Privacy-Preserving Admission to Mobile Peer-to-Peer Groups

Mark Manulis
Cryptographic Protocols Group
Department of Computer Science
TU Darmstadt & CASED, Germany
Email: mark@manulis.eu

***Abstract*—Mobile peer-to-peer groups, which do not require any pre-deployed infrastructure or trusted centralized authority are valuable for a variety of collaborative applications. This work is focused on how to securely admit new users to such groups. Existing mechanisms based on threshold cryptography require that prospective members collect sufficient number of individual votes from the group prior to obtaining a membership credential. However, this approach does not consider the desirable anonymity of group members towards the admitted or declined users. This paper presents an admission control mechanism in which group members decide collectively and notify prospective members on the outcome of their decision without revealing their identities to prospective members.**

***Keywords*-admission control, peer-to-peer groups, privacy**

I. Introduction

After entering our daily life in early nineties mobile communication has experienced its rapid development and is now used in different fields of transportation, medical care, commerce, education, entertainment and other industries. Especially, communication in mobile peer-to-peer networks is one of the most promising technologies for the near future since it does not rely on any additional infrastructure and provides higher mobility and flexibility. Many applications such as audio/video conferences, decision making systems, collaborative work-flow systems are based on the group communication technology where all involved participants are usually composed into a group. Obviously, due to the absence of any pre-deployed infrastructure and because of the decentralized nature of mobile peer-to-peer networks the initialization of the group and its maintenance upon dynamic changes is a challenging task and one of the most important research topics in the field of security.

In this paper we focus on the admission control mechanism for mobile peer-to-peer groups. We consider groups without any centralized (trusted) group authority which are initialized and maintained in a collective manner. This implicitly means that also decisions on the admission of prospective members should be carried out collectively. One popular form of such decision making process is based on voting according to some specified admission policy. According to [7] the admission policy for a collective decision can be either static or dynamic. In case of the *static admission policy* a fixed number of members' votes is required to grant the access to the group. This policy may work well in static groups, however if the number of group members falls below the specified fixed value then alternative policies are required. The *dynamic admission policy* requires a certain fraction v of votes to allow a prospective member become part of the group. Obviously, this policy is more flexible and suitable for mobile peer-to-peer groups. The dynamic admission policy is also fairer because every prospective member requires the same percentage of votes to be admitted to the group, whereas for the static policy this percentage depends on the actual group size.

In this paper we propose an admission control protocol for mobile peer-to-peer groups based on the dynamic admission policy. We show how founding members can initialize the group, how members can collectively decide about the admission of other users, and how a group member can prove own membership in the group towards other group members and also non-members. Unlike existing admission control schemes (c.f. Section II) we focus on privacy issues during the admission process. In particular, we wish to protect anonymity of users participating in the collective decision phase. In this context we consider anonymity of users towards the admitted or declined users during and after the admission phase as a prime privacy-preserving goal. We show that this goal is not provided by current solutions where prospective group members are required to collect individual confirmation votes from the group members; thus, being able to learn the identities of group members who voted for the admission and deduce the identities of group members who did not vote. Especially for collaborative applications this lack of anonymity is undesirable, since admitted group members may object collaboration with group members who did not express their consent. In order to meet this anonymity goal we our solution uses several modern cryptographic techniques.

Organization: Section II describes related work on the admission control for (mobile) peer-to-peer groups. Section III highlights the main ideas behind our solution. Section IV describes the proposed privacy-preserving admission control protocol in detail. Section V specifies the main security requirements and explains why our protocol achieves them.

978-1-4244-5328-3/10/$26.00 ©2010 IEEE

Section VI describes some additional features of our protocol for securing diverse forms of communication within and beyond the group.

II. Related Work and Analysis

Kim et al. [7] proposed an admission control framework suitable for different kinds of peer groups and suggested some realizations using various cryptographic techniques. Their framework relies on two basic elements: group charter and group authority. Group charter is a document which contains information about the group including the admission rules. Group authority is the entity which decides about group membership requests. The framework includes various admission policy types depending on the structure of the group authority: (i) public access control lists (ACLs) which can be used without any group authority; (ii) admission decision made by a centralized group authority (e.g. a group manager or a trusted third party), and (iii) admission decision made collectively by current group members. Obviously, public ACLs are unsuitable for dynamic mobile peer-to-peer groups because all prospective group members should be known in advance which is an unrealistic assumption. The centralized admission decision made by a group manager or a trusted third party violates the trust relationship in such groups where no such trusted parties are available. Hence, the admission policy based on collective decisions is surely the most suitable form for mobile peer-to-peer groups. The framework in [7] uses the notion of group membership certificates (GMCs) that allow members to prove own group membership. Every group member obtains own GMC by the end of the admission protocol.

Saxena et. al. [13] proposed an abstract model for the access control protocol based on the framework from [7] and provided three concrete realizations. The difference to [7] is that upon collecting enough approval votes prospective user U computes own GMC without further interaction. All three protocol use secret sharing techniques and require secure p2p-channels between the founding group members. All these protocols rely on the static admission policy since the threshold value used in the secret sharing scheme remains constant.

In an earlier work [10], Narasimha et al. described a solution based on RSA threshold signatures. Similar to [13] their protocols are expensive in terms of computation and communication and require secure point-to-point channels between participants during the initialization and the admission protocol. Also, group members who wish to approve U's request need to redistribute their threshold shares.

Later, Saxena et al. [14] described another admission control protocol for MANETs and P2P networks. The difference to the previous approaches is that members use their identities (which are bound to public keys) to compute the secret shares. Their protocol also distinguishes between internal and external group membership proofs.

Further, Saxena et al. [15] proposed two admission control protocols (UniAC and BiAC) for short-lived mobile ad-hoc networks. The difference to the previous approaches is that members do not require own membership certificates and are able to prove their membership over commitments to their secret shares derived from the shared polynomial. Similar to [13] the protocols can be initialized using a joint secret sharing scheme from [4]. However, the UniAC protocol is based on secret sharing with univariate polynomials $f(x)$ which results in a more efficient initialization procedure compared to the BiAC protocol which uses bivariate polynomials $f(x, y)$. This efficiency of the initialization in UniAC is achieved at the expense of the admission protocol since secret shares of t members who wish to approve the admission request have to be randomized using the inefficient *random shuffling* technique from [5] in order to prevent U from learning their secret shares. The additional strength of UniAC and BiAC protocols is that group members U_i and U_j after being admitted to the group are able to establish a secret key for the communication without additional interaction. This is a valuable property because pairwise key establishment is a frequent operation in peer-to-peer networks. We note that our protocol provides not only this property but goes a step further in setting up a secret key known to all group members as discussed in Section VI.

III. Overview of Our Solution

All proposals briefly described in Section II have the following common form: a prospective member U^* has to collect at least t votes signed by current group members before U^* can join to the group. Moreover, these votes must be transmitted confidentially through a secure communication channel. It is implicitly assumed that such channels exist. In practice, this transmission would, however, reveal the identities of voting members to the prospective members. Since the identities of group members (through their certificates) are publicly known U^* would be able to distinguish between members who have explicitly approved his admission and members who did not.

In order to achieve the desired anonymity of voters we apply a different technique: we let the founding group members execute a group key exchange (GKE) protocol in order to agree on a secret shared group key k. A crucial security requirement of GKE protocols is that the established group key k remains known only to the participants of the protocol. Moreover, GKE execution does not require any secure channels amongst the users and can be performed over a public, insecure network. Once the group is initialized any prospective member U^* can send own membership request to the group. Then, current group members can exchange their votes encrypted with k and in case the number of positive votes exceeds the fraction v a new session of GKE protocol is executed in which U^* is considered as a new protocol participant, i.e. each group key is valid for the

period between two successful admission procedures. The important property of our approach is that the voting phase itself does not involve U^*, i.e. the voting process is carried out amongst the members of the group and the individual votes remain hidden from U^* due to the applied encryption.

During the voting phase we let each user sign his own vote before encrypting it with the group key. In this way we ensure that the same user submits at most one vote per admission procedure. In our admission protocol each U_i is in possession of a private/public key pair (sk_i, pk_i) which will be used to generate and verify these signatures. A key pair (sk_i, pk_i) is, however, bound to the preceding execution of the GKE protocol and remains valid between two consecutive admission events. In this way we ensure that if U_i is in possession of sk_i for some pk_i that has been used in the preceding GKE execution then U_i is a valid member of the group.

We will use a special class of GKE protocols according to the following definition from [9] (here slightly modified to suit the context):

Definition 1 (GKE+P Protocol): A *group key exchange protocol enabling on-demand derivation of p2p keys (GKE+P)* amongst n users $U_1, \ldots, U_n$ consists of an interactive *group phase* at the end of which participants compute a shared group key k_G and an on-demand executable non-interactive *p2p phase* in which any two users U_i and U_j derive their own secret p2p key $k_{i,j}$.

As shown in [9] such GKE+P protocols can be constructed from any GroupDH protocols, i.e. GKE protocols that generalize the classical two-party key exchange protocol from [3] to a group setting as defined in the following (here slightly modified to suit the context):

Definition 2 (GroupDH Protocol): Let $\mathbb{G}$ be a cyclic group of prime order Q and $g \in \mathbb{G}$ its generator. It is assumed that the Discrete Logarithm Problem[1] is intractable in $\mathbb{G}$.

A *GroupDH protocol* is a GKE protocol amongst n users $U_1, \ldots, U_n$ such that during its execution each user U_i chooses own exponent $x_i \in_R \mathbb{Z}_Q$ and at the end all users compute a secret shared group element $k \in \mathbb{G}$ which is the output of $f(g, x_1, \ldots, x_n)$ for some function $f : \mathbb{G} \times \mathbb{Z}_Q^n \to \mathbb{G}$ specific to the protocol.

Some GroupDH protocols like [16], [1], [6], [8] require that each user U_i chooses own exponent $x_i \in_R \mathbb{Z}_Q$ in the first protocol round and broadcasts the corresponding public value $y_i = g^{x_i}$. However, (x_i, y_i) can also be seen as a Discrete Logarithm-based asymmetric key pair (sk_i, pk_i) of U_i. Since (x_i, y_i) is used to compute the group key k we immediately have a link between the public key $pk_i = y_i$ of U_i and the execution of the GKE+P protocol. Therefore, we can setup a group by executing the GKE+P protocol and consider $pk_i = y_i$ as a public key of group member U_i.

[1]Given $y_i \in \mathbb{G}$ find $x_i \in \mathbb{Z}_Q$ with $y_i = g^{x_i}$

In the admission control protocol we can let each U_i sign own votes with $sk_i = x_i$ so that other group members will be able to verify these signatures using pk_i whereby being assured that the signature has been produced by a legitimate member of the group.

IV. Privacy-Preserving Admission Control Protocol

A. Building Blocks

In the following we list the three cryptographic building blocks of our construction.

GKE+P Protocol $\mathcal{P}$: Let $\mathcal{P}$ denote a GKE+P protocol from Definition 1 which is in turn based on a GroupDH protocol from Definition 2 during which each user U_i chooses own secret exponent x_i and broadcasts the corresponding public value y_i in the first round. As mentioned earlier, such GroupDH protocol can be found in [16], [1], [6], [8].

A GKE+P protocol $\mathcal{P}$ is called *secure* if the group key k remains indistinguishable from some random value of the same length to any party which was not a legitimate participant of the protocol session in which k was established. In particular, each new session of $\mathcal{P}$ results in a new group key which is independent from any key computed in some other (earlier or later) session.

Digital Signature Scheme Σ: A digital signature scheme Σ consists of the signing and verification algorithms $(\mathtt{Sig}, \mathtt{Ver})$. By $\sigma := \mathtt{Sig}(sk, m)$ we denote the signature on a message m computed using the private key sk. The verification algorithm $\mathtt{Ver}(pk, m, \sigma)$ returns `true` if the signature on m is valid under the public key pk; otherwise the algorithm returns `false`.

A digital signature scheme Σ is called *existentially unforgeable under chosen message attacks (EUF-CMA)* if it is infeasible for a given public key pk to compute (m, σ) such that $\mathtt{Ver}(pk, m, \sigma) = \mathtt{true}$ after obtaining valid message-signature pairs (m', σ') for $m \neq m'$.

Since in our protocol we will have $(sk, pk) = (x, y)$ with $y = g^x$ the appropriate EUF-CMA secure signature schemes is, for example, DSA from [12].

Symmetric Encryption Scheme $\mathcal{E}$: A symmetric encryption scheme $\mathcal{E}$ such as AES from [11] consists of the encryption and decryption algorithms $(\mathtt{Enc}, \mathtt{Dec})$. By $c = \mathtt{Enc}(k, m)$ we denote the encryption of a message m using the symmetric key k. The corresponding decryption $\mathtt{Dec}(k, c)$ returns m.

A symmetric encryption scheme Σ is called *indistinguishable under chosen ciphertext attacks (IND-CCA)* if for any random key k it is infeasible to distinguish between $c_0 = \mathtt{Enc}(k, m_0)$ and $c_1 = \mathtt{Enc}(k, m_1)$ for any chosen $m_0 \neq m_1$ after obtaining decryptions $m' = \mathtt{Dec}(k, c')$ for any chosen $c' \neq c_0 \neq c_1$.

B. Group Charter

A group charter GC contains at least the following entries: a group id G, the admission fraction v (as explained in the context of the dynamic admission policy in Section I), and the current group public key PK. We assume that G and v are agreed upon in advance, while PK is initially set to be empty.

C. Initial Group Formation

In order to initialize G the founding members $U_1, \ldots, U_n$ execute a session of the GKE+P protocol $\mathcal{P}$. At the end of the protocol each U_i holds the group key $k = f(g, x_1, \ldots, x_n)$, own private/public key pair $(sk_i, pk_i) = (x_i, y_i)$ as well as public keys of all other group members $(pk_1, \ldots, pk_n) = (y_1, \ldots, y_n)$. Then, each U_i computes the group public key $PK := (pk_1, \ldots, pk_n)$ and broadcasts the signature $\sigma_i = \texttt{Sig}(sk_i, GC)$ where $GC = (G, v, PK)$. Finally every group member U_i verifies each received σ_j, computes $S = (\sigma_1, ..., \sigma_n)$, and stores (GC, S); this in addition to the secret values k and sk_i. The signature set S serves as an authenticator for the group charter GC. Since no centralized infrastructure is deployed all members have to authenticate GC collectively.

D. Our Admission Control Protocol

A prospective group member U^* can join the group G if at least a fraction v of current group members approves his admission request sent to the group. Our privacy-preserving admission control protocol consists of four stages detailed in the following.

Stage 1 (GC Fetch) U^* obtains (GC, S) and verifies the validity of the group charter GC by verifying each signature σ_i in S using the appropriate public key pk_i from PK (which is part of GC).

Stage 2 (Join Request) U^* proceeds as follows:

- choose $x^* \in_R \mathbb{Z}_Q$; compute $y^* = g^{x^*}$;
- define $(sk^*, pk^*) = (x^*, y^*)$;
- compute $\sigma^* = \texttt{Sig}(sk^*, GC)$;
- broadcast $U^*|pk^*|\sigma^*$ to the members of G.

Stage 3 (Collective Decision) Every group member U_i from G proceeds as follows:

- set $\texttt{vote}_i$ either to `accept` or `decline`;
- compute $\sigma_i = \texttt{Sig}(sk_i, \texttt{vote}_i|U^*|pk^*|GC)$;
- compute $c_i = \texttt{Enc}(k, \texttt{vote}_i|U^*|pk^*|GC|\sigma_i)$;
- broadcast c_i.

Upon receiving encrypted votes c_j from all other group members each U_i proceeds as follows:

- decrypt each $\texttt{vote}_j|U^*|pk^*|GC|\sigma_j = \texttt{Dec}(k, c_j)$;
- verify that each $\texttt{Ver}(pk_j, \texttt{vote}_j|U^*|pk^*|GC, \sigma_j) = \texttt{true}$;
- discard votes with invalid signatures and votes of users who voted twice;
- if the number of valid votes with value `accept` satisfies the admission fraction v then continue with Stage 4; otherwise abort (meaning that U^* will not be admitted to the group and PK remains unmodified).

Stage 4 (Group Admission) Users $U_1, \ldots, U_n$ and U^* execute a new session of a GKE+P protocol $\mathcal{P}$. For simplicity we set $U^* = U_{n+1}$. At the end of the protocol each U_i holds the updated group key $k = f(g, x_1, \ldots, x_n, x_{n+1})$ where $x_1, \ldots, x_n \in \mathbb{Z}_Q^n$ are *new* exponents chosen respectively by $U_1, \ldots, U_n$ in this session (i.e. replacing their exponents from the previous session) and $x_{n+1} (= x^*)$ is the exponent chosen by the new member U_{n+1} in Stage 2. Thus, this execution implies the update of all (sk_i, pk_i) for $i \in [1, n]$ and so of the group public key $PK = (pk_1, \ldots, pk_{n+1})$ in the GC.

In order to complete the protocol members have to collectively update the authenticator S of GC. This is done similarly to the initialization phase from Section IV-C. I.e., each U_i broadcasts the signature $\sigma_i = \texttt{Sig}(sk_i, GC)$ where $GC = (G, v, PK)$ and stores (GC, S) where $S = (\sigma_1, \ldots, \sigma_{n+1})$.

E. Proof of Group Membership

Any group member U_i can prove own group membership to some verifier V (whereby V can be some other group member or some third party) by proving the knowledge of the secret key $sk_i = x_i$ for which the corresponding public key pk_i is part of the current group public key PK from GC. This proof can be done via a simple challenge-response protocol, e.g. U_i signs some challenge r chosen by V, that is computes $\sigma_i = \texttt{Sig}(sk_i, r)$. Note that this proof is implicitly used in Stage 3 of our protocol to prevent double-voting.

Another property of our solution is that U_i can also prove own group membership to V without disclosing own public key pk_i. If V is some third party then U_i can use the (non-interactive) zero-knowledge proof of knowledge of 1-out-of-2 discrete logarithms from [2] extended to a 1-out-of-n version, i.e. U_i proves to V the knowledge of some $\log_G y_i$ from $\log_G y_1$, ..., $\log_G y_n$ without revealing the index i. If V is another group member, say U_j, then U_i and U_j can authenticate each other as group members using the shared group key k. Since k is known to all members this authentication does not provide non-repudiation within the group.

V. Security Requirements and Analysis

A. Desirable Security Requirements

One of basic security requirements for an admission control protocol is that no user can claim to be a member of some group if this user has not been admitted to that group before. This requirement intuitively assumes that users should not be able to forge their group membership proofs. This intuition serves as a basis for the following informal definition of unforgeability.

Definition 3 (Unforgeability): It must be computationally infeasible for a user U to prove the group membership in some group G without having own membership being approved through the admission control protocol.

While unforgeability is a necessary security requirement for any admission control protocol, it does not cover the desired aspects of privacy. In particular, our goal was to achieve privacy of votes during the admission phase against prospective group members. This can be done by requiring that prospective members do not obtain any information about the individual votes of group members. Our informal definition of anonymity given below is more general as it aims to protect the anonymity of votes against any non-member U.

Definition 4 (Anonymity): It must be computationally infeasible for a user U who was not a member of G during the execution of the admission control protocol to obtain any information about the individual votes $\texttt{vote}_i$ of group members U_i.

We remark that (as discussed in Section II) the anonymity requirement implicitly prevents any prospective member U from being able to distinguish between group members who approve or disapprove his join request. In particular, our definition also allows that U is admitted to the group at some later stage; thus, we also ensure anonymity of votes "in the future".

B. Security Analysis of Our Protocol

In this section we argue informally that our protocol from Section IV-D satisfies the desirable security requirements of unforgeability and anonymity defined in the previous paragraph if secure primitives, i.e. GKE+P protocol $\mathcal{P}$, digital signature scheme Σ, and symmetric encryption scheme $\mathcal{E}$, are used.

Unforgeability: Assume that some user U_j can prove own membership in G without being previously admitted to G. In this case U_j must be able to output signature σ_j such that $\texttt{Ver}(pk_i, \sigma_j, m) = \texttt{true}$ for some pk_i in PK. Since the signature scheme is assumed to be EUF-CMA secure this may occur only with negligible probability.

Anonymity: Assume that $\texttt{vote}_i$ submitted by some group member U_i during the Stage 3 of our protocol leaks some information, i.e. it is possible to distinguish whether $\texttt{vote}_i$ was set to `accept` or `decline`. Since each $\texttt{vote}_i$ is encrypted in c_i this attack would mean that the ciphertext produced by the underlying encryption scheme $\mathcal{E}$ leaks information about the encrypted plaintext. Since $\mathcal{E}$ is assumed to be IND-CCA secure and the secret group key k is known only to group members as guaranteed by the security of the used GKE+P protocol $\mathcal{P}$ this may occur only with negligible probability.

VI. Additional Properties of Our Protocol

Our privacy-preserving admission control protocol provides some further useful properties with regard to the communication in the peer-to-peer group:

Secure Group Communication: The secret group key k can be used by all group members to communicate securely within a group, i.e. this form of communication is given implicitly through the use of our protocol for the purpose of the admission. This is in contrast to admission protocols from [7], [10], [13], [14], [15] where secure group communication would require a separate execution of a GKE protocol.

Secure Peer-to-Peer Communication: Our protocol uses a GKE+P protocol as a building block. The distinguished feature of GKE+P protocols is that in addition to the secret group key any protocol participant U_i can derive an independent secret peer-to-peer key $k_{i,j}$ for any other participant U_j without any further communication. Therefore, our admission control protocol implicitly provides group members with the ability to communicate securely on a peer-to-peer basis. Note that secure peer-to-peer communication is also provided by the BiAC protocol from [15].

Secure Communication with Non-Members: Since in our protocol each U_i holds own private/public key pair (sk_i, pk_i) as a result of the underlying GKE+P protocol any non-member can send some confidential message m to any group member using some asymmetric public key encryption scheme and encrypting m with pk_i.

Furthermore, the secret group key k can be used to derive an asymmetric key pair (sk_G, pk_G). The public key pk_G can then be used by any non-member to send some confidential message to the whole group.

VII. Conclusion

In this paper we have proposed a new approach for the admission control in mobile peer-to-peer groups where group members collectively approve or decline the joining request of some user while remaining anonymous to that user. The anonymity of voting members remains preserved against the admitted group members. In addition to the desired privacy goals our protocol provides support for secure group and peer-to-peer communication both within the same group and between members of the group and non-members.

REFERENCES

[1] M. Burmester and Y. Desmedt. A Secure and Efficient Conference Key Distribution System. In *Advances in Cryptology— EUROCRYPT 1994*, *Lecture Notes in Computer Science*, vol. 950, pp. 275–286. Springer, 1994.

[2] J. Camenisch and M. Michels. A Group Signature Scheme with Improved Efficiency. In *International Conference on the Theory and Applications of Cryptology and Information Security*, pp. 160–174. Springer, 1998.

[3] W. Diffie and M. E. Hellman. New Directions in Cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654, 1976.

[4] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin. Secure Distributed Key Generation for Discrete-Log Based Cryptosystems. In *Advances in Cryptology—Eurocrypt 1999*, *Lecture Notes in Computer Science*, vol. 1592, pp. 295–310. Springer, 1999.

[5] A. Herzberg, S. Jarecki, H. Krawczyk, and M. Yung. Proactive Secret Sharing, Or How to Cope with Perpetual Leakage. In *Advances in Cryptology—CRYPTO 1995*, *Lecture Notes in Computer Science*, vol. 963, pp. 339–352. Springer, 1995.

[6] J. Katz and M. Yung. Scalable Protocols for Authenticated Group Key Exchange. In *Advances in Cryptology—CRYPTO 2003*, *Lecture Notes in Computer Science*, vol. 2729, pp. 110–125. Springer, 2003.

[7] Y. Kim, D. Mazzocchi, and G. Tsudik. Admission Control in Peer Groups. In *2nd IEEE International Symposium on Network Computing and Applications*, pp. 131–139. IEEE CS, 2003.

[8] Y. Kim, A. Perrig, and G. Tsudik. Tree-based Group Key Agreement. *ACM Transactions on Information and System Security*, 7(1):60–96, 2004.

[9] M. Manulis. Group Key Exchange Enabling On-Demand Derivation of Peer-to-Peer Keys. In *International Conference on Applied Cryptography and Network Security*, *Lecture Notes in Computer Science*, vol. 5536, pp. 1–19. Springer, 2009.

[10] M. Narasimha, G. Tsudik, and J. H. Yi. On the Utility of Distributed Cryptography in P2P and MANETs: The Case 0f Membership Control. In *IEEE International Conference on Network Protocols*, pp. 336–345. IEEE CS, 2003.

[11] National Institutes of Standards and Technology. FIPS 197: Advanced Encryption Standard, 2001.

[12] National Institutes of Standards and Technology. FIPS 186-3: Digital Signature Standard, 2009.

[13] N. Saxena, G. Tsudik, and J. H. Yi. Access Control in Ad Hoc Groups. In *International Workshop on Hot Topics in Peer-to-Peer Systems*, pp. 2–7. IEEE CS, 2004.

[14] N. Saxena, G. Tsudik, and J. H. Yi. Identity-Based Access Control for Ad-Hoc Groups. In *International Conference on Information Security and Cryptology*, *Lecture Notes in Computer Science*, vol. 3506, pp. 362–379. Springer, 2004.

[15] N. Saxena, G. Tsudik, and J. H. Yi. Efficient Node Admission for Short-lived Mobile Ad Hoc Groups. In *13th International Conference on Network Protocols*, pp. 269–278, IEEE CS, 2005.

[16] D. G. Steer, L. Strawczynski, W. Diffie, and M. J. Wiener. A Secure Audio Teleconference System. In *Advances in Cryptology—CRYPTO 1988*, *Lecture Notes in Computer Science*, vol. 403, pp. 520–528. Springer, 1990.