process of creating, distributing, and securing the cryptographic keys used for encryption and digital signatures.

This is a critical process for safeguarding encrypted communications and verifying the authenticity of the sender.

Key management revolves around a public-private key pair:

Public key [1]:

Shared openly with others. Anyone can use your public key to encrypt a message for you or to verify your digital signature.

Private key [2]:

Kept secret and protected by a strong passphrase. You use it to decrypt messages sent to you and to create digital signatures.

We discuss 4 Core PKI key management areas below.

1. Key generation

Create key pairs:

Use software like :abbr:`GnuPG (Gnu Privacy Guard)` (:abbr:`GPG (Gnu Privacy Guard)`) to generate a key pair.

Establish a master key and subkeys: Create a master key for certifications and then use subkeys for day-to-day operations like signing and encryption.

The master key should be stored offline, while the subkeys can be stored on your devices.

Use strong passphrases:

Protect your private key with a complex, unique passphrase. PGP security is only as strong as its passphrase.

Set an expiration date:

A good practice is to set an expiration date on your keys (e.g. within two years).

This forces you to generate a new key on a set schedule, making old, compromised, or forgotten keys obsolete over time.

Generate a revocation certificate:

Immediately after creating a new key, generate a revocation certificate.

Store this certificate in a safe place, separate from your private key. In case your private key is lost or compromised, this certificate allows you to publicly revoke the key.

2. Key distribution and verification

Share your public key:

Distribute your public key to those who need to send you encrypted messages or verify your signature.

This can be done by uploading it to a public keyserver or by sharing it as an ASCII-armored file (.asc).

Import public keys:

When you want to send an encrypted message to someone else, you must import their public key into your keyring first.

Verify key authenticity:

You should never blindly trust [3] a public key from a keyserver.

Instead, you should verify a key's authenticity by comparing its fingerprint over a trusted channel (e.g. over a phone call or in person).

Use the Web of Trust:

PGP's original model uses a "web of trust" to help verify key authenticity. Users can cryptographically sign another user's key to certify that they have verified the key belongs to that individual.

Your software can then use these third-party signatures to determine how much it trusts a key.

3. Ongoing maintenance

Secure your private key:

Your private key is your most valuable asset.

Store it on a machine with strong access controls, or on an encrypted, offline device like a USB stick.

Back up your keys:

Regularly back up both your public and private keyrings. This is crucial because if you lose your private key or forget your passphrase, you will not be able to decrypt any messages (old or current).

Change passphrases:

Change your passphrase regularly, especially if you suspect it may have been compromised.

Revoke compromised or expired keys:

If a private key is compromised, or a key reaches its expiration date, you should revoke it immediately and issue a new one.

The revocation certificate you generated earlier makes this possible.

4. Tools for :abbr:`PKI (Public Key Infrastructure )` key management

Manual PKI

is a free, open-source command-line tool that is a complete implementation of the OpenPGP standard. It is the most common tool for manual :abbr:`PGP (Pretty Good Privacy)` key management and integrates with many applications.

Practical Advice on GPG keys from Alex Cabal Article

Gpg4win:

A Windows version of GnuPG that includes graphical tools for managing keys and integrating with applications like Microsoft Outlook.

Kleopatra:

A certificate manager included with Gpg4win that provides a graphical user interface for key management.

Managed platforms:

For corporate environments, tools like IBM Sterling B2B Integrator and Symantec Encryption Management Server offer automated key management features, such as secure storage and server-side control over key expirations.

Webmail providers:

Services like Proton Mail integrate PGP key management seamlessly, handling the creation, encryption, and decryption in the background.

1. **Public Key**: Your security infrastructure must guard, against unauthorized write access, the location where public keys are made available. Allthough public keys are not kept private, they must must still be authentic. This is why public key servers need to vet insertions by requiring authentication codes during insertion transactions.
2. **Private Key**: Protect the private key file with a strong passphrase and don't allow anyone but the owner read access to it. Never share it. Anyone who has access to a private key can use it to make digiital signatures and also to decrypt secret messages. It is very important never to share these keys with anyone, the passphrase securing it should be strong. The key file encrypted by the passphrase should also be inaccessible to anyone but the owner, thereby making brute force techniques to crack the passphrase impractical.
3. **Trust**: When we speak of trusting public keys we are concerned with the possibility of someone other than the intended recipient of an encrypted message who published a maskerading key tricking you into encrypting a message which their own key rather than the real intended recipient key can decrypt. We are also concerned about a maskerader sending you digitally signed messages as though they came from the genuine sender. To mitigate these risks we need to be meticulous about where and how our public keys are shared and verified.