A MULTI-PARTY NON-REPUDIATION PROTOCOL

Steve Kremer and Olivier Markowitch
Université Libre de Bruxelles
Dept of Computer Science, Bd du Triomphe, C.P.212,
1050 Bruxelles, Belgium
skremer@ulb.ac.be, omarkow@ulb.ac.be

Abstract We present a multi-party non-repudiation protocol, based on a group encryption scheme. We define multi-party non-repudiation, compare it to multi-party fair exchange and show some fundamental differences between these two problems. This is the first effort to generalize non-repudiation to the multi-party case. Our definitions and the resulting protocol are more general than the ones given in the only comparable work, a multi-party certified mail protocol. Finally we also give an example of a group encryption scheme that can be used with our protocol.

Keywords: Non-repudiation, multi-party protocols, group encryption.

Introduction

During the last years the impressive growth of the Internet and more generally of open networks has created several security related problems. The non-repudiation and the related fair exchange problem are two of them. Non-repudiation must ensure that no party involved in a protocol can deny having participated in a part or the whole of the protocol. Therefore a non-repudiation protocol has to generate non-repudiation of origin evidences intended to the recipient parties and non-repudiation of receipt evidences destined to the originating parties. In case of a dispute (e.g. an originator denying having sent a given message or a recipient denying having received it) an adjudicator can evaluate these evidences and take a decision in favor of one of the parties without any ambiguity. In fair exchange two parties want to exchange items and the protocol ensures that either both parties receive the desired item, or none of them receives valuable information.

First solutions to those problems involve a trusted third party (TTP) that acts as an intermediary between the participating entities. The major disadvantage of this approach is the communication bottleneck created at the TTP. Therefore more efficient solutions have been proposed. Two different approaches have been considered: one consists in designing protocols without a TTP, the other tries to minimize its involvement.

The approach without TTP involvement is often based on a gradual release of the knowledge. However it generally requires that all involved parties have the same computational power. Another disadvantage is the important number of transmitted messages. In Ben-Or et al., 1990 a protocol for digital contract signing without TTP has been proposed: the probability that the contract has been signed, is increased each round until reaching one. The assumption of same computational power is not needed. Another probabilistic non-repudiation protocol, recently presented by Markowitch and Roggeman, 1999, also succeeded in relaxing the condition on the computational power. Here the idea is that the recipient does not know a priori which transmission will contain the message. The probability of guessing the transmission including the message is arbitrarily small. However to decrease the probability the number of messages has to be increased.

The other approach, trying to minimize the TTP involvement has got more attention during the last years.Asokan, 1998 presented the optimistic approach: usually all participants are honest and only in the case of a misbehaving party, the TTP has to be involved. Zhou and Gollman presented a protocol where the TTP intervenes during each execution as a 'low weight notary', rather than as an intermediary. They presented in Zhou and Gollmann, 1997 a variant of this protocol inspired by Asokan's optimistic approach as well.

Most of these protocols have been designed as two-party protocols. In fair exchange first works have been done to generalize them to the case of n participants: Asokan et al., 1996; Asokan et al., 1998; Franklin and Tsudik, 1998; Bao et al., 1999. Considering non-repudiation no investigations towards a generalization have yet been undertaken. The only to us known comparable work is the multi-party certified mail protocol proposed by Asokan et al. in Asokan et al., 1998. However the here proposed protocol is more general: as it will be outlined later in more details, the certified mail protocol only continues if the whole set of receivers is willing to do so. Our protocol leaves the choice to the sender to finish with only the subset of the responding receivers, or to stop in the case of a non-responding receiver, as the certified mail protocol does.

In this paper we will first define the multi-party non-repudiation problem showing the difference with the multi-party fair exchange. The requirements of a multi-party non-repudiation protocol shall be defined as well. Then we will remind a protocol designed by Zhou and Gollman and finally present a generalization of this protocol to the case of n parties, using a group encryption scheme.

1. MULTI-PARTY NON-REPUDIATION

In literature, different kinds of multi-party fair exchange have been considered. In Franklin and Tsudik, 1998 a classification has been proposed. One can differ between single-unit and multi-unit exchanges. Moreover different topologies are possible: Franklin and Tsudik, 1998 and Bao et al., 1999 concentrated on a ring topology. Each entity e_i $(0 \leq i \leq n-1)$ desires an item (or a set of items) from entity $e_{i \boxminus 1}$ and offers an item (or a set of items) to entity $e_{i \boxplus 1}$, where $\boxplus$ and $\boxminus$ respectively denote addition and subtraction modulo n. Another topology is the more general matrix topology, where each entity may desire items from a set of entities and offer items to a set of entities. In fact a ring topology is a special case of the matrix topology: if the cardinality of the sets of the entities offering and requiring items equals one, i.e. each entity offers items to exactly one entity and receives items from exactly one entity, we have one or several ring topologies. Such protocols have been proposed by Asokan et al. in Asokan et al., 1996 and Asokan et al., 1998.

A fundamental difference between non-repudiation and fair exchange is the fact that in non-repudiation protocols the originator does not expect an item from the recipient. The originator sends some data with a non-repudiation of origin evidence to a recipient, who has to respond with a non-repudiation of receipt evidence. The sent data is generally not known to the recipient a priori. In a fair exchange each entity offers an a priori known item and receives another item, also known a priori. In a multi-party fair exchange protocol one can imagine sending an item to one entity and receiving an item from a different one. In non-repudiation it does not make sense that one entity receives some data and a distinct entity sends the corresponding receipt. Thus a ring topology is not sound: one cannot have the case that an entity e_i sends a message to entity $e_{i \boxplus 1}$ and receives a non-repudiation token for the sent message from entity $e_{i \boxminus 1}$ if $n > 2$. The most natural and here considered generalization seems to be a one-to-many protocol, where one entity sends a message to $n-1$ receiving entities who respond to the sender. However other possibilities for generalization exist (many-to-one, many-to-many).

The expectations we have towards such a protocol are rather similar to the properties required in two-party non-repudiation. The first property to respect is non-repudiability. Therefore non-repudiation of origin must be guaranteed, as well as non-repudiation of receipt. This means that a party sending a message cannot deny being its originator—the recipient can prove the identity of the originator of the message to an external party—and the recipient cannot deny having receipt it—the originator can prove that the recipient actually received a given message. This property is generally assured using non-repudiation evidences that can be evaluated by an adjudicator.

A second property is fairness. One can distinguish between the two notions of *strong fairness* and *weak fairness.* A multi-party non-repudiation protocol is said to provide *strong fairness* if at the end of the protocol the sender has got a complete non-repudiation of receipt evidence for a given recipient if and only if this recipient has got the message with a complete corresponding non-repudiation of origin evidence. A multi-party non-repudiation protocol is said to provide *weak fairness* if either the protocol provides strong fairness, or the protocol provides evidences that can prove to an adjudicator until which state the protocol has been executed with each of the receivers.

Here we can clearly see the difference with the certified mail protocol proposed by Asokan et al. Their protocol requires that at the end of the protocol *all* receivers have got the message with corresponding origin evidence and that the sender has got a receipt for *every* receiver, or none of them gained any valuable information.

A last required property is the timeliness property: a protocol has to be completed before a finite amount of time if at least one of the parties is behaving correctly with respect to the protocol.

2. THE ZHOU-GOLLMAN PROTOCOL

In this section we present a non-repudiation protocol designed by Zhou and Gollman (Zhou and Gollmann, 1997), that we will generalize in the next section. The basic idea is to split the transaction into two parts. In a first time the originator sends a message ciphered under key k to the recipient, who responds by sending a receipt for this cipher. The second part consists in sending this key to a TTP, who signs and publishes it. The TTP's signature guarantees the key's origin and availability. Then, both the originator and the recipient can retrieve the signed key from a public directory, providing only read access, using the ftp get operation. Finally the recipient can decrypt the message using key k.

In the description of the protocol the following notation is used:

- O, R and TTP respectively denote the originator, the recipient and the trusted third party.
- $X \rightarrow Y$: transmission from entity X to entity Y.
- $X \leftrightarrow Y$: ftp get operation performed by X at Y.
- $S_X()$: the signature of entity X.
- $f_{\{Eoo,\ Eor,\ Sub,\ Con\}}$: a flag indicating the intention of the message: evidence of origin, evidence of receipt, submission of the deciphering key or confirmation of the submission of the deciphering key.
- l: a unique label $l = h(m, k)$ identifying the protocol run, where $h()$ denotes a one-way hash function.
- t: the submission deadline.
- c: the cipher resulting from symmetric-key encryption of message m under key k.
- $Eoo = S_O(f_{Eoo}, R, l, t, c)$: the evidence of origin for the cipher c.
- $Eor = S_R(f_{Eor}, O, l, t, c)$: the evidence of receipt for the cipher c.
- $Sub_k = S_O(f_{Sub}, R, l, t, k)$: the evidence of submission of the key k.
- $Con_k = S_{TTP}(f_{Con}, O, R, l, t, k)$: the evidence of confirmation of the key k.

$$\begin{array}{ll}
O \rightarrow R: & f_{Eoo}, R, l, t, c, Eoo \\
R \rightarrow O: & f_{Eor}, O, l, Eor \\
O \rightarrow TTP: & f_{Sub}, R, l, t, k, Sub_k \\
R \leftrightarrow TTP: & f_{Con}, O, R, l, k, Con_k \\
O \leftrightarrow TTP: & f_{Con}, O, R, l, k, Con_k
\end{array}$$

We shall now look at each step of the protocol in more detail. During the first message the originator sends the identity of the recipient, a unique label l, the deadline t as well as well as the cipher of the message. The label is used to link all messages of a protocol run. The deadline t precises the moment when the key will be published at the TTP. The cipher c, the encryption of message m under key k, is used as a commitment to m. The origin of the cipher is guaranteed by the originator's signature in the *Eoo* evidence.

As a second step, the recipient responds to the originator with the originator's identity and the label l. The *Eor* evidence contains the recipient's signature on the cipher and can thus be used as an evidence of receipt for c.

During the third step, the originator sends the key to the TTP. The TTP verifies the deadline t and, if it is respected, publishes the key in a public directory related to O, R and l. This directory only provides read access via ftp avoiding its content to be modified. O's signature of Sub_k guarantees the authenticity of the key to the TTP. Remark that if there is a risk of eavesdropping the key, this message must be encrypted. During the last two steps the recipient and the originator can get the key and the confirmation of submission signed by the TTP. The deadline assures that if the key is not published at time t, it will not be published at any time later.

The non-repudiation of origin evidence of the message is composed of the evidence of origin of the cipher (Eoo) and the confirmation of submission evidence of the key (Sub_k), proving the key's authenticity. The non-repudiation of receipt evidence is composed of the evidence of receipt of the cipher (Eor), as well as the confirmation of submission evidence of the key (Sub_k), proving the key's availability. The only hypothesis that has to be made on the underlying communication model is that the availability of the ftp get operation is not permanently broken. A more detailed discusion of the security of the protocol can be found in Zhou and Gollmann, 1997, also showing that the protocol provides strong fairness. Timeliness is trivially provided by the deadline and the assumption made on the availability of ftp get.

3. GROUP NON-REPUDIATION PROTOCOL

Now we shall give a description of the generalization of the previously discussed Zhou-Gollmann protocol to provide multi-party non-repudiation. First we have to introduce some additional notation:

- $X \Rightarrow \mathcal{E}$: multicast from entity X to the set of entities $\mathcal{E}$.
- $E_{\mathcal{E}}()$: a group encryption scheme E, that can be deciphered by each party $P \in \mathcal{E}$.
- $\mathcal{R}$: the set of receiving entities.
- $\mathcal{R}'$: the set of receiving entities having sent an evidence of receipt for the cipher to the originator.
- $Eoo = S_O(f_{Eoo}, R, l, t, c)$: the evidence of origin for the cipher c.
- $Eor_i = S_{R_i}(f_{Eor}, O, L, t, c)$: the evidence of receipt from the recipient R_i for the cipher c.
- $Sub_k = S_O(f_{Sub}, \mathcal{R}', l, t, E_{\mathcal{R}'}(k))$: the evidence of submission of the key k.
- $Con_k = S_{TTP}(f_{Con}, O, \mathcal{R}', l, t, E_{\mathcal{R}'}(k))$: the evidence of confirmation of tke key k.

$$\begin{array}{ll}
O \Rightarrow \mathcal{R}: & f_{Eoo}, \mathcal{R}, l, t, c, Eoo \\
R_i \rightarrow O: & f_{Eor}, O, R_i, l, Eor_i \text{ where } R_i \in \mathcal{R} \text{ and } i \in \{1, \ldots, |\mathcal{R}|\} \\
O \rightarrow TTP: & f_{Sub}, \mathcal{R}', l, t, E_{\mathcal{R}'}(k), Sub_k \\
R_i' \leftrightarrow TTP: & f_{Con}, O, \mathcal{R}', l, E_{\mathcal{R}'}(k), Con_k \text{ where } R_i' \in \mathcal{R}' \ \forall i: 1 \leq i \leq |\mathcal{R}'|
\end{array}$$

$O \leftrightarrow TTP: \ f_{Con}, O, \mathcal{R}', l, E_{\mathcal{R}'}(k), Con_k$

The idea of the protocol is closely related to the Zhou-Gollmann protocol. In a first phase the originator sends the cipher of the message to each of the receivers. Then the receivers (or possibly a subset of receivers) respond with an evidence of receipt. The set of receivers having done so is denoted as $\mathcal{R}'$. Note that the originator decides himself a suitable time to start the second phase. Every receipt arriving after the second phase has been initiated, is not considered and will, as we shall see, bring no valuable information to the originator. The second phase consists in sending the decryption key to the TTP. If there is a risk that the third message could be intercepted, it must be ciphered with a key shared only by the originator and the TTP. The originator can always decide to stop the protocol before starting the second phase if not all of the recipients have answered. In that case it provides the same service as the certified mail protocol in Asokan et al., 1998.

In a two-party protocol we do not have to worry about external adversaries. However in the multi-party case the situation is different: the key k should only be available to the recipients included in $\mathcal{R}'$, the recipients having sent an evidence of receipt for the encrypted message. If a receiver $R \in \mathcal{R} \backslash \mathcal{R}'$ could get the key at the TTP, he would possess a complete non-repudiation of origin evidence. However the originator does not have the evidence of receipt for the cipher from this recipient and thus does not possess a complete non-repudiation of receipt evidence. Hence, we have a violation of our fairness requirement. That is why we have to limit the access to the key. Just limiting the TTP access, for instance, by restricting the access permissions of the directory containing the key, is not an acceptable solution as the information could be obtained by eavesdropping, while another recipient is getting the key. The proposed solution is to use a public-key group encryption scheme. The idea is that the key can be ciphered in a way that only recipients $R_i' \in \mathcal{R}'$ can decipher it, but any entity can verify wether a given key k has been ciphered for a recipient R_i or not. A concrete example of such a schemes will be given in the following section.

Before the submission of $E_{\mathcal{R}'}(k)$ to the TTP by O, none of the involved parties has got any complete evidence of origin or receipt. This is due to the fact that non-repudiation of origin (receipt) evidences are composed of the *Eoo* (*Eor*) evidences and of the confirmation of submission evidence issued by the TTP. As the TTP makes this last evidence available to the orginator and the recipients at the same time t, neither the originator nor any of the recipients can get a complete evidence without the other also doing so. Hence, this protocol provides strong fairness. Timeliness can be shown using exactly the same arguments as in the Zhou-Gollman protocol. The non-repudiability is assured by the signature included in each of the protocol messages. Assuming that signatures cannot be forged, the non-repudiation evidences provide irrefutable proofs of the origin and the receipt of the message.

There is a more delicate case that should be discussed in detail. Consider a scenario where at the second step several receivers send the evidence of receipt. The originator will start the second phase and contact the TTP. Now, if a group of late receipts arrive at the originator, he will possess an evidence of receipt for the cipher c from these receivers. However the evidence published by the TTP will contain the set of receivers able to decipher the key. As the non-repudiation of receipt evidence is composed by both the receipt of the cipher and the receipt of the key, provided by the TTP, it is only complete if the recipient originating the cipher receipt is included in the confirmation of the key. Thus the too late arrived evidences of receipt do not provide valuable information to the originator.

Two kinds of disputes can arise: repudiation of origin and repudiation of receipt. Repudiation of origin arises when a recipient R_i claims having received a message m from an originator O, who denies having sent it. The recipient then has to provide Eoo, Con_k and (m, c, k, l). The message m and the key k have to be sent to the judge via a secure channel, for example using encryption. Otherwise a recipient $R \in \mathcal{R} \backslash \mathcal{R}'$ can recover the message. If any of these informations cannot be provided the recipient's claim is rejected. The judge validates the recipient's claim if he can successfully verify O's signature on Eoo, the TTP's signature on Con_k, that c and l match the Eoo and the Con_k, that $l = h(m, k)$, and that c corresponds to the cipher of m under key k.

Repudiation of receipt arises when the originator O claims having sent a message m to a recipient R_i who denies having received it. In order for the claim to be accepted the originator must provide Eor_i, Con_k and (m, c, k, l) to a judge. As above a secure channel is needed to transmit m and c. The judge validates the claim if he can succesfully verify R_i's signature on Eor_i, the TTP's signature on Con_k, that c and l match the Eor_i and the Con_k, that $R_i \in \mathcal{R}'$, that k has been ciphered for R_i in the cipher provided by Con_k, that $l = h(m, k)$, and that c corresponds to the cipher of m under key k.

4. GROUP ENCRYPTION SCHEMES

Several public-key group oriented cryptosystems exist, but the right choice is crucial to the security of the protocol. Boyd presented a generalization of RSA in Boyd, 1988. In a group of n entities n keys are needed: each entity i knows $n-1$ keys; he knows all keys behalve the i^{th} one. Encryption is possible for each subset of entities. The problem with this scheme is that it is 1-resilient: when two entities collude they can decipher every message as together they possess all keys. Hence this cryptosystem is not suitable.

We shall now present the scheme proposed in Chiou and Chen, 1989 and show how it can be used to instantiate our protocol. It is based on a public-key encryption scheme and on the chinese remainder theorem. This method is generic as it can use any secure public-key cryptosystem. Let n be the number of recipients ($n = |\mathcal{R}|$) and m be the number of recipients having sent a receipt for the initial cipher ($m = |\mathcal{R}'|$). We shall denote the public-key encryption operation as $E_{e_i}()$ and the decryption operation as $D_{d_i}()$, where e_i and d_i respectively denote the public and the private key of recipient i ($1 \leq i \leq n$). Each recipient also chooses a large integer N_i such that all N_i are pairwise relatively prime and $N_i > E_{e_i}(k)$. The public values for each recipient are e_i and N_i; the private value is d_i. To cipher the key k for the m recipients (indexed form j_1 to j_m) having sent an evidence of receipt for the cipher, the originator computes $C_{j_i} = E_{e_{j_i}}(k)$ ($1 \leq i \leq m$). Then he uses the chinese remainder theorem to find a solution to the following system: $X \equiv C_{j_i} \pmod{N_{j_i}}$ ($1 \leq i \leq m$). As all N_i are pairwise relatively prime we are sure that a unique solution exists. We now have that $E_{\mathcal{R}'}(k) = X$.
Each recipient j_i ($1 \leq i \leq m$) can decrypt $E_{\mathcal{R}'}(k)$ by computing $C_{j_i} = E_{\mathcal{R}'}(k) \mod N_{j_i}$ and $D_{d_{j_i}}(C_{j_i}) = D_{d_{j_i}}(E_{e_{j_i}}(k)) = k$.

When the judge needs to verify that a key k has been ciphered for R_i in $E_{\mathcal{R}'}(k)$ he just computes $E_{e_i}(k)$ and checks that $E_{\mathcal{R}'}(k) \mod N_i = E_{e_i}(k)$. So the judge can efficiently perform this verification without the need to completely recompute $E_{\mathcal{R}'}(k)$.

5. CONCLUSION AND FUTURE WORKS

In this paper we have defined multi-party non-repudiation. This is the first effort to generalize non-repudiation to the multi-party case. We have also shown the differences that exist between multi-party non-repudiation and multi-party fair exchange. Above all the communication topologies of multi-party fair exchange are different to those used in multi-party non-repudiation. The most closely related work that has been realised, is a multi-party certified mail protocol. However our definitions are more general.

The here presented protocol is a generalization of the Zhou-Gollman protocol. For this purpose we use a public-key group encryption scheme.

We have given Chiou and Chen's group encryption scheme as an example. This scheme has the advantage of providing an efficient way for the judge to perform his verifications. Moreover, using the idea of group encryption, other protocols can be generalized. Many protocols first send a cipher as a commitment and then send the corresponding key. Using the idea of ciphering this key for a given group makes the generalization rather straightforward. Above all it would be interesting to generalize a protocol based on the optimistic approach.

References

Asokan, N. (1998). *Fairness in Electronic Commerce.* PhD thesis, University of Waterloo.

Asokan, N., Baum-Waidner, B., Schunter, M., and Waidner, M. (1998). Optimistic synchronous multi-party contract signing. Research Report RZ 3089, IBM Research Division.

Asokan, N., Schunter, M., and Waidner, M. (1996). Optimistic protocols for multi-party fair exchange. Research Report RZ 2892 (# 90840), IBM Research.

Bao, F., Deng, R., Nguyen, K. Q., and Vardharajan, V. (1999). Multi-party fair exchange with an off-line trusted neutral party. In *DEXA'99 Workshop on Electronic Commerce and Security*, Florence, Italy.

Ben-Or, M., Goldreich, O., Micali, S., and Rivest, R. L. (1990). A fair protocol for signing contracts. *IEEE Transactions on Information Theory*, 36(1):40–46.

Boyd, C. (1988). Some applications of multiple key ciphers. In *Advances in Cryptology—EUROCRYPT 88*, volume 330 of *Lecture Notes in Computer Science*, pages 455–467, Davos, Switzerland. Springer-Verlag.

Chiou, G. and Chen, W. (1989). Secure broadcasting using the secure lock. *IEEE Transactions on Software Engineering*, 15(8):929–934.

Franklin, M. and Tsudik, G. (1998). Secure group barter: Multi-party fair exchange with semi-trusted neutral parties. *Lecture Notes in Computer Science*, 1465.

Markowitch, O. and Roggeman, Y. (1999). Probabilistic non-repudiation without trusted third party. In *Second Conference on Security in Communication Networks'99*, Amalfi, Italy.

Zhou, J. and Gollmann, D. (1997). An efficient non-repudiation protocol. In *Proceedings of The 10th Computer Security Foundations Workshop.* IEEE Computer Society Press.