Abstract— With the explosive growth of Internet, e-mail has undergone a sea change. E-mail, which once was a convenience, is now a necessity. E-mail is now used to send critical documents or important business communications, where as in the past it was just a quick way to communicate. The Simple Mail Transfer Protocol (SMTP) standard has made it possible for different e-mail systems connected to the internet to exchange information through e-mails. Although SMTP was designed to simplify the process of sending emails, however it was not designed to send them securely. To overcome this disadvantage, standards called S/MIME and PGP have been developed to secure the data that is being sent in an email.

The purpose of this project is to analyze the major existing software that is compatible with the S/MIME and PGP standards. The goal is to help users with moderate computer skills choose which combination of software and secure standards best fit their needs. The paper will also analyze if either PGP or S/MIME has become a realistic choice for widespread implementation. This paper will not analyze every possible implementation of PGP or S/Mime for two main reasons. First, the costs of analyzing some of the solutions are above the means of the authors and that the possibility of widespread implementation for some implementations is very slim. Windows still has an extremely dominant market share, accordingly, this project can affect the largest group of individuals by targeting this segment. With regards to Open PGP, plug-ins other than software created by the PGP corporation has not been certified secure. There is the possibility according to some sources that in using these plug-ins, the level of security PGP can offer is considerably weaker. Because of these reasons our paper only focuses on PGP and S/MIME on Windows machines.

Often with computers there is no such thing as a perfect solution. With this idea in mind the analysis for this project will use multiple perspectives. The project will analyze those of the home/personal user and the business user. The home user will be more concerned about costs and the ease of use, while the business user will be more concerned about the security obtained by standard. Cost is defined as the cost for security standard chosen (S/Mime or PGP) and the cost of the software needed to run the standard. Security involves the level of confidentiality, non-repudiation, authentication and integrity. Security will also include the ability to revoke keys or detect keys that are no longer valid.

Index Terms—

I. INTRODUCTION

This document analyses the implementations of SMIME and PGP. This first section of the paper explains general concepts about SMIME and PGP which would be helpful in understanding the rest of the paper. SMIME (secure/multipurpose email extension) provides a consistent way to send and receive secure Mime data. Based on the popular Internet Mime standard, SMIME provides the following cryptographic security services for electronic messaging applications: Authentication, message integrity and non-repudiation of origin (using digital signatures) and privacy and data security (using encryption). However both PGP and S/MIME have considerably different ways of providing these services.

A. S/MIME BASICS:

S/MIME was originally developed by RSA Data Security, Inc. It is based on the PKCS #7 data format for the messages, and the X.509v3 format for certificates.
SMIME basically provides two security services:

1. Digital signatures

   This is the most commonly used service in S/MIME. Digital signatures are the counterpart to legalized signatures on paper. The core functions of digital signatures are authentication, non-repudiation, and data integrity. Together they ensure the recipient that the message was infact from the sender and it has not been altered. However, digital signatures do not provide confidentiality. We need to use message encryption to provide confidentiality.

   For supporting non-s/mime clients, messages called clear signed and opaque signed messages have been defined. Clear signed messages are preferred when dealing with non-S/MIME clients, as the signature (signed message digest and digital ID) is carried as a standard MIME attachment outside the human-readable message body. In this way, the original message traverses the mail transport, which is followed by an unintelligible block of Base64-encoded S/MIME content. Standard mail clients simply see an attachment called `smime.p7s` full of illegible text, in addition to the standard message. The primary disadvantage of a clear signed message is that the mail gateways are free to translate character sets of the message, or re-encode MIME attachments, thereby altering the message’s contents. This breaks the digital signature (since the signature guarantees the original contents of the message are unaltered) and often yields false warning of message tampering.

   Opaque signing is designed to avoid problems caused by invasive mail gateways that can alter message formatting in transit, thus invalidating the signature. By wrapping the entire message in a MIME-encoded block of illegible text (usually Base64-encoded), the entire message is treated as an attachment, and is not altered by gateways that wish to translate character sets or re-code MIME headers. This guarantees the message, but it is illegible (not to be confused with encrypted) to the gateway. It is also illegible to non-S/MIME clients that can’t unwrap the envelope.


Message encryption is needed to conceal the messages so that no other person than the recipient can read the message. Message Encryption provides the following services:

- **Confidentiality**

   Message encryption is used to protect contents of a message. Only the intended recipient of the e-mail can read it contents and not anyone who can intercept and receive the e-mail. Message Encryption provides confidentiality while the e-mail is in transit and storage.

- **Data integrity**

   As similar to digital signatures, encryption provides data integrity because of the specific operations to do encryption.
While encrypting an e-mail we need the recipient unique information which is used to provide confidentiality.

When the recipient opens an encrypted message, a decryption operation is performed on the encrypted message. After the encrypted message is retrieved, unique recipient information (usually a public key) is also retrieved. Then decryption operation is performed using information that is unique to the recipient and known only to him. The message body is retrieved from the decrypted message.

Combining Digital signatures and Encryption.

Digital signatures and message encryption are not mutually exclusive. Both of them should be combined to attain all the security services provided by both encryption and digital signatures. Using only one of them will deny the user the security services provided by the other. Fig 5 and 6 depict how combining digital signatures and encryption can be done.

Key Management:

Public Key Cryptography and Digital Signatures

Public-Key Encryption uses a pair of keys. The one used for encryption is known as the public key, and this can be freely distributed. It can't decrypt messages, so it is not sensitive. The key used for decryption is the private key, which must be kept guarded. The reverse is also true, i.e. data encrypted with the private key can be decrypted with the public key. However, encrypting information with a truly public key is not a great practice, because all it takes to decrypt the message is the public key, which is often easily available. However using the private key on a message is useful for digital signatures.

When the message is received, the digital signature can be retrieved and the sender's public key applied in a decryption operation, which yields the original hash value of the message. A comparison of this hash value with the hash value of the received message can then be performed. Because only one private key can correspond to a public key, and only the owner of the public key could use it to encrypt the hash value successfully, decrypting the hash with the public key shows that the private key owner encrypted the hash value. Because the hash value is a numerical representation of the message text, if the encrypted hash value matches the hash value of the message received, it indicates that the message text that was sent matches the text that was received. When coupled with the fact that only the private key owner could have sent the message the result is that the recipient is assured that only the key owner sent the message. This provides authentication and, consequently, non-repudiation. It also shows that the message has not been changed, which provides data integrity. If the hash values did not match, the recipient would know that the message had either been altered in transit or that the public key used does not match the private key used. In both cases, the recipient knows that the message is not valid and should not be trusted.
Public Key Cryptography and Message Encryption

In public key cryptography as goal of message encryption is to ensure that only authorized recipients can view the message, the private key of each recipient is used to provide that service. Because the private key can only be successfully used by its owner, confidentiality is provided. As the public key is widely distributed, many people can send encrypted emails to one user.

In public key cryptography, message encryption is done but not for the entire message as using encryption and decryption on the whole message is expensive considering the complexity of the keys algorithms. So encryption here is just a part of the process that locks and unlocks information.

However, the key pair is not used on the entire message. This is because the encryption and decryption operation using a key pair is an expensive process, due to the necessary complexity of the keys' algorithms. Although a key pair needs to be used, it does not necessarily have to be used on the entire message. It needs to be part of the process that "locks" and "unlocks" the information. As long as the message is unreadable until the private key is presented, the goal of message encryption is met.

Symmetric keys called session keys are used to encrypt the message as they are computationally faster than the complex encrypting algorithms used in S/MIME. The session keys are encrypted using the public key of the recipient, who decrypts the session key using his own private key. After he retrieves the session key, he decrypts the message using that key. Figure 9 and Figure 10 clearly explain how that is done.

Digital Certificates:

A digital certificate is an electronic means of establishing your credentials when doing business or other transactions on the Web. It is issued by a certification authority (CA). User’s distinguished name, User’s public key, User’s Credentials, Serial number, Issuer name, Expiration date, copy of the certificate holder's public key (used for encrypting and decrypting messages and digital signatures), and the digital signature of the certificate-issuing authority so that a recipient can verify that the certificate is real.

Digital signature which meets ITU (International Telecommunications Union) Telecommunication Standardization (ITU-T) PKIX X.509 version 3 [RFC 2459] standard is generated based on

- Detailed information about the key holder
- An expiration date, after which the certificate is placed on the CA's CRL (Certificate Revocation List)
- A Compromised Key List (CKL)

The operating system stores a certificate locally on the computer or device that requested it or, in the case of a user, on the computer or device that the user used to request it. The storage location is called the certificate store. A certificate store will often have numerous certificates, possibly issued from a number of different certification authorities. Each implementation of S/MIME has its own certificate manager so that it can choose from the available certificates.

A free e-mail digital certificate from the certification authority Thawte Consulting Pvt Ltd was obtained for working on this project.
Understanding How Digital Certificates Are Structured

For a digital certificate to be useful, it has to be structured in an understandable and reliable way so that the information within the certificate can be easily retrieved and understood. For example, passports follow a similar structure allowing people to easily understand the information in a type of passport that they may never have seen before. In the same way, as long as digital certificates are standardized, they can be read and understood regardless of who issued the certificate.

The S/MIME standard specifies that digital certificates used for S/MIME conform to the International Telecommunications Union (ITU) X.509 standard. S/MIME version 3 specifically requires that digital certificates conform to version 3 of X.509. Because S/MIME relies on an established, recognized standard for the structure of digital certificates, the S/MIME standard builds on that standard's growth and thus increases its acceptance.

The X.509 standard specifies that digital certificates contain standardized information. Specifically, X.509 version 3 certificates contain the following fields:

- **Version number** The version of the X.509 standard to which the certificate conforms.
- **Serial number** A number that uniquely identifies the certificate and is issued by the certification authority.
- **Certificate algorithm identifier** The names of the specific public key algorithms that the certification authority has used to sign the digital certificate.
- **Issuer name** The identity of the certification authority who actually issued the certificate.
- **Validity period** The period of time for which a digital certificate is valid and contains both a start date and an expiration date.
- **Subject name** The name of the owner of the digital certificate.
- **Subject public key information** The public key that is associated with the owner of the digital certificate and the specific public key algorithms associated with the public key.
- **Issuer unique identifier** Information that can be used to uniquely identify the issuer of the digital certificate.
- **Subject unique identifier** Information that can be used to uniquely identify the owner of the digital certificate.
- **Extensions** Additional information that is related to the use and handling of the certificate.
- **Certification authority's digital signature** The actual digital signature made with the certification authority's private key using the algorithm specified in the certificate algorithm identifier field.

Because S/MIME requires an X.509 v3 certificate, this information also describes the characteristics S/MIME uses for its specific certificates.

Key Management in Digital Certificates

**Digital Certificates and Public Key Infrastructure**

One of the benefits of public key cryptography is that it reduces key management because one key pair takes the place of numerous symmetric keys. This benefit is further enhanced by digital certificates, which allow public keys to be distributed and managed. However, digital certificates are not self-managing. By design, digital certificates are widely circulated, so the management of these certificates must address the distributed nature of digital certificates. Digital certificates require a functioning infrastructure to manage the certificates in the context within which they are going to be used. Public key infrastructure (PKI) is inseparable from digital certificates. PKI is responsible for issuing certificates, ensuring the distribution of these certificates through a directory, and validating certificates. PKI is responsible for the underlying work that supports digital certificates and enables them to provide the capabilities that services such as S/MIME rely on.

Because of the size and complexity of the topic, PKI is beyond the scope of this book. The information presented here focuses on how PKI and digital certificates work in conjunction with message security. There are many excellent resources that address PKI. You can obtain more information about PKI from your PKI vendor's documentation and from other PKI-specific sources.

**How PKI Works with Message Security**

PKI provides the means for digital certificates to be used by issuing certificates and making them accessible through a directory. PKI also validates digital certificates by verifying the authenticity of the certificate, the validity of the certificate, and that the certificate is trustworthy. These services are crucial to digital certificates because digital certificates rely on a distributed model by using third-party certification authorities.
The specific way that digital certificates are issued and published to a directory depends on the specific PKI product and its implementation. In general, PKI issues digital certificates and publishes information about these certificates to a directory where that information can be accessed by other applications. Some of this information is used for validating digital certificates. As discussed in "Putting Public Key Cryptography Together with Message Security" earlier in this paper, message security operations require access to the public keys of both senders and recipients. Because the digital certificate provides this information, accessing users' digital certificates is crucial to a message security system. By providing access to digital certificates, PKI builds on the benefits that public key cryptography offers in terms of simplified key management by eliminating the need to manually exchange keys. Instead, PKI makes digital certificates available through a directory so that they can be retrieved by applications when needed.

To understand how PKI validates a certificate, remember the role that the certification authority has in issuing the digital certificate. The issuing certification authority vouches for the validity of the identity, and shows this by using its public key to sign the digital certificate. Checking the authenticity of a certificate means that the certification authority's digital signature must be verified. PKI validates a certificate by providing the means by which the issuing certificate authority's signature can be verified. If the signature cannot be verified, the certificate is known to be untrustworthy.

As mentioned in "Understanding Digital Signatures" earlier in this chapter, no security method is perfect. A digital certificate can be compromised, usually by loss of the private key. For digital certificates to be trustworthy, there must be a way to cancel or "revoke" a digital certificate before its expiration, much like a stolen credit card can be canceled. Certificate revocation is another of the critical services that PKI provides to support digital certificates and is another part of the process of verifying the digital certificate.

Because PKI ensures that digital certificates are trustworthy, PKI is an integral part of digital certificates. You cannot use digital signatures without PKI. Because Exchange Server 2003 supports X.509 v3 certificates, the specific PKI that supports an Exchange installation will depend on the digital certificates used with Exchange. From the standpoint of message security, however, all PKIs provide these fundamental services in support of digital certificates. The differences between specific PKIs are implementation and design related, and are specific to each PKI deployment.

**Putting Digital Certificates Together with Message Security**

With an understanding of digital certificates and how they support public key cryptography, the next step is to apply this information to message security. The next section shows you how digital certificates provide support for the core security services that comprise digital signatures and message encryption.

**How Digital Certificates Are Used for Digital Signatures**

The relationship of a public key to a user's private key allows a recipient to authenticate and validate a sender's message. Digital certificates provide support to public key cryptography by providing a reliable means to distribute and access public keys. When a sender is signing a message, the sender provides the private key that is associated with the public key available on the digital certificate. In turn, when the recipient is validating a digital signature on a message, the recipient is obtaining the public key to perform that operation from the sender's digital certificate. Figure 1.16 shows the sequence of signing with the addition of the supporting elements of digital certificates.

**How Digital Certificates Are Used for Message Encryption**

Just as digital certificates support digital signatures by making public keys available for the verification process, digital certificates also support message encryption by making public keys available so that the keys can be used for the encryption process. As discussed in "Public Key Cryptography and Message Encryption" earlier in this chapter, a sender can access the recipient's public key, which allows the sender to encrypt the message, knowing that only the recipient can decrypt the message. This time it is the recipient's digital certificate that makes the encryption possible. As with digital signatures, the public key from the digital certificate makes the operation possible.

**How Digital Certificates Are Used for Digital Signatures and Message Encryption**

Digital signatures and message encryption complement one another. Figure 11 shows the sequence of signing and encrypting with the addition of the supporting elements of a digital signature.
Figure 12 shows the sequence of decrypting and verifying the digital signature with the addition of the supporting elements of public key cryptography.

Obtaining a Digital Certificate

This flow chart describes how we obtain a digital certificate and how it is used for implementing all the security services as has been discussed in the previous sections.
The process of obtaining a digital certificate from Thawte has been described here:

The first step in Thawte's process was to create an ID number containing a country code and the national identification code.

What ever identity number is given to Thawte, that number is associated with the email account in the certificate.

This can't be a web-based mail account such as hotmail or Yahoo mail. The email that the sender and the receiver must use should be a POP3 or IMAP mail account such as GMU e-mail.

Then the CA asks the user to choose a password which must be remembered. Losing the password will lead us to dire consequences like not being able to renew or revoke certificates.

Then the CA sends an e-mail to the email account we specified to verify that the user who has requested the certificate is indeed himself. The user has to click on a verification link specified in that email to verify that he owns that e-mail account.

Then the username is created in their system and the user can login to the website of the CA and request a X.509 format certificate. The user requests the X.509 certificate and in the next step, he is asked about the browser he is using. The user selects a choice and after that he is allowed to configure the certificate extensions as shown in the figure below.

Then, the user is asked to chose the CSP (cryptographic service provider, code which comes installed with windows), which he does and he is asked to give his public key to the CA as shown in the figure. User’s private key is encrypted by the CSP after the key pair is generated and stored at C:\Documents and Settings\(username)\Application Data\Microsoft\Crypto\RSA in windows XP. The user cannot see his private key and it can only be accessed by the CSP to be used while signing or decrypting e-mails. He does that and the CA issues the certificate and sends an e-mail in a span of 10 minutes. The user can click on the link given in the email and can obtain the certificate, which is directly installed in to his browser’s certificate manager.

The method described above is to obtain your own personal certificate. To be able to send encrypted messages, we need the certificates of other people. There are 2 major methods to obtain them.

- Have the recipient send you a signed or encrypted email. You can save his certificate directly to your address book.
- Some organizations provide an LDAP directory from which you can obtain a user's certificate by clicking a link. The user’s certificate is saved on the hard disk and then imported into the address book (and in turn in to certificate store).

To determine whether we have someone's certificate, open the Certificate Manager and click the Other People's tab.

**Cryptographic algorithms used in S/MIME:**

S/MIME incorporates three public-key algorithms:

- Digital Signature Standard (DSS) is the preferred algorithm for digital signatures.
- Diffie-Hellman is the preferred algorithm for encrypting session keys.
- RSA is the preferred algorithms used for both signatures and session key encryption.

The hash function used to create the digital signature is SHA-1 but the 128 bit MD5 is also recommended for backward compatibility.

For Message Encryption, three key 3 DES is recommended. However, all the e-mail clients tested in the project support DES and RC2.

**How S/MIME Encrypted and Signed messages are sent using E-mail client**

After obtaining, verifying and importing the certificates to your certificate manager, we specify which certificate to be used for a particular e-mail account. After composing the e-mail, we click on the digitally sign and Encrypt options present on the e-mail clients’ toolbar and send the mail. The e-mail client automatically obtains the certificates of the user and his recipient from the certificate store. If it doesn’t find any of these certificates, it returns an error. Then it duly signs and encrypts the message using these certificates and then sends it.

**B. PGP Basics**

PGP is the second of the major solutions available to provide security to email. PGP was created in 1991 to allow individuals who wished to post on a bulletin board system (BBS) a way to send and receive secure messages. PGP’s concept is derived from a local grocery store near the home of creator Phil Zimmerman. The slogan was, “if you can't find it at Ralph's, you can probably get along without it.” This is the same theory that PGP has adopted. It may not be the most secure...
product on the market, but it does provide a high level of security for securing data.

In terms of general functionality, PGP is very similar to S/MIME in many regards but extremely different in others. PGP can secure any type of data, whereas S/MIME is focused on email. PGP can also be used to secure whole hard drives, but this project does not focus on these capabilities. PGP is an asymmetric key set up that provides message authentication and confidentiality through encryption and digital signatures. The PGP Corporation also claims that PGP provides non-repudiation through a web of trust. The encryption is through a set of public and private keys that either use Diffie-Hellman or RSA to create the keys. PGP can also sign a message with a SHA-1 hash. However unlike S/MIME its security is based in the overall trust of the user population. You can register any email to a PGP key regardless of whether you actually own that email or not. If enough people have clicked that they trust the user of the key, then a circle of trust is formed and the signature will appear as valid. The other downside of the “web of trust “ is that trust takes some time to build up, while with S/Mime the trust in the signature can be immediately accepted. In the case of PGP this level of trust from other users is the only thing that promises the recipient of the email, that the sender is who they say they are. With this system some level of non-repudiation can be accepted, but this level is still weaker than that of a traditional certificate authority. The security functions of PGP are the same regardless of the email program used. This is because PGP copies whatever is on the clipboard and then either encrypts, signs it or both. This can be done with both plaintext and files. The only added functionality that a PGP plug-in could provide is not having to copy the file or the text to the clipboard. After the file is pasted to the window, PGP has a wipe function that clears the clipboards memory. This process exceeds those of the Department of Defense’s media sanitization standards.

**Installation:**

Installation of PGP on a windows machine is extremely easy. All that is involved is saving the PGP freeware to your desktop, then clicking install. The file is self extracting and the default settings do not need to be changed. The only thing a user needs to choose is whether they want to install the features for instant messaging with ICQ or email using Eudora.

After the installation and the required reboot, a user generates their default key. The key generation process is also a very simple one. A user is prompted whether they want to use the default setting, or choose their own in the advanced window.

In the standard set up the user has to put in their name and email address. In the next step the user is asked to give a password that will be needed for any actions with the private key. Then the key and the subkey ring are generated. A possible problem with the key generation is that the key is generated on the user’s computer and the algorithms main input are the keyboard and mouse click of the user. Since in any language, some characters are used more than others, this presents the problem of having keys that are not uniformly distributed. If an attacker can have a higher probability of guessing the key, the encryption is considerably weaker. The advanced window has more options, these options include the algorithm type of either Diffie Hellman or RSA, as well as the key size. The key size can range from 1024 - to 4096, with 2048 as the default setting.

Any newly created keys can be uploaded to the server for use. There are two main servers for PGP keys, one in America and one in Europe. Uploading a key is a very simple process. All that is done to upload a key is click on the key tab in the PGP key section of the program, then select the server and click upload. The rest is done automatically. The key upload process is very easy to do, if you know where to look. However a user is never prompted by PGP to upload their keys to a server, so someone with less computer skill may be unaware that this step should be taken. It would be very easy for PGP to add a prompt asking a user if they would like to upload their public key to the server, and why it is good to do. Hopefully this can be in their next upgrade.

**Email Programs:**

The next step in using secure email is determining the email client to use. The main programs used by for email are Outlook and Outlook Express. Netscape’s email client, Eudora, as well as Mozilla’s thunderbird will also be examined for the business user. Ease of use will be determined by how many times a person has to use their own intuition or perform any mouse clicks that are outside of the main window of the email program they are using. Can the user just click sign to encrypt an email, or do they have to go through a process that involves them going to multiple windows or drop down menus? Measuring the functionality and security will involve comparing the different security options from each of the different email programs.

Netscape, Mozilla and AOL do not currently have plugins for PGP. However it is still possible to use PGP with these programs. This is because PGP works by encrypting the contents of the clipboard. It can be used to encrypt a word document, a picture or just text. If the email program being used is not listed above, this process will still work the same for any email program that does not have a plug in. This process is also the only
way to encrypt or sign data with the freeware version of the product, regardless of the email program used. Even though it installs the email plug-ins with the freeware, the functionality of the plug-ins is suspended unless you upgrade. This is to encourage users to spend the 60 dollars (or more) on getting a personal license (or higher), which is supposed to allow you to have one click encryption and decryption of files.

To encrypt or sign text, a user starts the PGP program and selects either current window or the clipboard. Then after they choose whether they want to encrypt, sign or both, the message is automatically copied to the clipboard and encrypted. In the case where the user has chosen to sign the message, they are prompted for their password, and if correct the message is signed. Then the results of the encryption and or signing are automatically posted back into the original window. This is slightly easier because it does not involve the user to cut and paste the contents of the file, which can actually be a problem for the more novice users.

To decrypt a file with a regular email client, such a Mozilla’s Thunderbird, the message would have to be decrypted to the plain text editor, and then it can be edited. This becomes a longer process when the message has an attachment. First the message has to be decrypted to the plain text editor, then it the attachment has to be downloaded to the hard drive. Once saved on the hard drive the file can be opened.

Eudora 6.2 is one of the three programs tested that PGP is supposed to support. However, the support the plug-in provides is far from perfect. The way Eudora is supposed to work is to remove the end user from having to cut and paste any data. First, the email is written just like any other email and any attachments are added. Next, the user has the option of encrypting the message, signing it or both by selecting the encrypt tab, the sign tab or both tabs. Each of these tabs are easily viewed in the window used to compose an email. Once send is clicked, the default settings are to send the email immediately, however this is not what happens. The email is first sent to the outbox where it waits for the program to be closed. When a user tries to close the program, it prompts the user if they would like the email to be sent. If the user clicks yes and there are no attachments in any of the emails selected to be encrypted with PGP, the program works. However if any of the emails have attachments, the program unexpectedly terminates and the Microsoft error reporting window appears. The message was finally sent on the fourth try after a reboot on Windows, and two crashes of Eudora. The email finally sent when the get mail button was pressed. However it broke the message into an unusual format. In this format the attachment was able to be downloaded, and the files were renamed 1.1 and 1.2. File 1.2 was the attachment and could be decrypted. However 1.1 was supposed to be the plaintext of the message and this part was unrecoverable.

Decrypting of a plaintext email in Eudora was a relatively simple process. All that a user needs to do is click the decrypt verify button which is clearly displayed. However the program once again failed to handle attachments in an appropriate manner. It prompts the user that the attached file, that is a PGP file, will not be able to be decrypted because it does not appear to the program that it is a PGP file. The only solution is to download the attachment and decrypt it like any of the programs that do not have PGP plug-ins. This is not an acceptable solution.

Outlook Express is another one of the programs that PGP is supposed to work with effortlessly, however this is not always the case. PGP is incorporated into one of the drop down menus of PGP with buttons that say sign, verify, encrypt and decrypt. It should work that a user can write out a message, and then choose whether they want to encrypt the message, sign the message or both. What actually happens is that PGP alerts the user that they have an encountered an error. Specifically it states that PGP is not able to copy the message to the clipboard for encryption. It requests that the user verify that the window they have selected is the window of the email they want to encrypt. In a first attempt to remedy this problem, the text to be encrypted is highlighted with a mouse or keyboard and once again the user clicks on the encrypt button, the sign button or both. Now the program does some of the steps correctly and some incorrectly. It acts just like when the encrypt/ sign options on the “use the active window” part of PGP are chosen. It will correctly encrypt or sign the message, but after it finishes it returns the same error message as before and posts a lower case a after the message. Attempting to decrypt message results in exactly the same problems. If the message is highlighted, it can be decrypted, but an error message is still displayed.

This is not the only problem with Outlook Express. Outlooks Express has a major problem with trusting its users to not open viruses. By default, it blocks almost all attachments as unsafe and does not allow the user to download them. This is great in the event that the attachment really is a virus, but in reality it blocks a number of useful attachments as well. One type of attachment it blocks is encrypted PGP attached files and public keys. This means that to even download the file, the file must be downloaded through another program. The idea that a user would have to forward the message on to a second email client just to be able to download the encrypted file is not acceptable. Because of all these problems, outlook express is the worst possible choice.
for an email client with PGP. It should work correctly, but what programs should do, and what they are able to is often two different worlds.

Outlook does everything that Outlook express is supposed to do. Microsoft full-featured email client, part of Microsoft office, seamlessly integrates PGP with its program. The sign and verify keys can be clearly seen from the main inbox menu, as well as a drop down menu named PGP. Unlike Outlook Express, these functions actually work. With two clicks in the main option menu, Outlook can be programmed to decrypt and verify messages automatically upon opening. This can also be done by clicking on the drop down menu to decrypt and verify on a per email basis. The automatic decryption function saves the user half the effort, because it will decrypt the message and the attachment in one click. Using the Encrypt and sign message automatically on send, the user can just attach the file like a regular email, instead of having to first encrypt the file and then attach it. Outlook makes PGP considerably easier to use, and provides the best functionality as well as ease of use.

### 3. FUNCTIONALITY

The following implementations of S/MIME and PGP have been analyzed for functionality:

**Microsoft Outlook 2003, Mozilla Thunderbird 0.8, Netscape 7.2 and Eudora 6.2**

This is a comparison of the various implementations for S/MIME and PGP. Eudora does not support S/MIME.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Outlook</th>
<th>Mozilla Thunderbird</th>
<th>Netscape</th>
<th>PGP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Installing Certificates</td>
<td>It uses Internet Explorer’s cache to obtain certificates. The certificate is exported to a location in the harddisk and then imported into Outlook for a specific e-mail account after providing a password. Other user’s certificates should be installed into the address book of Outlook.</td>
<td>The certificate is exported to a location in the harddisk and then imported into Thunderbird for a specific e-mail account after providing a password.</td>
<td>The certificate is imported directly from Netscape browser’s certificate cache. So, specifying which certificate to use is very easy in Netscape when compared to the other two implementations.</td>
<td>Not applicable</td>
</tr>
<tr>
<td>Ease of configuration</td>
<td>Hard to configure as we need to first export and again import providing the password 2 times.</td>
<td>Easier than outlook as we just import the certificate for an email account after providing the password only once.</td>
<td>It is the easiest of all. We just need to import from the browser cache. No need to provide any password.</td>
<td>Very easy for outlook.</td>
</tr>
<tr>
<td>Confidentiality</td>
<td>Yes. Using encryption provides confidentiality</td>
<td>Yes Using encryption provides confidentiality</td>
<td>Yes Using encryption provides confidentiality.</td>
<td>This is provided by either Diffie-Helman or RSA</td>
</tr>
<tr>
<td>Feature</td>
<td>Outlook Options</td>
<td>Other Email Options</td>
<td>Notes</td>
<td></td>
</tr>
<tr>
<td>-------------------------------</td>
<td>--------------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td><strong>Integrity</strong></td>
<td>Yes. Using encryption provides confidentiality</td>
<td>Yes. Using encryption provides confidentiality</td>
<td>This is provided through a SHA-1 Hash.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Authentication</strong></td>
<td>Yes. Using digital signature provides Authentication</td>
<td>Yes. Using digital signature provides Authentication</td>
<td>This is provided through a SHA-1 Hash.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Non Repudiation</strong></td>
<td>Yes. Provided through a certificate authority</td>
<td>Yes. Provided through a certificate authority</td>
<td>Provided through a web of trust</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>No of email accounts</strong></td>
<td>Multiple e-mail accounts are supported. You can use any number of email accounts with all of their outgoing servers (SMTP). But only one e-mail accounts signing and encryption certificate can be selected.</td>
<td>Multiple email accounts are supported but you can specify only one outgoing server (SMTP). This literally limits our secure e-mail usage to only one account. If you want to use other accounts also, you have to change SMTP settings each time you want to send a secure mail</td>
<td>Multiple email accounts are supported.</td>
<td></td>
</tr>
<tr>
<td><strong>Ability to choose the</strong></td>
<td>YES. Outlook has the option to let the user choose which encryption and hashing algorithm to use. It gives the options as SHA1 and md5 for hashing and 3DES, DES and RC2 (128, 64, 40bit) for encryption.</td>
<td>NO. It doesn’t give the user any such option though it supports all the algorithms SHA1, md5, 3DES and RC2 while receiving the email</td>
<td>Yes, AES is the default selection, with 3DES, Idea and Twofish being others.</td>
<td></td>
</tr>
<tr>
<td>algorithms to use.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Options for Validation</strong></td>
<td>Only the option that CRL checking can be enabled or not is given. Latest CRL’s have to be manually installed in to the client.</td>
<td>No settings are provided.</td>
<td>CRL is available but is not a default option. The CRL server and list has to be set up manually.</td>
<td></td>
</tr>
<tr>
<td>Revocation of certificates</td>
<td>Should request the CA. Doesn’t relate with the email client. Easy to remove the certificates after revocation.</td>
<td>Should request the CA. Doesn’t relate with the email client. Easy to remove the certificates after revocation.</td>
<td>Should request the CA. Doesn’t relate with the email client. Easy to remove the certificates after revocation.</td>
<td>Only a password is needed to revoke a key.</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
<td>-------------------------------------</td>
</tr>
<tr>
<td>Key exchange</td>
<td>Other people’s certificates have to be stored in to one’s address book or certificate manager to be able to decrypt and verify messages</td>
<td>Other people’s certificates have to be stored in to one’s address book or certificate manager to be able to decrypt and verify messages</td>
<td>Other people’s certificates have to be stored in to one’s address book or certificate manager to be able to decrypt and verify messages</td>
<td>This can be done through sending the public key or it can be downloaded from one of the two PGP servers.</td>
</tr>
<tr>
<td>Ability to chose which type of signed message to send (clear signed or opaque signed)</td>
<td>Yes. This option is available in Outlook</td>
<td>Not Available</td>
<td>Not available</td>
<td>Not available</td>
</tr>
<tr>
<td>Requesting S/MIME receipt for S/MIME signed messages.</td>
<td>Yes. This option is provided in Outlook. Using this option sender can get a receipt from the receiver that he received the sender’s message unaltered, which can use this to detect any attacks on the e-mail system.</td>
<td>No such option is provided</td>
<td>No such option is provided.</td>
<td>No such option is provided.</td>
</tr>
<tr>
<td>Ease with which message is decrypted.</td>
<td>The user just gets a warning that the message is encrypted and an application is requesting access to protected item (private key). Clicking on “ok” directly decrypts the message. The procedure is simpler.</td>
<td>The user is prompted for master password for accessing the CSP. This is a bit lengthy when compared to outlook. However the user can skip this by making the email client not to ask for master password each time.</td>
<td>The user is prompted for master password for accessing the CSP. This is a bit lengthy when compared to outlook. However the user can skip this by making the email client not to ask for master password each time.</td>
<td>This can be a lengthy process, as described in the email section of this paper. However, this is a very simple, one step process for Outlook.</td>
</tr>
<tr>
<td>Interoperability</td>
<td>Worked fine with secure e-mails sent using Netscape and Mozilla</td>
<td>Worked fine with secure e-mails sent using outlook and Netscape.</td>
<td>Worked fine with secure e-mails sent using outlook and Netscape.</td>
<td>Eudora, was not compatible with other programs. Everything else is compatible.</td>
</tr>
</tbody>
</table>
Vulnerabilities

Internet explorer’s certificate cache and windows certificate manager are unprotected. So any user logging on to the system can get access to the cryptographic services i.e. the private key. He can decrypt or send encrypted messages. We suggest putting a password for accessing cryptographic services as in other implementations.

Cost

Pretty high

Freely available.

There is a master password for accessing cryptographic services. So, only one user can access the private key.

There is a corporate version along with a free one which offers more features.

Can be set up for free or 120 for the Outlook set up.

3. Conclusion:

Recommendations for the end users

There are two main groups of users for email with two separate sets of specific needs. These groups are the individual or home user and the business user. Both the small and large businesses of the world have a higher need for secure data than a home user. Also, both business groups will probably have access to a computer specialist who could assure the more advanced features of these standards are installed correctly. Because of this, the paper has combined these two sub groups into a general group.

The home user will look for the easiest way to implement a product and low costs. Often they have marginal computer skills and do not require as high a level of security for their emails as the business user. The home user will most likely be using one of the following email programs: Outlook, Outlook Express, Eudora, Mozilla Thunderbird, Netscape or AOL. The business user will be concerned about functionality and security and will probably be using Outlook, Eudora, Netscape or Mozilla. While an end user for a business may not have a strong background in computers, they will have access to a system administrator who will.

For personal use where ease of use is the main concern Outlook with PGP is the best choice. This is because regardless of the users computer knowledge, the PGP default settings, along with Outlooks default settings work perfectly. The only thing a user would have to do is click the PGP drop down menu and choose encrypt and/ or sign. Anyone who can use email should be able to do this. The automatic decrypt and verify function works very well and does not require the encrypted files to be saved to the computer. This is also the only email program with plug-ins that did not have a major error when either decrypting or encrypting attached files.

The only downside of this set up is that neither Outlook nor the personal editions of PGP software are free. The cost for this set up ranges $120 to $145 (before tax and shipping). The PGP personal edition costs $60 and Microsoft Outlook ranges from $60 (for anyone who qualifies for the student and teacher edition) of Microsoft Outlook to $85 (for everyone else.) For people looking for a free set up that is still relatively easy to use, there are multiple options. The choices are PGP using Mozilla or Netscape and S/Mime using Netscape. Netscape with S/MIME is slightly easier to use for plaintext message and considerably easier to use for messages with attachments. Netscape with S/MIME offers all the basic security services like authentication, integrity, confidentiality and non-repudiation offered in other software packages. Netscape’s advantage is that it is very easy to configure when compared to other implementations of S/Mime. This is because the home user, who may be a novice computer user, may not know whether to send clear signed messages or which cryptographic algorithms to choose for encryption and hashing. The myriad of options in Outlook and paucity of options in Mozilla rule themselves out, as Netscape offers a good balance of security and ease of use. So, we recommend Netscape if a home user wants to use S/MIME. However, since obtaining a S/Mime certificate is, in general, harder than obtaining a key in
PGP, either Netscape or Mozilla is appropriate for the user concerned about installing the security service. Because of these advantages and disadvantages, the user looking for low cost and easy implementation does not have a clear cut winner.

While there is an argument for either PGP or S/Mime in a personal setting, the same is not true for the business situations. The damage to a business from a security violation is often considerably greater than that of a home user. While the web of trust provides adequate level of non-repudiation, S/MIME provides this at a higher level, assuming the user or company trusts the CA. Because of the lower level of trust associated with the PGP email, it is not as appropriate in a business setting. Another disadvantage of PGP is that it interferes with an employer’s ability to read employee email. In the United States emails are owned by the business, not the individual who sent them. Many employers need to have a way to recover the emails in case an employee has to be terminated in less than optimal circumstances. However, PGP private keys are protected by a hand chosen password, which leaves a couple of less than optimal choices. One of these is to have the system administrator create all the keys by hand after talking with the end users to find out their desired password. This presents a number of problems, with some of the main problems being large scale implementation and how the systems administrator and the end user securely share the password. The second option is to have some sort of program create random passwords so the system administrator does not have to spend his time doing this task. The problem with random passwords is that people may put the passwords on sticky notes so they don’t forget and this makes the passwords considerably less secure. Outlook with S/MIME has some very strong security features. Some of these features are: the ability to select the algorithms to encrypt and hash, ability to chose clear signed or opaque signed messages, best certificate manager among all the 3 implementations. Outlook is also very easy to set up in a corporate setting so that a user can easily request the company’s internal CA to issue a certificate. We recommend Outlook for small and large business use.

APPENDIX:

Configuring outlook, Mozilla thunderbird and netscape for S/MIME.

Netscape:
To send signed or encrypted emails, you must first configure one of your mail accounts for S/MIME. Follow these steps:

1. Open the Tasks menu and choose Mail & Newsgroups. (If you haven’t already set up your mail account for the test profile, you will be asked to do so before you can proceed.)
2. Open the Edit menu and choose Mail & Newsgroups Account Settings.
3. In the left panel, identify the account you want to configure and click Security under the name of that account.

The Security panel contains two sections: Digital Signing and Encryption:

Fig 14. Configuring Email accounts

The Digital Signing section allows you to select the certificate you want to use for signing email messages. The certificate you select is attached to every message you sign (including all encrypted messages), allowing the recipient to verify your digital signature. The signature is created using the private key for this certificate, which remains on your hard disk.

The Encryption section allows you to select the certificate you want to use for encrypting email messages. This certificate is attached to every signed and/or encrypted message you send, so that all recipients can subsequently use it to encrypt emails they send to you. Decryption is performed with the private key for this certificate, which remains on your hard disk.

4. In the Digital Signing section, click Select.

This dialog box appears:

Fig 15 Selecting Certificates
5. Select the signing certificate you want to use for testing, then click OK.

6. In the Digital Signing section, select the checkbox labeled "Digitally sign messages".

7. In the Encryption section, click Select. You can then use a similar dialog box to select the certificate you want to use for encryption. (Note that the certificate selected for signing can be the same as the one used for encryption if the certificate supports both signing and encryption.) For now, leave the radio button labeled "Never" selected. You will still be able to encrypt messages on an individual basis.

8. Click OK.

You must select both a signing certificate and an encryption certificate before you can start sending signed messages. When you sign a message, both your encryption and signing certificates are attached to it, so that recipients can verify your digital signature and, if they wish, send you encrypted email.

**Signing and Encrypting Email Messages**

1. Click the Compose button (or reply to, or forward, an existing message).

A new Compose window opens.

When you receive a message, the right side of the heading area (in current builds) indicates whether the message is signed or signed and encrypted. (Encrypted messages are always signed.)

If you have configured S/MIME as described above, you can digitally sign any email message. To encrypt a message, however, you must already have an encryption certificate for each of the recipients.

A. **Signing and Encrypting Email Messages**

To digitally sign an email message, follow these steps:

1. Click the Compose button (or reply to, or forward, an existing message).

A new Compose window opens.

2. Open the Options menu, then the Security submenu.

If you followed the directions in this document exactly, you'll notice that Digitally Sign is already checked, because of the Mail & Newsgroup Account option you selected earlier.

If you want to digitally sign the message only, without encrypting it, you don't need to do anything with this menu (assuming that the Digitally Sign item is already selected). If you want to encrypt the message as well as signing it, select Always Encrypt. (In current builds, Always Encrypt applies to this Compose window only.)

3. Address and compose your test message, then click Send.

When you receive a message, the right side of the heading area (in current builds) indicates whether the message is signed or signed and encrypted. (Encrypted messages are always signed.)

If you have configured S/MIME as described above, you can digitally sign any email message. To encrypt a message, however, you must already have an encryption certificate for each of the recipients.

Mozilla has a method very similar to this. So we are not going to discuss about that now.

**OUTLOOK**

**Signing All Outbound Messages**

1. Tools > Options…

2. Click the “Security” tab.

3. Check the “Add digital signature to outgoing messages” checkbox.

4. Also check the “Send clear text signed message when sending signed messages”.

Fig 17: Configuring Outlook for S/MIME

Back-up Certificates

1. Click the “Import/Export…” button.
2. Select the “Export your Digital ID to a file” radio button.
3. Click the “Select…” button.
4. Choose the Certificates you wish to export from the list and then click the “OK” button.
5. In the “Filename” field, type a filename for your exported certificate.
6. To protect your exported certificates, enter a password and confirm.
7. Click the “OK” button again.
8. You will need to enter the password for your certificate at this time and click “OK” (do not check the “Remember password” checkbox – this will defeat the “High” level of security on your certificate).
9. Click the “OK” button.

References:

[1]. Using S/MIME in Mozilla


[3]. S/MIME and PGP
http://www.imc.org/smime-pgmpmime.html

[4]. Microsoft TechNet
http://www.microsoft.com/technet/prodtechnol/Exchange/guides/E2k3MsgSecGuide/3a1da037-8c65-420a-b41a-5f7260731db2.mspx

[5]. Paper: S/MIME and PGP a comparison
http://www.thawte.com/support/index.html

[6]. Thawte Technical Support
http://www.thawte.com/support/index.html


[8]. Step by Step procedure to configure Mozilla thunderbird
http://www.lehigh.edu/helpdesk/docs/thunderbird/

[9]. Beginners Guide to secure email using S/MIME:

[10]. Review: Secure Email Clients: Not quite ready for S/MIME
http://www.nwc.com/902/902r2.html

[11]. Paper: S/MIME or PGP how will you secure your e-mail?
http://www.worldtalk.com/Standards%20and%20Tec