A MUTUAL NON-REPUDIATION PROTOCOL WITH PRIVACY

JIANG Xiao-ning(蒋晓宁), YE Cheng-qing(叶澄清)
(*Dept. of Computer Sci. & Tech., Zhejiang University, Hangzhou, 310027, China*)

Received June 1, 1999; revision accepted Nov.30,1999

Abstract: Non-repudiation services provide the parties involved in a transaction with protection against the other party by denying that a particular event or action took place. They collect irrefutable evidence to support the resolution of any such disagreement. We address this security issue by first examining the previous work done in this area, and then propose a novel protocol to achieve mutual non-repudiation service, encompassing both mandatory evidence of origin and mandatory evidence of receipt. By using two simple ideas, a conditional signature and a public notice board, the novel protocol can achieve this security service in a simple but effective manner. By applying cryptography technology, this protocol also provides privacy for the parties using the security service.

Key words: non-repudiation, privacy, electronic business, network security.
Document code: A **CLC number**: TP393; TP309

INTRODUCTION

Government and commercial organizations rely heavily on the use of information to conduct business. Loss of confidentiality, integrity, authenticity, accountability, availability, and unauthorized use of information can have an adverse impact on these organizations. There are three major trends now causing communications security concerns to escalate (Tanenbaum, 1996):

1. Increasing interconnection of systems and of networks, making any system potentially accessible to a rapidly growing population of (known and unknown) users;

2. Increasing use of computer networks for security-sensitive information, for example, electronic funds transfer, business data interchange, government unclassified but sensitive information, and corporate proprietary information;

3. Increasing ease of engineering a network attack, given the ready availability of sophisticated technology and the rapidly falling cost of such technology to a would-be attacker.

The potential motivations for attacks on commercial or unclassified government networks may vary. They include financial fraud, theft of telecommunications resources, industrial espionage, illicit eavesdropping for financial or political gain, egotistical and mischievous gratification. In addition to these deliberate attacks, communications security also needs protection against accidental exposures. Accidental connection of a sensitive communications session to the wrong address or accidental failure to protect sensitive information may prove as damaging as a successful deliberate attack.

Repudiation is one of the fundamental security threats existing in social and electronic environments. The motivation for non-repudiation services is not just the possibility that communicating parties may try to cheat each other. It is also a reflection of the reality that no system is perfect, and that circumstances can arise in which two parties end up with different views of something that happened. If two possibilities regarding an event cannot be distinguished, a party related to the event can deny it occurred.

Non-repudiation is concerned with preventing such a denial. With sender non-repudiation, the sender of data is provided with a proof of receipt (POR) showing that the other party in the data exchange received the data. Receiver non-repudiation provides the recipient with a proof of origin (POO) showing that the originator sent the data. The proofs of origin and receipt constitute non-repudiation evidence information. Principals can exchange evidence information, either through direct peer-to-peer communication or in-directly via a third-party intermediary.

The correct generation of evidence information is crucial to non-repudiation. The proof of origin must associate the identity of the origination with the data exchanged in such a manner that the originator cannot deny this association. Likewise, the identity of the recipient is associated with the proof of receipt. The evidence must be undeniable and unforgeable. These properties are achieved through the use of digital signatures.

A non-repudiation service must provide an arbitration framework for addressing disputes. If a dispute arises, it may be possible for the disputing principals to resolve it themselves by exchanging and examining the evidence information. If this does not suffice, then an agreed arbitrator, who is trusted by both principals, is called upon to mediate a settlement. The entities involved in the exchange present evidence to this arbitrator who uses a set of well-defined rules to decide, based on the evidence submitted, whether or not a data exchange took place.

This paper is organized as follows. Section 2 surveys the related work done in the area, and discusses its suitability with regard to a globally distributed and heterogeneous electronic commerce environment. Section 3 introduces the basic notation to be used in this paper. Section 4 presents a mutual protocol for non-repudiation service. Section 5 discusses some implementation issues. Some concluding remarks and future work are outlined in Section 6.

RELATED WORK

Some work has been done in designing protocols to support non-repudiation service for mutually distrustful business parties to electronically exchange messages such as signed contracts and certified mails. Generally speaking, fair non-repudiation protocols can be constructed in the following two ways:

1. The originator and the recipient exchange the message and/or non-repudiation evidence simultaneously;

2. Trusted third parties assist in fair exchange of the message and/or non-repudiation evidence.

In the first approach, e.g. (Bahreman et al., 1994), fairness is achieved by the gradual release of secrets over many rounds: during each round, some knowledge about the message and/or evidence is revealed. If either party stops before the protocol run is complete, both parties are left with comparable knowledge. This approach seems to be too cumbersome and inefficient for actual implementation. Moreover, fairness is based on the assumption of equal computational complexity, which makes sense only if the two parties have equal computing power, an often unrealistic and undesirable assumption (Anderson , 1994).

A few fair non-repudiation protocols based on the active involvement of trusted third parties exist (Zhou et al., 1997, Balenson, 1993, and Coffey etc., 1996). From the investigation, we found that they differ in

1. The degree to which a trusted third party has to be involved in a protocol run.

2. The assumptions on the availability of communications links.

We can classify the involvement of trusted third parties into three levels:

1) Non-repudiation with an in-line trusted third party: A trusted third party acts as an intermediary between the originator and the recipient and intervenes directly in a non-repudiation service, e.g. a delivery authority.

2) Non-repudiation with an on – line trusted third party: A trusted third party is actively involved in every instance of a non-repudiation service, e.g. a public notary.

3) Non-repudiation with an off-line trusted third party: A trusted third party supports non-repudiation without being involved in each instance of a service, e.g. a certification authority.

Our non-repudiation protocol only needs an on-line low weight notary and a weak assumption on the availability of communications links.

BASIC CONCEPTS AND NOTATION

Fairness may be a desirable property of a non-repudiation service. Protocols can achieve fairness through the involvement of a trusted third party, but the extent of the trusted third party's involvement can vary between protocols. The third party may become a performance bottleneck and not work efficiently. Hence, one of the goals of designing an efficient non-repudiation protocol is to reduce the workload of the trusted third party. Another goal is to provide privacy for principals since in many circumstances the trusted third party should not even know the contents of the messages transmitted.

We introduce the following notation to represent messages and protocols.

1) $X; Y$: concatenation of two messages X and Y, in the order specified.

2) $H\{X\}$: a one-way hash function of message X.

3) $eK\{X\}$: encryption of message X with key K.

4) V_A and S_A : the public and private key of principal A.

5) $sK\{X\}$: digital signature of message X with the private key K.

6) $A \rightarrow B$: X: principal A sends message X to principal B.

7) $A \leftrightarrow B$: I, X: principal A fetches message X from principal B in terms of identifier I, i.e. A first sends I to B who then replies with the message X associated with I.

OUR APPROACH

1. The assumption

There is a trusted third party called *NRS* (non-repudiation server). The responsibilities of *NRS* are; first, to make session keys public in due course, and second, to maintain a database of published session keys. This database is read-only, and must be protected from unauthorized modifications. By 'make a session key public' we mean publishing the session key in a way similar to a 'public notice-board'. The information on the 'public notice-board' is unconditionally readable to any viewers, regardless of their identities. The practical implementation of the ' public notice-board' may vary, e.g. it can be a known query-reply service provided by the server.

All parties including *NRS*' have access to conventional and public-key cryptosystems, as well as a Certification Authority (*CA*) that generates and manages public-key certificates (*PC*s) for the parties. For example, PC_A is party A's certificate signed by the *CA* which specifies A's distinguished name (e.g., name, organization, email address, etc.), the corresponding public key V_A, and its issuing date.

2. The protocol

The basic idea of this protocol is that the originator of a message first chooses a conventional session key and sends a recipient the cipher-text of the message encrypted using the session key. Then, the recipient acknowledges the receipt of the cipher-text message by returning a conditional signature valid if and only if the following two conditions are met:

1. The session key must be made public no later than a specified time.

2. The session key published can correctly decrypt the cipher-text message.

Upon the receipt of the conditional signature, the message originator sends *NRS* the session key together with its relevant information, and *NRS* simply publishes the session key and information received. Finally, the recipient fetches the session key from *NRS*, and uses it for the decryption of the cipher-text to recover the message.

To present the protocol in detail, let us suppose that A has a message M for B. To start with, A chooses a one-time conventional session key K_{AB}, and sends B the following message.

Step 1: $A \rightarrow B$: $eV_B(PC_A, eK_{AB}\{M\}), sS_A(H(eK_{AB}\{M\}), Lable)$

PC_A is included to facilitate B's decryption of A's signature. $eV_B(PC_A, , eK_{AB}\{M\})$ is the message part which is encrypted, using B's public key V_B certified in certificate PC_B, to protect the secrecy of message M, since session key K_{AB} will be made public eventually. $sS_A(H(eK_{AB}\{M\}), Lable)$ is the signature part. Here, *Lable* is a session-dependent number, e.g. a large freshly chosen random number, to ensure that the signature is only valid for this session. $H(eK_{AB}\{M\})$ is embodied for B to verify the originality and integrity of cipher-text $K_{AB}\{M\}$ in the message part, and the use of the hash function is for security reasons because the signature will be published along with session key K_{AB}.

Step 2: $B \rightarrow A$: $PC_B, sS_B(Lable, t_B)$

t_B is a future time of *NRS*'s clock, specified by B. This future time indicates that A must get session key K_{AB} published by NRS no later than t_B in order to obtain a valid receipt. In fact, t_B defines a condition for the validity of B' s signature, namely B' s signature will be invalid if key K_{AB} arrives at NRS later than t_B. This condition is imposed to stop A deliberately delaying the publication of the session key so as to gain some financial or business advantages. Note that the specification of t_B should take into account the worst-case clock drift among A, B and NRS. Normally a clock drift less than a few seconds is guaranteed by the current network time service.

Step 3: $A \rightarrow NRS$: $eV_{NRS}(PC_A, Lable, t_B)$, $sS_A(H(eK_{AB}\{M\}), K_{AB}, t_B)$

A' s signature comprises three important items that influence the successful completion of the current session, i.e. key K_{AB} is associated with $H(eK_{AB}\{M\})$ and must be published by NRS before t_B.

Step 4: $B \leftrightarrow NRS$: $Lable, I_A$

In this step NRS records the arrival time t_{NRS} of message 3, which is also assumed to be the publication time of the session key, and makes the following information accessible to any party:

$I_A = Lable, PC_{NRS}, sS_{NRS}(Lable, Late)$, if t_{NRS} is later than t_B.

$I_A = Lable, PC_{NRS}, sS_{NRS}(Lable, sS_A(H(eK_{AB}\{M\}), K_{AB}, t_B))$, otherwise.

In case message 3 has not reached NRS, any request on information with identifier *Label* will get the following reply:

$I_A = Lable, PC_{NRS}, sS_{NRS}(Lable, Null)$.

Step 5: $A \leftrightarrow NRS$: $Lable, I_A$

Though this step is optional, A is adviced to execute it to obtain and keep I_A as evidence for publication of the session key.

3. Handling dispute

Disputes can arise over the origin and receipt of a message M. In the first case, B claims to have received M from A while A denies having sent M to B. In the second case, A claims having sent M to B while B denies having received M. These disputes can be resolved by a judge who evaluates the evidence held by the participants and determines whether receipt or origin are as claimed.

Repudiation of origin

If B claims that it received M from A, the judge will require B to provide M, $eK_{AB}\{M\}$, *Lable*, and the non-repudiation evidence $sS_A(H(eK_{AB}\{M\}), Lable)$ and $sS_{NRS}(Lable, sS_A(H(eK_{AB}\{M\}), K_{AB,}, t_B))$. If B cannot provide this evidence or one of the following checks fails, B's claim will be judged invalid.

The judge checks that $sS_{NRS}(Lable, sS_A(H(eK_{AB}\{M\}), K_{AB}, , t_B))$ is NRS 's signature;

The judge checks that $sS_A(H(eK_{AB}\{M\}), Lable)$ is A's signature;

The judge checks that $M = dK_{AB}\{eK_{AB}\{M\}\}$.

If the first check is positive, the judge will assume that the entry in the NRS's directory was made because A had submitted the key K_{AB} for use with the message with label *Lable*. Here, the judge trusts the NRS to generate valid evidence. If the second check is positive, the judge will assume that A had sent $eK_{AB}\{M\}$ as its commitment for the message with label Lable. If the final check is positive, the judge will uphold B's claim.

Repudiation of receipt

If A claims that B had received M from A, the judge will require A to provide M, $eK_{AB}\{M\}$, *Lable* and the non-repudiation evidence $sS_B(Lable, t_B)$ and $sS_{NRS}(Lable, sS_A(H(eK_{AB}\{M\}), K_{AB}, , t_B))$ (or provided by NRS). If A cannot provide this evidence or one of the following checks fails, A's claim will be judged invalid.

The judge checks that $sS_{NRS}(Lable, sS_A(H(eK_{AB}\{M\}), K_{AB}, , t_B))$ is NRS 's signature;

The judge checks that $sS_B(Lable, t_B)$ is B' s signature;

The judge checks that $M = dK_{AB}\{eK_{AB}\{M\}\}$.

The first check is as above. If the second check is positive, the judge will assume that B had received $eK_{AB}\{M\}$ and is committed to retrieving the K_{AB} from NRS. If the final check is positive, the judge will uphold A's claim.

4. Reliable and secure *NRS*

A trusted third party NRS plays important roles in non-repudiation services. The reliability and security of a non-repudiation server is a practical problem, which needs to be addressed. For various reasons, a non-repudiation server may fail or make mistakes. In our non-repudiation protocol, if *NRS* crashed and lost the data of I_A at the time when one entity has collected I_A while another entity has not, fairness will be disrupted. To increase its reliability, *NRS* should be able to recover from network and system failures, and maintain a consistent directory accessible (read only) to the public. This requires proper backup and recovery mechanisms.

Non-repudiation services will rely on trustworthy functions of the non-repudiation server involved. Any dishonest actions of NRS will affect non-repudiation services. In the real world, an individual NRS may not be completely and permanently trustworthy. To enhance trust in non-repudiation services, one possible solution is to distribute trust among a group of non repudiation servers, which could be based on threshold schemes (Shamir, 1979). This idea has been used in distributed authentication services so that a minority of malicious and colluding servers cannot compromise security or disrupt service (Gong, 1993). However, this solution will increase the cost and downgrade the efficiency of non-repudiation services. Another possible solution is to put a trusted third party under supervision. This is outside the scope of the security technology.

EXPERIMENT AND CONCLUSION

A prototype of the non-repudiation protocol has been implemented. The experiment environment consisted of COMPAQ™ 5000 Server, Legend™ P233 Workstations and 10M Ethernet LAN. The operation system is Windows™ NT 4.0 plus SP3. Our non-repudiation protocol is based on CryptoAPI. We choice RC4 as the encryption/decryption algorithm, RSA the signature algorithm, and MD5 the hashing algorithm. The experiment result shows that a protocol run takes about 4 seconds. This speed for a fair business transaction, providing both mutual non-repudiation evidence and message privacy, is feasible.

The characters of our protocol can be summarized as follows:

1. Our protocol requires the existence of a trusted third party *NRS*, but neither the plaintext nor ciphertext of a message *M* is viewed or maintained by *NRS*. Privacy is achieved by this way.

2. NRS only handles publication of a session key. It is unlikely to become a performance bottleneck.

3. Our protocol is efficient in a sense that it requires normally four message transactions to perform a single message/receipt exchange.

4. It only requires clocks to be loosely synchronized, which can be easily satisfied by the current network technology.

ACKNOWLEDGEMENTS

The authors would like to thank the anonymous referees for their most helpful and constructive comments.

References

Anderson, R., 1994. Why cryptosystems fail. *Communications of the ACM*, **37**(11): 32 - 41.

Bahreman, A., Tygar, J. D., 1994. Certified electronic mail. *In*: Proc. Of the Internet Society Symposium on Network and Distributed System Security. IEEE Computer Society Press, New York, p. 3 - 19.

Balenson, D., 1993. Privacy enhancement for Internet electronic mail: Part III-Algorithms, modes, and identifiers. *RFC* 1423.

Coffey, T., Saidha P., 1996. Non-repudiation with mandatory proof of receipt. *Computer Communication Review*, **26**(1): 6 - 18.

Gong, L., 1993. Increasing availability and security of an authentication service. *IEEE Journal on Selected Areas in Communications*, **11**(6): 657 - 662.

Shamir, A., 1979. How to share a secret. *Communications of the ACM*, **22**(11): 612 - 613.

Tanenbaum, A.S., 1996. Computer Networks. 3rd edition. Inc., a Simon & Schuster Company, Prentice Hall, p. 577 - 622

Zhou, J., Gollmann, D., 1997. An efficient non-repudiation protocol. *In*: Proc. Of the 10th IEEE Computer Security Foundations Workshop, IEEE Computer Society Press, Rockport, Massachusetts, p. 126 - 132.