Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2011, Article ID 807684, 17 pages
doi:10.1155/2011/807684

Research Article

A Family of Key Agreement Mechanisms for Mission Critical Communications for Secure Mobile Ad Hoc and Wireless Mesh Internetworking

Ioannis G. Askoxylakis,[1,2] Theo Tryfonas,[2] John May,[2] Vasilios Siris,[1] and Apostolos Traganitis[1]

[1] *Foundation for Reserach and Technology-Hellas, Institute of Computer Science, N. Plastira 100, 70013 Heraklion, Greece*
[2] *Faculty of Engineering, University of Bristol, Queen's Building University Walk, Clifton, Bristol BS8 1TR, UK*

Correspondence should be addressed to Ioannis G. Askoxylakis, asko@ics.forth.gr

Received 30 June 2010; Accepted 17 September 2010

Academic Editor: Christos Verikoukis

Future wireless networks like mobile ad hoc networks and wireless mesh networks are expected to play important role in demanding communications such as mission critical communications. MANETs are ideal for emergency cases where the communication infrastructure has been completely destroyed and there is a need for quick set up of communications among the rescue/emergency workers. In such emergency scenarios wireless mesh networks may be employed in a later phase for providing advanced communications and services acting as a backbone network in the affected area. Internetworking of both types of future networks will provide a broad range of mission critical applications. While offering many advantages, such as flexibility, easy of deployment and low cost, MANETs and mesh networks face important security and resilience threats, especially for such demanding applications. We introduce a family of key agreement methods based on weak to strong authentication associated with several multiparty contributory key establishment methods. We examine the attributes of each key establishment method and how each method can be better applied in different scenarios. The proposed protocols support seamlessly both types of networks and consider system and application requirements such as efficient and secure internetworking, dynamicity of network topologies and support of thin clients.

1. Introduction

Consider a disaster situation, such as an earthquake, a flood, or a terrorist attack, where the commercial network infrastructure is destroyed or out of order. The objective of the rescue workers is to set up quickly, efficiently, and easily a wireless network among them in order to help in a coordinated way the affected population. Their goal is to interconnect all their computing and communication devices, in a way that will enable them to share all necessary information securely, in a way that they could be sure that possible "high tech" terrorists/attackers in their range will not be able to disrupt or intercept the rescue efforts.

In real disaster scenarios, emergency response does not take place all at once. We usually observe an escalation in the presence of several groups of rescue workers and prioritized escalation of their efforts. In the beginning, we usually observe ad hoc groups working as independent teams that arrive at place independently. These teams gradually become part of coordinated action by a central disaster management entity, which requires more time to arrive at place, set up its infrastructure, and become operational. Approaching this scenario from a networking perspective, a sufficient approach would be the support of the initial groups of rescue workers by communication devices with mobile ad hoc networking capabilities. In this respect, an efficient networking solution for the support of the central disaster management entity would be the employment of adaptive, self-organized networks with advances networking capabilities, and redundant characteristics like wireless mesh

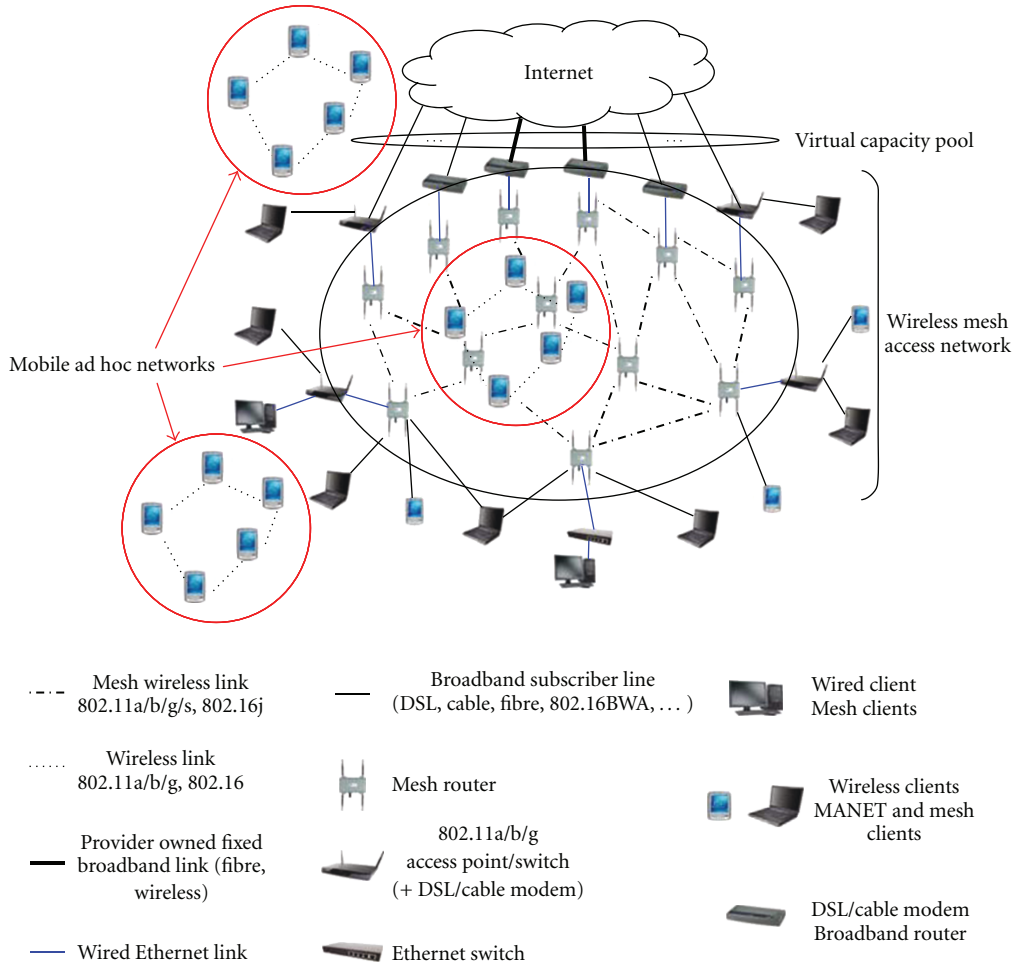

Figure 1: Network model.

networks. Seamless interworking of both types of networks would be a key requirement for such a scenario.

Security is a primary concern for providing protected communications in such environments where there is no available communication infrastructure and where networks of varying types and sizes must be established quickly and dynamically. Moreover, there might be situations where potentially large numbers of rescue workers, potentially from multiple government services or even nations must cooperate and coordinate their efforts in areas where natural or man-originated disasters have damaged or set temporarily out of order part or the entire telecommunication infrastructure. The unique nature and characteristics of mobile ad hoc Networks and wireless mesh networks make them ideal networking solution to the above situations. At the same time, their nature and characteristics pose a number of nontrivial challenges to their security design, architecture, and services.

In both MANETs and wireless mesh networks, like in any other type of network, trust cannot be created among the network nodes without the existence of predefined prior known information available to all nodes beforehand. This special kind of information is necessary in order to build trust between all participating nodes. A network is established among the existing nodes, if from this preexisting information known to all network nodes, we reach a state where a common *session key* is agreed among the nodes.

The technical goal is to make sure that no other entity outside the *group* should be able to gain access within the new network. However, since neither a certification authority nor a secure communication channel exists, potential attackers have the ability to eavesdrop and modify exchanged messages transmitted over the air. Additionally, since no central identification authority is present, group member impersonation is easy, jeopardizing the security of the whole system.

Considering all these issues, the main challenge that arises is the setting up of a wireless network where the legitimate members of a group will be able to establish a protected wireless network. Moreover, in the case where a new node arrives at place, desiring to become a member in an already established group, joining, without delaying or even intercepting the existing group, is also challenging. The case where a group member is captured by the enemy, and therefore the group key is compromised is also part of

the considered scenario. All the above considerations become even more challenging for the mobile ad hoc/wireless mesh internetworking scenario examined in this work.

The rest of the paper is organized as follows. In Section 2, we describe the system model. In Section 3, we describe the adversary model, and in Section 4, we present the security requirements. In Section 5, we present a review of the related work concerning two-party and multiparty key agreements, and we give a brief introduction on weak to strong authentication and the elliptic curve theory. In Section 6, we describe specific multiparty key agreement protocols and particularly, the BCC, the FCC algorithms, and the tetrahedral approach and examine their properties. Finally, in Section 7, we conclude with suggestions for future work.

2. System/Network Model

In this section, we consider a system/network model as illustrated in Figure 1. It consists of both wireless mesh networks and mobile ad hoc networks. While several detailed surveys on mesh network architectures can be found in the literature [1, 2], the proposed system model is the similar to the one defined for the EU-MESH Project (http://www.eu-mesh.eu/) as far as wireless mesh networking is concerned. Accordingly, this model a mesh network consists of mesh routers that form a network with very similar networking attributes and characteristics of a static wireless ad hoc network. The mesh routers can function either as gateways to the wired Internet, or as wireless access points for mobile mesh clients.

We assume that the mesh routers belong to multiple operators, and they cooperate for providing aggregate networking services to all of their mesh clients. In the disaster management scenarios, we consider as different operators different teams of rescue workers (firemen, policemen, etc.). Their cooperation model, which falls out of the scope of this paper, can be based on simple on field agreements or on business agreements similar to roaming agreements in the case of cellular networks. Mesh clients are mobile computing devices (smart phones, PDAs, netbooks, etc.) operated by customers that can be associated with one or more operators by contractual means.

The mesh network provides various services to its clients like Internet access, real-time communications within the mesh network, and so forth. In this model, the mesh network is also designed to provide QoS applications with client mobility support. This way mobile mesh clients can perform seamless handovers between access points.

In parallel to the wireless mesh architecture, in our system model, we have the presence of independent mobile ad hoc networks as shown in red in Figure 1. A MANET is a type of network, which is typically composed of equal mobile hosts that we call nodes. When the nodes are located within the same radio range, they can communicate directly with each other using wireless links. This direct communication is employed in a distributed manner without hierarchical control. The absence of hierarchical structure introduces several problems, such as configuration advertising, discovery, maintenance, as well as ad hoc addressing, self-routing, and security [3].

In our internetworking model, a MANET node can be also considered as a mesh client and can perform seamless handovers between access points of the mesh network or between the MANET and the mesh network.

3. Adversary Model

As usual, the first step in the identification of security requirements is the understanding of the potential attacks against the system. This understanding is summed up in the following adversary model that describes the classes of attackers, their objectives and their means to attack the network.

Taking into account the system model of mobile ad hoc and wireless mesh internetworking, the following types of attackers are identified.

External Attackers. These are attackers that have no legitimate access to the MANET or the wireless mesh network but they have appropriate equipment to use the wireless medium and interfere with the operation of the network protocols.

Compromised Nodes/Clients. These are legitimate node devices that have legitimate access to the MANET and/or the wireless mesh network services and they have been compromised by attackers (e.g., by stealing a device or by capturing a legitimate user in the field). The attackers have the knowledge to modify the behavior of these nodes and try to take advantage of this in order to interfere with the operation of the network or to gain illegal access to its services.

Dishonest Network Nodes/Clients. They are misbehaving end users that while they have legitimate access to the wireless networks and some or all of the network services, they try to take advantage of this in order to gain illegal access to services that are not subscribed to, or to obtain higher QoS in services that they are already subscribed.

Dishonest Network Operators. They are operators of the mesh infrastructure that do not honestly keep to cooperation agreements.

Next, we identify the following main objectives of attacks.

Denial-of-Service (DoS). The objective of this type of attack is to degrade the QoS provided by the mesh network and/or the MANET or even to completely disrupt the provided services. This is an objective of external adversaries.

Unauthorized Access to Services. This objective is mainly related to external adversaries and dishonest clients. Common services include internet access and real-time communications.

Unauthorized Access to Network Client Data and Meta-Data. Network client data are the messages exchanged in a service session and the corresponding objective is the violation of the confidentiality of the client whereas meta-data is information related to the client's location and service usage profile and

the objective is the violation of the privacy of the client. Primarily, this objective is related to external adversaries and dishonest network operators.

Fraudulent Improvement of Operator Profile. This could be the objective of dishonest operators that may mount attacks on the mesh network or specific network operators/competitors participating in the network in order to gain some advantage over them. This can be achieved either by reducing or destroying the reputation of the competitors, or by spuriously increasing their own reputation.

There is a broad range of attack mechanisms that can be used and combined in order to reach the goals described above. However, most of these mechanisms fall into either one of the following two categories.

(i) attacks on wireless communications (including eavesdropping, jamming, replay, and injection of messages, and traffic analysis).

(ii) compromising existing nodes (typically by physical tampering or logical break-in). The behavior of the fake or compromised nodes can be arbitrarily modified in order to help to achieve specific attack objectives. In such a scenario, the underlying security depends on the size and the randomness of the chosen password. However, the larger the password gets the more difficult it is to memorize and use. Moreover, since the response time is vital during emergency operations, the use of large passwords can be proved inconvenient. Therefore, the use of short, user-friendly passwords is an essential requirement;

(iii) setting up fake mesh routers or compromising unattended existing mesh routers.

4. Security Requirements

It is broadly known that security mechanisms cannot create trust [4]. The members of a team that wish to establish a group know and trust one another physically. Otherwise, they would never be able to achieve mutual trust regardless of the authentication mechanism used. Our goal is to exploit the existing physical mutual trust and create a secure group of communication for both types of networks that would operate in a seamless manner.

An efficient solution to this direction, without adding new requirements like the use of dedicated hardware (i.e., smart cards), would be a password authentication mechanism. A simple approach of a password-based authentication scheme could be the use of sufficiently large and randomly generated data strings employed as passwords. In such a scheme, all nodes could agree on a password and achieve mutual authentication supported by a trivial authentication protocol.

In such a scenario, the underlying security depends on the size and the randomness of the chosen password. However, the larger the password gets the more difficult it is to memorize and use. Moreover, since the response time is vital during emergency operations, the use of large passwords can be proved inconvenient. Therefore, the use of short, user-friendly passwords is an essential requirement.

The use of short passwords provides weak authentication since the password selection set is quite limited, and thus the corresponding authentication procedure is vulnerable to dictionary attacks [5]. Therefore, we need an authentication protocol that will lead to a reasonable degree of security even if the authentication procedure has been initiated from a small, weak password.

Below, we outline the main security requirements of the proposed architecture.

Weak-to-Strong Password-Based Authentication. Use of an authentication scheme that will lead to a reasonable degree of security although the authentication procedure has been initiated from a small, weak password.

Secure Authentication. Only the entities that hold the correct password will eventually become members of the network.

Forward Authentication. Even if a malicious partner manages to compromise a network entity in a later phase, he will still be unable to participate in the already existing network.

Contributory Key Establishment. The network is established when a session key is generated and agreed among all network nodes. The session key should be generated throughout in a contributory manner, by all participating entities.

Security Architecture for Thin Clients. In both types of networks, there are mobile devices/clients with limited processing power and energy consumption. The cryptographic algorithms used for authentication and key agreement should add minimal computational overhead.

Rare Key Reestablishment. Session key refreshments should be performed as rare as possible, since during every new key reestablishment session the network is unavailable for node communications.

Unified Security Architecture for Combined MANET-Mesh Secure Internetworking. The proposed key agreement mechanisms should apply in both types of networks, without requiring any network-specific adjustments.

5. Background Theory

5.1. Password-Based Key Exchange. Typical cryptographic protocols based on keys chosen by the users, are weak to password guessing attacks. Bellovin and Merritt [6] proposed a protocol called *Encrypted Key Exchange (EKE)* where a strong shared key is derived from a weak one. The basic concept of the generic protocol is the following: there are two parties A, B that share a password P. Both parties use a suitable symmetric cryptosystem but entity A has also the ability to create a random asymmetric key pair, (e_A, d_A). During the first step, A generates a random public key e_A and

encrypts it symmetrically using key P in order to produce $P(e_A)$. Then, A sends it to B

$$A : (A_{\text{id}}, P(e_A)) \longrightarrow B. \tag{1}$$

This message includes A's id in clear text.

Since A and B share the same password P, B decrypts the received message to obtain e_A. Node B generates a random secret key R and encrypts it in both asymmetric and symmetric cryptosystem using as an encryption key quantity e_A and P, respectively. So, B produces $P(e_A(R))$ and sends it to A

$$B : P(e_A(R)) \longrightarrow A. \tag{2}$$

Entity A now decrypts the received message to obtain R, generates a unique challenge challenge$_A$ and encrypts it with R to produce $R(\text{challenge}_A)$ and send it back to B,

$$A : R\left(\text{challenge}_A\right) \longrightarrow B. \tag{3}$$

Then, B decrypts the message to obtain A's challenge, generates a unique challenge B, and encrypts the two challenges with the secret key R to obtain $R(\text{challenge}_A; \text{challenge}_B)$. Node B is ready to transmit quantity $R(\text{challenge}_A; \text{challenge}_B)$ to node A

$$B : R\left(\text{challenge}_A; \text{challenge}_B\right) \longrightarrow A. \tag{4}$$

When A receives the message, it decrypts it to obtain challenge$_A$ and challenge$_B$, and it compares it with the previous challenge. If there is a match, A encrypts challenge$_B$ with R to obtain $R(\text{challenge}_B)$ and sends it to B

$$A : R\left(\text{challenge}_B\right) \longrightarrow B. \tag{5}$$

If the challenge response protocol has been successfully deployed, then the authentication process is successfully accomplished and both parties proceed, using the symmetric cryptosystem and the quantity R as the session key. However, this protocol has a major drawback. That is, the creation of the common session key R has taking place with unilateral prospective, that is, only by the entity that first initiate the whole procedure. Thus, the key agreement scheme is not contributory.

In [7], Asokan and Ginzboorg proposed a contributory version of the above protocol for both two party and multiparty case. Their proposal is described as follows.

(1) Two-party case

(i) $A \rightarrow B : A, P(e_A)$,

(ii) $B \rightarrow A : P(e_A(R, S_B))$,

(iii) $A \rightarrow B : R(S_A)$,

(iv) $A \rightarrow B : K(S_A, H(S_A, S_B))$,

(v) $B \rightarrow A : K(S_B, H(S_A, S_B))$,

where S_A, S_B are the random quantities generated from A, B, respectively, and K is the session key produced according the formula $K = F_1(S_A, S_B)$, where F_1 is an one way function, and $H()$ is a public hash function.

(2) Multiparty case

(i) $M_n \rightarrow \text{ALL} : M_n, P(E)$,

(ii) $M_i \rightarrow M_n : M_i, P(E(R_i, S_i)), i = 1, \ldots, n-1$,

(iii) $M_n \rightarrow M_i : R_i(\{S_j, j = 1, \ldots n\}), i = 1, \ldots, n-1$,

(iv) $M_i \rightarrow M_n : M_i, K(S_i, H(S_1, \ldots, S_n))$, forsomei,

where E is the Public key of M_n. S_i, for all i is the random quantities generated from M_i, and K is the session key produced according the formula $K = F_2(S_i)$, for all i. F_2 is an n-input one way function and $H()$ is a public hash function.

5.2. Password-Based Diffie-Hellman Key Exchange

5.2.1. Two Party Key Exchange. Diffie-Hellman is the first public key distribution protocol that opened new directions in cryptography [8]. In this important key distribution protocol, two entities A, B after having agreed on a prime number p and a generator g of the multiplicative group Z_p, can generate a secret session key. In [6], Bellovin and Merritt proposed a password authenticated key exchange which operates in the following way.

(i) A picks a random number, R_A calculates $P(g^{R_A} (\text{mod } p))$, and A sends A, $P(g^{R_A})$ to B; entity A's id is sent in clear text.

(ii) B picks a random number R_B and calculates $g^{R_B} (\text{mod } p)$. B uses the shared password P to decrypt $P(g^{R_A} \text{ mod } p)$ and calculates

$$g^{R_B R_A} \text{ mod } p. \tag{6}$$

(iii) The session key K is derived from this value by selecting a certain number of bits. Finally, a random challenge, challenge$_B$ is generated. Then, B transmits

$$P\left(g^{R_B} (\text{mod } p)\right), K\left(\text{challenge}_B\right). \tag{7}$$

(iv) A uses P to decrypt $P(g^{R_B} \text{ mod } p)$. From this, quantity K is calculated; K is in turn used to decrypt $K(\text{challenge}_B)$. A then generates a random challenge challenge$_A$. A sends

$$K\left(\text{challenge}_A, \text{challenge}_B\right). \tag{8}$$

(v) B decrypts $K(\text{challenge}_A, \text{challenge}_B)$, and verifies that challenge$_B$ is correct. B sends

$$K\left(\text{challenge}_A\right). \tag{9}$$

(vi) A decrypts to obtain challenge$_A$ and verifies that it matches the original message.

5.2.2. Elliptic Curve Diffie-Hellman. The original Diffie-Hellman algorithm is based on the multiplicative group modulo p. However, the elliptic curve Diffie-Hellman (ECDH) protocol is based on the additive elliptic curve group, and it is desribed below. We assume that two entities A, B have selected the underlying field, $GF(p)$ or $GF(2^k)$, the elliptic curve E with parameters a, b, and the base point P. The order of the base point P is equal to n. Also, we ensure that the selected elliptic curve has a prime order, in order to comply with the appropriate security standards [9, 10].

At the end of the protocol, the communicating parties end up with the same value K which represents a unique point on the curve. A part of this value can be used as a secret key to a secret-key encryption algorithm. We give a brief description of the protocol.

(i) Entity A selects an integer,

$$d_A : d_A \in [2, n-2]. \tag{10}$$

(ii) Entity B selects an integer

$$d_B : d_B \in [2, n-2]. \tag{11}$$

(iii) A computes $Q_A = d_A \times P$. The pair Q_A, d_A consists of A's public and private key.

(iv) B computes $Q_B = d_B \times P$. The pair Q_B, d_B consists of B's public and private key.

(v) A sends Q_A to B

$$A : Q_A \longrightarrow B. \tag{12}$$

(vi) B sends Q_B to A

$$B : Q_B \longrightarrow A. \tag{13}$$

(vii) A computes

$$K = d_A \times Q_B = d_A \times d_B \times P. \tag{14}$$

(viii) B computes

$$K = d_B \times Q_A = d_B \times d_A \times P. \tag{15}$$

Quantity K is now the common shared key between A and B. Moreover, it can also be used as a session key. Quantity n is the order of the base point P.

5.2.3. Password-Based Elliptic Curve Diffie-Hellman. The efficiency of elliptic curves in terms of security and calculation efficiency has been extensively discussed [10, 11, 12, 9a, 14]. Therefore, their employment in the password-based Diffie-Hellman process would significantly accelerate the key establishment procedure. The importance of this enhancement becomes even greater in the case of an emergency situation, where all actions should be performed in the fastest and more secure possible way consuming limited computing power.

We assume there two entities A, B that have agreed on the underlying field $GF(p), GF(2^p)$ on an elliptic curve E with coefficients α, β defined over the selected field, on the base point Q and the password P. The operation of the proposed protocol is as follows.

(i) A picks a random number $R_A : R_A \in [2, n-2]$, where n is the order of the base point Q and calculates $P(R_A \times Q)A$ sends

$$P(R_A \times Q) \tag{16}$$

to B; entity A's id is sent in clear.

(ii) B picks a random number $R_B : R_B \in [2, n-2]$ and calculates $R_B \times Q$. B also uses the shared password P to decrypt $P(R_A \times Q)$ and calculates

$$R_B \times R_A \times Q. \tag{17}$$

(iii) The session key K is derived from this value, perhaps by selecting certain bits. Finally, a random challenge challenge$_B$ is generated. B transmits

$$P(R_B \times Q), K\left(\text{challenge}_B\right). \tag{18}$$

(iv) A uses P to decrypt $P(R_B \times Q)$. From this, K is calculated; K is in turn used to decrypt $K(\text{challenge}_B)$. A then generates its own random challenge challenge$_A$. A sends

$$K\left(\text{challenge}_A, \text{challenge}_B\right). \tag{19}$$

(v) B decrypts $K(\text{challenge}_A, \text{challenge}_B)$ and verifies that challenge$_B$ is correct. B sends

$$K\left(\text{challenge}_A\right). \tag{20}$$

(vi) A decrypts to obtain challenge$_A$ and verifies that it matches the original message.

5.3. Efficient D-H-Based Multiparty Key Exchange

5.3.1. d-Cube Protocol Overview. For key establishment procedures in multiparty networks like MANETs and mesh networks, where several entities are involved, multiparty authentication protocols should be applied. A lot of research has been done in this direction. Becker and Wille [15] presented a method very efficient in terms of number of authentication rounds. According to this method, also known as d-cube protocol, all entities planning to participate in a network are initially arranged in a d-dimensional hypercube. Each potential network entity is represented as a vertex in the d dimensional-cube, and it is uniquely assigned a d-bit address. The addresses are assigned in a way so that two vertices connected along the ith dimension differ only in the ith bit. There are 2^d vertices each of which are connected to d other vertices.

5.3.2. DH d-Cube. Assume that there are $n = 2^d$ entities seeking to establish an ad hoc non infrastructural network. During the first step, each entity is assigned to a vertex in the hypercube, and it is given a unique d-bit address. The deployment of the address arrangement is out of the scope of this paper and will not be examined. The key establishment protocol is illustrated within d rounds. In every single round the entities are paired together, according to a specific procedure, and the Diffie-Hellman key exchange is performed. These pairwise operations are performed in parallel during every round. For example, during the ith round of the protocol a node with address a performs a two party Diffie-Hellman key exchange with the node whose address is $a \oplus 2^{i-1}$. So, in the ith round there will be 2^{i-1} pairs of groups, each group consisting of 2^{d-i} nodes. By the time the dth is completed, a contributory session key will have be created. Next, we will present graphically the 2-d and 3-d cases.

In the 2-d case ($d = 2 \rightarrow 2^d = 2^2 = 4$), there are four entities $\{A, B, C, \text{and } D\}$ aiming to establish a common session key. Let us assume that the address that were assigned to them are $\{00, 01, 11, 10\}$, respectively. Each entity contributes in order the common session key, ($K_{\text{session}} = K_{ABCD}$) can be created, so let us also assume that the contribution of each entity is (S_A, S_B, S_C, S_D). During the first round, two pairs will be created, pair_1 consisting of entities A, B and pair_2 consisting of entities C, D. The two pairs will be internally and in parallel perform a two party Diffie-Hellman yielding a pair of common keys (K_{AB} and K_{CD}) as shown in Figure 2.

During the second round, A will perform a two-party Diffie-Hellman with the node C while node B a two-party Diffie-Hellman with D. Each node will use the common key computed during the previous round, (round 1), in order to create, during the current round, (round 2), the resulting common session key. So, by the end of the second round, all nodes will be sharing the same contributory key (S_{ABCD}). This is presented graphically in Figure 3.

In [7], the authors incorporate the password-based authentication into the cube protocol. This is achieved by using the four-move two-party password authenticated

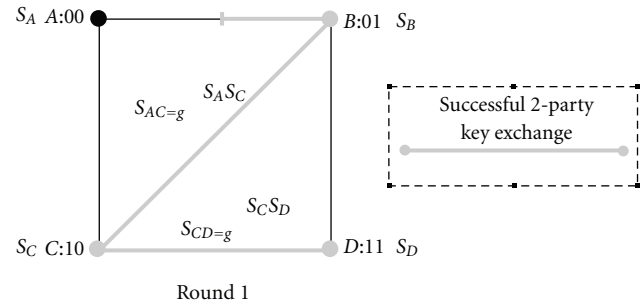

Figure 2: Asokan's 2-d cube round 1.

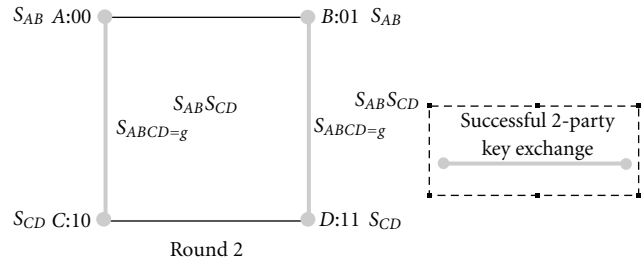

Figure 3: Asokan's 2-d cube round 2.

Diffie-Hellman protocol for pairwise exchanges in each round of the d-cube protocol.

The method is also applicable in the case where the number of players is not a power of 2. The solution for this case is given thought the use of the 2^d octopus proposed by Becker and Wille in [15]. This protocol manages to optimize the number of rounds performed. More precisely if the number of nodes n follows that $2^d < n < 2^{d+1}$, then the first 2^d nodes act as the central controllers and the remaining ones $(n - 2^d)$ are distributed among them as their wards. The controllers execute a two-party Diffie-Hellman with their ward, and then they are engaged in a d round cube protocol using information gathered from the previous stage. Finally, the derived key is distributed to the wards. Another important aspect that [7] introduced is the way that a node should behave when a two-party authentication procedure has failed. They propose an algorithm according which a node can select another potential partner until a nonfaulty one is found. For a single node N in a random round k, there are at most 2^{k-1} potential nodes and at most 2^{k-1} potential subrounds. Two basic requirements are set for node N.

(i) N must not match two nodes to the same partner in a given subround.

(ii) N must not select the same partner twice.

The work in [7] selects the closest partners before the more distant ones, in terms of Euclidean distance between the two corresponding address. The protocol depends on the current round performed, however each round can be consisted of several subrounds.A subround is executed when

a two party key exchange with the appropriate partner node cannot be established. The operation of a player during a given subround is divided in

(i) computation and transmission of all outgoing messages,

(ii) reading of all waiting messages and state transition accordingly.

The proposed algorithm is best illustrated through a simple example which is depicted in Figures 4 and 5.

Every node has a three bit address $\{x, x, x\}$ and a three bit mask, and it is labeled from A to H. Its key contribution is represented by the corresponding lowercase letter.

Labels next to the arrows indicate the nodes that have already contributed, directly or indirectly, to the key. Suppose that player G (with address 110) is unsuitable (unavailable or does not know the password). In round 1, player H (111) will initiate the procedure of selecting as a partner the node whose address is 110 and mask 000.

The exchange attempt with G fails and the mask is already \$000\$. So, H does nothing in this round. In round 2, E (\$100\$) will start with \$110\$ as candidate address and 001 as mask. The first recursive call will try \$110\$ as candidate address and \$000\$ as mask and will fail. The second recursive call will try \$111\$ as candidate address and \$000\$ as mask and will succeed. Similarly, in round 3 and Figure 4, node C (\$010\$) starts partner finding with \$110\$ as candidate. The work in [7] also considers the case, where the total number of nodes is not more than 2^d, while the number of the faulty nodes is $m : 2^k \leq m \leq 2k+1$ for some $0 \leq k \leq d$. The 2^{k-1} of them are located in a single k-cube C_1, and the rest of them in a k-cube C_2.

The number of subrounds required in rounds from $k+2$ to d where $k < d-1$ are at most $m+1$ per round. This is because in each of those rounds, there is always one subround with m faulty partners. The same faulty node may select using N each of the m faulty partners in sequence before being able to complete its round exchange, thus resulting $m+1$ rounds. Since there is no other subcube with more faults, $m+1$ is the maximum number of subrounds required.

In round $k+1$, the number of faulty players in C_1, is 2^k-1, resulting that the maximum number of subrounds is 2^k. So the total number of subround for the first $k+1$ rounds is therefore

$$\sum_{j=0}^{k} 2^j = 2^{k+1} - 1. \tag{21}$$

Thus, the total number of communication rounds required to complete the exchange is $2^{k+1} - 1 + (d-k-2)(m-1)$. This case incurs the maximum possible number of subrounds in the worst case during round 1 to $k+1$ round.

6. The Family of Key Agreement Protocols

In this section, we describe a family of key agreement protocols initially employed only in MANETs and the way that can be implemented in a MANET/mesh internetworking system.

In the approach described in Section 5, the only way to obtain a common session key when one or more nodes depart from the established MANET is to start over the algorithm from the very first step. Furthermore, there are no intermediate session keys stored between nodes that are still part of the network, which could be proven to be useful for node-to-node communication, when global session key is no longer valid due to network reform. Such approaches tend to be sufficient in relatively stable networks, where their topology does not change frequently. However, when network topology dynamicity increases, creating new global session keys very often is not the optimum solution.

The following algorithms propose efficient means for creation and use of intermediary session keys at the same time with the creation of the global network key, which can be used both for subgroup communications and as intermediate step for key refreshment of the global session key, without the obligation to restart the group key agreement.

6.1. The Body-Centered Cubic (BCC) Algorithm. The body-centered cubic algorithm [16] is a cryptographic key agreement algorithm that initiates from a tree-arrangement of 3-d cubes; it is based on the aggressive 3-d cube algorithm and employs the *body-centered cubic* (BCC) structure for the dynamic case. For simplicity purposes, in the rest of the paper, each bond in 2-d or 3-d space corresponds to a two-party password-based elliptic curve Diffie-Helman key exchange, as described in Section 5.2.3.

6.1.1. Initial Node Arrangement. The proposed system is based on the 3-d aggressive d-cube algorithm [17]. The initial key agreement procedure depends on the number of ad hoc nodes that wish to establish a MANET. We denote the number of nodes as n. In contrast to [17], in the proposed system, there is no need for d-dimension hyperspaces. The maximum order is the 3-d space. Nodes of the network are always arranged in the 3-d space, except the case that $n \leq 4$ where we can use the 2-d plane. Therefore, when we have a large number of nodes, they must be divided and arranged in 3-d cubes that each contains eight nodes. Each cube selects a leading node that will act as an intermediary between the corresponding cube nodes and the rest of the ad hoc network. The leading nodes constitute a new group; however, they follow the same rules for initial arrangement, that is, they are arranged in a new 3-d cube. In the case where the number of leading nodes is greater than eight (i.e., the number of all ad hoc nodes is greater than 64), they also need to elect leading nodes in their group that will act as their representatives to the ad hoc network. In such a case, the leading nodes elect higher level leaders in a tree model according to [18]. We consider the latter case as an extreme case since from a practical point of view typical ad hoc networks do not exceed 64 nodes. Figure 6 shows an initial arrangement of a 32-node network. Nodes are arranged in four independent cubes and each cube elects a leader (dashed annotation). Node arrangement and addressing can be performed in any way, as far as every simple-cube node has wireless connection with the rest of the seven nodes of the corresponding cube.

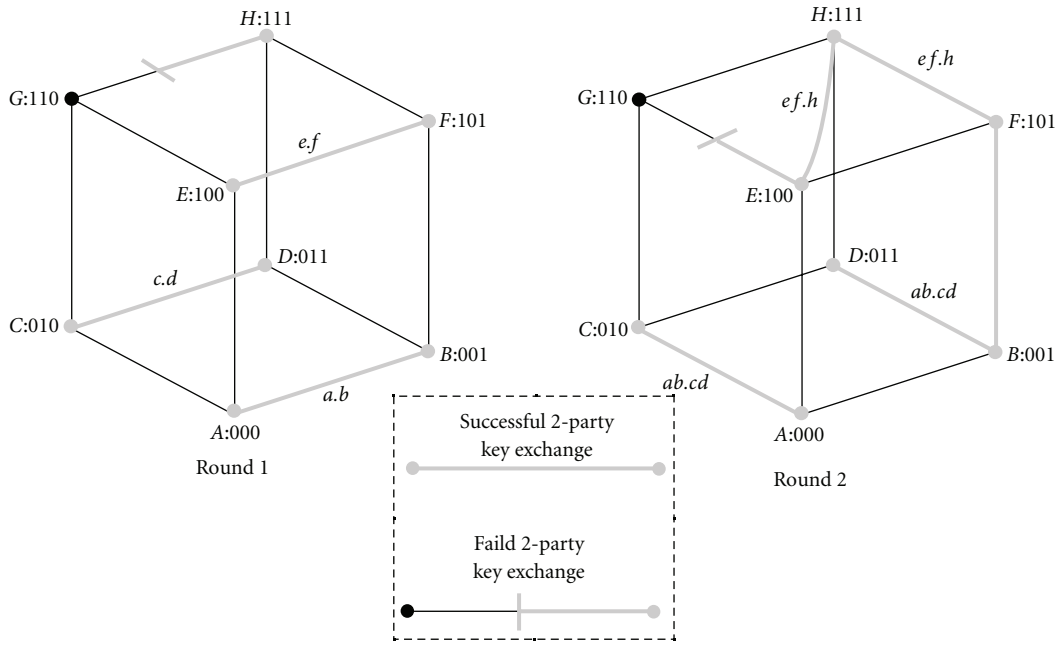

Figure 4: Asokan's 3-d cube round 1, 2.

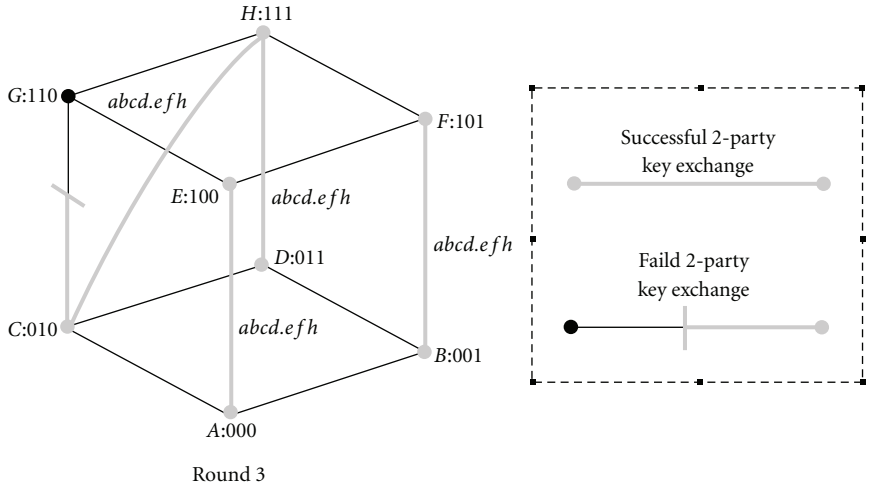

Figure 5: Asokan's 3-d cube round 3.

This requirement must be also fulfilled by the leading nodes among themselves; therefore, it is an important criterion for the selection of a leading node within a simple cube.

6.1.2. Initial (Static) Key Agreement. Next, after the initial 3-d arrangement, BCC creates a common network key. In the proposed system this is done in two steps.

During the first step, the leading nodes perform a 3-d aggressive cube algorithm and they create a global session key. In the second step, every group performs a 3-d aggressive d-cube and establishes a simple-cube session key. During the simple-cube key generation, the leading nodes transmit the global session key that they have already established in step 1 to the remaining seven nodes of the group. After the second step, every node has a contributory simple-cube session key K_{cube} for the cube that is part of, and the global session key of the entire network K_{global}.

In the first step, nodes (000) of cube a, (010) of cube b, (100) of cube c, and (110) of cube d are elected as leading nodes of the corresponding cubes. Since they are four, they perform a 2-d aggressive algorithm, and they establish a global session key K_{global}. If there are than four and equal or less than eight 3-d cubes, their leaders should perform a 3-d aggressive cube algorithm. In this case, the leading nodes can use the first two digits of their addressees as a 2-d address for the 2-d aggressive algorithm, that is, (00), (01), (10), and (11). If other nodes are elected as cube leaders due to communication constraints, they should be addressed

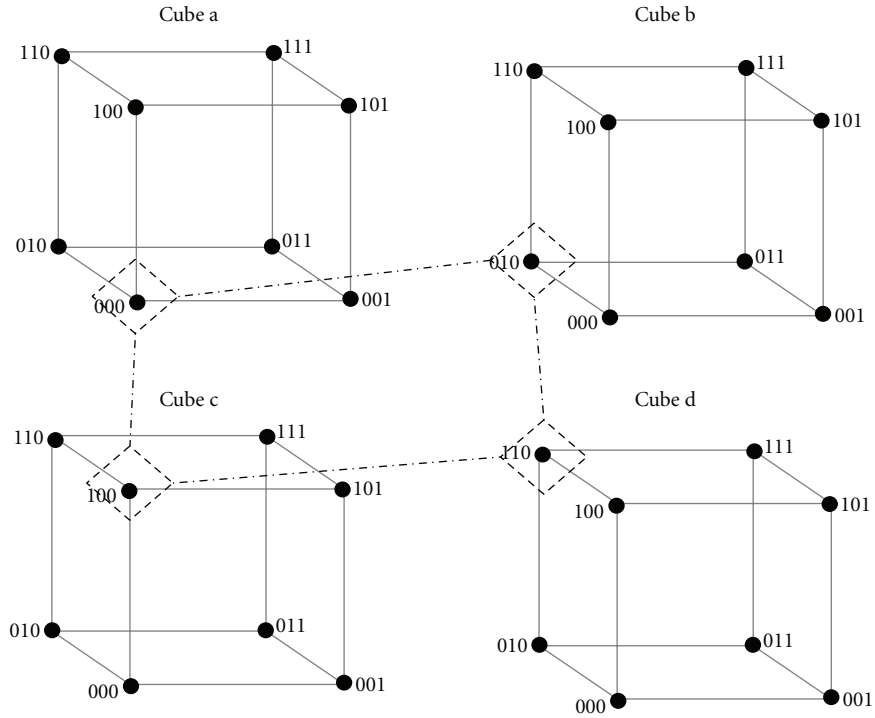

Figure 6: A BCC 4-cube example.

in a separate way than the one employed in their 3-d cube (second addressing is required).

Once the global session key of a group K_{global} is established the cubes perform a 3-d aggressive d-cube, and they establish the simple-cube session key K_{cube}. The final step is that each cube leader broadcasts the global key encrypted with the simple-cube key to the rest of the cube members. At the end of the protocol, every node has a simple-cube session key K_{cube} for secure communications among nodes of simple-cubes, and a global session key K_{global}, for the entire group (mesh or MANET).

6.1.3. BCC for the Dynamic Case. Above, we described the initial arrangement-addressing of nodes and the generation of a global and of simple-cube session keys. These keys are static, since if there is a need to add new nodes to the network, the key generation procedure must be repeated. Here, we describe an efficient method for dynamic key generation every time new nodes arrive to or depart from our network. The proposed dynamic algorithm is based on the body-centered cubic structure, and we call it BCC algorithm.

The body-centered cubic (bcc) structure is a cube with an additional node in the center. Figure 7(a) shows a typical cube while Figure 7(b) depicts a body-centered cube. If we consider the grid case, the bcc structure is a set of bcc cubes. The BCC algorithm for dynamic changing topologies is presented through two cases: addition of new nodes to an established network and extraction of network nodes.

Case 1 (Adding nodes to an established network). The BCC algorithm operates in the following way: assume that a group has been established as previously described. Assume that one simple cube of this network is depicted in Figure 7(a). At some point of time, seven new nodes arrive and request to join the network. If the number of the new arriving nodes m is a multiple of 8, that is, $m \bmod (8) = 0$, then in groups of 8, they perform aggressive cube algorithms and each group elects a leader that will contact leaders of new groups and leaders from the established network in order to create a new global session key. If $m \bmod (8) \neq 0$, then we will have k new groups of 8 nodes where k is the integer part of $m/8$ and l the number of the remaining nodes where $0 < l = m \bmod (8) < 8$ while the k groups of 8 nodes will perform new aggressive cube algorithms, the remaining node will attach to an existing cube of the network in the following way.

The first four new nodes are assigned addresses that correspond to the center of the existing cube the centers of the right, upper, and front cubes as shown in Figure 7(c) while the last three are assigned addresses that correspond to the centers of the left, back, and down cubes as shown in Figure 7(e). Keep in mind that the six neighboring cubes do not exist as network cubes; they are used as geometrical objects for demonstration purposes of the BCC algorithm. The first four new nodes (the body-centered cubic node and three central nodes of neighbor cubes) create a new cube with four nodes of the preexisting network cube as shown in Figure 7(d), and they perform a new aggressive cube algorithm. The latter 3 new nodes together with the body-centered cubic node (the node assigned to center of

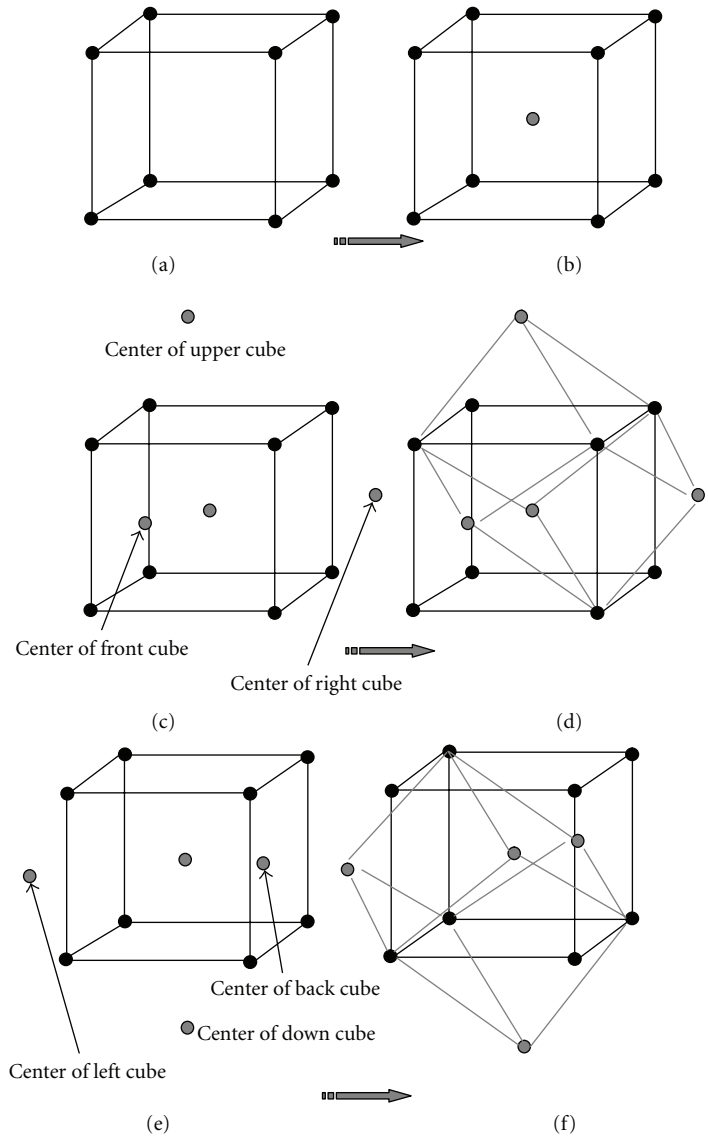

Figure 7: Body-centered cubic structure.

the existing cube of the network that has already participated in the previous new aggressive cube), and the remaining four nodes of the preexisting cube of the network perform a second new aggressive cube algorithm as shown in Figures 7(e) and 7(f).

The preexisting network nodes are colored black while the new arriving nodes are in gray color. Figure 8 demonstrates the BCC algorithm and the process that initiates with an existing network cube and concludes to two new cubes that both have the central node (body-centered cubic node) of the initial cube as common node.

Case 2 (Extracting nodes from an established network). The process for extraction of nodes from an established network (MANET or mesh) is similar to the addition of new nodes to such a network. The remaining nodes of a simple cube find close cubes and perform the BCC algorithm. This process changes only the global session key of the network, since according to BCC algorithm cube leaders have to establish a new global key; however, the simple cube session keys of the rest of the MANET cubes remain unchanged. Figure 9 demonstrates the simple case where one node leaves an established cube (node 100 of the left cube of Figure 9(a)). The remaining seven nodes take place according to the BCC algorithm at the centers of the geometrically neighboring cubes of the right cube of Figure 9(a), as shown in Figure 9(b). The BCC algorithm concludes with the generation of two new cubes as shown in Figure 9(c).

6.2. The Face-Centered Cubic (FCC) Algorithm. The face-centered cubic algorithm is shown in Figure 10. The first 8 nodes (or less) are arranged in a 3-d cube as shown in the left side of Figure 3. They perform an aggressive 3-d cube algorithm and obtain a common session key. The first

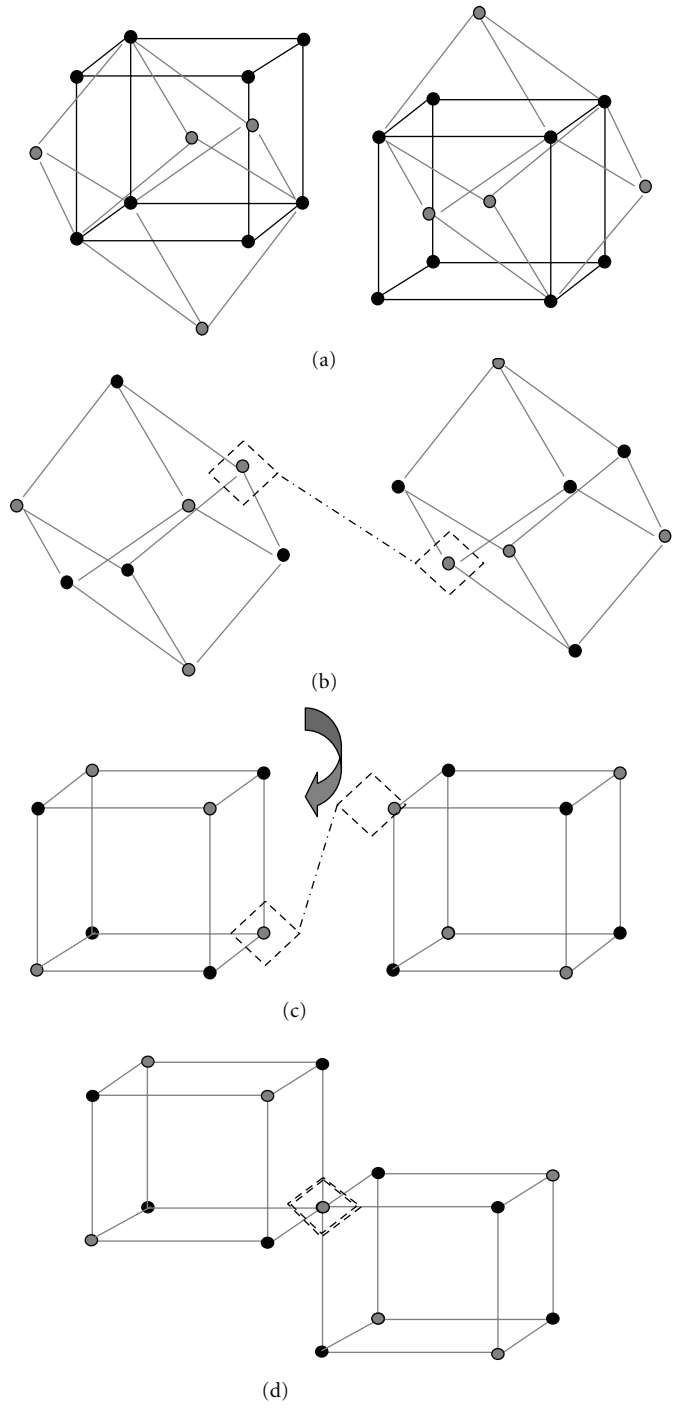

Figure 8: Body-centered cubic algorithm.

6 nodes that will arrive in a later phase will be arranged in the centers of the six faces of the cube as shown in the central picture of Figure 3.

The 6 new nodes together, with nodes (010) and (101) that contributed to the initial cube, create a new cube and perform a new (second) aggressive 3-d cube algorithm. This way the inner cube creates a second common session key. After the setup of the second session, key nodes (010) and (101) hold both session keys corresponding to both cubes. This privilege makes nodes (010) and (101) leading nodes for the established network since any communication between black and grey nodes should pass through them. If we wish to avoid this hierarchy in our network, during the set up of the common session key within the inner cube, nodes (010) and (101) propagate the common session key of the initial (black) cube to the new nodes. This way the first session key can be

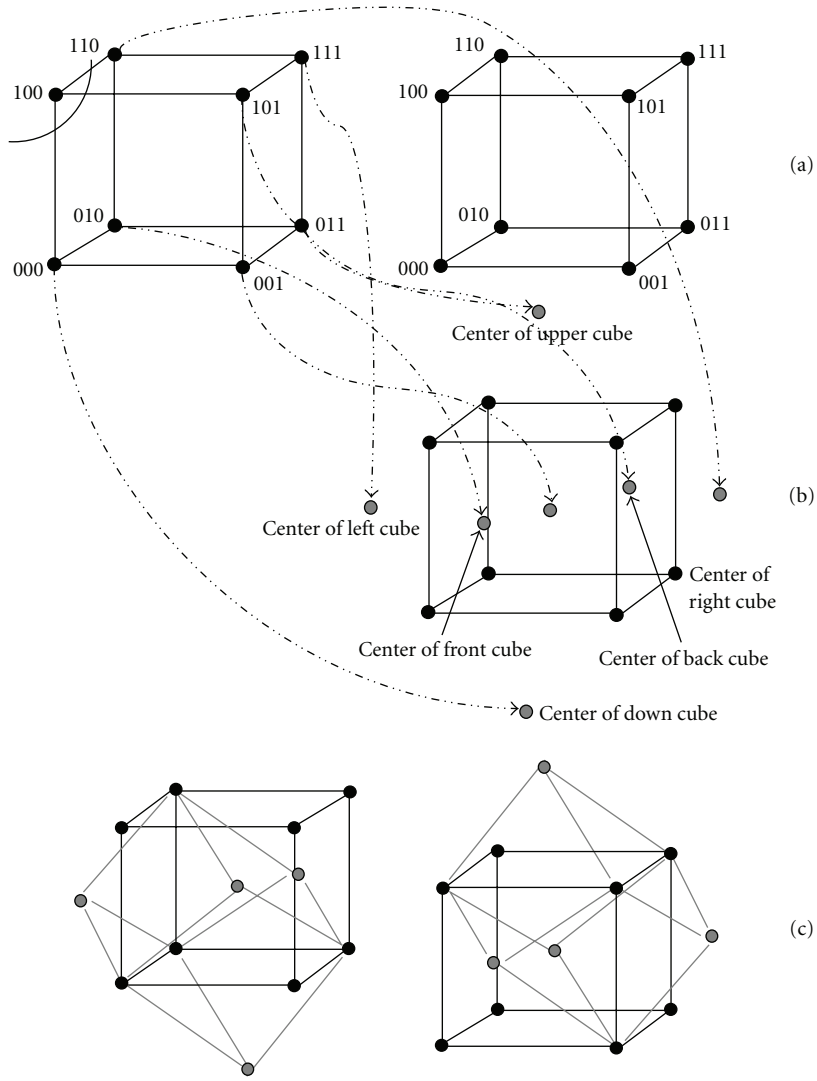

Figure 9: Extracting nodes from an established MANET.

used by all nodes to communicate securely with each other, while the second can be used for the secure communication of the internal (grey) cube.

The addressing of the new nodes is shown in Figure 11. We observe that the 2 old (black) nodes keep the same address with the one they had during the setup of the first session key. For the communication between nodes belonging to different cubes there is a separate metric (cube number) declaring the cube that the node is belonging to. In this example, black nodes are identified as cube 1 nodes, grey nodes are identified as cube 2 nodes, and nodes (010) and (101) have both identifiers since they belong to both cubes.

In the hierarchical model, where every cube has its independent session key, key refreshment due to departing nodes is easy. As soon as a node is leaving the network, the rest nodes of the common cube, perform a new aggressive 3-d cube algorithm and create a new session key. In case the leaving node is belonging to two consecutive cubes, a new aggressive 3-d cube algorithm is performed automatically to both cubes.

In the case where the previous session keys belonging to previous cubes are forwardly distributed to the next cubes, the key renewal should be performed to all previous cubes. This appears not to be a desired feature, since if there is a departure in the last cube all previous stages/cubes will be affected. However, this can be also avoided if the set up is a combination of the two solutions. Periodically the key forwarding method is interrupted by the hierarchical solution. This way, we create isolated groups of concatenated cubes and any necessary key refreshment is bounded within these groups.

6.3. The Tetrahedral Approach. In the tetrahedral approach [19], in contrast to [16, 20] all intermediate two-party ECDH keys are stored by each node. This way, when a global session key has been created, every node in the d-dimensional cube maintains also a list of all the two-party ECDH keys that has

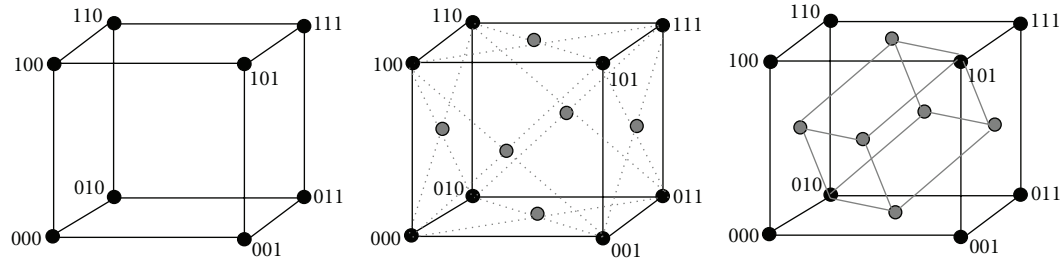

Figure 10: The face centered cubic algorithm.

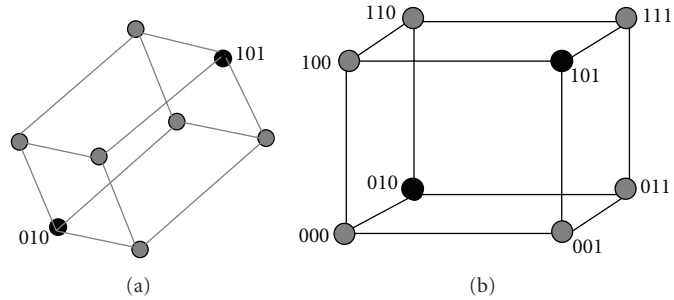

Figure 11: FCC algorithm: addressing of the new nodes.

created with every of her closest neighbors during the global key generation.

In the BCC and FCC approaches as described above, as soon as the initial phase of the proposed algorithm is completed, each node posses the global session key, and the two-party keys with her closest neighbors. At that point, if a node leaves the network, the global session key should be refreshed and in the meantime, secure communications would be only available between couples of closest neighbors that have already established two-party keys during the creation of the latest session key, which is no longer valid. This way, communication between distant neighbors, is only available through multihopping between nodes that, in couples, maintain valid two-party keys. This would add another requirement for routing metric information maintenance by all nodes, in order to serve each other to find the right secure path in the network. Solution to this direction is provided by the proposed integration of tetrahedral group key agreement in the existing cubes. The proposed structure covers both cases (d-cube and aggressive d-cube) and takes place right after the creation of the global session key.

Below, the procedure is explained in detail.

As soon as the global session key has been created (round 3 in the 3-d example) all nodes establish two-party ECDH with their second level closer neighbors. There nodes are actually the ones based on the diagonal of each cubic surface. The algorithm is better demonstrated in Figure 12. Figure 12(a) describes the cube created after the global key agreement, while Figures 12(b) and 12(c) demonstrate the two-party ECDH key exchanges between the second order neighbors. All these additional two-party ECDH key exchanges form the two internal tetrahedrons inside the cube as shown in Figures 12(b) and 12(c). Figure 13 depicts the two internal tetrahedrons isolated by the cube. The process for the establishment of these tetrahedrons, in terms of two-party ECDH key agreements, is depicted in detail in Figure 14. We can observe that two-party ECDH key agreements take place on nonconnected segments. Although that after the second round the tetrahedrons have formed their global session keys, the algorithm has another final step, by covering all available segments.

This is due to two reasons: every node has a two party key with all nodes of the cube except the most distant node. For example, a has two party keys with every node of the cube except node h. Besides the group session keys among every 4 nodes forming a square edge of the cube, the four triangles of each internal tetrahedron shares a common session key, since the ECDH key exchange this time is the party D-H key exchange instead of two-party in all other cases.

Let us provide an example to demonstrate the attributes of the proposed algorithm. The reference node for this example will be node a. During the initial 3-d or aggressive 3-d algorithm, node a creates three two-party keys with nodes b, c, and e and the global session key of the cube. During the tetrahedral algorithm, node a creates three two-party keys with nodes d, g and f. This way, node a maintains two-party keys with all nodes of the cube except h.

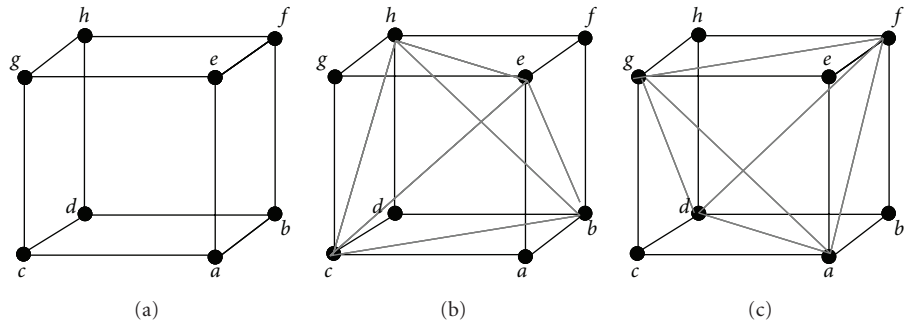

Figure 12: The proposed tetrahedral algorithm structure.

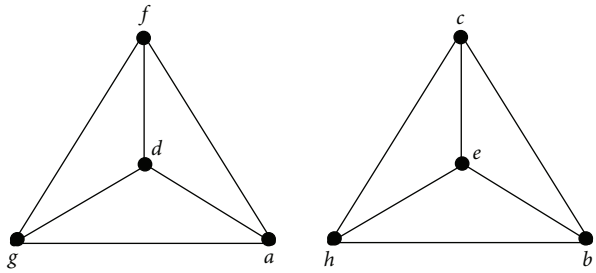

Figure 13: The internal tetrahedrons.

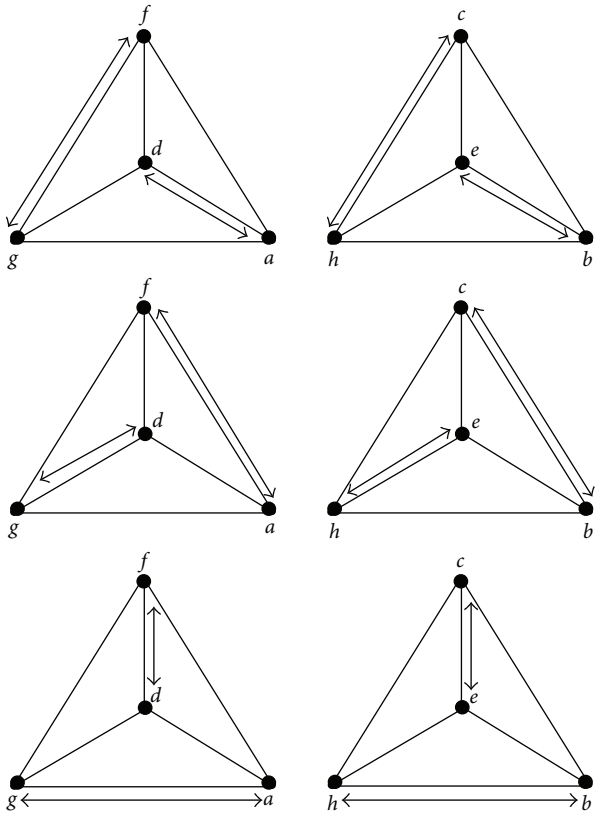

Figure 14: The two-party ECDH key agreement sequence in the internal tetrahedrons.

If any of its first and second order closest neighbor leaves the network, during the global key renewal node a will be still able to communicate with all expect one of the remaining nodes. However, this distant node (in the example node h) belongs to the other tetrahedron and has keys for communication with the remaining nodes.

In another case, if a decides to leave the network, its distant nodes, belonging to the left tetrahedron, do have keys for secure communications (a session key for the hole tetrahedron plus the two party keys among any pair of them), while the remaining three nodes b, c, and h of the right tetrahedron, besides the two-party keys, they have a three-party key established among them during the last stage of the tetrahedral key agreement algorithm. Therefore, when a node leaves the network, the remaining nodes have all the two-party session keys, a four-party session key of the tetrahedron that did not change formation, and a three-party session key of the triangle composed of the three remaining nodes of the tetrahedron that the leaving node was part of.

6.4. Analysis of the Algorithms. All proposed algorithms integrate elliptic curve cryptography together with the password-based DH key exchange protocols and extend this approach to the multiparty case where new network nodes arrive or existing network nodes depart from the network.

The static multiparty case is considered and after careful examination the aggressive d-cube algorithm is selected in the 3-d case. The algorithms are extended versions of the d-cube algorithm proposed in [7], with additional features that enhance the resistance against the dictionary attacks [5]. The basic idea behind the algorithms' design is to isolate the faulty nodes in the earliest possible stage. We managed to reduce the interaction with the faulty nodes and therefore minimize the exposure to dictionary attacks and other types of attacks. However, the aggressive behavior of the algorithm may lead to isolation of even legitimate members due to reasons such as the loss of commutation signal or the typo errors of the initial password during the authentication procedure. To overcome these kind of problems and to provide a more dynamic and robust solution, we propose the body-centered cubic (BCC) as well as the face-centered cubic (FCC) and the tetrahedral algorithm.

The BCC and the FCC algorithms can be considered as a series of different aggressive cube algorithms, each

one performs for a different set of network nodes. In terms of two-party ECDH procedures, the BCC and FCC complexity is not higher than any of the single aggressive cube algorithms. This is because within a single two-party ECDH procedure of each algorithm, three ECDH procedures run simultaneously, each one corresponding to a different round of the aggressive cube algorithm.

The tetrahedral algorithm has significant differences compared with the BCC or the FCC algorithms. In this algorithm, all intermediate two-party keys are stored locally by the nodes and may be used during the network reformation process. Moreover, three-party keys are stored, providing the ability for secure subgroup communications among nodes that share the corresponding keys.

Depending on the mobility pattern and the reformation speed of the network, the aggressive d-cube algorithm is more efficient for almost static networks, the FCC approach would be more convenient for small-medium network reformation, BCC for medium-high network reformation and the tetrahedral algorithm would better facilitate high dynamicity in network reformation.

7. Conclusion

Our research was motivated by the need to establish fast, reliable, efficient, and secure group communications without relying on preexisting infrastructures. The actual operational environment and the varying nature of the established networks impose further key issues (e.g., the ability to add or subtract nodes depending on operational and security considerations) that need to be taken into account.

We have reviewed existing proposals around two-party or multiparty authentication and introduced new key establishment methods. Our proposal overcomes some of the main issues (such as rapid deployment, accuracy, and dynamic and robust behaviour) of existing solutions and operational environments. The proposed solutions introduce the use of elliptic curve cryptography in such a scenario. ECC computations require less storage, less power, less memory, and less bandwidth than other systems. This allows implementation of cryptography in constrained platforms such as wireless devices, handheld computers, smart cards, and thin-clients. For a given security level, elliptic curve cryptography raises computational speed and this is important in ad hoc networks, where the majority of the clients have limited resources.

The proposed algorithms meet all security requirements according the initial specification and provide differentiated solutions depending on the network reformation dynamicity. They are designed to support in a uniform approach both MANETs and wireless networks, taking into account a broad range of application requirements including secure internetworking and dynamicity of network topologies.

The proposed algorithms leave several open issues for future work. Formal analysis is necessary. The incorporation of several new password-based key agreement protocols, which do not require the use of asymmetric encryption, is a challenging consideration. Studying in detail the dynamic case, where the network topology is rapidly changing, would be very interesting, especially for indentifying the thresholds of network reformation that should be employed for algorithm switching.

Acknowledgment

This work was supported in part by the European Commission in the 7th Framework Programme through project EU-MESH (Enhanced, Ubiquitous, and Dependable Broadband Access using MESH Networks), ICT-215320, http://www.eu-mesh.eu/.

References

[1] I. F. Akyildiz, X. Wang, and W. Wang, "Wireless mesh networks: a survey," *Computer Networks*, vol. 47, no. 4, pp. 445–487, 2005.

[2] R. Bruno, M. Conti, and E. Gregori, "Mesh networks: commodity multihop ad hoc networks," *IEEE Communications Magazine*, vol. 43, no. 3, pp. 123–131, 2005.

[3] C. Verikoukis, L. Alonso, and T. Giamalis, "Cross-layer optimization for wireless systems: a European research key challenge," *IEEE Communications Magazine*, vol. 43, no. 7, pp. 1–3, 2005.

[4] P.-F. Bonnefoi, D. Sauveron, and J. H. Park, "MANETS: an exclusive choice between use and security?" *Computing and Informatics*, vol. 27, no. 5, pp. 799–821, 2008.

[5] A. Narayanan and V. Shmatikov, "Fast dictionary attacks on passwords using time-space tradeoff," in *Proceedings of the 12th ACM Conference on Computer and Communications Security (CCS '05)*, pp. 364–372, Alexandria, Va, USA, November 2005.

[6] S. M. Bellovin and M. Merritt, "Encrypted key exchange: password-based protocols secure against dictionary attacks," in *Proceedings IEEE Computer Society Symposium on Research in Security and Privacy*, pp. 72–84, Oakland, Calif, USA, May 1992.

[7] N. Asokan and P. Ginzboorg, "Key agreement in ad hoc networks," *Computer Communications*, vol. 23, no. 17, pp. 1627–1637, 2000.

[8] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[9] D. Johnson and S. Vanstone, "The elliptic curve digital signature algorithm (ECDSA)," *International Journal on Information Security*, vol. 1, no. 1, pp. 36–63, 2001.

[10] F. Cucker and S. Smale, "Complexity estimates depending on condition and round-off error," *Journal of the ACM*, vol. 46, no. 1, pp. 113–184, 1999.

[11] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of Computation*, vol. 4, no. 8, pp. 203–209, 1987.

[12] A. J. Menezes, T. Okamoto, and S. A. Vanstone, "Reducing elliptic curve logarithms to logarithms in a finite field," *IEEE Transactions on Information Theory*, vol. 39, no. 5, pp. 1639–1646, 1993.

[13] A. Menezes, E. Teske, and A. Weng, "Weak fields for ECC," in *Topics in Cryptology (CT-RSA '04)*, T. Okamoto, Ed., vol. 2964 of *Lecture Notes in Computer Science*, pp. 366–386, Springer, 2004.

[14] A. Kalele and V. R. Sule, "Weak keys of pairing based Diffie-Hellman schemes on elliptic curves," Cryptology ePrint Archive 2005/30, 2005.

[15] C. Becker and U. Wille, "Communication complexity of group key distribution," in *Proceedings of the 5th ACM Conference on Computer and Communications Security*, San Francisco, Calif, USA, November 1998.

[16] I. Askoxylakis, D. Sauveron, K. Markantonakis, T. Tryfonas, and A. Traganitis, "A body-centered cubic method for key agreement in dynamic mobile ad hoc networks," in *Proceedings of the 2nd International Conference on Emerging Security Information, Systems and Technologies*, pp. 193–202, Cap Esterel, France, August 2008.

[17] I. G. Askoxylakis, D. D. Kastanis, and A. P. Traganitis, "Elliptic curve and password based dynamic key agreement in wireless ad-hoc networks," in *Proceedings of the 3rd IASTED International Conference on Communication, Network, and Information Security (CNIS '06)*, pp. 50–60, Cambridge, Mass, USA, October 2006.

[18] L. Liao and M. Manulis, "Tree-based group key agreement framework for mobile ad-hoc networks," *Future Generation Computer Systems*, vol. 23, no. 6, pp. 787–803, 2007.

[19] I. Askoxylakis, T. Tryfonas, J. May, and A. Traganitis, "A dynamic key agreement mechanism for mission critical mobile ad hoc networking," in *Proceedings of the 2nd International Conference on Mobile Lightweight Wireless Systems*, Barcelona, Spain, May 2010.

[20] I. Askoxylakis, K. Markantonakis, T. Tryfonas, J. May, and A. Traganitis, "A face centered cubic key agreement mechanism for mobile ad hoc networks," in *Proceedings of the 1st International Conference on Mobile Lightweight Wireless Systems (MOBILIGHT '09)*, pp. 103–113, Athens, Greece, May 2009.