Flaws in an E-Mail Protocol of Sun, Hsieh and Hwang

Alexander W. Dent

***Abstract*— Recently, Sun, Hseih and Hwang [1] proposed two methods of retrieving e-mail from a central e-mail server and claimed that these algorithms had perfect forward secrecy. We present a critique of one of their algorithms. In particular, we break the forward secrecy of the second proposed protocol.**

***Index Terms*— E-mail, network security, encryption, perfect forward secrecy.**

I. Introduction

Recently, a paper by Sun, Hseih and Hwang [1] proposed two protocols for retrieving e-mail from a central e-mail server in such a way that the e-mails had forward secrecy. Both of these protocols extend the standard hybrid encryption paradigm, where a number of long term asymmetric keys are used to generate and encrypt short term symmetric keys, and these short term symmetric keys are used to encrypt messages. It appears that their protocols were designed to have two security properties: the messages should be encrypted when they pass from the sender to the e-mail server, and from the e-mail server to the receiver; and that these encryptions should have perfect forward secrecy. They define perfect forward secrecy in the following way:

> A protocol providing perfect forward security means that even if one entity's long term secret key is compromised, it will never reveal any old short term keys used before.

The authors proposed two schemes that they claim achieve these goals. The first is a scheme based on the well-known result that a completely ephemeral version of the Diffie-Hellman key exchange protocol achieves perfect forward secrecy. The second is a scheme based on the concept of the Certificate of Encrypted Message Being a Signature (CEMBS) [2].

In proposing their scheme, Sun *et al.* use the following notational conventions. A sender B wishes to send e-mail to a receiver A via an e-mail server S. Each of these entities, for example A, may have a public encryption key PK_A and a private decryption key SK_A. Asymmetric encryption of a message M under a public key PK_A is denoted as $Enc_{PK_A}(M)$; the corresponding decryption operation on a ciphertext C under a private key SK_A is given by $Dec_{SK_A}(C)$. The symmetric encryption of a message M under a secret key K is given by $E_K(M)$; whilst the decryption operation of a ciphertext C under the same key is given by $D_K(C)$. Further notation is given in Table I.

In this letter, we review the second protocol proposed by Sun *et al.*, and show that it is not clearly defined, that the security analysis is flawed, and that it is does not, as claimed, provide forward secrecy.

Alexander Dent is with the Information Security Group at Royal Holloway, University of London (email: a.dent@rhul.ac.uk)

TABLE I
NOTATION USED IN PROTOCOLS

A	The receiver.
b	The private signing key of entity B. We will assume that this is an integer between 1 and p.
B	The sender.
$Cert$	A certificate of the ciphertext.
$D_K(C)$	The symmetric decryption of the ciphertext C under the key K.
$Dec_{SK}(C)$	The asymmetric decryption of ciphertext C under the private key SK
$E_K(M)$	The symmetric encryption of message M under the key K.
$Enc_{PK}(M)$	The asymmetric encryption of message M under the public key PK.
g	An element of $\mathbb{Z}_p^*$ that generates a large subgroup.
h	A hash function.
ID_X	A bit string that uniquely identifies entity X.
K	A secret symmetric key.
p	A large prime number.
PK_X	The public encryption key belonging to entity X. We will always assume that this is an element of the form $g^x \bmod p$ where x is the private key decryption key.
pwd	A password shared between A and S
S	The e-mail server.
$Sig_X(M)$	A signature generated on the message M using the private signing key of entity X.
SK_X	The private decryption key belonging to entity X.

II. The CEMBS Based E-Mail Protocol

We describe the second of Sun *et al.* [1] protocols in Figure 1[1]. This protocol is based on the concept of the Certificate of Encrypted Message Being a Signature (CEMBS) proposed by Bao, Deng and Mao [2].

Firstly, we note that the authors claim that the pair (r, s) acts as a Schnorr signature on the bit string ID_A. This is not the case. For a Schnorr signature, the value s would have to be computed as:

$$s = x + bh(ID_A||r) \bmod (p-1)$$

where b is some randomly chosen and secret integer between 1 and the order of g in $\mathbb{Z}_p^*$. Furthermore, the Schnorr signature would be the pair $(h(ID_A||r), s)$ and not the pair (r, s). We assume that this is what the authors meant and that the private signing key b is not used for any other purpose.

[1] It should be noted that we have altered slightly the composition of message (5) for clarity. We have explicitly shown the use of the password within the protocol. In the original paper, the encryption of y under the password shared by the server and A is denoted $Enc_{PK_A}(y)$ rather than $E_{pwd}(y)$.

Pre-computation:
B randomly generates integers x and w
B computes: $r = g^x \bmod p$
$s = b + h(ID_A||r) \bmod (p-1)$
$Sig_B(ID_A) := (r, s)$
$W = g^w \bmod p$
$V = r(PK_A)^w \bmod p$
$Enc_{PK_A}(r) := (V, W)$

Sending phase:
(1) $B \to S$(A is off-line) $Enc_{PK_A}(r), Cert, ID_A$
(2) $S \to B$(A is off-line) $g^y \bmod p, Sig_S(g^y \bmod p)$
B computes: $k = (g^y)^x \bmod p$
(3) $B \to S$(A is off-line) $E_k[M], h(k||g^y \bmod p)$

Receiving phase:
(4) $A \to S$(B is off-line) Request for new mail
(5) $S \to A$(B is off-line) $E_k[M], Enc_{PK_A}(r), Cert,$
$h(k||g^y \bmod p), E_{pwd}(y)$

Fig. 1. Proposed Secure Protocol for E-Mail System

If we now assume that $Cert$ is meant to be a CEMBS for the Schnorr signature under the ElGamal encryption scheme[2], then we note that there is no published method for creating such a CEMBS. Indeed, it is not clear why the authors wish to compute a signature on ID_A in the first place. If their intent is to somehow bind the ephemeral Diffie-Hellman key $g^x \bmod p$ to A's identity then it would be better to produce a signature on $ID_A||(g^x \bmod p)$. The effects of using $r = g^x$ both as the randomiser in the Schnorr signature scheme and as the ephemeral Diffie-Hellman key are unclear.

We note that Bao *et al.* do provide a CEMBS system for a DSA-like signature scheme [2] and that this may be usable as a basis for this protocol.

The main problem with the protocol, however, is that it does not have the claimed forward secrecy property. It is easy to see that any party in possession of both A's private decryption key and password may recover the symmetric key K from the information A receives from S in message (5), and may do so for any earlier interactions between A and S. More generally, no protocol in which A does not actively participate can ever result in perfect forward secrecy. In such cases, A must receive all the information required to compute the symmetric key K from the information received from the e-mail server and his own private keys. Clearly, if A's private keys are compromised then an attacker may compute any message to A in the same manner that A does.

We also question the value of S encrypting y under a password. Since A has published a public key suitable for use with the ElGamal encryption scheme, then surely it is simpler for S to encrypt y using this scheme.

[2]If we do not assume that $Cert$ is meant to be a CEMBS for the signature of ID_A then it is unclear what $Cert$ is meant to represent, and we note that s is never used in the protocol and need not be computed. At this point, the protocol reduces to a scheme whereby an ephemeral Diffie-Hellman key $g^x \bmod p$ is encrypted using an ElGamal system.

Lastly, in their security analysis, Sun *et al.* suggest that the CEMBS protocol is superior to the Diffie-Hellman based protocol proposed in the first part of the paper because the e-mail server S can compute the secret key K in the Diffie-Hellman based protocol, and is unable to compute the key K in the CEMBS based protocol. Careful examination of the schemes will show that this is not the case. The e-mail server S is unable to compute the key K when the Diffie-Hellman based protocol is used, but is able to compute the secret key K when the CEMBS based protocol is used.

III. Conclusion

It has been shown that, amongst other problems, the CEMBS based protocol given by Sun *et al.* [1] does not possess perfect forward secrecy as claimed. Hence, we recommend that this algorithm is not used.

Acknowledgements

The author gratefully acknowledges the financial support of the EPSRC and their Junior Research Fellowship programme. The author would also like to thank Chris Mitchell for useful comments and conversations.

References

[1] H.-M. Sun, B.-T. Hsieh, and H.-J. Hwang, "Secure E-mail protocols providing perfect forward secrecy," *IEEE Communications Letters*, vol. 9, no. 1, pp. 58–60, January 2005.
[2] F. Bao, R. H. Deng, and W. Mao, "Efficient and practical fair exchange protocols with off-line TTP," in *IEEE Symposium on Security and Privacy*, 1998, pp. 77–85.