

An Enhanced Pretty Good Privacy (EPGP) System with Mutual Non-Repudiation

Gregory L Vert
Assistant Professor of Computer Science,
Dept. of Computer Science, University of
Nevada-Reno, Reno, NV 89557
e-mail: gvert@cs.unr.edu

Manaf A Alfize
Instructor of Information Technology,
SYSCOMS College, Abu Dhabi,
United Arab Emirates
e-mail: manaf_alfize@hotmail.com

Abstract

Enhanced Pretty Good Privacy (EPGP) is a new cryptosystem based on Pretty Good Privacy (PGP), used for the purpose of secure e-mail message communication over an open network. The idea of EPGP, introduced in this paper, addresses PGP's main drawback of incomplete non-repudiation service, and therefore, attempts to increase the degree of security and efficiency of e-mail message communication.

Keywords: PGP, EPGP, Non-Repudiation, NRO, NRR, MNR, Security.

1 Introduction

Since the beginning of the era of e-mail message communication over open networks, the security of electronic mail (e-mail) has been a growing concern [1]. There have been several protocols developed and implemented to assure the security of such communication. These protocols provide the security services of message confidentiality and message authentication for an e-mail message, basically.

Some e-mail security protocols that have been developed include: Simple Mail Transfer Protocol (SMTP) [2], Multipurpose Internet Mail Extension (MIME), and its enhancement, known as Secure MIME (S/MIME) [3]. Other protocols are: Certified Exchange of Electronic Mail (CEEM) [4], Secure E-mail Protocol (SEP) [5], Privacy Enhanced Mail (PEM) [6], and Pretty Good Privacy (PGP) [7].

Pretty Good Privacy (PGP) [7] is an integrated cryptographic system, invented by Phillip Zimmerman in 1991, in order to establish the integrity, authenticity, compactness, confidentiality, and compatibility of e-mail message communication. Since its invention, PGP has been one of the most successful cryptographic systems used for e-mail security [7], because it has

come to offer a comprehensive system that covers a wide variety of security issues.

However, a major drawback of PGP can be seen in its unfair (incomplete) repudiation service, where there is a chance of false repudiation to occur [5]. False repudiation simply means the denial of such e-mailing service by one (or more) parts of e-mail communication [5]. This might be done in order to achieve an illegal benefit or deny a related commitment to such communication service.

Non-repudiation, on the other side, means providing evidence of such a delivery event, in order to protect against such occurring denial of it [8]. This can be done, for example, by providing the protected part by a receipt that serves as evidence of an action performed by the protected part of communication. In terms of e-mail communication, there are two basic types of e-mail non-repudiation services [9]:

(1) Non-Repudiation of Origin (NRO) provides the recipient of a message with evidence of sending the message by its sender, which protects against any attempt by the originator (sender) to falsely deny sending the message, in order to deny any related obligation to such performed action.

(2) Non-Repudiation of Receipt (NRR) provides the originator (sender) of a message with evidence of receiving the message by its receiver, which will protect against any attempt by the recipient to falsely deny receipt of the message, for the similar reason for which an originator would deny sending such an e-mail message.

NRO and NRR help to prevent the occurrence of denial-of-service attacks since every side of communication has evidence of a denied action, and therefore, assures the non-repudiation service of an e-mail communication to be complete, or fair. Although PGP does assure the NRO of e-mail message communication (as shown later) it does not assure its NRR, and its non-repudiation service is therefore, incomplete, or unfair.

Several protocols have been proposed to solve this problem, such as: Secure E-Mail Protocol (SEP) [5], Fair Non-Repudiation (FNP) [10], and its enhancement, a new Fair Non-Repudiation Protocol (FNRP) [9], Certified Mail Protocol (CMP) [11], and Certified Electronic Mail (CEM) [12].

In general, non-repudiation protocols are classified into two types whether is a Trusted Third Party (TTP) involved or not [10]. It is always desired to avoid the use of a TTP in the non-repudiation process [9], because this would decrease the number of trusted parties of communication, a thing which implies higher security within less cost and effort.

However, these protocols did not target the system PGP itself, but instead are proposed new protocols that would assure the fair non-repudiation of an e-mail communication service. In this paper, an Enhanced PGP (EPGP) system is introduced with a new feature of NRR, plus PGP's original feature of NRO, and therefore, assuring the new security service of *Mutual* Non-Repudiation (MNR) for an e-mail message communication. Before EPGP is introduced, a clear and illustrated explanation of PGP itself is shown in the following section.

2 Summary of PGP

Although PGP is a quite long and sophisticated process, it is possible to summarize the steps of its procedure by considering that sender A, who would like to send an e-mail message of any size, M, to a receiver B. The e-mail message is to be transmitted over an open network, where it is not possible to assure the security of the whole communication line of the network itself. Therefore, the following stages are performed:

Stage I: To assure the integrity of message M, user A's e-mail software computes a digital hash (or digest), M_1 , from message M, as follows:

$$A: M_1 = H(M) \quad (1)$$

Where $H(M)$ indicates the function of hashing the message M, using the Secure Hashing Algorithm-1 (SHA-1) to produce a digest for the message.

SHA-1 is a strong hashing algorithm that was developed in 1993. It takes an input message with a maximum length of $2^{64}-1$ bits, and processes it in 512-bit blocks, in order to produce a fixed-length output as a message digest of 160 bits [13].

Stage II: To assure the authenticity of message M, user A's e-mail software computes its digital signature from digest M_1 , and attaches it to the original message, M, as an authenticator. Therefore, M_2 is computed as follows:

$$A: M_2 = DS_{K_{RA}}[M_1] \parallel M \quad (2)$$

Where $[M]$ implies a message that would be encrypted or decrypted, and \parallel indicates concatenation (attachment) to the preceding message. $DS_{K_{RA}}[M_1]$ indicates the function of producing a digital signature for message M_1 , by A's private key, K_{RA} , using the Digital Signature Standard (DSS) scheme, as shown in figure (1). The RSA signature scheme may also be used as well.

DSS is a digital signature scheme that was developed in 1991. It computes a digital signature, or an authenticator, of a message in a quite similar way to SHA-1, considering the private key of the signer as part of the computation process of the digital signature to be attached to the message [13].

It is clear now that the NRO has also been achieved. Sender A cannot deny sending message M to receiver B, since its digital signature based on A's private key ($DS_{K_{RA}}[M_1]$) is attached to the message. This means that applying a digital signature on a message assures the receiver of: a) the authenticity of the sender, and b) the NRO.

Stage III: To assure the compactness of message M, user A's e-mail software compresses message M_2 to produce a reduced-size message, M_3 , as follows:

$$A: M_3 = Z(M_2) \quad (3)$$

Where Z indicates zipping the message by applying a strong ZIP compression algorithm, such as the Lempel-Ziv-1977 (LZ77) scheme [14]. Compression is applied after signing the message digitally, so that sender A can store a detached signature of message M, $DS_{K_{RA}}[M_1]$, as an archive.

Stage IV: To assure the secrecy, or confidentiality, of message M, user A's e-mail software encrypts message M_3 , to produce message M_4 , as follows:

$$A: M_4 = E_{K_s}[M_3] \parallel E_{K_{UB}}[K_s] \quad (4)$$

Where E_{K_s} indicates applying a symmetric black-box Data Encryption Standard (DES) in encryption algorithm in the Cipher Block Chaining (CBC) mode, such as CAST-128, using a 128-bit secret

session key, K_S , which A generates using a PGP pseudo-random key-generator (PRKG).

Also, $E_{K_{UB}}$ indicates RSA public-key encryption to the secret key K_S , using user B's public key, K_{UB} . The encrypted session key, $E_{K_{UB}}[K_S]$, is then attached to the encrypted e-mail message, $E_{K_S}[M_3]$, so that receiver B will be able to retrieve the secret key needed to decrypt the e-mail message later, as shown in figure (1).

Stage V: Finally, to assure the compatibility of message M to different e-mail systems, user A's e-mail software computes the final e-mail message to be transmitted, M_5 , as follows:

$$A \rightarrow B: M_5 = R_{64}(M_4) \quad (5)$$

Where $A \rightarrow B$ implies sending a message from A to B. R_{64} indicates applying Radix-64 conversion on message M_4 , which maps every keyboard-character of 6 bits into 8 bits of a general ASCII code, which is required by most e-mail systems. The entire procedure of the "transmission phase" of PGP is illustrated in figure (1).

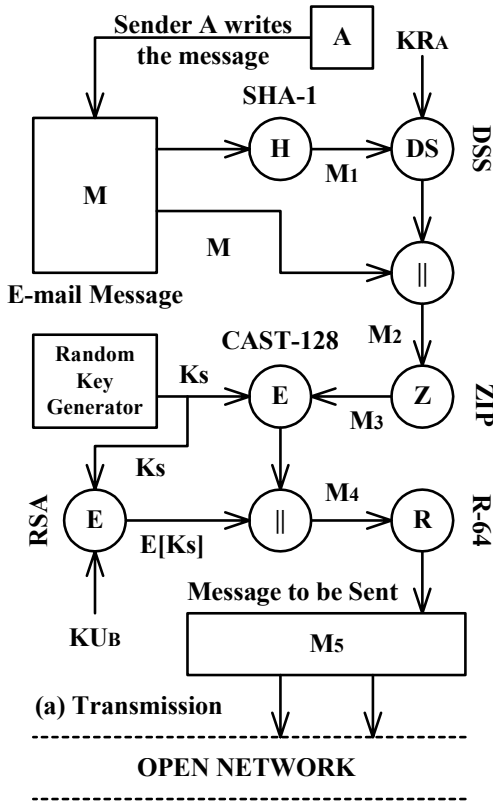


Figure (1): The PGP Transmission Phase

Stage VI: Now, the ciphertext message M_5 is transmitted to receiver B over the network. After

the transmitted message M_5 has reached its destination, receiver B performs the reverse PGP process, as illustrated in figure (2), to retain back the original e-mail message M. User B's e-mail software first applies the inverse Radix-64 conversion to retain back M_4 , as follows:

$$B: M_4 = R_{64}^{-1}(M_5) = E_{K_S}[M_3] \parallel E_{K_{UB}}[K_S] \quad (6)$$

Then, it retrieves the secret session key, K_S , by an RSA decryption process, using its private key, K_{RB} , as follows:

$$B: K_S = D_{K_{RB}}[E_{K_{UB}}[K_S]] \quad (7)$$

Then, it uses the obtained secret key, K_S , to decrypt message M_3 , as follows:

$$B: M_3 = D_{K_S}[E_{K_S}[M_4]] \quad (8)$$

Then, it applies the inverse ZIP compression process to retain back message M_2 , as follows:

$$B: M_2 = Z^{-1}(M_3) = DS_{K_{RA}}[M_1] \parallel M^* \quad (9)$$

At this point, the NRO of message M is achieved. Now, receiver B has already gotten the original e-mail message M, as M^* , but still not sure of the integrity and authenticity of the received message. Therefore, B verifies the digital signature, $DS_{K_{RA}}[M_1]$, using user A's public key, K_{UA} , as follows:

$$B: M_1 = DS^{-1}_{K_{UA}}[DS_{K_{RA}}[M_1]] \quad (10)$$

Now, the final required step of check is to retrieve back the original e-mail message M, from M_1 . Since it is not possible to retain back an original message, M, from its digest, $H(M)$, user B may check the integrity of the received message, M^* , by re-hashing M^* , and comparing it to the digest of the original message, $H(M)$, as illustrated in figure (2), and shown below:

$$B: \text{IF } H(M^*) = H(M) \text{ THEN } M^* = M \quad (11)$$

Equation (11) implies that if the two hashes are equal, then the two messages must be identical, since no two different messages can lead to the same digest, when hashed by a strong hashing algorithm, like SHA-1 [3].

In this case, user B accepts the received message, M^* , as the original e-mail message, M, sent by user

A over the network, and the entire PGP process is complete. Otherwise, user B would reject the received message, M^* . The entire procedure of the "reception phase" of PGP is illustrated in figure (2).

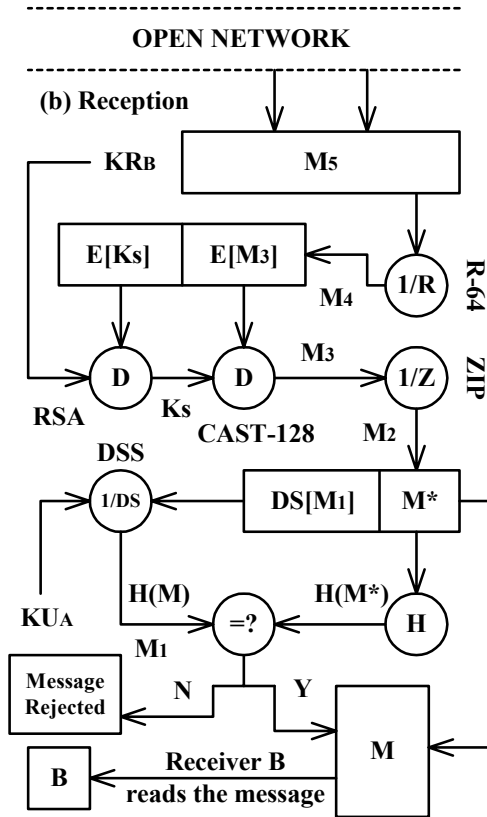


Figure (2): The PGP Reception Phase

3 Motivation for EPGP

The problem of the current PGP system, as mentioned before, is that it does not provide its users with complete, or fair, non-repudiation service for an e-mail message communication over an open network. Therefore, the motivation of EPGP is to address this problem, basically.

It has become clear from the previous example that sender A cannot deny sending message M to receiver B, since his digital signature, $DS_{KRA}[M_1]$, has been already attached to the transmitted message. This means that PGP does assure the NRO of an e-mail message communication session over a network.

However, it is also clear that receiver B can easily deny receiving an e-mail message M, from sender A, since A has no evidence on such event. This means that PGP does not assure the NRR of an e-

mail message communication session over a network. The idea of PGP's incomplete (unfair) non-repudiation is illustrated in figure (3).

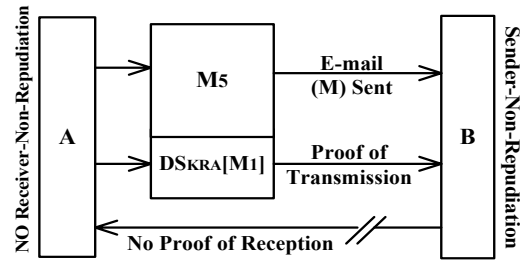


Figure (3): Unfair Non-Repudiation of PGP

In many important e-mail message communication sessions, a receiver's denial of the reception of a sent e-mail message by its sender, may lead to a variety of problems. For example, in business communications, a receiver's denial of reception would make the receiver able to deny any corresponding commitment of a business contract. At the court, the judge is unable to prove the reception process without an evidence of receipt. Although PGP's feature of NRO makes it a very good system, EPGP's feature of NRR, and therefore, MNR, would make it a better and more secure one.

4 Enhanced PGP (EPGP)

It is known that an e-mail communication process is a connectionless-oriented type of communication, that is, it is not necessary for both sides of communication to be in direct contact with each other simultaneously during the transmission and reception phases [2].

Instead, an e-mail message M_5 that sender A sends is uploaded to a 24-hour-available trusted e-mail software server D. Then whenever receiver B opens its e-mail inbox, message M_5 is downloaded from e-mail server D to B's machine, where B's e-mail software performs the reverse PGP process to retain back the original text of e-mail message M.

It is not necessary for B to be online when A sends the message, neither is it necessary for A to be online when B receives the message, since the e-mail server D is online all the time. The procedure of EPGP relies heavily on an e-mail server, D, to accomplish the enhancement feature of MNR, as shown later.

Server D is *not* a Trusted Third Party (TTP) from outside the communication link, but it is an embedded part in the whole process that takes on

the role of message delivery. The entire EPGP process consists of three main phases, described as follows:

Phase I: This is the "transmission phase", where quite similar steps of PGP are to be applied here. User A's e-mail software computes message M_1 , by hashing message M , using the SHA-1 hashing algorithm as follows:

$$A: M_1 = H(M) \quad (1)$$

Then, user A's e-mail software computes M_2 as a digital signature of message M_1 , using the DSS digital signature scheme. The attached digital signature of sender A, $DS_{KRA}[M_1]$, on to the message will assure the feature of NRO, which is already achieved by PGP as well, as follows:

$$A: M_2 = DS_{KRA}[M_1] \parallel M \quad (2)$$

Then, user A's e-mail software compresses message M_2 as message M_3 , using the LZ77 ZIP algorithm, as follows:

$$A: M_3 = Z(M_2) \quad (3)$$

Then, user A's e-mail software computes M_4 , by encrypting message M_3 , by the secret key K_S , using a DES-CBC symmetric encryption algorithm. Then, the secret key, K_S of A is encrypted by RSA public-key encryption, using the public key of server D, K_{UD} , *not* the public key of receiver B as shown in (4). B will *not* be able to retrieve the secret key K_S , and thus, not become able to decrypt the e-mail message. The computation of message M_4 is shown as follows:

$$A: M_4 = E_{K_S}[M_3] \parallel E_{K_{UD}}[K_S] \quad (12)$$

Finally, user A's e-mail software computes M_5 by applying Radix-64 conversion to ASCII on message M_4 , and sends the final message to e-mail server D, as follows:

$$A \rightarrow D: M_5 = R_{64}(M_4) \quad (5)$$

Now, the message has been sent to receiver B via server D over the open network. It is clear now that receiver B till now is still not able to decrypt the message since it has not gotten yet the secret key K_S , nor server D's private key, K_{RD} . The enhancement of NNR is applied here as shown in the next phase of the EPGP process. The entire "transmission phase" of EPGP is illustrated in figure (4).

Phase II: This is called the "NNR phase", which is the major enhancement of EPGP. Once receiver B, opens its e-mail inbox, downloads message M_5 from server D, and attempts to open message M , user B's e-mail software will establish a communication session with server D to get the secret key, K_S , to decrypt the message. First of all, server D forwards message M_5 to B, as follows:

$$D \rightarrow B: M_5 \quad (13)$$

Server D will not grant receiver B the secret key, K_S , unless receiver B handles its digital signature on the unopened message, M_5 , to server D first. This will serve as evidence of message reception, and therefore, the MNR of the whole process. Receiver B submits server D its digital signature on the received message, M_5 , encrypted by RSA, using user A's public key, K_{UA} , as follows:

$$B \rightarrow D: M_6 = E_{K_{UA}}[DS_{KRB}[M_5]] \quad (14)$$

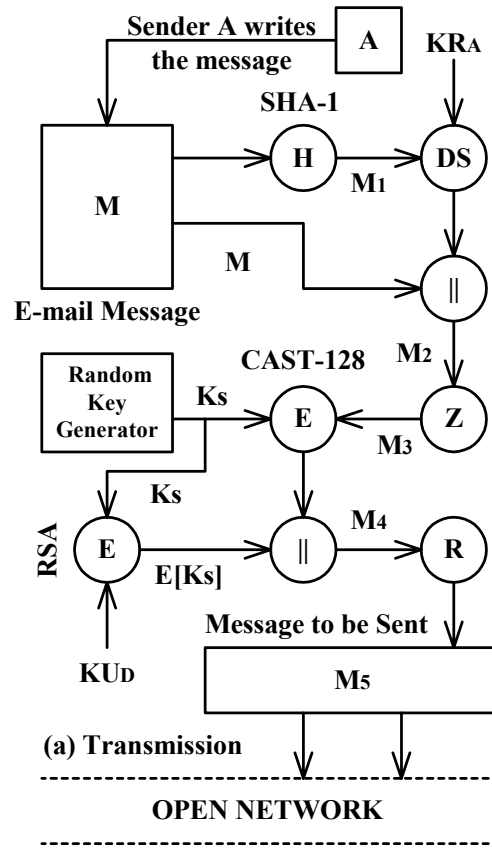


Figure (4): The EPGP Transmission Phase
Then, server D may send the secret key, K_S , to receiver B, encrypted by RSA, using B's public key, K_{UB} , after D obtains the key itself by encrypting it using its own private key, K_{RD} , as follows:

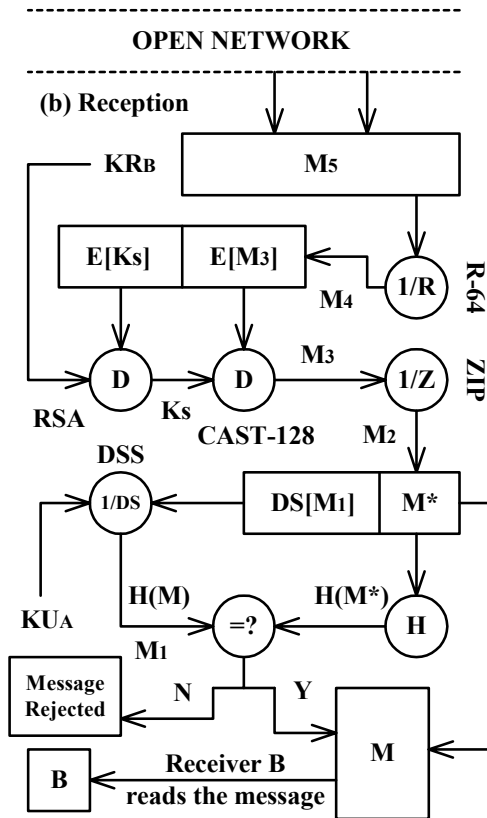


Figure (6): The EPGP Reception Phase

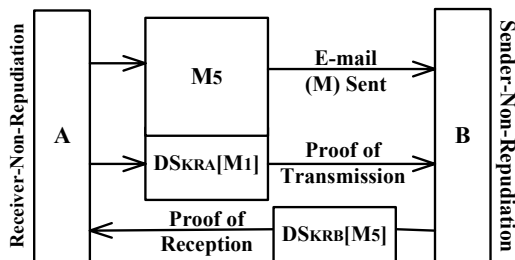


Figure (7): Fair MNR of EPGP

6 Conclusion

A new protocol and architectural design is the result of this initial work on enhancing PGP to prevent non repudiation. It appears that the design of an Enhanced Pretty Good Privacy (EPGP) system is relatively complete, and the objective of Mutual Non-Repudiation (MNR) is accomplished. This means that the level of security of e-mail message communication can be improved by the MNR major enhancement.

Future work may also consider a simulation of EPGP, plus any other improvement of the EPGP model to solve any other drawbacks of PGP, in

order to enhance the level of security and efficiency of EPGP for the purpose of secure e-mail message communication.

7 References

- [1] Drummond R, Cox N, "Lan Times E-mail Resource Guide", Osborne McGraw-Hill, 1994.
- [2] Tanenbaum A, Computer Networks, Prentice Hall, UpperSaddle River, NJ, c1996.
- [3] Stallings W, Network Security Essentials, Prentice Hall, Upper Saddle River, NJ, c2003.
- [4] Al-Hammadi B and Shamsavari M, "Certified Exchange of Electronic Mail (CEEM)", Southeastcon '99, IEEE Proceedings, 1990, pp. 40-43.
- [5] Bai L, Achuthanadam R, and Kam M, "Access Revocation and Prevention of False Repudiation in Secure Email Exchanges", Fifth International Symposium on Autonomous Decentralized Systems, Dallas, Texas, March 26-28 2001, pp. 419-425.
- [6] Linn J, "Privacy Enhancement for Electronic Mail", <http://www.cis.ohio-stat.edu/rfc/rfc1321.txt>, April 1992.
- [7] Garfield S, PGP: Pretty Good Privacy, O'Reilly Associates, Inc, c1994.
- [8] Zhou J and Gollmann D, "Evidence and Non-Repudiation", Journal of Network and Computer Applications, London: Academic Press, 1997.
- [9] Zhou J and Gollmann D, "An Efficient Non-Repudiation Protocol", Computer Security Foundations Workshop, 1997 Proceedings, 10th, 10-12 Jun 1997, pp. 126-137.
- [10] Meng B, Wang S, and Xiong Q, "A Fair Non-Repudiation Protocol", Proceedings of IEEE symposium on security and privacy, 1996, pp. 51-61.
- [11] Deng R, Gong L, Lazar A, and Wang W, "Practical protocols for certified electronic mail", Journal of Network Security Manager, 4(3), 1996, pp. 279-297.
- [12] Baherman A and Tygar J D, "certified Electronic Mail (CEM)", Proceedings of the internet security symposium on network and distributed system security, San Diego, California, February 1994, pp. 3-19.
- [13] Stallings W, Cryptography and Network Security, Prentice Hall, UpperSaddle River, NJ, c2003.
- [14] Ziv J and Lempel A, "A Universal Algorithm for Sequential Data Compression", IEEE Transactions on Information Theory, Vol. 23, No. 3, pp. 337-343.