A Distributed Key Establishment Scheme for Wireless Mesh Networks using Identity-Based Cryptography*

Duygu Karaoglan
Sabanci University
Istanbul, Turkey
duyguk@sabanciuniv.edu

Albert Levi
Sabanci University
Istanbul, Turkey
levi@sabanciuniv.edu

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

ABSTRACT

In this paper, we propose a secure and efficient key establishment scheme designed with respect to the unique requirements of Wireless Mesh Networks. Our security model is based on Identity-based key establishment scheme without the utilization of a trusted authority for private key operations. Rather, this task is performed by a collaboration of users; a threshold number of users come together in a coalition so that they generate the private key. We performed simulative performance evaluation in order to show the effect of both the network size and the threshold value. Results show a tradeoff between resiliency and efficiency: increasing the threshold value or the number of mesh nodes also increases the resiliency but negatively effects the efficiency. For threshold values smaller than 8 and for number of mesh nodes in between 40 and 100, at least 90% of the mesh nodes can compute their private keys within at most 70 seconds. On the other hand, at threshold value 8, an increase in the number of mesh nodes from 40 to 100 results in 25% increase in the rate of successful private key generations.

Categories and Subject Descriptors

C.2.0 [**Computer Communication Networks**]: General—*Security and protection*; C.2.1 [**Computer Communication Networks**]: Network Architecture and Design—*Wireless communication*; E.3 [**Data Encryption**]: [Public key cryptosystems]

General Terms

Design, Experimentation, Performance, Reliability, Security

Keywords

Wireless Mesh Networks, Key Establishment, Identity-Based Cryptography, Threshold Secret Sharing

*This work is supported by Scientific and Technological Research Council of Turkey (TÜBİTAK) under grant 104E071.

Q2SWinet'10, October 20–21, 2010, Bodrum, Turkey.

1. INTRODUCTION

Wireless Mesh Metworks (WMNs) are wireless networks in which the nodes are able to carry out mesh routing by utilizing multi hop communication. They are dynamically self-organized, self-healed and self-configured; meaning that the mesh nodes form a network on the fly. Furthermore, they offer both low-cost and high-speed network services for the end users. Along with the ease of their deployment, they provide mobility, flexibility, robustness and increased coverage with an effective level of scalability. To have those advantages, utilization of WMNs is preferred in the areas that do not have wired infrastructure or are in terrain of difficult deployment.

WMNs are enclosed with mesh routers and mesh clients, where the infrastructure, given in Figure 1, shows a cooperation in wireless communication carried out among a number of mesh nodes [17]. The difference between these two types of mesh nodes is not only in their mobility, being the mesh routers stationary while the mesh clients either stationary or mobile, but also in the energy consumption constraints they have. Mesh clients are known to be more limited in energy consumption. Therefore, the load of functionalities that require high computational power and bandwidth can be burdened on the mesh routers. Additionally, at any time, any node can either join or leave the network without affecting network functionality.

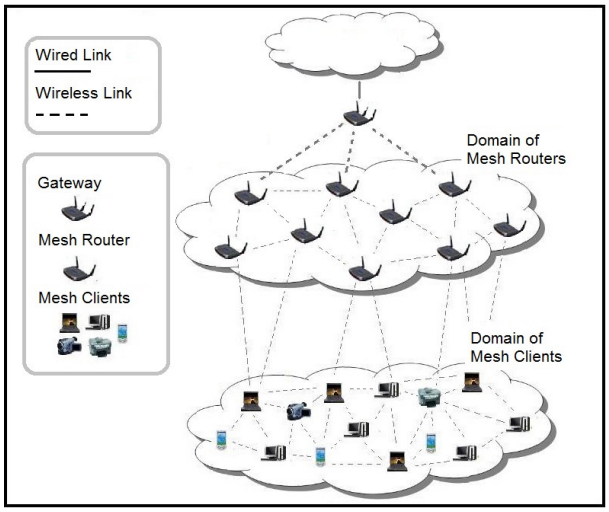

Figure 1: Infrastructure of a WMN

Nevermore, multi hop communication and the nature of wireless channel make WMNs prone to both passive and active attacks. In a WMN, a passive attack will result in violation of confidentiality whereas an active attack will compromise resiliency, integrity, authentication and non-repudiation [17]. Thus, communication security between the mesh nodes is an important research area. In order to maintain mutual trust and secure communication among the mesh nodes, a key establishment service must be provided. Considering the network being dynamically self-organized and self-configured along with the lack of complete central administration, standard methods such as Public Key Infrastructure (PKI)-based schemes are inapplicable in establishing keys within WMNs.

Essentially, Identity-based Cryptography (IBC) seems to be a more efficient approach for WMNs since it eliminates the certificate based public key distribution indispensible in the conventional PKI-based schemes. It basically avoids the need for users to generate private keys and to distribute them throughout the network, which reduces the computation necessary to join the network. Additionally, in IBC, users only need to propagate their identities, which is typically included in the messages, instead of propagating both the public keys and the signatures on them as it is done in the traditional PKI-based schemes. This reduces the utilized network bandwidth considerably. However, IBC assumes a trusted third party (TTP) to generate and distribute the private keys of the users, which does not fit with the characteristics of WMNs. Additionally, using a TTP in a security providing protocol is neither rational nor practical due to the fact that such a system will be prone to single point of failure. Therefore, it can be said that the limitations of conventional solutions necessitate the development of a brand-new security architecture to cope with the unique requirements of WMNs [1].

1.1 Related Work

Zhang and Fang [21, 20] propose UPASS/ARSA scheme, a secure authentication and billing architecture to enable an omni-present network with faultness roaming. Their trust model is built upon both PKI and IBC, which is not practical since they force the users to perform both protocols' operations that require high computational power of different types. Besides, ISA scheme proposed by Li [12] defines a good key revocation method that provides an efficient network access based on IBC with the assumption of the gateway router being the TTP.

Protocols mentioned above regarding secure key management assume a trusted authority. In practice, it is not very feasible to make such an assumption because of the hardness of maintaining such a server safely and keeping it available all the time. In order to eliminate that assumption, threshold secret sharing is used in [22] and [8]. Zhou and Hass [22] present a key management protocol based on the conventional PKI-based schemes, in which a group of nodes share the role of the Certification Authority (CA). As in (n, k)-threshold schemes [15], any k partially signed certificates can collaboratively construct a signed certificate which befits to a CA signed certificate. A similar approach is proposed by Kong [8], in which the RSA certificate signing key is distributed among all the nodes of the network. These two schemes, [22] and [8], differ only in the name of the number of shareholders, but in both, shares of the certificate signing keys are generated and distributed by a TTP. Thus, they do not provide a fully distributed key management.

On the other hand, Deng and Agrawal [7] propose a mechanism to secure the Dynamic Source Routing (DSR) protocol for ad-hoc networks in which the trusted authority is fully eliminated by the utilization of IBC along with threshold secret sharing. Their problem is application specific and so is their solution. Khalili et al. [10] and Deng et al. [5] also use IBC with threshold secret sharing but to secure the management of the keys for ad-hoc networks by offering the nodes to generate the shares and the secret collaboratively, which is an application free solution. Both of those mechanisms enable flexible and efficient key management for ad-hoc networks and are fully distributed. The difference is that [10] gives general information on the system with the options to be selected for the public keys of the users and the IBC to be utilized whereas [5] describes a specific solution and evaluates it. However, those schemes cannot be applied directly to WMNs as discussed in the subsection below.

1.2 Our Contributions

We propose a secure and efficient key establishment protocol by customizing the work in [5] for the sake of the significatives of WMNs. In their scheme, all of the nodes come together in a coalition so that they generate the requisite shares, which has two important disadvantages in comparison with the characteristics of WMNs:

1. *Large transmission delay*: number of users that collaboratively compute the master private key directly affects the amount of utilized network bandwidth. If we assume that n users are in such a coalition, due to the utilized secret sharing scheme described in Section 2.2.2, at least $n \times (n-1)$ packets needs to be transmitted among the nodes.

2. *Number of collaborative nodes dependent network resiliency*: due to the fact that any k nodes can collaboratively compute any other node's private key, network is tolerant to $k-1$ compromised nodes, where k is the threshold value. Thus, resiliency of the network can only be increased by increasing the value of k, which is infeasible since that value determines the required number of the neighboring nodes by protocol definition.

WMNs' characteristics provide us a way to centralize the network to an extent by which we can ameliorate the above-mentioned disadvantages. As discussed above, mesh routers can be distinguished by the parameters they hold and by the operations they perform. Thus, we imposed the burden of the master key generation on them, which decreased the number of nodes present in that phase. Besides, we assumed that it is harder to compromise the mesh routers than the mesh clients. With this assumption, we increased the number of shares needed in the reconstruction process by increasing the number of shares that the mesh routers hold. As a consequence, resiliency of the system is increased without increasing the number of required neighboring nodes.

The rest of the paper is organized as follows. Section 2 gives necessary cryptographic background and Section 3 describes our proposed solution in detail. In Section 4, resiliency of our proposed solution is analyzed while in Section 5 its simulative performance evaluation is discussed. Finally, we conclude our work in Section 6.

2. BACKGROUND INFORMATION

In the following subsections Identity-based Cryptography (IBC) and Secret Sharing schemes are explained in detail.

2.1 Identity-based Cryptography (IBC)

The concept of IBC is introduced by Adi Shamir [16] in 1985. The basic idea is that the public key of a user can be an arbitrary string (i.e. IP address, e-mail address, name, etc.) that uniquely identifies him in such a way that the denial is impossible. Many schemes has been proposed regarding IBC, which can be examined in detail from [4], [19], [11] and [3], where [4] is based on quadratic residues while the others use pairing operation defined over Elliptic Curve Cryptography (ECC)[1]. We preferred the IBC scheme proposed by Boneh and Franklin [3] in which Weil Tate Pairing is utilized as the bilinear mapping on ECC, as it has a performance comparable to ElGamal encryption and the chosen cipher text security in the random oracle model.

In ECC based IBC, master key, which is a public-private key pair, is generated by a trusted authority, known as the Private Key Generator (PKG). The public part of the key is assumed to be known by every user while the private part is kept secret in the PKG. When a user demands his private key, PKG computes and delivers it to the requestor. After delivering the private key, PKG does not involve in any other operation. Thus, the network does not need to be centralized and the solution is applicable for closed groups of users [16].

IBC consists of four phases:

1. *Setup*: Global parameters and the master key of the system are generated by the PKG. In our constructions, we used finite field $\mathcal{F}_q$, where q is a sufficiently large prime. The master private key, MK^{priv}, is randomly selected and the master public key is computed as $MK^{pub} = MK^{priv} \times P$, where P is the generator of elliptic curve group.

2. *Extract*: PKG uses the master private key and the public key of the requestor to construct the user's private key. Assuming that the user's public key is $Q_{ID,i} = H_1(ID_i)$, where H_1 is a hash function, its private key is then computed as $PK_i = MK^{priv} \times Q_{ID,i}$.

3. *Encryption*: Plaintext is encrypted using receiver's public key, $Q_{ID,receiver}$.

4. *Decryption*: Ciphertext is decrypted using receiver's private key, $PK^{receiver}$.

One of the most important features of IBC is that the sending party does not need to have its private key to be able to send a message since it does not need for receiving party's certificate. Likewise, the receiving party does not need to have its private key to be able to receive a message. Besides, exchanged messages remain secret as long as an adversary does not have either of the master private key or the user private keys. Additionally, obtaining a number of users' private keys does not help to break another user's private key; thus the security of the system is ensured.

[1]In ECC, the difficulty is based on the difficulty of elliptic curve discrete logarithm problem and it has almost the same cryptographic security as 1024-bit key length used in RSA [18].

2.2 Secret Sharing

Secret Sharing is a method that allows a secret to be distributed among a group of users, in such a way that no single user can deduce the secret from his share alone. The secret cannot be reconstructed unless a certain condition is met, and that condition is generally a coalition of a sufficient number of shareholders. All the secret sharing schemes uses a field structure, for which we use $\mathcal{F}_q$, where q is a prime.

2.2.1 Additive Secret Sharing (AdSS)

In AdSS schemes, reconstruction is performed by adding up all the shares, thus, it is impossible to reconstruct the secret unless all the shareholders collaborate. AdSS assumes the existence of a TTP by whom the shares are generated and transmitted securely[2] to the corresponding shareholders. What TPP performs is to choose a large prime q, a secret $s \in \mathcal{F}_q$ and $n-1$ random numbers $s_1, s_2, \ldots, s_{n-1}$ to be the shares of the secret. Then he computes the last share of the secret by Equation 1 and sends the shares s_i to the corresponding shareholders. The reconstruction of the secret is then performed as all the shareholders come together in a coalition and evaluate Equation 2.

$$s_n = s - \sum_{k=1}^{n-1} s_k \pmod q \tag{1}$$

$$s = \sum_{i=1}^{n} s_i \pmod q \tag{2}$$

2.2.2 Threshold Secret Sharing (ThSS)

In ThSS schemes, reconstruction is performed by any subset of k users, but no subset of smaller size. These schemes are also known as (n, k)-ThSS schemes. They are secure against active adversaries for $k < n/2$ and against passive adversaries for $k < n$.

Shamir's ThSS.

One of the widely used ThSS schemes is proposed by Adi Shamir [15] in 1979, which is based on the Lagrange Interpolation Polynomial[3]. The existence of a TTP is also assumed in Shamir's ThSS scheme, whose role is to generate and distribute the shares. TTP first chooses a large prime q, a secret $s \in \mathcal{F}_q$ and a polynomial $f(z)$ of degree $k-1$, such that $f(0) = s$. Then he evaluates the polynomial for each user to generate their shares, s_i, via Equation 3 and sends them to the corresponding shareholders. As for the reconstruction of the secret, k of the shareholders combine their shares performing the calculations in Equations 4 and 5.

$$s_i = f(i) \pmod q \tag{3}$$

$$s = \sum_{j=1}^{k} s_j l_j(0) \pmod q \tag{4}$$

[2]The trusted authority is assumed to be powerful enough to establish a secure pairwise communication link with every shareholder.

[3]It is a linear polynomial interpolation, in which given a set of k data points in the 2-dimensional plane (x_i, y_i), there is one and only one polynomial $f(x)$ of degree $k-1$ such that $f(x) = y_i$ for all i for distinct values of x_i's [15].

$$l_j(0) = \prod_{i=1,\, i\neq j}^{k} \frac{i}{j-i} \pmod q \tag{5}$$

Shamir's ThSS without a Trusted Authority.

The problem of Shamir's ThSS stems from the assumption of a TTP, which can be eliminated by the idea of the nodes being collaboratively computing the secret s. Each node contributing to the generation of the secret has an equal influence on its value. For the collaborative key generation, each node N_i selects a secret x_i and a polynomial $f_i(z)$ of degree $k-1$, such that $f_i(0) = x_i$, generates the shares $x_{i,j}$ of x_i as in Equation 3 and sends them to the corresponding node N_j, where$j = 0, 1, 2, \ldots, n$. When node N_i receives $n-1$ of $x_{j,i}$'s, it can compute its shared secret via Equation 6. The computed share is equivalent to the share distributed by the TTP in a (n, k)-ThSS. Therefore, with the collaboration of k shareholders, the secret can be reconstructed as it is done in the (n, k)-ThSS scheme.

$$s_i = \sum_{j=1}^{n} x_{j,i} \pmod q \tag{6}$$

Variations on ThSS.

The abovementioned ThSS schemes consider splitting the secret s among n users by giving each of them one share. However, we might have different levels of trust for different users or we might want to make some of the users more important than the others. In such a situation, one way of handling this is to give a larger number of shares to the users we trust more: if we give x shares to the trusted users, then we give y shares to the others, where $x > y$. Thus, the scheme becomes a $(ax+by, k)$-ThSS, where a is the number of users that we trust more and b is the number of regular users. Another approach is to share the secret additively among two groups whereby the additive shares are shared again with a ThSS scheme within each group. To be more precise, let us assume that we have $n = n_1 + n_2$ users for the share to be distributed. Let the secret be $s = s_1 + s_2$ with s_1 being shared in a (n_1, k_1)-ThSS fashion among the first group and s_2 being shared in a (n_2, k_2)-ThSS fashion among the second group. Then, k_1 users from the first group and k_2 users from the second group need to collaborate in order to reconstruct the secret s.

3. PROPOSED METHOD: DISTRIBUTED KEY ESTABLISHMENT (DKE)

Our approach is composed of three phases: master private key share generation, master private key share distribution and user private key generation. First phase consists of collaborative generation of the master private key shares performed by the mesh routers. In the second phase, generated master private key shares are distributed to the mesh clients. As soon as a mesh client receives its master private key share, it can also contribute to this process. Last phase provides a private key generation service, by which each mesh node can obtain its private key. This service is carried out by a collaboration of a defined number of mesh nodes, determined by the threshold value.

Table 1: Symbols used in Protocol Definition

Number of mesh nodes	n
Number of mesh routers	m
Number of mesh clients	l
Number of shares for mesh routers	x
Threshold value	k
A mesh node	MN
A mesh router	MR
A mesh client	MC
Secret	s
Subshare of a secret	ss
Master public key	MK^{pub}
Master public key share	MKS^{pub}
Master private key	MK^{priv}
Master private key share	MKS^{priv}
Master private key partial share	$MKPS$
User public key	Q
User private key	PK
User private key share	PKS

In the following subsections, we give detailed information on the phases just after defining our assumptions. Symbols used in the protocol definition can be found in Table 1.

3.1 Assumptions

Mesh nodes, especially the mesh routers, do not collude to reveal any other mesh node's private key. Our security solution does not rely on the existence of a trusted authority and there is no pre-defined mutual trust among the mesh nodes. All the keys are generated collaboratively by the mesh routers and distributed accordingly to the mesh clients. What we assume here is that the mesh nodes will not misbehave on their own or conspire with either of the parties unless they are captured. Thus, we measure the resiliency of the network only against the adversaries.

It is harder to compromise the mesh routers and they are arranged in a specific way to cover the network area. Mesh routers are the mesh nodes that form the backbone of WMNs; we know that they are there, for sure. Moreover, the mesh routers are deployed in such a way that they cover the network area in order to maintain continous connectivity. Thus, there should be a design in the placements of the mesh routers. At this point, we assume that the physical locations of the mesh routers are selected in such a way that physical capture of these nodes becomes hard. For example, the mesh routers can be placed at the top of street lamps or at the roofs of properties where the physical capture requires an effort equals to break-in.

Identities of the mesh nodes are unique and each node has a mechanism to discover its one-hop neighbors. As in all IBC schemes, there is the assumption of the identity of the node being unique, since they are used as the public keys of the users. In order to easily overcome this uniqueness issue, the identities of the nodes are selected to be their addresses. This can be simply obtained through dynamic address allocation such as DHCP (Dynamic Host Configuration Protocol) where a centralized server ensures uniqueness of the addresses.

Communication among the mesh nodes is limited to neighbourhood. An adversary can simply decrease the bandwidth share by increasing the number of hops in a route between the source and destination nodes that a packet will traverse [2, 9]. In order to prevent this type of action, thus to improve the capacity of the network, a node should only communicate with nearby nodes as the analytical upper and lower bounds of a network capacity implies [6]. This is accompolished by broadcasting except for the first phase of our protocol.

3.2 Master Private Key Share Generation

In this phase, master private key, MK^{priv}, is collaboratively computed by all of the mesh routers. Thus, the total number of shares present in the system depends on the number of shares that the mesh routers hold, x, and the number of mesh routers, m, yielding a total of $m \times x$ shares to be distributed among the nodes of the network. In our scheme, AdSS is also applied along with $(m \times x,\, k)$-ThSS: the master private key of the network is defined as $MK^{priv} = MK^{priv,\,1} + MK^{priv,\,2}$, where $MK^{priv,\,1}$ is known by all the mesh routers while $MK^{priv,\,2}$ is shared among the mesh nodes in a $(m \times x,\, k)$-ThSS fashion.

Setup phase of ECC based IBC without master key generation process and the decision of the threshold value is performed before any other operation. It can either be hardcoded to the mesh nodes or a negotiation protocol might be used in between the mesh routers to decide on the parameters that will be used. Here, we preferred the first option for simplicity. Just after the process of realizing the parameters, the very first thing that the mesh routers perform is the computation of the network's master key. For master private key share generation, each mesh router MR_i computes subshares $ss_{i,\,j,\,a}$, as described in Section 2.2.2 and sends them to the corresponding mesh router MR_j, where $j = 1, 2, \ldots, m$ and $a = 1, 2, 3, \ldots, x$. As a mesh router MR_i receives $(m-1) \times x$ subshares, it computes its master private key share via Equation 7, where $l_i(0)$ is the Lagrange coefficient computed via the Equation 5. Additionally, withholding its master private key share, each MR_i computes its master public key share via Equation 8 and publishes it.

$$MKS_i^{priv} = \sum_{a=1}^{x} (\sum_{j=1}^{m} ss_{j,\,i,\,a} \pmod q) \times l_i(0) \qquad (7)$$

$$MKS_i^{pub} = MKS_i^{priv} \times P \qquad (8)$$

In order for a mesh router to compute the actual value of the master public key, it needs to hold $(m-1) \times x$ master public key shares and the reconstruction is performed via Equation 9. This equation corresponds to reconstructing the additively shared master public key whose one of the additive parts is shared among the mesh nodes in a threshold manner. Thus, first we combine the shares coming from k mesh routers MR_j and then we combine this result with the share of its own.

$$MK^{pub} = (\sum_{i=1}^{k} MKS_i^{priv} \times l_i(0) \times P) + (MK_j^{priv,\,1} \times P) \qquad (9)$$

3.3 Master Private Key Distribution

For every mesh node, second phase starts as soon as a mesh client recognizes that one of its neighboring mesh nodes finished computing its master key share. When a mesh client receives a broadcast message consisting of the master public key share, it generates a request message for the distribution of the master private key share. Upon receiving that request, the neighbouring mesh node MN_j computes the master private key partial share of the requesting mesh client MC_i via Equation 10 , where $l_j(i)$ is the Lagrange coefficient computed via the Equation 5, and sends it to MC_i. As the requesting mesh client MC_i receives sufficient number of such shares, it computes its master private key share by simply adding up all the received partial shares as in Equation 11. Additionally, MC_i reconstructs its master public key share as in Equation 8 using the information gathered from the broadcast messages of the previous step.

$$MKPS_{j,\,i} = MKS_j^{priv} \times l_j(i) \qquad (10)$$

$$MKS_i^{priv} = \sum_{j=1}^{k} MKPS_{j,\,i} \pmod q \qquad (11)$$

3.4 User Private Key Generation

After a mesh node finishes computing its master private key share, it can make use of the private key generation service. For a mesh node MN_i to reconstruct its private key, it broadcasts a private key generation service request message. Upon receiving such a request, a neighbouring mesh node MN_j computes the user private key share for MN_i via Equation 12, where Q_j is the public key of the requesting node. Once the requesting node MN_i receives sufficient number shares, it can reconstruct its private key as in Equation 13. This equation corresponds to reconstructing the additively shared user private key whose one of the additive parts is shared among the mesh nodes in a threshold manner. Thus, first we combine the shares coming from k mesh nodes MN_j and then we combine this result with the share coming from a mesh router.

$$PKS_{j,\,i} = MKS_i \times Q_j \qquad (12)$$

$$PK_i = (\sum_{j=1}^{k} PKS_{j,\,i} \times l_j(0)\,) + MK_p^{priv,\,1} \times Q_j \qquad (13)$$

3.5 Timeout Method

The most outstanding characteristic of the reconstruction operations is that if a mesh node does not have sufficient number of neighboring nodes, it simply can compute neither the master key shares nor its private key. However, a situation as the following may also occur: a packet sent by a mesh node consisting of a service request drops due to collisions. As a result, that mesh node cannot compute either of the keys in spite of having sufficient number of neighboring nodes.

In order to overcome such a problem, a timeout method is adopted. In this method, after sending a service request for either master key share or user private key computations, a mesh node sets a timer in correspondance with that request.

For the master private key share computation performed by the mesh routers, each mesh router should receive $m-1$ subshares to compute the master private key share of their own, where m is the number of mesh routers. Therefore, a mesh router keeps sending request packets periodically until it receives all the desired data. As for either master private key share reconstruction performed by the mesh clients or master public key share and user private key computations performed by the mesh nodes, a mesh node needs k corresponding shares. However, a mesh node might not has that many neighbours. Thus, when there is a doubt on the reception of the demanded data, mesh nodes repeat their requests periodically only a number of times.

4. RESILIENCY ANALYSIS

The resiliency of a network is the maximum number of compromised nodes by which the security of the network is not affected. If an adversary compromises a number of mesh nodes holding a sufficient number of shares of the master private key, then he can compute all the user private keys. Thus, the resiliency of the network can be increased by increasing the threshold value.

As described in Section 3.2, our scheme ensures that a mesh router must always contribute to any of the reconstruction processes. Therefore, in order for an adversary to be successful, capturing a mesh router is a must. In other words, as long as a mesh router is not compromised, no matter how many mesh clients are captured, the resiliency of the network is conserved. On the other hand, if a mesh router is compromised, then the network is still resilient to some extent as discussed below. Suppose we use $(m \times x,\, k)$-ThSS scheme, where m is the number of mesh routers each of which has x shares of the master private key. In such a scenario, an adversary must capture a number of nodes wihholding a total of at least k shares of the master private key in order to reconstruct the master private key of the network. As a consequence, the resiliency of the network is conserved even if an adversary compromises q mesh routers and p mesh clients satisfying $k < (q \times x) + p$.

5. PERFORMANCE EVALUATION

We simulatively analyze the performance of our proposed scheme. In this section, we first give simulation environment and performance metrics and then, we discuss the results obtained.

5.1 Simulation Environment and Performance Metrics

We used Network Simulator 2 (ns2) [14] version 2.33 to evaluate the performance of our proposed scheme. For the simulation scenario, we modeled the network as having $n = 30, 40, 50, \ldots, 100$ nodes within an area of 2000×2000 square meters. We simulated the performance of the network for the threshold values $k = 2, 4, 6, 8, 10, 12$. As we make the assumption that the mesh routers cover the network area, we have 25 mesh routers. Each mesh router has 2 shares of the master private key and they are deployed in grid manner as to cover the network area. In this deployment model, each mesh router is in the transmission range of its neighboring mesh routers. On the other hand, mesh clients are deployed within the area using bivariate uniform random distribution.

All the simulations are run on a personal computer with the following configuration: Windows Vista (32-bit), Intel Core 2 Duo T5450 Processor at 1.66 GHz, 2 GB RAM and GCC 4.3.3 on Cygwin 1.5.25-15. Morever,for the configuration of ns2, we used MAC and network interface types of 802.11p [13], omni-directional antenna with a transmission range of 375 meters, priority queue defined under drop tail queue, Dynamic Source Routing and Transmission Control Protocol.

We consider two metrics in our performance model:

1. *Latency of Key Establishment* is defined as the time elapsed between the initial deployment and the end of key establishment processes of all nodes.

2. *Success Percentage for Private Key Generation* is the ratio of the number of mesh nodes that can compute their user private keys over the total number of the mesh nodes present within the network.

5.2 Simulation Results

As mentioned in Section 3.5, private key generation service is carried out succesfully if and only if the requesting mesh node receives sufficient number of shares from its neighboring nodes that finished computing their master private key shares. Thus, the success percentage not only depends on the number of shares received but also to the number of neighboring nodes that actually have their master private key shares. Figure 2 shows the change in success percentage of private key generation with respect to the threshold value for different network sizes. When the threshold value is 4, all the nodes compute their user private keys while when it is 6, at least 90% of them can perform that computation. As we increase the threshold value, the success ratio decreases; meaning that some nodes cannot compute their user private keys due to insufficient amount of received shares. On the other hand, as we increase network size, for the same threshold value, we achieve a higher success ratio. This is because of the fact that more shares become accessible by the requesting node as the number of neighboring nodes increase.

In Figure 3, change in latency is examined with respect to the threshold value for different network sizes. Since the most of the burden is in the first phase, i.e. mesh routers generate the master private key shares, latency of key establishment has not been affected by the total number of mesh nodes. Also, the threshold values smaller than 6 do not affect the latency. This is due to the fact that almost all the mesh nodes can compute their private keys for those threshold values. However, as we increase threshold value, some of the nodes cannot compute their private keys and they use the timeout method described in Section 3.5. This, consequently, causes an increase in latency.

6. CONCLUSIONS

WMNs are an emerging research area representing a good solution for providing low-cost and high-speed network services for the end users. In this paper, we propose a secure and efficient key establishment protocol for WMNs. Our scheme is based on IBC by which the communication within the network is secured. Moreover, we make use of a variant of Shamir's $(n,\, k)$-ThSS scheme along with AdSS, where trusted authority is abrogated with the collaborative gener-

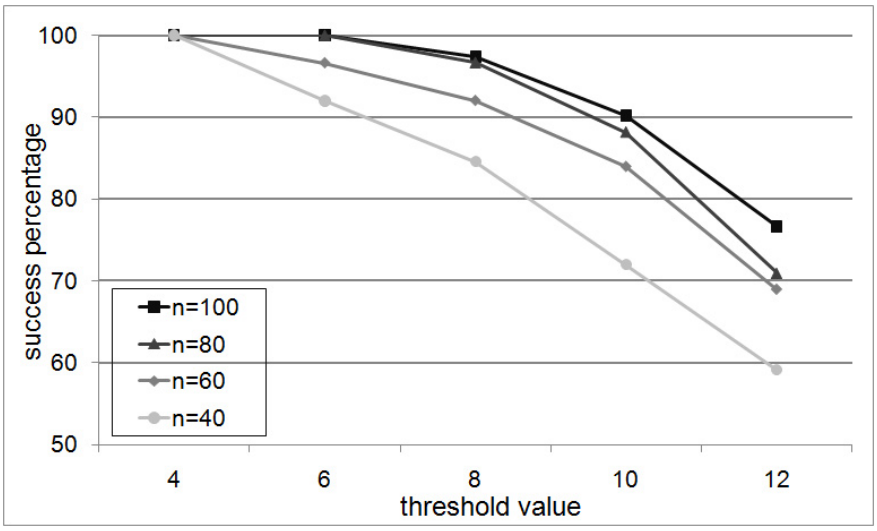

Figure 2: Success percentage of the proposed DKE scheme with respect to the threshold value for different numbers of nodes

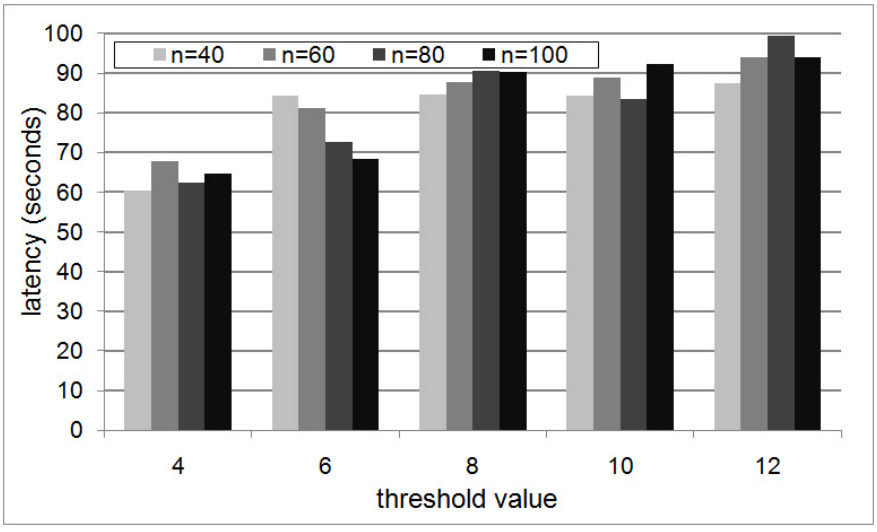

Figure 3: Latency of the proposed DKE scheme with respect to the threshold value for different numbers of nodes

ation of the secrets, yielding the improved resiliency against attacks.

Our simulations show that 100% of the mesh nodes can compute their private keys within at most 60 seconds, regardless of the number of mesh nodes, for the threshold value 4. For the threshold values $4 < k \leq 8$ and for the number of mesh nodes $40 \leq n \leq 100$, at least 90% of the mesh nodes can compute their private keys within at most 70 seconds. For the worst case, i.e. a network with 40 nodes performing at threshold level 12, at least 58% of the mesh nodes can compute their private keys within 90 seconds on the average. On the other hand, at threshold value 8, an increase in the number of mesh nodes from 40 to 100 results in a 25% increase in the rate of successful private key generations. These results clearly show the tradeoff between the resiliency of the network and the efficiency of our scheme: an increase in either the threshold value or the number of mesh nodes results in an increase in resiliency while decreasing the efficieny.

7. REFERENCES

[1] I. F. Akyildiz, X. Wang, and W. Wang. Wireless mesh networks: a survey. *Computer Networks*, 47(4):445–487, March 2005.

[2] N. Ben Salem and J.-P. Hubaux. A Fair Scheduling for Wireless Mesh Networks. In *The First IEEE Workshop on Wireless Mesh Networks (WiMesh)*, 2005.

[3] D. Boneh and M. Franklin. Identity-based encryption from the weil pairing. *Advances in Cryptology - CRYPTO 2001*, pages 213–229, 2001.

[4] C. Cocks. An identity based encryption scheme based on quadratic residues. *Cryptography and Coding*, pages 360–363, 2001.

[5] H. Deng, A. Mukherjee, and D. P. Agrawal. Threshold and identity-based key management and authentication for wireless ad hoc networks. In *Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004. International Conference on*, volume 1, pages 107–111 Vol.1, 2004.

[6] P. Gupta and P. R. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, 46(2):388–404, March 2000.

[7] D. P. A. Hongmei Deng. Tids: threshold and identity-based security scheme for wireless ad hoc networks. In *Ad Hoc Networks*, volume 2, pages 291–307, 2004.

[8] K. Jiejun, Z. Petros, H. Luo, S. Lu, and L. Zhang. Providing robust and ubiquitous security support for mobile ad-hoc networks. In *Network Protocols, 2001. Ninth International Conference on*, pages 251–260, 2001.

[9] J. Jun and M. L. Sichitiu. The nominal capacity of wireless mesh networks. *IEEE Wireless Communications*, 10(5):8–14, 2003.

[10] A. Khalili, J. Katz, and W. A. Arbaugh. Toward secure key distribution in truly ad-hoc networks. In *SAINT-W '03: Proceedings of the 2003 Symposium on Applications and the Internet Workshops (SAINT'03 Workshops)*, page 342, Washington, DC, USA, 2003. IEEE Computer Society.

[11] W.-B. Lee and K.-C. Liao. Constructing identity-based cryptosystems for discrete logarithm based cryptosystems. *J. Netw. Comput. Appl.*, 27(4):191–199, 2004.

[12] G. Li. An identity-based security architecture for wireless mesh networks. In *NPC '07: Proceedings of the 2007 IFIP International Conference on Network and Parallel Computing Workshops*, pages 223–226, Washington, DC, USA, 2007. IEEE Computer Society.

[13] IEEE Draft Standard for Information Technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment : Wireless Access in Vehicular Environments. `http://ieeexplore.ieee.org/servlet/opac?punumber=5174142`.

[14] The Network Simulator NS-2. `http://www.isi.edu/nsnam/ns/`.

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

[16] A. Shamir. Identity-based cryptosystems and signature schemes. In *Proceedings of CRYPTO 84 on Advances in cryptology*, pages 47–53, New York, NY, USA, 1985. Springer-Verlag New York, Inc.

[17] M. S. Siddiqui and C. S. Hong. Security issues in wireless mesh networks. In *MUE '07: Proceedings of the 2007 International Conference on Multimedia and Ubiquitous Engineering*, pages 717–722, Washington, DC, USA, 2007. IEEE Computer Society.

[18] W. Stallings. *Cryptography and Network Security: Principles and Practice (5th Edition)*. Prentice Hall, 5th edition, 2010.

[19] S. Tsuji and T.Itoh. An id-based cryptosystem based on the discrete logarithm problem. *IEEE Journal of Selected Areas in Communications*, 7(4):467–473, 1989.

[20] Y. Zhang and Y. Fang. Arsa: An attack-resilient security architecture for multihop wireless mesh networks. *IEEE Journal on Selected Areas in Communications*, 24(10):1916–1928, 2006.

[21] Y. Zhang and Y. Fang. A secure authentication and billing architecture for wireless mesh networks. *Wirel. Netw.*, 13(5):663–678, 2007.

[22] L. Zhou and Z. J. Haas. Securing ad hoc networks. *IEEE Network Magazine*, 13:24–30, 1999.