Brief Announcement: Distributed Trust Management and Revocation

Dmitriy Kuptsov‡ and
Andrei Gurtov
Network Research Group
Aalto University, HIIT
Espoo, Finland
{dmitriy.kuptsov,
gurtov}@hiit.fi

Oscar Garcia-Morchon‡
Distributed Sensor Systems,
Philips Research Europe,
Eindhoven, The Netherlands
oscar.garcia@philips.com

Klaus Wehrle
Distributed Systems Group,
RWTH Aachen University
Aachen, Germany
klaus.wehrle@cs.rwth-aachen.de

ABSTRACT

Fair node and network operation is a key to ensure the correct system operation. The problem arises when some nodes become compromised or faulty endangering the overall system. This is especially challenging in sensor networks because they are often deployed in hostile environments and have to endure both passive and active attacks. Therefore, a node should only communicate with trusted nodes, while non-trusted nodes should be removed from the system to prevent them from further disrupting its normal operation. To address such threats, we introduce the Efficient Cooperative Security (ECoSec) – a distributed and adaptive protocol that allows a network to control the admission and revocation of nodes in a cooperative and democratic way during two voting rounds. Whereas the contributions of the protocol to the family of cooperative security protocols are two fold. First, it introduces the use of polynomial-based votes showing that its operation, and in general, operation of cooperative security protocols, can endure up to 33% of misbehaving nodes. Second, the protocol applies correlated keying material structures to verify the node admission and node revocation voting procedures reducing the overall communication overhead.

Categories and Subject Descriptors: F.0 [**Theory**]: Miscellaneous; C.2 [**Computer-Communication Networks**]: Network Protocols

General Terms: Theory, Security, Reliability, Algorithms

Keywords: Distributed Protocols, Security, Trust Management

1. PROTOCOL DESCRIPTION

Based on the concept of Cooperative Security [1], ECoSec relies on a trusted party to configure each node in the sensor network with its *own* revocation keying material. Before a deployed node can start communicating with other nodes, it has to join the network by gaining a trust during the first admission voting round. To this end, each joining node reveals its revocation information to its neighbors in the form of verifiable *partial revocation votes (PRVs)*. These neighbors form the node's *Dynamic Trusted Security Domain (DTSD)*, if they confirm that the node disclosed its revocation information correctly during this initial voting. In this case, the joining node is admitted into the network by its DTSD, becoming fully operable. Otherwise it is not admitted so that it cannot endanger the system. The second voting round is triggered by the network to revoke the node if the node is found to be malicious. This is decided by the node's DTSD that can vote on its revocation from the network. Again, if sufficient number of members agree on its removal, a *revocation vote (RV)* is constructed by combining the PRVs disclosed by the node during the joining procedure, and thus, allowing for the removal of the faulty node in the whole network.

To enforce correct operation the protocol makes use of cryptographic keying material for identity authentication as well as the verification of the disclosed PRVs and reconstructed RVs. The keying material structure comprises polynomials and three hierarchically connected Merkle [2] trees. First, a top-level tree – *Global Non-Rekeying Verification Tree (GNRVT)* – is used to verify the identities of the n nodes in the network. Second, n subtrees G_ζ denoted by *Rekeying Verification Trees* (RVTs) – unique to each node in the network and bound to the corresponding n leaves of *GNRVT* – are used to verify the s communication sessions of the n sensors in the network. Here, a communication session is defined as a period of time during which node remains trusted by its DTSD members. Finally, each leaf node L_ζ^k in G_ζ stores the hash of a root element of the third, bottom-level, tree B_ζ^k concatenated with the hash of a RV value, or $f_\zeta^k(0)$, i.e., $L_\zeta^k = H(H(B_\zeta^k)\|H(f_\zeta^k(0))$, where H is a cryptographic hash function. In turn, each of the w leaf elements of tree B_ζ^k stores a double hash of a PRV, i.e., $H(H(PRV))$, such that each PRV is a polynomial share generated from a polynomial $f_\zeta^k(x)$ of degree t. The PRV, its hash, and its double hash are used to authenticate up to three voting instances within single communication session of a node. Note that the RV value $f_\zeta^k(0)$ can be recomputed from a set of at least $t+1$ PRVs [3].

Based on the above keying material structure, ECoSec manages the admission and revocation of nodes by means of two voting procedures during a communication session. The voting presented in this work (i) assumes availability of a failure-free broadcast channel, and (ii) uses direct disclosure of PRVs or their hashes as a type of verifiable broadcast voting.

Admission Voting Procedure. In order for a node to join the network, it has to disclose λ PRVs (shares of a polynomial) to each selected neighbor forming its DTSD that comprises a total of q nodes. The distribution of PRVs is done in a secure way, e.g., pairwise keys are used to secure the communication links such that each DTSD member learns only its λ PRVs. The nodes receiving those votes comprise the DTSD of the joining node. Each member can verify the votes by means of the Merkle tree paths and the common root known to all nodes. In a second step, all the DTSD members cooperate to find out whether sufficient information has

‡Joint work

PODC'10, July 25–28, 2010, Zurich, Switzerland.
ACM 978-1-60558-888-9/10/07.

been disclosed by the DTSD owner. To this end, DTSD members vote by broadcasting the double hash of the received λ PRVs. Since the votes are verifiable, each DTSD member (i) counts the number of disclosed PRVs, and (ii) if the number is sufficient, i.e., at least $2t + 1$, each DTSD member shall trust the DTSD owner and admit it into the network. Note that if the node does not disclose enough information, it cannot join, and thus, it cannot endanger the network. If it does disclose enough information, it becomes trusted and joins the network. On the other hand, the protocol ensures that every admitted node have disclosed enough information, such that it can be later removed if it is found to be malicious.

Revocation Voting Procedure. After node admission, the DTSD monitors the operation of a node by means of *intruder detection system (IDS)*, which, in this paper, we assume to be ideal or faultless IDS: if any honest node detects the misbehavior, then all other honest nodes will do so as well. To this end, if the node is detected to be corrupted, the DTSD starts a revocation procedure. During this phase the nodes broadcast the PRVs previously received from the DTSD owner. It follows that the revocation succeeds if a sufficient number of PRVs is collected. Later these PRVs allow each DTSD member, individually, to reconstruct the secret $f_{\varsigma}^{k}(0)$, i.e., the RV, by polynomial interpolation. Such RV together with a corresponding Merkle tree path form a network-wide verifiable revocation message, which is then sent via broadcast to isolate the node from the whole network.

2. PROTOCOL ANALYSIS

ECoSec relies on a secret key sharing scheme based on polynomials of degree t in which each DTSD member receives λ PRVs. Thus, the system is secure under the collusion of up to $c = \lfloor t/\lambda \rfloor$ attackers because they cannot recompute the hidden secret. Given this maximum threshold for c,we have to analyze what is the minimum DTSD size q that ensures that c attackers cannot disrupt the system operation within a DTSD during the voting procedures, such that they (i) cannot admit another attacker, or (ii) prevent a good node from joining, or (iii) hinder the network from removing an intruder, or (iv) cannot remove an honest node. Theorem 1 analyzes this:

THEOREM 1. *A collusion of $c = \lfloor t/\lambda \rfloor$ intruders cannot subvert protocol operation if the DTSD comprises at least $q = \lfloor 3t/\lambda \rfloor + 1$ nodes and the underlying IDS operates faultlessly.*

To prove this, we show that above conditions (i), (ii), (iii), and (iv) hold. Note that there are at least $q - c \geq 2\lfloor t/\lambda \rfloor + 1$ honest nodes in the DTSD, and an attacker can only join by distributing authentic PRVs verified by means of the verification trees and the public root.

(i) The joining attacker has to disclose enough information, however, it can collude with up to c attackers. If these attackers within the DTSD disclose the received $H(H(PRV))$, then at least another $\lfloor t/\lambda \rfloor + 1$ honest nodes must do it as well to make sure the network has enough revocation information to reconstruct RV and revoke the new node in future. The disclosure of $2t+1$ $H(H(PRVs))$ by $\lfloor 2t/\lambda \rfloor + 1$ nodes is, therefore, mandatory: If the voting procedure does not confirm the reception of $2t + 1$ votes it can only mean that a joining attacker tries to fool the DTSD members by disclosing less PRVs to honest nodes than required.

(ii) If the compromised nodes within the DTSD try to prevent the node from joining, c nodes will not disclose their $H(H(PRVs))$ stating that they have not received them. We know from (i) that the DTSD must see at least $\lfloor 2t/\lambda \rfloor + 1$ nodes disclosing $H(H(PRV))$. Hence, the DTSD must comprise at least $q \geq c + (\lfloor 2t/\lambda \rfloor + 1) = \lfloor 3t/\lambda \rfloor + 1$ nodes. In this way, (at least) $\lfloor 2t/\lambda \rfloor + 1$ honest nodes will disclose their $H(H(PRV))$ or vote positively, and thus, the honest node will be allowed to join the network.

(iii) The c attackers may try to remove the honest node by disclosing all their t PRVs. However, as the IDS of the honest nodes does not trigger any alarm (it is faultless), then, the last and needed PRV will not be disclosed by any honest node. And eventually, the attackers will fail.

(iv) If honest nodes in a DTSD find a node to be an attacker, the IDS of all the nodes will trigger an alarm. Subsequently, all nodes will disclose in total (and at least) $2t+1$ distinct PRVs allowing for the reconstruction of the RV. As a result, this leads to a network-wide revocation.

COROLLARY 1. *ECoSec can endure up to $c = \lfloor t/\lambda \rfloor$ compromised nodes within a DTSD. From Theorem 1, the system operates correctly if $q \geq \lfloor 3t/\lambda \rfloor + 1$. The ratio between corrupted nodes and number of DTSD members is maximized when c is maximum and q is minimum. Thus, ECoSec can endure up to 33% and the optimal DTSD size is $q = \lfloor 3t/\lambda \rfloor + 1$ nodes*

The above results improve the ratio of endured compromised nodes within a DTSD when compared with [1]. Additionally, the overall approach presents some other advantages. First, the keying material structure allows reducing the communication overhead because the same Merkle tree is used for the verification of the votes during the admission voting procedure and node revocation during the second voting. Second, by varying the λ value, the protocol can adapt the operation regarding the maximum number of endured faulty nodes, the DTSD size, number of communication messages, and spent computational resources. For instance, to maximize the DTSD size and its security, each DTSD member receives exactly one PRV. On the other hand, it might occur that the node does not have the minimum number of required neighbors to ensure a secure protocol operation. In this case, the node would distribute more than one PRV to each DTSD member. Another advantage refers to the fact that ECoSec only requires a node to carry its own revocation information and not the revocation information for all other nodes in the network. Thus, ECoSec reduces the memory overhead by a factor of n with respect to the results presented in the related work [4].

3. CONCLUSIONS

In this work we have presented a preliminary description and analysis of ECoSec protocol regarding its operation, voting strategy, and thresholds. Assuming faultless IDS as a decision maker for triggering the revocation procedure our investigation shows that the protocol can endure up to 33% of faulty or compromised nodes and allows for adaptive operation. Our future work will focus on the design of advanced voting strategies when faulty IDS are involved and further analyzing the protocol complexity, including communication and computation overheads.

4. REFERENCES

[1] Garcia-Morchon, O., Baldus, H., Heer, T., Wehrle, K.: *Cooperative Security in Distributed Sensor Networks*,in Proceedings of the 2007 International Conference on Collaborative Computing: Networking, Applications and Worksharing (COLABORATECOMM '07), pp. 96–105, 2007

[2] Merkle, R.: *Secrecy, authentication, and public key systems*, Ph.D. dissertation, Dept. of Electrical Engineering, Stanford Univ., 1979

[3] Shamir, A., *How to share a Secret*, in Proceedings of Communications of the ACM Volume 22, pp. 612–613, 1979

[4] Chan, H., Gligor, V., Perrig, A., Muralidharan, G.: *On the Distribution and Revocation of Cryptographic Keys in Sensor Networks*, IEEE Transactions on Dependable and Secure Computing, pp. 233-247, 2005