E-Mail Security:  PGP (Pretty Good Privacy) & PEM (Privacy-Enhanced Mail)

Konstantinos Raptis

Supervisor: Sokratis Katsikas

Samos, September ‘99
Table of Contents:

Introduction ........................................................................................................... 3
Email Security ......................................................................................................... 3
Pretty Good Privacy (PGP) .................................................................................... 4
Privacy-Enhanced Mail (PEM) ............................................................................... 7
Conclusion ............................................................................................................. 9
References ............................................................................................................. 10
Introduction

On the Internet, the notions of privacy and security are practically non-existent. Although email is one of the most popular uses of the Internet, security experts have estimated that only about one in every 100 messages is secured against interception and alteration.

Many people may think that sending an email in plain text is privacy-protected and enhancement of privacy is not necessary. This is simply not the fact. Whether you realize it or not, those messages you've been sending to business partners or friends over the Internet have been sent in the clear; information you thought was enclosed in a sealed envelope was instead sent just like a postcard.

When an email message is sent between two distant sites, it will generally transit dozens of machines on the way. Any of these machines can read the message and/or record it for future work.

Email Security

Let’s look at some of the assumptions many people have about the security and integrity of email [1].

Authenticity

Many people assume that the name given as the sender of an email message identifies who actually sent it. In fact, this depends on the honesty of the sender and the flexibility of their mail package. For example, the Netscape Navigator mail function allows people to enter their own description of who they are, and what their email address is. While this will not allow them to receive mail that is not properly addressed to them, they can still send mail.

Integrity

When you send a message via email, there is no guarantee that it will be received, or that what is received is exactly what you sent. You have no way of knowing that your message was not read or forwarded by third parties. This is due to the passing of messages from machine to machine, between your email server and that of the intended recipient.

At any point along the way, the mail server could lose the message, or the staff supporting the server could read and/or alter it. This is obvious if you consider that a mail message is only a file that gets passed from person to person along a delivery chain. Any person in the chain can drop the whole file in the garbage, or copy, add, delete, or replace documents in it. The next person in the chain doesn't know it's coming, what's in it, or how big it should be. These people don't work for the same company, and quite possibly aren’t even on the same continent.

If you mis-spell the recipient's address, the mail server at their end may send the note back to you as undeliverable. However, it may also send it to somebody else, who happens to have the address you typed, or it may send it to the “Postmaster”, who administers the system. Normally the postmaster will re-send it to the appropriate...
person, but this is a manual process, which may take some time, or it may not be done at all.

To add to the confusion, incoming and outgoing mail is stored in plain text files on a hard disk in your mail server. These files can be altered by authorized administrators or by anybody capable of assuming authority. While University employees do not do this on a whim, the capability exists.

Reliability
As a sender, you have no way of knowing when a message was delivered. It could have been delayed due to system problems at an intermediate link in the delivery chain. Also, there is no standard way of requesting a receipt when the message is read. If you request a return receipt, and the receiver’s mail system does not recognize that function, it will not send you an email note confirming delivery.

Because of the wide-spread nature of these problems, a number of competing solutions are being developed that address the authentication and integrity issues. The general consensus is to use some form of public-key cryptography, so that messages can be decrypted only by the intended recipient, are unalterable, and can be verified as coming from the sender.

Pretty Good Privacy, PGP, and Privacy-Enhanced Mail, PEM, are both “systems” that provide secrecy and non-repudiation of data that is sent over the Internet, mostly by email (figure 1).

Pretty Good Privacy (PGP)
Pretty Good Privacy (PGP) is a popular program used to encrypt and decrypt email over the Internet. It can also be used to send an encrypted digital signature that lets the receiver verify the sender’s identity and know that the message was not changed en route. Available both as freeware and in a low-cost commercial version, PGP is the most widely used privacy-ensuring program by individuals and is also used by many corporations. Developed by Philip R. Zimmermann in 1991, PGP has become a de facto standard for e-mail security. PGP can also be used to encrypt files being stored so that they are unreadable by other users or intruders.

PGP can be used basically for 4 things:
- Encrypting a message or file so that only the recipient can decrypt and read it.
  The sender, by digitally signing with PGP, can also guarantee to the recipient, that the message or file must have come from the sender and not an impostor.
• Clear signing a plain text message guarantees that it can only have come from the sender and not an impostor.
• Encrypting computer files so that they can't be decrypted by anyone other than the person who encrypted them.
• Really deleting files (i.e. overwriting the content so that it can't be recovered and read by anyone else) rather than just removing the file name from a directory/folder.

As we mentioned, PGP provides two services: encryption and digital signatures [2]. Encryption allows a user to encode a file for storage locally or for transmission as an e-mail message. The local storage option is handy if you are worried about other people having access to files on your machine. The e-mail option enables PGP to be used for private exchanges over a network. PGP encrypts the entire contents of the message in such a way that only the intended recipient can decode and read the message. Anyone else who attempts to capture or copy the message en route will receive meaningless garble.

The digital signature service allows a user to 'sign' a document before transmission in such a way that anyone can verify that the signature is genuine and belongs with a particular document. If someone alters the message or substitutes a different message, the signature will no longer be valid. And any recipient can verify that the message has been signed by its true creator and not an impostor.

PGP's confidentiality and encoding services use the most popular public-key encryption scheme, known as RSA. All public-key encryption systems make use of an encoding and decoding algorithm and a related pair of keys. The input to the encryption algorithm is the text to be encrypted--known as plaintext--and a key. The algorithm takes the input and produces scrambled output known as ciphertext. To use the decryption algorithm, you input the ciphertext plus the key that matches the one used for encryption, and the original plaintext is produced as output.

The two keys used in any public-key encryption scheme, including RSA, are called the public key and private key. The public key, as the name suggests, is made public. The idea is to make your public key available to people with whom you correspond. You keep your private key secure, and it should be known only to you. These two keys can be used to provide confidentiality and encryption.

For confidentiality, PGP encrypts messages with an efficient single-key or conventional encryption algorithm known as IDEA. It then uses RSA to encrypt, with the receiver's public key, the IDEA key used to encrypt the message. The receiver can use RSA to recover the IDEA key and use that key to recover the message.

For digital signatures, PGP uses an efficient algorithm known as MD5 to produce a summary code, or hash code, of the message that is, for all practical purposes, unique to that message. PGP then uses RSA to encrypt the hash code with the sender's private key. The receiver can use RSA to recover the hash code and verify that it is the correct hash code for the message. If it is correct, then only the alleged sender could have prepared the encrypted hash code.
But how safe is PGP? Will it really protect my privacy? As we know, nothing can be 100% secure. For PGP we have to consider three main points. First, top-rate civilian cryptographers and computer experts have tried unsuccessfully to break PGP. Second, whoever proves that he or she can unravel PGP will earn quick fame in crypto circles. He or she will be applauded at banquets and attract grant money. Third, PGP's most knowledgeable users around the world will broadcast this news at once.

People often claim that PGP is illegal. There are three separate reasons why they might claim so. It is probable that PGP falls under the ITAR (International Traffic in Arms Regulations) restrictions, which control the export of munitions and cryptographic technology from the US and CANADA. If this is the case, it is illegal to export PGP from the USA or Canada to any other country. Of course, if you don’t get PGP from the US or Canada, the issue is moot. In some countries, the use of cryptography is restricted by law or, it is outright illegal to encrypt data at all. In other countries, they’re working on it.

In a few words, we can summarize the legal status of PGP as follows:

- If you live in the USA or Canada, you should buy ViaCrypt PGP, or use MIT PGP 2.6. Otherwise you will be infringing the patents held by PKP, which you merely paid for.
- If you live outside the USA or Canada, you can use PGP without having to worry about infringing the RSA patent (though the IDEA patent still stands if you wish to use PGP commercially). If you use International PGP (version 2.6.2i), you should have no problems talking to other versions. 2.6.2i is endorsed by Phil Zimmermann for use outside the USA, and is faster and more compatible than any other freeware version of PGP.
- It may be illegal to send encrypted messages in some countries or on some networks.
- You should not export PGP from the USA or Canada to any other country.

An example of a PGP key is the follow:

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: 2.6.i
mQBNAi6w7BsAAEACANCXkbb8RjCuSVhLLb+dkRJnDJ83ktiCY/UOzv9tDMSRYVvplL
TM9PSmEqvD4ghiaW1vnFL6djYAEVM/BVFBdeATkABRG0OEhhcmFszCBOvdVpdcCB
bHzl3c3ryYW5kIDxIXJhbQquVC5BbHZ1c3rjYW5kQHVuaW51dHQwbm8+iQCVAqUO
Ly+cZH98CXqjtTaSTqFqSvqAxtz+Fp3Mm6jRgariX5HRZ+oDMPfaRbtlhf8eF5
aMoFRO3ZGhzI2kfqn1GCF1iyxzh4kZV4vNfNoav5kYTzrvZ7isUidaCVFKH6oA
mWUeweWAw2zkLmbbbk8YTj3EKKXbcH7XKrcIYuwj0Z3eS1+CUNkdH6/6bqhz
qqCJAHUCBR AwQqjy89bnwTbXqBBsQAwDk+qY515K22zSGdQb6x0kKLAeOEb55
D2FBiyFCwVpQ089jPqDpao7Ts2NhVUOBhpFk63shKksvkspUHP46UHNwrUCYV841
BdFPUdUArG8cqlLOjmdXH4fYrklv2s/fuaJAJUDAyQmCvlnhf3d8rIvEBAc1J
A/4g8d8tWhDyGFr0Cxb4mtvX5S+XWtcIiKd/OZzWXlX4xMJaAtGMeN7PQwWG6
FsoayC3V4jvUqGwQ3t+3tyqHgnp9xU3trVht82CTAuYzp5h6fV3YU1GXA8ccwDF
2A50ZCYH5Wp-nMPSIFvJdNY3Hbwr+9qgx+QP+ezYXzokAVJFEC6xCl4zFW8U8
F14BOQEBwMCALcsmMQH2CITEh9uxTcSwatqbgEFIpwjw/YGeYfzmA40a7iY7Jd2
NG4aylYaUX7mQoNHGi1lD0s3JtCDnW/JK0CGrh0YUBkYxl
=3Qxp
-----END PGP PUBLIC KEY BLOCK-----
```
Privacy-Enhanced Mail (PEM)

Privacy-Enhanced Mail (PEM) is an Internet standard that provides for secure exchange of electronic mail. PEM employs a range of cryptographic techniques to allow for confidentiality, sender authentication, and message integrity. The message integrity aspects allow the user to ensure that a message hasn't been modified during transport from the sender. The sender authentication allows a user to verify that the PEM message that they have received is truly from the person who claims to have sent it. The confidentiality feature allows a message to be kept secret from people to whom the message was not addressed.

PEM does not require the use of a specific algorithm. On the contrary, it allows use of several algorithms for data encryption, key management, and data integrity. The details of PEM are described in four Internet RFCs as the following: RFC 1421 [3] describes message encryption and authentication procedure; RFC 1422 [4] addresses certificate-based key management including the key management architecture and infrastructure using public-key certificates; RFC 1423 [5] describes the encryption and message integrity algorithms, including key management; RFC 1424 [6] describes three types of services to support PEM, including key certification, certificate-revocation list (CRL) storage, and CRL retrieval.

PEM provides a range of security features. They include originator authentication, (optional) message confidentiality, and data integrity. Each of these will be discussed in turn.

Originator Authentication

In RFC 1422 [4] an authentication scheme for PEM is defined. It uses a hierarchical authentication framework compatible X.509, "The Directory --- Authentication Framework." Central to the PEM authentication framework are certificates, which contain items such as the digital signature algorithm used to sign the certificate, the subject's Distinguished Name, the certificate issuer's Distinguished name, a validity period, indicating the starting and ending dates the certificate should be considered valid, the subject's public key along with the accompanying algorithm. This hierarchical authentication framework has four entities.

The first entity is a central authority called the Internet Policy Registration Authority (IPRA), acting as the root of the hierarchy and forming the foundation of all certificate validation in the hierarchy. It is responsible for certifying and reviewing the policies of the entities in the next lower level. These entities are called Policy Certification Authorities (PCAs), which are responsible for certifying the next lower level of authorities. The next lower level consists of Certification Authorities (CAs), responsible for certifying both subordinate CAs and also individual users. Individual users are on the lowest level of the hierarchy.

This hierarchical approach to certification allows one to be reasonably sure that certificates coming users, assuming one trusts the policies of the intervening CAs and PCAs and the policy of the IPRA itself, actually came from the person whose name is associated with it. This hierarchy also makes it more difficult to spoof a certificate because it is likely that few people will trust or use certificates that have untraceable
certification trails, and in order to generate a false certificate one would need to subvert at least a CA, and possibly the certifying PCA and the IPRA itself.

Message Confidentiality

Message confidentiality in PEM is implemented by using standardized cryptographic algorithms. RFC 1423 [5] defines both symmetric and asymmetric encryption algorithms to be used in PEM key management and message encryption. Currently, the only standardized algorithm for message encryption is the Data Encryption Standard (DES) in Cipher Block Chaining (CBC) mode. Currently, DES in both Electronic Code Book (ECB) mode and Encrypt-Decrypt-Encrypt (EDE) mode, using a pair of 64-bit keys, are standardized for symmetric key management. For asymmetric key management, the RSA algorithm is used.

Data Integrity

In order to provide data integrity, PEM implements a concept known as a message digest. The message digests that PEM uses are known as RSA-MD2 and RSA-MD5 for both symmetric and asymmetric key management modes. Essentially both algorithms take arbitrary-length "messages," which could be any message or file, and produce a 16-octet value. This value is then encrypted with whichever key management technique is currently in use. When the message is received, the recipient can also run the message digest on the message, and if it hasn't been modified in-transit, the recipient can be reasonably assured that the message hasn't been tampered with maliciously. The reason message digests are used is because they're relatively fast to compute, and finding two different meaningful messages that produce the same value is nearly impossible.

There are at least two different implementations of PEM available. The first one is Riordan's Internet Privacy Enhanced Mail (RIPEM), written by Mark Riordan [7]. The other implementation of PEM was originally called TIS/PEM [8] (version 7.0), written by Trusted Information Systems, Inc. However, TIS/PEM has since been succeeded by TIS/MISS (version 7.1), a program which implements PEM with MIME extensions added to it.

Like PGP, PEM uses RSA encryption. As we mentioned, the US government has strict export controls over foreign use of this technology, so people outside the US may have a difficult time finding programs which perform the encryption.

An example of a PEM key is the follow:

```
-----BEGIN PRIVACY-ENHANCED MESSAGE-----
Proc-Type: 4,MAC-CLEAR
Content-Domain: RFC822
Originator-Certificate:
MIIBhDCB7gIIBjANBgkqhkiG9w0BAQIFAfAwRQYDVQQGEwJOTzEQMA4GA1UE
ChMHdW5pbmV0dDAsFw05NjAxMDUxMjU4MjNaFw05ODAxMDExMjAwMDBaMD4xMzAJ
BgNVBAYTAk5PMRAwDgYDVQQKEwdVTklORVRUMR0wGwYDVQQDExRIYXJhbGgVC4g
QWx2ZWN0cmFuc2ZBMQoGA1UHEQYDVQQLExRIYXJhbGgVC4g
-----BEGIN PRIVACY-ENHANCED MESSAGE-----
Proc-Type: 4,MAC-CLEAR
Content-Domain: RFC822
Originator-Certificate:
MIIBhDCB7gIIBjANBgkqhkiG9w0BAQIFAfAwRQYDVQQGEwJOTzEQMA4GA1UE
ChMHdW5pbmV0dDAsFw05NjAxMDUxMjU4MjNaFw05ODAxMDExMjAwMDBaMD4xMzAJ
BgNVBAYTAk5PMRAwDgYDVQQKEwdVTklORVRUMR0wGwYDVQQDExRIYXJhbGgVC4g
QWx2ZWN0cmFuc2ZBMQoGA1UHEQYDVQQLExRIYXJhbGgVC4g
```

---
Conclusion

As we mentioned, Pretty Good Privacy, PGP, and Privacy-Enhanced Mail, PEM, are both “systems” that provide secrecy and non-repudiation of data that is sent over the Internet, mostly by email. PGP and PEM allow two parties to communicate in a way which does not allow third parties to read them, and which certify that the person who sent the message is really who they claim they are.

The table below presents the basic similarities and differences of PGP and PEM.

<table>
<thead>
<tr>
<th>Item</th>
<th>PGP</th>
<th>PEM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supports encryption?</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Supports authentication?</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Supports non-repudiation?</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Supports compression?</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Supports canonicalization?</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Supports mailing lists?</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Uses base64 coding?</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Secret-key algorithm</td>
<td>IDEA</td>
<td>DES, TDES</td>
</tr>
<tr>
<td>Secret-key length (bits)</td>
<td>128</td>
<td>64</td>
</tr>
<tr>
<td>Public-key algorithm</td>
<td>RSA</td>
<td>RSA</td>
</tr>
<tr>
<td>Public-key length (bits)</td>
<td>384/512/1024</td>
<td>Variable</td>
</tr>
<tr>
<td>User name space</td>
<td>User defined</td>
<td>X.400</td>
</tr>
</tbody>
</table>

Dette er signert med PEM
-----END PRIVACY-ENHANCED MESSAGE-----
<table>
<thead>
<tr>
<th>X.509 conformant</th>
<th>No</th>
<th>Yes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Do you have to trust anyone?</td>
<td>No</td>
<td>Yes (IRPA)</td>
</tr>
<tr>
<td>Key revocation</td>
<td>Haphazard</td>
<td>Better</td>
</tr>
<tr>
<td>Can eavesdroppers read messages?</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Can eavesdroppers read signatures?</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Internet standard?</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Exportable from US?</td>
<td>No</td>
<td>Depends</td>
</tr>
<tr>
<td>Can get full source?</td>
<td>Depends</td>
<td>Yes</td>
</tr>
<tr>
<td>Cost?</td>
<td>Free/Low cost</td>
<td>Free</td>
</tr>
</tbody>
</table>

PGP is a popular program used to encrypt and decrypt email over the Internet. Available both as freeware and in a low-cost commercial version, PGP is the most widely used privacy-ensuring program by individuals and is also used by many corporations.

PEM, in contrast to PGP, does not use widely. There are hooks for using both PEM, specifically RIPEM although TIS/PEM should work as well, and PGP in the NCSA httpd program for providing secure web communications with NCSA Mosaic. There are also extensions to the Emacs editor which allow one to use either PGP or a PEM implementation in conjunction with mail or any other Emacs buffer. There is also a product put out by SecureWare (http://www.secureware.com) called SecureMail that implements PEM.

References


