A New Approach of Signing Documents with Symmetric Cryptosystems and an Arbitrator

Nol Premasathian

inolhian@kmutt.ac.th

Faculty of Science
King Mongkut's University of Technology Thonburi
Bangkok, Thailand

Abstract

Signing documents can be done by using some public-key algorithms. Alternatively, there is a protocol to sign documents with symmetric cryptosystems and an arbitrator. The protocol requires each the sender and the receiver to maintain a secret key with the arbitrator. Messages and acknowledgments are sent through the arbitrator, who will decrypt it using the private key of the sender and encrypt it using the private key of the receiver. The arbitrator has to keep records of every signed message and acknowledgment transmitted. Since the arbitrator may work for several pairs of people, it can be a bottleneck in the transmission. This paper presents a new approach of signing documents with symmetric cryptosystems and an arbitrator. The new approach uses three private keys instead of two and has three improvements. First, it reduces the number of transmissions from four to three. It also reduces the number of cryptographic operations performed by the arbitrator. Second, the arbitrator needs not keep a record of each transmission. The proof of sending is kept by the message receiver while the proof of the message acknowledgment is kept by the message sender. The sender and the receiver naturally feel more secure to have the proof with them. Third, although the arbitrator is trusted by both parties, it doesn't mean that they want to reveal the content of the message to the arbitrator. In the new approach, the arbitrator will not perceive the content of the message. This paper explains how the new approach can be used to send a signed message, to acknowledge the message and how to prove the sending or receiving when a dispute occurs.

Keywords: digital signatures, symmetric cryptosystems, private key, secret key.

1 Introduction

With the expansion and the use of Internet, the electronic means of communication (exchange of information) are becoming progressively more important [2]. It is often useful to prove that a message was generated by a particular individual, if the individual is not necessarily around to be asked about authorship of the message [3]. The sender can sign a message using digital signature, which depends on the contents of the message. Most previously proposed signature schemes were based on well-known public key systems such as RSA system [4] and ElGamal system [1][7]. A disadvantage of public key systems is the speed [5]. It is possible to sign a document using private key systems and an arbitrator or a third person who is trusted by both the sender and the receiver.

2 The Existing Scheme

The protocol that provides digital signatures using symmetric cryptosystems has been invented for some time [6]. It requires the help of an arbitrator. The sender shares a secret key K_s with the arbitrator while the receiver shares a secret key K_r with the arbitrator. A signed message can be sent as follows.

1. The sender encrypts the message with K_s and sends it to the arbitrator.
2. The arbitrator decrypts the message with K_s.
3. The arbitrator takes the decrypted message and a statement that he has received this message from the sender, and encrypts them with K_r.
4. The arbitrator sends the encrypted message and the statement to the receiver.
5. The receiver decrypts the message and the statement with K_r.

A message can be acknowledged in a similar way. The receiver composes an acknowledgment statement, encrypts it with K_r and sends it to the arbitrator, who will decrypt it with K_r, encrypt it with K_s and send it to the sender. Four transmissions are required to send a signed message and receive a signed acknowledgment. The arbitrator has to keep a record for every message and acknowledgment transmitted. If the sender refuses to recognise the sending of a message or the receiver refuses to recognise the acknowledgment of a message, disputes can be dissolved according to the records kept by the arbitrator. If the record is lost, the dispute cannot be dissolved. The protocol also requires the arbitrator to encrypt and decrypt all messages and acknowledgements. If messages are large and the arbitrator works for several pairs, this can be a bottleneck in communication. In addition, the content of the message is revealed to the arbitrator.

3 The Proposed Scheme

In this section, we propose a new protocol to sign and acknowledge a message. In this protocol, the sender shares a secret key K_s with the arbitrator, the receiver shares a secret key K_r with the arbitrator, and the sender shares a secret key K_m with the receiver. A signed message M can be sent and acknowledged as follows.

1. The sender computes the hash of the message H(M), encrypts the hash with K_s, encrypts the message with K_m and sends $K_s(H(M))$ and $K_m(M)$ to the arbitrator.
2. The arbitrator decrypts the hash with K_s to get H(M), computes $K_r(K_r(H(M)))$, hashes it to get $H(K_r(K_r(H(M))))$, combines $H(K_r(K_r(H(M))))$ with H(M) to get $H(K_r(K_r(H(M))))+H(M)$ and encrypts it with K_s to get $K_s(H(K_r(K_r(H(M))))+H(M))$.
3. The arbitrator sends $K_m(M)$, $K_s(H(M))$, $K_r(K_r(H(M)))$, and $K_s(H(K_r(K_r(H(M))))+H(M))$ to the receiver.
4. The receiver decrypts $K_m(M)$ to get the message, computes $K_r(K_r(H(M)))$ from the message and compare it with the $K_r(K_r(H(M)))$ that he received from the arbitrator. If they are the same, it means that the signature $K_s(H(M))$ is valid. The receiver then sends $K_r(K_r(H(M)))$ and $K_s(H(K_r(K_r(H(M))))+H(M))$ to the sender.
5. The sender decrypts $K_s(H(K_r(K_r(H(M))))+H(M))$ to get $H(K_r(K_r(H(M))))$ and H(M). He can verify H(M) with the original one to check the validity of the acknowledgement. If they are the same and the hash of the received $K_r(K_r(H(M)))$ is the same as the $H(K_r(K_r(H(M))))$ decrypted from $K_s(H(K_r(K_r(H(M))))+H(M))$, the acknowledgment $K_r(K_r(H(M)))$ is valid.

In the proposed scheme, the content of the message M is not revealed to the arbitrator since it is encrypted with K_m, which is shared only between the sender and the receiver. The encrypted message is forwarded to the receiver without a modification along with other information including the signature $K_s(H(M))$. $K_s(H(M))$ can be used as a signature because of two reasons. First, it is computed from the message. Second, the sender possesses the key K_s that can compute the signature from the message whereas the receiver does not. Although the arbitrator also possesses this key but we must assume that the arbitrator is trusted by both the sender and the receiver and therefore we assume that he will not cheat. The receiver knows that the signature $K_s(H(M))$ is a valid signature for the message M, though he does not have K_s to decrypt the signature and verify it. That is because he also receives $K_r(K_r(H(M)))$, that was computed from the same H(M) the signature $K_s(H(M))$ by the arbitrator. He can verify the validity of $K_r(K_r(H(M)))$ by hashing the message M, encrypting it with K_r twice and compare it with the received $K_r(K_r(H(M)))$. In this way, the validity of the signature is verified. Similarly, $K_r(K_r(H(M)))$ received by the sender can be used as the acknowledgment of the message M. It is computed from the message M using the key K_r that is possessed by the receiver but not the sender. The sender can verify the validity of the acknowledgment by decrypting $K_s(H(K_r(K_r(H(M))))+H(M))$, that was computed by the arbitrator and forwarded to the sender from the receiver, to get $H(K_r(K_r(H(M))))+H(M)$. He can verify H(M) with the original one. The arbitrator used the same H(M) to compute $H(K_r(K_r(H(M))))$ and therefore it can be used to verify the validity of the acknowledgment by hashing the acknowledgment and compare it with the received $H(K_r(K_r(H(M))))$. H(M) is encrypted by K_r twice to make it differ from the signature. Note that the receiver cannot modify $K_s(H(K_r(K_r(H(M))))+H(M))$ since it is encrypted with K_s.

In the proposed scheme, there are three transmissions, each in step 1, 3 and 4. The cryptographic operations that the arbitrator has to perform are encrypting hash values and hashing encrypted hash values. These operations perform on fixed-length data regardless of the size of the message. He does not hash or encrypt the whole message, which can be long and may take a lot of time. This thus increases the efficiency of the arbitrator.

4 Dissolving Disputes

There are two kinds of disputes here. The sender does not recognise the signature and the receiver does not recognise the acknowledgment. Both disputes can be dissolved by the arbitrator without revealing the key K_m to the arbitrator. However, the content of the disputed message must be revealed.

If the sender refuses the sending of a message, the receiver can present the message M, and the signature $K_s(H(M))$ to the arbitrator. The arbitrator hashes M to get H(M), encrypts it using K_s and compares it with the signature $K_s(H(M))$ presented by the receiver. If they are the same, the arbitrator will declare the signature valid. The sender cannot claim that the signature $K_s(H(M))$ was produced by the receiver since the receiver does not possess the key K_s.

If the receiver refuses the receiving of a message, the sender can present the message M, and the acknowledgment $K_r(K_r(H(M)))$ to the arbitrator. The arbitrator hashes M to get H(M), encrypts it using K_r twice and compares it with the acknowledgment $K_r(K_r(H(M)))$. If they are the same, the arbitrator will declare acknowledgment valid. The receiver cannot claim that the acknowledgment $K_r(K_r(H(M)))$ was produced by the sender since the sender does not possess the key K_r.

When a key is updated, the old key must be properly archived so that any dispute about a signature or an acknowledgment using the old key can be dissolved in the future.

5 Conclusion

This paper presents a new approach of signing and acknowledging documents using symmetric key cryptosystems and an arbitrator. In the new approach, the number of transmissions is reduced from four to three, the arbitrator is not required to keep a record of each sending, the number of cryptographic operations performed by the arbitrator is reduced and the content of the message is not revealed to the arbitrator. The paper does not specify a particular encryption algorithm or a hash function to be used. Any person who is interested in implementing this protocol should choose an encryption algorithm carefully as some of them may be vulnerable to a certain attack when used in this protocol.

References

[1] ElGamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms, *IEEE Trans. Inform. Theory* 31 (4) , pp.469-472, 1985.

[2] Janka, J.: Use of public key infrastructure, in *Proc. Security and Protection of Information*, Brno, Czech Republic, 2001.

[3] Kaufman, C., Perlman, R., and Speciner, M.: Network security, Prentice Hall, 1995.

[4] Rivest, R.L., Shamir, A., Andelman, L.: A method for obtaining digital signature and public key cryptosystem, *Comm. ACM* 21(2), pp. 120-126, 1978.

[5] RSA Security Inc.: RSA Laboratories frequently asked questions about today's cryptography, Version 4.1, http://www.rsasecurity.com/rsalabs/faq/2-1-3.html, 2003.

[6] Schneier, B.: Applied cryptography, Wiley, 1996.

[7] Tseng, Y., Jan, J., Chien, H.: Digital signature with message recovery using self-certified public keys and its variants, *Journal of Applied Mathematics and Computation* 136, pp. 203-214, 2003.