An Approach of Communication Topology for Wireless Mesh Networks

M.Madhavi[1], B.Swetha[2]

[1] Computer Science and Engineering, Anurag Engineering College, Kodad,Andhra Pradesh 507001,India
[2] Computer Science and Engineering, Anurag Engineering College, Kodad,Andhra Pradesh 507001,India

Abstract

Recently, multihop wireless mesh networks (WMNs) have involved increasing attention and deployment as a low-cost move toward to give broadband Internet access at an urban scale. Security and privacy issues are of the major concern in pushing the success of WMNs for their broad deployment and for behind service-oriented applications. Regardless of the required, partial security research has been conducted toward privacy protection in WMNs. This motivates us to develop Anonymous and Accountable communication topology (AACT), a novel secure communication framework, tailored for WMNs. On one hand, AACT implements harsh user access control to cope with both free riders and spiteful users. On the other hand, AACT offers complicated user privacy protection beside both adversaries and a range of other network entities. AACT is accessible as a suite of authentication and key agreement protocols built upon our AACT. Our analysis demonstrates that AACT is resilient to a number of security and privacy related attacks. Additional methods were also discussed to further improve scheme efficiency.

Keywords: Wireless Mesh network, Anonimity, Onion ring cryptography, user security , user accountability**.**

1. Introduction

Wireless mesh networks (WMNs) have recently concerned rising attention and deployment as a promising low-cost approach to give last-mile high speed Internet access at urban scale [2], [3]. Typically, a WMN is a multihop layered wireless. The first layer consists of access points, which are high-speed wired Internet way in points. In the second layer, stationary mesh routers form a multihop spine via long-range high-speed wireless methods such as WiMAX [**6**]. The wireless spine connects to wired access points at some mesh routers through high speed wireless links. The third layer consists of a huge number of mobile network users. These network users access the network either by a direct wireless link or through a chain of other peer users to a nearby mesh router. WMNs correspond to a unique marriage of the ubiquitous coverage of large area cellular networks with the ease and the speed of the local area Wi-Fi networks [4]. The compensation of WMNs also contains low deployment costs, self-configuration and self maintenance, good scalability, high robustness, etc. [**2**].

Security and privacy issues are of mainly a concern in pushing the success of WMNs for their large deployment and for supporting service-oriented applications. Due to the essentially open and distributed nature of WMNs, it is necessary to enforce network access control to cope with both free riders and spiteful attackers. Dynamic access to WMNs should be subject to successful user authentication based on the correctly pre recognized trust among users and the network operator; otherwise, network access should be forbidden. On the other hand, it is also dangerous to provide good provisioning over user privacy as WMN communications regularly contain a vast amount of sensitive user details. The wireless standard, open network structural design, and be lacking in of physical protection over mesh routers render WMNs extremely vulnerable to different privacy-oriented attacks. These attacks range from passive eavesdropping to active message Phishing, interception, and modification, which could simply lead to the leakage of user information. Obviously, the wide deployment of WMNs can succeed only after users are assured for their capability to manage privacy risks and preserve their desired level of anonymity. Included with sensors and cameras, the WMN may also be used to gather information of interest. Perceptibly, all these communications include different kinds of sensitive user information like individual identities, actions, position information, fiscal information, transaction summaries, social/business connections, and so on. Once disclosed to the attackers, this information could negotiation any user's privacy, and when further associated together, can cause even more overwhelming consequences. Hence, securing user privacy is of paramount practical importance in WMNs. Moreover, for both billing purpose and avoiding the neglect of network resources, it is also necessary to exclude free riders and let only legitimate residents access WMNs.

1. CRYPTOGRAPHY SPECIFICS

1.1 Onion ring strategy [31]

The Onion routing [31] achieves communication privacy by making communication ends as unable to link. An Onion routing network consists of a number of interconnected Onion routers (ORs); each OR has a pair of public/private keys. Each OR knows the topology of the Onion network as well as the public keys of other ORs. An end user that requires an anonymous communication will send a request to an OR that it trusts; this OR is known as the Onion Proxy (OP) for the user. The communication between an end user and its OP is protected from the adversaries. The OP determines a route that consists of a series of ORs and constructs an "Onion" using the public keys of the routers en route. The "Onion" is constructed in a way such that the most inner part is the message to the intended destination. The message is wrapped, i.e., encrypted using the public keys of the ORs in the route, in the same order as the ORs appears in the route. Once an OR receives the Onioned message, it uses its private key to peel, i.e., decrypt, the "Onion", to obtain the information such as the next hop and the session key. It then forwards the rest of the "Onion" to the next hop. This process is repeated until the "Onion" reaches the last OR, which peels the last layer of the "Onion" and obtain the exit information, i.e., the destination.

For example, if the private route is $R_1 \rightarrow R_2 .... \rightarrow R_n$, where R_i is the i^{th} OR, and the last router R_n will connect to the exit funnel of the' ORs ', which will further communicate with the address requested by the session initiator; the message flow and the "Onion"(s) received at each router in the route are as follows:

$$E_{k_pR_1}\left(R_2, k_1, E_{k_pR_2}\left(....E_{k_pR_n}\left(k_n, \text{exit}\right)....\right)\right) \quad (1)$$

$$\rightarrow E_{k_pR_2}\left(...E_{k_pR_n}\left(k_n, \text{exit}\right)...\right)... \rightarrow E_{k_pR_n}\left(k_n, \text{exit}\right).$$

'k_pR_i' and 'k_i' are the public key and assigned session key for the i^{th} router. After the route is built up, session keys are used for constructing "Onion"s, and anonymous circuit ID (ACI) is used for routing. For the reverse path, data packet was encrypted with the session keys. The OP receives the "Onion" in the reverse path and peels it using the session keys it assigned to the ORs, and sends the raw data to the end user.

For an Onion route, only the proxy knows the ¯rst and the last router. Any OR in the route only knows its previous hop and next hop. For both outside attackers and inside attackers (i.e., compromised ORs), as encryption or decryption is processed at every OR, it is di±cult to link any two links (a link is a connection between two Onion routers) to the same route. Therefore, for a communication going through the Onion routers, the entry OR and exit OR are unable to link. When there are a large number of connections, it is di±cult to ¯nd out the two communication ends for any connection that applies Onion routing.

To avoid that the change of "Onion" size in the route built-up stage may give adversary hints about routing in- formation, an "Onion" has to be padded when part of its information has been read and removed, so that the length of the "Onion" keeps the same and it is difficult for an inside observer to obtain the routing information. Refer to [10], if the maximum number of Onion routers in a private route is N, the OP will construct a message of N "Onions" to build an Onion route. When an router receives the "Onion"s, it decrypts all the "Onion"s and obtain the routing information only from the ¯rst one. It then adds a dummy packet at the end, and forward the "Onion"s further.

For example, if the maximum hop count N is 5, and the private route is as $\mathbf{OP \rightarrow R_1 \rightarrow R_2 \rightarrow R_3}$, the message flow and the messages sent at each router are as follows:

$$OP \rightarrow R_1 : E_{k_pR_1}\left(R_2, k_1\right), E_{k_pR_1}\left(E_{k_pR_2}\left(R_3, k_2\right)\right), \quad (2)$$

$$E_{k_pR_1}\left(E_{k_pR_2}\left(E_{k_pR_3}\left(\text{exit}, k_3\right)\right)\right),$$

dummy,dummy

$$R_1 \circledR R_2 : E_{k_pR_2}\left(R_3, k_2\right), E_{k_pR_2}\left(E_{k_pR_3}\left(\text{exit}, k_3\right)\right),$$

dummy;dummy;dummy

$$R_2 \circledR R_3 : E_{k_pR_3}\left(\text{exit}, k_3\right),$$

dummy;dummy;dummy;dummy

1.2 Group Signature

Group signature schemes are a comparatively recent cryptographic concept introduced by Chaum and van Heyst in 1991 [9]. A group signature scheme is a technique for allowing a member of a group to sign a message on behalf of the group. In contrast to ordinary signatures, it gives anonymity to the signer, i.e., A verifier can only tell that a member of any group signed. However, in outstanding cases, such as a legal argument, any group signature can be "opened" by a designated group manager to make known clearly the identity of the signature's originator. Some group signature schemes support revocation, where group membership can be disabled. One of the most recent group signature schemes is the one proposed by Boneh and Shacham [8], which has an extremely short signature size that is similar to that of an RSA-1024 signature [10]. This scheme is based on the following two problems that are believed to be hard. Let G_1, G_2 , g_1, g_2 as defined above.

q-Strong Diffie-Hellman problem: The q-SDH problem in (G_1, G_2) is defined as follows: given a (q + **2**)-tuple $(g_1, g_2, g_2^{\gamma}, g_2^{(\gamma^2)}, ..., g_2^{(\gamma^q)})$ as input, output a pair $(g_1^{1/(\gamma+x)}, x)$, where $x \in Z_p^*$.

Decision linear on G_1: Given random generators ***u, v, h*** of G_1 and $u^a, v^b, h^c \in G1$ as input, output yes if ***a*** + b = c, and no, otherwise.

3 PROBLEM FORMULATION AND THE SCHEME OVERVIEW

3.1 Network Architecture and System Assumptions

The three-layer architecture in Fig. 1 consider a metropolitan-scale WMN under the manage of a network operator (NO). The network operator deploys a number of APs and mesh routers and forms a well-connected WMN that covers the whole area of a city and gives network services to network users, i.e., the citizens. Network users, on the other hand, subscribe to the network operator for the services and use their mobile clients to freely access the network from wherever within the city. The membership of network users may be **1**) completed/renewed according to user- operator agreement in an episodic manner or **2**) dynamically revoked by ***NO*** in case of argument/attack.

Similar to [4], [11], we assume that the downlink from a mesh router to all users within its reporting is one hop. However, the uplink from a user to a mesh router may be one or several hops. That is, a network user wants to transmit packets in multiple hops to a mesh router beyond his direct transmission range. In this case, network users cooperate with each other on relaying the packets to mesh routers. We further assume that all the network traffic has to go through a mesh router except the communication between two direct neighboring users. We assume so as it is probable that communications to and from a mesh router will constitute the majority of traffic in a WMN [12]. Moreover, this assumption would considerably reduce the routing complexity from the users' point of view as mesh routers will take the responsibility.

We assume that ***NO*** can always communicate with mesh routers through pre recognized secure channels, and so are mesh themselves. The WMN is assumed to be deployed with redundancy in mind so that revocation of individual mesh routers will not affect network connection. We assume the survival of an offline trusted third party (***TTP***), which is trusted for not disclosing the information it stores. ***TTP*** is essential only during the system setup. We further assume that there is a secure channel among ***TTP*** and each network user.

3.2 Threat Model and Security Requirements

Due to the open medium and spatially distributed nature, WMNs are susceptible to both passive and active attacks. The passive attacks include eavesdropping, while active attacks range from message relaying, bogus message injection, Phishing, active imitation to mesh router cooperation. Hence, for a practical threat model, we consider an adversary that is able to eavesdrop all network communications, as well as inject random fake messages. In addition, the adversary can compromise and control a small number of users and mesh routers subject to his option; it may also set up rogue mesh routers to phish user accesses. The purposes of the adversary contain 1) illegal and unaccountable network access, **2**) the privacy of genuine network users, and 3) denial-of-service (DoS) attacks against service accessibility.

In light of the above threat model, the following security requirements are necessary to make sure that a WMN functions correctly and strongly as purposed.

.User-router shared authentication and key agreement: A mesh router and a user should equally authenticate each other to stop both unauthorized network access and Phishing attacks. The user and the mesh router should also set up a shared pairwise symmetric key for session authentication and message encryption.

.User-user mutual authentication and key agreement: Users should also authenticate each other by cooperation in observing to message relaying and routing. Moreover, symmetric keys should be established and efficiently maintained to give session authentication and message encryption over the equivalent traffic.

.Sophisticated user privacy protection: The privacy of users should be well secluded, and we distinguish user privacy against dissimilar entities such as the adversary, ***NO***, and the law authority, as will be complicated in the next section.

.User accountability: In the cases of attacks and argument, the responsible users and/or user groups should be capable to be audited and pinpointed. On the other hand, no innocent users can be framed for disputes/attacks they are not concerned with.

.Membership maintenance: The network should be capable to handle membership dynamics with membership revocation, renewing, and addition.

.DoS resilience: The WMN should maintain service accessibility despite of DoS attacks.

4. AACT: ANONYMOUS, ACCOUNTABLE COMMUNICATION TOPOLOGY

When designing AACT, we find that none of the obtainable anonymous accountable cryptographic primitives, such as blind signature and group signature schemes, suits our purpose given the security and privacy requirements discussed above. Blind signature and group signature schemes can only give binding anonymously, while AACT demands user accountability, and hence, revocable anonymity. Existing group signature schemes do give revocable secrecy, but cannot support complicated user privacy. This inspiring us to tailor a group signature scheme by combining with onion ring strategy to convene all the necessities. AACT is then built on this onion ring based group signature difference by further integrating it into the authentication and key agreement protocol design.

4.1 AACT Key Management

The following setup operations are performed in an offline manner by all the entities in AACT, namely NO, a ***TTP,*** mesh routers, network users, and user group managers. AACT works under bilinear groups (G_1, G_2) with isomorphism ψ and respective generators g_1 and g_2, as in Section 2.1. AACT also employs hash functions H_0 and H, with respective ranges G_2^2 and Z_p. The notation below mainly follows [**8**].

NO is responsible for the key generation operation. Specifically, ***NO*** proceeds as follows:

1.Select a generator g_2 in G_2 uniformly at random and set $g_1 \leftarrow \psi(g_2)$. Select $\gamma \underset{\leftarrow}{R} Z_p^*$ and set $w = g_2^{\gamma}$.

1. Select $grp_i \underset{\leftarrow}{R} Z_p^*$

For a registered user group ***I.***

3. Using γ, generate an SDH tuple $(A_{i,j}, grp_i, x_j)$ by selecting $x_j \underset{\leftarrow}{R} Z_p^*$ such that $\gamma + grp_i + x_j \neq 0$, and setting $A_{ij} \leftarrow g_1^{1/(\gamma + grp_i + x_j)}$.

4. Repeat Step 3 for a prearranged number of times that are mutually agreed by ***NO*** and the user group manager GM_i.

5. Send $GM_i \{[i, j], grp_i, x_j) | \forall j\}$ via a secure channel.

6. Repeat Steps 2, 3, and 4 for every user group.

7. Send ***TTP:*** $GM_i \{[i, j], A_{i,j} \otimes x_j) | \forall i, j\}$ via a secure channel, where **0** denotes bitwise ***exclusive OR*** operation**.**

The above operation generates the group public key ***gpk*** and a number of private keys ***gsk:***

$$\begin{cases} gpk = (g_1, g_2, w) \\ \{gsk[i, j] = (A_{i,j}, grp_i, x_j) | \forall i, j\}. \end{cases}$$

Furthermore, ***NO*** obtains a set of revocation tokens, ***grt,*** with grt[i,j] $= A_{i,j}$ and also keep the mapping among group id ***i*** and grp_i for all user groups. Note that γ is the system secret only known to ***NO***. For the purpose of non denial, ***NO*** signs on Steps 5 and 7 under a standard digital signature scheme, such as ECDSA [13]. In AACT, we suppose that ECDSA-160 is used. For the same purpose, GM_i and ***TTP*** also sign on these messages upon receiving and send the resulted signature back to ***NO***.

Additionally, ***NO*** prepares every mesh router MR_k a public/private key pair, denoted by (RPK_k, RSK_k). Each mesh router also gets an accompanied public key

A certificate signed by NO to prove key authenticity. The signing key pair of NO is denoted by (NPK, NSK). The certificate has the following fields at the minimum:

$$Cert_k = \{MR_k, RPK_k, ExpT, Sig_{NSK}\},$$

Where ***ExpT*** is the expiration time and ***Sig,*** denotes an ECDSA-160 signature signed on a given message using a private key •.

Before accessing the WMN, a network user has to validate himself to his fit in user groups**.** For each such user group i, a network user uid_j is assigned a casual group private key as follows:

1. GM_i sends $uid_j(|i, j|, grp_i, x_j)$ as well as the related system parameters.

2. GM_i requests ***TTP*** to send $uid_j(|i, j|, A_{i,j} \otimes x_j)$ by providing the index [i, ***j***].

3. uid_j assembles his group private key as $gsk[i, j] = (A_{i,j}, grp_i, x_j)$**.**

Note that in our setting,

- GM_i only keeps the mapping of $(uid_j(|i, j|, grp_i, x_j))$ but has no knowledge of the corresponding $A_{i,j}$.
- ***NO*** only knows the mapping of $(GM_i, gsk[i, j])$ but has no knowledge about to whom ***gsk*** [***i, j***] is assigned.
- ***TTP*** has the mapping of $(uid_j(A_{i,j} \otimes x_j, grp_i))$ as it sends uid_j this information through a safe channel among the two upon the request from GM_i. But ***TTP*** has no

knowledge of the corresponding x_j or $A_{i,j}$.

Here, we use uid_j the user's necessary attribute information. For the purpose of non repudiation, uid_j signs on the messages it receives from GM_i and ***TTP*** under ECDSA-160, and sends back GM_i the equivalent signature.

4.2 User-Router Mutual Authentication and Key Agreement

To access the WMN, a network user follows the user-router common authentication and key agreement protocol as particular below, when a mesh router is within his direct communication range.

1. The mesh router MR_k first picks a random nonce $r_R \underset{\leftarrow}{R} Z_p^*$ and a random generator ***g*** in G_1 and then computes $g^{r_R}.MR_k$ further signs on $\boldsymbol{g}\, g^{r_R}$, and the current time stamp ts_1 , using ECDSA-160. ***MR_k*** then broadcasts

$$g, g^{r_R} ts_1, Sig_{RSK_k}, Cert_k, CRL, URL \qquad \textbf{\textit{(M.1)}}$$

As part of ***beacon message*** that is periodically broadcast to declare service existence. Here, ***CRL*** and ***URL*** denote the mesh router certificate revocation list and the user revocation list, respectively. Specifically, ***URL*** contains a set of revocation tokens that corresponds to the revoked group
private keys, which is a subset of ***grt***. Both ***CRL*** and ***URL*** are signed by ***NO***.
Upon receipt of (M.1), a network user uidj
proceeds as follows:
Check the time stamp ts$_{\mathbf{1}}$ to prevent replay attack.
Examine $Cert_k$ to confirm public key authenticity and the certificate expiration time; examine ***CRL*** and see if $Cert_k$ has been revoked by applying NPK. Further verify the authenticity of Sig_{RSK} by applying RPK_k .

Upon positive check results, uid_j believes that MR_k is legitimate and does the following:

Pick two random nonce $r, r_j \underset{\leftarrow}{R} Z_p^*$, compute g^{r_j} , and prepare the current timestamp ts_2 . Further get two generators $(\hat{u},\hat{v})$ in G_2 from H_0 as

$$(\hat{u},\hat{v}) \leftarrow H_0(gpk, g^{r_j}, g^{r_R} ts_2, r) \in G_2^2, \qquad \textit{(1)}$$

And compute their images
in $G_1 : u \leftarrow \psi(\hat{u}) and v \leftarrow \psi(\hat{v})$.

Compute $T_1 \leftarrow u^{\alpha} and T_2 \leftarrow A_{i,j} v^{\alpha}$ by selecting an exponent $\alpha \underset{\leftarrow}{R} Z_p .Set \delta \leftarrow (grp_i + x_j)\alpha \in Z_p$.

Pick blinding values r_α, r_x , and $r_\delta \underset{\leftarrow}{R} Z_p$.

Compute helper values R_1, R_2 , and R_3 :

$R_1 \leftarrow u^{r_\alpha}, R_2 \leftarrow e(T_2, g_2)^{r_x} .e(v, w)^{-r_\alpha} .e(v, g_2)^{-r_\delta}$,

and $R_3 \leftarrow T_1^{r_x} .u^{-r_\alpha}$. Compute a challenge value $c \in Z_p$ using ***H:***

$$c \leftarrow H(gpk, g^{r_j}, g^{r_R}, ts_2, r, T_1, T_2, R_1, R_2, R_3) \in Z_{p.}$$

Compute $s_\alpha = r_\alpha + c\alpha, s_x = r_x + c(grp_i + x_j)$ an d $s_\delta = r_\delta + c\delta \in Z_p$. Obtain the group signature on $\{g^{r_j}, g^{r_R}, ts_2\}$ as

$$SIG_{gsk[i,j]} \leftarrow (r, T_1, T_2, c, s_\alpha, s_x, s_\delta).$$

Compute the shared symmetric key with MR_k ***:***

$$K_{k,j} = (g^{r_R})^{r_j} .$$

Unicast back to MR_k

$$g^{r_j}, g^{r_R}, ts_2, SIG_{gsk[i,j]} . \quad \textbf{\textit{(M.2)}}$$

Upon receipt of (M.2), MR_k carries out the following to authenticate uid_j :

Check g^{r_R} and ts_2 make sure the freshness of (M.2).

Check that $SIG_{gsk[i,j]}$ is a valid signature by applying the group public key ***gpk*** as follows:
Compute $\hat{u}$ and $\hat{v}$ using (**1**), and their images
u and ***v*** in $G_1 : u \leftarrow \psi(\hat{u}) and v \leftarrow \psi(\hat{v})$.
Retrieve R_1, R_2 and R_3 as:

$$\tilde{R}_1 \leftarrow u^{s_\alpha} / T_1^c$$

$$\tilde{R}_2 \leftarrow e(T_2, g_2)^{s_z} .e(v, w)^{-s_\delta} .(e(T_2, w) / e(g_1, g_2))^c,$$

And $\tilde{R}_3 \leftarrow T_1^{s_z} .u^{-s_\delta}$.
Check that the challenge c is correct:

$$c \underset{=}{?} H(gpk, g^{r_j}, g^{r_R}, ts_2, r, T_1, T_2, \tilde{R}_1, \tilde{R}_2, \tilde{R}_3).$$

(2)
For each revocation token A ∈ URL, check
whether A is encoded in (T_1, T_2) by checking if

$$e(T_2 / A, \hat{u}) \underset{=}{?} e(T_1, \hat{v}). \quad (3)$$

If no revocation token of the URL is encoded in (T_1, T_2) , then the signer of $SIG_{gsk[i,j]}$ has not been revoked.

If all the above checks succeed, MR_k is now assured that the current user is a legitimate network user, although MR_k does not know which particular user this is. Note that uid_j is never disclosed or transmitted during protocol execution.

a. MR_k Further computes the shared symmetric key as $K_{k,j} = (g^{r_j})^{r_R}$ and sends back uid_j:

$$g^{r_j}, g^{r_R}, E_{K_{k,j}}(MR_k, g^{r_j}, g^{r_R}), \qquad \text{(M.3)}$$

Where E denotes the symmetric encryption of the given message within the brackets using key •.

The above protocol allows explicit mutual authentication among a mesh router and a genuine network user; it also enables unilateral anonymous authentication for the network user. Upon successful completion of the protocol, the mesh router and the user also create a shared symmetric key used for the succeeding communication session. And this session is uniquely identified through (g^{r_j}, g^{r_R}).

Remarks

Equation (2) holds because

$$\tilde{R}_1 = u^{s_\alpha} / T_1^c = u^{r_\alpha + c\alpha} / (u^\alpha) = u^\alpha = R_1.$$

$$\tilde{R}_2 = e(T_2, g_2)^{s_z}.e(v,w)^{-s_\alpha}.e(v,g_2)^{s_\delta \cdot}\left(\frac{e(T_2,w)}{e(g_1,g_2)}\right)^c = (e(T_2,g_2)^{r_z}.e(v,w)^{-r_\alpha}.e(v,g_2)^{s_\delta \cdot}).(e(T_2,g_2)^{grp_i+x_j}.e(v,w)^{-\alpha}.e(v,g_2)^{-(grp_i+x_j)\alpha}.\frac{e(T_2,w)}{e(g_1,g_2)}c$$

$$= R_2.\left(\frac{e(T_2v^{-\alpha}, wg_2^{grp_i+x_j})}{e(g_1,g_2)}\right)^c = R_2.\left(\frac{e(A_{i,j}, wg_2^{grp_i+x_j})}{e(g_1,g_2)}\right)^c = R_2.\left(\frac{e(g_1,g_2)}{e(g_1,g_2)}\right)^c = R_2.$$

$$\tilde{R}_3 = T_1^{s_z}u^{-s_\delta} = (u^\alpha)^{r_z + c(grp_i+x_j)}.u^{-r_\delta - c\alpha(grp_i+x_j} = (u^\alpha)^{r_z}.u^{-r_\delta} = T_1^{r_z}u^{-r_\delta} = R_3.$$

Equation (3) holds when there is an element A of URL encoded in (T_1, T_2) because of the following.

We know that ψ: $G_2 \to G_1$ is an isomorphism such that $\psi(g_2) = g_1$. According to the definition of isomorphism, we have $\psi(PQ) = \psi(P)\psi(Q)$ for any P, Q $\in G_2$. Using this property and mathematical induction, it is easy to know the following fact: For any natural number $m \in N, \psi(g_2^m) = g_1^m$.

Hence, if a group private key $(A_{i,j}, grp_i, x_j)$ with $A_{i,j} \in URL$ signed the group signature σ. For simplicity, let $\hat{u} = g_2^\alpha and \hat{v} = g_2^b$ for some integers a and b. On one hand,

$$e(T_2 / A_{i,j}, \hat{u}) = e(A_{i,j}v^\alpha / A_{i,j}, \hat{u}) = e(v^\alpha, \hat{u}) = e((\psi(\hat{v}))^\alpha, \hat{u}) = e((\psi(g_2^b))^\alpha, \hat{u}) = e((g_1^b)^\alpha, g_2^\alpha) = e(g_1, g_2)^{ab\alpha}.$$

On the other hand,

$$e(T_1, \hat{v}) = e(u^\alpha, \hat{v})) = e((\psi(\hat{u}))^\alpha, \hat{v}) = e((\psi(g_2^\alpha))^\alpha, \hat{v}) = e((g_1^a)^\alpha, g_2^b) = e(g_1, g_2)^{ab\alpha}.$$

Therefore, $e(T_2 / A_{i,j}, \hat{u}) = e(T_1, \hat{v}).$

4.3 User-User Mutual Authentication and Key Agreement In AACT

Adjacent genuine network users may help to relay each other's traffic. To this end, two network users within each other's direct communication range first authenticate each other and create shared secret pairwise key as follows:

uid_j picks a random nonce $r_j \underset{\leftarrow}{R} Z_p^*$ and computes where g^{r_j} is obtained from the inspirational messages broadcasted by the current service mesh router. uid_j further signs on g, g^{r_j}, and current time stamp ts1, using his group private key gsk[i,j] following Steps 2b(i) to 2b(iv), as in Section 4.2. uid_j Then locally broadcasts

$$g, g^{r_j}, ts_1, SIG_{gsk[i,j]}. \qquad \text{(M.1)}$$

2. Upon receipt of $(\tilde{M}.1)$, uid_I checks the time stamp and verifies the authenticity of $SIG_{gsk[i,j]}$ by applying the group key gpk following Step 3b, as in Section 4.2. uid_I further checks if the signature is generated from a revoked group private key following Step 3c, as in Section 4.2. Note that URL can always be obtained from the ***beacon messages.***

If all checks succeed, uid_I is assured that the current user it communicates with is legitimate. uid_I proceeds to pick a random nonce $r_I \underset{\leftarrow}{R} Z_p^*$ and computes $g^{r_I}.uid_I$ further signs on g^{r_j}, g^{r_I}, and current time stamp ts_2, using an appropriate group private key gsk[t, I] of his. uid_I also computes the shared pairwise session key as $K_{r_j,r_I} = (g^{r_j})^{r_I}$.

then replies uid_I

$$g^{r_j}, g^{r_I}, ts_2, SIG_{gsk[t,I].} \qquad (\tilde{M}.2)$$

3. Upon receipt of $(\tilde{M}.2)$, uid_j first delay window. uid_j checks whether $ts_2 - ts_1$ is within the acceptable delay window. uid_j also examines $SIG_{gsk[i,j]}$ and ***URL*** as uid_j did above. If all checks succeed, uid_j is also assured that its communicating counterpart is legitimate. uid_j Computes the shared pairwise session key as $K_{r_j,r_I} = (g^{r_I})^{r_j}$. uid_j Finally replies uid_I

$$g^{r_j}, g^{r_I}, E_{K_{r_j,r_I}} = (g^{r_I}, g^{r_I}, ts_1, ts_2). \qquad (\tilde{M}.3)$$

Upon receipt of $(\tilde{M}.3)$ and successful decryption of $E_{K_{r_j,r_I}} = (g^{r_I}, g^{r_I}, ts_1, ts_2)$. uid_I is assured that uid_j has successfully completed the authentication protocol and recognized the shared key for their subsequent communication session, which is uniquely identified through (g^{r_j}, g^{r_I}). This design of AACT protects user privacy in a complicated manner, while still maintaining user accountability.

4.5.1 User Anonymity against the Adversary, the User Groups, and *TTP*

In AACT, a user only authenticates himself as a genuine service subscriber without disclosing any of his identifying information by make use of the group signature method. Neither the adversary nor the user group managers can tell which meticulous user generates a given signature. The adversary, even by compromising mesh routers and other network users, that is, knowing a number of group private keys in addition to the group public key, still cannot infer any information concerning the meticulous group private key used for signature generation. This is due to the rigidity of the underlying q-SDH problem, where q is a 1,020-bit prime number. Due to the similar reason, neither a user group manager can distinguish whether or not one of his group members has signed a meticulous signature as he has no knowledge of the corresponding $A_{i,j}$s nor can he compute them. The same termination also holds for TTP as TTP can compute neither x_j nor $A_{i,j}$ given $A_{i,j} \otimes x_j$. Furthermore, each data session in AACT is identified only through pairs of fresh random numbers, which again discloses nothing concerning the user identity information. In addition, AACT needs a network user to refresh session identifiers and the shared symmetric keys for each different session. This further eliminates the ability to link among any two sessions initiated by the same network user. We note that even with the help of compromised mesh routers and other network users, the opponent still cannot judge whether two communication sessions are from the similar user. This is because, basically, none of them can tell whether two signatures are from the same user, given q- SDH problem and decision linear on G problem are hard.User Privacy against ***NO*** and User Accountability:Since *NO* knows grt, it can always tell which gsk[i, j] produces a given signature. However, *NO* has no knowledge about to whom gsk[i, j] is assigned as AACT allows a late compulsory among group private keys and network users. Furthermore, it is user group managers' sole responsibility to assign group private keys to every network user without any participation of NO. Therefore, *NO* could only map gsk[i, j] to the user group *i* based on grp_i. Because no other entities except *NO* and the key holder himself has the knowledge of the corresponding $A_{i,j}$, and can therefore, generate the given signature, the key holder must be a member of the user group i. This audit result serves us both necessities. On one hand, the result only discloses partial nonessential attribute information of the user and still protects user privacy to an extent. On the other hand, the result is adequate for user accountability purposes for ***NO.***

When ***NO*** (on behalf of mesh routers) finds a certain communication session disputable or suspicion, it conducts the following protocol to audit the responsible entity:

1. Given the link and the session identifier, find the equivalent authentication session message $(M.2) = g^{r_j}, g^{r_R}, ts_2, SIG_{gsk[i,j]}$ from the network log file.

2. For each revocation token $A_{i,j} \in grt$, check whether $e(T_2 / A_{i,j}, \hat{u}) \underset{=}{?} e(T_1, \hat{v})$. Output the first element $A_{i,j} \in grt$ such that $e(T_2 / A_{i,j}, \hat{u}) \underset{=}{?} e(T_1, \hat{v})$.

3. For the found revocation token $A_{i,j}$, output the corresponding mapping between $A_{i,j}$ and grp_i.

Since grp_i maps to a particular user group i, now a responsible entity has been found from the perspective of NO.

From the user's perspective, only part of his unneeded attribute information is disclosed from the audit. But such unneeded attribute information will not reveal his necessary attribute information. For example, the above audit may find that the dependable user is a member of Company XYZ but cannot reveal any other information about the user. Yet NO still has adequate proof to prove to Company XYZ that one of his members violates certain network access rule so that Company XYZ should take the corresponding responsibility specified in their service contribution agreement.Revocable User Anonymity against Law Authority: When law authority decides to track the meticulous attacker that is responsible for a certain communication session, the following procedure is taken: NO reports to the law authority $(A_{i,j}, grp_i)$ by executing the above protocol against the session in audit. $(A_{i,j}, grp_i)$ is then further forwarded to GM_i. GM_i Checks its local record, finds out the mapping between $(grp_i andx_i)$, and hence, the corresponding user uniqueness information uid_j, to whom gsk[i,j] is assigned during the system setup. GM_i then replies uid_j to the law authority. At this point, law authority and only law authority get to know about which particular user is conscientious for the communication session in the audit. We point out that this tracing procedure has the non denial property because 1) GM_i signed on all *gsks* that are assigned from NO as the proof of receipt; 2) uid_j also signed on the messages when obtaining gsk[i, j] from GM_i and TTP as the proof of receipt. AACT also not able to frame because no one else knows gsk[i, j] except NO and uid_j or is able to forge a signature on **behalf of** uid_j **.**

5 PERFORMANCE ANALYSIS OF AACT

5.1 System Security Analysis

As its basic security functionality, AACT enforces network access control. Hence, we are the majority concerned with the following three different types of attacks, i.e., Bogus data injection attacks, data Phishing attacks, and DoS attacks.

Bogus data injection attacks: In such attacks, the opponent needs to inject bogus data to the WMN aimed at using the network service for free. The sources of the bogus data could be outsiders, revoked users, or revoked mesh routers.

Data phishing attacks: In such attacks, the opponent may set up bogus mesh routers and try to phish user connections to such routers. In this way, the opponent could control network connection and analyze users' data traffic for their benefits. The Phishing mesh routers can be either completely new mesh routers or revoked mesh routers both at the adversary's control.

DoS attacks: In such attacks, the opponent may flood a huge number of illegal access request messages to mesh routers. The purpose is to exhaust their resources and render them less capable of serving legitimate users. In AACT, for every access request message (M.2), the corresponding mesh router has to confirm a group signature and check the validity of the signer. Both operations involve costly pairing operations, which, hence, can simply be exploited by the opponent. To deal with this issue, we assume the same client-puzzle approach as adopted in [18]. The idea of this approach is as follows: When there is no proof of the attack, a mesh router process (M.2) usualy. But, when under a suspected DoS attack, the mesh router will attach a cryptographic puzzle to every (M. **1**) and need the solution to the puzzle be attached to every (M.2). The mesh router commits resources to process (M.2) only when the solution is correct. Typically, solving a client puzzle needs a brute-force search in the solution space, while the solution conformation is trivial [18].

5.2 User Privacy and Accountability Analysis

AACT protects user privacy in a complicated manner, while still maintain user's responsibility. First, AACT enables user anonymity against the opponent, the user group managers, and TTP. In AACT, a network user only authenticates himself as a genuine service subscriber without disclosing any of his identity information by using the group signature method. Neither the opponent nor the user group managers can tell which meticulous user generates a given signature. The adversary, even by compromising mesh routers and other network users, that is, knowing a number of group private keys in addition to the group public key, still cannot deduce any information about the particular group private key used for signature generation. This is due to the rigidity of the underlying q-SDH problem, where q is a 1,020-bit prime number. Due to the same reason, a user group manager also cannot differentiate whether or not one of his group members has signed a particular signature as he has no knowledge of the corresponding $A_{i,j}$ s nor can he compute them. The same finish also holds for TTP as TTP can compute neither Xj nor $A_{i,j}$ given $A_{i,j} \otimes x_j$. Furthermore, every data session in AACT is recognized only through pairs of fresh random numbers, which again discloses nothing about user identity information. In addition, AACT requires a network user to refresh session identifiers and the shared symmetric keys for every different session. This further eliminates the linkage among any two sessions originated from the same network user. We note that even with the help of compromised mesh routers and other network users, the adversary still cannot judge whether two communication sessions are from the same user. This is because, basically, none of them can tell whether two signatures are from the same user, given q- SDH problem and decision linear problems on G_1 are hard.AACT gives adequate user privacy protection against NO while maintaining user accountability.

7. Conclusion

In this paper, we proposed AACT, which, to the most excellent of our knowledge, is the first attempt to set up an liable security framework with a complicated user privacy protection model tailored WMNs. We tailored group signature scheme[8] that combined with onion ring strategy [31]. We then built AACT on this new model by further integrating it into the authentication and key agreement protocol design. On one hand, AACT enforces strict user access control to cope with both free riders and spiteful users. On the other hand, AACT offers complicated user privacy protection against both adversaries and different other network entities. Our analysis showed that AACT is elastic to a number of security and privacy related attacks. Additional methods were also discussed to further improve the scheme efficiency

Acknowledgments

Insert acknowledgment, if any. The preferred spelling of the word “acknowledgment” in American English is without an “e” after the “g.” Use the singular heading even if you have many acknowledgments. Avoid expressions such as “One of us (S.B.A.) would like to thank ... .” Instead, write “F. A. Author thanks ... .” Sponsor and financial support acknowledgments are also placed here.

References

[1] K. Ren and W. Lou, "A Sophisticated Privacy-Enhanced Yet Accountable Security Framework for Wireless Mesh Networks," Proc. 28th Int'l Conf. Distributed Computing Systems (ICDCS '08), June 2008.

[2] I.F. Akyildiz, X. Wang, and W. Wang, "Wireless Mesh Networks: A Survey," Computer Networks, vol. 47, no. 4, pp. 445-487, Mar. 2005.

[3] "Self Organizing Neighborhood Wireless Mesh Networks," http://www.research.microsoft.com/mesh/ , 2009.

[4] Y. Zhang and Y. Fang, "A Secure Authentication and Billing Architecture for Wireless Mesh Networks," ACM Wireless Networks, to be published.

[5] Y. Zhang and Y. Fang, "ARSA: An Attack-Resilient Security Architecture for Multi-Hop Wireless Mesh Networks," IEEE J. Selected Areas in Comm., vol. 24, no. 10, pp. 1916-1928, Oct.2006.

[6] "The Wimax Forum,"http://www.wimaxforum.org. 2009.

[7] "Boston Suburb Secures Metro-Scale Wireless Mesh Network with Bluesocket," http://www.tmcnet.com/usubmit/2006/09/27/ 1936581.htm, Sept. 2006.

[8] D. Boneh and H. Shacham, "Group Signatures with Verifier-Local Revocation," Proc. ACM Conf. Computer and Comm. Security (CCS), pp. 168-177, 2004.

[9] D. Chaum and E. van Heyst, "Group Signatures," Proc. Conf. Eurocrypt, pp. 257-265, 1991.

[10] R. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," Comm. ACM, vol. 21, no. 2, pp. 120-126, 1978.

[11] M. Jakobsson, J. Hubaux, and L. Buttyan, "A Charging and Rewarding Scheme for Packet Forwarding in Multi-Hop Cellular Networks," Proc. Seventh Int'l Conf. Financial Cryptography (FC), 2003.

[12] N. Salem, L. Buttyan, J. Hubaux, and M. Jakobsson, "A MicroPayment Scheme Encouraging Collaboration in Multi-Hop Cellular Networks," Proc. ACM MobiHoc, 2003.

[13] D. Hankerson, A. Menezes, and S. Vanstone, Guide to Elliptic Curve Cryptography. Springer-Verlag, 2004.

[14] Y. Zhang, W. Liu, and W. Lou, "Anonymous Communications in Mobile Ad Hoc Networks," Proc. IEEE INFOCOM, Mar. 2005.

M.Madhavi is working as Assoc.Prof in Anurag Engineering College,Kodad. She is the life member of ISTE,member of CSI.She has presented a paper on stegnography in National Level Conference, Published a paper on cloud computing in international journal. She completed B.Tech and M.tech in Computer Science Engineering.She has Organized National level workshop on Design and Analysis of Algorithms.

M.Swetha has completed her B.tech in ADAM’s Engineering . College. Organized a state level student technical fest named as "NEXUS' 09".Participated a workshop on "image processing and pattern recognition"organized by ADAM’s Engineering . College. She is pursuing M.Tech in Anurag Engineering College.