PRETTY GOOD PRIVACY

22z209	Anandkumar NS	22z242	Naveen Ragav K
22z214	Bragadeesh V	22z254	Rohith Prakash
22z231	Karthik Srinivas	22z256	Karthikeyan Sivarasu
22z233	M Raj Ragavender		

Introduction to PGP

M Raj Ragavender - 22Z233

What is PRETTY GOOD PRIVACY (PGP)

- It is an **encryption program** that provides **data security for emails, files, and digital communication**.
- PGP helps people send private and trusted messages over the Internet.
- Uses a mix of **symmetric encryption** (for speed) and **public-key encryption** (for secure key exchange).
- Ensures three main goals:
  - **Confidentiality** – only the intended person can read the message.
  - **Integrity** – message can't be altered undetected.
  - **Authentication** – verifies the sender's identity.

Pretty Good Privacy
PGP

Why It Was Created?

- PGP was developed by an American computer scientist, **Phil Zimmermann in 1991.**
- In the early 1990s, **email communication was growing**, but it was **not secure at all.**
- Anyone could **intercept, read, or modify** emails during transmission.
- Governments and organizations could easily **spy on private messages.**
- There was **no easy tool** for ordinary users to **encrypt their personal communications.**
- Phil Zimmermann wanted to give **privacy to individuals**, not just big corporations or governments.
- Hence, he created PGP to make **strong encryption accessible to everyone.**

Evolution into OpenPGP (Internet Standard)

- As PGP became widely used, multiple versions and vendors appeared.
- To ensure compatibility, the **OpenPGP standard** was developed in **1997** under the **IETF (Internet Engineering Task Force).**
- **OpenPGP** defines a universal standard for **encrypting and signing data** across platforms.
- Today, OpenPGP is used in many **email clients, servers, and security tools.**

WORKING PRINCIPLE

Karthik Srinivas S - 22Z231

WORKING PRINCIPLES

PGP (Pretty Good Privacy) works on the principle of combining symmetric and asymmetric encryption to secure data.

It ensures confidentiality, integrity, and authenticity of messages.

Instead of using just one encryption method, it uses both — to balance speed and security.

3 CASES

- Authenticity only
- Confidentiality only
- Authenticity + Confidentiality

Hybrid Cryptosystem Explained

PGP combines two encryption techniques for speed and security

Symmetric Encryption

Fast • For Data

- **Session key** generated for each message
- Encrypts message using **AES** or **CAST**
- **Very fast** — ideal for large data

Asymmetric Encryption

Secure • For Session Key

- Session key encrypted with **receiver's public key**
- Uses **RSA** or **ElGamal**
- Only **receiver's private key** can decrypt

Why Hybrid?

Symmetric encryption is fast but requires a shared key. **Asymmetric encryption** is secure but slow. PGP uses asymmetric encryption to securely share the symmetric session key, then uses that fast symmetric key to encrypt the actual message.

Role of Keys in PGP

Public Key

Shared with Everyone

Used For:

Encrypt session key and verify signatures

✉ **Example:**

Think of it like an **open mailbox**

Private Key

Kept Secret by User

Used For:

Decrypt session key and create digital signatures

Example:

Only you have the **mailbox key**

Session Key

One-Time Use

Used For:

Symmetric key to encrypt the actual message

Example:

Changes for every message

How They Work Together

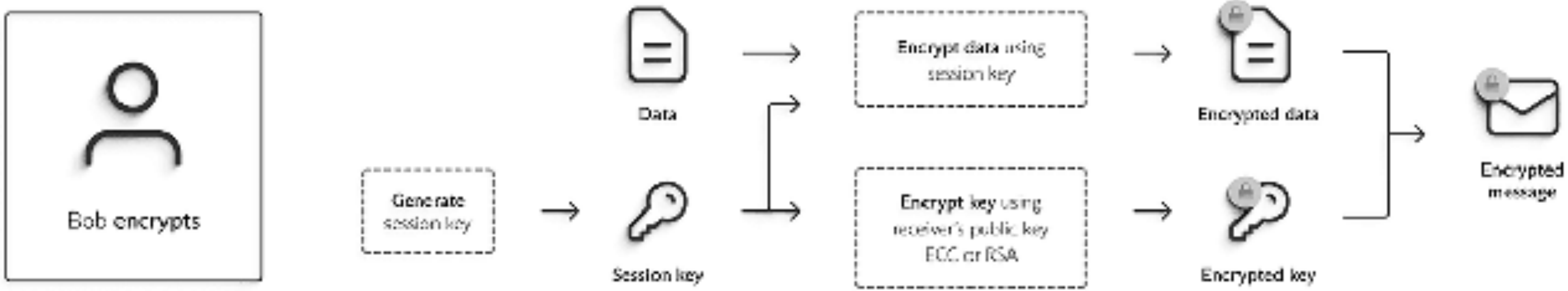
1. A random session key is generated for each message
2. The message is encrypted using the session key
3. The session key is encrypted using the recipient's public key
4. The recipient uses their private key to decrypt the session key
5. The session key is then used to decrypt the actual message

Encryption & Decryption

Anandkumar NS - 22Z209

Steps in PGP Encryption

1. **Compression** – Reduces size and adds security.
2. **Session Key Generation** – Random symmetric key for message.
3. **Symmetric Encryption** – Encrypts the actual message.
4. **Asymmetric Encryption** – Encrypts the session key using receiver’s public key.
5. **Packaging** – Combines encrypted message + encrypted key.



Bob encrypts

Generate session key

Data

Session key

Encrypt data using session key

Encrypt key using receiver's public key ECC or RSA

Encrypted data

Encrypted key

Encrypted message

1. Compression

1. Message is compressed before encryption
2. Saves space and hides text patterns
3. Common tools: ZIP, ZLIB

2. Session Key Gen

1. A random symmetric key (eg. AES) is generated
2. Each message has a unique key
3. Enables fast encryption

3. Symmetric Encryption

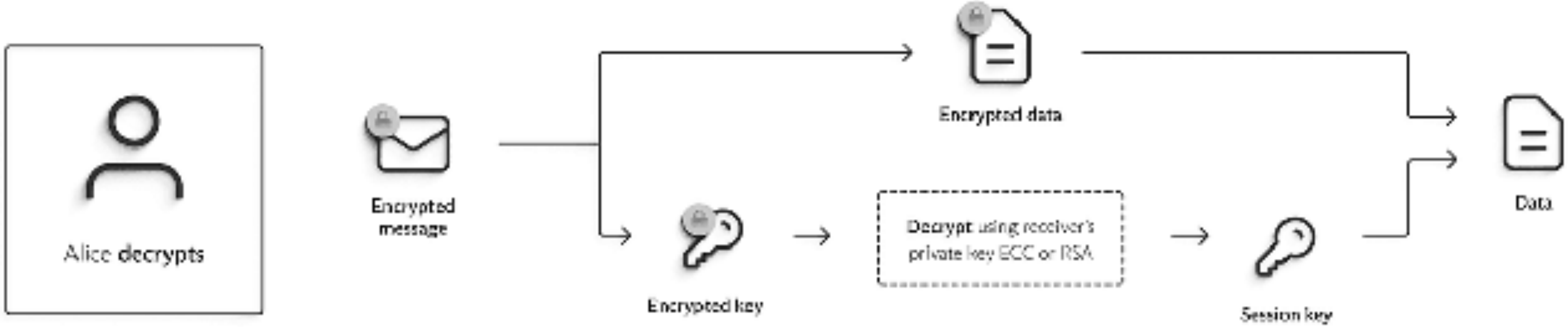
1. The message is encrypted using the session key.
2. Algorithms: AES, CAST, IDEA
3. Produces the encrypted message.

4. Asymmetric Encryption of Session Key

1. The session key is encrypted with the receiver’s public key.
2. Only receiver’s private key can unlock it.

Steps in PGP Decryption

1. **Decrypt Session Key** using private key (asymmetric).
2. **Decrypt Message** using session key (symmetric).
3. **Decompress** to restore the original message.



Alice decrypts

Encrypted message

Encrypted data

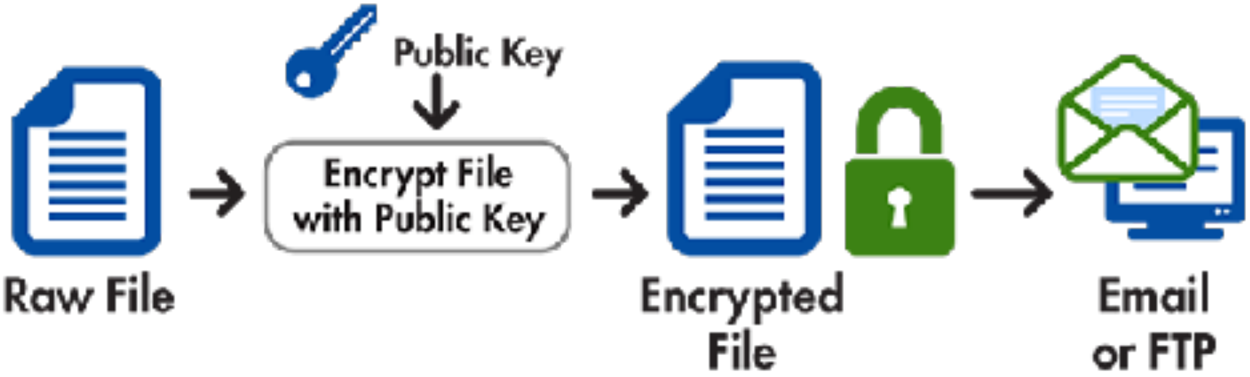
Encrypted key

Decrypt using receiver's private key ECC or RSA

Session key

Data

Encryption Process



Public Key

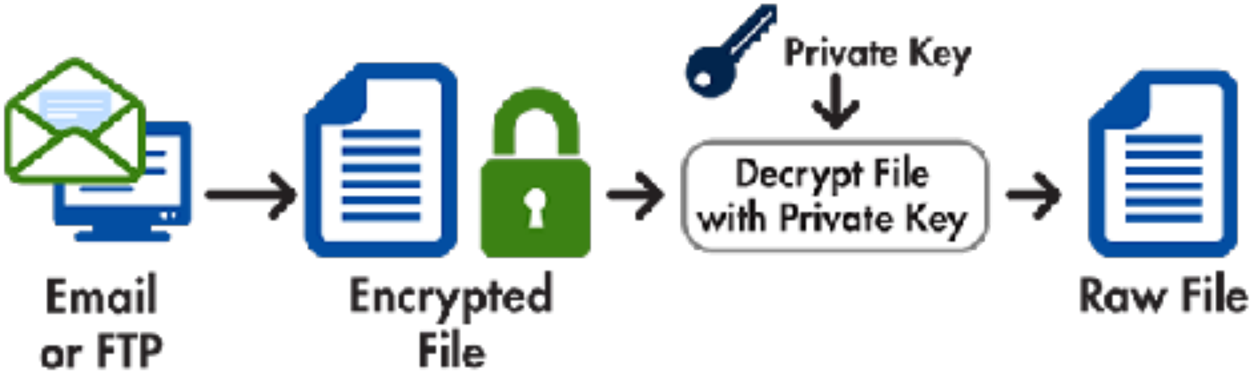
Raw File

Encrypt File with Public Key

Encrypted File

Email or FTP

Decryption Process



Private Key

Email or FTP

Encrypted File

Decrypt File with Private Key

Raw File

Digital Signature & Authentication

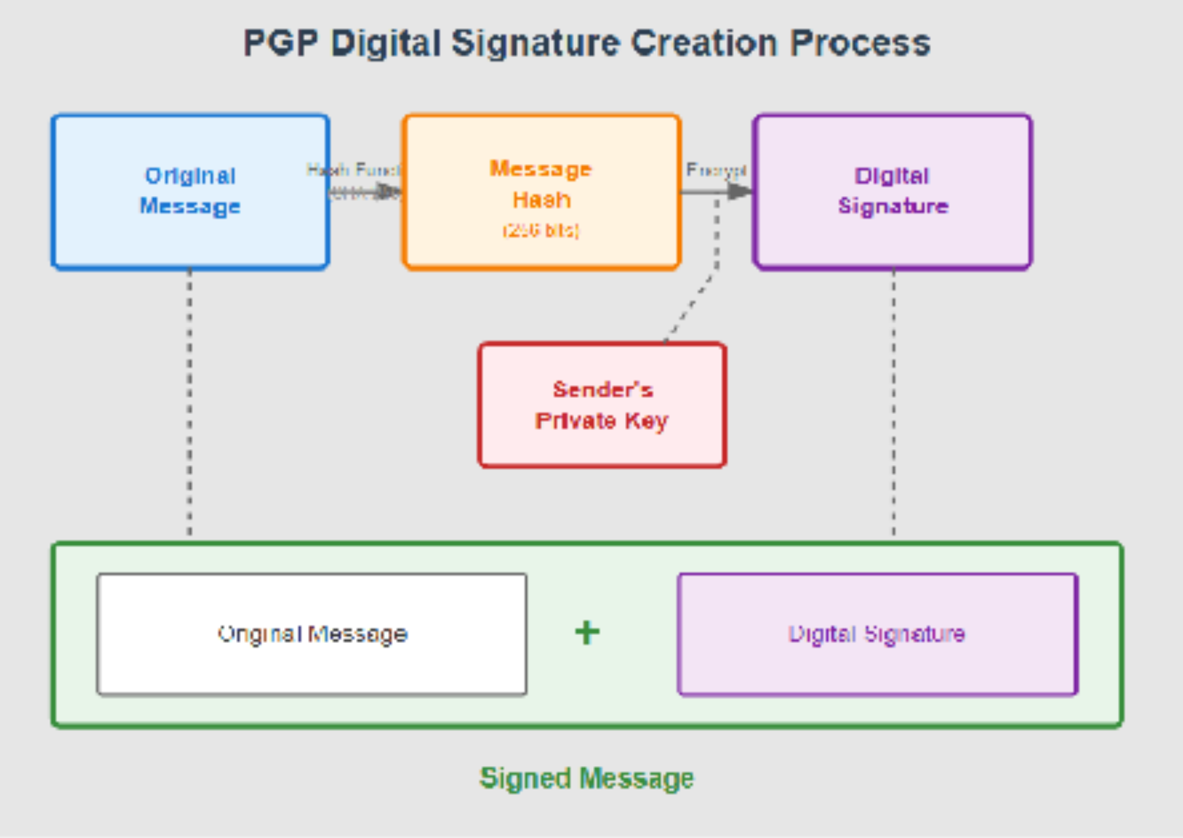
Rohith Prakash (22z254)

Introduction

A **digital signature** is a cryptographic construct that lets a recipient verify three things about a message:

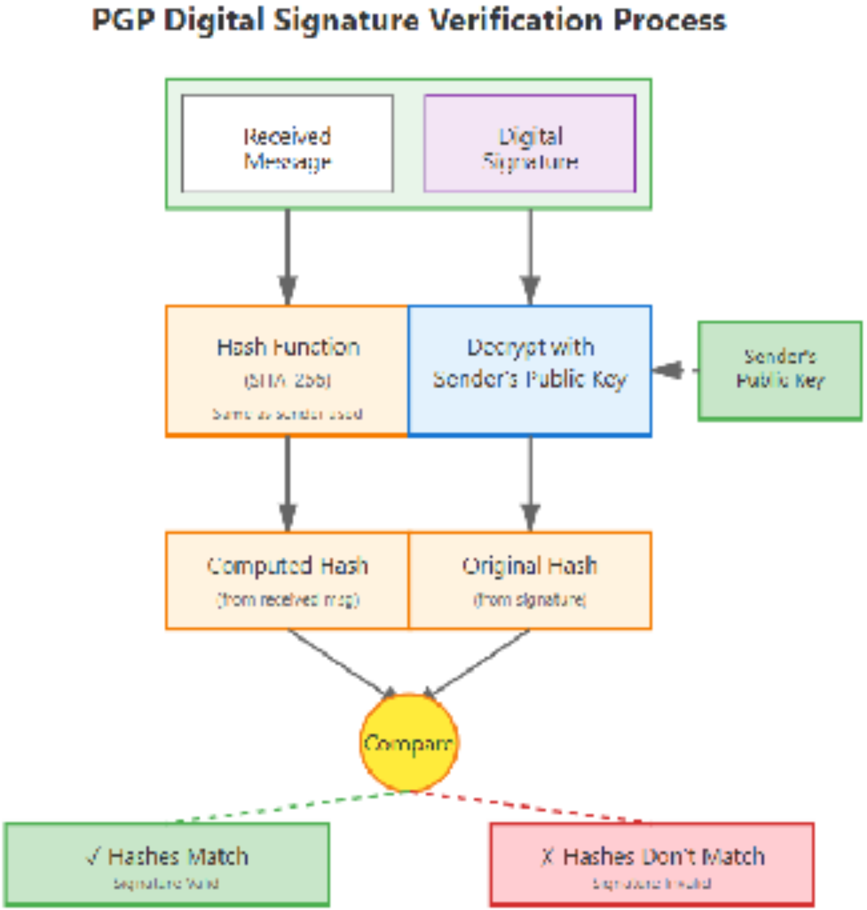
- **Authenticity** — it came from the claimed sender (because only the sender has their private key).
- **Integrity** — the message wasn’t changed after signing (a change breaks the signature).
- **Non-repudiation** — the sender cannot plausibly deny sending it (assuming their private key wasn’t compromised).

How PGP creates Digital Signatures?



- **Hash Function Selection** - SHA-256/SHA-512 for message digest
- **Private Key Operation** - Encrypting hash with sender's private key
- **Signature Attachment** - Appending signature to message
- **Cleartext vs Detached** - Different signature formats

Signature Verification Process



- **Extract Signature** - Separate signature from message
- **Decrypt with Public Key** - Recover original hash
- **Independent Hashing** - Compute fresh hash of received message
- **Hash Comparison** - Match validates authenticity & integrity

How signatures fit into PGP message flow ?

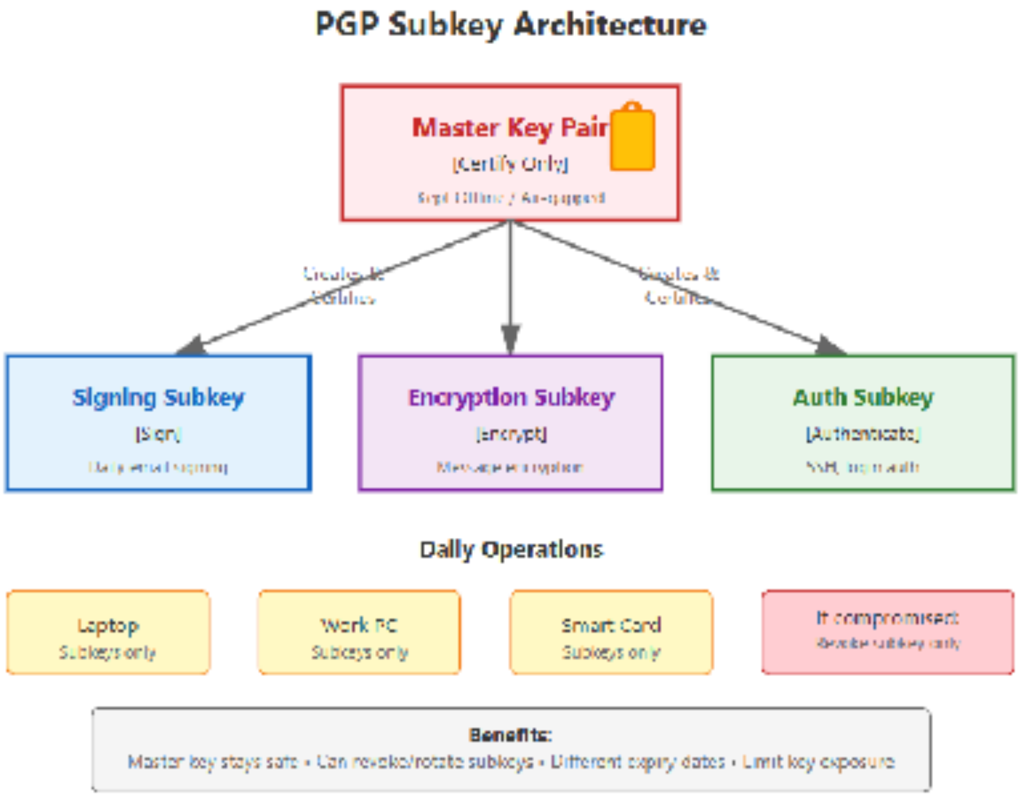
Sign → Compress (optional) → Encrypt

- Signing the plain data creates a signature tied to the original content.
- Then encrypting both the content and the signature keeps the signature confidential and prevents attackers from using signatures as metadata.
- PGP supports “one-pass” signature packets for streaming: the signer sends a one-pass signature packet, the literal data, then a signature packet.

PGP Signature Types

- Binary Signatures (0x00)-They are used for files like executables, images, or compressed data - the exact bytes are signed
- Text Signatures (0x01)- Handles the problem of different operating systems using different line endings
- Certification Signatures-Four trust levels for key signing(Generic (0x10) , Persona (0x11), Casual (0x12), Positive(0x13))
- Timestamp signatures-Proves document existed at different times

Subkey Architecture



- **Master key for identity** - Kept offline, extremely secure
- **Subkeys for daily use** - Signing, encryption, authentication
- **Key rotation without losing identity** - Replace compromised subkeys
- **Reduced exposure risk** - Master key rarely used

Key Management and

Bragadeesh V (22z214)

What is Key Management?

- Process of generating, distributing, storing, and revoking cryptographic keys
- Ensures that the right people have access and unauthorized users cannot decrypt messages.

- PGP Context:
    - Each user has a public key (shared) and a private key (kept secret).
    - Keys must be trusted before use.

Web of Trust

- A decentralized trust model used by PGP to validate public keys.
- How it works:
  - Users sign each other’s keys to vouch for authenticity.
  - Trust is transitive: if A trusts B, and B trusts C, A can have some trust in C.
- Trust is user-driven, not controlled by a central authority.

Key Generation & Exchange

- Key Generation:
  - PGP creates a public-private key pair.
  - Users choose key type (RSA, DSA, ECC) and key size.

- Key Exchange:
  - Public keys can be shared via key servers or directly between users.
  - Users verify keys before trusting them.

Trust Levels & Validation

- Trust Levels:
  - Full Trust: Key owner is fully trusted.
  - Marginal Trust: Partial trust, needs multiple confirmations.
  - Unknown Trust: Not enough information.

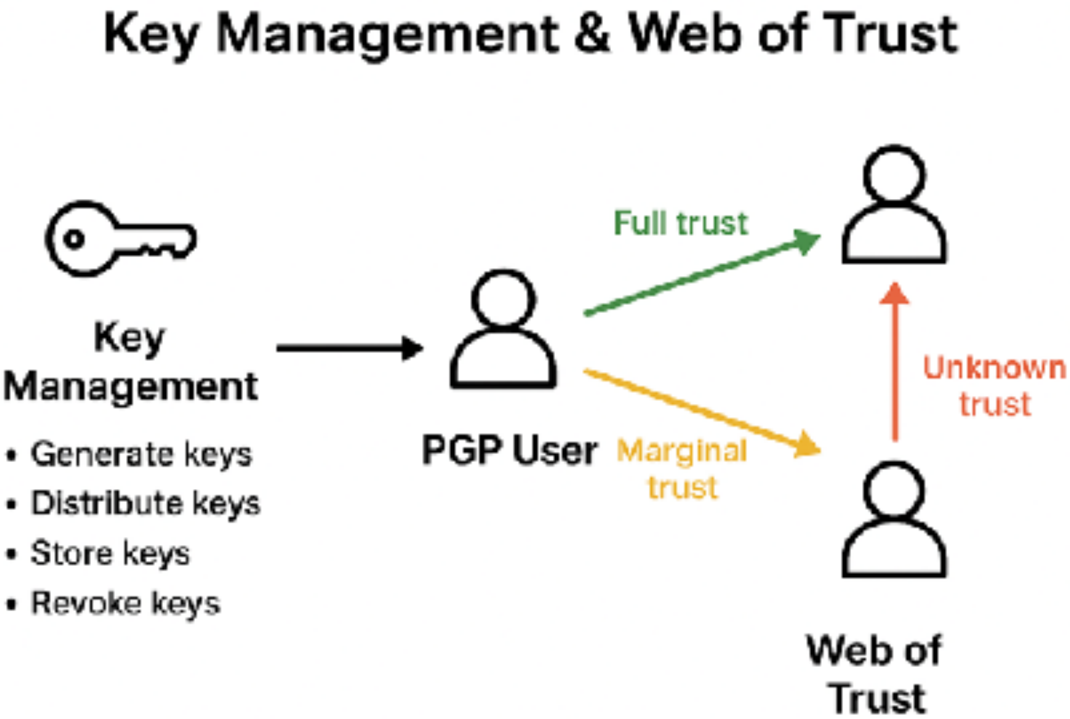
- Key Signatures:
  - Digital signatures from other trusted users confirm authenticity.

- Revocation:
  - Keys can be revoked if compromised.

Key Servers

- Publicly store and distribute public keys.
- Examples:
    - MIT PGP Key Server,
    - SKS Key Server.
- Usage:
    - Users upload their public keys.
    - Others can search and download trusted keys.

Diagram



Key Management & Web of Trust

Key Management

- Generate keys
- Distribute keys
- Store keys
- Revoke keys

Real-World Use Cases and Applications

K.Naveen Ragav

1.Secure Email Communication

- Encrypts and signs emails to ensure privacy and authenticity.
- Prevents unauthorized access or message tampering.
- Used by journalists, businesses, and government agencies.
- Tools: ProtonMail, Tutanota, Outlook (with Gpg4win).

2.File and Data Encryption

- Protects files and folders from unauthorized access.
- Used for securing backups, research data, and cloud uploads.
- Ensures data remains confidential even if stolen.
- Tools: Gpg4win, Kleopatra, Symantec Encryption Desktop.

3.Software Distribution and Code Signing

- Verifies the authenticity of software and updates.
- Prevents tampering during downloads or installations.
- Widely used in open-source projects (Linux, Git).
- Tools: GPG, Debian package signing, Homebrew verification.

4.Secure Communication for Activists & Whistleblowers

- Ensures private and anonymous communication.
- Protects users in journalism or activism from surveillance.
- Maintains message integrity and confidentiality.
- Tools: SecureDrop, Tor, Freedom of the Press Foundation.

3 uses of PGP Encryption

Encrypting Emails

Digital Signature Verification

Encrypting Files

Advantages,Limitations and Future of PGP

Karthikeyan Sivarasu(22z256)

- **End-to-End Armor: Ensures only sender and receiver can read the message.**

- **Digital Trust Seal: Uses signatures to verify identity and prevent tampering.**

- **Hybrid Power: Combines speed of symmetric keys with security of asymmetric keys.**

- **Platform Free: Works across emails, files, and networks — flexible protection.**

- **Privacy Ownership: No dependency on centralized authorities; user holds the keys.**

Limitations of PGP

- **Complex Setup: Key generation and management can be confusing for non-tech users.**

- **No Key Recovery: Lose your private key — lose your access forever.**

- **Trust Web Confusion: “Web of Trust” can be tricky to build and maintain.**

- **Compatibility Hurdles: Not all email clients or systems support PGP natively.**

- **Performance Dip: Encryption and decryption take noticeable processing time.**

Future of PGP

- **Seamless Integration: Built into apps and browsers for one-click encryption.**

- **Quantum-Resistant Keys: Adapting to next-gen encryption to beat quantum attacks.**

- **User-Centric Privacy: Simplified interfaces and automatic key handling.**

- **Cloud-Ready Security: Secure file sharing and backups via decentralized storage.**

- **Zero-Knowledge Networks: PGP evolving for encrypted collaboration and messaging.**

Do You Know??

- ***PGP was written in just 5 months by Phil Zimmermann in 1991 — and it became the world’s most popular encryption tool!***

- ***PGP was once considered a “weapon” by the U.S. government — exporting it was illegal under arms control laws!***

- ***Modern PGP algorithms (like RSA 4096-bit) are trillions of times more secure than traditional password systems.***

THANKS FOR YOUR PATIENT LISTENING !!! ☺️