ISSN 22773061

Fair Certified Email Protocol

Abdullah Mohammed Alaraj

IT department, College of Computer, Qassim University, Saudi Arabia

arj@qu.edu.sa

ABSTRACT

In this paper, a new fair certified email protocol is proposed. The proposed protocol uses offline trusted third party (TTP) that will only be involved if a party misbehaves, otherwise the TTP is inactive. The protocol is based on the verifiable and recoverable encryption of keys. In this regard, the recipient of an email will be able to verify if the encrypted key used to encrypt the email is correct. The protocol comprises of only three messages in the exchange sub-protocol and three messages in the recovery sub-protocol. The proposed protocol is more efficient compared with related protocols.

Indexing terms/Keywords

fair exchange protocols, certified email, protocols, non-repudiation, fairness

Academic Discipline And Sub-Disciplines

Computer science; information security

SUBJECT CLASSIFICATION

Computer science

Council for Innovative Research

Peer Review Research Publishing System

Journal: INTERNATIONAL JOURNAL OF COMPUTERS & TECHNOLOGY

Vol 10, No 1

editor@cirworld.com
www.cirworld.com, member.cirworld.com

1. Introduction

Alice would like to send an electronic mail (email) to Bob and she needs to receive an evidence of receipt from Bob. To provide this service, certified email protocols are used. As in certified postal mail where the postman delivers the mail and asks the recipient to sign a receipt. In the certified email protocols, the items to be exchanged between the sender and the recipient are the email and the evidence of receipt. If the recipient receives the email and the sender receives the evidence of receipt then fairness is ensured for all parties.

The evidence of receipt is called non-repudiation of receipt. It ensures that the recipient cannot deny receiving the email. Some certified email protocols provide evidence called non-repudiation of origin which ensures that the sender cannot deny sending the email. Non-repudiation of origin is optional in certified email protocols [7].

Fair certified email protocols are like other fair exchange protocols using which two parties exchange their items in a fair manner (some protocols are designed to be used by more than two parties). That is, either both parties get each other items or none does.

Early fair exchange protocols are not based on a Trusted Third Party (TTP) [5, 6, 8]. Parties using these protocols are gradually exchanging their items in parts until the whole items are exchanged between the two parties. These protocols assume that the two parties have the same computational power. Additionally, the number of rounds needed to complete the exchange can be high.

TTP is involved in other fair exchange protocols to overcome the problems with protocols that do not involve a TTP. The involvement of the TTP can be inline, online or offline. The inline TTP is used to deliver the exchanged items between parties [3]. The sender sends the email to the TTP. The TTP encrypts the email and sends it to the recipient. The recipient signs the receipt and sends it to the TTP. Finally, the TTP forwards the decryption key to the recipient to decrypt the email and the receipt to the sender. Fairness is ensured but the problem with this approach is that the TTP is heavily involved in the exchange of items because the TTP is used to deliver the items to the parties.

To reduce the involvement of the inline TTP, the online TTP is introduced. The online TTP is used during the protocol execution but its use is not for delivering the items. An example of the use of online TTP is for validating items [2]. Therefore, the online TTP-based protocols reduced the load on the TTP but the problem is that it needs to be active during the exchange of items.

To overcome the problems of inline and online TTP-based protocols, the offline TTP-based protocols are introduced [1, 9, 10, 11, 12, 15]. The offline TTP is not used in the normal execution of the protocol. It is going to be used if something goes wrong during the exchange of items such as one party misbehaves. The offline TTP-based protocols are the best among the TTP-based protocols because the use of the TTP is kept to minimum and hence the cost of running a TTP is reduced. Additionally, the problem of having the TTP as a source of bottleneck is reduced in the offline TTP-based protocols.

Micali [9] proposed a certified email protocol. Micali's protocol consists of two sub-protocols; the exchange sub-protocol consists of three messages and the dispute resolution sub-protocol consists of three messages. The exchange sub-protocol works as follows. Alice sends the first message that includes the email encrypted with Bob's public key and the first message is encrypted with the TTP's public key. When Bob receives the first message, he signs it with his private key. If Alice finds the signed receipt is correct then she sends the email encrypted with Bob's public key.

The dispute resolution sub-protocol of Micali's protocol works as follows. Bob sends to the TTP, (a) the encrypted email that he received from Alice, and (b) Bob's signature on the encrypted email. If the TTP finds Bob's signature is correct then the TTP will send Bob's signature to Alice and the encrypted email to Bob.

Puigserver et al [15] proposed a certified email protocol. Their protocol works as follows. Alice sends the first message to Bob. The first message includes the encrypted email, the key used to encrypt the email encrypted with the public key of the TTP, commitment of Alice to finish the exchange. Once Bob receives the first message, he verifies it and if it is correct then he sends to Alice his commitment to finish the exchange. Then, Alice sends to Bob the decryption key encrypted with Alice's private key. Finally, Bob signs the received message from Alice to acknowledge the receipt of the message. If Bob fails to send his signature to Alice then she contacts the TTP to resolve the dispute.

Another certified email protocol is proposed by Ninadic et al [12]. Their protocol is based on verifiable and recoverable encryption of signatures. Using this approach, a party is be able to verify whether or not the encrypted signature is correct and if yes then the party can send its item to the other party. If the other party fails to send their item then the TTP can recover their item. The protocol starts by Alice sending the following to Bob: her signature on the email, and encrypted email. On receiving Alice's message, Bob verifies it and if it is correct then he sends his signature on the encrypted email, and verifiable and recoverable encryption of signature. Alice then verifies Bob's message and if it is correct then she sends to Bob the decryption key to decrypt the email. Bob finally sends the decryption key to Alice to get Bob's signature. If Alice fails to get the decryption key from Bob then she contacts the TTP to recover the decryption key.

In this paper, a new fair certified email protocol is proposed. The proposed protocol is based on verifiable and recoverable encryption of keys. Using this approach, a party will be able to verify if the encrypted key used to encrypt the email is correct. If it is correct then it is safe for them to release their signature on the email because if the other party fails to send the decryption key then the TTP will be able to recover it. Protocols that are based verifiable and recoverable encryption of signatures (such as in [12]) consist of four messages in the exchange sub-protocol, whereas our protocol consists of three messages. This is because the design of the protocols that are based on verifiable and recoverable encryption of signature is different in that the two parties firstly exchange the encrypted email for the verifiable and recoverable

encryption of signature. If all verifications are correct then the two parties exchange the decryption keys to decrypt the email and the signature of the recipient of the email (evidence of receipt). The design of our approach reduced the number of messages to only three messages as will be explained in the next section.

2. The Proposed Protocol

The proposed protocol consists of two sub-protocols, the exchange sub-protocol and the recovery sub-protocol. The normal execution of the protocol is executed in the exchange sub-protocol where the sender and the recipient of the email behave honestly. The recovery sub-protocol will be executed if one of the parties behaves dishonestly.

The idea of the proposed protocol is as follows. Alice and Bob would like to exchange an email for a receipt. To start the protocol, Alice sends the encrypted email to Bob. Bob will be able to verify the encrypted email using special certificates (M-Cert and C.at that will be discussed in the Notations section). If the encrypted email is correct then Bob will send the receipt to Alice. Finally, Alice sends the decryption key to Bob to get the email. If Alice fails to send the decryption key to Bob then Bob contact the TTP to recover it.

In this approach, Alice is enforced to be honest by enforcing her to send a correct encrypted email to get the receipt. If the encrypted email is incorrect then Bob will not send the receipt. Therefore, Alice has to send the correct encrypted email.

The proposed protocol is based on the following assumptions:

- All parties will use the same encryption, decryption and hash algorithms
- Communication channels are resilient, meaning that all sent messages will be received by their intended receivers
- Necessary timestamps are used in related messages to prevent the replay attacks
- Identifiers will be used to identify the sender and the receiver of the messages. For simplicity they are omitted from the protocol messages

2.1 Notations

The following represents the notations used in the proposed protocol.

- P_a, P_b, and P_t: parties Alice, Bob, and TTP, respectively.
- $C._{at}$: the shared public key certificate between P_a and the TTP. The shared public key between P_a and TTP is denoted as *pkat* = (eat, nat) and its corresponding private key is denoted as skat = (dat, nat). The TTP will have a copy of skat. After creating the shared public key, the TTP will issue the certificate $C._{at}$ of the shared public key and send it to P_a.
- M: the email
- M-Cert: the email's certificate. It is issued by Certificate Authority CA before the protocol starts. It includes the following:
    1. hM: hash value of the email
    2. heM: hash value of encrypted email with k_a
    3. *heKa:* hash value of encrypted *ka* with *pkat*
    4. CA's signature on M-Cert
- K_a: a symmetric key used by P_a
- $Pk_x = (e_x, n_x)$: RSA Public Key [16] of the party x, where n_x is a public RSA modulus and e_x is a public exponent
- $Sk_x = (d_x, n_x)$: RSA Private Key [16] of the party x, where n_x is a public RSA modulus and d_x is a private exponent
- h(M): a strong-collision-resistant one-way hash function
- $enc.pk_x(M)$: an RSA [16] encryption of message M using the public key pk_x (e_x, n_x). The encryption of M is computed as follows. $enc.pk_x(M) = M^{ex} \bmod n_x$
- $enc.sk_x(Z)$: an RSA [16] decryption of Z using the private key sk_x (d_x, n_x). The decryption of Z is computed as follows. $enc.sk_x(Z) = Z^{dx} \bmod n_x$
- $Sig._x(M)$: the RSA digital signature [16] of the party *x* on M. The digital signature of party *x* on M is computed by encrypting the hash value of M using the private key sk_x (d_x, n_x).
- $P_x \rightarrow P_y$: M, means party x sends message M to party y
- X + Y: concatenation of X and Y

2.2 The Exchange Sub-Protocol

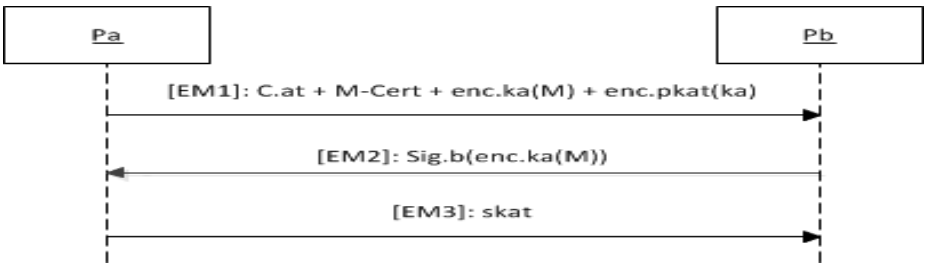

Figure 1: The Exchange Sub-Protocol

ISSN 22773061

The exchange sub-protocol consists of the following steps (messages) as shown in Fig. 1:

1- Alice starts the exchange sub-protocol by sending to Bob the first message that includes the following items: the shared public key certificate (C.at), the email's certificate (M-Cert), and the email encrypted with k_a, and the encryption of k_a with shard public key pk_{at}
2- On receiving the first message, Bob will do the following verifications:
   a. Verify the correctness of both $C._{at}$ and M-Cert by checking their signatures.
   b. Compute the hash value of "$enc._{ka}(M)$" then compare it with "heM" that is included in M-Cert
   c. Compute the hash value of "$enc.pk_{at}(k_a)$" then compare it with "heKa" that is included in M-Cert

   If all verifications are correct then Bob will send the second message to Alice. The second message includes Bob's signature on the encrypted email that was received in the first message. Bob's signature represents that Bob has received the email i.e. it is the non-repudiation of receipt
3- Once Alice receives Bob's message, she will verify the signature. If the signature is correct then she will send the third message to Bob. The third message includes the shared private key to decrypt the encrypted email that was received by Bob in the first message.

At this stage, Alice has received the receipt that confirms that Bob received the email. Also, Bob has received the email. Therefore, fairness is ensured.

If Alice refuses to send the decryption key in the third message then Bob can request it from the TTP as will be described in the next section.

2.3 The Recovery Sub-Protocol

If Bob fails to receive the decryption key in the third message of the exchange sub-protocol, then he can raise a dispute to the TTP as follows (as shown in Fig. 2):

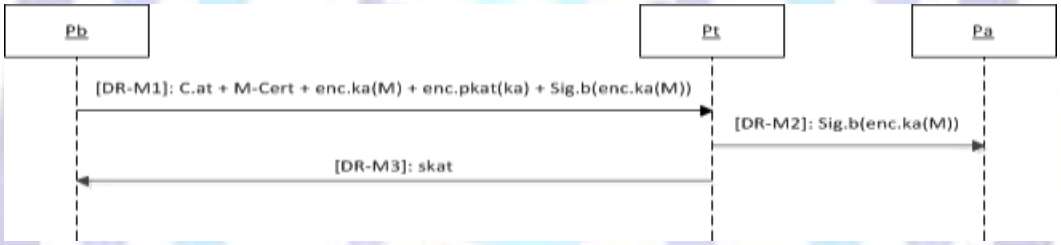

Figure 2: The Recovery Sub-Protocol

1- Bob sends the first message that he received from Alice in addition to Bob's signature on the encrypted email
2- On receiving the first message, TTP will do the following verifications:
   a. Verify the correctness of both $C._{at}$ and M-Cert by checking their signatures.
   b. Compute the hash value of "$enc.k_a(M)$" then compare it with "heM" that is included in M-Cert
   c. Compute the hash value of "$enc.pk_{at}(k_a)$" then compare it with "heKa" that is included in M-Cert
   d. Verify Bob's signature on the encrypted email

   If all verifications are correct, then TTP will send Bob's signature on the encrypted email to Alice and the decryption key to Bob.

The reason for sending Bob's signature to Alice is to ensure fairness for both parties as Bob may contact the TTP before sending his signature on the email to Alice.

3. Discussion

The following discusses the fairness and non-repudiation properties of the proposed protocol.

Fairness: in the certified email protocols, fairness for both the sender and the recipient of the email need to be ensured i.e. either both get each other item or none does. That is, Bob receives the email and Alice receives a confirmation that Bob has received the email.

Regarding the fairness property in the exchange sub-protocol, the following cases will be studied.

1- Messages that Bob needs to receive from Alice to complete the exchange sub-protocol are EM1 and EM3:
   a. If Bob receives correct messages EM1 and EM3 and also Alice receives correct EM2 then fairness is ensured.
   b. If Bob receives EM1 but does not receive EM3 then if Bob does not send EM2 then fairness is not compromised because Alice does not receive Bob's signature on the email and also Alice does not send the decryption key to Bob.
   c. If Bob receives EM1 and sends correct EM2 to Alice but does not receive EM3 then fairness is not ensured and hence Bob needs to use the recovery sub-protocol to ensure fairness.

2- Message that Alice needs to receive from Bob to complete the exchange sub-protocol is EM2:
   a. If Alice receives incorrect EM2 then she will not send EM3 and fairness is ensured.
   b. If Alice receives correct EM2 and send correct EM3 to Bob then fairness is ensured.
   c. If Alice receives correct EM2 but does not send EM3 or sends incorrect EM3 then fairness is not ensured and hence Bob needs to use the recovery sub-protocol to ensure fairness.

Regarding the fairness property in the recovery sub-protocol, the following cases will be studied.

1- Message that Bob needs to receive from TTP to complete the recovery sub-protocol is DR-M3:
   a. If the TTP receives correct DR-M1 from Bob then TTP will send DR-M3 to Bob.
   b. If the TTP receives incorrect DR-M1 from Bob then the TTP will not send DR-M3. Therefore, to ensure fairness Bob needs to send correct DR-M1 to TTP.
2- Message that Alice needs to receive from TTP to complete the recovery sub-protocol is DR-M2:
   a. If the TTP receives correct DR-M1 from Bob then TTP will send DR-M2 to Alice and DR-M3 to Bob. The reason for sending DR-M2 to Alice even Bob is the one who raises dispute is to ensure fairness for both Alice and Bob.
   b. If the TTP receives incorrect DR-M1 from Bob then the TTP will not send DR-M2.

Non-repudiation: it means that the party cannot falsely deny the transaction. The proposed protocol provides non-repudiation of receipt. That is, Bob cannot deny that he has not signed the receipt sent in message EM2 because the receipt is signed using Bob's private key. Non-repudiation of origin is not supported in the proposed protocol as it is optional in certified email protocols [7]. However, if we need to provide it then message EM1 should include Alice's signature on the email.

4. Comparison

In the following, the comparison between our protocol and related protocols is presented. As can be seen in Table 1, our protocol and Micali's protocol [9] has the lowest number of messages in the exchange sub-protocol with only three messages. All protocols have three messages in the dispute sub-protocol. Regarding the number of modular exponentiations which is considered to be the most expensive operation [1], our protocol has the lowest number of modular exponentiations in the exchange sub-protocol with only six. In the dispute sub-protocol, our protocol and Micali's protocol [9] have the highest number of modular exponentiations with three.

Table 1. Protocols Comparisons

	Micali protocol [9]	Nenadic et al Protocol [12]	Puigserver et al Protocol [15]	Our protocol
# messages in the exchange sub-protocol	3	4	4	3
# messages in the dispute sub-protocol	3	3	3	3
# modular exponentiations in exchange protocol	7	14	9	6
# modular exponentiations in dispute resolution protocol	3	2	2	3

5. Conclusion

In this paper, a new fair certified email protocol was proposed. The proposed protocol is based on offline TTP that will only be involved if one party misbehaved. The protocol comprises of only three messages in the exchange sub-protocol and three messages in the recovery sub-protocol. The approach used to design the protocol is based on verifiable and recoverable encryption of keys. The use of this approach resulted in having less number of messages and also less number of modular exponentiations comparing with other certified email protocols.

The future work will include formal verification of the proposed protocol and implementing it.

References

1. Alaraj, A. (2012). Simple and Efficient Contract Signing Protocol. International Journal of Advanced Computer Science and Applications (IJACSA), Vol. 3, No. 3, 67-71.
2. Abadi, M., Glew, N., Home, B. & Pinkas, B. (2002). Certified email with a light on-line trusted third party: design and implementation. In Proceedings of the International World Wide Web Conference –WWW '02, ACM Press, New York, USA, 387–395.
3. Bahreman, A. & Tygar, J. (1994). Certified electronic mail. In Proceedings of the 1994 Network and Distributed Systems Security Conference, February 1994, 3-19.

4. Basagiannis, S., Petridou, S., Alexiou, N., Papadimitriou, G., & Katsaros, P. (2011). Quantitative analysis of a certified e-mail protocol in mobile environments: A probabilistic model checking approach. computers & security, 30, 257-272.
5. Ben-Or, M., Goldreich, O., Micali, S. & Rivest, R. (1990). A fair protocol for signing contracts. IEEE Transactions on Information Theory, vol. 36, no. 1, 40-46.
6. Blum, M. (1983). How to exchange (secret) Keys, *ACM Transactions on Computer Systems* 1(2), 175–193.
7. Ferrer-Gomillaa, J., Onievab, J., Payerasa, M., & Lopezb. J. (2010). Certified electronic mail: Properties revisited. computers & security 29, 167–179.
8. Markowitch, O. & Roggeman, Y. (1999). Probabilistic non-repudiation without trusted third party. In *Proceedings of the Second Conference on Security in Communication Networks – SCN '99*,
9. Micali, S. (2003). Simple and fast optimistic protocols for fair electronic exchange. In Proceedings of the twenty-second annual symposium on Principles of distributed computing, PODC '03, July 13-16, 2003, Boston, Massachusetts, 12-19.
10. Nenadic, A., Zhang, N. & Barton, S. (2004). A security protocol for certified e-goods delivery. In *Proceedings of the IEEE International Conference on Information Technology, Coding and Computing –ITCC '04*, IEEE Computer Society, 22–28.
11. Nenadic, A., Zhang, N. & Barton, S. (2004). Fair certified e-mail delivery. In *Proceedings of the ACM Symposium on Applied Computing – SAC '04*, 391–396.
12. Nenadić, A., Zhang, N., Shi, Q. (2005). RSA-based Verifiable and Recoverable Encryption of Signatures and its application in certified e-mail delivery. Journal of Computer Security 13 (5) IOS Press, 757–777.
13. Onieva, J., Zhou, J., Lopez, J. (2004). Enhancing certified email service for timeliness and multicast. In Fourth international network conference, 327–335.
14. Oppliger, R. (2004). Certified mail: the next challenge for secure messaging. ACM Press. Communications of the ACM, 75–79.
15. Puigserver, M., Gomila, J., Rotger, L. (2005). Certified e-mail protocol with verifiable third party. In Proceedings of the 2005 IEEE international conference on e-technology, e-commerce and eservice, EEE '05, 548–551.
16. Rivest, R., Shamir, A. & Adleman, L. (19787). A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, vol. 21, no. 2, 120 –126.
17. Shao, M., Wang, G., Zhou, J. (2006). Some common attacks against certified email protocols and the countermeasures. Computer Communications (29), 2759–2769.