International Journal on Future Revolution in Computer Science & Communication Engineering
Volume: 5 Issue: 5
ISSN: 2454-4248
80 – 83

Pretty Good Privacy: An e-mail Security Protocol

Vibha Ojha[1], Ravinder Singh[2]
Govt. Engineering College, Ajmer
[1]*vibha.ojha@gmail.com*, [2]*ravikaviya@gmail.com*

Abstract : Security has been an issue in mail from ancient times. Security is still important today.E-mail is as fast and casual as a voice phone call, but can be save and retrieved withinfinitely greater efficiency than paper letters or taped conversations. Security in maildeals first with reliable delivery to the addressee. Security, that is confidential, reliableand known delivery is essential to the success of e-mail. In other words people will notuse a mail system that they cannot trust to deliver their messages. This paper describes the basic approaches for e-mail security and discusses the advanced email security mechanism i.e. Pretty Good Privacy (PGP)

Keywords *: E-mail, Cryptography, PGP*

*****

1. Introduction

An e-mail system is made up of two primary components that reside in an organization's IT infrastructure: mail clients and mail servers.

Users read, compose, send, and store their e-mail using mail clients. Mail is formatted and sent from the mail client via the network infrastructure to a mail server. The mail server is the computer that delivers, forwards, and stores e-mail messages. All components-the mail servers, the mail clients, and the infrastructure that connects and supports them-must be protected.

Voluntary industry standards (e.g., SMTP, ESMTP, POP, IMAP) for formatting, processing, transmitting, delivering, and displaying e-mail ensure interoperability among the many different mail client and server solutions.

E-mail security relies on principles of good planning and management that provide for the security of both the e-mail system and the IT infrastructure. With proper planning, system management, and continuous monitoring, organizations can implement and maintain effective security.

2. E-mail Security

In its simplest form, mail is a direct message from the sender's machine to the receiver's machine. But machines are often only intermittently connected. Thus the electronic post officebox. A computer connects every so often to the electronic post office box to exchange mail. Since the machine implementing the post office is constantly receiving messages, mail is no longer dropped.

The mail infrastructure consists of a mesh of mail forwarders, called Message Transfer Agents **o**r MTAs. Typically there is more than one path between a sender and receiver. Usually, the rules for forwarding mail are configured manually.

Email has a slew of security issues:

- **Privacy**: The ability to keep anyone but the intended receiver to read the email.
- **Message flow confidentiality**: In addition to privacy, a third party cannot even know whether there was an email or not.
- **Authentication**: The receiver knows the identity of the sender.
- **Integrity**: The receiver knows that the message has not been altered after leaving the control of the sender.
- **Non-repudiation**: The receiver can prove to a third party that the sender really did send this message.
- **Proof of submission**: The sender can prove that *this* message to the receiver left his control.
- **Proof of delivery**: Verification that the recipient received the message.
- **Anonymity**: The ability to send a message so that the receiver cannot find out the identity of the sender.
- **Containment**: The ability of the network to keep certain security levels of information from leaking out of a particular region.
- **Audit**: The ability of the network to record events that might have security relevance.
- **Accounting**: The ability of the network to maintain usage statistics, e.g. for charging users.
- **Self destruction**: An option for the sender that allows the message to be destroyed after the receiver received it.
- **Message sequence integrity**: Reassurance that all emails were delivered in the order that they were sent.

 ISSN: 2454-4248
80 – 83

3. Basic Approaches for E-mail Security

Confidentiality is about making sure that no one (other than the intended receipient) can read your emails. Ordinary emails have no confidentiality because they are sent in the clear for anyone to read. They usually pass through many different parties before reaching the receiver, so any them or other evesdropers could read it. It is like writing your message on a postcard when you would rather have the message put inside an envelope.

Integrity is about making sure that the message the receiver gets is the same message the sender sent. Ordinary emails have no integrity, since there is no way for the receiver to detect that a message has been tampered with.

Authentication is about making sure of who the email came from. Ordinary emails have no authentication, and anyone can make a fake email claiming it came from you.

Secure email can give us confidentiality, integrity and authentication.

3.1 Encrypting

A message encrypted with your public key can only be decrypted using the corresponding private key. Encryption (as you know from spy movies) is scrambling a message and decryption is unscrambling it.

If someone wants to send you a secret message, they simply encrypt it with your public key. Anyone can get hold of your public key, so anyone can send you an encrypted message. But only you have the private key, so only you can decrypt and read the message. No one else has your private key, so no one else can read the message. Similarly, if you want to send someone a secret message, you would encrypt it with their public key.

Encrypting an email is how we get confidentiality.

3.2 Signing

A message signed by a private key can have that signature verified by the corresponding public key.

Think of a signature as encryption with the two keys reversed. You sign a message by "encrypting" it with your private key. Only the corresponding key can decrypt it. But everyone can have your public key, so anyone can decrypt it. If the decrypted data matches the message, then the receiver can be sure that only you (the holder of the corresponding private key) could have created that signature. No one else could have forged that signature, because no one else can guess what your private key is. Also, if the message has been tampered with, the "decrypted" signature will not successfully match the message so the receiver will know that it is not what you signed.

Digital signatures on an email proves integrity and provides authentication of the sender.

3.3 Certificates

We have assumed that the other person knows that a particular public key is yours. This is where certificates come in.

A certificate is a piece of data used to establish an identity. It contains your public key and information about you (e.g. name and email address). This way, a person can get hold of your public key and also know that it is your public key.

The certificate is issued by an authority. The certificate is digitally signed by the authority, so (like a signed email) it cannot be forged or tampered with. The theory goes: if you trust an authority, then you trust that they are associating the correct public key with the correct person.

There can be a chain of certificates, where each certificate is digitally signed by another private key which has an associated certificate. This continues until the root certificate which is a self signed certificate -- one that is signed by its own private key. Email programs are usually preloaded with a set of trusted root certificates.

For our purposes, we will nearly always deal with certificates. You will very rarely handle just the raw public key. If a program want to use the public key, we will usually give it the certificate and it will extract the public key from it.

Note: a lot of programs and documents incorrectly use the term "certificate" to refer to the public key or (worse) refer to the private key as a certificate. They also sometimes calls the key-pair a certificate. They also sometimes refer to a private key and one or more certificates as a certificate. This is very confusing. In this article, a certificate will only refer to the digitally signed data issued by a certificate authority (it contains a public key as well as other information).

3.4 S/MIME and X.509v3

This guide will be using Secure Multipurpose Internet Mail Extensions (S/MIME), which is a specification for how to represent digitally signed and encrypted emails.

S/MIME uses X.509v3 certificates, which is a standard that specifies the format and information inside a certificate. We will be using X.509v3 certificates issues by commercial certificate providers.

4. Pretty Good Privacy (PGP)

Pretty Good Privacy or PGP is a popular program used to encrypt and decrypt email over the Internet, as well as authenticate messages with digital signatures and encrypted stored files.

Previously available as freeware and now only available as a low-cost commercial version, PGP was once the most widely used privacy-ensuring program by individuals and is also used by many corporations. It was developed by Philip R. Zimmermann in 1991 and has become a de facto standard for email security.

 ISSN: 2454-4248
80 – 83

Pretty Good Privacy uses a variation of the public key system. In this system, each user has an encryption key that is publicly known and a private key that is known only to that user. You encrypt a message you send to someone else using their public key. When they receive it, they decrypt it using their private key. Since encrypting an entire message can be time-consuming, PGP uses a faster encryption algorithm to encrypt the message and then uses the public key to encrypt the shorter key that was used to encrypt the entire message. Both the encrypted message and the short key are sent to the receiver who first uses the receiver's private key to decrypt the short key and then uses that key to decrypt the message.

PGP comes in two public key versions -- Rivest-Shamir-Adleman (RSA) and Diffie-Hellman. The RSA version, for which PGP must pay a license fee to RSA, uses the IDEA algorithm to generate a short key for the entire message and RSA to encrypt the short key. The Diffie-Hellman version uses the CAST algorithm for the short key to encrypt the message and the Diffie-Hellman algorithm to encrypt the short key.

When sending digital signatures, PGP uses an efficient algorithm that generates a hash (a mathematical summary) from the user's name and other signature information. This hash code is then encrypted with the sender's private key. The receiver uses the sender's public key to decrypt the hash code. If it matches the hash code sent as the digital signature for the message, the receiver is sure that the message has arrived securely from the stated sender. PGP's RSA version uses the MD5 algorithm to generate the hash code. PGP's Diffie-Hellman version uses the SHA-1 algorithm to generate the hash code.

5. Benefits of PGP Protocol

PGP combines some of the best features of both conventional and public key cryptography. PGP is a hybrid cryptosystem. When a user encrypts plaintext with PGP, PGP first compresses the plaintext. Data compression saves modem transmission time and disk space and, more importantly, strengthens cryptographic security. Most cryptanalysis techniques exploit patterns found in the plaintext to crack the cipher. Compression reduces these patterns in the plaintext, thereby greatly enhancing resistance to cryptanalysis. Followings are the some benefits to use PGP protocol:

- Sensitive information is always protected. It cannot be stolen or viewed by others on the internet. It assures that the information that is sent or received was not modified in transmission and that files were not changed without your knowledge.
- Information can be shared securely with others including groups of users and entire departments.
- You can be certain who the email is from and who it is for. PGP verifies the sender of the information to ensure that the email was not intercepted by a third party.
- Your secure emails and messages cannot be penetrated by hackers or infected by email attacks.
- Others cannot recover sensitive messages or files once you have deleted them.

Conclusion

E-mail security remains unstandardized and unstructured. Even though there are various software programs and schemes on the market to address this issue, most businesses can expend resources on stop gap measures until standards are uniform or do nothing at all. Companies are now forced to choose between hardware solutions such as firewalls or some of the various combination of non-interoperable software. PGP protocol is the option for such requirements.

The PGP protocol will eventually mature and reach the point in which e-mail security will be the norm. It will allow users to send unsigned encrypted messages, signed but unencrypted messages and signed and encrypted. The program should encrypt messages for storage. It will make it possible to send messages to a single receiver or to multiple receivers. The PGP protocol will be available on all sorts of platforms.

References

[1] S. Garfinkel, PGP: Pretty Good Privacy. Sebastopol, CA:O'Reilly Media, Inc., 1995.

[2] A.Whitten and J. Tygar, "Why johnny can't encrypt: A usabilityevaluation of pgp 5.0," in Proceedings of the 8th USENIXSecurity Symposium, vol. 99, 1999.

[3] "FREQUENTLY ASKED QUESTIONS – FAQs." Hosting and Internet Access URL:http://www.walkontheweb.com/faq/main_faq.htm

[4] N. Borisov, I. Goldberg, and E. Brewer, "Off-the-recordcommunication, or, why not to use pgp," in Proceedings ofthe 2004 ACM Workshop on Privacy in the Electronic Society,ser. WPES '04. New York, NY, USA: ACM, 2004, pp.77–84.

[5] S. L. Garfinkel and R. C. Miller, "Johnny 2: A user test of keycontinuity management with S/MIME and Outlook Express,"in First Symposium on Usable Privacy and Security (SOUPS2005). Pitsburg, PA: ACM, 2005, pp. 13–24.

[6] Tim Richardson . "Simple Notes on Internet Security and Email." May 28, 2001. URL:http://www.tim-richardson.net/misc/security.html

[7] S. Sheng, L. Broderick, C. Koranda, and J. Hyland, "WhyJohnny still can't encrypt: Evaluating the usability of emailencryption software," in Poster Session at the Symposium OnUsable Privacy and Security, Pitsburg, PA, 2006.

ISSN: 2454-4248
80 – 83

[8] Schneier, Bruce. E-mail security, How to keep your electronic messages private,(ISBN 0-471-05318-X) John Wiley & Sons,1999

[9] C. Herley, "So long, and no thanks for the externalities: Therational rejection of security advice by users," in SeventeenthNew Security Paradigms Workshop (NSPW 2009). Oxford,England: ACM, 2009, pp. 133–144.

[10] Canter,Sheryl . "E-Mail Encryption." PC Magazine. The 1997 Utility Guide.URL http://www.zdnet.com/pcmag/features/utility/encrypt/_open.htm

[11] S. Fahl, M. Harbach, T. Muders, and M. Smith, "Confidentialityas a service–usable security for the cloud," in EleventhInternational Conference on Trust, Security and Privacy inComputing and Communications (TrustCom 2012). Liverpool,England: IEEE Computer Society, 2012, pp. 153–162.

[12] N. Unger, S. Dechand, J. Bonneau, S. Fahl, H. Perl, I. Goldberg,and M. Smith, "SoK: Secure messaging," in Thirty-Sixth IEEESymposium on Security and Privacy (S&P 2015). San Jose,CA: IEEE Computer Society, 2015, pp. 232–249.

[13] Google, "Browser security handbook, part 2," accessed October13, 2015. https://code.google.com/p/browsersec/wiki/Part2.

[14] P. Stone, "Pixel perfect timing attacks with HTML5,"2013, accessed October 13, 2015. http://www.contextis.com/documents/2/Browser Timing Attacks.pdf.

[15] W. C. Garrison III, A. Shull, S. Myers, and A. J. Lee,"On the practicality of cryptographically enforcing dynamicaccess control policies in the cloud," in Thirty-Seventh IEEESymposium on Security and Privacy (S&P 2016). San Jose,CA: IEEE Computer Society, 2016.