A Key Establishment Scheme for Wireless Mesh Networks using Identity-based Cryptography and Threshold Secret Sharing

by

DUYGU KARAOĞLAN

Submitted to the Graduate School of Sabancı University
in partial fulfillment of the requirements for the degree of
Master of Science

Sabancı University

Spring, 2009

A Key Establishment Scheme for Wireless Mesh Networks
using Identity-based Cryptography and Threshold Secret Sharing

Approved by:

Assoc. Prof. Dr. Albert Levi .....
(Thesis Supervisor)

Assoc. Prof. Dr. Erkay Savaş .....
(Thesis Supervisor)

Assist. Prof. Dr. Selim Balcısoy .....

Assoc. Prof. Dr. Özgür Gürbüz .....

Assist. Prof. Dr. Kemalettin Erbatur .....

Date of Approval: .....

A KEY ESTABLISHMENT SCHEME FOR WIRELESS MESH NETWORKS USING IDENTITY-BASED CRYPTOGRAPHY AND THRESHOLD SECRET SHARING

Duygu KARAOĞLAN

Computer Science and Engineering, Master's Thesis, 2009

Thesis Supervisors: Assoc. Prof. Dr. Albert Levi, Assoc. Prof. Dr. Erkay Savaş

Keywords: Wireless Mesh Networks, Key Establishment, Identity-based Cryptography, Threshold Secret Sharing, Additive Secret Sharing

Abstract

Wireless Mesh Networks (WMNs) are an emerging research area that provide low-cost and high-speed network services for the end users. Key establishment, on the other hand, is the most important and critical security concern for WMNs as all the other types of wireless networks. However, the conventional solutions for key establishment do not fit in the unique constraints and requirements of WMNs.

In this thesis, we propose two efficient and secure key establishment protocols elaborated at the sake of WMNs. Our security model is based on Identity-based Cryptography (IBC) and Threshold Secret

Sharing (ThSS). By the utilization of IBC, we eliminate the necessity of certificates used in infrastructure based schemes along with meeting the security requirements. With the utilization of ThSS, we provide a more resilient network working in a self-organizing way to provide the key establishment service, without the assumption of a trusted authority.

In the schemes we propose, master private key of the network is distributed among the mesh nodes. The user private key generation service is handled with collaboration of k mesh nodes, where k is the threshold value. A high threshold value increases the resiliency of the network against attacks; however, this negatively affects the system performance. We performed simulative performance evaluation in order to show the effect of both the number of mesh nodes in the network and the threshold value k on the performance. For the threshold values smaller than 8, at least 90% of the mesh nodes compute their private keys within at most 70 seconds. When we increase the number of mesh nodes in the network from 40 to 100, the rate of successful private key generations increase from 75% to 100% at the threshold value 8 where the latency of the key establishment is around 80 seconds. Considering the same increase in the number of mesh nodes, network performs up to 42% better at worst case, for the threshold values larger than 8, and the latency becomes at most 90 seconds on the average.

TELSİZ IZGARA AĞLAR İÇİN KİMLİK TABANLI KRİPTOGRAFİ VE EŞİK SIR PAYLAŞIMI KULLANAN BİR ANAHTAR TESİS MEKANİZMASI

Duygu KARAOĞLAN

Bilgisayar Bilimi ve Mühendisliği, Yüksek Lisans Tezi, 2009

Tez Danışmanları: Doç. Dr. Albert Levi, Doç. Dr. Erkay Savaş

Anahtar Kelimeler: Telsiz Izgara Ağlar, Anahtar Tesis Etme Mekanizması, Kimlik Tabanlı Kriptografi, Eşik Sır Paylaşımı, Katkılı Sır Paylaşımı

Özet

Telsiz Izgara Ağlar gelişmekte olan bir araştırma alanıdır ve kullanıcılara hem ucuz hem de hızlı servis sağlamaktadırlar. Öte yandan, anahtar tesis etme mekanizması, her türlü ağda olduğu gibi Telsiz Izgara Ağlar için de çok önemli ve kritik bir güvenlik kaygısıdır. Ancak, anahtar tesis etmek için kullanılan geleneksel yöntemler Telsiz Izgara Ağlar'ın benzersiz özelliklerine ve kısıtlamalarına uymamaktadır.

Bu tez ile, Telsiz Izgara Ağlar'a özel tasarlanmış iki verimli ve güvenli anahtar tesis etme mekanizması sunuyoruz. Güvenlik modelimiz Kimlik Tabanlı Kriptografi ve Eşik Sır Paylaşımına dayalı. Kimlik Tabanlı Kriptografi

kullanımı güvenlik gerekliliklerini sağlamakla birlikte geleneksel sistemlerin gerektirdiği sertifikaları da ortadan kaldırmaktadır. Diğer yandan, Eşik Sır Paylaşımı ağın daha esnek olmasına olanak vermekle birlikte kendi kendine düzenlenen bir anahtar tesis etme mekanizmasının oluşturulabilmesini sağlamaktadır.

Sunduğumuz iki mekanizmada da ağın ana şifresi kullanıcılar tarafından paylaşılmaktadır ve kullanıcıların şifrelerinin hesaplanması ancak yeterli sayıda kullanıcının - eşik değerini sağlayacak şekilde - biraraya gelmesi ile gerçekleşmektedir. Eşik değerini arttırdığımızda ağın saldırılara karşı olan esnekliği de artar ama bu durum sistemin performansını kötüleştirmektedir. Toplam kullanıcı sayısının ve eşik değerinin performans üzerindeki etkilerini görebilmek için bir takım similasyonlar yaptık: 8'den küçük eşik değerleri için kullanıcıların en az %90'ı kendi şifrelerini en fazla 70 saniyede oluşturabilmektedir. Eşik değerini 8'e sabitleyerek, kullanıcı sayısını 40'dan 100'e yükseltirsek, kullanıcı şifrelerinin oluşturulabilme yüzdesi de %75'den %100'e yükselmektedir ve işlemler 80 saniyede tamamlanmaktadır. Eşik değerini 8'in üstüne çıkardığımızda ise, kullanıcı sayısındaki aynı artış en kötü durumda bile ağın %42 daha verimli olduğunu göstermektedir ve işlemler bu koşullarda en fazla 90 saniyede son bulmaktadır.

to my beloved family

Acknowledgements

I wish to express my sincere gratitute to Assoc. Prof. Albert Levi and Assoc. Prof. Erkay Savaş, for their continuous support and wortwhile guidance thoughout my masters studies. Assoc. Prof. Albert Levi was always accessible and willing to help; I feel proud as he placed confidence in me more than I do. Also I am thankful to my thesis defense committee members: Assoc. Prof. Ozgur Gurbuz, Assist. Prof. Selim Balcısoy and Assist. Prof. Kemalettin Erbatur for their support and presence.

I appreciate Ayşegül Karatop, Ismail Fatih Yıldırım and Can Berk Güder for their help during the implementation process. I would also like to thank to my friends Emre Kaplan and Murat Ergun for their help in the cirruculum courses. Özlem Kocabaş and Çetin Akdere deserve special thanks for their precious support.

Last, but not the least, I am immensely thankful to my family, for being there when I needed them to be.

Contents

1 Introduction **1**

2 Background Information and Related Work **3**

2.1 Wireless Mesh Networks (WMNs) 3

2.1.1 Characteristics of WMNs 5

2.1.2 Security Requirements 7

2.2 Key Establishment 8

2.2.1 Symmetric Key Establishment 9

2.2.2 Asymmetric Key Establishment 9

2.3 Cryptographic Overview 10

2.3.1 Identity-based Cryptography (IBC) 11

2.3.2 Secret Sharing 15

2.4 Related Work 21

3 Motivation and Contribution of the Thesis **25**

3.1 Motivation 25

3.2 Contribution of the Thesis 26

4 Proposed Distributed Key Establishment (DKE) **29**

4.1 Assumptions 29
4.2 General Methodology 30
4.2.1 Master Private Key Share Generation 32
4.2.2 Master Private Key Distribution 35
4.2.3 User Private Key Generation 37
4.2.4 Timeout Method 38
4.3 Specialized Methodologies 40
4.3.1 DKE with use of ThSS 40
4.3.2 DKE with use of both ThSS and AdSS 43

5 Security and Resiliency Analysis 45

5.1 Security Analysis 45
5.2 Resiliency Analysis 45
5.2.1 Resiliency Analysis of DKE with ThSS 46
5.2.2 Resiliency Analysis of DKE with ThSS and AdSS . . . 47

6 Communication and Computational Overheads 48

6.1 Communication Overhead 48
6.2 Computational Overhead 49

7 Performance Evalution 51

7.1 Simulation Setup 51
7.2 State of the Network 52
7.2.1 Channel, MAC and Network Interface Types 52
7.2.2 Antenna and Radio Propogation Models 53
7.2.3 Queue Type 53
7.2.4 Routing Protocol 54
7.2.5 Transport Layer Communication Protocol 55
7.3 Implementation Details 55
7.3.1 Cryptographic Operation Latencies 55
7.3.2 Performance Metrics 57
7.4 Results 58

8 Conclusions and Future Work **65**

List of Figures

1 Infrastructure of a WMN 4

2 A Wireless Mesh Network (WMN) 4

3 IBC Framework 14

4 An Example for Shared Secret Construction 20

5 Success Percentage of DKE with ThSS 59

6 Success Percentage of DKE with ThSS and AdSS 60

7 Latency of DKE with ThSS 61

8 Latency of DKE with ThSS and AdSS 62

9 Success Percentage for an Ad hoc Network 63

10 Latency for an Ad hoc Network 64

List of Tables

1 The Symbols used in Protocol Definition 31

2 Computational Overheads for DKE with ThSS 50

3 Static Latency Benchmark 56

4 Dynamic Latency Benchmark for the Second Protocol 56

1 Introduction

Wireless Mesh Metworks (WMNs) are wireless networks in which nodes are able to carry out mesh routing by the utilization of 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, high robustness and increased coverage with an effective level of scalability. To have those advantages, the utilization of WMNs became a convincing choice and is preferred in the areas that do not have wired infrastructure or in the territories on which a temporary wireless network will be deployed.

Nevermore, multi hop cummunication and the nature of wireless channel make the WMNs prone to both passive and active attacks. Thus, the communication security between the mesh nodes is the most important problem to take a strong interest in. In order to maintain mutual trust and secure communication among the mesh nodes, a key establishment service must be provided. The limitations of conventional solutions necessitate the development of a brand-new security architecture to cope with the unique requirements of WMNs [1].

In this thesis, we propose two efficient and secure key establishment protocols which are designed with respect to the requirements and constraints of WMNs. The utilization of Identity-based Cryptography (IBC) along with Threshold Secret Sharing (ThSS) is preferred to overcome the problems at

present, namely network bandwidth consumption, network resiliency and single point of failure. In addition to those, we also achieved all of the security requirements of WMNs with the use of IBC.

The rest of the thesis is organized as follows: Section 2 contains background information on WMNs, key establishment and the cryptographic algorithms that form the basis of a secure key establishment scheme together with the related work. In Section 3, motivation and contributions of the thesis are presented. Then we describe our proposed solutions in detail in Section 4. In Section 5, to what extend the security requirements are met and in Section 6 the computational and communicational complexities are examined. Section 7 consists of the evaluation of our proposed solutions. Finally, we conclude the thesis in Section 8.

2 Background Information and Related Work

In this section, we explicate the characterictics of Wireless Mesh Networks (WMNs) along with their security requirements. Then, we define the cryptographic protocols we utilized and we give an introductory information on key establishment. Finally, we conclude the section with the related work done in the field of WMN security.

2.1 Wireless Mesh Networks (WMNs)

Wireless Mesh Networks (WMNs) are enclosed with mesh routers and mesh clients, where mesh routers are stationary while mesh clients can either be stationary or mobile. The backbone of a WMN consists of mesh routers and the whole WMN is formed by the appendage of mesh clients. Along with integrating stationary and mobile nodes, a WMN can optionally provide Internet access [11].

A typical WMN, having an infrastucture as in Figure 1 is shown in Figure 2.

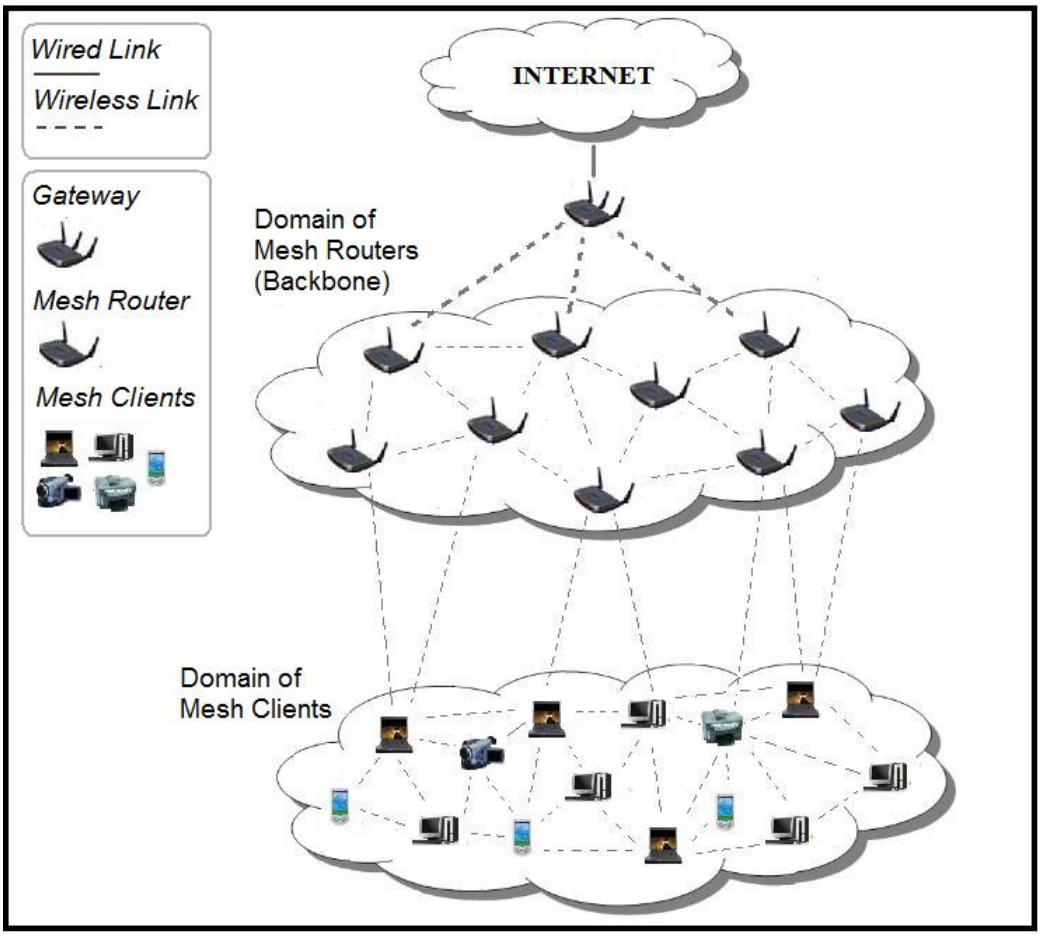

Figure 1: Infrastructure of a WMN

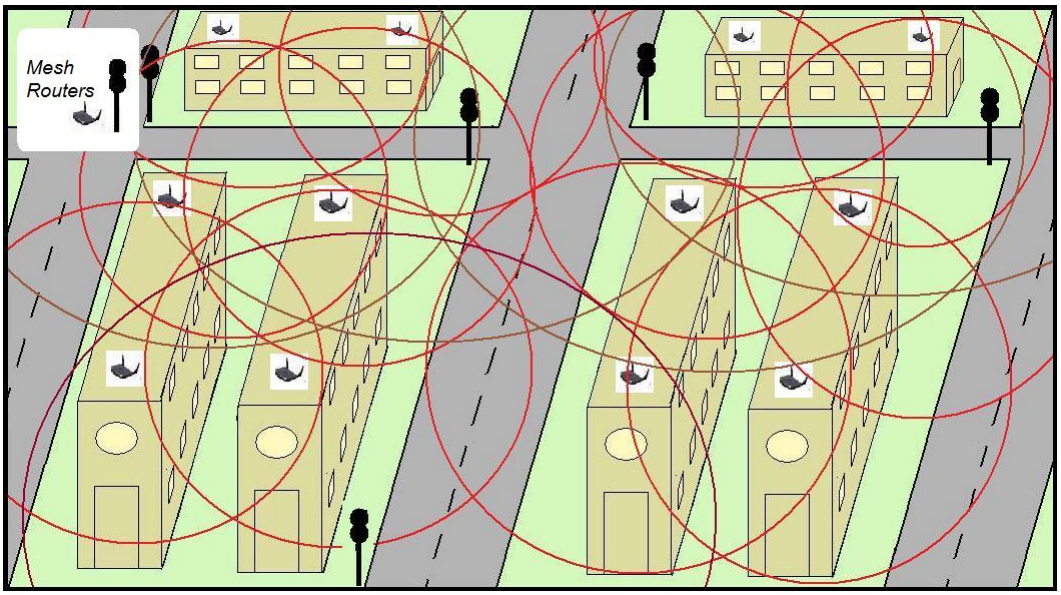

Figure 2: A Wireless Mesh Network (WMN)

2.1.1 Characteristics of WMNs

Multi hop wireless network Power of the signal is maintained by splitting the long distances into shorter hops. Each mesh node acts as a repeater that forwards data on behalf of the source node until the data reaches the destination. Thus, WMNs achieve a network with higher bandwidth in comparison to other wireless networks of whose coverage areas are the same [1].

Infrastructure and mobility The infrastructure can be defined as a wireless cooperative communication carried out in between a number of mesh nodes [24]. At any time, any node can either join or leave the network and that does not affect any network functionality. On the contrary, joining nodes enlarge the network coverage and provide a larger connectivity since they also act as forwarders. Besides, if a mesh node crushes or decides to leave the network, a neighbor of it can be in the routing path instead of itself. This characteristic increases the availability of the network. Additionally, with the fact that the mesh routers are stationary, continuous connectivity throughout the network is achieved without compromising the performance.

Dedicated configuration WMNs consist of mesh routers and mesh clients, as mentioned above. The difference between these two types of mesh nodes underlies not only in their mobility but also in the energy consumption constraints they have. Mesh clients are assumed to have a larger amount of energy consumption limitation. Therefore, the load of functionalities that require a higher computational power and bandwidth can burden on the

mesh routers.

Integration WMNs enable integration of various existing networks through the gateway functionalities of the mesh routers [1]. This provides that an end user within a network can utilize a service of another network through a WMN.

All the abovementioned characteristic brings out a different advantageous aspect of WMNs. However, WMNs also have disadvantages, as one should expect. Although the utilization of multi hop communication yields advantegous characteristics, it is also one of the derogations of WMNs. Due to the nature of wireless channel, all wireless networks are prone to passive attacks. However, the communication carried out in a multi hop fashion results in the possibility of active attacks [28]. In a WMN, a passive attack will result in the violation of confidentiality whereas an active attack will compromise resiliency, integrity, authentication and non-repudiation [25]. Therefore, maintaining the communication security between the mesh nodes is the most important problem to take a strong interest in. In addition to those, mesh nodes have both limited power and limited storage area, because of which thet cannot perform large computations.

2.1.2 Security Requirements

Authentication ensures that the communicating entity has the identity that it claims to have; meaning that the origin is correctly identified. In a group of wireless nodes, this is achieved by either using pairwise keys, using a group key or with the use of Public Key Infrastructure (PKI)-based schemes. Unless authenticity is accomplished, an adversary can masquerade a node and gain unauthorized network access.

Confidentiality ensures that only the ones who are authorized to have access to specific data can access that data. In other words, confidentiality hides the contents of the information exchanged, thus protects the data from unauthorized disclosure. This is achieved by encrypting the data and giving access to the authorized party for decryption. Obviously, first the authentication must be achieved.

Integrity is the assurance of the fact that only the authorized parties can modify the data. By this, the validity of the data exchanged is satisfied. As a resuly, when integrity is achieved, any party can understand whether the information sent is modified, replayed, deleted or not. This is generally ensured by the use of a number referred as a Message Integrity/Authentication Code, which is computed with both the data and a shared secret and is appended to the end of the data. When the receiving party gets the information, it computes the extension part using the secret and checks whether it is equal to or not to the received extension part. Alternatively, it is also achieved

by using session keys, which are the symmetric keys that the communicating parties hold. The exchanged data in a session between the communicating parties is encrypted and decrypted with this session key.

Non-repudiation requires that neither of the authorized parties deny the information being exchanged. In other words, it is the protection against denial by either of the communicating parties. This security requirement is actually useful in the detection of compromised nodes; it allows a user receiving an errornous message to decide whether the sending node is compromised or not. Non-repudiation is ensured by using a signature scheme in which the data to be sent is encrypted with the sender's private key.

2.2 Key Establishment

In order to establish and maintain mutual trust and secure communication among the mesh nodes, a key establishment service must be provided. This leads to the significance of how the keys are managed to be exchanged or distributed. There are basically two approaches: symmetric key establishment and asymmetric key establishment.

In the decision of the key establishment protocol that will be utilized, characteristics and constraints of a network plays an important role. Because of the fact that symmetric key algorithms have a lower computational complexity than that of asymmetric ones, the commonly preferred way of ensuring a secure communication passes over using an asymmetric key es-

tablishment protocol to agree on a symmetric session key.

2.2.1 Symmetric Key Establishment

Symmetric key establishment involves the distribution of the symmetric keys, which are used in both encryption and decryption within a communication session. This type of key establishment is provided in two different ways. In the first way, a trusted authority, which generates and distributes the keys, is assumed. This is impractical due to the hardness of keeping a server available everytime it is needed to be used. In the second way, the burden of the key generation is given to one of the communicating parties. In other words, one of the parties generates the secret key to be used and sends it securely to the other party. However, in both types of the symmetric key establishment, there is the risk of being prone to single point of failure.

2.2.2 Asymmetric Key Establishment

Public Key Cryptography (PKC) is first proposed by Diffie and Hellman in 1976 [7] and is considered to the be the most important breakthrough in the history of cryptography [26].

In PKC, each user has a pair of public and private keys. The private key of the user is kept secret while the public key is widely distributed. Basically the public and the private keys are related to each other; however it is not mathematically feasible to derive the private key from the public key. And

most importantly, the key that is used to encyrpt a message is different from the key by which the corresponding message is decrypted.

Public Key Infrastructure (PKI) is the most important characteristic of the traditional PKC. It ensures an infrastructure that keeps track of the public keys with both the use of certification, by which the public keys are bind to the users, and validation, by which the certificates are guaranteed to be applicable. Certificates consist of the user information along with the public keys of that user most commonly signed by a certification authority (CA). Since the CAs are trusted authorities and either known or reachable by every user; its public key is used by the users in the validation process.

Beside the PKI-based schemes, Identity-based Cryptography (IBC), which is explained in Section 2.3.1, is another type of PKC which is utilized in the asymmetric key establishment. Essentially, 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.

2.3 Cryptographic Overview

Any adversary can monitor a mesh node easily due to the utilization of wireless channel along with multi hop communication and mobility [31]. This brings out the fact that WMNs are prone to both passive and active attacks. In a WMN, a passive attack will result in violation of confidentiality whereas an active attack may compromise availability, integrity, authentication and non-repudiation [28]. Therefore, the difficulty of providing communication

security between the mesh nodes is one of the main drawbacks of WMNs.

The other important drawback is the constraints of WMNs, discussed in Section 2.1.1. For authentication and encryption, traditional Public Key Infrastructure (PKI) based schemes are hard to deploy for WMNs, since the capabilities of mesh nodes are limited in the sense of resource and power. Thus, the need for the utilization of symmetric key cryptography arises. However, to use that approach, there is the need for a good mechanism to distribute the keys.

In the following subsections, we introduce cryptographic methods that can be used for the maintanence of such schemes.

2.3.1 Identity-based Cryptography (IBC)

The concept of Identity-based Cryptography (IBC) is put forward by Adi Shamir [23] in 1985. The basic idea of IBC is to find an approach by which the public key of a user is defined as an arbitrary string that uniquely identifies him in such a way that the denial is impossible. It may be the IP address, e-mail address, name, etc., which eliminates the need for certificates along with the need for Certificate Authorities (CAs). As a consequence, users in IBC do not have to exchange public keys, certificates, etc [23]. In IBC, users may also choose random looking public keys to achieve anonymity.

In IBC, all the user private keys are generated by a trusted authority, namely the Private Key Generator (PKG). PKG holds a master key (public/private key pair), with which the user private key generation is performed.

To be clear, without the knowledge of the master private key, none of the user's private keys can be generated. After having its user private key, a node can encrypt/sign and decrypt/verify a message. After delivering the private key, the PKG does not involve in any other operation. Thus, the network does not need to be a centralized one and the solution is applicable for closed groups of users [23].

IBC consists of four phases:

1. Setup Phase (Algorithm 7 in Appendix): Global parameters and the master key of the system are generated by the PKG. The global parameters consist of q, G_1,G_2,H_1,H_2,$\hat{e}$ and P. First of all, G_1 and G_2 are two groups of order q, which is a sufficiently large prime. Secondly, H_1 and H_2 are cryptographic hash functions that map arbitrary strings to non-zero elements in G_1 and in the finite field $\mathcal{F}_q$ respectively. H_1 is used to map the identity of the user to a point on the curve, whilst H_2 is used to map the session key. Finally, $\hat{e}$ is the bilinear map such that $\hat{e} : G_1G_1 \longrightarrow G_2$ and P is the generator of G_1. Along with those, master key has two pairs: master private key, s_{priv}, and master public key, s_{pub}, which is defined as in Equation 2.1.

$$s_{pub} = s_{priv} \times P \tag{2.1}$$

2. Extract Phase (Algorithm 8 in Appendix): PKG uses the master key along with the public key of the requesting user to construct the user's private key. Assuming that the user's public key is ID_i, the private

key is computed as in Equation 2.2.

$$PK_i = s_{priv} Q_{ID_i} \tag{2.2}$$

where, Q_{ID_i} is defined as in Equation 2.3.

$$Q_{ID_i} = H_1(ID_i) \tag{2.3}$$

3. Encryption Phase: In the encryption phase, the message to be transmitted is encrypted with the sending user's private key. This operation is carried out on the side of the party who will send a message.

4. Decryption Phase: In the decryption phase, the received message is decrypted with the sending node's public key, which is computed from the identity of the sender as in Equation 2.3. As expected, the user that receives a message performs this operation.

The framework of an IBC is as seen in Figure 3. For instance, let Alice be the sender and Bob be the receiver. When Alice wants to send a message to Bob, she simply encrypts the message with Bob's public key, i.e. bob@su.sabanciuniv.edu. On the other hand, when Bob receives the message that Alice sent to him, he decrypts it with his user private key. At this point of time, if he does not yet has his user private key, he contacts with PKG and sends a request after authenticating himself.

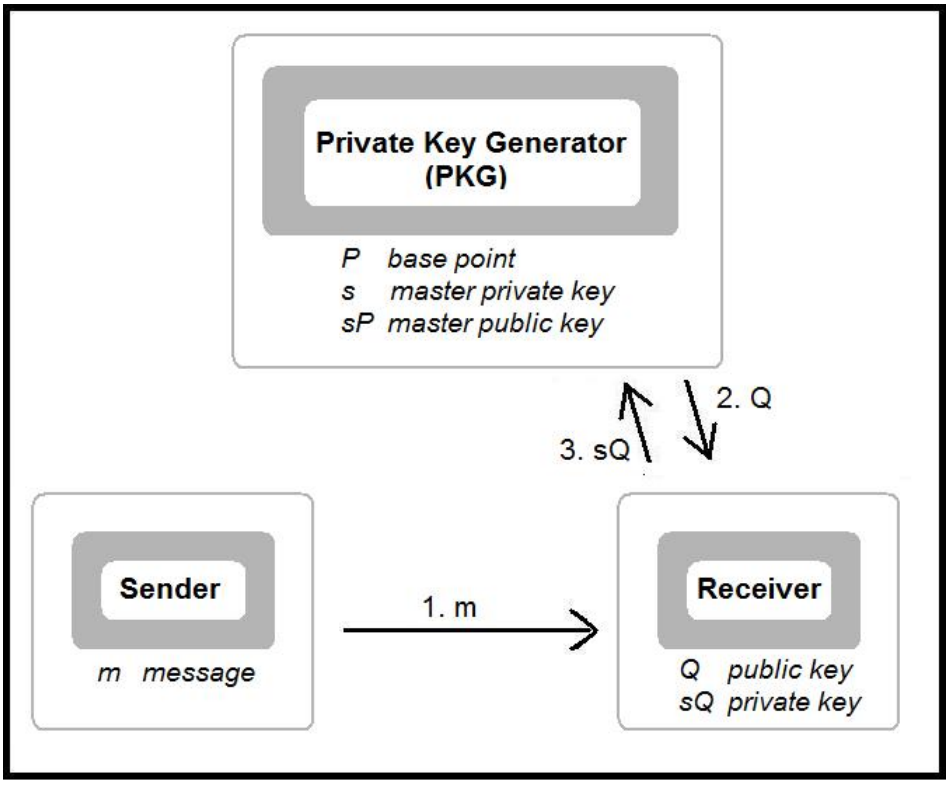

Figure 3: IBC Framework

The important point here is that the receiving party does not need to have its user private key to be able to receive a message. That is actually due to the fact that the sending party does not need for receiving party's certificate.

Quite a few schemes proposed in the field of IBC, which can be examined in detail from [5, 12, 14, 3]. [5] is based on quadratic residues while the others use pairing operation defined over Elliptic Curve Cryptography (ECC)[1]. The

[1]ECC is based on the the difficulty of elliptic curve discrete logarithm problem and has almost the same cryptographic security as 1024-bit key length used in RSA [27, 26].

most practical one is the one proposed by Boneh and Franklin [3], which uses Weil Pairing as the bilinear mapping on ECC, due to the fact that it has a performance comparable with ElGamal encryption and it has the chosen cipher text security in the random oracle model.

2.3.2 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[2] share alone. The secret cannot be reconstructed unless a certain condition is met, and that condition is generally a coalition among a sufficient number of shareholders.

All the secret sharing schemes are based on a field structure and have the characteristic that a secret s is shared among n participants. What differs them is the required number of collaborators needed for the reconstruction process. Henceforth in our constructions, we use $\mathcal{F}_q$ field, where q is prime for simplicity.

Additive Secret Sharing

In Additive Secret Sharing (AdSS) schemes, the secret s is distributed among n users in a way that adding up all the shares gives the secret. In other words, it is impossible to reconstruct the secret unless all the shareholders collude.

[2]No gender implication.

AdSS assumes the existence of a trusted third party (TTP), by whom the shares are generated and transmitted securely[3] to the corresponding shareholders. What TPP performs is as follows:

1. chooses a large prime q and a secret $s \in \mathcal{F}_q$.
2. chooses $n-1$ random numbers s_1, s_2, s_3, ..., s_{n-1} to be the shares of the secret.
3. computes the last share of the secret by Equation 2.4.

$$s_n = s - \sum_{k=1}^{n-1} s_k \pmod{q} \tag{2.4}$$

4. sends the shares s_i to the corresponding shareholders, u_i.

The reconstruction of the secret in AdSS is performed with the collaboration of all the shareholders evaluating Equation 2.5.

$$s = \sum_{i=1}^{n} s_i \pmod{q} \tag{2.5}$$

Threshold Secret Sharing

In Threshold Secret Sharing (ThSS) schemes, the secret s is distributed among n users in such a way that any subset of k users can reconstruct the secret s, but no subset of smaller size can. These schemes are also known as (n, k)-ThSS schemes.

[3]The trusted authority is assumed to be powerful enough to establish a secure communication link

Shamir's ThSS

One of the widely used ThSS schemes is proposed by Adi Shamir [22] in 1979. The basis of his scheme is 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 [22].

The Lagrange Interpolation Polynomial is a linear interpolation polynomial in which the data points are in the Lagrange form. Given a set of k data points in the 2-dimensional plane (x_i, y_i), the Lagrange polynomial is defined as the linear combination given in Equation 2.6 of the Lagrange coefficients defined by Equation 2.7.

$$L(x) = \sum_{j=1}^{k} y_j l_j(x) \tag{2.6}$$

$$l_j(x) = \prod_{i=1, i\neq j}^{k} \frac{x - x_i}{x_i - x_j} \tag{2.7}$$

The existence of TTP is also assumed in Shamir's ThSS scheme, whose role is to generate and to distribute the shares. TTP performs these operations as in Algorithm 11 given in Apendix and the operations are as follows:

1. chooses a large prime q, a secret $s \in \mathcal{F}_q$ and a polynomial $f(z)$ of degree $k-1$, such that $f(0) = s$.

2. evaluates the polynomial for each user to generate their shares via

Equation 2.8.

$$s_i = f(i) \pmod q \tag{2.8}$$

3. sends the shares s_i to the corresponding shareholders, u_i.

As for the reconstruction of the secret, k of the shareholders combine their shares as it is given in Algorithm 12 in Appendix, performing Equation 2.9.

$$f(a) = \sum_{j=1}^{k} s_j l_j(a) \pmod q \tag{2.9}$$

$$l_j(a) = \prod_{i=1,\, i\neq j}^{k} \frac{a-i}{i-j} \pmod q \tag{2.10}$$

Shamir's ThSS Without a TTP

The problem of Shamir's ThSS stems from the assumption of the 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 performs the following operations:

1. selects a secret x_i and a polynomial $f_i(z)$ of degree $k-1$, such that $f_i(0) = x_i$.[4]

2. generates the shares $x_{i,j}$[5] of x_i, where $j = 0, 1, 2, \ldots, n$, as described

[4]Modulus is assumed to be known by all the nodes.
[5]The subscript i, j is defined as *by i for j*.

in Section 2.3.2.

3. sends $x_{i,j}$ to N_j, where $j = 0, 1, 2, \ldots, n$ and $j \neq i$.

When node N_i receives $n-1$ of $x_{j,i}$'s, where $j = 1, 2, 3, \ldots, n$ and $j \neq i$, it can compute its shared secret (Algorithm 1) via Equation 2.11.

$$s_i = \sum_{j=1}^{k} x_{j,i} \pmod{q} \tag{2.11}$$

Algorithm 1 COMPUTE-SHARE-OF-THE-SECRET $(x_{i,j})$

```
(1) sharedData ← x_{j,j}
(2) i ← 0
(3) while i < n do
(4)     if i ≠ j then
(5)         sharedData ← (sharedData + x_{i,j}) mod q
(6)     end if
(7)     i ← i + 1
(8) end while
(9) return sharedData
```

Figure 4 below, shows an instance of a share construction performed by three users. Alice, Bob and Charlie first selects a secret and then evaluates it on the polynomial he/she has selected. The resulting three shares of the chosen secret correspond to the subshares of the actual secret to be shared. As either of them receives two subshares, he/she can compute his/her share of the actual secret.

This computed share befits to the share distributed by the TTP in an (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.

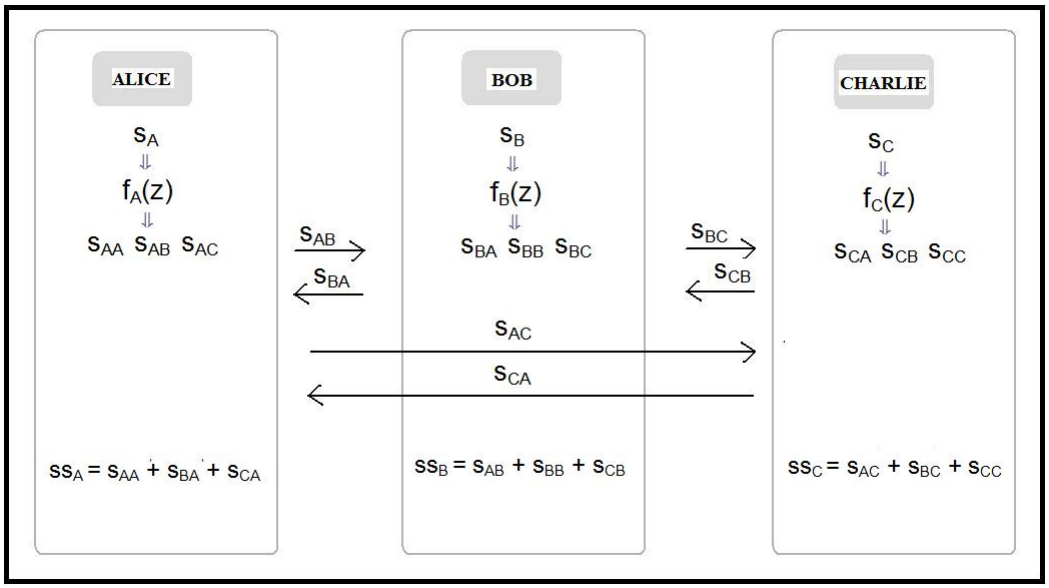

Figure 4: An Example for Shared Secret Construction

Variations on ThSS

The abovementioned ThSS schemes consider splitting the secret s, in between 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, we give y shares to the others, with $x > y$. Thus, the scheme becomes an ($ax + by$, k)-ThSS in which 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. To be more precise, let us assume that we have $n = n_1 + n_2$ users for the share to be distributed among. 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.

2.4 Related Work

Salem and Hubaux [2] describe specifics of WMNs and identify three fundemental network security requirements: detection of compromised mesh routers, utilization of secure routing and fairness. In [28] Wu and Li propose Onion Routing, a private routing algorithm, which utilizes layered encryption in the achievement of end user privacy. Using this scheme, a group of users can connect to the Internet through the Onion routers without revealing the routing information. In [16], Siddique et al. proposes a secure multi hop routing protocol for WMNs. Their network model consists of several mesh networks and they propose a routing algorithm with four components of which the main characteristic is that they utilize both proactive and reactive routing protocols.

Besides the routing related solutions mentioned above, secure authen-

tication protocols are also proposed for WMNs. For example, in [29, 30], Zhang and Fang propose UPASS/ARSA, a secure authentication and billing architecture to enable an omnipresent network with faultness roaming. In UPASS, the network is divided into domains each having a key of its own and a number of trusted authorities, as CAs, are assumed. When a mesh client wants access to the network, it first connects to the trusted authority to get its private key and then connects to the mesh router of the domain in which it stands. Thus, trust model of UPASS is built upon both PKI and IBC, which is not practical due to the fact that the users need to perform both CBC and IBC operations. Additinally, the scheme does not provide an efficient mechanism for key revocation. On the contrary, ISA proposed by Li [13] defines a good key revocation method. The necessity of the key revocation is determined by a neighbor detection mechanism in which if a certain number of nodes accuses a specific node, that node is treated as compromised. Moreover, ISA provides an efficient network access based on IBC with the assumption of the gateway router being the trusted authority. All the operations, i.e. key generation, key revocation and key renewal, are performed on the gateway router. When a new mesh client wants access to the network, it first connects to the gateway router to get its private key and then it implements a 3-way handshake protocol with the mesh router to compute a shared key. In spite of providing a leightweight network access and a good mechanism for the marking of compromised nodes, the assumption of a trust authority diminishes its practicability.

All the abovementioned protocols assume a trusted authority for efficient and secure key management. However, 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 the assumption of a trusted authority, threshold secret sharing is used in [31] and [9]. Zhou and Hass [31] presents a key management protocol based on the traditional PKI scheme, in which a group of nodes share the role of the CA. The nodes that withhold a share of the certificate signing key are able to generate partially signed certificates. As in the (n, k)-threshold scheme, any k partially signed certificates can collaboratively construct a signed certificate which befits to a certificate that is signed by a CA of the traditional PKI-based schemes. A similar approach is proposed by Kong [9], in which the RSA certificate signing key is distributed among all the nodes of the network. The two schemes differ only in the name of the number of shareholders. When they are compared, the one proposed by Kong seems to have an advantage of providing a better availability since it is easier to get in contact with k neighbors in that scheme. However, in both protocols, the shares of the certificate signing key is generated and distributed by a trusted authority. Thus, they do not provide a fully distributed key management.

On the other hand, Deng et al. [6] proposed a secure key management scheme for ad hoc networks which is fully distributed; meaning that no trusted authority is assumed in either parts of the protocol. The combination of IBC and (n, k)-threshold Secret Sharing forms the basis of their solution; in which both the shares and the secret are generated collaboratively.

In this thesis, we propose two secure and efficient key establishment protocols by taking the work proposed by Deng et al. as basis. In other words,

we customize their solutions at the sake of the requirements and constraints of WMNs. In their scheme, due to the the idea of distributing the secret among all the nodes, the shares are also generated by the collaboration of all nodes. This makes their scheme inefficient with respect to the communication overhead introduced and network bandwidth used. We attenuate these disadvantages by the advantageous characteristics of WMNs.

3 Motivation and Contribution of the Thesis

This section includes information on why we selected this subject and what contributions we made.

3.1 Motivation

Like all the other types of networks, Wireless Mesh Networks (WMNs) also need a way of secure distribution of the private keys. In WMNs, the most suitable cryptographic approach for the secure key establishment is the utilization of Identity-based Cryptography (IBC). However, IBC assumes a trusted third party (TTP) which does not fit 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. What we need is to distribute the role of the TTP assumed in IBC.

As described in Section 2.3.1, in IBC, the role of the TTP is to generate and distribute the private keys of the users. To perform that computation, TTP holds a master secret key that belongs to the network. Therefore, in order to distribute the role of it, the master secret key of the network must be distributed.

The distribution of a secret can be done by the utilization of a Secret Sharing scheme. In Additive Secret Sharing (AdSS) discussed in Section 2.3.2, the number of nodes collaboratively reconstruct the secret must be

equal to the number of nodes in between which the secret is shared. Thus, within a network with large number of nodes, using only AdSS is unreasonable. What we left with is the Threshold Secret Sharing (ThSS), in which the secret is shared in such a way that a defined number of users withholding a total of k shares can collaboratively reconstruct the it, as described in Section 2.3.2. However, the most widely used ThSS, Shamir's ThSS, also assumes the existance of TTP. Therefore, the role of the TTP assumed in Shamir's ThSS should also be distributed and that can be done by using the extended version of the Shamir's ThSS scheme, which is described in Section 2.3.2.

3.2 Contribution of the Thesis

We examined the protocols proposed for the secure key establishment of different types of wireless networks and tried to apprehend the most suitable one for WMNs. Considering the constraints and the security requirements of WMNs, we agreed on a key establishment scheme that combines Identity-based Cryptography (IBC) and Threshold Secret Sharing (ThSS).

The proposed protocols using these techniques, discussed in Section 2.4, have two important disadvantages:

1. *Large transmission delays*: the number of users that collaboratively compute the master private key directly affects the amount of used network bandwidth. If we assume that n users are in such collaboration, then at least $n \times (n-1)$ packets will be transmitted in between the

nodes. This is due to the nature of the utilized secret sharing scheme, which is described in Section 2.3.2.

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

The characteristics of WMNs provide us a way to centralize the network to an extent. As discussed in Section 2.1.1, the mesh routers can be distinguished by the parameters they hold and/or by the operations they perform. Thus, we imposed the burden of the master key generation on them. This resulted in the reduction of the number of nodes present in the master key generation operation, which clearly eliminated the first abovementioned disadvantage. Additionally, we assumed that it is hard to compromise the mesh routers. 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, the resiliency of the system is increased without increasing the number of required neighboring nodes.

At this point, it is important to mention the importance of not increasing the neighboring node count. Since all the mesh nodes act as routers, the throughput of a mesh node is mostly dependent on the network topology and the number of neighbors of the node that are in its transmission range [25].

It is shown that a node having six neighbors has the optimal transmission power intensity in a stationary multi hop network [1].

In brief, we ameliorated the disadvantages mentioned above with the aid of the characteristics of WMNs.

4 Proposed Distributed Key Establishment (DKE)

This section provides a detailed explanation of our contributions. First, we define our assumptions. Then we give the general methodology of our scheme. Finally, we explain the specifications for two different proposed solutions.

4.1 Assumptions

Security solution does not rely on the existence of any trusted entity and there is no pre-defined mutual trust among the mesh nodes. However, mesh nodes will not collude to reveal any other mesh node's private key, especially the mesh routers.

By the characteristics of WMNs, we propose two secure and efficient key establishment schemes that does not rely on any trust authority to generate and distribute the private keys of the nodes. In other words, there is no underlying key establishment system. All the keys are generated collaboratively by the mesh routers and distributed accordingly to the mesh clients.

It is hard 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 the WMNs; we know that they are there, for sure. We turned this characteristic into an advantage by assuming that it is hard to compromise them. Additionally, we deployed the mesh routers in such a way that they cover the network area in

order to maintain continous connectivity. Obviously, mesh clients also have a role in the coverage area.

Identities of the mesh nodes are unique and each node have a mechanism to discover its one-hop neighbors.

As in all IBC systems, there is the assumption of the identity of the node being unique. In order to easily overcome this uniqueness issue, the identities of the nodes are selected to be their addresses, which simply can be obtained through dynamic address allocation. On the other hand, it can be said that 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, 10]. 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 [8]. Accordingly, we assume that each mesh node is able to discover its neighbors and find out their identities.

4.2 General Methodology

Our proposed 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 owns its master private key share, it can also contribute to this distribution process. Last phase provides a private key generation service, by which each mesh node[6] can compute their user private keys. This service is carried out by a collaboration of a defined number of mesh nodes.

Let us assume that we have a WMN of $n = m + l$ nodes, where m is the number of mesh routers and l is the number of mesh clients. In the following subsections, we give detailed information on how these phases are performed.

Table 1: The 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
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 private key	MK^{priv}
master private key share	MKS^{priv}
master public key share	MKS^{pub}
master private key partial share	$MKPS$
user public key	Q
user private key	PK

[6]Both mesh routers and mesh clients

4.2.1 Master Private Key Share Generation

The milestone of all the operations performed is the master private key MK^{priv}, which will be shared among all the mesh nodes. As mentioned above, generation of this key is carried out only by the mesh routers. Thus, the total number of shares present in the system depends on the number of shares that the mesh routers hold, namely x. This means that a total of $m \times x$ shares will be distributed among the nodes of the network.

Just after the deployment of the mesh routers, the very first thing they perform is the setup phase of the Identity-based Cryptography (IBC) system, which is described in Section 2.3.1. The parameters of IBC are set and the curve is constructed. Last two operations of IBC setup include the selection of the master private key and the computation of the corresponding master public key. As there is no trusted authority to construct and distribute the keys to the mesh nodes, these operations are not performed as it is defined in the original setup phase of IBC (Algorithm 7). Instead, the mesh routers collaboratively generate the shares of the master private key.

Each mesh router MR_i performs the following for the collaborative generation of the master private key shares:

1. computes subshares $ss_{i,j,a}$, where $j = 1, 2, \ldots, m$ and $a = 1, 2, \ldots, x$, as described in Section 2.3.2.

2. sends $ss_{i,j,a}$ to MR_j, where $j = 1, 2, \ldots, m$, $a = 1, 2, 3, \ldots, x$ and $j \neq i$.

The corresponding algorithm for the generation of the subshares can be found in Algorithm 2.

As a mesh router MR_i receives its first subshare, it starts a timer, whose reason is explained in Section 4.2.4. When MR_i receives $(m-1)\times x$ subshares, it cancels the timer and computes its master private key share via Equation 4.2. Additionally, withholding its master private key share, MR_i computes its master public key share via Equation 4.1 and publishes it. The operations performed upon a receipt of a subshare can be found in Algorithm 3.

$$MKS_i^{priv} = \sum_{a=1}^{x} MKS_{i,a}^{priv} \times l_i(0) \pmod q \tag{4.1}$$

where, $MKS_{i,a}^{priv}$ is defined as in Equation 4.2 and $l_i(0)$ is the Lagrange coefficient computed via the Equation 2.7.

$$MKS_{i,a}^{priv} = \sum_{j=1}^{m} ss_{j,i,a} \pmod q \tag{4.2}$$

.

$$MKS_i^{pub} = MKS_i^{priv} \times P \tag{4.3}$$

where, P is a common parameter used in IBC.

Algorithm 2 MASTER-PRIVATE-KEY-SUBSHARE-ESTABLISHMENT (s, m, x, k)

```
(1)  a ← 0
(2)  b ← 0
(3)  index ← 0
(4)  while a < x do
(5)     while b < m do
(6)        subshares_index ← GENERATE − SECRET−
             WITH − SHAMIR − ThSS(s, m, k)
(7)        b ← b + 1
(8)        index ← index + 1
(9)     end while
(10) a ← a + 1
(11) end while
(12) send the subshares to the corresponding nodes
```

In order for a mesh router to compute the actual value of the master public key, it needs to hold sufficient number of these types of shares. With that information, a mesh router reconstructs the master public key of the network as described in Sections 4.3.1 and 4.3.2 in correspondence with the definition of adequacy.

Algorithm 3 RECEIVE-MASTER-PRIVATE-KEY-SUBSHARE ($s_{senderAddr, myID}$)

```
(1)  if not received from senderAddr yet then
(2)     if isFirstSubshareReceived≠ true then
(3)        isFirstSubshareReceived ← true
(4)        subshareTimer.start() for some time interval
(5)     end if
(6)     subshares_senderAddr ← s_senderAddr, myID
(7)     subshareCount ← subshareCount + x
(8)     if subshareCount = m × x − x and
          masterPrivKeyShareSet = false then
(9)        if subshareTimer is on then
(10)          subshareTimer.cancel
(11)       end if
(12)       a ← 1
(13)       while a < m do
(14)          if a does not correspond to my identity then
(15)             MKS^priv ← MKS^priv + subshares_a
(16)          end if
(17)          a ← a + 1
(18)       end while
(19)       MKS^pub ← MKS^priv × P
(20)       broadcast MKS^pub
(21)    end if
(22) end if
```

4.2.2 Master Private Key Distribution

Second phase starts as a mesh client recognizes that one of its neighboring nodes finished computing its master private key share. This recognition is achieved with the message by which a mesh router publishes its master public key share. Upon receiving such a message, mesh client MC_i makes a request

from whose reply it will learn which of its neighboring nodes will help for the reconstruction of its master key shares[7].

As a sufficient number of its neighboring nodes reply, the requesting mesh client MC_i generates another request message which contains a list of the willing collaborators and broadcasts that message. Upon receiving the second request message, mesh node MN_j checks whether its identity is concatenated in the collaborators list or not. If it is, then MN_j computes the master private key partial share of MC_i via Equation 4.4 and sends it to MC_i.

$$MKPS_{j,i} = MKS_j^{priv} \times l_j(i) \pmod q \tag{4.4}$$

where, $l_j(i)$ is the Lagrange coefficient computed via the Equation 2.7.

On the other hand, if its identity does not appear in the collaborators list, MN_j simply discards the message.

When the requesting mesh client MC_i receives all the information it asks for, it computes its master private key share by simply adding up all the received partial shares as in Equation 4.5.

$$MKS_i^{priv} = \sum_{j=1}^{k} MKPS_{j,i} \pmod q \tag{4.5}$$

Additionally, MC_i reconstructs its master public key share as described in Sections 4.3.1 and 4.3.2.

[7]Both the master private and the master public key shares.

4.2.3 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.

In order to reconstruct its user private key, mesh node MN_i broadcasts a request message. Upon receiving user private key generation request, mesh node MN_j computes the user private key share for MN_i via Equation 4.6, if it has already computed its master private key share. In order to do the computation, MN_j first retrieves the public key of MN_i. However, if MN_j does not have its master private key share yet, it cashes the request to be able to send a reply after it finishes its master private key share computation (Algorithm 4).

$$PKS_{j,i} = MKS_i \times Q_j \tag{4.6}$$

where, Q_j is the public key of the requesting node.

As the requesting node MN_i receives sufficient number shares, it can reconstruct its user private key as will be described in Sections 4.3.1 and 4.3.2 in correspondence with the definition of adequacy.

Algorithm 4 SEND-PKG-REPLY ($destAddr$)

```
(1)  if MK^priv_myID is set then
(2)     if MKPS^priv_destAddr is not computed then
(3)        MKPS^priv_destAddr ←
                    EXTRACT − IBC(MKS^priv, destAddr)
(4)     end if
(5)     send MKPS^priv_destAddr to MN_destAddr
(6)  else
(7)     cash destAddr as requester
(8)  end if
```

4.2.4 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 the user private key. However, a situation as the following may also occur: 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 (Algorithm 5) is adopted. In this method, after sending a service request for either master key share computation or user private key generation, a mesh node sets a timer in correspondance with that request. If the mesh node makes this request on a data which will be received for sure, i.e. master private subshare exchanged in between the mesh routers, it keeps sending request packets periodically until the desired data is received. On the other hand, if there is a doubt on

the reception of the demanded data, i.e. user private key share, then the mesh node repeats its request periodically only a number of times.

Algorithm 5 TIMEOUT

```
(1) if type = subshareTimer then
(2)    if a MR then
(3)       if enough subshares has not received then
(4)          request subshare from which
                has not received yet
(5)       end if
(6)       subshareTimer.start(3)
(7)    else
(8)       timerCount ← timerCount + 1
(9)       if timerCount < 10 then
(10)         broadcast a request
(11)         subshareTimer.start(3)
(12)      end if
(13)   end if
(14) else if type = pkgTimer then
(15)    timerCount ← timerCount + 1
(16)    if timerCount < 10 then
(17)       request private key generation
(18)       pkgTimer.start(3)
(19)    end if
(20) else if type = partialShareTimer then
(21)    if enough shares has not received yet then
(22)       request partial share from which
             has not received yet
(23)    end if
(24)    partialShareTimer.start(3)
(25) end if
```

4.3 Specialized Methodologies

In all of the (n, k)-Threshold Secret Sharing (ThSS) schemes, k is defined as the sufficient number of shares needed for the reconstruction of the distributed secret. Thus, it is the numerical value of adequacy for the reconstruction process of the ThSS. As discussed in Section 3.2, that value determines the resiliency of the network, which should be increased without increasing the value of the number of neighboring nodes.

We propose two different protocols that overcomes the problem, which differ in the name of adequacy. In the first solution, k shares are enough for a mesh node to reconstruct a desired value whilst in the second solution the number of enough shares is $k + 1$. Actually, the main difference of those proposed solutions is the use of the Secret Sharing method(s), which in turn differs the solutions with respect both to the level of security they provide and to the resiliency of the network.

In the following subsections, we describe how the master key shares are distributed among the mesh nodes, and how the reconstruction is performed, for both solutions.

4.3.1 DKE with use of ThSS

In this scheme, a ($m \times x$, k)-ThSS is applied, where $m \times x$ is defined as the total number of shares to be distributed among the mesh nodes.

Within the protocol, mesh nodes perform reconstruction while computing

the master public key shares and their user private keys. All the other computations/reconstructions are performed as described in Section 4.2.

In IBC, master public key is computed by the trusted authority via Equation 4.7.

$$MK^{pub} = MK^{priv} \times P \tag{4.7}$$

Since we distributed the value MK^{priv} among the mesh nodes, each mesh node MN_i that has already computed its master private key share, computes its master public key share, MKS_i^{pub}, by Equation 4.8.

$$MKS_i^{pub} = MKS_i^{priv} \times P \tag{4.8}$$

Thus, the actual value of the master public key can only be reconstructed by a collaboration of k such shares via Equation 4.9.

$$MK^{pub} = \sum_{i=1}^{k} MKS_i^{pub} \times l_i(0) \tag{4.9}$$

where, $l_i(0)$ is the Lagrange coefficient.

As for the user private key reconstruction, it is defined in IBC as in Equation 4.10.

$$PK_i = MK^{priv} \times Q_i \tag{4.10}$$

where, Q_i is the public key of a mesh node.

Because of the same abovementioned reasons, in our scheme, this computation corresponds to that of given by Equation 4.11 performed by a col-

laboration of k shareholders.

$$PK_j = \sum_{i=1}^{k} PKS_{i,j} \times l_i(0) \tag{4.11}$$

where, $l_i(0)$ is the Lagrange coefficient and $PKS_{i,j}$ is the user private key share of MN_j computed by MN_i.

When a mesh node MN_i receives a reply for its user private key generation request, it increments the number of shares it received according to the type of the replying mesh node. When MN_i receives sufficient number of shares of its user private key, then it can perform the corresponding computation, as given in Algorithm 6.

Algorithm 6 RECEIVE-PKG-REPLY ($PK_{senderAddr,myID}$)

```
(1)  if not received from senderAddr yet then
(2)     PKshares_senderAddr ← PK_senderAddr,myID
(3)     if MN_senderAddr is a MR then
(4)        receivedPKGreplies ← receivedPKGreplies + x
(5)     else
(6)        receivedPKGreplies ← receivedPKGreplies + 1
(7)     end if
(8)     if sufficient receivedPKGreplies received then
(9)        PK ← RECONCTRUCT − SECRET−
             WITH − SHAMIR−
             ThSS(PKshares, receivedPKreplies)
(10)    end if
(11) end if
```

4.3.2 DKE with use of both ThSS and AdSS

In this scheme, an Additive Secret Sharing (AdSS) is applied along with a $(m \times x,\ k)$-ThSS: the master private key of the network is defined as in Equation 4.12.

$$MK^{priv} = MK^{priv,1} + MK^{priv,2} \tag{4.12}$$

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 as described in Section 4.2.

As the sharing method implies, for any type of reconstruction, i.e. master public key reconstruction and user private key reconstruction, a share from a mesh router is now a must. The important point here is that each mesh node needs to keep track of the identities of the mesh nodes from which they receive a share.

As mentioned in Section 4.3.1, mesh nodes perform reconstruction while computing either the master public key or their user private keys. For both of the reconstruction operations, a mesh node MN_i should have k shares computed with $MK^{priv,2}$ and a share computed with $MK^{priv,1}$.

Upon the receipt of sufficient number of shares, the master public key is reconstructed via Equation 4.13.

$$MK^{pub} = (\sum_{i=1}^{k} MKS_i^{pub} \times l_i(0)\ ) + MKS_j^{pub} \tag{4.13}$$

where, MKS_i^{pub} is computed by a mesh node MN_i as given in Equation 4.14 and MKS_j^{pub} is computed by a mesh router MR_j via Equation 4.15.

$$MKS_i^{pub} = MKS_i^{priv,\,2} \times P \tag{4.14}$$

$$MKS_j^{pub} = MKS_j^{priv,\,1} \times P \tag{4.15}$$

On the other hand, a mesh node MN_i can reconstruct its user private key as in Equation 4.16.

$$PK_i = (\sum_{j=1}^{k} PKS_{j,\,i} \times l_j(0)\,) + PKS_{p,\,i} \tag{4.16}$$

where, $PKS_{j,\,i}$ is computed by a mesh node MN_j via Equation 4.17 and $PKS_{p,\,j}$ is computed by a mesh router MR_p as in Equation 4.15.

$$PKS_{j,\,i} = MKS_i \times Q_j \tag{4.17}$$

$$PKS_{j,\,p} = MKS_p \times Q_j \pmod q \tag{4.18}$$

5 Security and Resiliency Analysis

In this section, after analysing to what extent the security requirements of the Wireless Mesh Networks (WMNs) are met, we will analyse the resiliency of the network.

5.1 Security Analysis

When the key establishment process finishes, each mesh node withholding its user private key, the mesh nodes can utilize the network services. Since we assumed that the identities of the mesh nodes uniquely identifies themselves, denial of a mesh node of being what he claimed to be is impossible. The confidentiality of the data transmitted along with authentication and non-repudiation is achieved by encrypting the message both with the sending node's private key and the public key of the destined node. Moreover, with the session key exchanged between the communicating nodes by the first message transmitted, integrity is achieved.

Therefore, all of the security requirements listed in Section 2.1.2 are met with the utilization of IBC, which is described in Section 2.3.1.

5.2 Resiliency Analysis

The resiliency of the network is the maximum number of compromised mesh nodes by which the security of the network is not affected. If an adversary

compromises a number of mesh nodes holding a total of k shares of the master private key, he can compute all the user private keys. Therefore, the resiliency of the network can be increased by increasing the threshold value.

In the following subsections, we analyse both DKE with ThSS and DKE with ThSS and AdSS with respect to the resiliency of the network.

5.2.1 Resiliency Analysis of DKE with ThSS

In this scheme, each mesh router has x shares while each mesh client has 1 share of the master private key and we are using a $(m \times x,\ k)$-ThSS scheme, where m is the number of mesh routers. 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 Equation 5.1.

$$k < (q \times x) + p \tag{5.1}$$

For instance, in a network with 3 mesh routers and 4 mesh clients, where the master private key is distributed in a (6, 4)-ThSS fashion, each mesh router has 2 shares. In such a network, an adversary can compute all the user private keys if he compromises either 1 mesh router and 2 mesh clients or 2 mesh routers or 4 mesh clients. Thus, this network is resilient to either $q = 1$ or $p = 3$, where q is the number of captured mesh routers and p is the

number of captured mesh clients. When we increase the threshold value to 6, an adversary prospers if he compromises either 3 mesh routers or 2 mesh routers and 2 mesh clients or 1 mesh router and 4 mesh clients. Thus, the resiliency of the network is satisfied when either 2 mesh routers or 4 mesh clients are compromised.

5.2.2 Resiliency Analysis of DKE with ThSS and AdSS

As mentioned in Section 4.3.2, this scheme ensures that a mesh router will always contribute to any of the reconstruction processes. Therefore, in order for an adversary to be successful, he needs to capture a mesh router. 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 resilient to the number of captured mesh routers and mesh clients, as decribed in the previous subsection.

6 Communication and Computational Overheads

Let us assume a WMN with $n = m + l$ nodes, where m is the number of mesh routers and l is the number of mesh clients. Additionally, let us assume that each mesh router holds x shares. Retaining those, we examine the communication and computational overheads introduced by our proposed solutions in the following subsections.

6.1 Communication Overhead

The communication overhead is introduced by the master key generation and distribution along with the user private key generation operations.

Since we disarranged the roles of the trusted third parties (TTPs) defined in IBC and Shamir's ThSS schemes, explained in Sections 2.3.1 and 2.3.2, the master key of the network is generated collaboratively. As described in Section 4.2, mesh routers are the ones to construct the master private key shares and distribute them to the mesh clients.

The generation of the master private key shares requires at least[8] $m \times (m-1)$ packets to be sent of each is a unicast message. For the other operations performed following this phase, there are a number of things that affect the number of packets sent: number of mesh clients realizing the first fraction of the operations is finished, number of mesh nodes that can respond to a request, number of mesh nodes that computed their master private key

[8]If a drop occurs, the packets are retransmitted.

shares, etc. Nevermore, none of the operations, i.e. master public and user private key generations, master private key share distribution, introduce a larger number of packet transmissions. In other words, the number of packets transmitted after master private key share generation is considerably small. As a consequence, the communicational complexity of the proposed solutions is $O(m^2)$ in terms of the number of packets transmitted.

6.2 Computational Overhead

The computational overhead is introduced by the use of Identity-based Cryptography (IBC) along with both Threshold Secret Sharing (ThSS) and Additive Secret Sharing (AdSS), described in Sections 2.3.2 and 2.3.2.

First of all, each mesh router distributes its randomly selected secret, which contains $m \times (k-1)$ modular exponentiation and $(m+1) \times (k-1)$ modular addition operations. As the receipt of $m-x$ subshares, each mesh router performs a modular addition of $m \times x$ values. After a mesh router computes its master private key share, it computes the master public key share of its own, which consists of an ECC multiplication. Then, for the computation of the partial shares that will be sent to the corresponding mesh clients, k modular multiplications and k modular additions are performed. Finally, each mesh client reconstructs its master private key share, the master public key and their user private keys seperately by $3k \times (k-1)$ modular multiplications along with $k \times (k-1)$ modular inverse operation, k ECC multiplications and k ECC additions. For those reconstructions to

be carried out, each mesh node that responds to a request performs 1 ECC multiplication for the computation of the requested share. As for the master public key and the user private key computation of the mesh routers, the same operations are used. The total computational overhead can be found in Table 2 with respect to the type of operations performed.

Table 2: Computational Overheads for DKE with ThSS

Modular Exponentiation	$m^2 \times (k-1)$
Modular Addition	$m \times ((m+1) \times (k-1) + (m \times x + k))$
Modular Multiplication	$m \times (k + 6k \times (k-1) + 3k \times l \times (k-1))$
Modular Inverse	$2m \times k \times (k-1) + l \times k \times (k-1)$
ECC Addition	$(2m \times k) + k \times l$
ECC Multiplication	$m \times (m + 2k) + (k \times l)$

As for the second proposed solution, DKM with ThSS and AdSS, we have the computational overhead introduced by AdSS along with the above-mentioned ones. In this solution, AdSS is used only in the reconstruction operations and involves an ECC addition. Since we have 2 reconstruction operations for each mesh node, i.e. master public key and user private key reconstructions, and an additional reconstruction operation for each mesh client, i.e. master private key share reconstruction, a total of $2m + 3l$ ECC additions is performed. Neverthless, for each of these reconstruction requests, an additional share is computed by an ECC multiplication.

7 Performance Evalution

We used Network Simulator 2 (ns2) [17], which is an open source discrete event network simulator, to evaluate the performance of the solutions that we propose. In the following subsections, we present our simulation setup, we introduce the implemetation details and finally, we discuss the simulation results.

7.1 Simulation Setup

Since we propose two different solutions for secure key generation and distibution in WMNs, we simulated two different scenarios. For each scenario, we modeled the network as having $n = 30, 40, 50, \ldots, 100$ nodes within an area of 2000×2000 square meter. Since the make the assumption that the mesh routers cover the network area, we have 25 mesh routers in each model and each has 2 shares of the master private key. In the simplest form, the mesh routers dwell on the coordinates as to cover the network area. Each mesh router is in the transmission range of its neighboring mesh routers. On the other hand, mesh clients are disposed within the area randomly. Additionally, we simulated the behavior of the network for the threshold values $k = 2, 4, 6, 8, 10, 12$.

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
- GCC 4.3.3 on Cygwin 1.5.25-15
- ns2 version 2.33

7.2 State of the Network

State of the network consists of the placement of the nodes and the options defined for them. Since the comparison of two different protocols is considered, the mesh nodes are placed at the same coordinates on each protocol. However, the coordinates of the mesh clients are selected randomly within the specified area.

As for the options of the nodes, there are several of them described below:

7.2.1 Channel, MAC and Network Interface Types

As the medium implies the channel type is wireless channel and thus the MAC type is 802.11. However, it is important to mention that the MAC type that we used is implemented by the company named Mercedes [20]. This is because of the fact that it is more stable than the one that is defined inside the ns2. Secondly, the network interface type used is the wireless physical layer and the version that Mercedes implemented is used due to the same stability concerns.

7.2.2 Antenna and Radio Propogation Models

Omni-directional antenna with a transmission range of 375 meters is used as for the antenna model. The reason we selected this type is that it is both easy and inexpensive to set and use them. As for the radio-propagation model, it is defined as two ray ground, which considers both the direct path and a ground reflection path; giving an accurate prediction at long distances [21, 17].

7.2.3 Queue Type

The interface queue type is selected as the priority queue defined under drop tail queue. It implements FIFO scheduling and drop-on-overflow buffer management among the classes of the same priority level [17]. Upon a receipt of a message, a node checks if the block flag, which contains information on whether a packet is already in process or not, is set. If the flag is not set, the mesh node sets the block flag and processes the packet. Otherwise, the mesh node puts the packet into the queue for later processing. After a packet is processed, the block flag is released with a callback function.

At this point, it is important to mention that using a queue that implements single forwarding affects the fairness of the system. For instance, even in a network with three nodes, when two of the nodes want to send a packet to the third node at the same instant of time, one of the packets came across with starvation. As a result, the throughput of a node decreases as the number of nodes increases [10].

7.2.4 Routing Protocol

There are three types of routing protocols that are proposed by ad hoc networks, thus applicable to WMNs: proactive, reactive and hybrid [4]. The proactive routing protocols act just as the same as the routing protocols that are designed for wired networks; at least one route is retained to any destination. In opposition to those, in the reactive routing protocols a route is found if and only if a source node has data to send to a destination node that it has not have a route to send the data yet. As for the hybrid routing protocols, as its name implies, it is the combination of proactive and reactive routing protocols.

Since WMNs are not stable, i.e. new nodes may join and/or existing nodes may leave the network, the proactive routing protocols cannot be used. The selected routing protocol should be able to construct a new route when an existing node is failed to operate or decided to leave the network. Accordingly, it should be able to construct a new route for newly joining mesh nodes. In other words, it must be flexible in sense of changing topology. On the other hand, the protocols that have a high overhead and that require global information are not applicable to WMNs due to their constraints.

Considering those facts, we selected Dynamic Source Routing (DSR) as the routing protocol for our simulations.

7.2.5 Transport Layer Communication Protocol

We simulated our models with UDP during the first runs. Since the master private key share generation process needs the guarantee of all the packets to be delivered athough UDP does not, simulation results did not seemed satisfactory. Most of the mesh nodes were not even able to compute their master private key shares. Thus, we turned out attention on TCP.

The reason of this choice actually stems from the fact that TCP uses flow control. To be more clear, with the utilization of TCP, data is guaranteed to be delivered. On the contrary, since it is designed initially for wired networks, it mostly captures the packet losses that occur because of the buffer overflows. However, in WMNs the losses may occur not only by the buffer overflows but also due to collisions, mobility and the utilization of wireless links [11]. Therefore, TCP shows a lower performance than in wired networks. This is actually the reason that we introduced the timeout method mentioned in Section 4.

7.3 Implementation Details

7.3.1 Cryptographic Operation Latencies

Due to the fact that the computational latencies are not taken into account by ns2, they are computed separately and inserted into the protocol either just after a message is received or just before a message is sent. Those latencies include the time consumed in the Identity-base Cryptography (IBC)

setup, (n, k)-Threshold Secret Sharing (ThSS) setup, master private key share (MKS_i^{priv}) computation performed by mesh routers, master public key share (MKS_i^{pub}) and user private key share ($PK_{i,j}$) computations performed by the mesh nodes ; all of which remain same as the threshold value k changes and are given in Table 3.

Table 3: Static Latency Benchmark

Setup of IBC	$0,00013$ ms
Setup of (n, k)-ThSS	$0,070526$ ms
Computation of MKS_i^{priv}	$0,001712$ ms
Computation of MKS_i^{pub}	$0,00903$ ms
Computation of $PK_{i,j}$	$0,00937$ ms

Additionally, there are the latencies that are changing with respect to the threshold value k. Those latencies consist of the computation of the master private key partial share ($MKS_{i,j}^{priv}$) along with the reconstructions of MK^{pub} and MSK. The values of the dynamic latencies can be found in Table 4.

Table 4: Dynamic Latency Benchmark for the Second Protocol

k	computation of $MKS_{i,j}^{priv}$	reconstructions of MK^{pub} and MSK
4	$0,0002$ ms	$0,05106$ ms
6	$0,0003$ ms	$0,1011$ ms
8	$0,0004$ ms	$0,1679$ ms
10	$0,0006$ ms	$0,19178$ ms
12	$0,0007$ ms	$0,21726$ ms

We did not implemented IBC but we used the implemented one present in Multiprecision Integer and Rational Arithmetic C/C++ Library (Miracl) [15]. In this implementation, the cryptographic parameter q is defined as in Equation 7.1.

$$q = 2^{159} + 2^{17} + 1 \tag{7.1}$$

The other parameters are set as given in Algorithm 7 and has the following properties:

- $q \ / \ p + 1$
- $p \,\%\, 3 \ = 2$
- $p \,\%\, 8 \ = 3$
- $y^2 = x^3 + 1 \pmod{p}$

7.3.2 Performance Metrics

We consider two metrics for each model: elapsed time and success percentage.

Latency of Key Establishment is defined as the time difference between the deployment of the mesh nodes and the end of the processes of each node. As mentioned in Section 4.2, a mesh node can either compute its user private key or not, and that depends on the number of neighboring nodes that the mesh node has. Since we introduced a timeout period for the mesh nodes to understand whether they can compute its user private key or not; at the end, a mesh node or a mesh router either has its user private key or it quits trying to compute one. When the last node finishes its process, the latency of key establishment is realized.

Success Percentage for Private Key Generation is the ratio of the number of mesh nodes that can compute their user private keys to the total number of the mesh nodes present within the network. In other words, it is the percentage of the mesh nodes that can compute their user private keys.

7.4 Results

Figures 5 and 6 show the difference in the success ratio for the private key generation with respect to the changes in both n and k, where n is the total number of mesh nodes and k is the threshold value. As mentioned in Section 4.2.4, the private key generation service is carried out succesfully if and only if the requesting mesh node receives a total of k shares from its neighboring nodes, who 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. 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 do the computation. As we increase the value of k, the success ratio decreases; meaning that the nodes cannot compute their user private keys due to the absence in the received shares. On the other hand, as we increase the total number of mesh nodes, for the same value of k, we achieve a higher success ratio. This is because of the fact that the number of neighboring nodes increase as the network size increases. In other words, more shares become accessible by the requesting node.

If we were to compare those two different patterns, a decrease in the success percentage in Figure 6 with respect to Figure 5 is clearly seen. The reason being is that in the protocol that corresponds to Figure 6, receiving a share from a mesh router is a must. Thus, if a mesh node cannot receive a share from a mesh router, for any reason[9], even if it receives k shares from other mesh nodes, it cannot reconstruct its user private key. Essentially, this is one of the tradeoffs between the two proposed protocols: as the security level increases, the success percentage decrease.

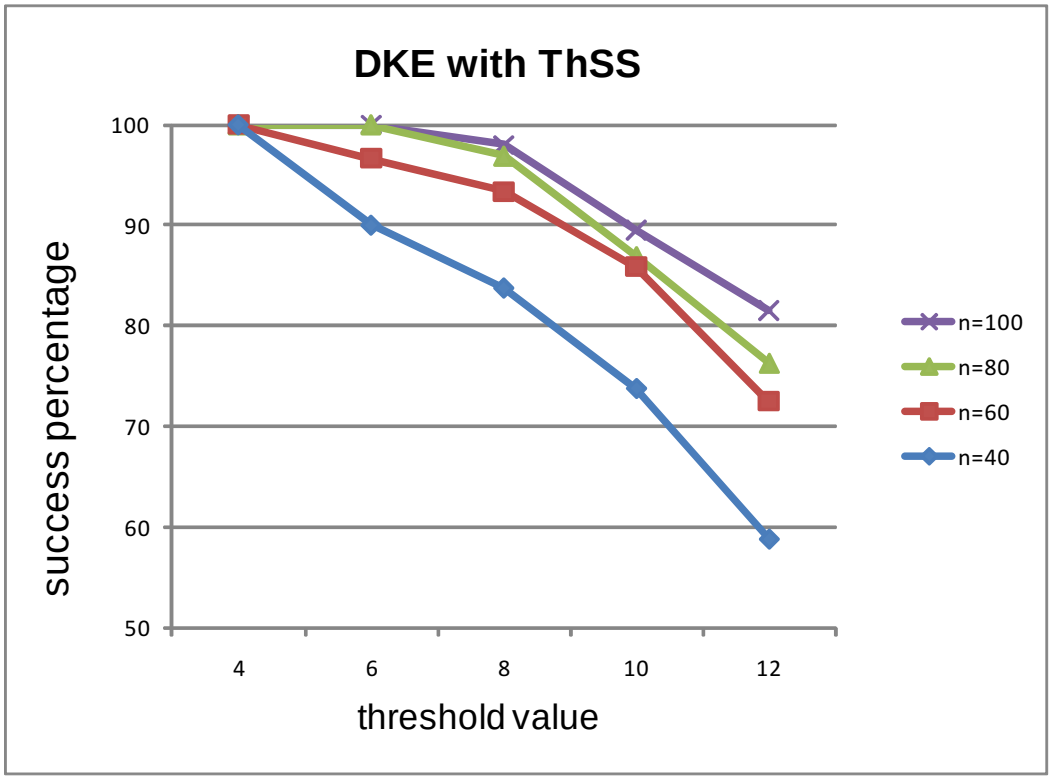

Figure 5: Success Percentage of DKE with ThSS

[9]With the increase in the network size, the number of packets transmitted increases; yielding more packets to collide and/or drop.

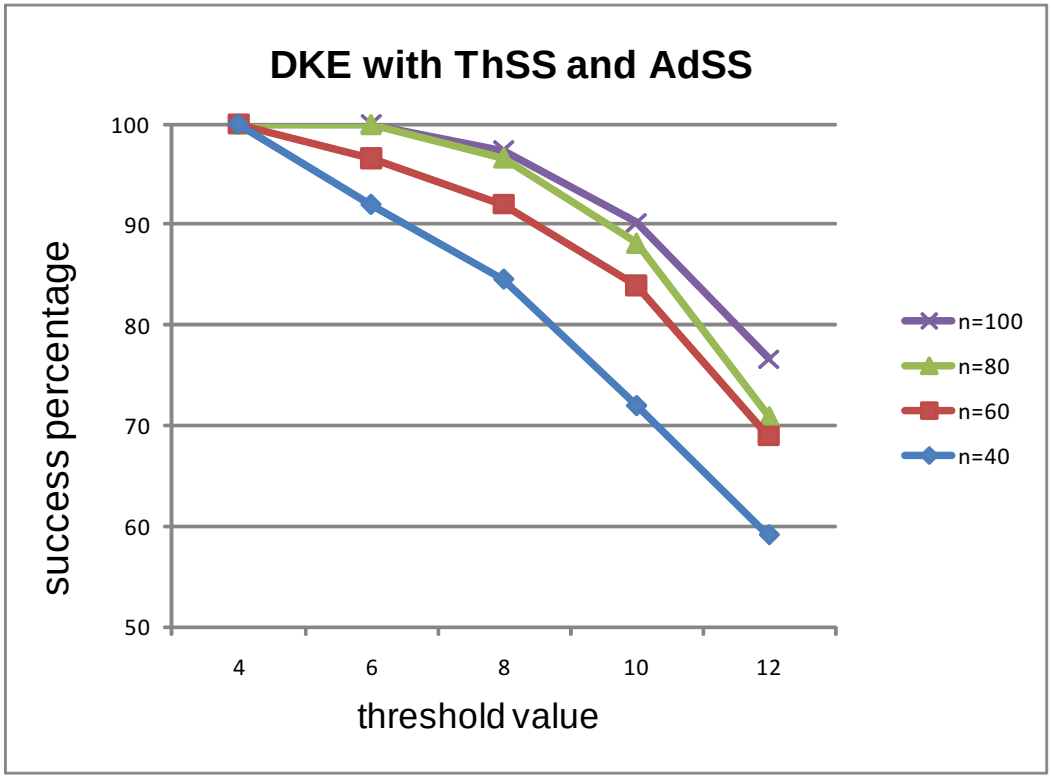

Figure 6: Success Percentage of DKE with ThSS and AdSS

In the Figures 7 and 8 the time diversification is examined with respect to the changes in n and k. Since the most of the burden is on the first phase, i.e. mesh routers generate the master private key shares, the latency of key establishment does not 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 values of k. However, as we increase k, the decrease in the success percentage results in an increase in the latency. This is due to the fact that some of the nodes cannot compute their private keys and they use the timeout method proposed, described in Section 4.2.4.

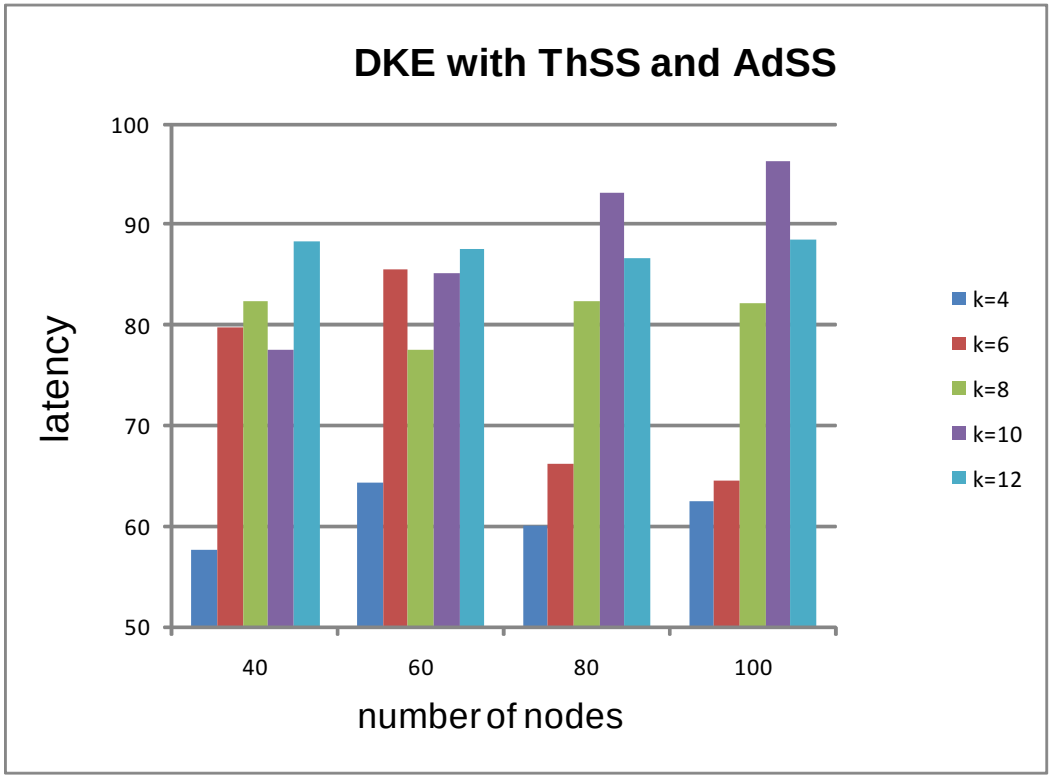

Figure 7: Latency of DKE with ThSS

Also we adopt the proposed schemes to an ad hoc network, as it is done in [6]. The main difference between an ad hoc network and a WMN is their infrastructures. In ad hoc networks, connectivity depends on the movements of the end users because of the fact that they do not possess dedicated nodes as mesh routers. Therefore, the network becomes less resilient in comparison to WMNs. Essentially, an ad hoc network can be considered as a subset of WMNs. In the knowledge of that information, in an ad hoc network, the master private key share generation is performed with the collaboration of all the nodes present in the network. As expected, this increases the number of packets transmitted, which yields to a larger number of collisions and thus

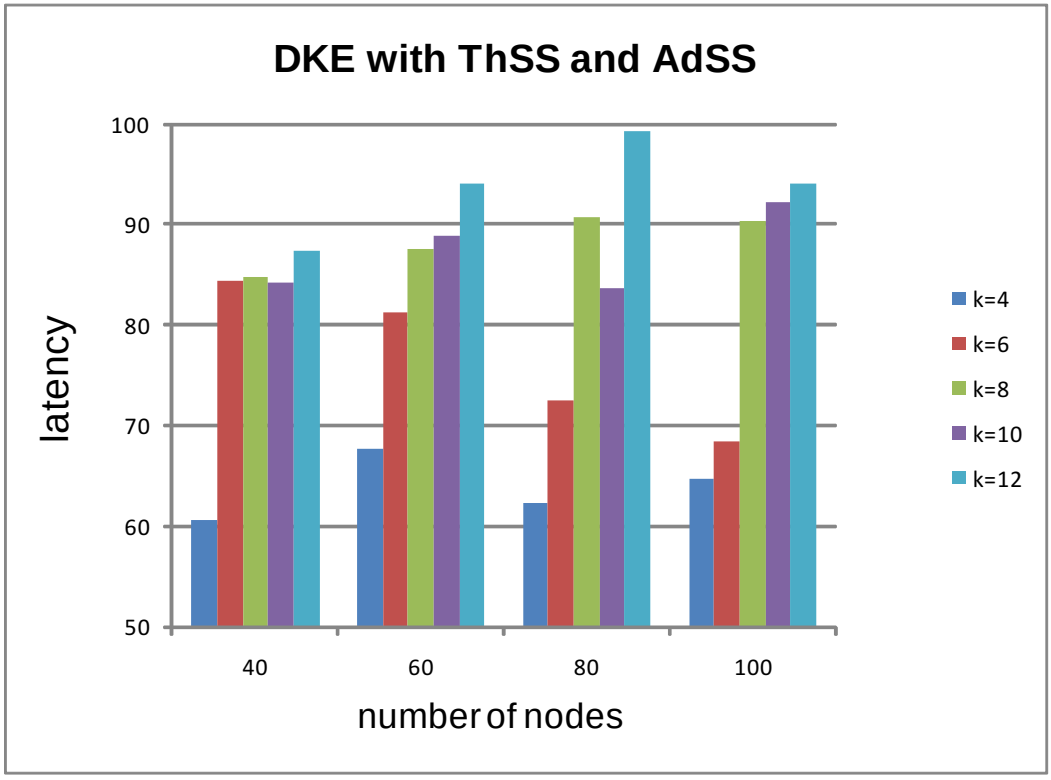

Figure 8: Latency of DKE with ThSS and AdSS

larger number of drops.

The success percentage for private key generation and the latency of key establishment show a pattern as in Figures 9 and 10. The timeout method used for the drops becomes unsatisfactory in this scheme. This results in a network that is inadequate for the values $n > 60$ no matter the value of k is, and for the values $k > 8$ no matter the network size is. Besides, because of the drops, some of the mesh nodes cannot even compute their master private key shares. Thus, the number of neighboring nodes of a requesting node that has computed its master private key share lessens. This is why a sharp drop

is seen as the threshold value increases. As for the latency, all the values seem to be as expected. An increase in the value of k increases the latency, just like the effect of the increase in the total number of nodes.

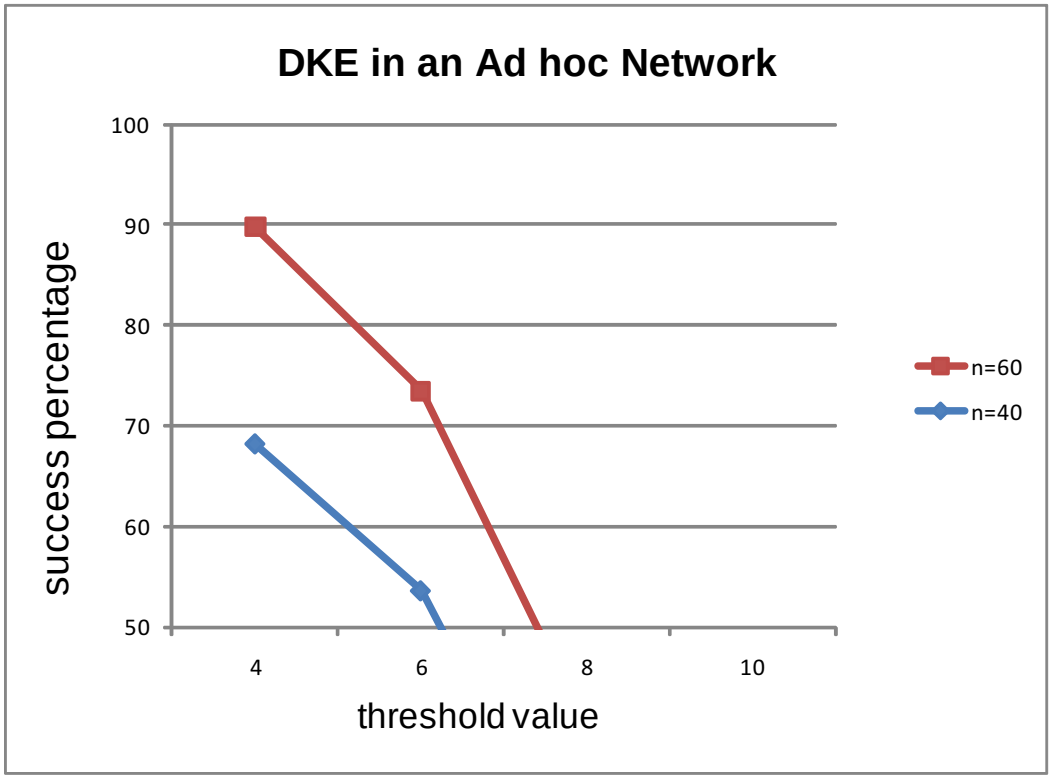

Figure 9: Success Percentage for an Ad hoc Network

As we compare these results with that of ours, we can easily realize that the success percentages for private key generation are higher in our schemes. This is due to the abovementioned disadvantage, i.e. increased number of drops, which unfortunately stems from the infrastructural characteristics of the ad hoc networks. Moreover, the latency of key establishment is higher in the ad hoc version even the number of successful private key generation operations is lesser. This actually is related to the timeout method used.

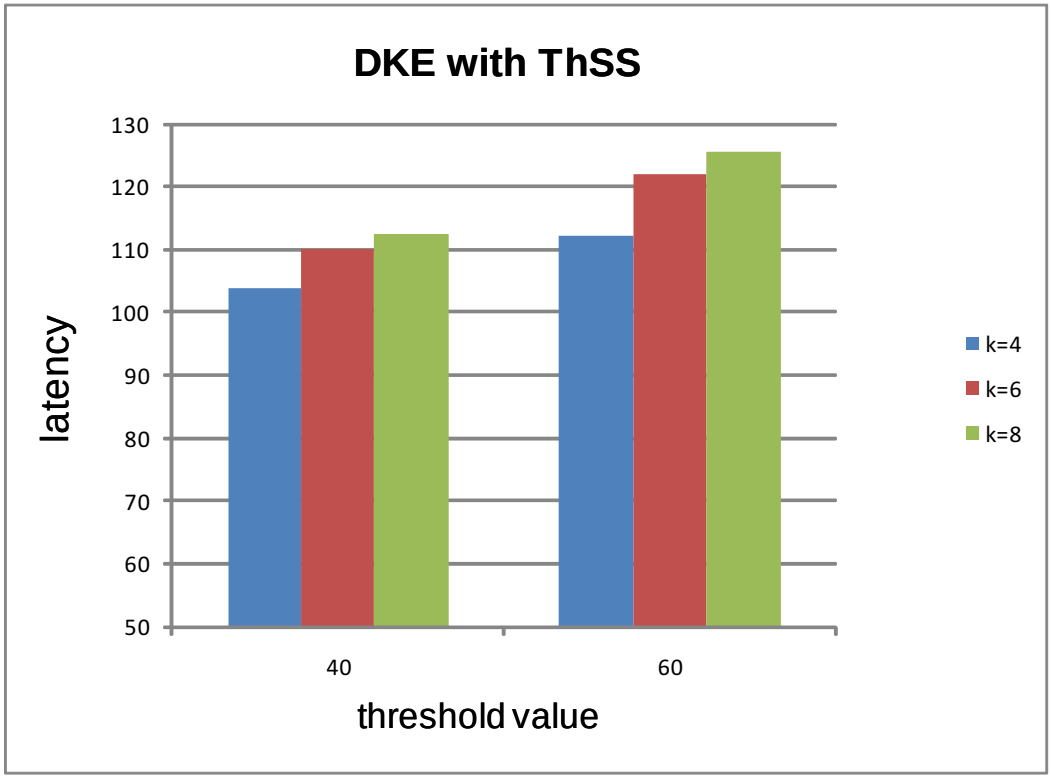

Figure 10: Latency for an Ad hoc Network

To be more clear, since the number of nodes that are capable of performing the generation and/or reconstruction operations is reduced, the time passes during the retrial increases.

Consequently, for the success percentage, our scheme shows a better performance up to %32 for $n = 40$ and up to %37 for $n = 60$. As for the latency, considering the worst case values, our scheme accomplishes the operations almost %23 faster.

8 Conclusions and Future Work

Wireless Mesh Networks (WMNs) are an emerging research area representing a good solution for providing low-cost and high-speed network services for the end users. However, the achievement of the security requirements is not trivial due to the constraints they have. On the contrary, key establishment is the most important and critical security concern that any type of wireless network need to have.

In this thesis, we propose two efficient and secure key establishment protocols designed with respect to the advantages and disadvantages of WMNs. On the basis, they make use of Identity-based Cryptography (IBC), which eliminates the necessity of the certificate based public key distribution indispensible in the conventional PKI-based schemes. The problems arise from the assumptions of IBC are eliminated by the use of a (n, k)-Threshold Secret Sharing scheme; trusted authority is abrogated with the collaborative generation of the secrets. Additionally, with the utilization of a variant of Shamir's Threshold Secret Sharing scheme, resiliency of the network is increased. Most importantly, a more secure solution appeared by the adoption of the Additive Secret Sharing.

Our simulations show that 100% of the mesh nodes can compute their private keys within at most 60 seconds, regardless of the value of the number of mesh nodes for the threshold value 4. For the worst case, i.e. a network with 40 nodes performing at the threshold level 12, at least 58% of the mesh nodes can compute their private keys within 90 seconds on the average. As we

increase the threshold value, the success percentage decreases, as expected. However, with such increase the effect of the number of nodes appears: the larger the number of nodes, the larger the success percentage.

However, as in all the systems, not always everything shapes up; there always seems to be a tradeoff in between the requirements. The simulation results implied a drawback for DKE with ThSS and AdSS: with the increase in the security level, the latency of key establishment is increased and the success percentage for the private key generation is decreased as compared to DKE with ThSS. Nevertheless, the difference between these two schemes are not too much, which makes DKE with ThSS and AdSS still useful for more security-demanding applications.

Above all, there are still some details remained intact and needs to be considered. First of all, during the process of collaboratively computing the share of the master private key, the messages are transmitted in plain; i.e. unecrypted. An authentication mechanism for the mesh nodes should be provided. Secondly, since it is assumed that there is no mutual trust in between the mesh nodes, the subshares generated and distributed by a mesh node should be verifiable by the receiving node. This can be achieved by adopting one of the verifiable secret sharing scheme proposed [18, 19]. Finally, as mention in Section 7.3.1, the benchmarks are taken from the runs used by the implemented IBC that is embedded in Miracl. However, those cryptographic parametes do not fit with that of ThSS's. Thus, a compatible implementation of IBC can be carried out.

APPENDIX - Algorithms used in the cryptographic approaches

Algorithm 7 is the setup phase and Algorithm 8 is the extract phase of the Identity-based Cryptography (IBC). Algorithms 9 and 10 are used in the extact phase of IBC. On the other hand, Algorithms 11 and 12 define how the secret is generated, distributed and reconstructed in an (n, k)-Threshold Secret Sharing (ThSS) scheme.

Algorithm 7 SETUP-IBC (q)

(1) t $\leftarrow 2^{511}/2q$
(2) $s \leftarrow (2^{511}-1)/2q$
(3) while true
(4) $\quad n \leftarrow$ random integer
(5) $\quad n \leftarrow n \mod t$
(6) $\quad p \leftarrow 2(n \times q) - 1$
(7) $\quad$ if p is a prime then
(8) $\quad\quad$ break
(9) $\quad$ end if
(10) end while
(11) $coef \leftarrow 2n$
(12) $EC \leftarrow$ elliptic curve
(13) while true
(14) $\quad temp \leftarrow$ random point
(15) $\quad cube \leftarrow temp^{\frac{p+1}{3}}$
(16) $\quad cube \leftarrow cube^{p-1}$
(17) $\quad$ if $cube$ is unity then
(18) $\quad\quad$ break
(19) $\quad$ end if
(20) end while
(21) while true
(22) $\quad P \leftarrow$ random point
(23) $\quad P \leftarrow P \times coef$
(24) $\quad$ if $P \neq 0$ then
(25) $\quad\quad$ break
(26) $\quad$ end if
(27) end while
(28) $s \leftarrow$ random integer
(29) $Ppub \leftarrow s \times P$

Algorithm 8 EXTRACT-IBC (s, id)

(1) $Q_{id} \leftarrow MAP - TO - POINT(id)$
(2) $D_{id} \leftarrow s \times Q_{id}$
(3) $privKey \leftarrow GET - X(D_{id})$
(4) `return` $privKey$

Algorithm 9 GET-X (y, p)

(1) $t \leftarrow ((y+1) \times (y-1)) \,\%\, p$
(2) `return` $(t^{\frac{(2p-1)}{3}}) \,\%\, p$

Algorithm 10 MAP-TO-POINT (id)

(1) $y \leftarrow H1(id)$
(2) $x \leftarrow GET - X(y)$
(3) $Q \leftarrow$ `set point with` x `and` y
(4) `return` Q

Algorithm 11 SHARE-SECRET-WITH-SHAMIR-ThSS (s, n, k)

```
(1)  coefficients : integer array of size k − 1
(2)  sharedDataTmp : integer array of size n
(3)  i ← 0
(4)  while i < k − 1 do
(5)     coefficients_i ← random integer
(6)     coefficients_i ← coefficients_i mod q
(7)     i ← i + 1
(8)  end while
(9)  i ← 0
(10) while i < n do
(11)    sharedDataTmp_i ← 0
(12)    j ← 0
(13)    while j < k − 1 do
(14)       coeffTmp ← ((i + 1)^(j+1)) mod q
(15)       sharedDataTmp_i ←(sharedDataTmp_i
             +(coefficients_j × coeffTmp)) mod q
(16)       j ← j + 1
(17)    end while
(18)    i ← i + 1
(17) end while
(18) sharedData ← template array of size n
(19) i ← 0
(20) while i < n do
(21)     sharedDataTmp_i ← (sharedDataTmp_i + s) mod q
(22)     sharedData_i.data ← sharedDataTmp_i
(23)     sharedData_i.id ← i + 1
(24)     i ← i + 1
(25) end while
(26) return sharedData
```

Algorithm 12 RECONSTRUCT-SECRET-WITH-SHAMIR-ThSS ($sharedData$, m)

```
(1)  nominators : integer array of size m
(2)  denominators : integer array of size m
(3)  i ← 0
(4)  while i < m do
(5)     nominators_i ← 1
(6)     denominators_i ← 1
(7)     j ← 0
(8)     while j < i do
(9)        if i ≠ j then
(10)          nominators_i ← nominators_i
                ×sharedData_j.id
(11)          denominators_i ← denominators_i
                ×(sharedData_j.id − sharedData_i.id)
(12)          nominators_i ← (nominators_i×
                denominators_i^{-1}) mod q
(13)       end if
(14)       j ← j + 1
(15)    end while
(16)    i ← i + 1
(17) end while
(18) i ← 0
(19) while i < m do
(20)    nominators_i ← (nominators_i×
          sharedData_j.data) mod q
(21)    i ← i + 1
(22) end while
(23) reconstructedData ← 0
(24) i ← 0
(25) while i < m do
(26)    reconstructedData ← (reconstructedData
          +nominators_i) mod q
(27)    i ← i + 1
(28) end while
(29) return reconstructedData
```

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. pages 213–229. 2001.

[4] M. E. M. Campista, P. M. Esposito, I. M. Moraes, L. H. M. K. Costa, O. C. M. B. Duarte, D. G. Passos, C. V. N. De Albuquerque, D. C. M. Saade, M. G. Rubinstein, and M. G. Rubinstein. Routing metrics and protocols for wireless mesh networks. *Network, IEEE*, 22(1):6–12, 2008.

[5] C. Cocks. An identity based encryption scheme based on quadratic residues. pages 360–363. 2001.

[6] 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.

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

[8] P. Gupta and P. R. Kumar. The capacity of wireless networks, 1999.

[9] 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.

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

[11] S. Lakshmanan, R. Sivakumar, and K. Sundaresan. Multi-gateway association in wireless mesh networks. *Ad Hoc Netw.*, 7(3):622–637, 2009.

[12] 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.

[13] G. Li. An identity-based security architecture for wireless mesh networks. *Network and Parallel Computing Workshops, IFIP International Conference on*, 0:223–226, 2007.

[14] R. Lu, Z. Cao, and X. Dong. A new practical limited identity-based encryption scheme. *Fundam. Inf.*, 80(4):461–474, 2008.

[15] Multiprecision Integer and Rational Arithmetic C/C++ Library. `http://www.shamus.ie/`.

[16] J. H. K.-C. S. H. Muhammad Shoaib Siddiqui, Syed Obaid Amin. Mhrp: A secure multi-path hybrid routing protocol for wireless mesh networks. *IIEEE Conference*, 2007.

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

[18] T. P. Pedersen. A threshold cryptosystem without a trusted party. In *EUROCRYPT*, volume 547, pages 522–526. Springer-Verlag, 1991.

[19] T. P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In *CRYPTO '91: Proceedings of the 11th Annual International Cryptology Conference on Advances in Cryptology*, pages 129–140, London, UK, 1992. Springer-Verlag.

[20] D. J. M. T.-M. L. D. Q. Chen, F. Schmidt-Eisenlohr and H. Hartenstein. Overhaul of IEEE 802.11 modeling and simulation in ns-2. `http://dsn.tm.uni-karlsruhe.de/english/Overhaul_NS-2.php/`, 2008.

[21] M. Rahnema. *UMTS Network Planning, Optimization, and Inter-Operation with GSM*. Wiley-IEEE Press, 2007.

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

[23] A. Shamir. Identity-based cryptosystems and signature schemes. pages 47–53. 1985.

[24] M. S. Siddiqui and C. S. v. 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.

[25] M. S. Siddiqui and C. S. v. 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.

[26] W. Stallings. *Cryptography and Network Security: Principles and Practice (2nd Edition)*. Prentice Hall, 2nd edition, July 1998.

[27] W. Trappe and L. C. Washington. *Introduction to Cryptography with Coding Theory (2nd Edition)*. Prentice Hall, July 2005.

[28] X. Wu and N. Li. Achieving privacy in mesh networks. In *SASN '06: Proceedings of the fourth ACM workshop on Security of ad hoc and sensor networks*, pages 13–22, New York, NY, USA, 2006. ACM Press.

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

[30] Y. Zhang, Y. Fang, and S. Member. Arsa: An attack-resilient security architecture for multihop wireless mesh networks. 2008.

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