# *Access Control in Ad Hoc Groups*

Nitesh Saxena, Gene Tsudik, Jeong Hyun Yi
School of Information and Computer Science
University of California at Irvine
Irvine, CA 92697, USA
{nitesh,gts,jhyi}@ics.uci.edu

## Abstract

*Ad hoc groups, such as peer-to-peer (P2P) systems and mobile ad hoc networks (MANETs) represent recent technological advancements. They support low-cost, scalable and fault-tolerant computing and communication. Since such groups do not require any pre-deployed infrastructure or any trusted centralized authority they have many valuable applications in military and commercial settings as well as in emergency and rescue operations. However, due to lack of centralized control, ad hoc groups are inherently insecure and vulnerable to attacks from both within and outside the group.*

*Decentralized* access control *is the fundamental security service for ad hoc groups. It is needed not only to prevent unauthorized nodes from becoming members but also to bootstrap other security services such as key management. In this paper, we survey a number of practical distributed access control mechanisms based on various flavors of threshold signatures.*

## 1 Introduction

Ad hoc groups, such as peer-to-peer (P2P) systems and mobile ad hoc networks (MANETs), are very popular in today's computing, especially in the research community. They lack infrastructure and do not need any trusted authority. Moreover, they are inherently scalable and fault tolerant. Such characteristics find many interesting applications in military and commercial settings as well as in emergency and rescue operations. However, their open nature and lack of centralized control result in some security challenges.

The security research community recognized the need for specialized security services in ad hoc groups. *Access Control* is particularly important since most other traditional security are based upon it. In this context, an access control mechanism must prevent unauthorized nodes from becoming a part of the group and to establish trust among members in the absence of a trusted authority. Access control is also essential to bootstrap other security services, such as group key management and secure routing.

Zhou and Haas [35] first suggested using threshold cryptography [35] to secure mobile ad hoc networks. Their intuition was to distribute trust among the nodes of the network such that no less than a certain threshold of nodes are trusted. They proposed a distributed certification authority (CA) [13] which issues certificates (using some threshold signature [7] protocol) to the nodes joining the network. Certificates enable the nodes to communicate with each other in a secure and authenticated manner. This work also led to the development of COCA [36], an online certification authority for wired networks. Although attractive, this idea is not applicable to ad hoc groups. Their approach is hierarchical: only select nodes can serve as part of the certification authority and thus take part in admission decisions. Moreover, contacting the distributed CA nodes in a MANET setting is difficult since such nodes might be many hops away.

Luo, et al. considered the same problem in [22] and Kong, et al. in [20, 19] as well as [23, 21]. This body of work proposed a set of protocols for ubiquitous and robust access control in MANETs. They amended the model of Zhou and Haas to allow every member to participate in access control decisions, thus maintaining the true "peer" nature of ad hoc groups and providing increased availability. Unfortunately, this otherwise elegant scheme has been shown to be insecure [24, 16].

Recently, Kim, et al. [17] developed a group admission control framework based on a menu of cryptographic techniques. This framework classifies group admission policy according to the entity (or entities) making admission decisions. The classification included simple admission control policies, such as static ACL (Access Control List)- or attribute-based admission, as well as admission based on the decision of a fixed entity: external (e.g., a CA or a TTP) or internal (e.g., a group founder). Such simple policies are relatively easy to support and do not present much of a technical challenge. However, they are inflexible and ultimately unsuitable for dynamic ad hoc networks. Static ACLs enumerate all possible members and hence cannot support truly dynamic membership (although they work well for closed networks). Admission decisions made by a TTP or a group founder violate the

peer nature of the underlying ad hoc group.

In this paper, we consider several decentralized access control mechanisms that employ threshold cryptography (more specifically, threshold signatures). We also propose a new access control mechanism based on a provably secure threshold Schnorr signature scheme [12].

**Organization:** Section 2 overviews the generic group access control protocol. Sections 3-6 discuss the use of various types of threshold signatures in building effective decentralized access control mechanisms. Then, all proposed mechanisms are evaluated and compared based on the basis of features and efficiency factors in Section 7.

## 2 Group Access Control

$(t+1, n)$ threshold cryptography employs the secret sharing of the group secret among $n$ members in such a manner that any set of $t+1$ members can recover the group secret and perform a cryptographic operation jointly.

A threshold signature scheme [7] enables any subgroup of $t+1$ members in a group to collaboratively sign messages on behalf of that group. This is achieved by secret-sharing the signature key among the group members, and allowing them to compute a signature on some message via a distributed protocol in which the members use the shares of the signature key instead of the key itself. Threshold signature schemes can tolerate up to $t$ corruptions in the whole lifetime of the system.

The idea of threshold signatures applies directly to build access control mechanisms by making collaborative decisions. Next, we overview a generic access control protocol.

**Access Control Protocol Overview.** The access control mechanism is initiated by a prospective member (or node) or an "applicant". At the end, given that enough honest members (or nodes) ($\geq t+1$) approve admission, the applicant becomes a member of the group (or network) and possesses its group membership certificate.

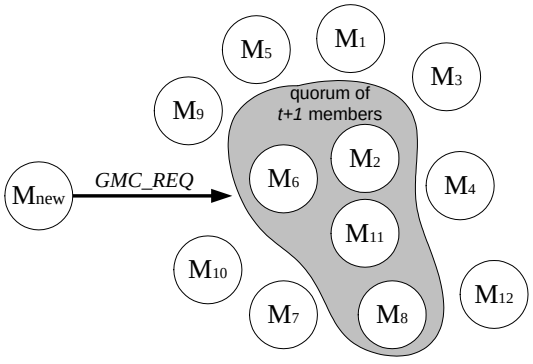


**Figure 1. Access Control**

*Step 0. Bootstrapping:* A prospective member $M_{new}$ obtains the *group charter* [17] which contains admission policies and various security parameters out of band and then the information of current group size from some bootstrap node. This process is performed only once per admission.

*Step 1. Certification Request:* $M_{new}$ initiates the protocol by sending a signed certification request (`GMC_REQ`) message to the group. `GMC_REQ` includes the name of the group and the usual fields, such as issuance time and the validity interval. It may reference $M_{new}$'s identity PKC $PKC_{new}$. Alternatively, or additionally, it may also contain a distinct public key (for which $M_{new}$ knows the corresponding secret key) to be used as $M_{new}$'s unique security credential within the group. How this request is sent to the group is application-dependent.

*Step 2. Admission Decision:* Upon receipt of `GMC_REQ`, a group member $M_i$ first extracts the sender's $PKC_{new}$ and verifies the signature. If $M_i$ approves of admission, it replies with a signed vote $vote_i$ (or a partial signature) on `GMC_REQ` message and its group membership certificate $GMC_i$. Several signature schemes (as will be described in following sections) can be used for this purpose. Note that, depending on the underlying signature scheme, this step may involve multiple rounds and/or co-ordination among signing members.

*Step 3. GMC Issuance*[1]*:* On receiving $k$ ($\geq t+1$) messages in the above step, $M_{new}$ randomly selects $t+1$ out of $k$ of these sponsoring members, validates the membership of selected $M_i$-s by asserting the correctness of the corresponding $GMC_i$-s, verifies the individual votes, and, from them, composes its own $GMC_{new}$.

Various flavors of threshold signatures exist in literature: RSA based, DSA based, Schnorr based and more recently, BLS [5] based. However, known provably secure threshold RSA signatures *do not* yield access control mechanisms for ad hoc networks. We begin by carefully considering various threshold RSA schemes and explain why they are not applicable for access control in ad hoc networks.

## 3 Threshold RSA Signatures

**Schemes by Frankel et al. [9, 10] and Rabin [28].** The currently known provably secure threshold RSA signature schemes, two schemes by Frankel, et al. [9, 10] and a scheme by Rabin [28], are not easily applicable for access control in ad hoc networks. In particular, the RSA signature scheme of [9] is practical only for small networks, while in the other two provably secure threshold RSA schemes known today [10, 28] (which employ additive secret sharing as opposed to polynomial secret sharing of [31]) the members participating in the threshold signature protocol need to reconstruct the secret shares of the nodes that are currently inaccessible to them. In this way both protocols essentially equate a temporarily inaccessible node with a corrupt one, whose secrets might just as

[1]The joining member must also be issued its share of the secret signing key (so that it can participate in future admission decisions). This process, referred to as *partial secret share shuffling* is described in [20]. We denote the shuffled partial share provided by member $M_i$ to $M_{new}$ by $pss_i(id_{new})$.

well be reconstructed. This is an undesirable feature for asynchronous ad hoc networks where members are often inaccessible to one another. In such settings we need to enable isolated but large enough subgroups of nodes to operate without reconstructing everyone else's secrets. COCA [36] employs a modified version of Rabin's RSA scheme which overcomes this problem of availability. However, this scheme, which is based on combinatorial secret sharing as opposed to the additive sharing of Rabin, is applicable only for small networks, because in large networks the number of combinations $\binom{n}{t}$ becomes intractable.

**Scheme by Shoup [32].** Another well known and more recent provably secure threshold RSA scheme was proposed by Shoup [32]. This scheme is more elegant than the ones discussed above because the signature generation and verification is fully non-interactive and it also avoids the inaccessibility problem by employing the polynomial $(t+1, n)$ secret sharing of Shamir [31]. However, since the secret sharing is performed over secret modulo $\phi(N)$ (unlike over publicly known integers in the schemes discussed above), it is not possible for the nodes in the network to provide a new node with its secret share in the access control protocol. Moreover, Shoup's scheme requires a trusted dealer to generate the RSA keys which is an undesirable feature in an ad hoc network. Boneh and Franklin [4] developed a method to generate an RSA modulus in a distributed fashion. Alas, it might not be possible to use this method, since Shoup's scheme require that the common RSA modulus $N$ be a product of two *safe* primes. (Informally, a large prime $p$ is *safe* if $p = 2q + 1$ where $q$ is itself a large prime.) Furthermore, we believe that using any method to generate RSA keys in a distributed manner involves prohibitively high communication and/or computation overhead which severely impacts the practicality of such techniques in many group setting (such as MANETs).

# 4 Threshold DSA based Access Control

In this section, we describe the access control mechanism (referred to as TS-DSA) [24, 29] based on the threshold DSA scheme [11] of Gennaro, et al.

TS-DSA can be initialized by either: (1) a trusted dealer or (2) a group of $3t + 1$ or more founding members.

## 4.1 Setup

INITIALIZATION BY DEALER. The trusted dealer generates the system parameters $(p, q, g)$, selects a random polynomial $f(z) = f_0 + f_1 z + \cdots + f_t z^t$ over $\mathbb{Z}_q$ of degree $t$, such that the group secret is $f(0) = f_0 = x$. In order to enable verifiable secret sharing, VSS in short, (refer to [8]), the dealer computes and publishes the witnesses $W_i = g^{f_i}$ for $(i = 0, \cdots, t)$. The witness value $W_0 = g^x$, also denoted by $y$, is actually the group public key. Next, for each $M_i$, the dealer computes the secret share $ss_i$ such that $ss_i = f(id_i) \pmod q$ and issues the group membership certificate $GMC_i$. Note that the dealer is not required hereafter.

SELF-INITIALIZATION BY FOUNDING MEMBERS. The founding members $M_i(|i| \geq 3t+1)$ select individual polynomials $f_i(z)$ over $\mathbb{Z}_q$ of degree $t$, such that $f_{i0} = x_i$. Then, using the joint secret sharing (JSS) protocol (refer to [26]), each $M_i$ computes its own secret share $ss_i$, such that $ss_i = \sum_{j=1}^{l} f_j(id_i) \pmod q$ $(|l| \geq 3t+1)$. Also, the dealing process supports VSS. Now, in order to provide each member with a membership certificate, any set of $2t+1$ founding members must collaborate.

## 4.2 Admission Process

Let $n$ $(\geq 3t+1)$ be the number of current group members. The prospective member $M_{new}$ invokes the admission process described in section 2. The first three steps of the protocol are exactly the same as in section 2 (with the admission threshold $2t+1$), the remaining steps are described in the following.

| | | |
|---|---|---|
| $M_{new} \rightarrow M_i$: | $m, S_{new}(m), PKC_{new}$ | (1) |
| $M_{new} \leftarrow M_i$: | $GMC_i$ | (2) |
| $M_{new} \rightarrow M_j$: | $SL_{new}$ | (3) |
| $M_{new} \leftarrow M_j$: | $u_j,\ v_j$ | (4) |
| $M_{new} \rightarrow M_j$: | $r$ | (5) |
| $M_{new} \leftarrow M_j$: | $vote_j, pss_j(id_{new})$ | (6) |

$m$=GMC_REQ$_{new}$
$vote_j = k_j(m + ss_j r) \pmod q$

**Figure 2. TS-DSA Admission Protocol**

1. Each $M_j$ randomly chooses a polynomial $k_j(z),\ a_j(z)$ in $\mathbb{Z}_q$ of degree $t$. $M_j$ computes $k_j(i)$ and $a_j(i)$ for all signers $M_i(i = 1, \cdots, 2t+1)$ in $SL_{new}$, and then distributes $k_j(i)$ and $a_j(i)$ to all co-signers. After receiving the partial shares from other co-signers, $M_j$ computes $k_j$ and $a_j$ such that $k_j = k(j) = \sum_{l=1}^{2t+1} k_l(j),\ a_j = a(j) = \sum_{l=1}^{2t+1} a_l(j) \pmod q$. Then, $M_j$ computes $u_j$ and $v_j$ such that $u_j = k_j \cdot a_j \pmod q,\ v_j = g^{a_j} \pmod p$, and sends $u_j$ and $v_j$ back to $M_{new}$.
2. $M_{new}$ now computes $u$ and $v$ such that $u = \sum_{j=1}^{2t+1} u_j\ l_j(0) \pmod q$ ($l_j(0)$ is the Lagrange's interpolation polynomial computed at point 0) which finally equals to $ka \pmod q$, $v = \prod_{j=1}^{2t+1} (v_j)^{l_j(0)} \pmod p$ which equals $g^a \pmod p$. Next, $M_{new}$ computes the inverse $u^{-1} \pmod q$ and finally computes $r$ as $r = (v^{u^{-1}} \bmod p) \bmod q$ which equals $(g^{k^{-1}} \bmod p) \bmod q$. Then, $M_{new}$ sends $r$ to $M_j$.
3. $M_j$ computes partial signature $vote_j$ such that $vote_j = k_j(m + ss_j r) \bmod q$, where $m$ is information derived from the $GMC\_REQ_{new}$. Then, $M_j$ sends $vote_j$ to $M_{new}$.
4. $M_{new}$ computes the complete signature $s$ such that $s = \sum_{j=1}^{2t+1} vote_j \cdot l_j(0) \pmod q$ which equals $k(m + xr)$

$(\mathrm{mod}\ q)$. $M_{new}$ verifies the signature by using the equation $g^{m \cdot s^{-1}} \cdot y^{r \cdot s^{-1}} = r \pmod p$i. In case this verification fails, $M_{new}$ verifies the validity of each $vote_j$ by using the equation $g^{m \cdot vote_j^{-1}} \cdot y_j^{R \cdot vote_j^{-1}} = g^{k_j^{-1}} \pmod p$, and in the process traces the malicious group member. Here $y_j = g^{ss_j} \pmod p$, which can be calculated using VSS. $g^{k_j^{-1}} \pmod p$ can be similarly computed using the public witnesses of the polynomials $k_j(z)$-s which will be broadcast by the members $M_j$-s in the step (1) above.

# 5 Threshold BLS based Access Control

We now describe the access control mechanism (referred to as TS-BLS) based on the threshold BLS [5] scheme of [3]. The identity-based version of this scheme appeared in [30].

## 5.1 Setup

Similar to the TS-DSA, TS-BLS can also be initialized by either: (1) a trusted dealer or (2) a group of $2t + 1$ or more founding members.

In either case, the dealer or the founding members first initialize and generate the appropriate elliptic curve domain parameters $(p, \mathbb{F}_p, a, b, A, q)$. $\mathbb{G}_1$ is set to be a group of order $q$ generated by $A$, $\mathbb{G}_2$ is a subgroup of $\mathbb{F}^*_{p^2}$ of order $q$, and $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ is defined to be a public bilinear mapping. Also, $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1^*$ is the hash function that maps binary strings to non–zero points in $\mathbb{G}_1$. All of this information is published and all group members (as well as prospective members) are assumed to have access to it.

The set-up procedure is exactly the same as in TS-DSA (refer to section 4.1). Only difference is that the VSS operations are performed in the elliptic curve domain and threshold BLS signing is used to issue certificates.

## 5.2 Admission Process

Let $n(\geq 2t + 1)$ be the number of current group members. The prospective member $M_{new}$ invokes the admission process. The first three steps of the protocol are exactly the same as in section 2, the remaining steps are described in the following.

| | | |
|---|---|---|
| $M_{new} \rightarrow M_i$: | $m, S_{new}(m), PKC_{new}$ | (1) |
| $M_{new} \leftarrow M_i$: | $GMC_i$ | (2) |
| $M_{new} \rightarrow M_j$: | $SL_{new}$ | (3) |
| $M_{new} \leftarrow M_j$: | $vote_j, pss_j(id_{new})$ | (4) |
| $m$=GMC_REQ$_{new}$ | | |
| $vote_j = ss_j H_1(m)$ | | |

**Figure 3. TS-BLS Admission Protocol**

1. $M_j$ produces the signature on message $m$ as $vote_j = ss_j H_1(m)$ and sends it to $M_{new}$.

2. $M_{new}$ recovers the threshold signature $s$ as $s = \sum_{j=1}^{t} vote_j l_j(0)$. $M_{new}$ then verifies the correctness of $s$ by checking $\hat{e}(s, A) = \hat{e}(H_1(m), y)$ (recall that $y = xA$ is the group public key here). On failure of this verification, $M_{new}$ traces the faulty member(s) by verifying the correctness of each $vote_j$ as $\hat{e}(vote_j, A) = \hat{e}(H_1(m), y_j)$. Each $y_j = ss_j A$ can be computed using the public witness values as in VSS.

# 6 Threshold Schnorr based Access Control

In this section, we propose the access control mechanism (referred to as TS-Schnorr) based on the threshold Schnorr scheme [12].

## 6.1 Setup

TS-Schnorr can be initialized by either: (1) a trusted dealer or (2) a group of $2t + 1$ or more founding members. In either case, the set-up procedure is exactly the same as in TS-DSA (refer to section 4.1). Here the admission threshold is $t + 1$.

## 6.2 Admission Process

Let $n$ $(\geq 2t + 1)$ be the number of current group members. The prospective member $M_{new}$ invokes the admission process described in section 2. The first three steps of the protocol are exactly the same as in section 2, the remaining steps are described in the following.

| | | |
|---|---|---|
| $M_{new} \rightarrow M_i$: | $m, S_{new}(m), PKC_{new}$ | (1) |
| $M_{new} \leftarrow M_i$: | $GMC_i$ | (2) |
| $M_{new} \rightarrow M_j$: | $SL_{new}$ | (3) |
| $M_{new} \leftarrow M_j$: | $vote_j, pss_j(id_{new})$ | (4) |
| $m$=GMC_REQ$_{new}$ | | |
| $vote_j = k_j + c.ss_j \pmod q$, $c = H(m\|\|r)$ | | |

**Figure 4. TS-Schnorr Admission Protocol**

1. Each $M_j$ randomly chooses a polynomial $k_j(z)$ in $\mathbb{Z}_q$ of degree $t$. $M_j$ computes $k_j(i)$ for all signers $M_i (i = 1, \cdots, t + 1)$ in $SL_{new}$, and then distributes $k_j(i)$ to all co-signers. After receiving the partial shares from other co-signers, $M_j$ computes $k_j$ such that $k_j = k(j) = \sum_{l=1}^{t+1} k_l(j) \pmod q$. Each $M_j$ then computes $r = \prod_{l=1}^{t} r_j^{l_j(0)} \pmod p$, where $r_j = g^{k_j} \pmod p$ values are computed using the broadcast public witnesses (as in VSS) of the polynomials $k_j(z)$-s. $M_j$ then computes $c = H(m||r)$, the signature $vote_j$ such that $vote_j = k_j + c \cdot ss_j \pmod q$, and sends it to $M_{new}$.

2. $M_{new}$ first computes $r = \prod_{l=1}^{t} r_j^{l_j(0)} \pmod p$, where $r_j = g^{k_j} \pmod p$ values are computed using the public witnesses (as in VSS) of the polynomials $k_j(z)$-s. $M_{new}$ then computes $c = H(m||r)$ and interpolates the threshold signature $s$ such that $s = \sum_{j=1}^{t+1} vote_j \cdot l_j(0)$

$(\mathrm{mod}\ q)$ which equals $k + cx\ (\mathrm{mod}\ q)$. $M_{new}$ verifies the signature by using the equation $g^s = ry^c\ (\mathrm{mod}\ p)$. In case this verification fails, $M_{new}$ verifies the validity of each $vote_j$ by using the equation $g^{vote_j} = g^{k_j} {y_j}^c\ (\mathrm{mod}\ p)$, and in the process traces the malicious group member(s). Here $y_j = g^{ss_j}\ (\mathrm{mod}\ p)$, which is calculated using VSS.

## 7 Comparison

The access control mechanisms discussed thus far offer very different alternatives for ad hoc networks. The security of both *TS-DSA* and *TS-Schnorr* is based on discrete logarithm problem (DLP) and inherently the generation of a random value (and in turn interaction) among signers is required, while the security of *TS-BLS* is based on elliptic curve discrete logarithm problem (ECDLP) and has a fully non-interactive signature generation.

Table 1 compares the respective computation and communication costs required for a new node to join the network. With *TS-DSA*, $2t + 1$ signers are required in order to be able to tolerate $t$ faults, while $t + 1$ partial shares are needed to reconstruct the secret share for joining. Both *TS-Schnorr* and *TS-BLS* schemes require $t + 1$ partial signatures as well as partial shares. Note that the partial signature with *TS-BLS* is constructed using elliptic curve operations which are relatively costly.

As for the communication cost, with *TS-DSA*, new member must contact $2t+1$ existing members for signature reconstruction and $t+1$ members out of $2t+1$ of these committing members for new share acquisition. For this share computation, $t + 1$ designated members should securely communicate among themselves due to a secret share shuffling [20] which requires $(t+1)t$ extra rounds. Note that shuffling technique is required in all other schemes too. Since admission protocol requires 6 rounds with *TS-DSA*, the total communication rounds are $(2t + 1) * 6 + (t + 1)t = t^2 + 13t + 6$. Also the bandwidth used with *TS-DSA* admission protocol is given by the sizes of $2t + 1$ partial signatures, $t + 1$ partial shares, and $(t+1)t$ random numbers. Assuming the signature key size to be $1024$ bits, the bandwidth requirement is more than $(2t + 1) * 1024 + (t + 1) * 160 + (t + 1)t * 160 = (2t+1)*1024+(t+1)^2*160$, since certificate size is greater than $1024$ bits. Similarly, we can analyze the communication rounds and the bandwidth required for *TS-Schnorr* and *TS-BLS*. As shown in Table 1, due to comparatively lesser communication, *TS-BLS* is more applicable for MANETs, in which the amount of communication is directly related to the battery power of the mobile devices (refer to [2]). On the other hand, since the partial signature computation with *TS-BLS* requires $t + 1$ expensive scalar-point-multiplication operations in elliptic curves, we expect that *TS-Schnorr* would perform better in terms of the computation time. Hence, *TS-Schnorr* will be a good candidate for wired P2P networks where communication cost is not of great concern.

A new member must ascertain the validity of the acquired certificate and secret share. This is what we call as the *verifiability*. Also, when the new member detects that its certificate and/or secret are not valid, it must be able to trace the bogus partial signatures and/or partial shares. This functionality is termed as the *traceability*.

Table 2 compares the costs for the verifiability and traceabilty procedures for *TS-DSA*,*TS-Schnorr* and *TS-BLS*.[2] Both *TS-DSA* and *TS-Schnorr* require two and $t+2$ exponentiations for certificate and share verification, respectively, whereas *TS-BLS* uses two tate-pairing operations and $t + 2$ scalar-point-multiplication operations which are more expensive. This means that *TS-DSA* and *TS-Schnorr* would perform better that *TS-BLS* as far as the verifiabilty is concerned. The same pattern is expected for the traceablity. Note that the *verifiability* is always required, however the *traceability* is only necessary when a member detects (from the verifiability service) that its reconstructed certificate and/or share are not valid.

## References


[1] Y. Amir, Y. Kim, C. Nita-Rotaru, J. Schultz, J. Stanton, and G. Tsudik. Exploring robustness in group key agreement. In *IEEE International Conference on Distributed Computing Systems,*, pages 399–408, April 2001.

[2] K. Barr and K. Asanovic. Energy Aware Lossless Data Compression. In *International Conference on Mobile Systems, Applications, and Services (MobiSys)*, May 2003.

[3] A. Boldyreva. Effi cient threshold signatures, multisignatures and blind signatures based on the gap-diffi e-hellman-group signature scheme. In *Proceedings of International Workshop on Practice and Theory in Public Key Cryptography*, volume 2567 of *LNCS*, pages 31–46, 2003.

[4] D. Boneh and M. Franklin. Effi cient Generation of Shared RSA keys. In B. Kaliski, editor, *CRYPTO '97*, number 1294 in LNCS, pages 425–439. IACR, 1997.

[5] D. Boneh, B. Lynn, and H. Shacham. Short Signatures from the Weil Pairing. In C. Boyd, editor, *ASIACRYPT'01*, number 2248 in LNCS, pages 514–532. IACR, 2001.

[6] B. Dahill, B. Levine, E. Royer, and C. Shields. A secure routing protocol for ad hoc networks. Technical Report UM-CS-2001-037, University of Massachusetts, August 2001.

[7] Y. Desmedt and Y. Frankel. Threshold cryptosystems. In G. Brassard, editor, *CRYPTO '89*, number 435 in LNCS, pages 307–315. IACR, 1990.

[8] P. Feldman. A practical scheme for non-interactive verifi able secret sharing. In *28th Symposium on Foundations of Computer Science (FOCS)*, pages 427–437, 1987.

[9] Y. Frankel, P. Gemmell, P. D. MacKenzie, and M. Yung. Proactive RSA. In *Proc. of Crypto'97*, pages 440–454, 1997.

[10] Y. Frankel, P. D. MacKenzie, and M. Yung. Adaptive security for the additive-sharing based proactive rsa. In *Public Key Cryptography 2001*, pages 240–263, 2001.

[11] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin. Robust Threshold DSS Signatures. In U. Maurer, editor, *EUROCRYPT '96*, number 1070 in LNCS, pages 354–371. IACR, 1996.


[2] Note that for proper security of discrete log in $\mathbb{F}_{p^2}$ for an elliptic curve of *TS-BLS* the prime $p$ should be at least 512-bits long which is equivalently as secure as 1024-bit $p$ in *TS-DSA* and *TS-Schnorr*

**Table 1. Admission Complexity**

| | TS-DSA | TS-Schnorr | TS-BLS |
|---|---|---|---|
| Certifi cate size (bits) | $> 1024$ | $> 1024$ | $> 512$ |
| Secret share size (bits) | 160 | 160 | 160 |
| Partial signatures | $2t + 1$ | $t + 1$ | $t + 1$ |
| Partial shares | $t + 1$ | $t + 1$ | $t + 1$ |
| Comm. Rounds | $t^2 + 13t + 6$ | $(t + 1)(t + 4)$ | $(t + 1)(t + 4)$ |
| Bandwidth (bits) | $> (2t + 1) * 1024 + (t + 1)^2 * 160$ | $> (t + 1) * 1024 + (t + 1)^2 * 160$ | $> (t + 1)^2 * 160$ |

**Table 2. Verifiablity and Traceability Complexity**

| | TS-DSA | TS-Schnorr | TS-BLS |
|---|---|---|---|
| Signature verify | 2 EXP | 2 EXP | 2 TPO |
| Share verify | $t + 2$ EXP | $t + 2$ EXP | $t + 2$ SPM |
| Partial sig. trace | $3(2t + 1)$ EXP | $3(t + 1)$ EXP | $2(t + 1)$ TPO |
| Partial share trace | $(t + 1)(t + 2)$ EXP | $(t + 1)(t + 2)$ EXP | $(t + 1)(t + 2)$ SPM |

EXP: Modular Exponentiation SPM: Scalar-Point-Multiplication

TPO: Tate Pairing operation


[12] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin. Secure Applications of Pedersen's Distributed Key Generation Protocol . In *The Cryptographers' Track at the RSA Conference (CT-RSA)*, April 2003.

[13] R. Housley, W. Polk, W. Ford, and D. Solo. Internet X.509 Public Key Infrastructure Certifi cate and Certifi cate Revocation List (CRL) Profi le. RFC 3280, IETF, Apr. 2002.

[14] Y.-C. Hu, D. Johnson, and A. Perrig. Sead: Secure effi cient distance vector routing for mobile wireless ad hoc networks. In *IEEE Workshop on Mobile Computing Systems and Applications*, June 2002.

[15] Y.-C. Hu, A. Perrig, and D. B. Johnson. Ariadne: A secure on-demand routing protocol for ad hoc networks. In *Proceedings of the Eighth ACM International Conference on Mobile Computing and Networking (Mobicom 2002)*, Sept. 2002.

[16] S. Jarecki, N. Saxena, and J. H. Yi. An Attack on the Proactive RSA Signature Scheme in the URSA Ad Hoc Network Access Control Protocol. In *ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN)*, October 2004.

[17] Y. Kim, D. Mazzocchi, and G. Tsudik. Admission Control in Peer Groups. In *IEEE International Symposium on Network Computing and Applications (NCA)*, Apr. 2003.

[18] Y. Kim, A. Perring, and G. Tsudik. Simple and fault-tolerant key agreement for dynamic collaborative groups. In *ACM Conference on Computer and Communications Security*, pages 235–244, November 2000.

[19] J. Kong, H. Luo, K. Xu, D. L. Gu, M. Gerla, and S. Lu. Adaptive Security for Multi-level Ad-hoc Networks. In *Journal of Wireless Communications and Mobile Computing (WCMC)*, volume 2, pages 533–547, 2002.

[20] J. Kong, P. Zerfos, H. Luo, S. Lu, and L. Zhang. Providing Robust and Ubiquitous Security Support for MANET. In *IEEE 9th International Conference on Network Protocols (ICNP)*, 2001.

[21] H. Luo, J. Kong, P. Zerfos, S. Lu, and L. Zhang. URSA: Ubiquitous and Robust Access Control for Mobile Ad Hoc Networks, available on-line at `http://www.cs.ucla.edu/wing/publication/publication.html`. In *IEEE/ACM Transactions on Networking (ToN)*, to appear, Oct 2004.

[22] H. Luo and S. Lu. Ubiquitous and Robust Authentication Services for Ad Hoc Wireless Networks, available on-line at `http://citeseer.ist.psu.edu/luo00ubiquitous.html`. Technical Report TR-200030, Dept. of Computer Science, UCLA, 2000.

[23] H. Luo, P. Zerfos, J. Kong, S. Lu, and L. Zhang. Self-securing Ad Hoc Wireless Networks. In *Seventh IEEE Symposium on Computers and Communications (ISCC '02)*, 2002.

[24] M. Narasimha, G. Tsudik, and J. H. Yi. On the Utility of Distributed Cryptography in P2P and MANETs: The Case of Membership Control. In *IEEE International Conference on Network Protocol (ICNP)*, pages 336–345, November 2003.

[25] P. Papadimitratos and Z. Haas. Secure routing for mobile ad hoc networks. In *Communication Networks and Distributed Systems Modeling and Simulation Conference*, 2002.

[26] T. P. Pedersen. A threshold cryptosystem without a trusted party. In D. Davies, editor, *EUROCRYPT '91*, number 547 in LNCS, pages 552–526. IACR, 1991.

[27] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. D. Tygar. Spins: Security protocols for sensor networks. In *Mobile Computing and Networking*, 2001.

[28] T. Rabin. A Simplifi ed Approach to Threshold and Proactive RSA. In H. Krawczyk, editor, *CRYPTO '98*, number 1462 in LNCS, pages 89 – 104. IACR, 1998.

[29] N. Saxena, G. Tsudik, and J. H. Yi. Admission Control in Peer-to-Peer: Design and Performance Evaluation. In *ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN)*, pages 104–114, October 2003.

[30] N. Saxena, G. Tsudik, and J. H. Yi. Identity-based Access Control for Ad-Hoc Groups. In *International Conference on Information Security and Cryptology (ICISC)*, 2004 To appear.

[31] A. Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, Nov. 1979.

[32] V. Shoup. Practical Threshold Signatures. In B. Preneel, editor, *EUROCRYPT '00*, number 1807 in LNCS, pages 207–220. IACR, 2000.

[33] M. Steiner, G. Tsudik, and M. Waidner. Cliques: A new approach to group key agreement. *IEEE Transactions on Parallel and Distributed Systems*, August 2000.

[34] M. Steiner, G. Tsudik, and M. Waidner. Key agreement in dynamic peer groups. In *IEEE Transactions on Parallel and Distributed Systems*, July 2000.

[35] L. Zhou and Z. J. Haas. Securing Ad Hoc Networks. *IEEE Network Magazine*, 13(6):24–30, 1999.

[36] L. Zhou, F. Schneider, and R. van Renesse. Coca: A secure distributed on-line certifi cation authority. *ACM Transactions on Computer Systems*, 20(4):329–368, November 2002.