Journal of Zhejiang University-SCIENCE C (Computers & Electronics)
ISSN 1869-1951 (Print); ISSN 1869-196X (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

A new protocol of wide use for e-mail with perfect forward secrecy*

Tzung-her CHEN†, Yan-ting WU

(*Department of Computer Science and Information Engineering, National Chiayi University, Taiwan 60004, Chiayi City*)

†E-mail: thchen@mail.ncyu.edu.tw

Received Feb. 27, 2009; Revision accepted May 1, 2009; Crosschecked Sept. 27, 2009

Abstract: Recently, Sun *et al*. (2005) highlighted the essential property of perfect forward secrecy (PFS) for e-mail protocols when a higher security level is desirable. Furthermore, Sun *et al.* (2005)'s protocols take only a single e-mail server into account. Actually, it is much more common that the sender and the recipient register at different e-mail servers. Compared to existing protocols, the protocol proposed in this paper takes into account the scenario that the sender and the recipient register at different servers. The proposed protocol is skillfully designed to achieve PFS and end-to-end security as well as to satisfy the requirements of confidentiality, origin, integrity and easy key management. The comparison in terms of functionality and computational efficiency demonstrates the superiority of the present scheme.

Key words: E-mail protocol, Perfect forward secrecy (PFS), Confidentiality, End-to-end security
doi:10.1631/jzus.A0910126 **Document code:** A **CLC number:** TP309

1 Introduction

Nowadays, people are used to communicating with each other by e-mail. Since e-mail is always transmitted over open networks, it is not secure against eavesdropping, modification, and so on. In e-mail protocols, keeping the confidentiality and integrity of transmitted e-mail is so crucial that e-mail protocols such as pretty good privacy (PGP) (Zimmermann, 1995) and security/multipurpose Internet mail extensions (S/MIME) (Thompson, 1995) have been proposed.

PGP and S/MIME e-mail protocols use the hybrid system by combining the public-key cryptosystem (Diffie and Hellman, 1976) and the symmetric cryptosystem. A sender generates the digest of e-mail content M, say $H(M)$. Next, while the digest is encrypted the sender's private key PR_S is adopted, which is then concatenated with M, say $Z=F_{PR_S}(H(M))||M$. The sender uses a session key k_S to encrypt Z, say $E_{k_S}[Z]$. Finally, the session key k_S and $E_{k_S}[Z]$ are encrypted with the public key of the recipient before the mail is sent to the recipient. Using PGP and S/MIME, the confidentiality and integrity are easy to achieve.

Certified e-mail protocols (Schneier and Riordan, 1998; Park *et al*., 2003; Puigserver *et al*., 2005) have recently been proposed to achieve fair exchange. That is, when a sender sends an e-mail to a recipient, the sender can confirm whether the recipient successfully receives the e-mail or not. On the other hand, the recipient can also confirm whether the e-mail has been sent from the sender.

Perfect forward secrecy (PFS) is important for a security protocol. It means that when the long-term keys of either the sender or the recipient are compromised, the previous short-term keys are still kept secret. Unfortunately, the above-mentioned schemes (Thompson, 1995; Zimmermann, 1995; Schneier and Riordan, 1998; Park *et al*., 2003; Puigserver *et al*., 2005) do not have this property.

Sun *et al*. (2005) proposed two secure e-mail protocols that could achieve PFS. However, Dent (2005) pointed out that Sun *et al*. (2005)'s second protocol did not have the property of PFS. Phan (2008)

* Project supported by the National Science Council (No. NSC 98-2221-E-415-006-)

further showed that Sun *et al*. (2005)'s first protocol could not resist replay attacks or unknown key-share attacks. Thus, some e-mail protocols (Kim *et al*., 2006; Lin *et al*., 2006) have been presented to improve the security of Sun *et al*. (2005)'s.

It is worthwhile to note that the above-mentioned e-mail protocols take only a single e-mail server into account. Actually, it is much more common that the sender and the recipient register at different e-mail servers.

The guidelines of designing a secure e-mail protocol are highlighted as follows:

1. End-to-end confidentiality: An e-mail should be transmitted in the form of ciphertext so that it cannot be decrypted by anyone else (including e-mail servers) but the intended recipient.

2. Origin and integrity: The recipient can check the origin of the received e-mail and the integrity of the content of the e-mail. This property can be guaranteed through adopting message authentication code (MAC) or digital signature.

3. PFS: This property is in particular suitable for applications with the essential need of high security. It can be achieved with the characteristic of Diffie-Hellman key agreement scheme (RSA Laboratories, 2000)

4. Public key infrastructure (PKI) complexity: It is well known that it is expensive to maintain a PKI environment. Security protocols that intuitively adopt public key cryptosystems often suffer from this.

In this paper, a new e-mail protocol, in which the sender and the recipient register at two different e-mail servers, is proposed to meet all the properties mentioned above. The comparison in terms of functionality and computational efficiency between the protocols in related works and the proposed protocol demonstrates that the present scheme does work well.

2 Proposed method

The proposed e-mail protocol consists of three phases: registration, sending, and receiving. Some notations shown in Table 1 are depicted to help understand the proposed protocol.

2.1 Registration phase

Either the sender or the recipient has to register at an individual e-mail server at the beginning. For example, when a participant A (resp. B) registers at e-mail server S_A (resp. S_B), it implies that A shares password Q_1 with S_A. A submits ID_A and (g^{aQ_1} mod n) to S_A, where n is a big prime number, g is a generator with order $n-1$ over GF(n), and a is a random number. S_A computes the registration information (g^a mod n) with Q_1^{-1} and stores (g^a mod n). Likewise, the participant B shares Q_2 with e-mail server S_B. S_B stores (g^b mod n) for B. For simplicity, 'mod n' is omitted hereafter.

Table 1 Notations used in the proposed scheme

Notation	Description
ID_A, ID_B	A or B's identity, including the field of the identity of an e-mail server at which A or B has registered
S_A, S_B	The two e-mail servers at which A and B have registered
$[\cdot]k$	The symmetric encryption or decryption operation with the key k
Q_1, Q_2	The integers computed from the passwords are shared between the sender and the e-mail server or e-mail and the recipient. And the size is large enough to resist dictionary attacks
n	A large prime number
g	A generator with order $n-1$ over GF(n)
$\mathrm{MAC}_k()$	A one-way hash function with the key k
K	The secret key shared in advance between two e-mail servers S_A and S_B
$X \rightarrow Y$: M	X sends the message M to Y

2.2 Sending phase

When sender A intends to send an e-mail to recipient B, the operation goes as follows:

Step 1: $A \rightarrow S_A$: Request.

If A wants to deliver an e-mail to B, he should send the request to S_A firstly.

Step 2: $S_A \rightarrow S_B$: Request.

S_A forwards the request to S_B to ask for the registration information of B.

Step 3: $S_B \rightarrow S_A$: ID_B, g^b, $\mathrm{MAC}_K(\mathrm{ID}_B, g^b)$.

S_B finds g^b of B. Then S_B computes the MAC value of ID_B, g^b with K, and sends ID_B, g^b, $\mathrm{MAC}_K(\mathrm{ID}_B, g^b)$ to S_A.

Step 4: $S_A \rightarrow A$: ID_B, g^b, $\mathrm{MAC}_{Q_1}(\mathrm{ID}_B, g^b)$.

In order to check the validation of the received message, S_A computes $\mathrm{MAC}_K(\mathrm{ID}_B, g^b)$ and checks if the computed MAC value is equal to the received MAC value. If it holds, S_A computes the MAC value

of ID_B, g^b with Q_1 and sends ID_B, g^b, $MAC_{Q_1}(ID_B, g^b)$ to A.

Step 5: $A \rightarrow S_A$: ID_A, ID_B, $[M]_{g^{xb}}$, g^x, $MAC_{Q_1}(ID_A, ID_B, [M]_{g^{xb}}, g^x)$.

Upon receiving the message, A computes $MAC_{Q_1}(ID_B, g^b)$ and checks if the computed MAC value is equal to the received MAC value. If it holds, A computes g^x with a random number x and g^{xb} by computing $(g^b)^x$. A encrypts M with g^{xb}, where M is the content of the e-mail. A computes the MAC value of ID_A, ID_B, $[M]_{g^{xb}}$, g^x with Q_1 and sends ID_A, ID_B, $[M]_{g^{xb}}$, g^x, $MAC_{Q_1}(ID_A, ID_B, [M]_{g^{xb}}, g^x)$ to S_A.

Step 6: $S_A \rightarrow S_B$: ID_A, ID_B, $[M]_{g^{xb}}$, g^x, $MAC_K(ID_A, ID_B, [M]_{g^{xb}}, g^x)$.

S_A checks the validation of the received message. It computes $MAC_{Q_1}(ID_A, ID_B, [M]_{g^{xb}}, g^x)$ and checks if the computed value is equal to the received value. If it holds, S_A computes the MAC value of ID_A, ID_B, $[M]_{g^{xb}}$, g^x with K and sends ID_A, ID_B, $[M]_{g^{xb}}$, g^x, $MAC_K(ID_A, ID_B, [M]_{g^{xb}}, g^x)$ to S_B. After receiving the message, S_B stores the e-mail message for B.

2.3 Receiving phase

Step 7: $B \rightarrow S_B$: ID_B, $g^{b'}$, $MAC_{Q_2}(ID_B, g^{b'}, g^b)$.

When B is on-line and intends to check e-mails, he will compute $g^{b'}$ with a new random number b' and $MAC_{Q_2}(ID_B, g^{b'}, g^b)$. Then B sends ID_B, $g^{b'Q_2}$, $MAC_{Q_2}(ID_B, g^{b'}, g^b)$ to S_B.

Step 8: $S_B \rightarrow B$: ID_A, ID_B, $[M]_{g^{xb}}$, g^x, $MAC_{Q_2}(ID_A, ID_B, [M]_{g^{xb}}, g^{b'}, g^b)$.

Upon S_B receiving the message, S_B verifies $MAC_{Q_2}(ID_B, g^{b'}, g^b)$. If the verification fails, S_B will reject the request from B; otherwise, S_B updates g^b with $g^{b'}$. Lastly, S_B computes the MAC value of ID_A, ID_B, $[M]_{g^{xb}}$, g^x, $g^{b'}$ with Q_2 and sends ID_A, ID_B, $[M]_{g^{xb}}$, g^x, $g^{b'}$, $MAC_{Q_2}(ID_A, ID_B, [M]_{g^{xb}}, g^x, g^{b'})$ to B.

When B receives the message from S_B, he computes $MAC_{Q_2}(ID_A, ID_B, [M]_{g^{xb}}, g^x, g^{b'})$. B checks if the computed MAC value is equal to the received MAC value. If it holds, he computes g^{xb} by computing $(g^x)^b$ to decrypt $[M]_{g^{xb}}$.

3 Security analysis

This section provides the security analysis of the proposed e-mail protocol. Before explaining the security, we make some assumptions to help prove the security.

Assumption 1 (Discrete logarithm problem, DLP) Given a prime n, a generator g of Z_n^*, and an element $b \in Z_n^*$, find an integer a, $1 \le a \le n-2$, such that $g^a \equiv b \bmod n$. The DLP assumption is that there is no feasible way to solve the DLP problem, computing a by giving b, for all probabilistic polynomial time.

Assumption 2 (Computational Diffie-Hellman problem, CDH) Given g^a and g^b, there is no feasible way to compute g^{ab} in probabilistic polynomial time.

Assumption 3 (One-way hash assumption) A one-way hash function is defined as that the hash function can take input of any size and output the fixed-size result. For a secure one-way hash function $y=h(x)$, given x to compute y is easy but given y to compute x is hard.

Next, we analyze three security properties: confidentiality, integrity, and PFS. Confidentiality is the protection of the transmitted e-mail: only the authorized recipient can obtain the content of the mail. The general method is encrypting the e-mail. Secondly, integrity ensures the e-mail not modified on the way of transmitting. Finally, PFS is a security-sensitive property. If the long-term secret keys or passwords are compromised, the previous session keys will be still kept secret and, thus, the confidentiality of the past e-mail will be guaranteed. Besides, replay attacks are one of the most common attacks to a security protocol. In an e-mail protocol, they aim at cheating the e-mail recipient.

Proposition 1 (Confidentiality with end-to-end security) This protocol provides confidentiality of the transmitted e-mail by keeping the encryption key secret.

Proof Cipher-e-mail is delivered from A, S_A, S_B to the recipient B. Although the attacker can intercept g^x and g^b, he has no feasible way to compute g^{xb} based on Assumption 2. The cipher-e-mail is not decrypted until the recipient does it. On the way of delivery, neither outsiders nor e-mail servers can open it. Thus, confidentiality with end-to-end security is achieved.

Proposition 2 (Integrity) This protocol provides integrity of the transmitted e-mail.

Proof In this protocol, the integrity of the transmitted e-mail is enabled by introducing MAC functions. The integrity of e-mail is guaranteed by means of protecting the integrity of $[M]_{g^{xb}}$.

The integrity of $[M]_{g^{xb}}$ between sender A and e-mail server S_A is verified by S_A (Step 5) with Q_1. Secondly, the integrity of $[M]_{g^{xb}}$ between e-mail server S_A and recipient S_B is verified by S_B (Step 6) with the secret key K. Thirdly, the integrity of $[M]_{g^{xb}}$ between e-mail server S_B and recipient B is verified by B (Step 8). Hence, this protocol can provide the integrity of the transmitted e-mail.

Proposition 3 (Perfect forward secrecy, PFS) This proposed protocol provides the property of PFS.

Proof If the passwords Q_1, Q_2 and the secret key K are disclosed to an attacker for some reason, and the attacker can compute g^x and g^b, the attacker has no efficient way to compute the session key g^{xb} based on Assumption 2. Also he cannot solve x (resp. b) to compute $(g^b)^x$ (resp. $(g^x)^b$) based on Assumption 1. It implies this protocol provides the property of PFS.

Proposition 4 (Resistance to replay attacks) This protocol can resist replay attacks.

Proof Two nonce numbers a and b are involved in all MAC functions in Steps 3–8 to guarantee the freshness of transmitted messages. Each participant can easily measure the freshness of message traveling. Accordingly, the freshness of the received message can be guaranteed by checking the validation of MAC values.

4 Discussion

The following discussions are given to highlight the advantages of the present protocol.

1. Generalization. In order to meet the scenario of common e-mail protocol, i.e., involving the two server e-mail servers for two participants, the proposed protocol enables two participants registering at different servers to transfer an e-mail.

2. Computation complexity. The present e-mail protocol involves cost-efficient hash operations instead of cost-expensive asymmetric cryptographic operations. Hash operations are about 100-10000 times faster than asymmetric cryptographic operations (Rivest and Shamir, 2001), and symmetric operations are about 1000 times faster than asymmetric cryptographic (RSA Laboratories, 2000). Therefore, the computation complexity can be decreased efficiently in the proposed scheme.

3. Key management. Complexity of PKI arises from public-key dictionary maintenance including public-key certificate generation and public-key revoking in the system. However, such certificate-based public-key cryptosystems have mainly the following potential problems. Firstly, the cost for verifying the certificate is necessary. Secondly, the heavy burden of maintaining PKI is involved. The proposed scheme removes these shortcomings of the related protocols in (Sun *et al*., 2005; Kim *et al*., 2006; Lin *et al*., 2006).

Table 2 summarizes the comparisons of security and functionality between the related e-mail protocols and ours. Table 3 shows the comparison of performance between the related e-mail protocols and ours.

Table 2 Functionality comparison between the related e-mail protocols and ours

Protocol	Reference	Confidentiality	Integrity	PFS	RTRA	PKI	NR
PGP	Zimmermann, 1995	Y	Y	N	Y	Y	Y
S/MIME	Thompson, 1995	Y	Y	N	Y	Y	Y
SHH	Sun *et al*., 2005	Y	N	Y	N	Y	Y
		Y	N	N	N	Y	Y
LLW	Lin *et al*., 2006	Y	N	Y	Y	Y	Y
KKL	Kim *et al*., 2006	Y	N	Y	Y	Y	Y
		Y	N	Y	Y	Y	Y
SR	Schneier and Riordan, 1998	N	N	Y	Y	Y	N
PRCS	Park *et al*., 2003	Y	N	N	N	Y	Y
PGR	Puigserver *et al*., 2005	N	N	N	N	Y	Y
Ours		Y	Y	Y	Y	N	Y

Y: the e-mail protocol can provide the property; N: the e-mail protocol can not provide the property; NR: non-repudiation; PFS: perfect forward secrecy; PGP: pretty good privacy; PKI: public key infrastructure; RTRA: resistance to replay attacks; S/MIME: security/multipurpose Internet mail extensions

Table 3 Computation comparison between the related e-mail protocols and ours

Computation	*A*/*S*/*B*					*A*/S_A/S_B/*B*
	SHH		LLW	KKL		Ours
Public-key en(de)cryption	2/2/2	2/2/2	2/2/2	2/1/2	1/0/0	0/0/0/0
Modular exponential	2/1/2	1/1/1	1/1/1	2/3/3	2/3/3	3/0/0/2
Session-key en(de)cryption	1/0/1	1/0/1	1/0/1	1/1/2	1/1/2	1/0/0/1
Hash function	1/0/1	1/0/1	1/0/1	0/0/0	1/0/1	2/4/4/2

KKL (Kim *et al.*, 2006); LWW (Lin *et al*, 2006); SHH (Sun *et al*, 2005)

5 Conclusion

In this paper, a more generalized protocol providing perfect forward secrecy and end-to-end security is presented. This protocol can guarantee not only the high security of end-to-end confidentiality, origin, integrity and perfect forward secrecy, but also the performance in terms of generality, key management, and computation efficiency.

References

Dent, A.W., 2005. Flaws in an e-mail protocol of Sun, Hsieh and Hwang. *IEEE Commun. Lett.*, **9**(8):718-719. [doi:10.1109/LCOMM.2005.1496593]

Diffie, W., Hellman, M.E., 1976. New directions in cryptography. *IEEE Trans. Inf. Theory*, **22**(6):644-654. [doi:10.1109/TIT.1976.1055638]

Kim, B.H., Koo, J.H., Lee, D.H., 2006. Robust e-mail protocols with perfect forward secrecy. *IEEE Commun. Lett.*, **10**(6):510-512. [doi:10.1109/LCOMM.2006.1638632]

Lin, I.C., Lin, Y.B., Wang, C.M., 2006. An Improvement on Secure E-Mail Protocols Providing Perfect Forward Secrecy. Proc. 9th Joint Conf. on Information Sciences, p.697-700. [doi:10.2991/jcis.2006.102]

Park, J.M., Ray, I., Chong, E.K.P., Siegel, H.J., 2003. A Certified E-Mail Protocol Suitable for Mobile Environments. Proc. IEEE Global Telecommunications Conf., p.1394-1398. [doi:10.1109/GLOCOM.2003.1258467]

Phan, R.C.W., 2008. Cryptanalysis of e-mail protocols providing perfect forward secrecy. *Comput. Stand. Inter.*, **30**(3):101-105. [doi:10.1016/j.csi.2007.08.007]

Puigserver, M.M., Gomila, J.L.F., Rotger, L.H., 2005. Certified E-Mail Protocol with Verifiable Third Party. Proc. IEEE Int. Conf. on e-Technology, e-Commerce and e-Service, p.548-551. [doi:10.1109/EEE.2005.46]

Rivest, R.L., Shamir, A., 2001. PayWord and MicroMint: Two Simple Micropayment Schemes. MIT Laboratory for Computer Science, Cambridge, MA, p.1-18. [doi:10.1007/3-540-62494-5_6]

RSA Laboratories, 2000. RSA Laboratories' Frequently Asked Questions about Today's Cryptography, V4.1. Bedford, USA.

Schneier, B., Riordan, J., 1998. A Certified E-Mail Protocol. Proc. 14th Annual Computer Security Applications Conf., p.347-352.

Sun, H.M., Hsieh, B.T., Hwang, H.J., 2005. Secure e-mail protocols providing perfect forward secrecy. *IEEE Commun. Lett.*, **9**(1):58-60. [doi:10.1109/LCOMM.2005.01004]

Thompson, J., 1996. S/MIME Message Specification: PKCS Security Services for MIME. RSA Data Security Inc. Available from http://www.rsa.com/ [Accessed on Feb. 22, 2009].

Zimmermann, P.R., 1995. The Official PGP User's Guide. MIT Press, Cambridge, Massachusetts, USA.