Securing E-mail with Digital Certificates

Author: Steve Moitozo <smoitozo@bates.edu>

Created: 11/15/2006
Revised: 02/04/2009
Revision: 4

Table of Contents

Introduction

By default e-mail is insecure. Most people who use e-mail have heard and believe this statement. However, many do not know why e-mail is insecure and how to secure it. Securing e-mail doesn't have to be difficult. Armed with an understanding of the risks and techniques for mitigating them, people can take responsibility for their own e-mail security.

Risks to E-mail Security

Why is e-mail insecure? In order to understand why e-mail is insecure we must first have a clear understanding of what we're attempting to protect. We want each individual message to be safe. We're not attempting to secure the transmission system or the storage system. We simply want some assurance that our messages are safe.

Safe from what? There are many risks to e-mail messages:

1. Unauthorized deletion of a message: someone could gain access to your e-mail account and delete one or more messages.
2. Unauthorized modification of a message: someone who has access to your e-mail message could change it to say something different.
3. Unauthorized viewing of a message: someone who has access to your e-mail could read one or more messages.
4. Sender impersonation: someone could send you a message that appears to be from someone else.
5. Discoverable channels of communication: someone who has access to your e-mail could see who you are communicating with.

Number 5 is beyond the scope of this document. If you want to communicate with someone and don't want anyone to know with whom, e-mail is not the best tool for you.

Mitigating Risks

Unauthorized deletion of a message

There are a number of ways this could happen; here are a couple:

1. A person with access to the mail server could delete one or more messages. Good management and security measures at the server level can mitigate this risk.
2. An unauthorized person could access your e-mail account and delete one or more messages. Good password management and the use of encrypted authentication systems can mitigate this risk.

Unauthorized modification of a message

A person who has access to your e-mail could modify one or more messages without your consent or knowledge. The use of message digests, produced by a cryptographic hash function,[1] help to mitigate this risk or at least alert you to such occurrences.

Unauthorized viewing of a message

A person who has access to your e-mail could view one or more messages without your consent or knowledge. The use of encryption[2] can mitigate this risk.

[1] http://en.wikipedia.org/wiki/Cryptographic_hash_function

[2] http://en.wikipedia.org/wiki/Encryption

Sender Impersonation

Anyone can configure an e-mail client to send an e-mail to you which appears to be from someone else (e.g., Joe can send and e-mail to Alice which appears to come from Jane). The use of digital signatures[3] can help to verify the sender's identity.

Encryption

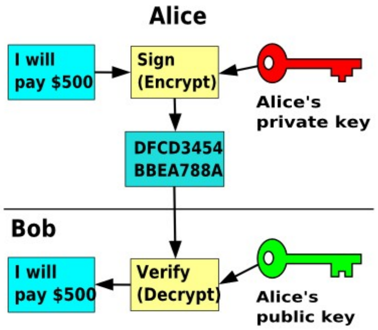
Mention encryption and most people think of the symmetric-key cryptography.[4] A basic example is the secret decoder ring. However, symmetric-key cryptography becomes unwieldy when more than two or three people need to communicate because of the issues associated with key distribution (e.g., decoder rings) and the implications of a stolen key. Public-key cryptography[5] provides a better solution for securing e-mail communications with larger numbers of people. Two well-known examples of public-key cryptography in action are Transport Layer Security[6] (TLS) and its predecessor Secure Sockets Layer (SSL).

Crash Course in Public-key Cryptography

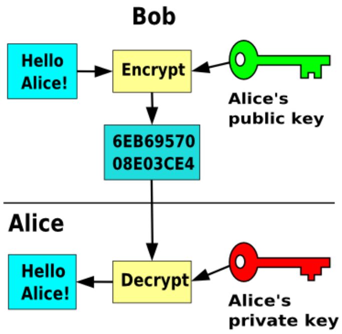
Public-key cryptography uses a two part key for encryption. Part one is the private key, the secret and highly protected part. Part two is the public key, the non-secret and widely distributed part. The public key is used for encrypting and the private key is used for digitally signing.

Digitally signing a message has a number of benefits. A digital signature gives the recipient a method of verifying the identity of the sender. This is known as sender authentication. It also provides a way of telling if the message was altered after it left the sender, known as message integrity. Lastly, a digital signature makes it very difficult for the sender to claim that they never sent a message, or non-repudiation. Anyone can digitally sign e-mail.

Message encryption provides secrecy. Encrypting a message with a person's public key makes the message accessible only to that person. All other people who come in contact with the message will see a large block of meaningless characters, making it extremely difficult to eavesdrop.



Message Signing: Alice signs the message with her private key and Bob verifies the message signature with Alice's public key.



Message Encryption: Bob encrypts the message with Alice's public key and Alice decrypts the message with her private key.

[3] http://en.wikipedia.org/wiki/Digital_signature
[4] http://en.wikipedia.org/wiki/Symmetric_key_cryptography
[5] http://en.wikipedia.org/wiki/Public-key_cryptography
[6] http://en.wikipedia.org/wiki/Transport_Layer_Security

Whatever action is performed with the private key is reversed with the public key and whatever action is performed with the public key is reversed with the private key. If Alice signs with her private key, Bob verifies the signature with her public key. If Bob encrypts with Alice's public key, Alice decrypts with her private key.

It is important to note that when something has been encrypted using the public key it cannot be decrypted with the public key, the way it works with symmetric-key cryptography. Public-key cryptography is sometimes referred to as asymmetric cryptography for this reason.

Once these basics are understood a few questions arise. How does Bob get Alice's public key? Public-key cryptography was invented to deal with the issue of key distribution. This feature is what makes Public-key cryptography so much better for the purpose of securing communication than symmetric-key cryptography. In the above examples depicting message signing, Alice signs a message and sends it to Bob. Alice's public key is sent along with the signed message. The simple act of signing and sending a message is all that Alice needs to do in order to get Bob her public key.

The second question that is usually asked at this point is, “When Bob receives Alice's public key how does he know it's authentic?” This issue of trust has resulted in two types of trust frameworks. The first is called the web of trust[7] (WOT) and the second is called public key infrastructure[8] (PKI). While PKI is a centralized key management approach, WOT is a distributed approach. While WOT binds a public key and a user (identity certificate[9]) to other identity certificates in a web, PKI relies on a central certificate authority[10] or chain of authorities. This can get very technical very quickly; I will try to keep this simple.

WOT is similar to a social networking situation where Bob introduces Alice to Jane thereby transferring at least a minimal amount of trust on behalf of Alice to Jane. This is done when Bob signs Alice's public key. As Jane discovers that more of the people in her web also know Alice (by looking at the signatures on Alice's public key) her level of assurance about Alice's identity grows.

PKI takes a different approach. A group of people get together and they all decide that they will trust a central authority that will sign all identity certificates. In a PKI environment Jane simply decides to trust the same certificate authority that signed Alice's public key. The fact that Bob knows Alice is irrelevant.

Implementations of Public Key Cryptography for Securing E-mail

PGP/GPG: a Web of Trust Solution

PGP and its compatible, open sibling GPG are implementations of the web of trust approach to securing e-mail. They provide the ability to encrypt e-mail as well as files on a computer. Typically these solutions are not tightly integrated with applications such as e-mail clients and require extensions and/or plug-ins in order to function. PGP/GPG solutions for securing e-mail typically have a bit of a “geek factor” to them. That is to say, they require more than an average understanding of the technology and are probably not the best solution for widespread deployment, particularly to non-technical people.

S/MIME: a Public Key Infrastructure Solution

S/MIME is a standard method of integrating PKI into e-mail clients. It provides an elegant, tightly integrated environment in which to secure e-mail communication. Many modern e-mail clients support S/MIME out of

[7] http://en.wikipedia.org/wiki/Web_of_trust
[8] http://en.wikipedia.org/wiki/Public_key_infrastructure
[9] http://en.wikipedia.org/wiki/Identity_certificate
[10] http://en.wikipedia.org/wiki/Certificate_authority

the box, making it an easy choice for securing e-mail communications within an organization of mostly non-technical people.

S/MIME Secured Communication Infrastructure

The important elements of a S/MIME secured communication infrastructure are:

1. A user base that is informed about threats to e-mail and methods of protecting e-mail.
2. A certificate authority (CA) to sign user certificates.
   2.1. A secure method of requesting a certificate from the CA.
   2.2. A secure method of retrieving the signed certificate from the CA.
   2.3.A secure method for the CA to revoke certificates.
3. Wide distribution and support of e-mail clients that implement S/MIME.

Using S/MIME to Secure E-mail

Getting a Certificate

To begin using S/MIME to secure e-mail a user must request a certificate from a certificate authority. The CA will use various methods to authenticate the identity of the user. When the CA is satisfied that the user is who they claim to be the CA will generate and issue the certificate. In this case the certificate is composed of the user's new private key and public key bundled together and protected with a password in a PKCS#12[11] file format.

NOTE: All certificates generated by a CA have an expiration date. Typically, S/MIME certificates expire in 365 days.

The following is a list of certificate authorities which issue S/MIME certificates for securing e-mail communications.

Certificate Authority	*URL*	*Cost*	*Expires*	*Product Name*	*Notes*
CAcert	http://www.cacert.org	free	365		CA is not yet included in all browsers and e-mail clients
Comodo/InstantSSL	http://www.comodo.com http://www.instantssl.com	free or $7.20	365 days		Free for non-business use
GeoTrust	http://www.geotrust.com	$19.95	365 days	My Credential	
ipsCA	http://www.ipsca.com	free	3 mos	Personal E-mail Certificates	
Thawte	http://www.thawte.com	free	365 days	Personal E-mail Certificates	combines S/MIME with a web of trust assurance model
VeriSign	http://www.verisign.com	$19.95	365 days	Digital IDs for Secure Email	

[11] http://en.wikipedia.org/wiki/PKCS

Some institutions choose to establish their own certificate authorities used for issuing S/MIME certificates. Examples include Dartmouth College[12]. Bates College[13] used to use its own CA but has been using a commercial CA for since 2007.

Installing a Certificate

Without specifically mentioning client software the overall process of installing an S/MIME certificate is as follows:

1. Launch the e-mail client software.
2. Navigate to the area within the application that provides functionality for managing certificates.
3. Import the certificate.
4. Typically e-mail clients that support multiple accounts will require the additional step of mapping the certificate to the e-mail account for which the certificate was issued.

Bates College provides instructions for requesting, installing and configuring Thunderbird to use S/MIME at the following URL: http://www.bates.edu/smime-certificate.xml

Protecting the Private Key

What if an unauthorized person could send digitally signed e-mail that has been signed with your certificate? The result could be catastrophic. For this reason, it is important to protect the S/MIME private key with a strong passphrase.[14] It is never advisable to use a weak password or, even worse, no passphrase at all.

Signing Messages

It is usually recommended that e-mail client software be configured to digitally sign messages by default. Signing messages is typically transparent and the act of sending a message causes the e-mail client to automatically apply the signature to the outgoing message. There is no need for the recipient's public key in the signing process.

Distributing Your Public Key

Since there is no need for the recipient's public key in the signing process the simple act of sending a signed message is a great way to distribute a person's public key. The sender's public key is included in the signature. Modern e-mail clients that support S/MIME will automatically validate the signature and save the sender's public key, as long as they recognize the CA. Thus the act of sending a signed message to someone is all that is needed in order to prime future encrypted e-mail communications.

Sending Encrypting Messages

Typically, e-mail client software will provide a simple button or menu selection that the operator can use to indicate that an outgoing e-mail should be encrypted. For example, the mechanism in Thunderbird is a lock icon in the composition window labeled S/MIME. Clicking the arrow next to the lock reveals a few options

[12] https://collegeca.dartmouth.edu
[13] http://www.bates.edu/crypto/
[14] http://en.wikipedia.org/wiki/Passphrase

regarding encryption and signing.

Before a person can send an encrypted e-mail to someone the sender must first have the recipient's public key. As mentioned above the simplest method for obtaining someone's public key is to have them send a signed e-mail message. Since the process of sending encrypted e-mail requires the public key of the recipient it is not usually advisable to configure e-mail client software to encrypt by default. It will depend upon the environment.

Receiving Encrypted Messages

When receiving encrypted messages the recipient's private key is required. Without it the recipient will not be able to read any messages that were encrypted using the corresponding public key. This is extremely important, if the recipient's private key has been deleted any encrypted e-mails that have been received will be totally unreadable. It is also important to note that an expired private key can still be used to access messages that have been encrypted using the corresponding public key. Therefore, it is important that expired certificates remain installed if the user desires to access past encrypted messages.

Certificate Revocation Lists (CRLs)

A Certificate Revocation List[15] is a certificate authority's list of certificates that it would like to revoke. I say it this way because certificate authority's would like to revoke these certificates but they really can't unless people have configured their software to pay attention to the CRLs. Unfortunately, most people don't know what CRLs are, let alone take the time to configure their software to check them. In addition many SSL/TLS-enabled programs are finicky about how they handle CRLs, if they even handle them at all.

Ideally a person would review the certificate authorities that have been configured in the software being used and would disable the ones that are not trusted and add/enable the ones that are. With that step complete the user would configure the software to use the CRLs for the trusted CAs. However, the process of deciding which CAs to trust and which ones not to trust requires a level of understanding and research capabilities that is either beyond the average user or simply not perceived as being valuable. In fact, I know very few (fewer than three) people who actually do all this.

Nevertheless, it is advisable that people configure their software to consult the CRLs for the CAs that the software is set up to trust.

Conclusion

The security of e-mail is at risk. Still, it is possible to mitigate many of the risks so that messages are safe from all but the most resourceful attackers. A person who is armed with an understanding of the risks as well as some tools and techniques for mitigating them can do an adequate job of securing e-mail communications.

[15] http://en.wikipedia.org/wiki/Certificate_revocation_list