André König, Matthias Hollick, Ralf Steinmetz:
A Stochastic Analysis of Secure Joint Decision Processes in Peer-to-Peer Systems. In: Mohammed Atiquzzaman, Marcus Brunner, Latif Latid, Ivica Rimac: Proceedings of the Third IEEE International Conference on Communications (ICC 2009), p. 1--5, June 2009. ISBN 978-1-4244-3435-0.

A Stochastic Analysis of Secure Joint Decision Processes in Peer-to-Peer Systems

André König, Matthias Hollick, Ralf Steinmetz
Multimedia Communications Lab (KOM), Technische Universität Darmstadt
{Andre.Koenig, Matthias.Hollick, Ralf.Steinmetz}@KOM.tu-darmstadt.de

***Abstract*—Central trusted instances as well as predefined security policies are not available in spontaneously established peer-to-peer environments. The former can be addressed by joint decision processes based on threshold cryptography. To compensate the latter, users can be involved directly in security-relevant decisions. In this case, minimizing the number of users involved is a necessary optimization goal to keep user-based joint decisions feasible for real-world deployment. Still, a certain redundancy has to be introduced when taking into account users that do not provide their decision in a reasonable amount of time. In this paper we scrutinize different interaction schemes for joint decision processes. We develop stochastic models that describe the outcome subject to the number of users requested and the probability with which one user provides his decision in time. The derived closed-form representation of the models serves as a tool for governing the decision process, allowing for a real-time minimization of the number of users involved.**

I. Introduction

Peer-to-peer (P2P) systems enable enhanced communication services in environments where client-server-based solutions can not be established due to e.g. time and/or cost constraints. Large scale emergency response scenarios are a prominent application domain. Here, the exchange of information and services between aid organizations can offer a considerable benefit for the coordination of search and rescue or of reconstruction efforts[1]. Yet, information and services should not be accessible in an unrestricted way. By means of authentication and access control, the admission to services and information can be governed. However, contemporary means for authentication and access control such as Kerberos [3] are based on central trusted instances. Thus, these mechanisms can not be transferred directly from the client-server domain to a P2P environment. In the absence of central trusted instances, security objectives such as authentication and access control can be implemented by threshold cryptography. Here, the cooperation of (at least) a certain number of peers is required to perform cryptographic operations. No single (possibly compromised) peer is able to e.g. sign and issue certificates that grant the access to restricted services. This way, a security level comparable to that of centralized solutions can be achieved.

The applicability and performance of threshold cryptography in P2P systems has been studied comprehensively. However, only little attention has been paid to the fact that (cooperative) security-relevant decisions require a well-defined set of regulations, shall they be performed automatically. In the application scenario outlined, any interaction desired can hardly be foreseen. Thus, the availability of predefined security policies can not be assumed. To deal with this, we consider the case of authorized users being involved directly in security-relevant decisions. In this case, performance issues of the network as well as of the threshold cryptography schemes deployed are negligible compared to the delay that is introduced by the users themselves. Rather, the number of users involved per decision has to be minimized to keep the approach feasible for real-world deployment. Nevertheless, the minimization has to take into account users that do not provide their decision in a reasonable amount of time. To deal with this, a certain redundancy has to be introduced regarding the number of users requested and the number of users that have to cooperate as specified by the threshold scheme deployed.

In the following, we describe how security-relevant decisions can be performed if neither a central trusted instance nor predefined security policies are available. We briefly identify variants of threshold cryptography that are able to deal with the resulting challenges. Having laid these basics, we describe different interaction schemes between the peer that requests a decision and the potential decision makers. We provide a stochastic analysis of the different interaction schemes. From this, we derive a closed-form description that allows for the real-time minimization of the number of users involved in a joint decision.

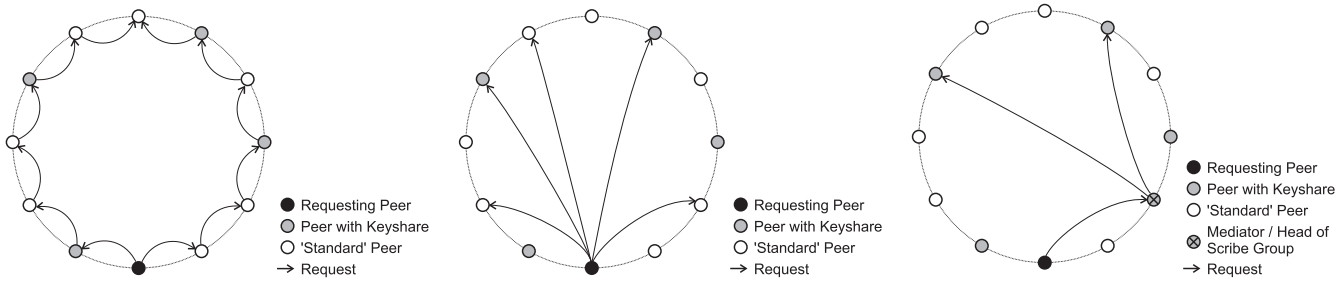
II. Related Work

In this section, we present related work that has motivated and influenced our research. We focus on basics of threshold cryptography and on studies on the performance of threshold cryptography in P2P environments.

The operation of threshold cryptography is based on shares of a secret key that are generated by choosing a polynomial $p(x)$ of degree n such that the shared secret key equals $p(0)$. The peers $Peer_1, ..., Peer_m$ receive the keyshares $p(1), ..., p(m)$ where $m > n$. With the keyshares, the peers are able to produce partial signatures. By Lagrange interpolation, a full signature can be computed from $n + 1$ partial signatures.

Most of the threshold cryptography schemes that have been proposed require that the set of all partial signers contributing to one full signature has to be known to each partial signer in advance. This results in multiple rounds of communication

[1]Related projects that apply P2P technology in large scale emergency response scenarios (but do not consider cooperative security mechanisms) are e.g. DUMBO [1] and SoKNOS [2].



(a) Broadcast - Unknown Signers (b) Multicast - Unknown Signers (c) Multicast - Known Signers

Fig. 1. Schematic representation of the different interaction schemes

required between the signers. A partial signature is valid only within the set of co-signers specified. In the context of our application scenario we have to consider devices that are connected wirelessly and, thus, may be subject to disconnections. What follows is that if one member of the set does not provide its partial signature in time and has to be replaced, all other partial signatures have to be discarded. The threshold scheme deployed should therefore be able to deal with signers that do not provide a signature without having to discard partial signatures that have been provided already. To make the system as reliable as possible, we build upon a threshold cryptography scheme that does not need any interaction between the signers involved. The scheme proposed in [4], as an enhancement of [5], meets this requirement.

In [6], the authors compare threshold signature schemes with respect to their performance in controlling access to closed user groups in P2P systems. Performance is measured in terms of basic operation costs (the time needed to produce partial signatures) and join time (the amount of time a new peer needs to join a closed user group). User interactions have not been considered. Non-interactive signature schemes were not part of the evaluation. The analysis of [6] has been extended for additional signature schemes in [7].

A non-interactive mechanism for access control in mobile ad hoc networks has been proposed in [8] and [9]. In contrast to [5], the protocol proposed in [8] and [9] is not based on a cryptographic key that is shared among multiple parties, but on bivariate polynomials that can be used to establish pairwise shared secret keys. A performance evaluation comparable to [6] and [7] has been performed. User interactions have not been considered. Because 'standard' signed certificates as required in our scenario can not be produced with this approach, we have not taken [8] into account as a possible cryptographic mechanism for our application scenario.

To the best of our knowledge, a stochastic model of user-based joint decision processes including closed-form descriptions for real-time deployment has not been proposed so far.

III. Interaction Schemes

A peer that requests a security-relevant decision has to send this request to a set of peers that are equipped with keyshares. Each of these may take part in granting access by issuing a partially signed certificate. The strategy according to which the requests are disseminated within the P2P overlay directly affects the number of users requested and the probability to receive enough partially signed certificates to be able to interpolate a full signature. In the following we describe different interaction schemes between requesting peers and peers that contribute to the signature process. We discuss broadcast and multicast approaches. We also take into account different levels of knowledge about which peers are equipped with keyshares. We demonstrate the interaction schemes with a Pastry P2P overlay [10] since we plan to verify the models in a Pastry-based testbed. Yet, the interaction schemes as proposed in the following are independent from the overlay and hold for structured as well as for unstructured or hybrid P2P systems.

A. Broadcast - Unknown Signers

We assume that the requesting peer has no knowledge about which peers in the network are equipped with keyshares and would potentially answer a request. In this case, as shown in Figure 1(a), a simple (but expensive) strategy to request a joint decision is to send a broadcast initiated by the requesting peer which is disseminated in the entire P2P overlay. This approach is shown schematically in Figure 1(a).

In the Pastry topology, the broadcast can be realized e.g. by forwarding messages to direct neighbors in the ID space or to the complete leaf set to make the process more stable.

B. Multicast - Unknown Signers

Due to the high number of users involved that is caused by a broadcast in the P2P overlay, the applicability of a broadcast is limited in our scenario. Instead, a multicast approach is reasonable. We still assume that the multicast is initiated by the requesting peer which has no knowledge about which peers are equipped with keyshares. In this case, as shown in Figure 1(b), a multicast can be realized by sending requests to a set of IDs that are selected randomly (if an ID is not used, Pastry will route the request to the ID closest to the one selected).

C. Multicast - Known Signers

In our application scenario we assume administrative restrictions that limit the number of peers which are authorized to take part in security-relevant decisions. Thus, a random selection of the peers to which a request is sent may not reach enough peers that are equipped with keyshares. To

TABLE I
NOTATIONS OF FORMULAE

n_{thres}	Number of partially signed certificates required to be able to compute a full signature
p_{rep}	Probability with which a single peer answers a request
n_{olay}	Total number of peers in the P2P overlay
n_{keys}	Number of peers equipped with keyshares
n_{ready}	Number of peers (users) ready to contribute to a decision
n_{mult}	Number of peers to which a request is sent
n_{rep}	Number of replies received for one request
$p(n_{rep})$	Probability for receiving n_{rep} replies from one request
$p_b(n_{rep})$	$p(n_{rep})$ for a broadcast with unknown signers
$p_{mu}(n_{rep})$	$p(n_{rep})$ for a multicast with unknown signers
$p_{mk}(n_{rep})$	$p(n_{rep})$ for a multicast with known signers
p_{succ}	Probability for receiving a sufficient (w.r.t. n_{thres}) number of replies from one request
$p_{succ\_b}(n_{thres})$	p_{succ} for a broadcast with unknown signers
$p_{succ\_mu}(n_{thres})$	p_{succ} for a multicast with unknown signers
$p_{succ\_mk}(n_{thres})$	p_{succ} for a multicast with known signers

increase reliability, it is reasonable to base the dissemination of requests on some knowledge about peers that are equipped with keyshares (and about the status of their users). This approach can be based e.g. on a peer that acts as a mediator for the decision process. The mediator keeps track of the potential signers and accepts and relays requests appropriately, as shown in Figure 1(c).

While introducing a mediator can be done independently from the particular P2P overlay, a way to implement a multicast with knowledge about potential signers on top of a Pastry P2P overlay is to make use of Scribe multicast groups [11]. Peers that are equipped with keyshares (and with users ready to contribute to a joint decision) subscribe to a corresponding multicast group. Requests can be sent to this group along with the requested number of partial signatures.

IV. STOCHASTIC MODELS

In this section we develop stochastic models that describe the success probability p_{succ} of the different interaction schemes. A request is considered successful if a sufficient number of partially signed certificates is received such that a fully signed certificate can be interpolated. The number of partially signed certificates received is sufficient if it is greater or equal to the threshold n_{thres} that is defined by the threshold cryptography scheme deployed. Table I provides an overview on the notation we use in the following.

We provide graphical examples of the models using a scenario that consists of $n_{olay} = 500$ peers of which $n_{keys} = 25$ are equipped with keyshares. The probability p_{rep} that a peer that holds a keyshare provides its answer in a reasonable amount of time is assumed to be 50% for the broadcast and the multicast with unknown signers. For the multicast with known signers (and known status of their users), we assume $p_{rep} = 95\%$ (we account a loss of 5% for the network itself) and $n_{ready} = 13$ (which corresponds to $\lceil 50\% \rceil$ of n_{keys}). With these values given, we show the influence of n_{thres} and n_{mult} on the success probability p_{succ}. Please note that the values are chosen for visualization purposes such that the system operates in reasonable boundaries. The stochastic models themselves are independent from this particular instantiation. Yet, these parameters strongly affect the performance of cooperative decisions in terms of requests issued and users involved. In turn, in a real-world scenario where the parameters are defined by the system itself, our models can be used to choose the appropriate mode of interaction.

A. Broadcast - Unknown Signers

The broadcast with unknown signers can be modeled by a binomial random variable. The binomial random variable describes the outcome of repeated Bernoulli experiments each of which has a certain probability to be successful. Thus, in our context this distribution describes the probability $p_b(n_{rep})$ for receiving a certain amount n_{rep} of replies to a request for a decision. We obtain

$$p_b(n_{rep}) = \binom{n_{keys}}{n_{rep}} p_{rep}^{n_{rep}} (1 - p_{rep})^{n_{keys} - n_{rep}}$$

For a request to be successful it is not important to receive a certain number of replies but to receive *at least* enough partial signatures to compute a full signature. I.e., a request is successful if the amount n_{rep} of replies received is greater than or equal to n_{thres}. The success probability $p_{succ\_b}(n_{thres})$ with respect to n_{thres} thus can be described as the sum of the probabilities for receiving a certain amount n_{rep} of replies starting from n_{thres}. The upper bound is given by the number of peers equipped with keyshares n_{keys}. We get

$$p_{succ\_b}(n_{thres}) = p(n_{rep} \geq n_{thres}) = \sum_{n_{rep}=n_{thres}}^{n_{keys}} p_b(n_{rep})$$

Figure 2 shows how the success probability is influenced by the number of partial signatures required assuming $n_{olay} = 500$, $n_{keys} = 25$, and $p_{rep} = 0.5$. Since a broadcast reaches all potential signers, we only show the influence of n_{thres}. Intuitively, the probability $p_{succ\_b}(n_{thres})$ for receiving a sufficient number of replies decreases if the number of replies that are required to compute a fully signed certificate increases and all other parameters are fixed.

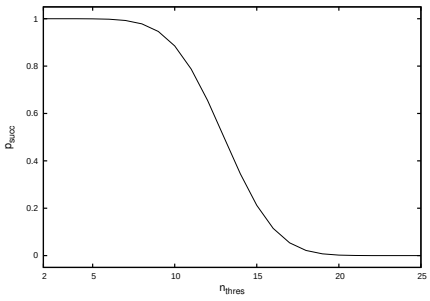


Fig. 2. Success probability $p_{succ\_b}(n_{thres})$

The broadcast with unknown signers can be considered as best-case benchmark regarding p_{succ} since the request is sent

to all potential signers. Due to the same reason it is the worst-case regarding the number of users involved. As an example, we generate 500 requests to involve 25 users (all peers that hold keyshares) to get a success probability of approximately 95% for a threshold of $n_{thres} = 8$.

B. Multicast - Unknown Signers

For the multicast scheme with unknown signers we assume a random limitation of requested peers with respect to the distribution of keyshares and to the status of peers. I.e., the scheme does not consider whether a peer to which a request is sent is allowed to answer or able to answer within an acceptable time frame. This random restriction can be modeled by a hypergeometric random variable. In our case, the hypergeometric variable describes the intersection of the set of peers to which a request is sent and the set of peers that would potentially reply to a request received. Thus, for the probability $p_{mu}(n_{rep})$ of receiving a certain amount n_{rep} of replies, we get

$$p_{mu}(n_{rep}) = \frac{\binom{n_{keys} \cdot p_{rep}}{n_{rep}} \binom{n_{olay} - (n_{keys} \cdot p_{rep})}{n_{mult} - n_{rep}}}{\binom{n_{olay}}{n_{mult}}}$$

As for the broadcast with unknown signers, a request is successful if the amount n_{rep} of answers received is greater than or equal to n_{thres}. The success probability $p_{succ\_mu}(n_{thres})$ thus is again the sum of the probabilities for receiving a certain amount n_{rep} of replies starting from n_{thres}. The upper bound of the sum is given by the number n_{mult} of peers to which a request is sent. We obtain

$$p_{succ\_mu}(n_{thres}) = p(n_{rep} \geq n_{thres}) = \sum_{n_{rep}=n_{thres}}^{n_{mult}} p_{mu}(n_{rep})$$

The resulting success probability subject to n_{mult} and n_{olay} is shown in Figure 3. As for the broadcast with unknown signers, the success probability decreases if n_{thres} increases (note that for reasons of presentation, the corresponding axis is inverted) and increases along with the number of users requested.

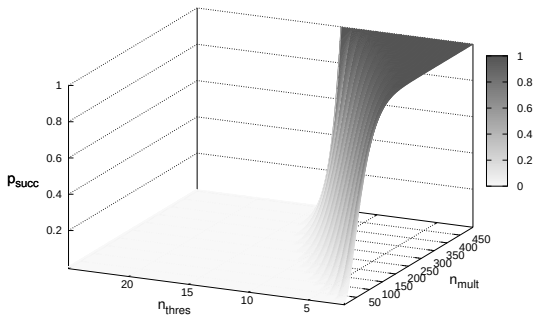


Fig. 3. Success probability $p_{succ\_mu}(n_{thres})$

For comparison with the broadcast with unknown signers, if we assume a threshold of 8, the requesting peer has to send 389 requests in order to 'hit' enough peers that are equipped with keyshares (and provide their reply in reasonable time) to reach a success probability of approximately 95%. If we assume an equal distribution of the peers that hold keyshares in the Pastry ID-space (i.e., every peer that holds a keyshare is 'hit' only once), this means that $\frac{389 \cdot 25}{500} \approx 20$ users have to be involved.

If n_{mult} is sufficiently small with respect to n_{olay}, the hypergeometric random variable can be approximated by a binomial random variable. Since our goal is to send a request to the least number of peers possible, the binomial approximation to the hypergeometric random variable is applicable for our needs. Figure 4 shows the resulting error of the success probability p_{succ} which is calculated as $err_{succ} = p_{succ\_b} - p_{succ\_mu}$. The approximation error is in low percentage range for reasonable values of n_{thres} and n_{mult} with respect to n_{olay}. Therefore, we represent the multicast with unknown signers by a binomial random variable. This allows us to combine the models for the broadcast with unknown signers and the multicast with unknown signers.

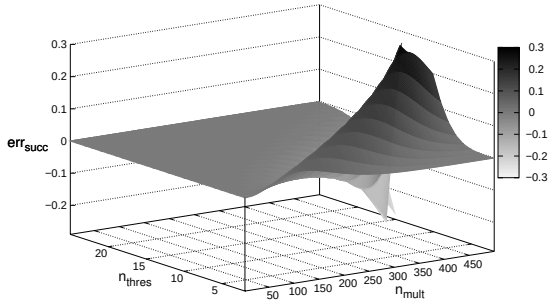


Fig. 4. Approximation error err_{succ}

C. Multicast - Known Signers

For the multicast with known signers, the probability $p_{mk}(n_{rep})$ of receiving a certain amount n_{rep} of replies to a request can (also) be modeled as a binomial random variable. In contrast to the broadcast with unknown signers, it is not parameterized by the total number of peers equipped with keyshares, but by the number of peers to which a request is sent n_{mult}. This results in

$$p_{mk}(n_{rep}) = \binom{n_{mult}}{n_{rep}} p_{rep}^{n_{rep}} (1 - p_{rep})^{n_{mult} - n_{rep}}$$

The success probability $p_{succ\_mk}(n_{thres})$ with respect to n_{thres} again can be described as the sum of the probabilities for receiving a certain amount n_{rep} of replies starting from n_{thres}. The upper bound of the sum is given by the number n_{mult} of peers to which a request is sent. We get

$$p_{succ\_mk}(n_{thres}) = p(n_{rep} \geq n_{thres}) = \sum_{n_{rep}=n_{thres}}^{n_{mult}} p(n_{rep})$$

The success probability subject to n_{thres} and n_{mult} is shown in Figure 5. As for the multicast with unknown signers,

the success probability increases along with the number of users requested and decreases if n_{thres} increases. 10 requests have to be generated to involve 10 users in order to obtain a success probability of more than 95% for a threshold of 8.

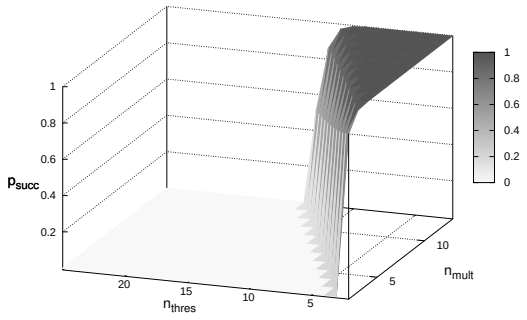


Fig. 5. Success probability $p_{succ\_mk}(n_{thres})$

D. Closed-Form Representation

We now derive a formula that provides a lower bound for the success probability p_{succ}. Since we have been able to describe all interaction schemes based on a binomial random variable with different parameters, the resulting formula is applicable for all interaction schemes discussed.

Let $p_{fail} = 1 - p_{succ} = p(n_{rep} \leq n_{thres} - 1)$ be the probability that our system fails (i.e., not enough partial signatures have been issued). We apply Chernoff's bound [12]

$$p(n_{rep} \leq n_{thres} - 1) \leq e^{(-\tau(n_{thres}-1))} M(\tau) \quad \forall \tau < 0$$

to obtain an upper bound for p_{fail}. The moment generating function $M(\tau)$ for n_{rep} is given as

$$M(\tau) = p_{rep} e^{\tau} + (1 - p_{rep})^{n_{mult}}$$

thus,

$$p_{fail} \leq e^{-\tau(n_{thres}-1)} (p_{rep} e^{\tau} + (1 - p_{rep}))^{n_{mult}}$$

To obtain an upper bound for p_{fail} (thus a lower bound for p_{succ}) we have to find the τ that minimizes the right hand side. Let

$$f(\tau) = e^{-\tau(n_{thres}-1)} (p_{rep} e^{\tau} + (1 - p_{rep}))^{n_{mult}}$$

thus

$$\begin{aligned}\frac{df}{d\tau}(\tau) =& e^{-\tau(n_{thres}-1)} n_{mult} (p_{rep} e^{\tau} + (1-p_{rep}))^{n_{mult}-1} p_{rep} e^{\tau} \\ & - n_{thres} e^{-\tau(n_{thres}-1)} (p_{rep} e^{\tau} + (1-p_{rep}))^{n_{mult}}\end{aligned}$$

By setting $\frac{df}{d\tau} = 0$ and resolving to τ we obtain

$$\tau = \ln \left(\frac{(n_{thres} - 1)(1 - p_{rep})}{n_{mult} p_{rep} - p_{rep}(n_{thres} - 1)} \right)$$

Thus, as a whole,

$$\begin{aligned}p_{succ} \geq 1 - & \left(\frac{n_{mult} p_{rep} - p_{rep}(n_{thres} - 1)}{(n_{thres} - 1)(1 - p_{rep})} \right)^{n_{thres-1}} \\ & \left(\frac{(n_{thres} - 1)(1 - p_{rep})}{n_{mult} - n_{thres} + 1} + (1 - p_{rep}) \right)^{n_{mult}}\end{aligned}$$

With this, we are able to adjust the system parameters during runtime. Assuming e.g. the amount of replies required to be able to compute a fully signed certificate is given. Furthermore, we assume that the probability that one peer answers within an acceptable time frame is known (measured) and changes during runtime of the system (but not within our control). In this case, the formula can be used to dynamically adjust the number of peers to which a request is sent in order to guarantee a minimum success probability and a minimum number of users involved.

V. Conclusion & Outlook

In this paper, we have discussed user-based joint decisions in P2P systems as a means to counterbalance missing central trusted instances and predefined security policies in order to achieve basic security objectives such as authentication and access control. We have described different interaction schemes for the joint decision process. For each interaction scheme, we have developed a stochastic model that describes the performance of the scheme. The models developed hold for non-interactive threshold signature and multisignature schemes which we have identified to be the most appropriate tools for enabling joint decisions in our application scenario. We have provided a closed-form description of the stochastic models which allows the relevant parameters of the joint decision process to be adjusted during runtime.

In future work, we plan to assess the performance of the interaction schemes introduced in a real P2P system. The stochastic models shall be validated by experimental results. For this, we use a pastry-based implementation that will be deployed in PlanetLab.

Acknowledgments

Special thanks go to Markus Fidler who helped us to get this work into the right direction.

References

[1] K. Kanchanasut et al., "A Multimedia Communication System for Collaborative Emergency Response Operation in Disaster-affected Areas," Asian Institute of Technology (AIT), Tech. Rep. TR_2007-1, 2007.
[2] Federal Ministry of Education and Research, Germany, "SoKNOS Project Homepage," *http://www.soknos.de*, 2008.
[3] C. Neuman et al., "The Kerberos Network Authentication Service (V5)," *IETF RFC 4120*, 2005.
[4] R. Gennaro et al., "Threshold RSA for Dynamic and Ad-Hoc Groups," in *Proc. of EUROCRYPT '08*.
[5] V. Shoup, "Practical Threshold Signatures," in *Proc. of EUROCRPYT 2000*.
[6] M. Narasimha et al., "On the Utility of Distributed Cryptography in P2P and MANETs: The Case of Membership Control," in *Proc. of ICNP '03*.
[7] N. Saxena et al., "Admission Control in Peer-to-Peer: Design and Performance Evaluation," in *Proc. of SASN '03*.
[8] ——, "Efficient Node Admission for Short-lived Mobile Ad Hoc Networks," in *Proc. of ICNP'05*.
[9] J. H. Yi, "Energy-Efficient and Non-interactive Self-certification in MANETs," in *Proc. of SSS 2006*.
[10] A. Rowstron et al., "Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems," in *Proc. of Middleware 2001*.
[11] M. Castro et al., "Scalable application-level anycast for highly dynamic groups," in *Proc. of NGC 2003*.
[12] S. M. Ross, *Probability Models*, 8th ed. Academic Press, 2003.