

**~ COLORADO TECHNICAL UNIVERSITY ~**

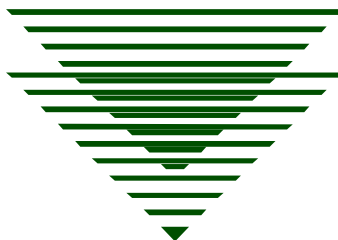
# **Overview of Message Security Protocols**

## **Securing Email with PGP and S/MIME**

**Professor: Dr. Karl Seifert**  
**Submitted in Partial Fulfillment of the Requirements for**  
**CS-653**  
**Network Security**

**By**

**William P. Flinn**  
**Fort Collins, Colorado**  
**September, 2005**



### Abstract

This paper discusses message security protocols. Particularly, Pretty Good Privacy (PGP) and Secure Multipurpose Internet Mail Extensions (S/MIME) will be discussed. An outline of the fundamental operations is given, as well as a discussion of how each uses public key infrastructure (PKI). The primary difference between PGP and S/MIME is that PGP is a software application, whereas S/MIME is a standard that must be supported by the message agent (email application) if S/MIME is to be used. This paper briefly discusses that risk analysis should be performed in order to decide whether or not to employ message security protocols, and if so, which ones to use.

## Securing Email with PGP and S/MIME

Email has become one of the most important and most widely used methods of communications today. It could be argued, for example, that using email to send messages has become more commonplace than mailing letters. The technological advantages of using email are great compared to the paper counterpart. There are certain advantages enjoyed by email users include convenience, speed, and even the ability to obtain delivery receipts at no extra costs. When comparing standard paper mail to email, however, it is clear that keeping email messages private as well as ensuring authenticity of the sender has become an important issue that needs to be effectively dealt with. For example, mailing a letter by way of conventional means at least ensures the letter is hidden from plain view because of envelopes and other outer wrappings. Of course the letter can be stolen or compromised in other ways, but generally speaking it is difficult for an unauthorized person to see the contents of the letter without intercepting it and opening it.

Securing email messages has become important in order to ensure privacy, and also to ensure that only the authorized recipient receives and opens the message. There are various types of security measures that exist in the various layers of the OSI communications model. There are protocols, for example, that ensure end-to-end reliability, and yet other protocols that are used to build a secure communications channel over which the data can travel. But these protocols don't always ensure confidentiality of the message, nor do they ensure that the sender is in fact who he or she claims to be. That is where the role of message security protocols plays an important part in electronic communications. Message security protocols can encrypt the data as well as provide digital signatures to help ensure the authenticity of the sender. By securing the transfer of the

message, and making the message unreadable by anyone but the intended recipient, electronic mail can be assured of a more secure environment.

The purpose of this paper is to explore different types of message security protocols. While there are several types which have been used throughout the history of electronic mail, this paper will primarily discuss the methodologies, advantages, and disadvantages of two primary message security systems: Pretty Good Privacy (PGP and Secure Multipurpose Internet Mail Extensions (S/MIME). The basic processes of each will be discussed, as well as an explanation of the evolution of each type throughout various iterations of technologies used.

### *Securing Messages*

The need to ensure message privacy is an important consideration. Email has replaced many forms of paper communication. As such, it has become apparent that the need to secure email is a little more difficult than simply putting it in an envelope and sending it by way of registered mail, as is the case with regular paper communications. There are several ways to ensure email privacy. It can either be sent through a secure communications channel, such as a virtual private network (VPN) or it can be encrypted so that even if the message is intercepted, it cannot be read in such a way as to make sense of the actual message (Oppliger, 2002, pg. 313-314). Although email is said to be secure (for most purposes) if the privacy is adequately assured, it must still be remembered that the value of the message must be weighed against the cost of the measures used to protect it (Daniels, 1997, 207).

Although email is an electronic media, the principles and methodologies for securing electronic mail are much the same as they are for protecting regular mail. Email can be sealed in a “digital” envelope, “signed” by a sender, and received by an authorized recipient. The primary focus of this paper will be to discuss methods for ensuring message privacy through encryption

as opposed to VPN technologies mentioned above. The technologies discussed in this paper, PGP and S/MIME, both rely, for the most part, on what are known as public key infrastructure (PKI) where public keys and private keys are used for encrypting and decrypting messages. Although it is not the scope of this paper to discuss encryption methods in detail, the basic technique of public key encryption is offered.

Essentially, the way that PKI is used for message confidentiality is as follows: Two keys, a public key and a private key are used for encrypting and decrypting messages. A party (Party A for this discussion) has a private key, which they guard and keep secret. They also have a public key which they make available to anyone (Party B for this discussion) who will be sending them a message. The Party B can use Party A's public key to encrypt an email message. Party A receives the message and can decrypt it with their private key. Thus the message is encrypted from sender to receiver, and the recipient can be relatively assured that the message came from the person who claimed to send it. The keys also have a reciprocal relationship in that the private key can also be used to encrypt and the public key can be used to decrypt messages.

### *Pretty Good Privacy (PGP)*

Pretty Good Privacy (PGP) was developed by Philip Zimmermann in the early 1990's using some fairly sophisticated encryption algorithms. PGP is essentially a software package that is installed on an end-user system. Initially, the way PGP worked was that the software was installed in the system, creating two key-rings on the computer. One key-ring called "sebring.pgp" for storing private keys and one called "pubring.pgp" for storing the public key. Once the appropriate keys are stored, they can be used to encrypt and decrypt messages.

In early implementations of PGP, the encryption and decryption of messages was a somewhat cumbersome process. The process for sending and opening an encrypted message using PGP

involved a number of steps. 1) The message is written using a word processor or other text creation application. 2) The message was then encrypted by using the installed PGP applications software. 3) The message is sent and received by the intended recipient. 4) The recipient uses the installed PGP software to then decrypt the message so that it can be read. The PGP application was not integrated with the client email application in this case. The private key is protected using IDEA encryption which uses a password or passphrase to protect access to its contents (Daniels, 1997, 209). One notable feature is in the discussion of key exchange in PGP environments. PGP public keys are usually exchanged by individuals either directly giving another party their public key, or even making them available on web sites. This is often referred to as a “web of trust” environment, and is well suited to small groups where sharing of public keys is a fairly easy thing to do (Oppliger, 2002, pg. 325).

The security feature of PGP software that prevents message tampering exists in that PGP can also use digital signatures, which is an integral part of PKI operations. PGP’s digital signature functionality uses an algorithm to calculate a digital code or “checksum” of the message content and encrypt it using the private key of the sender. This is known as the digital signature. The digital signature is decrypted by the recipient using the sender’s public key. If the decryption is successful, then the recipient is assured that the signature is authentic, and thus was sent by the person whom the recipient thinks has sent it (Mascha and Miller, 2002, pg. 63).

Having mentioned that PGP is a separate software package requiring additional steps to secure the confidentiality of a message, it could be presumed that a certain amount of risk analysis would have to be done to determine if the confidentiality of a particular message was important enough to warrant the use of PGP. It may very well be that all messages are not

deemed worthy of this type of protection. The user would then determine which messages to secure, and which could simply be sent as plain text.

As of the writing of this paper, PGP Desktop 9 is now available which provides an entire suite of security tools for encrypting not only messages but entire hard drives. The tools come complete with the PGP application for encrypting messages, plus a variety of other tools for “shredding data” (deleted files) and zeroing out free space. Unlike earlier iterations of PGP which required numerous steps and did not integrate with the email client application, PGP Desktop 9 is integrated so as to work with the email client. All a user is required to do is compose their email as usual using their email client. PGP Desktop 9 then “intercepts” the message on its way out to delivery and performs the encryption (Brandt, 2005, ¶ 3). Also, variants of PGP, such as Open PGP/MIME and PGP/MIME are implemented in email products such as Qualcomm Eudora version 4.3 (Oppliger, 2002, 321). The notion of having the email application integrate with the PGP application would then imply that the above discussion of a user having to think of whether or not to secure a message based on the risk of losing confidentiality for that message would no longer be an issue.

#### *Secure Multipurpose Internet Mail Extensions (S/MIME)*

In contrast to PGP, one of the most notable differences with Secure Multipurpose Internet Mail Extensions (S/MIME) is that rather than being a product or software package, S/MIME is more appropriately described as a standard. That is to say that S/MIME is a specification of how encryption works with email clients. As is the case with PGP, S/MIME uses public key encryption technologies. One of the notable differences, however, is in that S/MIME relies of key issuance to come from hierarchical certificate authorities (CAs), as opposed to PGP’s use of key exchange through direct or indirect key exchange (Oppliger, 2002, pg. 325). Digital

certificates can be purchased from a certificate provider such as Verisign or Nortel, for prices ranging from \$9.95 for a class 1 certificate in which an applicant completes only an identity form, to \$1,000+ for a class 3 certificate in which an applicant must undergo a background check in order to obtain a certificate (Mascha and Miller, 2002, 63). The important note here is then that the certificate is issued and can be verified by an authority as opposed to simply being shared by an individual.

Also in contrast to PGP, which was discussed in this paper as not being easily integrated with, or even not meant to be integrated with user agents (email software) S/MIME implementations are supported by such email clients as Microsoft Outlook, Lotus Notes, and Novell GroupWise, just to name a few (Fisher, 2004, ¶10). Again, this is because it should be noted that S/MIME is a standard that is supported by email applications as opposed to being an actual product or software package that is used separately from the email application (agent). But much like PGP, the concept of providing proof of the originator of a message is in the digital signature ability of S/MIME, which is able to help ensure authenticity of the sender of a message (Fisher, 2004, ¶9).

Using the “Party A” and “Party B” scenario once again, sending the message using the S/MIME principles works something like this: Party A encrypts a message to Party B with Party A’s private key. The message is signed using Party A’s digital certificate, and includes Party B’s digital certificate if message confidentiality is desired. Party B receives the message and compares the digital certificate received in the message with the digital certificate being held by a certificate authority – remember that while PGP uses trusts between individuals to exchange keys, S/MIME uses CAs to verify key authenticity. If the keys match, the message is authentic (Mascha and Miller, 2002, pg., 63).

Returning to the same risk analysis scenario discussed in the section of this paper on PGP, it could be speculated that using S/MIME supporting applications would not require quite as much thought by the user. If S/MIME is already integrated with the application, the message would be simply composed and sent without requiring the user to do risk analysis each time a message is sent. The choice would already be made in that S/MIME would be applied and used as a default condition of using the email client application.

### *Summary and Conclusions*

The purpose of this paper was to discuss ways to secure messages, specifically email messages. Two most of the most common methods of doing so are Pretty Good Privacy (PGP) and Secure Multipurpose Internet Mail Extensions (S/MIME). While both perform similar operations insofar as ensuring confidentiality and authenticity of messages is concerned, they each have notable differences in the way the user interacts with the encryption processes, and the way that the message is secured. The use of public and private key pairs by each is a similarity, but the common method of obtaining and exchanging certificates is somewhat different. PGP public keys are commonly exchanged directly or made available through means such as posting on web sites or sending via email. S/MIME, on the other hand, uses hierarchical certificate authorities to obtain keys.

PGP is an actual product, that is to say software application, that can be installed on the computer and used as a standalone program. It was even noted that PGP was not even initially meant to be incorporated into email agents, but as discussed in this paper, later iterations of PGP were built into desktop security suites, and have the ability to be automatically called by the email client in order to secure the message. This feature helps to get PGP into more of a “mainstream” presence by making it easier for the average end-user to install and use for

message security. In fact, it can be speculated that many ordinary email users do not take advantage of message security protocols because of the perception that they are too hard to install and use, or require too many steps. As mentioned previously, a certain amount of risk analysis has to be performed. The confidentiality requirements of a particular message may very well not justify the added complexity and required learning curve of using such a product as PGP.

S/MIME, on the other hand, is more of a standard that is implemented in email products. S/MIME is a standard incorporated into the email agent, but it should be noted that not all email clients support the use of MIME or S/MIME standards. In order to take advantage of message encryption, the message agent (email application) must have support for the S/MIME functionality. Some of the more well-known email clients such as Microsoft Outlook, Lotus Notes, and Novell GroupWise do support S/MIME and are thus well suited as email clients of choice.

Message encryption techniques should be adapted that will allow the business needs to be satisfied in an as efficient a manner as possible. Using the discussions of risk analysis discussed earlier, along with an understanding of what messaging (email) applications are already in use, a message security methodology can be chosen that will be compatible with an existing system and will provide the required message confidentiality and authenticity.

## References

Brandt, Andrew, (2005) PGP Desktop 9 Delivers a Sweet Security Suite, PC World Magazine, September, 2005; 23,9

Daniels, Shirley. (1997) Making E-Mail Secure. Work Study, Volume 46, Number 6, MCB University Press.

Fisher, Joseph. (2004) E-mail Authentication Slams Spam, Computer Technology Review; November, 2004; 24,11; ABI/INFORM Global

Mascha, Maureen and Miller, Cathleen. (2002) Stop the E-Mail Snoops. Journal of Accountancy, July 2002; ABI/INFORM Global.

Oppliger, Rolph (2002) Internet and Intranet Security, Second Edition. Norwood, MA: Artech House, Inc.