A Dynamic Key Agreement Mechanism for Mission Critical Mobile Ad Hoc Networking*

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

[1] Foundation for Reserach and Technology-Hellas – Institute of Computer Science,
N. Plastira 100, 70013 Heraklion, Greece
{asko,tragani}@ics.forth.gr
[2] University of Bristol, Faculty of Engineering,
Queen's Building
University Walk,Clifton, Bristol, BS8 1TR
{t.tryfonas,j.may}@bristol.ac.uk

Abstract. Mobile ad hoc networks are expected to play an important role in demanding communications such as military and emergency response. In Mobile ad hoc networking each node relies on adjacent nodes in order to achieve and maintain connectivity and functionality. While offering many advantages, such as flexibility, easy of deployment and low cost, mobile ad hoc networking faces important security threats that could be proven vital in future telecommunication applications. This paper introduces a key dynamic agreement method based on a weak to strong authentication mechanism associated with a multiparty contributory key establishment method. It is designed for dynamic changing topologies, it employs elliptic curve cryptography to best serve thin clients with energy constrains, and reduces significantly key re-establishment due to network formation changes.

Keywords: MANET security, password authentication, elliptic curve cryptography, key agreement.

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 and at the same time the need for establishing a network is crucial The objective of the rescue workers is to set up quickly, efficiently and easily a wireless network among themselves 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 won't be able to disrupt or intercept the rescue efforts.

* 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*

P. Chatzimisios et al. (Eds.): MobiLight 2010, LNICST 45, pp. 216–228, 2010.

Security is a primary concern for providing protected communications to mobile nodes that operate in such hostile environments where there is no readily available infrastructure and where networks of varying 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 (MANETs) 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.

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 [1].

In MANETs, such as in any other network trust cannot be provided among the nodes of the network without the existence of pre-defined prior known information to all nodes. This special kind of information is necessary in order to build trust between all participating nodes. A MANET is established among the existing nodes, if from preexisting, commonly known information, 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* (*we define all the legitimate members of the established wireless network as group, e.g., soldiers of a military unit*) 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.

The rest of the paper is organized as follows: In section 2 we describe the adversary model and in section 3 we present the security requirements. In section 4, we present a review of the previous 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 5 we describe the state of the art in multiparty key agreement protocols and particularly the d-cube and the aggressive d-cube algorithms and examine their properties and we describe our proposal that could be considered as an extension of both algorithms, designed for the dynamic changing topologies. Finally, in Section 6, we provide our concluding remarks along with suggestions for future work.

2 Adversary Model

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

Attackers' classes. Taking into account the system model of a MANET we can identify tho types of attackers:

- External attackers: These are attackers that have no legitimate access to the MANET but they have appropriate equipment to use the wireless medium and interfere with the operation of the network protocols.
- Compromised nodes: These are legitimate node devices that have legitimate access to the MANET 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

Objectives of attacks. We identify the following main objectives of attacks:

- Unauthorized access to the services provided by the MANET: Primarily, this objective is relevant for both classes of attackers.
- Unauthorized access to node data and meta-data: Here node data means the content of the messages exchanged in a service session, whereas meta-data refers to information on the nodes location and service usage profile (e.g., which applications are used and how often). Thus, the first objective is related to violating the confidentiality, and the second is related to violating the privacy of the node. Primarily, this objective is relevant for external adversaries.
- Denial-of-Service (DoS): This objective is related to degrading the QoS offered by the network (including the complete disruption of services). Primarily, this is relevant for external adversaries.

Attack mechanisms. There are a multitude 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:

- Attacks on wireless communications (including eavesdropping, jamming, replay, and injection of messages, and traffic analysis);
- 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.

3 Security Requirements

It is broadly known that security mechanisms cannot create trust [2]. The members of a team that wish to establish a MANET 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 in order to secure the ad hoc network.

A password authentication mechanism seems to be a rational approach that can deliver a proper solution without adding new requirements like the use of dedicated hardware (i.e smart cards). In a password based authentication scheme the use of a sufficiently large and randomly generated data string that can be used as a password would be an obvious approach. This way all nodes could agree on a password and, by using a trivial authentication protocol, achieve mutual authentication.

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 [3]. 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:

Security architecture designed for thin clients. A MANET is typically composed of mobile devices with limited processing power and energy consumption. The cryptographic algorithms used for authentication and key agreement should have minimal impact in terms of computational overhead.

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 MANET.

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 MANET 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.

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

4 Related Work

4.1 Key Exchange and Elliptic Curve Cryptography

Common cryptographic protocols based on keys chosen by the users are weak to dictionary attacks. Bellowin and Merrit [4] proposed a protocol called *encrypted key exchange (EKE)* where a strong shared key is derived from a weak one. However, this protocol has a disadvantage. The creation of the common session key takes place with unilateral prospective, that is, only by the entity that first initiated the whole procedure. Thus the key agreement scheme is not contributory.

Diffie–Hellman is the first public key distribution protocol that opened new directions in cryptography [5]. In this important protocol for key distribution, 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.

An essential property for the majority of cryptographic applications is the need for fast and precise arithmetic. Calculations over the set of real numbers are slow and inaccurate due to round-off error [6]. Finite arithmetic groups, such as

$$F_p, F_{2^m}.$$

which have a finite number of points, is used in practice. All practical public-key systems today exploit the properties of arithmetic using large finite groups. Additionally, elliptic curves can provide versions of public-key methods that, in some cases, are faster and use smaller keys, while providing an equivalent level of security. Consequently, the use of ECC can result in faster computations, lower power consumption, as well as memory and bandwidth savings. This is very useful for mobile devices, like the ones used in ad hoc networks, which face limitation in terms of CPU, power, and network connectivity.

An elliptic curve [7] consists of elements (x, y) satisfying the equation:

$$y^2 = x^3 + \alpha x + \beta (\text{mod} p). \tag{1}$$

for two numbers α, β. If (x, y) satisfies the above equation then $P = (x, y)$ is a point on the elliptic curve.

The elliptic curve discrete logarithm problem (ECDLP) can be stated as follows:

Fix a prime p and an elliptic curve E. Let xP represent the point P added to itself x times. Suppose Q is a multiple of P, so that $Q = xP$ for some x, then the ECDLP is to determine x given P and Q.

The general conclusion of leading cryptographers is that the ECDLP requires fully exponential time to solve. The security of ECC is dependent on the difficulty of solving the ECDLP.

Research community has given considerable attention to the ECDLP. Like the other types of cryptographic problems, no efficient algorithm is known to solve the ECDLP. The ECDLP seems to be particularly harder to solve. Moderate security can be achieved with the ECC using an elliptic curve defined over Z_p where the prime p is several times shorter than 230 decimal digits.

An elliptic curve cryptosystem implemented over a 160-bit field currently offers roughly the same resistance to attack, as would a 1024-bit RSA [8]. However, there have been weak classes of elliptic curves identified such as super singular elliptic curves [9] and some anomalous elliptic curves [10]. Implementations, such as ECDSA [11], merely check for weaknesses and eliminate any possibility of using these "weak" curves [12].

4.2 Elliptic Curve Diffie–Hellman

The original Diffie–Hellman (D-H) 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 as 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 to comply with the appropriate security standards [11].

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.

Entity *A* selects an integer,

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

Entity *B* selects an integer

$$d_B : d_B \in [2, n-2] . \tag{3}$$

A computes

$$Q_A = d_A \times P . \tag{4}$$

The pair Q_A, d_A consists A's public and private key.

B computes

$$Q_B = d_B \times P . \tag{5}$$

The pair Q_B, d_B consists B's public and private key.

A sends Q_A to B,

$$A : Q_A \rightarrow B . \tag{6}$$

B sends Q_B to A,

$$B : Q_B \rightarrow A . \tag{7}$$

A computes

$$K = d_A \times Q_B = d_A \times d_B \times P . \tag{8}$$

B computes

$$K = d_B \times Q_A = d_B \times d_A \times P . \quad (9)$$

Quantity K is now the commonly 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 The Proposed Architecture

The dynamic topology of mobile ad-hoc networks introduces challenging security issues. The continuous flow of incoming and departing nodes is a key issue for designing a key agreement mechanism. Furthermore, when a node publicly claims that it is leaving the network it does not mean that it looses its ability to "hear" the messages exchanged among the remaining nodes, unless action is taken.

In all the approaches described in section 4, the only way to obtain a common session key when one or more modes depart from the established MANET, is to start over each algorithm from the very first step Furthermore, there are no intermediate session keys stored between nodes that are still part of the network, that could be proven to be useful for node-to-node communication, when global session key is no longer valid due to network reform. It is obvious that such approaches tend to be sufficient in relatively stable MANETs, where their topology does not change frequently. However, when network topology dynamicity increases, creating new global session keys very often would not be the best solution.

The proposed architecture proposes an efficient way for creation and use of intermediary session keys at the same time with the creation of the global MANET key, that can be used both for subgroup communications and as intermediate steps for key refreshment of the global session key, without the obligation to restart the group key agreement from scratch.

5.1 D-Cube Initiation

The proposed architecture is based on [16] and [17]. A session starts, based on either case, (d-cube algorithm or aggressive d-cube algorithm) and concludes with a common, contributory created, session key among all nodes of the MANET. It is best illustrated through a simple example which is depicted in figure 3.

Every node have a three bit address {x,x,x} a three bit mask, and is labeled from A to H. Its key contribution is represented by the corresponding lower-case 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. Asokan and Ginzboorg, in [16] also consider the

case, where the total number of nodes is not more than 2^d, while the number of the faulty nodes is, $m: 2^k \le m \le 2^{k+1}$, for some $0 \le k \le d$. 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.

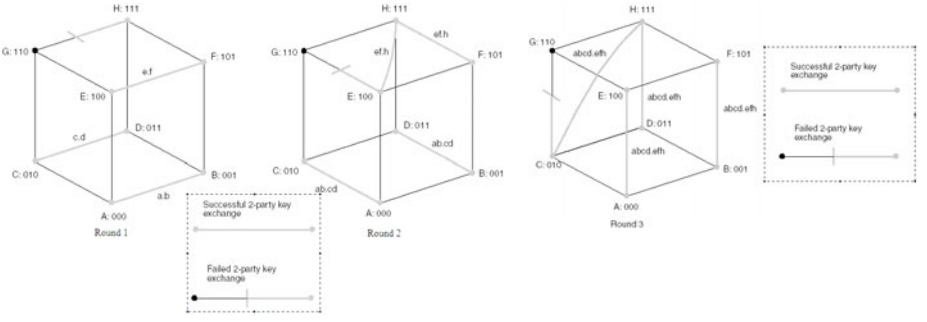

Fig. 1. 3-d cube rounds 1,2 and 3

The number of sub-rounds 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 sub-round 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 sub-cube with more faults, $m+1$ is the maximum number of sub rounds required.

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

$$\sum_{j=0}^{k} 2^j = 2^{k+1} - 1. \tag{3}$$

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 sub-rounds in the worst case during round 1 to $k+1$ round.

Next, we will detailed describe the aggressive 3-d cube example. (See fig.2) In this case we assume that node G is the faulty partner. During the first round the DH key exchange procedure performed between G:110 and H:111 will fail, since node G is a faulty one. However, instead of remaining idle and wait for the next round (as in the previous case), node H starts a DH key exchange with node E:100. Meanwhile Node E has already performed a successful DH key exchange with F:101, during the first half of the first round, so this key exchange will be the second successful one for this round. Node E having being notified by H that G is a faulty node will remain idle until the third round, instead of having attempted unnecessary DH exchanges with G.

In the next round (round 2) H performs a DH with node F and a DH with node C:010. Given that C has performed two successful DH with D:011 and H respectively, he will remain idle in the next round. However C has already performed a successful DH with A:000, during round one.

In total node C has performed three successful DH, with three different nodes, which means that C has completed all the appropriate procedures. Thus it will remain idle for the next round, which is the last round in our case. Summarizing the logic of this procedure we would say that the upper bound of the total successful DH procedures for a node participating in an aggressive d-cube algorithm is equal to d. In the described example d=3. During the third and final round there will be three more successfully accomplished DH key exchanges. One between H and D, one between F and B , and one between A and F.

Through this example it becomes obvious that using the aggressive 3-d cube algorithm, the faulty partner is being isolated. He only participates in one DH key exchange, the one performed in round 1 with node H, and since then he is excluded from all the subsequent DH key exchanges. Consequently, the faulty node loses the ability to have another change, during the generation process of the common session key, to compromise the security of the system.

In the following figures we give a graphical representation of the example.

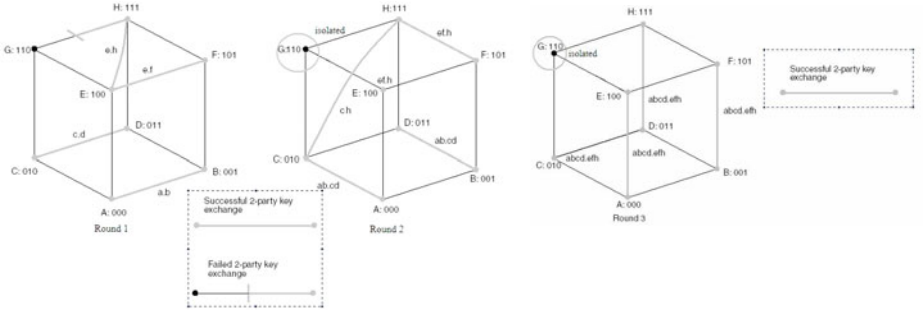

Fig. 2. Aggressive 3-d cube rounds 1, 2 and 3

5.2 Storage of All Intermediate Two-Party Keys

In the proposed algorithm, in contrast to [16] and [17] all intermediate two-party ECDH keys are stores 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 created with every of her closest neighbors during the global key generation.

5.3 The Internal Tetrahedrons Integration

So far, as described in sections 5,1 and 5.2, 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 this point, if a node leaves the network, the global session

key should be refreshed and in the meantime, secure communications are only available between couples of closest neighbors that participated in a two-party manner during the creation of the latest session key, which is no longer valid. This way, communication between distant neighbors, is only available through multi-hopping 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 correct 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.

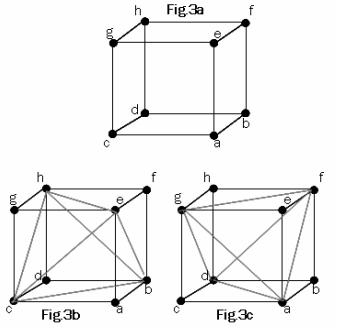

Fig. 3. The proposed tetrahedral algorithm structure

The procedure is the following:

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 3. Figure 3a describes the cube created after the global key agreement, while figures 3b and 3c, 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 3b and 3c. Figure 4, 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 5. We can observe that two-party ECDH key agreements take place on non-connected 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 thee party D-H key exchange instead of two-party in all other cases.

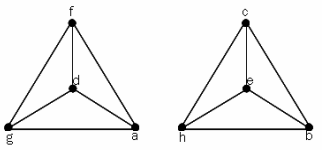

Fig. 4. The internal tetrahedrons

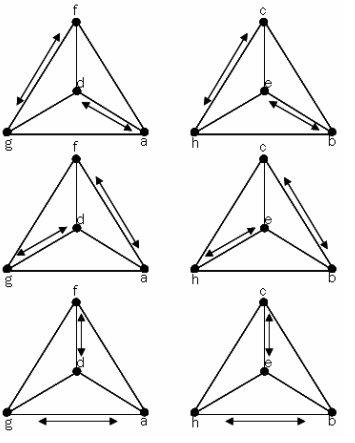

Fig. 5. The two-party ECDH key agreement sequence in the internal tetrahedrons

Let's 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.

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, her 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 Conclusion

Our research was motivated from the requirement of certain groups to establish fast, reliable, efficient and secure MANET's without relying on pre-existing infrastructures. The actual operational environment and the very 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 a new key establishment method. 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 solution introduces 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 protocol meets all security requirements according the initial specification and it is stronger in terms of security. Finally, we have proposed secure and resilient architecture for dynamic MANETs, where the composition of the network changes in time with the arrival and departure of nodes. The secure dynamic recomposition of the network could become an important requirement in scenarios like battlefields where a soldier, under threat of capture, signs off the network on time.

The proposed tetrahedral algorithm can be applicable in several other scenarios such as for groups of people meeting in a room, (like students in a classroom, business meetings, mobile social networks etc). The password-based feature of our work could be used in cases where a group of people meets one another in person for the first time, and would like to go back home and set up a secure network among them.

The proposed algorithm leaves several open issues for future work. Formal analysis is necessary. The employment of the proposed internal algorithm to other algorithms like [18, 19] could also be useful. The incorporation of several new password-based key agreement protocols, which do not require the use of asymmetric encryption, is a challenging consideration. The dynamic case, where the network topology is rapidly changing, is also very interesting.

References

1. Verikoukis, C., Alonso, L., Giamalis, T.: Cross-Layer Optimization for Wireless Systems: A European Research Key Challenge. IEEE Communications Magazine 43(7), 1–3 (2005)
2. Bonnefoi, P.-F., Sauveron, D., Park, J.H.: MANETS: an exclusive choice between use and security? Special Issue on Interactive Multimedia & Intelligent Services in Mobile and Ubiquitous Computing (MUC) of Computing And Informatics 27(5) (2008)
3. Narayanan, A., Shmatikov, V.: Fast dictionary attacks on passwords using time-space trade-off. In: Proceedings of the 12th ACM conference on Computer and communications security, Alexandria, VA, USA (2005)

4. Bellovin, S.M., Merrit, M.: Encrypted key exchange: Password based protocols secure against dictionary attacks. In: Proceedings of the IEEE Symposium on Research in Security and Privacy, Oakland, USA (May 1992)
5. Diffie, W., Hellman, M.E.: New directions in cryptography. IEEE Transactions on Information Theory 22, 644–654 (1976)
6. Cucker, F., Smale, S.: Complexity estimates depending on condition and round off error. Journal of the Association for Computing Machinery 46(1), 113–184 (2000)
7. Koblitz, N.: Elliptic curve cryptosystems. Mathematics of Computation 4(8), 203–209 (1987)
8. Rivest, R., Shamir, A., Adleman, L.M.: A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Communications of the ACM 21(2), 120–126 (1978)
9. Menezes, A., Okamoto, T., Vanstone, S.: Reducing elliptic curve logarithms to logarithms in a finite field. IEEE Transactions on Information Theory 39, 1639–1646 (1993)
10. Menezes, A., Teske, E., Weng, A.: Weak Fields for ECC. In: Okamoto, T. (ed.) CT-RSA 2004. LNCS, vol. 2964, pp. 366–386. Springer, Heidelberg (2004)
11. Johnson, D., Vanstone, S.: The elliptic curve digital signature algorithm (ECDSA). International Journal on Information Security 1, 36–63 (2001)
12. Kalele, A., Sule, V.R.: Weak keys of pairing based Diffie-Hellman schemes on elliptic curves, Cryptology ePrint Archive 2005/30 (2005)
13. Zheng, D., Chen, K., You, J.: Multiparty authentication services and key agreement protocols with semi-trusted third party. Journal of Computer Science and Technology archive 17(6), 749–756 (2002)
14. Ateniese, G., Steiner, M., Tsudik, G.: New Multiparty Authentication Services and Key Agreement Protocols. IEEE Journal of Selected Areas in Communications 18(4) (April 2000)
15. Becker, C., Wille, U.: Communication complexity of group key distribution. In: 5th ACM Conference on Computer and Communications Security, San Francisco, California (November 1998)
16. Asokan, N., Ginzboorg, P.: Key agreement in ad hoc networks. Computer Communications 23, 1627–1637 (2000)
17. Askoxylakis, I.G., Kastanis, D.D., Traganitis, A.P.: Elliptic curve and password based dynamic key agreement in wireless ad-hoc networks, Communications, Networks and Information Security CNIS-2006, Cambridge, USA (October 2006)
18. Askoxylakis, I.G., Sauveron, D., Markantonakis, K., Tryfonas, T., Traganitis, A.: A Body-Centered Cubic Method for Key Agreement in Dynamic Mobile Ad Hoc Networks. In: Second International Conference on Emerging Security Information, Systems and Technologies, Cap Esterel, France, August 25-29, pp. 193–202 (2008)
19. Askoxylakis, I.G., Markantonakis, K., Tryfonas, T., May, J., Traganitis, A.: A Face Centered Cubic Key Agreement Mechanism for Mobile Ad Hoc Networks. In: First International ICST Conference on Mobile Lightweight Wireless Systems, MOBILIGHT 2009, Athens, Greece, May 18-20, pp. 103–113 (2009)