Pretty Good Privacy

Dan Fleck

CS 469: Security Engineering

Pretty Good Privacy

Various crypto algorithms and products provide strong encryption, but are not particularly easy to use.

Phil Zimmermann had the goal of providing strong encryption to everyone, in the form of an email encryption system that is:

- extremely strong, using state of the art cryptographic algorithms;
- easy to use and accessible to all. – Ver 1 (1991)

PGP is "the closest you're likely to get to military-grade encryption." –Bruce Schneier, *Applied Cryptography*

Coming up: Zimmermann's Motivation

Zimmermann’s Motivation

Zimmermann had a strong distrust of the government, and believed strongly that everyone had an absolute right to privacy.

The government generally believes that the right to privacy is limited by the need of the government to read messages under certain circumstances. Historically, the government restricted access to strong encryption.

PGP is a “end-run” around government restrictions, and almost landed Zimmermann in jail.

Coming up: Did Zimmermann Succeed?

Did Zimmermann Succeed?

From Wikipedia page on PGP:

In 2003, an incident involving seized Psion PDAs belonging to members of the Red Brigade indicated that neither the Italian police nor the FBI were able to decode PGP-encrypted files stored on them.

A more recent incident in December 2006 (see United States v. Boucher) involving US customs agents and a seized laptop PC which allegedly contained child pornography indicates that US Government agencies find it "nearly impossible" to access PGP-encrypted files.

PGP

Zimmermann developed PGP (Pretty Good Privacy) in the late 1980’s and early 1990’s. Some characteristics include:

1. Uses the best available cryptographic algorithms as building blocks.
2. Integrates these into a general-purpose algorithm that is processor-independent and easy to use.
3. Package and documentation, including source code, are freely available on-line.
4. PGP is now provided by Viacrypt in a compatible, low-cost commercial version.

Why would anyone buy this software from Viacrypt when it’s available free?

Growth of PGP

PGP has grown explosively and is widely used.

1. Available free worldwide for Windows, UNIX, Macintosh, and others. The commercial version satisfies businesses needing vendor support.

2. Based on algorithms with extensive public review.
   - Public key encryption: RSA, DSS, Diffie-Hellman.
   - Symmetric encryption: CAST-128, IDEA, and 3DES.
   - Hash coding: SHA-1.

3. Wide applicability: standardized scheme for encryption, supports secure communication over Internet and other networks.

4. Not developed by or controlled by any government.

5. Now on track to become an Internet standard (OpenPGP RFC 3156)

Lessons

- PGP illustrates that strong encryption can be packaged conveniently and accessible to everyone.

- PGP is very widely used and extremely secure

PGP Services

PGP supplies five basic services:

1. Authentication
2. Confidentiality
3. Compression
4. Email compatibility
5. Segmentation

Coming up: PGP Authentication

PGP Authentication

This is a digital signature function.

1. Sender creates a message M.
2. Sender generates a hash of M.
3. Sender signs the hash using his private key and prepends the result to the message.
4. Receiver uses the sender’s public key to verify the signature and recover the hash code.
5. Receiver generates a new hash code for M and compares it with the decrypted hash code.

Abstractly:

$$S \rightarrow R : \{h(M)\}_{K_s^{-1}}, M$$

PGP Confidentiality

PGP provides encryption for messages sent or stored as files.

1. Sender generates a message M and a random session key K.
2. M is encrypted using key K.
3. K is encrypted using the recipient's public key, and prepended to the message.
4. Receiver uses his private key to recover the session key.
5. The session key is used to decrypt the message.

But why? Why not just use private key to encrypt the message instead of session key?

Abstractly:

$$S \rightarrow R : \{K\}_{K_r}, \{M\}_K$$

Confidentiality and Authentication

Both authentication and confidentiality may be combined for a given message.

1. Apply the authentication step to the original message.
2. Apply the confidentiality step to the resulting message.

Why is it preferable to generate a signature for the plaintext message, rather than for the encrypted message?

Lessons

- PGP offers five basic services.
- Two of those are authentication and confidentiality; these can be combined.

PGP Services

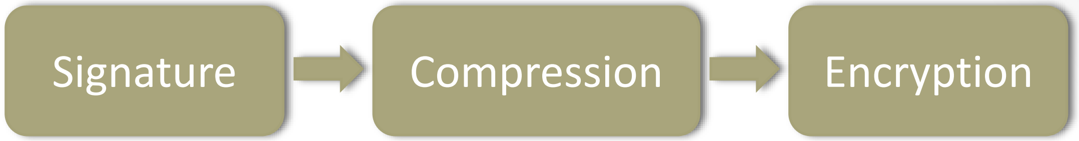
Recall that PGP supplies five basic services:

1. Authentication
2. Confidentiality
3. Compression
4. Email compatibility
5. Segmentation

Actually, only authentication and confidentiality are really "services." The others are engineering features designed to make PGP efficient and robust.

Coming up: Compression

Compression



As a default, PGP compresses the message, using the ZIP compression algorithm, after applying the signature and before encryption.

It is done in this order because:

- It is preferable to sign an uncompressed message so that the signature does not depend on the compression algorithm.

- Versions of the compression algorithm behave slightly differently, though all version are interoperable.

- Encryption after compression strengthens the encryption, since compression reduces redundancy in the message.

Email Compatibility

PGP always involves encryption. Encrypted text contains arbitrary 8-bit octets. However, many email systems would choke on certain bit strings they'd interpret as control commands.

PGP uses radix-64 conversion to map groups of three octets into four ASCII characters. Also appends a CRC for data error checking. By default, even ASCII is converted.

Use of radix-64 expands the message by 33%. This is usually more than offset by the compression.

Coming up: Segmentation and Reassembly

Segmentation and Reassembly

Email systems often restrict message length. Longer messages must be broken into segments, which are mailed separately.

PGP automatically segments messages that are too large. This is done after all of the other steps, including radix-64 conversion. Thus, signature and session key appear only once.

At the receiving end, PGP strips off mail headers and reassembles the message from its component pieces.

Lessons

- PGP provides the “services” of compression, email compatibility, and segmentation to make the system more robust and efficient.

PGP: Key Management

Dan Fleck

CS 469: Security Engineering

Key Management

PGP makes use of four types of keys: one-time session symmetric keys, public keys, private keys, passphrase-based symmetric keys.

Session keys: used once and generated for each new message

Public keys: used in asymmetric encryption

Private keys: also used in asymmetric encryption

Passphrase-based keys: used to protect private keys

A single user can have multiple public/private key pairs.

Coming up: Session Key Generation

Session Key Generation

Each session key is associated with a single message and used only once. Key size depends on the chosen encryption algorithm E; e.g. CAST-128: 128 bits, 3DES: 168-bits, etc.

The encryption algorithm E is used to generate a new n-bit key from a previous session key and two n/2-bit blocks generated based on user keystrokes, including keystroke timing. The two blocks are encrypted using E and the previous key, and combined to form the new key.

Public/Private Key Generation

For new RSA keys, an odd number n of sufficient size (usually > 200 bits) is generated and tested for primality. If it is not prime, then repeat with another randomly generated number, until a prime is found.

Primes appear in the neighborhood of n about every $\ln(n) = \lg_e(n)$ numbers. Since we can exclude even numbers, to find a prime of around 200 bits, it takes about $\ln(2^{200})/2 = 70$ tries.

This is an expensive operation, but performed relatively infrequently.

Coming up: Encrypting the Private Key

Encrypting the Private Key

The private key is stored encrypted with a user-supplied passphrase:

1. The user selects a passphrase for encrypting private keys.
2. When a new public/private key pair is generated, the system asks for the passphrase. Using SHA-1, a 160-bit hash code is generated from the passphrase, which is discarded.
3. The private key is encrypted using CAST-128 with 128 bits of the hash code as key. The key is then discarded.

Whenever the user wants to access the private key, he must supply the passphrase.

Lessons

- PGP uses four kinds of keys: session keys, public and private keys, and passphrase generated keys.
- Public / private key pairs are the most expensive to generate.
- Since the security of the system depends on protecting private keys, these are encrypted using a passphrase system.

Key Management

- In PGP, session keys and passphrase-based keys are generated on the fly, used once and discarded.
- Public and private keys are persistent and need to be preserved and managed. Recall that a user can have multiple public/private key pairs.

Coming up: Managing Key Pairs

Managing Key Pairs

Given that a user may have multiple public/private key pairs, how do we know which public key was used to encrypt a message.

- Send the public key along with the message. *Inefficient, since the key might be thousands of bits.*
- Associate a unique ID with each key pair and send that with the message. *Would require that all senders know that mapping of keys to ID's for all recipients.*
- Generate an ID likely to be unique for a given user. This is PGP's solution. *Use the least significant 64-bits of the key as the ID.*

This is used by the receiver to verify that he has such a key on his "key ring." The associated private key is used for the decryption.

Coming up: Key Rings: Private Key Ring

Key Rings: Private Key Ring

Each user maintains two key ring data structures: a private-key ring for his own public/private key pairs, and a public-key ring for the public keys of correspondents.

The private key ring is a table of rows containing:

Timestamp: when the key pair was generated.

Key ID: 64 least significant digits of the public key.

Public key: the public portion of the key.

Private key: the private portion, encrypted using a passphrase.

User ID: usually the user's email address. May be different for different key pairs.

Coming up: Public Key Ring

Public Key Ring

Public keys of other users are stored on a user’s public-key ring.

This is a table of rows containing (among other fields):

Timestamp:	when the entry was generated.
Key ID:	64 least significant digits of this entry.
Public key:	the public key for the entry.
User ID:	Identifier for the owner of this key. Multiple IDs may be associated with a single public key.

The public key can be indexed by either User ID or Key ID.

Retrieving a Private Key

Whenever PGP must use a private key, it must decrypt it. For example, suppose R receives a message encrypted with K_R.

1. PGP retrieves receiver’s encrypted private key from the private-key ring, using the Key ID field in the session key component of the message as an index.

2. PGP prompts the user for the passphrase to recover the unencrypted private key.

3. PGP recovers the session key and decrypts the message.

Validity of Public Key

Associated with each public key in the user's public key ring is a **key legitimacy field** that indicates the extent to which PGP trusts that this is a valid public key for this user.

Legitimacy is determined from certificates and chains of certificates, the user's assessment of the trust to be assigned to the key, and various heuristics for computing trust.

Coming up: Revoking Public Keys

Revoking Public Keys

A user may wish to revoke a public key because:

- compromise is suspected, or
- to limit the period of use of the key.

The owner issues a signed key revocation certificate. Recipients are expected to update their public-key rings.

Lessons

- Each PGP user must manage his own private keys and the public keys of others.
- These are stored on separate keys rings.
- Private keys are protected by encryption; public keys are stored with certificates attesting to their trustworthiness.
- Keys can be revoked.

End of presentation