Taking responsibility for one's own security

This is my agenda for establishing a personal digital security strategy.

Motivation

Nowadays, we must interact electronically with the various financial, medical, governmental institutions, and with our community in general.

We need to guard against others maskerading as ourselves, others gaining unauthorized access to our assets, others tampering with documents, media, software, data we share, others discovering our private information. In some cases we need to protect our anonymity.

if we do not address these issues we open ourselves to many risks.

Discussion on Key Pairs

Specifically PKI [1] key pairs.

These key pairs consist of two keys, one key is private which we must keep secret, another is public to be shared with all other people.

With these two keys various cryptographic functions are available to us:

- Message encryption for privacy.

  It aims to keep message body decypherable by only the intended message receiver.

  Using a public key one can encrypt a message body that only the holder of the private key can decrypt.

  For large files, one can use a secret key (which is faster) to encrypt the message body, and then one shares an encrypted copy of the secret key using public key encryption. That is also a useful mechanism for sharing among several users by making the encrypted data available online and sharing only the secret encryption key itself among authorized viewers.

- Message Integrity.

  We can generate a cryptographic hash (a unique fixed length string produced by applying a known hash function to the message body).

  The hash is a brief, accurate and precise representation of the body of the message.

  We send both the original body of the message and the hash. The receiver, by applying the same hash function on the received message body and comparing the result to the received hash, can verify that there has been no tampering.

  To rule out the possibilty of tampering with both message and hash, we must encrypt the hash itself using the sender's private key thus confirming the hash authenticity when the receiver decrypts it using the sender public key.

- Digital Signature

  The sender generates a hash of the message body and uses its own private key to encrypt that hash. The receiver decrypts the hash using the sender public key, applies same hash function to the message body and compares the two. This at once verifies the integrity of the message body and the authenticity of the sender.

- Combined

  Hash the message body, sign the hash, encrypt both using he recipient public key.

  The receiver uses its own private key to decrypt both message body and signature. Re-generates hash, decrypts signed hash, and compares hashes.

By so doing Privacy, integrity and authentication are all achieved. Encryption with receiver public key assures privacy. Decryption of sent hash using sender public key authenticates, comparison of hashes verifies integrity.

We adopt a policy of using key pairs because of the benefits described in the above discussion.

Mitigating Risks When Using PKI Processes.

The trust in processes described above hinges on the receiver's confidence that the public key they use to verify a signature actually belongs to the sender. This is where digital certificates and Certificate Authorities (CAs) come in.

Even when we can trust the public key, keys can be compromised, so although it does belongs to the sender, when compromised someone other than the sender may be using it. That is where revocation comes in. The owner of the key pair must revoke the key as soon as possible when it may have been compromised (E.G. Stolen laptop). All receivers must be diligent about verifying that keys have not been revoked.

It is a sensible assumption that sooner or later every key will be compromised, in light of that our policy needs to concern itself with the lifespan that key pairs remain valid, thus reducing the extent of damage of a single compromise.

Interoperability

Since the range of platforms used by any community of interest is broad. The availability of functions we rely on to implement our security protocols must be equally broad.

Conclusion

Our policy needs to concern itself with the creation, distribution, revocation, and longevity of PKI [1] key pairs. It needs to concern itself with mechanisms for discovering sender and receiver public keys.

Our policy needs to confine itself to what is reasonably available to all participants in our community of interest.

Policy

We mandate the use of the OpenPGP standard for our PKI [1]. This standard has remained consistent, reliable and widespread for over a decade. However because of a recent Schism in the OpenPGP community two diverging standards (Crypto Refresh - RFC 9580 - and LibrePGP) have emerged.

So to avoid incompatibilities our policy will mandate only what their standards hold in common. They both descend from RFC 4880. That is the common ground. So we adopt the policy of limiting ourselves to only to RFC 4880 key formats until this rift is resolved one way or another and we are back to a single standard.

For the sake of interoperability, we confine ourselves to the *Compatibility Policy* out lined below.

Compatibility Policy.

Key type

- All parties generate RSA key types

Key length

- Minimum key length 3072, 4096 is a good choice.

Algorithms

- Symmetric Encryption use AES (e.g. AES-256)
- Hash use SHA-256 or SHA-512

Summary

- Use RSA keys with a length of 4096.
- Use AES-256 for symmetric encryption.

- Exchange and verify key fingerprints through a secure, out-of-band channel such as whatsap.

1(1, 2, 3) PKI: Public Key Infrastructure.