On Applications of Cooperative Security in Distributed Networks

Dmitriy Kuptsov[1], Oscar Garcia-Morchon[2], Klaus Wehrle[3] and Andrei Gurtov[1,4]

[1] Helsinki Institute for Information Technology, Aalto University, Finland
{dmitriy.kuptsov,gurtov}@hiit.fi
[2] Philips Research Europe, Distributed Sensor Systems, Eindhoven, The Netherlands
oscar.garcia@philips.com
[3] RWTH Aachen University, Distributed Systems Group, Aachen, Germany
klaus.wehrle@cs.rwth-aachen.de
[4] Center for Wireless Communications, University of Oulu, Finland

Abstract. Many applications running on the Internet operate in fully or semi-distributed fashion including P2P networks or social networks. Distributed applications exhibit many advantages over classical client-server models regarding scalability, fault tolerance, and cost. Unfortunately, the distributed system operation also brings many security threats along that challenge their performance and reliability. In particular, faulty or misbehaving nodes cannot collude to subvert the system operation.
This paper addresses the above threats by applying cooperative security techniques to relevant distributed systems in the Internet. Our goal is to present methods that allow the peers to bootstrap basic trust relationships at the time of joining a distributed network and remove the peers if trust is lost. We consider the specific security caveats of the analyzed systems, investigate the applicability of existing cooperative security-based protocols, and propose general design guidelines for cooperative-security protocol in described distributed systems.

1 Introduction

Distributed systems have several advantages over a classical client-server model regarding scalability, fault tolerance, reliability, and cost efficiency. In contrast with centralized networks, where the network can be governed by a *trusted third party (TTP)*, a central entity which is trusted network wide, the nodes in distributed networks are operating autonomously, and therefore, equally responsible for different types of functionality, such as routing, node admission, or revocation. Security assertions are specially challenging: first, they are made in a collaborative way; second, they are constrained by compromised nodes that might collude subverting the correct operation of the whole network; finally, due to lack of trustworthiness between nodes in the network. We are interested in the design of self-organizing and self-healing networks. Networks that are capable to control the admission of new nodes, hence bootstrapping initial trust relationships between nodes. Second, cryptographically remove the compromised nodes after reaching the consensus when the trust towards misbehaving node was lost.

This paper analyzes the above security issues in three types of distributed networks, namely self-managed P2P networks [10, 7, 5], managed P2P networks [1], and web services [19, 21], and proposes the application of the concept of cooperative security [4] to deal with those security threats. We present the existing opportunities and challenges for the design of fully distributed cooperative security protocols for decentralized networks such as P2P networks. Moreover, we analyze the underlying advantages that those protocols would bring along to manage trust relationships between nodes in a distributed fashion.

The rest of the paper is organized as follows. First, we introduce the principals of cooperative security-based protocols in Section 2. Second, Section 3 analyzes relevant distributed applications that can benefit from cooperative security-based protocols and present some design concepts. Third, in Section 4 we introduce the state-of-art approaches targeting to solve the problem of node admission and revocation in decentralized networks. Finally, Section 5 concludes the paper and summarizes future research directions.

2 Cooperative Security Overview

The concept of *Cooperative Security* [4] has been applied to *distributed sensor networks (DSN)* comprising resource-constrained wireless sensor nodes. This approach allows a network to decide on the admission and revocation of the nodes in a distributed and cooperative way according to three fundamental directives.

First, in a cooperative-based protocol each node deployed in the network carries its own *partial revocation information (PRVs)*, i.e., information that can be used to build a revocation message against the node. A single PRV does not allow to revoke the node, but a set of PRVs does.

Second, a node must disclose PRVs of its own revocation information to its neighbors to be able to join the network. A group of nodes receiving the revocation information of a node is called the *dynamic trusted security domain, or DTSD* of the node. Here, if a node refuses to disclose the information upon joining the network, it is not accepted by the network. Thus, it loses the connectivity within the network and cannot endanger the rest of individuals. To this end, the network decides whether enough PRVs has been distributed to the node's neighbors. This fulfills the requirement that only cooperation of the minimum number of nodes makes the revocation possible.

Finally, the DTSD cooperates to monitor and revoke misbehaving nodes. If a node is detected to behave in suspicious manner, the members of its DTSD exchange the pieces of information, reconstruct a network wide revocation vote and eventually revoke the node in the whole network.

3 Cooperative-security in large-scale distributed networks

The operation of distributed networks in the Internet – such as managed P2P networks, non-managed P2P networks, and web services – is fundamentally different to the distributed sensor networks. First, nodes do not suffer from the resource limitations. Second, the network deployment scenarios are different than those used in sensor networks having a central authority in charge of rolling out

Table 1. Design Requirements for Cooperative Security-based Protocols

Features and Challenges	DSN	Self-managed P2P	Managed P2P	Web Services
TTP availability	Semi-Online	No	Online	Online
Initial Node Admission	Centralized	Collaborative	Centralized	Centralized
Key material certification	TTP	Group of nodes	TTP	TTP
Revocation	Voting	Voting	Voting	Feedback
Scalability	1000s	Millions	Millions	Millions
Number of DTSD participants	A few	Tens	Tens	Tens
Cryptographic Capabilities	Symmetric	Asymmetric	Asymmetric	Asymmetric

the network. And third, the networks we overview are dynamic in nature, where nodes are subjects to arbitrary leaving and joining the network.

These differences pose new design requirements that existing cooperative-security protocols do not fulfill. For instance, the keying material structure in [4] is based on hash chains to fit the resource-constrained nature of sensor nodes. However, what is an advantage for sensor networks becomes a limitation regarding network scalability for the above applications.

In this context, we detail in each of the following three sub-sections the existing challenges, design considerations, and advantages when applying cooperative-security approaches to the three above distributed networks. Table 1 summarizes the design requirements for each of the considered networks.

3.1 Managed P2P applications

Application A class of distributed managed networks is characterized by the presence of TTP which allows the nodes to bootstrap the initial trust and distribute key material. This is in turn the only role of the TTP in the network, and otherwise, the nodes operate in completely distributed manner. The example can be Skype [1] network. Such scalable design, on the other hand, gives a possibility for various types of misbehavior. For example, attacker controlling compromised accounts or faulty software can freely disseminate spam messages, which may contain anything, including links to compromised web sites. The application of cooperative security can be seen, therefore, as a direct solution for isolating such misbehavior.

For instance, following the principals of cooperative security a node should reveal the information which will enable its own revocation before communicating with any other node. If the receiving party suspects that a user is misbehaving it can publicize an obtained piece of information to enforce, or chip in the process of, the node revocation. Based on the settings if the number of votes exceeds the threshold, the server that is controlling the login information can afford to isolate the misbehaving node for a predefined time period depending on the policies.

Overall, the advantage of resulting system will be ability to isolate the misbehaving nodes from the network and decrease the chance of violating the system policies. Moreover, the reconstructed revocation votes by means of nodes cooperation, which can be verified network wide, can be used to build more sophisticated security services. For instance, the received revocation vote against any other node can be used as basis for issuing a punishment for a misbeaved node.

Design considerations While the structure for the revocation key material can be directly taken from [4], the application of the cooperative security protocols to this use case requires some further refinements.

As managed P2P network can comprise millions of nodes, with new nodes registering, joining, and leaving the network frequently, the key material structure must be scalable, maintainable, and flexible. Talking about scalability, it is important that rekeying of one node does not lead to rekeying of any other nodes in the network. Thus, in the network, such as Skype [1], where the login server can play the TTP role, it is more practical that each node maintains it own, local, verification tree (similar to the one described in [4]) where the root node of the tree is signed with TTP private key. Such an approach allows to build more scalable system, where the scalability factor of the system depends not on the number of nodes (and hence the height of the tree) but on the number of revocation sessions per node.

Cooperative security-based protocols were first designed for wireless sensor networks which support broadcast at the link layer. However, the absence of the efficient broadcast possibility in P2P networks deployed in the Internet will require to reconsider the voting strategy.

Design ideas Managed P2P networks involve communication between application running on platforms with higher capabilities when compared to sensor nodes. Despite that the structure based on Merkle trees and hash chains in [4] is efficient it lacks scalability. Therefore, the advantage of higher computational resources should be utilized to design more flexible revocation key material for managed P2P applications. Here, the usage of public key-based approach together with the concept of the TTP can overcome these limitations. In this setting, the TTP can assign revocation information to the nodes signed with its private key. This will allow the peers to authenticate the PRVs by verifying the signature of TTP. As opposed to the design proposed in [4], efficient and scalable key material structure for cooperative security protocol for such P2P network can be achieved using polynomials (such as in Shamir secret sharing [15]) to generate the PRVs for each node. For instance, each peer A can own a polynomial $f(x)^s_A$ to be used at time s. The time s can be understood as a revocation session, as described in [4]. Thus, a node A needs to distribute a minimum number of polynomial shares $f(i)^s_A$ to the network in order to start communicating with other peers in the network.

3.2 Self-managed P2P applications

Application There exist many examples of dynamic P2P applications which may not have TTP. Such systems operate in a completely distributed manner and not rely on a central authority. Just to name few, these can be P2P file sharing [10], distributed file systems [7], publish-subscribe [8], multicast [6] and P2P SIP [5]. In contrary to centralized networks, where the nodes are governed by a centralized authority, the nodes in self-managed P2P networks are equally responsible for the decisions, including security assertions, made in collaborative way.

The list of attacks associated to misbehaving nodes in self-managed P2P networks can be extensive. For sake of brevity here we introduce just two examples. One type of attack on content poisoning [17] has been identified in the literature. The attack refers to the situation when a node intentionally advertises a corrupted resource, e.g. file, which is consumed by nodes in the network. Another common type of attack is unfair resource allocation [18], or *free-riding.* This attack relates to an excessive resource consumption by a particular node which is much higher than resources advertised for utilization by other members of the network.

Despite that proactive protection mechanisms against these and other attacks are present in the literature, little attention has been paid to protocols that allow for the secure exclusion of the bad nodes, and thus, prevent future instances of attacks. Thus, we think that cooperative security-based protocols can leverage the security of the system and enforce proper nodes operation. In addition to this, the resulting system will also gain an ability to securely admit nodes into the network.

Design considerations We know that these self-managed P2P networks operate similarly to managed P2P networks. On the other hand, node admission decisions and key material generation should be an effort of group of nodes because such networks deficit the centralized authority responsible for node enrollment.

The major difference from managed P2P networks, is that in self-managed networks the availability of the TTP is not guaranteed, and consequently there is no central entity responsible for key material generation and node admission. Particularly, this circumstance raises several additional research questions which require further consideration. For instance, how to securely produce a network-wide verifiable key material for new nodes in the network. Accordingly, preliminary node cooperation is mandatory such that nodes are able to admit new entities into the network, generate key material, and still be able to revoke these entities if a misbehavior detected. On the other hand, it is important to ensure that colluding attackers are not able to generate forged key material and revoke some honest node by impersonating such key material.

Design ideas To solve the problem of distributed key material generation the concepts of distributed certification [3, 2] or identity-based pairing signatures with distributed public key generator (PKG) [11, 9] schemes can be adopted. Regarding the second issue each node should participate in the process of key material generation. This will prevent other nodes from generating spoofed revocation key material. However, both problems are left as open questions in this paper.

3.3 Trust in on-line web services

Applications The quality and success of the users' interaction in on-line web services, such as eBay [20] and Huuto.net [19], and decentralized social networks [21], depends on the trust between entities. Naturally, during the network lifetime, users should be able to adapt the trust level to unfair entities.

To enforce the fairness and establish a trust relationship in such systems, a user carries some authentic information (i.e. PRVs) which will be later revealed to other participants. Later, users involved in any kind of interaction with a target user will possess this authentic information. Such information together with revocation history will be a warranty for establishing a certain level of trust. Moreover, if some unfair activity is suspected this information revealed to agree that certain fraction of users distrusted the target. Thus, as compared to existing solutions, such as rating, keying material in our approach allow the owner of the content to ensure that only those users that hold the authentic information can affect its trust. All this may serve better as a defense against cheating clients.

Design considerations When designing a cooperative security-based protocols for such applications the following issues should be considered. First, in on-line web services, the joining user and its DTSD participants may not have direct communication channels with each other. Thus, the voting strategy needs to be changed. Second, similarly to managed distributed systems the number of DTSD participants varies and, therefore, the precautions are similar to those as in Section 3.1. Third, to support the verifiable revocation history, key material must be flexible enough to (i) guarantee that the node cannot lie about previous revocation decisions, (ii) allow all nodes to verify each history record, and (iii) allow to separate and to aggregate revocation sessions into different levels of granularity.

Design ideas Limiting the lifetime of the revocation information to just one instance of misbehavior allows to provide a verifiable isolation decisions. However, a more flexible and scalable reputation system can be build when the revocation key material is represented as a *timed hierarchy.* For instance, the hierarchical key material bound to a time periods, e.g., day, week, month, etc., may allow to maintain a verifiable revocation history of the user. This can be favorable when the users can utilize past revocation decisions as aggregated community opinions to assess the trust level more accurately in future. Second, we mentioned above, that the voting strategy in such systems require some modifications. In this context, a feedback to a back-end system, which stores the ranking information, can be a suitable solution for the users to reach the consensus.

4 Related Work

A concept of threshold node revocation based on distributed certification, which uses the RSA cryptosystem for group members [3] as a building block, was introduced by Lesueur et al. [2]. The approach was designed for P2P resource sharing networks such as in [10], which in the context of this paper falls under category of self-managed and managed applications. The protocol allows revoking a node if the group successfully reconstructs the key and produces the valid signature against target node. However, the system have several disadvantages, for instance, a collusion of half of members of a group can generate an unverifiable signature, thus, preventing potentially malicious node from being revoked.

Clulow et al. [12] introduced a radical strategy for node revocation for sensor networks, a so called *suicide for common good* strategy. The approach uses the fact that no node, neither attacker, nor honest node, will risk their presence in the network. Therefore, any node can revoke any other node in the network, but then it will result its own revocation as well. The disadvantage of the approach, when applied to P2P networks, is that it may endanger the network with fast depletion of nodes when misbehavior detection algorithms yield false positives.

Cholez et al. [13] introduced an architecture based on peer reputation and remote accounts, in which users' accounts are mapped to corresponding reputation information and stored in a *distributed hash table (DHT)* [14]. In such system every node can obtain the reputation information about group member, and therefore, judge whether to trust or not to such node. Despite that the protocol was designed for P2P file sharing networks, a class of self-managed distributed applications, applying the protocol to web services can also leverage their security. Particular limitation of the approach refers to the fact that only nodes that are responsible for monitoring the behavior of a particular target node can adjust its reputation information.

5 Conclusions and Future Work

Trust establishment during node admission to the network and node revocation as a consequence of lost trust are two aspects of paramount importance for distributed networks and systems in the Internet where not isolated misbehaving nodes, users, or elements can cause tangible damage to the network, system, or community.

Our contributions are twofold. First, we overview three application domains that can benefit from cooperative-secure primitives and present their security challenges. We argue that in P2P networks, which we classify into managed and self-managed, the node admission, hence initial trust establishment, and revocation, i.e., a counter-measure to a lost trust to a peer, should be an effort of a group of nodes. Whereas, the correct network operation depends on the ability of the network to isolate the misbehaving nodes. As opposed, web-services maintain the reputation information coupled with the revocation history, and thus, it is needed to enforce users fairness and build strong trust relationship to ensure the trusted system operation. Second, we show how cooperative security-based protocols can help to leverage the security of various applications by isolating bad nodes from systems. Moreover, by identifying new design guidelines, we try to fit the protocol into these application domains which have their own requirements, limitations, and challenges.

This research work serves, therefore, as a springboard for deeper studies and the design of new protocols for the above networks in the Internet. And, as for future research, we are planning to elaborate on trust management in P2P networks and alternative designs for distributed key generation algorithms.

References

1. Baset, S. A., Schulzrinne, H. G.: *An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol*, in Proceedings of the 25th IEEE International Conference on

Computer Communications, pp 1–11, 2008

2. François, L., Ludovic, M., Valérie, V.: *A Distributed Certification System for Structured P2P Networks*, in Proceedings of the 2nd international conference on Autonomous Infrastructure, Management and Security, pp. 40–52, 2008
3. Boneh, D., Franklin, M.: *Efficient generation of shared RSA keys*, in Advances in Cryptology – CRYPTO 97, pp. 425–439, 1997
4. Garcia-Morchon, O., Baldus, H., Heer, T., Wehrle, K.: *Cooperative Security in Distributed Sensor Networks*, Proceedings of the 3rd International Conference on Collaborative Computing, pp. 96-105, 2007
5. Bryan, A.D., Lowekamp, B.B., Jennings, C.: *SOSIMPLE: A serverless, standard-based, P2P SIP communication system*, in Proceedings of the International Workshop on Advanced Architectures and Algorithms for Internet Delivery and Applications, 2005
6. Castro, M., Druschel, M., Kermarrec, A-M., Nandi, A., Rowstron, A., Singh, A.: *Splitstream: high-bandwidth multicast in cooperative environments*, in Proceedings of the 19th ACM Symposium on Operating Systems Principles, pp. 298–313, ACM Press, 2003.
7. Dabek, F., Kaashoek, F., Karger, D., Morris, R., Stoica, I.: *Wide-area cooperative storage with CFS*, in Proceedings of the 18th ACM Symposium on Operating Systems Principals, pp. 202–215, 2001
8. Gupta, A., Sahin, O., Agrawal, D., Abbadi, A.: *Content-based publish/subscribe over P2P networks*, in Proceedings of the ACM/IFIP/USENIX International Middleware Conference, pp. 254–273, 2004
9. Joonsang, B., Yuliang, Z.: *Identity-Based Threshold Signature Scheme from the Bilinear Pairings*, in Proceedings of the International Conference on Information Technology: Coding and Computing, pp. 124, 2004
10. Stoica, I., Morris, R., Karger, D., Kaashoek, F., Balakrishnan, H.: *Chord: A scalable peer-to-peer lookup service for Internet applications*, in Proceedings of the ACM SIGCOMM Conference, pp. 149–160, 2001
11. Kate, A., Goldberg, I.: *A Distributed Private-Key Generator for Identity-Based Cryptography*
12. Clulow, J., Moore., T.: *Suicide for the Common Good: a New Strategy for Credential Revocation in Self-Organizing Systems*, in SIGOPS Operating Systems Rev., pp. 18 – 21, 2006
13. Cholez, T., Chrisment, I., Festor, O.:*A Distributed and Adaptive Revocation Mechanism for P2P networks*, in Proceedings of the Seventh International Conference on Networking, pp. 290–295, 2008
14. Balakrishnan, H., Kaashoek, M.F., Karger, D, Morris, R., Stoica, I.: *Looking up data in P2P systems*, in Communications of the ACM, February 2003.
15. Shamir, A., *How to share a Secret*, in Proceedings of Communications of the ACM Volume 22, pp. 612–613, 1979
16. Merkle, R.: *Secrecy, authentication, and public key systems*, Ph.D. dissertation, Dept. of Electrical Engineering, Stanford Univ., 1979
17. Liang, J., Kumar, R.: *Pollution in P2P File Sharing Systems*, in Proceedings of IEEE INFOCOM, pp. 1174–1185, 2005
18. Adar, E., Huberman. B.: *Free riding on Gnutella*, `http://firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/article/view/792/701`, 2000
19. Huuto online marketplace, `http://huuto.net`
20. eBay online marketplace, `http://www.ebay.com`
21. Fusion: P2P social network `http://p2p-fusion.org/`