys are exchanged through **key servers**, and users validate authenticity via **key fingerprints**. * **Comparison:** Unlike **S/MIME**, which relies on a **central CA**, PGP uses a **web of trust model**, making it more flexible but requiring manual key verification. |

| **Implementation Details:** |
| --- |
| Text:  Encryption:    Decryption at receiver’s end:      Decryption of sender’s e-mail:        Image:  Encryption:          Decryption at receiver’s end:                  Decryption of sender’s image:            Audio:    Encryption:          Decryption at receiver’s end:    Decryption of sender’s audio: |

| **Conclusion:** |
| --- |
| PGP effectively enhances **email security** using encryption and digital signatures, ensuring **confidentiality, integrity, and authentication** in secure communications. |

**Post Lab Questions:**

Q1] In PGP, explain how Bob and Alice exchange the secret key for encrypting the messages?

In **Pretty Good Privacy (PGP)**, Alice and Bob use a combination of **asymmetric encryption** and **symmetric encryption** to securely exchange messages. Here's how the process works:

1. **Asymmetric Encryption**: Alice and Bob each have a **pair of keys**: a public key (to encrypt) and a private key (to decrypt). They each distribute their **public keys** to each other, while keeping their **private keys** secret.
2. **Message Encryption**: When Alice wants to send an encrypted message to Bob, she uses the following process:  
   * Alice generates a **random symmetric key** (also called a **session key**) that will be used to encrypt the actual message.
   * Alice encrypts the message using the **symmetric encryption** algorithm with the session key.
3. **Session Key Exchange**:  
   * To send the symmetric session key securely, Alice encrypts the **session key** itself using Bob's **public key** (asymmetric encryption).
   * Alice sends both the encrypted message and the encrypted session key to Bob.
4. **Decryption by Bob**:  
   * Bob receives the message and the encrypted session key.
   * Bob uses his **private key** to decrypt the **session key**.
   * After retrieving the session key, Bob uses it to decrypt the actual message.

This way, the secret key (the symmetric session key) is exchanged securely using asymmetric encryption, and the message itself is encrypted using symmetric encryption, ensuring both security and efficiency.

Q2] List the types of algorithms used in PGP.

PGP uses a combination of different cryptographic algorithms to achieve secure communication. The key types of algorithms used in PGP are:

1. **Symmetric Encryption Algorithms**:  
   * **IDEA (International Data Encryption Algorithm)**: Often used for encrypting the actual message because of its speed.
   * **AES (Advanced Encryption Standard)**: A more modern symmetric encryption algorithm used for encrypting messages.
2. **Asymmetric Encryption Algorithms**:  
   * **RSA (Rivest–Shamir–Adleman)**: One of the most widely used algorithms for public-key encryption and digital signatures.
   * **ElGamal**: Another public-key encryption algorithm used in some PGP implementations for key exchange and encryption.
3. **Hashing Algorithms**:  
   * **MD5 (Message Digest Algorithm 5)**: A hashing algorithm used to generate a fixed-length hash value of the message. Note: MD5 is less secure today and has been largely replaced by more secure alternatives.
   * **SHA (Secure Hash Algorithm)**: A more secure family of hashing algorithms, like **SHA-1** and **SHA-256**, used in modern implementations of PGP.
4. **Digital Signature Algorithms**:  
   * **RSA**: Can also be used to create digital signatures, ensuring message integrity and authenticity.
   * **DSA (Digital Signature Algorithm)**: Used for generating digital signatures in PGP.

Q3] Explain the significance of key rings in PGP.

In **PGP**, a **key ring** is a file or collection of files where public and private keys are stored. It is a crucial part of PGP's key management system. The significance of key rings in PGP are:

1. **Key Storage**: PGP uses key rings to store both the **public keys** (which are used for encrypting messages or verifying signatures) and the **private keys** (which are used for decrypting messages or signing messages).
2. **Separation of Keys**:  
   * **Public Key Ring**: Stores all the public keys that a user has received or generated. It allows the user to access the public keys of others to encrypt messages to them or verify their signatures.
   * **Private Key Ring**: Stores the user's private keys, which are required to decrypt messages or create digital signatures. The private keys are encrypted and protected by a passphrase.
3. **Key Management**: Key rings enable the easy management of multiple keys, allowing users to import, export, and organize their public and private keys. It helps users keep track of trusted contacts and maintain a secure method of communication.
4. **Key Trust**: Key rings also manage trust levels associated with each key. In PGP, users can assign trust levels to keys based on how they trust the authenticity of the key’s owner.

Q4] Distinguish between PGP and S/MIME.

**PGP (Pretty Good Privacy)** and **S/MIME (Secure/Multipurpose Internet Mail Extensions)** are both encryption standards used to secure email communications, but they have several key differences:

| **Aspect** | **PGP (Pretty Good Privacy)** | **S/MIME (Secure/Multipurpose Internet Mail Extensions)** |
| --- | --- | --- |
| **Encryption Type** | **PGP uses a hybrid encryption system, combining symmetric encryption for data and asymmetric encryption for key exchange.** | **S/MIME uses asymmetric encryption for both key exchange and data encryption.** |
| **Key Management** | **PGP uses a decentralized web of trust model, where users themselves verify keys and trust levels.** | **S/MIME uses a centralized model with digital certificates issued by a trusted Certificate Authority (CA).** |
| **Standardization** | **PGP is not an official standard, though it is widely adopted.** | **S/MIME is a formal standard and is based on X.509 certificates.** |
| **Certificate Authorities** | **PGP does not require a centralized certificate authority (CA); it uses trust models between users.** | **S/MIME requires trusted Certificate Authorities (CA) for issuing certificates.** |
| **User Interface** | **PGP often requires manual key management by the user, which can be complex.** | **S/MIME is integrated into email clients (like Outlook or Thunderbird), offering a more seamless user experience.** |
| **Compatibility** | **PGP is supported by many email clients, but often requires additional software or plugins.** | **S/MIME is natively supported in many enterprise-level email systems and clients.** |
| **Message Integrity** | **PGP uses hashing algorithms and digital signatures to ensure message integrity and authenticity.** | **S/MIME also uses digital signatures and hashing for message integrity.** |

In summary:

* **PGP** is more flexible but requires more manual configuration, offering a decentralized approach to key management.
* **S/MIME** is a more standardized and widely integrated protocol in enterprise systems but relies on centralized authorities for key management.