Non-repudiation mechanisms for Peer-to-Peer networks

Enabling technology for peer-to-peer economic markets

Michael Conrad
Institute for Telematics
Universität Karlsruhe (TH), Germany
conrad@tm.uka.de

ABSTRACT

Recent peer-to-peer applications focus on end-to-end transport security. However, with future applications like distributed market places on the rise, it is likely, that the security focus will shift to other security mechanisms. In distributed market places, non-repudiation of contract data is an issue to allow economic transactions. This paper presents a non-repudiation protocol, which also satisfies requirements emerged by the application in a peer-to-peer network.

Keywords

non-repudiation, proof of reception, peer-to-peer networks

1. INTRODUCTION

Recently, peer-to-peer technologies are widely distributed in file sharing and instant messaging applications, but peer-to-peer could be an enabling technology for commercial platforms e.g. distributed market places. Offering better scalability and better robustness then classical client/server-based technology combined with lower transaction costs are great advantages of peer-to-peer technology.

But most peer-to-peer technology has one disadvantage today, only a few necessary security requirements, like end-to-end transport security, are available. To be fully qualified for commercial market platforms additional security requirements like authenticity of members or prove of transactions are crucial.

This paper presents a novel non-repudiation protocol for proof of reception. Proof of reception is a key element for providing secure contract conclusion between members on a market place. The key principle is the involvement of other peers. These peers act as witnesses (see figure 1) and assist the non-repudiation protocol operations. In summary, the set of witnesses acts as a replacement for the trusted third party known from classical non-repudiation protocols [1]. In figure 1 peers W_1, W_2 and W_3 are selected as witness peers. These peers assist the proof of reception protocol between the peers A and B.

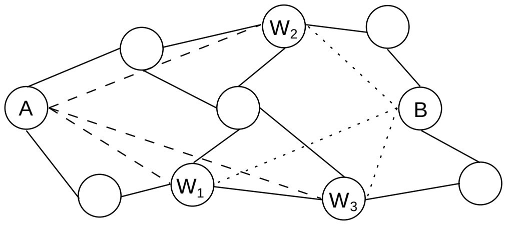

Figure 1: Involving other peers into the protocol

The paper is structured as follows. In section 2 requirements for a non-repudiation protocols which is suitable for peer-to-peer networks are stated. Section 3 gives an overview over existing protocols. Thereafter we present a short treat analysis in section 4 followed by the design of our non-repudiation protocol in section 5. Section 6 concludes our work.

2. REQUIREMENTS

We identified the following requirements for a non-repudiation protocol for a peer-to-peer environment:

- *Distributed control (R1):*
  Distribution of control is one of the core principles of peer-to-peer networks. A peer-to-peer non-repudiation protocol should follow this principle to avoid introducing a single point of failure into the peer-to-peer network.
- *Data confidentiality (R2):*
  If other peers are involved, the confidentiality of the data exchanged between sender and recipient have to be guaranteed.
- *Non-repudiation of reception for recipient (R3):*
  For a secure non-repudiation protocol it is crucial, that the reception can not be denied by the recipient.
- *Non-repudiation of content for sender (R4):*
  Non-repudiation of content is another requirement. This is necessary to avoid, that the sender can falsify the delivery of another document.
- *Robust selection of involved parties (R5):*
  If additional peers are involved in the non-repudiation protocol, the selection of such peers must be traceable, deterministic and not influenceable by the participants.

3. RELATED WORK

[1] gives an overview of existing non-repudiation protocols. Fundamental mechanisms are described in [2]. Most of them rely on a third party, which is trusted by all involved parties. Such a party is likely to not exist in in a peer-to-peer network, because it conflicts with the distributed control principle of peer-to-peer networks, hence, these protocols are not applicable in a peer-to-peer environment and the requirement R1 can not be fulfilled.

4. THREAT ANALYSIS

In our threat analysis we pay attention to following attacks:

- *Denial by recipient (A1):*
  Recipient B could deny the reception of a dedicated document O from sender A.
- *Fraud by sender (A2):*
  Sender A tries to falsify the reception of another document by recipient B.
- *Witness peer selection (A3):*
  One of the participants tries to precompute the witness peer set and place malicious peers as witness peers, which manipulate the protocol.

5. DESIGN

Our design consists of two parts. The first one is the witness selection, which provide a robust selection of additional peers. These peers assist the communication protocol for the proof of reception, which is the second element of our design.

In this section we are using following notations: $H(Z)$ is the hash value of Z, $K(Z)$ is the symmetric encryption with key K, $S_X(Z)$ is a signature of X, $E_X(Z)$ is the encryption with public key of X and $D_X(Z)$ is the decryption with the private key of X.

5.1 Witness selection

Without a central trusted third party other mechanism are required to enable non-repudiation protocols. One possibility is the involvement of other available peers, which assist the protocol between sender and recipient and acting as witness. The key problem is the selection of witness peers, if one of the participants is able to compute the witness peer set using a brute-force attack, he is able to place malicious peers and manipulate the non-repudiation protocol.
Therefore, both, sender and recipient, must be involved into the witness peer selection. To ensure this, the sender A requests a nonce value by sending a signed nonce request $S_A(H(H(O)), N_A)$ to the recipient B, including the hashed hash value of O and the nonce value N_A of A. B answers with a signed nonce response $S_B(S_A(H(H(O)), N_A), N_B)$ containing the original request and the nonce value N_B of B. After exchanging nonce values, sender A computes the set of witness peers using following formula.

$$Peer\ ID_{P_i} = H(i, S_B(S_A(H(O), N_A)N_B))$$

While the calculation of witness peers depends on the nonce value of A and B, none of them is able to place malicious peers before starting proof of reception protocol. Selecting witness peers using the described algorithm, the requirement R5 can be fulfilled and attack A3 can be avoided.

5.2 Communication protocol

Figure 2 shows the protocol for a proof of reception of document O of recipient B started by sender A. For simplification the figure only shows one witness peer P_i.

At the beginning A sends two messages (Req_A, Key_A) to each witness peer P_i. Each Peer P_i requests a reception confirmation of the encrypted document O by forwarding Req_A to B. If B answers with a valid Res_B the witness peer P_i sends Key_A to recipient B. With Key_A the recipient B is able to decrypt $K(O)$ extracting K from $E_B(K)$. After delivering Key_A to the recipient B, each peer P_i returns the reception confirmation Res_B to the sender A. The protocol works also correctly, if only one of the witness peers follows this procedure. Using n witness peers, the protocol requires $6n$ messages, 1 symmetric and 5 asymmetric operations at A and B, and 3 signature checks on each witness peer.

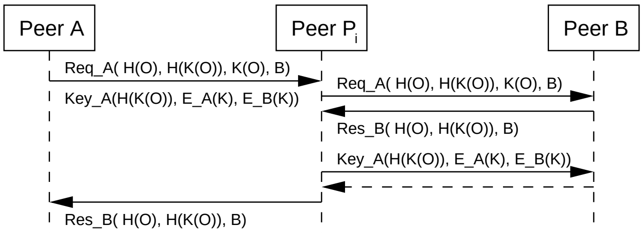
Peer A
Peer P_i
Peer B
Req_A(H(O), H(K(O)), K(O), B)
Key_A(H(K(O)), E_A(K), E_B(K))
Req_A(H(O), H(K(O)), K(O), B)
Res_B(H(O), H(K(O)), B)
Key_A(H(K(O)), E_A(K), E_B(K))
Res_B(H(O), H(K(O)), B)

Figure 2: Communication protocol

By the integration of witness peers, there is no need for a global trusted third party. For each proof of reception other witness peers are selected. Combining these two features, requirement R1 can be fulfilled. To meet requirement R2 the delivered document is encrypted by sender A and can only be decrypted by the recipient B. The requirement R3 can be fulfilled (and attack A1 can be avoided), because the recipient has to sign a reception confirmation before getting the key K to decrypt the document O. Each witness peer P_i returns this confirmation to the sender A, which is able to prove the reception confirmation. The same procedure helps to meet requirement R4. By applying a signature under the reception request and key message from sender A, recipient B is able to validate the content of document O. Thereby sender A is unable to falsify the reception of another document O' and attack A2 is unsuccessful.

6. CONCLUSION

In this paper we presented the current status of a non-repudiation protocol designed for peer-to-peer networks. Due the lack of a trusted third party, we propose the involvement of other peers into the evidence process.
Our design archives non-repudiation of the reception for the recipient, avoid fraud by the sender and provide a robust selection of witness peers, which acting as a replacement for the trusted third party, known from existing protocols.

7. REFERENCES

[1] S. Kremer, O. Markowitch, and J. Zhou. An intensive survey of non-repudiation protocols. Technical Report 473, 2002.

[2] P. Louridas. Some guidelines for non-repudiation protocols. *SIGCOMM Comput. Commun. Rev.*, 30(5):29–38, 2000.