

A-MAKE: An Efficient, Anonymous and Accountable Authentication Framework for WMNs

Ahmet Onur Durahim
Sabanci University
Istanbul, Turkey
durahim@sabanciuniv.edu

Erkay Savaş
Sabanci University
Istanbul, Turkey
erkays@sabanciuniv.edu

Abstract—In this paper, we propose a framework, named as A-MAKE, which efficiently provides security, privacy, and accountability for communications in wireless mesh networks. More specifically, the framework provides an anonymous mutual authentication protocol whereby legitimate users can connect to network from anywhere without being identified or tracked. No single party (e.g., network operator) can violate the privacy of a user, which is provided in our framework in the strongest sense. Our framework utilizes group signatures, where the private key and the credentials of the users are generated through a secure three-party protocol. User accountability is implemented via user revocation protocol that can be executed by two semi-trusted authorities, one of which is the network operator. The assumptions about the trust level of the network operator are relaxed. Our framework makes use of much more efficient signature generation and verification algorithms in terms of computation complexity than their counterparts in literature, where signature size is comparable to the shortest signatures proposed for similar purposes so far.

Index Terms—Wireless Mesh Networks; Anonymous Authentication; Pairing; Group Signatures;

I. INTRODUCTION

Wireless mesh networks (WMNs) emerge as a promising technology to provide low cost and scalable solutions for high speed Internet access and additional services. Thus, it is no surprise that it has been the focus of increasing attention of all quarters from research community to industry and military. A WMN is a dynamically self-organized and self-configured network, where nodes automatically establish and maintain mesh connectivity in a collaborative fashion. The collaborative nature of the mesh networks results in low up-front cost, easy network maintenance, robustness and reliable service coverage [1].

In their simplest form, WMNs are comprised of mesh routers and mesh clients (network users) such as desktops, laptops, phones, etc. Hybrid architectures [1] (cf. Fig. 1) are the most popular since both mesh routers and mesh users perform routing and configuration functionalities for other user nodes to help improve the connectivity and coverage of the network.

In order to achieve wide user-acceptance and deployment of WMNs, security and privacy concerns of users need to be addressed in an efficient manner. Effective access control mechanisms to guarantee the registered users a reliable network connectivity and other security services for the protection of network communication are among the essentials due to the

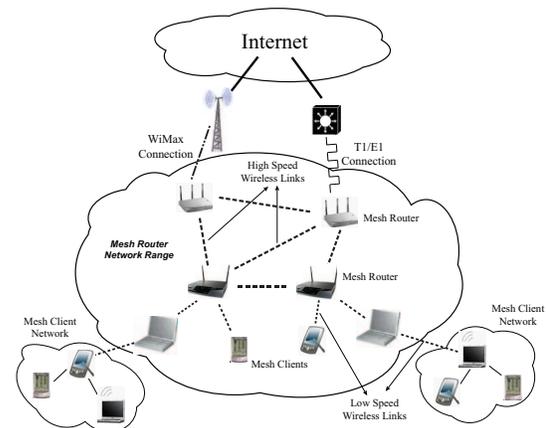


Fig. 1. Hybrid WMN architecture

dynamic and open nature of the network. By the same token, the services delivered to users may violate their privacy since they need to be authenticated first to connect to the network. Another related issue is user accountability, which aims to detect misbehaving users and, if needed, to deny network access to them via revoking. However, access control, security, user privacy and accountability can be conflicting objectives, which may not be easy to reconcile in the same framework.

In WMNs, it is essential to provide legitimate, privacy-aware network users with anonymous access to the network while unauthorized access by all others must be prevented. It is not immediately obvious as to how to block unregistered users when everybody is anonymous in the network. Furthermore, protecting the network from misbehaving users requires "identification" capability built into the network to achieve user accountability whereby users are held accountable for their (unacceptable) actions. Identification capability and anonymity are indeed conflicting goals since while the latter is trying to hide the user identity, the former is trying to reveal it.

In this paper, we propose a framework that is a collection of protocols to manage these conflicting objectives successfully. More formally; Security, User Privacy (Anonymity + Unlinkability), User Accountability and Key Revocation are the objectives efficiently achieved in our framework.

In our framework, users connect to the WMN using an anonymous mutual authentication protocol based on group signatures [8] where both signature generation and verification operations are efficient. Since the signature scheme is anonymous and unlinkable it is not possible to identify and track users, thus providing them with strong privacy. User accountability is achieved through an efficient user revocation protocol that can be executed only by a coalition of certain semi-trusted non-colluding parties. The revocation protocol can also be used for the users whose subscriptions expire while the backward security is guaranteed for users who are revoked. Our framework is practical and its protocols outperform similar protocols proposed for WMNs in literature [9].

The paper is organized as follows. In Section II, background information on pairings and group signatures are given and related work is discussed. In Section III, our construction is introduced. Then, detailed description of our security framework for privacy preserving authentication and key establishment will be given. In Section IV, security and privacy properties along with the performance analysis of our scheme will be examined. And Section V concludes the paper.

II. BACKGROUND AND RELATED WORK

The proposed framework makes use of anonymous group signatures, based on primitives such as elliptic curve arithmetic and pairing operations. Here, we give a brief introduction to bilinear pairings and group signatures utilized in our construction assuming an ample knowledge on elliptic curves.

Let G_1 , G_2 and G_M be cyclic groups of some large prime order q . Then, $\hat{e} : G_1 \times G_2 \rightarrow G_M$ is a **bilinear map**, which is efficiently computable and has the following property:

Bilinearity: $\forall P \in G_1$, and $\forall Q \in G_2$, and $\forall a, b \in \mathbb{Z}_q$, $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$.

In our construction, we use additive groups of elliptic curves for G_1 and G_2 , and multiplicative group in an extension of a finite field for G_M .

Group signatures are first introduced by Chaum and Heyst [7], which allow a group member to sign a message on behalf of the group without revealing its identity. Group signatures have been adopted in diverse application areas where anonymity is required [8, 9]. One of the most recent group signature application is the Direct Anonymous Attestation (DAA) scheme, which is originally proposed in [6].

A related framework for Accountable and Anonymous Authentication is proposed in Tsang et al. [10], in which service providers (SPs) authenticate users. In this work, there is no trusted third party (TTP) and accountability is provided by the blacklists of users that are held at SP side. Thus, the framework provides accountability on the SP side only. Therefore, it is not suitable for WMNs, where distributed accountability is required.

Ren and Lou [9] propose a more related framework to ours called PEACE which stands for SoPhisticated privacy-Enhanced yet Accountable seCurity framEwork for WMNs. PEACE is the first scheme that demonstrates that two conflicting goals, namely user privacy and accountability, can co-exist

in a practical and efficient framework. In PEACE, privacy is achieved through the use of short group signature scheme by Boneh and Shacham [5].

In PEACE, privacy against the network operator (NO) is obtained via the *late binding* of private keys by group managers to their corresponding users. Simply put, in late binding, the group manager (GM) determines which user will get which private key; and with the help of TTP, a user in the group can reconstruct its designated key. Although NO generates all the private keys, it is not a part of the late binding process and does not know user-private key mapping. NO can extract a private key used in a group signature, and determine the group id, and revoke all the private keys in that group. It, however, cannot trace it to the specific user who actually generates the signature. If any two of the three parties, i.e., the NO, the TTP, and the GM, collude, privacy and security can be forfeited for any given user.

Although the NO cannot reveal the identity of a specific user by only knowing the key used in a signature, it can trace any signature up to its group and use this information to violate the anonymity of the signer. Furthermore, the NO can link two anonymous signatures generated by the same user. The question here is "Is it sufficient to hide the identity of the user to protect his privacy?" This question reminds us of the infamous AOL Internet web search data release case that is a paramount example of privacy breach [3].

In this incident an AOL user whose identity was suppressed was easily tracked down and re-identified by knowing the web pages she visited. In summary, if we de-identify a user but allow him to be tracked, then we violate the privacy of that user. In PEACE, NO can track down the users in the network. Since the NO deploys the mesh routers and forms a well connected wireless mesh network (WMN), it can collect valuable data such as location and time of users' connections to the network.

Conclusion, then, is that user private keys should not be given to or generated by a single entity, especially by the NO due to its advantageously situated position. Furthermore, the NO generally is not the best choice for acting as the authorized party that we can easily bestow the trust of users; not to mention the cost associated with bearing such trust.

In PEACE the verification algorithm needs to check whether the signer is in user revocation list (UserRL) by computing two costly pairings per user in the list, which degrades the performance. Thus, a more efficient user revocation list checking algorithm is needed to enhance the performance of the framework.

III. A-MAKE: ANONYMOUS AND ACCOUNTABLE MUTUAL AUTHENTICATION AND KEY AGREEMENT

To address security and privacy concerns in WMNs in an efficient manner, we propose a privacy preserving mutual authentication and key agreement scheme with revocation capabilities. Our mutual authentication and key agreement protocol is based on the anonymous group signature used in DAA scheme proposed by Chen et al. [8]. In this section,

we give the details of our anonymous and accountable mutual authentication and key agreement framework (A-MAKE).

A. Network Architecture and Problem Formulation

Our WMN architecture comprises four entities; a network operator (NO), a trusted third party (TTP), mesh routers (MRs), and network users (NUs).

NO deploys a number of access points and mesh routers in order to provide network services to users. Network users subscribe to NO to use the network from anywhere within the WMN. In order to provide network access only to the legitimate users and protect network against malicious users, NO authenticates them via mesh routers. In addition, whenever it detects a misbehaving user or whenever a user's subscription period ends, it revokes the user and denies further access to the user. Naturally, the NO cannot be trusted to perform the revocation process by itself since this means the compromise of the user identity. Therefore, we stipulate that a revocation process requires involvement of both the TTP and the NO.

In WMNs, users connect to the network through both mesh routers and other users already connected to the network. Users that act as routers should also be able to authenticate the other users that are outside the range of mesh routers but still need to connect to the network. In addition, users must use necessary cryptographic means to protect their communication against eavesdropping, altering, and also sophisticated attacks aimed to compromise privacy. As a result, there is a need for a privacy preserving mutual authentication scheme with revocation capabilities for anonymous authorization of users and for a key agreement scheme to provide confidentiality and integrity.

The primary job of the TTP is to participate in the Join protocol (Section III-B2) for the generation of private keys and an associated credential for a user. The Join protocol gives the user a private key while the NO and the TTP obtains a share of it (without knowing the other party's share).

In order to provide confidentiality and integrity, a key agreement scheme is incorporated into the authentication scheme (Section III-B3).

Users that have the private key and the associated credential can perform two-party mutual authentication and key agreement protocol, MAKE, by mesh routers and other users. User accountability is achieved through user revocation protocol (Section III-C), which needs to bring the NO and the TTP together to revoke a user.

We have two important assumptions on TTP and NO. TTP and NO are semi-trusted non-colluding parties that follow the steps of the protocols. This is a relaxation compared to fully trusted model where trusted parties are usually in possession of private keys as is the case with [9]. An entity which is similar to a certificate authority (CA) in classical public key setting is an example as to how TTP is implemented in real world. Since user registration is performed once for every user and revocation of users is needed occasionally, TTP does not have to be highly accessible.

B. Our Construction

A-MAKE framework consists of four protocols (Setup, Join, MAKE, Revoke). In the following, we specify the detailed steps of the first three of these protocols.

1) **Setup:** Given the security parameter 1^k as the input, TTP performs the following steps:

- 1) Generates two groups G_1, G_2 of prime order $q \approx 2^k$ for which an asymmetric pairing can be defined. Solving decisional Diffie-Hellman problem (DDHP) and Gap-DLP [8] in G_1 is computationally hard,
- 2) Selects two generators P_1, P_2 such that $G_1 = \langle P_1 \rangle, G_2 = \langle P_2 \rangle$,
- 3) Selects a pairing such that $\hat{e} : G_1 \times G_2 \mapsto G_M$, where G_M is a multiplicative group of order q and the DLP in G_M is computationally hard,
- 4) Determines hash function $H_1 : \{0, 1\}^* \mapsto Z_q$ along with a key generating function $H_K : G_1 \mapsto Z_q$,
- 5) Generates its own public and private keys
 - a) Private key: (x, y) and $x, y \xleftarrow{R} Z_q$,
 - b) Public key: $(X, Y) \leftarrow (xP_2, yP_2) \in G_2$
- 6) Publishes public parameters, $\{G_1, G_2, G_M, \hat{e}, q, P_1, P_2, H_1, H_K, (X, Y)\}$

2) **Join:** The Join protocol is a three-party protocol that involves the user, the TTP and the NO. In this protocol, the NO, the TTP, and the user jointly generates the user private key, which is fully known only to the user. The NO and the TTP have its random shares, which contain no information about the private key. They store these shares along with corresponding user identities for revocation purposes in future. User privacy is guaranteed against the TTP and the NO) and the user can anonymously authenticate that it is a legitimate member of the network. In the following, protocol steps of the Join protocol for network user NU_i are described. Note that conventional PKC can be used for Steps 1, 2.c, 4.c, and 5.a.

- 1) NU_i generates a random number $r_{US_i} \xleftarrow{R} Z_q$ and sends it protected to NO
- 2) NO (for user NU_i where 'i' is the user identity)
 - a) Generates randomly the partial key $f_{NO_i} \xleftarrow{R} Z_q$
 - b) Keeps the binding (i, f_{NO_i})
 - c) Sends $r_{US_i} + f_{NO_i}$ protected to TTP together with $f_{NO_i} \cdot P_1$
- 3) TTP (for NU_i)
 - a) Generates randomly the partial key $f_{TTP_i} \xleftarrow{R} Z_q$
 - b) Stores the binding (i, f_{TTP_i})
 - c) Computes $f_{temp} \leftarrow r_{US_i} + f_{NO_i} + f_{TTP_i}$ and $f_{TTP_i} \cdot P_1$
 - d) Calculates $F_{NU_i} \leftarrow f_{NO_i} \cdot P_1 + f_{TTP_i} \cdot P_1$ ¹
- 4) TTP (generation of the credential for NU_i)
 - a) Generates a random number $r \xleftarrow{R} Z_q$
 - b) Calculates the credential
 - i) $A_i \leftarrow rP_1; B_i \leftarrow yA_i; C_i \leftarrow (xA_i + rxyF_{NU_i})$
 - ii) $cred_i \leftarrow (A_i, B_i, C_i)$

¹This value is used in generation of the NU_i 's credential.

- c) Encrypts both; $EC_i \leftarrow Enc_{PK_{NU_i}}(cred_i, f_{temp})$,
- d) Sends EC_i to NU_i

5) NU_i

- a) Decrypts EC_i in order to obtain the credential, $(cred_i, f_{temp}) \leftarrow Dec_{SK_{NU_i}}(EC_i)$
- b) Calculates its private key $f_i \leftarrow f_{temp} - r_{US_i}$ and $E_i = f_i \cdot B_i$
- c) Checks if the credential is generated appropriately:
If $\hat{e}(A_i, Y) \neq \hat{e}(B_i, P_2)$ or $\hat{e}(A_i + E_i, X) \neq \hat{e}(C_i, P_2)$, then abort protocol².

3) **MAKE - Mutual Authentication and Key agrEement:**

MAKE allows a user to authenticate itself to the network anonymously and obtain a symmetric secret key. It consists of three parts; namely, Sign, Key eXchange (KX), and Verify. First, we analyze the case when a user tries to connect to the network using a mesh router, and then give discussion for the case where a user acts as the router. For the latter case, user, which acts as a router in user-user authentication, should have UserRL in order to avoid granting rogue network users access to the network. Thus, UserRL is sent to a network user whenever it requests for it from any one of the Mesh Routers in a related beacon. So, in the following, it is assumed that all the agents acting as a relaying agent has the UserRL at their side. Alternative solution is that the user acting as a router gets help from a router to perform this check.

• **MAKE for User-Router Interaction (MAKE-UR)**

- 1) MR broadcasts a beacon periodically that contains an authentication payload (The following four steps are almost the same as those in [9]):

- a) Picks a random nonce $r_{MR} \xleftarrow{R} Z_q$, a time stamp ts_{MR} and a random generator $P_{MR} \in G_1$
- b) Computes $T_{MR} \leftarrow r_{MR} \cdot P_{MR}$
- c) Signs P_{MR} , T_{MR} and ts_{MR} using a conventional digital signature algorithm (e.g., ECDSA):
 $\sigma_{MR} \leftarrow Sign_{SK_{MR}}(P_{MR}, T_{MR}, ts_{MR})$
- d) Broadcasts $M_{sgMR} \leftarrow \{P_{MR}, T_{MR}, ts_{MR}, \sigma_{MR}, Cert_{MR}\}$ as a part of the beacon.

- 2) NU , upon receipt of M_{sgMR} , authenticates MR :

- a) Checks if the timestamp ts_{MR} is fresh
- b) Validates the CertMR using Online Certificate Status Protocol (OCSP) or with a similar protocol depending on the infrastructure design.
- c) Verifies signature σ_{MR} generated by MR . (non-anonymous authentication via conventional PKC).

- 3) NU authenticates itself to MR and initiates the authenticated key-exchange algorithm:

- a) For symmetric key establishment, NU
 - i) Picks a random nonce $r_{NU} \xleftarrow{R} Z_q$ and computes $T_{NU} \leftarrow r_{NU} \cdot P_{MR}$
 - ii) Calculates mutual key using H_K :
 $K_{UR} \leftarrow H_K(r_{NU} \cdot T_{MR})$
- b) For signature generation, NU

- i) Generates timestamp ts_{NU} to prove freshness
- ii) Generates a random point, $J \xleftarrow{R} G_1$, and a random number $t \xleftarrow{R} Z_q$
- iii) Randomizes the credential
 $(A', B', C') \leftarrow (t \cdot A, t \cdot B, t \cdot C)$
- iv) Calculate signature proofs of knowledge
 - A) $K \leftarrow f_i \cdot J$
 - B) Selects a random value $z \xleftarrow{R} Z_q$
 - C) Calculates pairing value to be supplied into the challenge and the witness
 $\rho'_D \leftarrow \hat{e}(z \cdot B', X)$, and $L \leftarrow z \cdot J$
 - D) Calculates challenge value
 $c \leftarrow H_1(params || A' || B' || C' || J || K || L || \rho'_D || K_{UR} || ts_{MR} || ts_{NU})$
 - E) Calculates response value
 $s \leftarrow z + c \cdot f_i \pmod{q}$
- v) Assembles the signature σ_{NU}
 $\sigma_{NU} \leftarrow (A', B', C', K, J, c, s)$
- vi) Sends signature σ_{NU} together with D-H key exchange share, T_{NU} , and timestamp ts_{NU}
 $M_{sgNU} \leftarrow \{\sigma_{NU}, T_{NU}, ts_{NU}\}$

- 4) MR verifies NU anonymously and obtains the shared key K_{UR} :

- a) Checks if the timestamp ts_{NU} is fresh
- b) Computes the shared secret key
 $K_{UR} \leftarrow H_K(r_{MR} \cdot T_{NU})$
- c) Checks if NU is in UserRL
If $\exists f_i \in UserRL$, such that $K = f_i \cdot J$, then rejects the signature and aborts the protocol.
- d) Checks the correctness of A' and B'
If $\hat{e}(A', Y) \neq \hat{e}(B', P_2)$, then rejects the signature and aborts the protocol.
- e) Verifies the Signature (Correctness of Proofs)
 - i) Performs the following computations
 $\rho'_A \leftarrow \hat{e}(A', X)$; $\rho'_B \leftarrow \hat{e}(B', X)$;
 $\rho'_C \leftarrow \hat{e}(C', P_2)$; $\rho'_D \leftarrow (\rho'_B)^s \cdot (\rho'_C / \rho'_A)^{-c}$
 $L' \leftarrow sJ - cK$
 - ii) Validates the challenge
If $c \neq H_1(params || A' || B' || C' || J || K || L' || \rho'_D || K_{UR} || ts_{MR} || ts_{NU})$, then rejects the signature and aborts the protocol.
- f) Allows the user to connect to the network.

Upon successful completion of the protocol, user and router can use the shared secret key K_{UR} to secure further communication between them.

• **MAKE for User-User Interaction (MAKE-UU)**

In case a user cannot find a router within its reception range but another user already connected to the network, two users can run a similar algorithm. The only difference from the previous scheme is that the relaying user also uses anonymous signature to authenticate to the connecting user. More precisely, the relaying user broadcasts beacon messages, which he signs using the anonymous

²This step is necessary also for checking the correctness of private key f_i

group signature scheme. As a result, both users mutually authenticate each other anonymously using their credentials they acquire in Join protocol.

C. User Accountability and Key Revocation

For user accountability, it is necessary to identify misbehaving users. Our protocol is designed to provide privacy in the strong sense [4], meaning that it is not possible to determine whether two signatures are generated by the same user. However, by performing this protocol on a given signature, TTP and NO can identify and revoke a user by revealing its private key. The TTP has the authority of revoking the user. Note that both TTP and NO maintain a list of pairs, (i, f_{TTP_i}) and (i, f_{NO_i}) , respectively.

1) **User (Private Key) Revocation:** If the owner of a signature σ_O is to be revoked, signer's private key is reconstructed by TTP and NO through the following protocol:

- 1) TTP performs the following operations:
 - a) Verifies the signature σ_O , where $\sigma_O = \{A_O, B_O, C_O, J_O, K_O, c_O, s_O\}$
 - b) If the signature verifies, then it sends J_O to NO and requests for the corresponding partial proofs (i.e., $f_{NO_i} \cdot J_O$ for each user).
- 2) Upon receiving J_O , the NO
 - a) Calculates partial proofs for every registered user $NU_i \in RU$, where RU stands for the list of registered users and $|RU| = n$ for the number of registered users; $\{\forall NU_i \in RU, f_{NO_i}; K_{NO_i} \leftarrow f_{NO_i} \cdot J_O\}$
 - b) Sends n proof pairs (i, K_{NO_i}) to TTP.
- 3) Using the proof pairs received from NO, TTP;
 - a) Calculates corresponding partial proofs using secret shares in its list: $\{\forall NU_i \in RU, f_{TTP_i}; K_{TTP_i} \leftarrow f_{TTP_i} \cdot J_O\}$
 - b) Calculates proofs by adding partial proofs K_{TTP_i} and K_{NO_i} and compare the result with $K_O = f_i \cdot J_O$:
 - i) $\forall NU_i \in RU$, calculate $K_i = K_{TTP_i} + K_{NO_i}$ and check if $K_i = K_O$
 - ii) If $\exists i$ for which $K_i = K_O$ then output i as the corresponding signer
- 4) On output ' i ', TTP asks for user NU_i 's partial private key value from NO by sending user id ' i '.
- 5) NO sends corresponding private key share f_{NO_i} to TTP.
- 6) Upon receiving partial secret, TTP
 - a) Computes secret key $f_i \leftarrow f_{TTP_i} + f_{NO_i}$
 - b) Adds f_i to user revocation list (UserRL) and
 - c) Adds user id ' i ' in another list to prevent rogue user from performing Join protocol in future.

IV. SECURITY AND PERFORMANCE ANALYSIS

In this section, we give security and performance analysis of our mutual authentication and key agreement architecture. The proposed architecture provides user-router mutual authentication where only the user is anonymous and user-user authentication whereby both users remain anonymous after the authentication.

A. Security Analysis

We assume that there exist pair-wise secure channels connecting the NO, the TTP and a user during the Join protocol where all exchanged information is protected. Since privacy and anonymity is not an aim of the Join protocol, securing the Join protocol can be performed using conventional methods.

User Anonymity against other users, NO and TTP: Our construction makes use of anonymous group signature scheme [8] to protect the anonymity of a user against the other users, mesh routers, the NO, and even against the TTP. Since no single entity within the network knows the private key of any user, no one is able to identify the owner of a given signature or link signatures generated by the same user. The TTP cannot link two signatures by the same user (even if TTP records the credential-user pairs) since the credential of a user is re-randomized for every authentication session. To identify, track (by linking signatures) and revoke a user, the NO and the TTP have to collaborate and run revocation protocol.

Confidentiality and Integrity: Communicating entities establish a shared symmetric key to secure their communications. In our proposal, we use authenticated Diffie-Hellman procedure to construct such a key between the communicating parties. A user that wants to connect to the network should generate random nonces to guarantee that a different key is generated in every session.

User Accountability: User accountability is made possible by the revocation capability incorporated into the scheme. Whenever malicious activity is observed, it can be reported to the TTP via providing a signature used by the malicious user for authentication. Then in accordance with the situation, TTP decides on whether to revoke reported user's private key or not. Also, network operator invalidates user subscription by again utilizing revocation protocol.

B. Performance Analysis

Performance analysis of our scheme, A-MAKE, in terms of computational and communication overhead is performed in this section in comparison with the security framework of PEACE [9]. We do not consider the Join protocol in our comparison since it is performed once for each user.

Computational Overhead: As seen from Table I, sign operation requires a single pairing and six elliptic curve multiplications (i.e., terms $1P$ and $6G_1$). The same operations in PEACE [9], takes two pairings and eight elliptic curve multiplications. The saving in the number of pairings and elliptic curve operations are extremely important for resource-constraint mesh clients.

In our scheme, verification algorithm needs five pairing operations (i.e., $5P$); $|UserRL|$ (i.e., the number of revoked users) elliptic curve multiplications; single multi-exponentiation (the term G_M^2) and two simultaneous point multiplications in elliptic curve group G_1 (the term G_1^2). On the other hand, PEACE requires six elliptic curve multiplications in addition to $3 + 2|UserRL|$ pairing computations.

| Operation | Party | Cost - AMAKE | Cost - PEACE [9] |
|-----------|------------------------------------|---|---------------------------|
| Join | TTP Network User Mesh Router | $3P + (2 + UserRL)G_1 + 2G_1^2$ $4P + 3G_1 + 1Sign$ $6G_1 + 1P$ | - |
| Sign | Network User | $1P + 6G_1$ | $2P + 8G_1$ |
| Verify | Mesh Router | $5P + UserRL G_1 + G_M^2 + G_1^2$ | $(3 + 2 UserRL)P + 6G_1$ |

TABLE I
COMPUTATIONAL OVERHEAD IN A-MAKE

Considering the user revocation list is not empty, the term $(|UserRL|)G_1$ in our protocol and $2(|UserRL|)P$ in PEACE dominate the computations. Since elliptic curve multiplication is much faster than pairing operation, our protocol outperforms the PEACE even for short user revocation lists. For example, whenever $|UserRL| > 2$, verification step of A-MAKE is carried out with less computational overhead than the one performed in PEACE.

Efficient verification algorithm for anonymous signatures is a crucial requirement in hybrid mesh network where regular users also perform verification of anonymous signatures while they act as routers. In summary, both our signature generation and verification algorithms are more efficient than their counterparts in PEACE.

Communication Overhead: In Table II, we provide the analysis of the communication overhead (due to anonymous authentication) of A-MAKE along with a comparison with PEACE. As seen from the table, the computational overhead

| Architecture | Communication Overhead | Total Bit Size |
|--------------|------------------------|-----------------------|
| PEACE | $2G_1 + 5Z_q$ | $7q + 2 \approx 1192$ |
| A-MAKE | $5G_1 + 2Z_q$ | $7q + 5 \approx 1195$ |

TABLE II
COMPARISON OF THE COMMUNICATION OVERHEAD (SIGNATURE SIZES)

of A-MAKE is comparable to PEACE.

V. CONCLUSION AND FUTURE WORK

We proposed a mutual anonymous authentication and key agreement framework for wireless mesh networks. Due to sophisticated yet efficient group signature protocol that is the pillar of our framework, the protocols are well suited to WMNs. Registered users can connect to the network from anywhere a router or another connected user is available without being identified or tracked. Therefore, our framework provides the user privacy in the strongest sense and the user accountability, which has been provided by none of the previous proposals in the literature.

An important contribution is the three-party Join protocol that reconciles the user privacy in the strongest sense and user accountability in an efficient and scalable manner. User revocation is possible only through the collaboration of two semi-trusted parties, namely the TTP and NO; therefore nobody can violate the privacy of users alone. Revocation procedure is efficiently performed under the control of the TTP.

We analyzed the performance of signature generation and verification algorithms in comparison with similar algorithms in literature. Our algorithms are more efficient in terms of computational complexity while the communication overhead is almost the same.

As a future work, we plan to implement the proposed framework and previous solutions in a simulation environment in order to provide some hard figures about the efficiency of the proposed protocols.

REFERENCES

- [1] I. F. Akyildiz, X. Wang, and W. Wang. Wireless mesh networks: a survey. *Computer Networks*, 47(4):445–487, 2005.
- [2] V. Atluri, B. Pfitzmann, and P. D. McDaniel, editors. *ACM CCS 2004, Washington, DC, USA, October 25-29, 2004*, 2004. ACM.
- [3] M. Barbaro and T. Jr. Zeller. A face is exposed for AOL searcher no. 4417749. *The New York Times*, 2006. URL <http://www.nytimes.com/2006/08/09/technology/09aol.html>. Accessed 25-February-2010.
- [4] M. Bellare, H. Shi, and C. Zhang. Foundations of group signatures: The case of dynamic groups. In Alfred Menezes, editor, *CT-RSA*, volume 3376 of *LNCS*, pages 136–153. Springer, 2005.
- [5] D. Boneh and H. Shacham. Group signatures with verifier-local revocation. In Atluri et al. [2], pages 168–177.
- [6] E. F. Brickell, J. Camenisch, and L. Chen. Direct anonymous attestation. In Atluri et al. [2], pages 132–145.
- [7] D. Chaum and E. van Heyst. Group signatures. In *EUROCRYPT*, pages 257–265, 1991.
- [8] L. Chen, P. Morrissey, and N.P. Smart. DAA: Fixing the pairing based protocols. *Cryptology ePrint Archive*, Report 2009/198, 2009. URL <http://eprint.iacr.org/>. Accessed 25-February-2010.
- [9] K. Ren and W. Lou. A sophisticated privacy-enhanced yet accountable security framework for metropolitan wireless mesh networks. In *ICDCS*, pages 286–294. IEEE Computer Society, 2008.
- [10] P. P. Tsang, M. H. Au, A. Kapadia, and S. W. Smith. Blacklistable anonymous credentials: Blocking misbehaving users without ttps. In *ACM CCS 2007*, pages 72–81. ACM, 2007.