

The Case for Email Security

by Erik Kangas, President, Lux Scientiae

Section 1: Introduction to Email Security



You may already know that email is insecure; however, it may surprise you to learn just how insecure it really is. For example, did you know that messages which you thought were deleted years ago may be sitting on servers half-way around the world? Or that your messages can be read and modified in transit, even before they reach their destination? Or even that the username and password that you use to login to your email servers can be stolen and used by hackers?

This article is designed to teach you about how email really works, what the real security issues are, what solutions exist, and how you can avoid security risks.

Information Security and integrity are becoming more important as we use email for personal communication and business. While you are reading this article imagine how security problems can affect your business or personal life.... if they have not already.

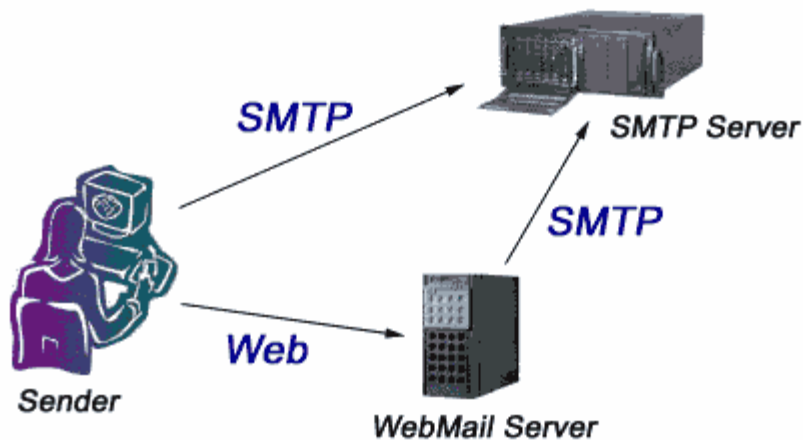
Section 2: How Email Works

This section describes the general mechanisms and paths taken by an email message on its route from sender to recipient. This should give you an overview of the different protocols (languages) involved, the different types of servers involved, and the distributed nature of email networks. The examples I present are representative of many common email solutions, but are by no means exhaustive.

Sending an Email Message

Sending an email message is like sending a letter. When you send a letter you drop it off at your local post office. Your local post office looks at the address and figures out which regional post office the letter should go to. Then the regional post office looks at the address and figures out which local post office is closest to your recipient. Finally, the recipient's local post office delivers your letter to its recipient. Computers are like "post offices", and the "Simple Mail Transport Protocol" (SMTP) is the "procedure" which a post office uses to figure out where to send the letter next. Any program that sends an email message uses SMTP to deliver that message to the next "post office" for relaying to its final destination.





Most people send mail in two ways - with a web-based interface like Yahoo! or Hotmail, or with an "email client" program like Outlook or Eudora.

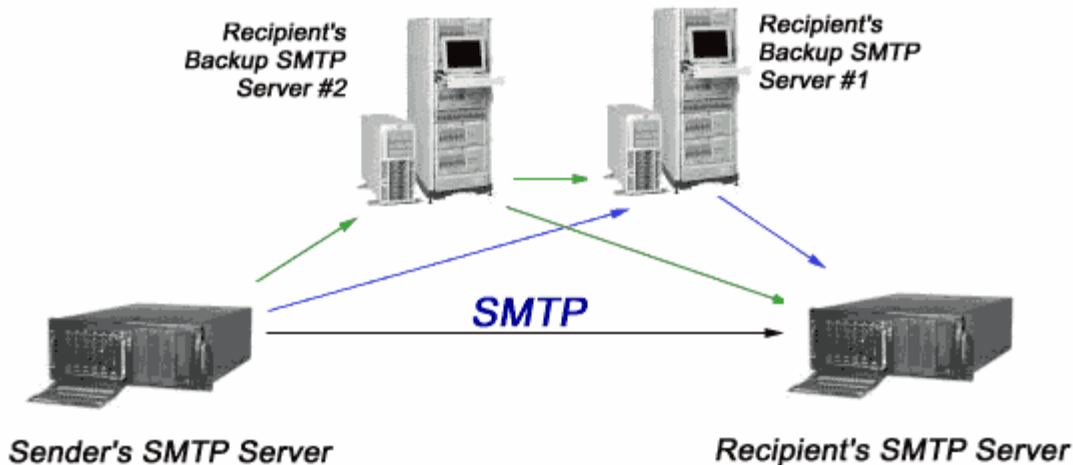
When you send a message with an email program on your personal computer (or your cell phone or PDA), you have to specify a server so that your email program knows where to send the message. This server is like your local post office. Your email program talks directly to the server using the computer protocol (language) known as SMTP. This is like dropping off a letter at the local post office.

When you use **WebMail** your personal computer uses an internet connection to communicate with a web server. The "language" that the internet connection uses is HTTP - "HyperText Transfer Protocol". When you send your message with WebMail the web server contacts its SMTP server and sends your message to it.

Delivery of email from your SMTP Server to your recipient's SMTP Server:

When an SMTP Server receives an email message, it first checks if it has an inbox for the message recipient. If it does not it must "relay" that email message to another SMTP server closer to the recipient. This is analogous to how your local post office forwards your letter to a regional post office. This process is known as "email relaying".

How does your SMTP Server know where to relay the message to? If the recipient's email address is "bob@luxsci.net", then the recipient's domain name is "luxsci.net". Part of the "DNS settings" for the recipient's domain (these are the "mail exchange" or MX records for the domain; see also **Understanding Domain Name Service (DNS)**) includes an ordered list of SMTP Servers that expect to receive email for this recipient. The highest priority SMTP Server listed is the recipient's actual SMTP Server; the others are "backup SMTP Servers". These backup servers merely queue email for later delivery to the recipient's actual SMTP Server.



There are many scenarios that govern the path an email message may take from the sender's to the recipient's SMTP Server. Some of these include:

1. The sender's server successfully contacts the recipient's server and sends the email message directly (black line in the figure).
2. The sender's server can not contact the recipient's actual SMTP server (maybe the recipient's server is busy, down, or has some other connection problem). In this case the sender's server tries to contact and deliver the message to the recipient's first backup server.
3. The sender's server can not contact the recipient's actual SMTP server or its first backup server. In this case the sender's server tries to contact and deliver the message the recipient's second backup server.
4. The sender's server can not contact any of the recipient's servers. In this case it will queue the message and try to send it later. It will keep retrying periodically for several days until it succeeds in sending or gives up.

Any message delivered to the backup servers goes through the same process of trying to contact the recipient's actual SMTP Server, or a higher priority backup server. Backup servers may also queue email for later sending. (Note that a recipient may have zero or more backup servers, not necessarily two as in this example).

Once the email message arrives at the recipient's SMTP Server and is delivered to the recipient's email box, the recipient may pick up the message and read it whenever s/he chooses (discussed below).

Each server that receives your message adds its "Received" stamp to the message. This stamp identifies what server received the message, at what time, and *from what other server*. This information allows the recipient to see a message's entire journey.

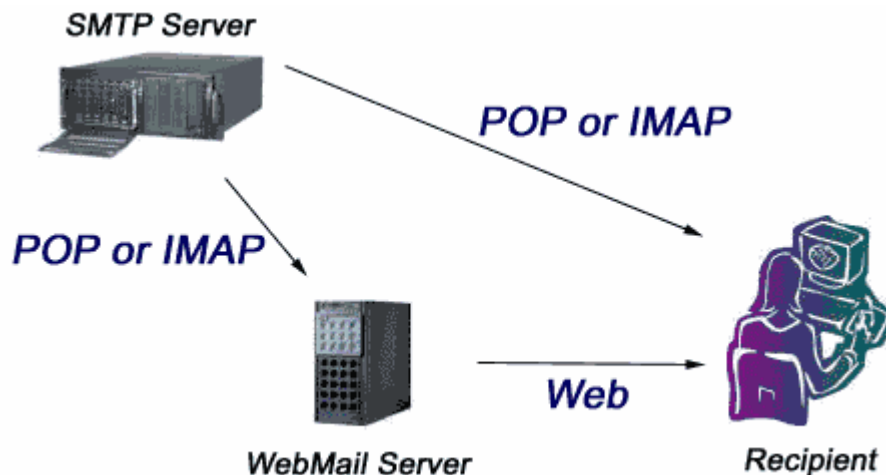
What should be clear from this discussion so far is that:

- All email servers communicate with each other using SMTP

- You never know how long it will take an email message to get from sender to recipient because you don't know how busy the servers are, how much traffic there is on the Internet, what machines are down for maintenance, etc.
- Your messages may sit in queues on any number of servers for any amount of time. Some of these servers may belong to third parties (i.e. may not be under the purview of either the sender or the recipient).
- Your recipients can determine the Internet address and name of the computer from which you are sending your messages.

Retrieving Email From an SMTP Server

When you receive an email message it sits in a file in your SMTP Server. If you wish to view this email message you must access this file. Any computer wishing to access your file must speak one of the languages the SMTP Server does. With some exceptions, there are really only 2 languages that email computers understand (for email retrieval, as opposed to email sending, for which they use SMTP), one is called the "Internet Message Access Protocol" (IMAP) and one is called the "Post Office Protocol" (POP). (We will not discuss the details of these here, but you may be interested in [Understanding Email Services](#) for information about them.)



As a recipient, you can generally retrieve your email by either using a web-based interface known as "WebMail", or via an "email client" program, such as Microsoft Outlook or Eudora, running on your personal computer. The email client programs will talk directly to your email server and speak IMAP or POP. With WebMail, your computer will talk to a WebMail server using a web connection (speaking HTTP); the WebMail server will, in turn, talk to your email server using POP or IMAP.

The Lack of Security in Email

Email is inherently insecure. In the following sections, we will see just how insecure it is. At this stage, it is important to point out the insecurity in the email delivery pathway just discussed:

- **WebMail:** If the connection to your WebMail server is "insecure" (i.e. the address is http:// and NOT https://), then all information including your username and password is not encrypted as it passes between the WebMail server and your computer.
- **SMTP:** SMTP does not encrypt messages. All communications between SMTP servers send your messages in plain text for any eavesdropper to see. Additionally, if your email server requests that you send your username and password to "login" to the SMTP server in order to relay messages to other servers, then these are also sent in plain text, subject to eavesdropping. Finally, messages sent via SMTP include information about which computer they were sent from and what email program was used. This information, available to all recipients, may be a privacy concern.
- **POP and IMAP:** The POP and IMAP protocols require that you send your username and password to login; these credentials are not encrypted. So, your messages and credentials can be read by any eavesdropper listening to the flow of information between your personal computer and your email service provider's computer.
- **BACKUPS:** Email messages are stored on SMTP servers in plain, unencrypted text. Backups of the data on these servers may be made at any time and administrators can read any of the data on these machines. The email messages you send may be saved unexpectedly and indefinitely and may be read by unknown persons as a result.

These are just a few of the security problems inherent in email. In the next section, we will talk about communications security problems in general so we can see what else can go wrong. Later on, we will see how these problems can be solved.

Section 3: Security Threats to Your Email Communications



This section describes many of the common security problems involved in communications and email in particular.

Eavesdropping: The Internet is a big place with a lot of people on it. It is very easy for someone who has access to the computers or networks through which your information is traveling to capture this information and read it. Just like someone in the next room listening in on your phone conversation, people using computers "near" the path your email takes through the Internet can potentially read and copy your messages!

Identity Theft: If someone can obtain the username and password that you use to access your email servers, they can read your email and send false email messages as you. Very often, these credentials can be obtained by eavesdropping on SMTP, POP, IMAP, or WebMail connections, by reading email messages in which you include this information, or through other means.

Invasion of Privacy: If you are very concerned about your privacy, then you should consider the possibility of "unprotected backups", listed below. You may also be concerned about letting your recipients know the IP address of your computer. This information may be used to tell in what city you are located or even to find out what your address is in some cases! This is not an issue

with WebMail, **POP**, or **IMAP**, but is an issue when sending email, securely or insecurely, from any email client over **SMTP**.

Message Modification: Anyone who has system administrator permission on any of the SMTP Servers that your message visits, can not only read your message, but they can delete or change the message before it continues on to its destination. Your recipient has no way to tell if the email message that you sent has been altered! If the message was merely deleted they wouldn't even know it had been sent.

False Messages: It is very easy to construct messages that appear to be sent by someone else. Many viruses take advantage of this situation to propagate themselves. In general, there is no way to be sure that the apparent sender of a message is the true sender - the sender's name could have been easily fabricated.

Message Replay: Just as a message can be modified, messages can be saved, modified, and re-sent later! You could receive a valid original message, but then receive subsequent faked messages that appear to be valid.

Unprotected Backups: Messages are stored in plain text on all SMTP Servers. Thus, backups of these servers' disks contain plain text copies of your messages. As backups can be kept for years and can be read by anyone with access to them, your messages could still be exposed in insecure places even after you think that all copies have been "deleted".

Repudiation: Because normal email messages can be forged, there is no way for you to prove that someone sent you a particular message. This means that even if someone DID send you a message, they can successfully deny it. This has implications with regards to using email for contracts, business communications, electronic commerce, etc.

Section 4: Symmetric and Asymmetric Encryption in a Nutshell

In order to understand how we can mitigate the security problems described in Sections 2 and 3, a basic knowledge of the two main types of encryption will be very useful. This section presents these concepts in a simple, straightforward form.

Message Digests are quick ways to check to see if a message has been altered. If you have a digest of the original message and compare it with a digest of the message you just received and they match, then you know that the message has been unaltered.

Asymmetric Key Encryption

In asymmetric key encryption, also known as "public key" encryption, each person has TWO keys. Any cyphertext created using one of the keys can ONLY be decrypted using the other key. For example, say you have keys "K1" and "K2". If you encrypt your message with K1, then ONLY K2 can be used to decrypt it. Similarly, if you encrypt using K2, ONLY K1 can be used to decrypt it. This is distinctly different from symmetric key encryption where you only have one key that performs both functions on the same message.

In asymmetric key encryption, the two keys that each person possesses are commonly named the "private" and "public" keys because the "public" one is published or given out freely to anyone who wants a copy and the "private" one is kept secret. The security of asymmetric key encryption depends only on whether you can keep your private key secret.

Asymmetric key encryption allows you to do many clever things:

- **Send an Encrypted Message:** To send a secure message to someone, all you have to do is encrypt it with their public key! Only the intended recipient who has the matching private key will be able to decrypt and read the message. This solves the problem of eavesdropping and the problem of sending secret keys that is inherent in symmetric key encryption.
- **Prove You Sent A Message:** To prove to someone that you sent a message, you can encrypt the message (or just a piece of it) with your private key. Then, anyone can decrypt it with your public key and read the contents. The fact that your public key decrypts the message proves that only you could have sent it.
- **Sign a Message:** A message signature proves that you sent the message AND allows the recipient to determine if the message was altered in transit. This is done by using your private key to encrypt a digest of a message at the time of sending. The recipient can decrypt this digest and compare it to a digest of the received message. If they match, then the message is unaltered and was sent by you.
- **Encrypted, Signed Messages:** The most secure form of communication is to first add a signature to the message and then to encrypt the message plus signature with the recipient's public key. This combines all of the benefits of all of the techniques: security against eavesdropping and unexpected storage, proof of sender, and proof on message integrity.

Section 5: Securing Your Email With SSL

The easiest thing you can do to make your email more secure is to use an email provider that supports "Secure Socket Layer" (SSL) for their WebMail, POP, IMAP, and SMTP servers.

SSL is a combination of asymmetric and symmetric key encryption mechanisms. If you connect to a server using SSL, the following things happen (roughly):

1. The server uses its private key to prove to you that it is in fact the server that you are trying to connect to. This lets you know that you are not connecting to a "middleman" that is trying to intercept your communications.
2. You send the server your public key.
3. The server generates a "secret key" and sends it to you encrypted using your public key.
4. You and the server then communicate using symmetric key encryption using this shared secret key. (Symmetric key encryption is faster than asymmetric key encryption).

The benefits of SSL are twofold: 1. you can determine if you are connecting to the right server, and 2. you and the server can communicate securely.

If you get any warning messages when connecting to a server using SSL, you should think twice about ignoring them. While your provider may just have a small technical problem that is causing the warning, these warnings can also indicate that your communications are being intercepted. These warnings usually indicate one of the following:

1. The server's SSL "certificate" (i.e. public/private key pair) has expired.
2. Some of the information in the certificate doesn't match the information you expect -- i.e. the certificate was issued for a different server name than the one you are trying to connect to. (You could be inadvertently connecting to the wrong server.)
3. The certificate was issued by an untrusted agency.

SSL certificates are (generally) issued by third party agencies such as Thawte.com or Verisign. These 3rd party companies do a background check on companies that request certificates, and only issue certificates if the companies have a right to them. The certificate includes the name of the company, the name of the issuing company, and the name of the server to which it is issued. When you connect to an SSL server you can verify this embedded information and the fact that it was issued by a third party company that you trust. If the certificate is valid then you can have a high degree of confidence that the server you are connecting to is the server you want to reach.

By using SSL for WebMail, POP, IMAP, and SMTP you ensure that communications between your personal computer and your email service provider's computers will be encrypted. Your message contents, username, and password will be hidden from eavesdroppers -- but only hidden from eavesdroppers between you and your service provider! SSL services do not protect your messages once they leave your SMTP Server and head to their destinations. So, it doesn't really protect your message contents, but it does completely protect your username and password from detection. This is very important because it prevents identity theft, forged messages, etc.

SSL is very easy to use. It usually only involves clicking a few checkboxes in the configuration of your email client. It is transparent to your recipients - you can use SSL for these services even if your recipients do not. These measures protect you and your password. Because it is so easy and because the security you receive is much better than no security, we strongly encourage the use of SSL for email communications whenever possible.

Section 6: Privacy with Anonymous SMTP

In Sections 2 and 3 we wrote that when you send email via any email client (but not WebMail), your computer's Internet address is included in the message for all recipients to see. Depending on your Internet Service Provider's privacy standards and what kind of connection and service you have, this information may be used to determine what region or the country you are in, what city you are in, or even what your address is! This could be a serious issue for people very concerned about their privacy. Additionally, other information such as what email program you are using, is also visible to the recipient.

Anonymous SMTP services, or re-mailers, provide a good way of keeping all of the functionality of SMTP that you require, while giving you back your privacy. These services typically receive your message via (Authenticated) SMTP (ideally they would also support SMTP over SSL, as described above) and then "scrub" the message, removing all information about your computer's address, your email program, and any other non-standard information. These services then re-mail your scrubbed message to the intended recipients.

The end result is that the recipients get the message just as they would have without the "Anonymous" service, except that they can only track the message back to your Anonymous SMTP server. They know who you are, based on your email address and your message content, but they have no way of knowing **where you are** or what email programs you are using.

Most anonymous SMTP services log all information that they scrub out of messages and track all activity. So, while your recipients do not know where you are, your email service provider *does*. This prevents this type of service from providing any real benefit to people who send unsolicited or forged email -- their service provider can quickly respond to complaints or abuse, identify the sender, and terminate the account and/or bring legal measures to bear.

What it does provide is a level of privacy in sending of email that is functionally equivalent to the level of privacy you get from sending through a WebMail interface (unless the WebMail in question makes it a point to add your computer's address into the outgoing message -- most do not).

LuxSci offers Anonymous SMTP services to all clients who have SMTP Relaying services. This service is compatible with LuxSci's SMTP over SSL services as well and only requires changing the port you use to connect to LuxSci's SMTP servers.

Section 7: Asymmetric Key Encryption and Email (PGP and S/MIME)

While SSL protects your password and your message contents to some extent, it does not solve any of the other problems we have discussed: repudiation, encryption, unwanted backups, message modification, etc. This is because SSL only protects the message path between you and your SMTP Server and stops there. Even with SSL, the messages are stored on your SMTP Server in plain text.

The ultimate solution is to use asymmetric key encryption to provide message signatures and/or encryption. This completely solves the issues of:

- Eavesdropping (everything is always encrypted)
- Message modification (message digests are used)
- Message replay (you can include a timestamp in the signature)
- Repudiation (signatures allow proof of who sent the message)
- Unprotected backups (everything is always encrypted)

Asymmetric key encryption should be used in combination with SSL so that your username and password are also protected. Why? These credentials are not part of the message and thus would not be encrypted along with the message.

Fortunately (or unfortunately), there are two widely used forms of asymmetric key encryption for email: S/MIME and PGP. Both allow you to add signatures and/or encryption to your messages. PGP can be obtained from PGP.com and is compatible with standard email clients. S/MIME is built into many email clients like Microsoft Outlook, but you must obtain an S/MIME certificate from a third-party company such as Thawte.com.

Interoperability Problems

PGP and S/MIME solve many problems, but they also create another: interoperability. One interoperability issue is that PGP and S/MIME are completely incompatible! If you are using PGP and your friend is using S/MIME, you will not be able to send each other secure messages.

That said, PGP has been an Internet standard (OpenPGP - RFC 2440) since 1997 and PGP-encrypted email accounts for well over 90% of the current encrypted email traffic on the Internet. So, using PGP will make you compatible with the majority. However, the majority doesn't matter when you're trying to contact the minority that use S/MIME. It is useful to know that some email clients, such as Microsoft Outlook, can be configured to use BOTH PGP and S/MIME so that you can correspond securely using whatever method is necessary at the moment.

The other interoperability issue involves "key exchange". If you want to send your friend an encrypted message, you first need his/her public key; if your friend wants to prove that you signed a message or that the message that you sent him/her was unaltered, s/he first needs your public key. So there is the necessity of trading public keys before secure communication can begin. There are various ways to trade keys (including email). PGP offers "key servers" from which your correspondents' keys can be downloaded to make the process easier. However, not everyone has their PGP keys listed on a key server, let alone the same key server, and not everyone uses PGP, so the key exchange issue is still an impediment to sending secure messages -- especially if you have to send them quickly.

Section 8: Compatible Security with Escrow Encryption

Escrow encryption uses a trusted "encryption middleman" to give you the same security offered by asymmetric key encryption, but with universal compatibility. Here is how it works:

1. The sender connects to the middleman's web mail portal on a secure SSL connection
2. The middleman validates the sender.
3. The sender creates a message.
4. The message sender makes a secret question and answer for the recipient. Only the recipient knows the answer to the question.
5. The middleman encrypts the message and stores it on his server.
6. The middleman sends a plain text message to the recipient that contains only a secure link to the middleman's web mail portal, and a unique message password that is part of the encryption key. The middleman then 'forgets' this password so that he **can not** decrypt the message until he gets the password back from the recipient.
7. The recipient connects to the middleman's portal over a secure SSL connection and logs in with the message password .
8. The middleman asks the recipient the secret question. If she answers correctly the middleman decrypts the message and presents it to the recipient.

The encryption middleman handles all the encryption dirty work; it doesn't matter if the sender uses PGP and the recipient uses S/MIME. In fact, it doesn't matter if either uses encryption at all! All that the sender and recipient need is a secure internet connection. The middleman takes care of everything else.

How does it solve the security problems we mentioned earlier?

- Eavesdropping: No one can eavesdrop on the message because the sender and recipient connect to the middleman on a secure SSL connection.
- Identity Theft: No one can steal the sender's login information or the secret question/answer because both the sender and the recipient use SSL connections.
- Invasion of Privacy: The recipient knows nothing about the sender's computer, email client, or location. She only knows that he used the middleman.
- Message Modification: No one can modify the message because it never leaves the middleman's server.
- False Messages: The message is only accessed on the middleman's server, so no one can pretend to send it.
- Message Replay: No one can modify the message because it never leaves the middleman's server.
- Unprotected Backups: The message is encrypted when it is stored, so it is secure even in backups.
- Repudiation: The recipient knows that the sender really did send the message because he was validated by the middleman.

In addition the middleman can keep a log of who accesses the message and at what times. Thus the sender can audit the message to see who has viewed it. Notice that the message is secure and anonymous; The message is encrypted and stored on the middleman's servers, so it is not subject to the security of intermediate relaying servers. Only the middleman can encrypt and decrypt the message, and only authorized recipients can access the message. The recipient knows nothing about the sender's computer, only that he used the middleman. As long as the middleman is

trustworthy, the message is completely secure, completely anonymous, and completely compatible.

LuxSci's *SecureLine* service provides complete, compatible security through escrow encryption, as well as supporting PGP and S/MIME modes of encryption.

Section 9: Conclusions

Email is, in general, *Completely Insecure!* The security issues include:

- Eavesdropping
- Identity Theft
- Invasion of Privacy
- Message Modification
- False Messages
- Message Replay
- Unprotected Backups
- Repudiation (Sender denies that s/he sent it)

SSL: It is simple and easy to use SSL to secure the communications between your computers and your email service provider's computers. This works no matter who your recipients are. SSL improves security in these ways:

- It establishes that you are contacting your service provider's computers and not someone else's
- It encrypts the username and password that you use to login to these servers. This mitigates identity theft and other issues.
- It protects your message from eavesdroppers between your computer and your SMTP server.

Anonymity: If you have access to an Anonymous SMTP server, you have an easy way to increase your Internet privacy. Anonymous SMTP provides:

- IP address privacy so that message recipients can not determine your computer's Internet address (and thus your location).
- Email client privacy so that the recipients of your email messages cannot determine what type of email client you are using.
- A means to strip out any other non-standard "email header" data that may be lurking in your outbound messages.

PGP and S/MIME: PGP and S/MIME keys use asymmetric key encryption to protect the contents of your messages throughout their complete journeys. They provide:

- Protection against eavesdropping and unwanted backups
- Message Digests to detect whether messages have been altered in transit
- Signatures to prove sender authenticity

I highly recommend the use of SSL for email communications. Unfortunately, PGP and S/MIME are not being used as extensively as they should be. In my experience, more and more companies are using SSL to encrypt communications with their email servers, but few are using PGP or S/MIME for encryption. I see the impediment being that the effort needed to setup, to enforce usage, and to train employees is seen as much larger (or costlier) than the benefit of use. Clearly, the cost savings gained by using secure messaging is in having less information leakage or modification which is very difficult to quantify, especially as most companies assume that they don't (or won't) have significant problems in this arena anyway. These assumptions will be changing.

Escrow encryption avoids the impediments of PGP and S/MIME encryption while providing complete security, anonymity, and other features. There is no learning curve for escrow encryption because it can be easily integrated into an existing web mail system. Any one who is concerned about security but doesn't know how to set it up can use escrow encryption as easily as they use web mail. In addition, it can communicate securely with anyone who has an internet connection.

Unlike computer break-ins and other security problems, problems with email security are very hard to detect. You cannot tell if someone is reading your email or modifying messages subtly until it is too late. You cannot quantify the cost of email and information security problems until it is too late - imagine all of the things people write in their messages.... and think twice.