A Cluster-Based Framework for the Security of Medical Sensor Environments

Eleni Klaoudatou, Elisavet Konstantinou, Georgios Kambourakis, and Stefanos Gritzalis

Laboratory of Information and Communication Systems Security
Department of Information and Communication Systems Engineering
University of the Aegean, Karlovassi, GR-83200 Samos, Greece
{eklad,ekonstantinou,gkamb,sgritz}@aegean.gr

Abstract. The adoption of Wireless Sensor Networks (WSNs) in the healthcare sector poses many security issues, mainly because medical information is considered particularly sensitive. The security mechanisms employed are expected to be more efficient in terms of energy consumption and scalability in order to cope with the constrained capabilities of WSNs and patients' mobility. Towards this goal, cluster-based medical WSNs can substantially improve efficiency and scalability. In this context, we have proposed a general framework for cluster-based medical environments on top of which security mechanisms can rely. This framework fully covers the varying needs of both in-hospital environments and environments formed ad hoc for medical emergencies. In this paper, we further elaborate on the security of our proposed solution. We specifically focus on key establishment mechanisms and investigate the group key agreement protocols that can best fit in our framework. Keywords: Wireless Sensor Networks; Security; Medical Environments; Clustering; Group key management.

Keywords: Wireless Sensor Networks; Security; Medical Environments; Clustering; Group key management.

1 Introduction

One of the most promising contributions of Wireless Sensor Networks (WSN) is their adoption in the healthcare sector. Their use can improve the quality of medical care provided and facilitate patients' every day living. However, the adoption of WSNs in the healthcare sector also introduces many security issues and challenges mostly because medical services and the associated to them information are considered particularly sensitive. In general, every information system deployed in medical premises must comply with the following security requirements: confidentiality, integrity, availability, authentication, privacy, non-repudiation, authorization and accountability.

The mechanisms employed in medical WSNs should also consider patients' mobility without compromising the needs for security and efficiency. In order to provide the necessary security in such networks, we must rely on key establishment and management mechanisms which will guarantee the secure communication between the nodes. The choice of a key establishment protocol for the creation of a shared, secret

S. Fischer-Hübner et al. (Eds.): TrustBus 2009, LNCS 5695, pp. 52–62, 2009.

key must be done very carefully, taking into consideration all possible limitations of medical WSNs, e.g. nodes mobility, frequent topology changes and scalability needs. In particular, group key management mainly includes activities for the establishment and maintenance of a group key. Potentially, group key establishment is more suitable than pairwise key establishment as devices do not waste energy every time they need to communicate with another device by establishing a new shared secret key. Most of the traditional group key management protocols cannot cope with the dynamic nature and limitations of medical WSNs. However, key management protocols that are based on a cluster formation of the nodes have been proved to be more efficient and scalable especially when they are applied on WSNs. Clearly, these protocols can be envisioned in medical environments providing the most efficient solutions. In this context, many modern applications for medical environments assume a cluster-based sensor network (e.g. [1-3]) but they do not specifically focus: (a) on how clustering is applied and (b) how key management mechanisms can fit and be particularly effective on top of the clustered network.

In our previous work [4], we proposed a general framework for cluster-based medical environments on top of which security mechanisms can rely. This framework fully covers the varying needs of both intra-hospital environments and environments formed ad hoc for medical emergencies but it can be realized for hybrid scenarios as well. Despite the scalability of this framework, a question that remains unanswered is which key management protocols can be applied over it. Group key Agreement (GKA) protocols which do not require the presence of central entities seem to be more suitable for our framework. In this paper we elaborate on the most important cluster-based GKA protocols and we discuss which of them can be custom-tailored and thus profitable to medical applications. The performance of each examined GKA protocol is theoretically investigated as well.

The rest of this paper is organized as follows: Section 2 describes related work. Section 3 presents a short analysis of our two scenarios for sensor clustering in medical environments. Section 4 examines existing authenticated GKA mechanisms that can fit in our framework. Section 5 concludes the paper and gives pointers to future work.

2 Related Work

Clustering has been used in medical environments in various applications and for many purposes [1-3], [5-8]. In [1] clusters are formed ad hoc to accommodate emergency situations. In [2] the authors have applied their location-aware group membership middleware in an e-care scenario where cluster heads are responsible to send user information to every group the user joins. In medical environments, location awareness can help in tracking the nearest specialist to the location of the patient. In [5] clusters are created based on an infrastructure that uses the base station to elect the leaders of the clusters. In [6] a 3G telemedicine application is proposed based on energy-efficiency mechanisms in large-scale and a multi-class admission mechanism. It uses super-sensors as cluster-heads to query sensors for medical data and perform data aggregation, filtering and compression, and forward them towards the medical center. Clustering is based on the Zone Routing Protocol (ZRP). For security

purposes, the authors of [3] assume that every patient forms a different cluster and the cluster coordinator is a special entity called PSP (Patient Security Processor) that not only relays data to the gateway but is also responsible for the distribution of the symmetric key needed for encryption. Moreover, [7] and [8] describe two mechanisms for cluster-based key management for medical applications. The first one proposes a rekeying mechanism for tree-based networks while the latter proposes a cluster-based group key management mechanism for wireless networks. They apply this mechanism in a medical environment and use a bottom-up approach to specify and distribute group keys.

Compared to previous work, our framework [4], described shortly in the next section, has the advantage that is not case-oriented and can be applied in any cluster-based medical environment. The flexibility of this framework allows the efficient application of various security mechanisms which can cover all the security requirements of medical environments.

3 A Cluster-Based Framework for Medical Environments

WSNs can be deployed in several medical environments, like intra-hospital or medical emergencies. For this reason, our framework is comprised of two different scenarios based on the nature of the medical environment. Scenario I copes with medical environments which have a fixed infrastructure while scenario II considers infrastructureless environments. In both cases, a number of wireless sensors are implanted on every patient's body in order to collect and transfer real time vital sign data to a central database.

For the first scenario we consider a hierarchical network with Cluster-Heads (CH). This scenario is more suitable for environments where we can afford some more powerful nodes, which can play the role of CH, like intra-hospital environments. Consequently, we consider that CHs are fixed and energy consumption is not a key issue for them. The hospital sensor network can be decomposed to several clusters, based on their geographical location. For example, we can realize one cluster per one or more neighbouring rooms and one or more clusters for the external area of the hospital. This grouping scheme minimizes frequent topology changes each time a patient roams within the boundaries of her cluster. Clusters' number and size may vary according to the size of the hospital premises, the different units and the number of sensors as well as the number of fixed-nodes or CH available and their level of wireless coverage. For this scenario we assume that Cluster Members (CM) communicate with their CH every time they need to transfer data and that communication between each node and the CH is typically one-hop. As a result, the sensors used in this scenario do not need to have special processing capabilities and can be very cheap. The CH collects medical data from all nodes and forwards them towards the central database. Additionally, the CH can perform aggregation and filtering of the collected data. This method eliminates even more the amount of data in transit improving resource utilization too.

For the second scenario we assume that there aren't any powerful nodes to act as CHs. This architecture is more suitable for medical environments where there is no full coverage or no fixed infrastructure at all, as is the case of medical emergencies.

According to this scenario, sensors can be dynamically grouped into clusters which can be overlapping or not. Every time a node has some information to transmit, the node closer to the gateway (best path) is selected as the Cluster Leader (CL). The CL can either forward the data directly to the gateway, if it is located nearby, or forward the data via the CLs of adjacent clusters located near the gateway. To do so the CL must implement a multi-hop routing scheme. Communication between each node and the CL might also be multi-hop. Having in mind that the sensors' location may change very often, the CL responsibility will be automatically assigned to the node that is located closest to the gateway or to the CLs of neighbouring clusters located near the gateway. This means that all nodes should be able to potentially become CLs. As a result, nodes need to have some computing power and capabilities and therefore be more expensive than the ones employed in the first scenario.

A detailed description of the proposed framework can be found in [4]. Since nodes that belong to a cluster need to communicate securely there is also a need for a key management mechanism. In this paper, we mainly focus on GKA protocols that can best fit to our framework. In the following section we investigate GKA protocols that can best match with our scenarios and we evaluate their performance.

4 Authenticated Group Key Agreement Protocols

By nature WSNs are vulnerable to a number of threats already identified in several works, e.g. [9-11]. Every security solution adopted for the protection against these threats requires the employment and management of cryptographic keys. Group-key agreement protocols are considered to be more efficient than centralized key establishment schemes for WSN because devices do not waste energy every time they wish to communicate with another device by establishing a new shared secret key. However, most of the proposed GKA protocols offer protection only from passive attacks and an intruder can easily realize a man-in-the-middle attack during the key-agreement phase, to gain all the information needed for the keys. We therefore focus on authenticated GKA protocols where every node is authenticated in order to ensure that only valid group members participate in the key setup phase.

Clearly, GKA protocols that are based on cluster-based formation of the network nodes are more suitable for our framework (both for scenario I and II). In most of the cluster-based key agreement schemes proposed so far, a general key agreement protocol is applied in every cluster and then the clusters' keys are used by the same or another key agreement mechanism to form the final group key. Protocols that are relying on clustering are [12-20]. These recent works consider different cluster sizes and they are based on either two-party key agreement protocols, (like Diffie-Hellman protocol) or on other GKA protocols (like the well known protocol of Burmester and Desmedt (BD) [21]). The cluster-based protocol described in [12] uses virtual nodes and backbone nodes in addition to the real ones and requires many rounds (linear to the number of the nodes). The ID-AGKA protocol given in [13] has an extra communication overhead caused by the fact that each node must communicate with the Key Generation Centre (KGC) in every round. The protocol described in [14], considers clusters of arbitrary sizes and executes the Diffie-Hellman agreement between every two nodes in a cluster. A very recent work [15] presents a clique-based GKA which

uses the efficient BD protocol [21] within every clique. A clique is actually a cluster where each member can communicate in one-hop with every other member of the cluster. This is clearly a restriction for the clustering formation procedure. Another GKA protocol that is based on the communication efficiency of the BD protocol is [16]. In this work, the nodes are separated in clusters of specific size, then the CLs are organized in clusters of the same size and so on until a final root cluster is constructed. The protocol is energy efficient but this particular clustering formation of the nodes can be restrictive in a more general framework like ours.

The protocols described in [17-20] are more general and we will, therefore, examine their applicability in our two scenarios. The first two ([17] and [18]) are well suited for hierarchical networks while the other two ([19] and [20]) can be used in an ad hoc infrastructureless network. Their detailed description and the way they can be applied in our framework follow in the next subsections. We also evaluate the computational and communication cost of each solution in a comparative analysis. After the execution of each protocol, every node will have at its disposal several keys which can be used from the group members in order to communicate with each other securely.

4.1 Authenticated GKA Protocols for Scenario I

The GKA protocols presented in [17] and [18] are well suited for hierarchical networks and therefore can be applied in scenario I of our framework. We present here the main features of each protocol followed by a comparative evaluation in terms of computational and communication costs.

The protocol proposed in [17] is a password based GKA Protocol, for hierarchical wireless networks. Three main entities exist in the network, the main controller C (highest layer), various subgroup controllers (S_i) and several members (M). The protocol assumes that each subgroup member holds a password and a pairwise secret key shared with the subgroup controller S_i. Also the subgroup controller holds a password and a pairwise secret key which are shared with the main controller. Note that both the pairwise secret and the password are securely pre-loaded in the devices.

We can distinguish 3 phases for the establishment of a common group key. During Phase 1, S_i interacts with the subgroup members to compute the subgroup key K_i. In Phase 2, S_i interacts with the controller C to obtain the final group key K. Finally, in Phase 3, the group key K is sent downward securely by the controller to the subgroup controllers which are responsible to securely send the K to their members. Key confirmation messages are also sent together to verify and confirm the subgroup key K_i and final group key K. Clearly, this mechanism can fit into scenario I of our framework, where a hierarchical WSN is configured, i.e. if we consider that the main controller is the BS, the several subgroup controllers are the CHs of various clusters formed with several members each.

The authors in [18] propose an Identity-Based (ID-Based) scheme for secure communication in a hierarchical cluster-based ad hoc network. The scheme consists of two phases, Authentication and Communication phase. For the needs of our study, we will focus on the authentication phase. The protocol assumes that there is a Trusted Authority (TA) which is responsible for generating and issuing the secret information to the nodes and also that all CHs in the system are trusted entities. During the system

setup phase, the TA generates a pair of {public, private} keys, which are then used to compute a secret key K_i for each cluster member and a secret key CK_j for each CH. Then, the TA sends these keys to the corresponding nodes, along with the result of a public one-way hash function, and a number N, which is the product of four relatively prime numbers.

The first phase of the authentication procedure begins when a node joins a cluster and has to be authenticated by the CH. The node therefore generates an authentication token and sends it to the CH along with its ID and a timestamp. During the second phase, the CH checks the validity of the message sent by the node and if it succeeds it accepts the join request for the node. After that the CH computes a secret key K_{MH}, using the group key GK_j. This key is sent to the node along with the ID of the CH. In the third phase, the cluster-member checks the validity of K_{MH} and GK_j in order to prove that the ID of the cluster head (i.e., the $CHID_j$) is legal. Finally, those keys can be used by the CM as the secret key shared between him the CH and the group key.

In order to have a comparative evaluation of the two examined mechanisms, we can make the following assumptions. The TA in [18] performs similar functions to the controller in [17], therefore the role controller in the comparative tables 1 and 2 stands for both entities. The CH in [18] performs similar function to the S_i in [17] therefore we will adopt the description CH for both entities, as it is closest to our framework. Summarizing, we consider a WSN group of nodes comprised by: One controller, N nodes and n_c clusters. Finally, the symbol n_j signifies the number of members in the *j-th* cluster (or subgroup) and symbols CH_j and $CM_{i,j}$ signify the CH of the *j-th* cluster and the *i-th* CM of *j-th* cluster, accordingly. Table 1 shows a comparative analysis of the two mechanisms discussed, in terms of computational and communication costs. Additionally, Table 2 presents the prerequisites, i.e., the number of keys that need to be stored and the number of keys available after the execution of each protocol.

Table 1. Comparative Analysis of the two Mechanisms for Scenario I

Protocol		Mod. Exp.	Messages Sent	Messages Received
[17]	C	$2n_c+2$	2	n_c
	CH_j	$2n_j+5$	3	n_j+2
	$CM_{i,j}$	3	1	2
[18]	C	$N+1$	$N+1$	0
	CH_j	n_j	$2n_j$	n_j+2
	$CM_{i,j}$	1	1	5

Table 2. Prerequisites – Keys Stored – Keys Created (Scenario I)

Protocol	Prerequisites	Keys stored	Keys created
[17]	password shared with S_i and C ($pw_{i,j}$) password shared with the controller (pw_i) pairwise secret key between $CM_{i,j}$ and S_i ($K_{i,j}$) pairwise secret key between S_i and C(K_i)	K_i, $K_{i,j}$	subgroup key K_i final group key K
[18]	secret key for node *i* (K_i) a secret key for $CHID_j$ (CK_j)	K_i, CK_j	a group key GK_j K_{MH} (CM-to-CH)

The results of Table 1 show that protocol [18] has a lower communication and computation cost per node. However, this is something we should expect since the protocol is not contributory (i.e. not all the members contribute to the creation of the group key) and therefore the load of computation and communication operations is shifted to the hierarchically higher members, namely the CHs and the Controller. The protocol described in [17] on the other hand, is contributory which means that computation and communication operations are distributed to all nodes. Moreover, protocol [18] also calculates various Hash functions and performs some extra symmetric operations which are not included in our evaluation.

4.2 Authenticated GKA Protocols for Scenario II

The protocols described in [19] and [20] are well suited for our second scenario. This is because they can be applied in dynamic networks comprised of nodes with similar capabilities. Here, we present the main features and a comparative evaluation of these two protocols.

The protocol described in [19] is an authenticated group-key agreement protocol for ad hoc networks and it is based on hierarchical authentication. The protocol is ID-based and uses pairwise keys for entity authentication. It comprises of two phases: (1) organization of the nodes into clusters and (2) generation of the group session key. For the needs or our study we will focus on the latter phase. The protocol assumes that each node is equipped with a secret Group Identity Key (KIG), a one-way hash function H() and a local identifier (ID). Also, every node is able to compute its weight. This weight is a numerical quantity that expresses the node's current status in terms of node's mobility, battery power level, distance from the other nodes, and values related to the surrounding environment (terrain, temperature, battery power, etc). Moreover, the protocol assumes that identities are publicly available. A TA is also needed during the setup phase to generate the private keys for every node.

We can distinguish 3 different phases of the protocol, for the construction and distribution of the final group key. During the first phase, which is also the setup phase, every node computes the pairwise shared secret key using the secret key and the hash of the ID of the other node. This pairwise secret key is used by each member for the authentication phase. Therefore, this phase does not require communication between members. In the second phase, a mutual authentication procedure takes place between members that belong to the same cluster (inner-cluster authentication). The result of this phase is that every node within the cluster is authenticated with the others and with the CL and that each cluster holds a cluster session key. During the third phase, the protocol is repeated with the upper level clusters, considering that the CLs that participated in the previous phase are now the children for their one level hierarchically higher CL. The CL in the lower level has to decrypt every message he receives from upper levels to first check the identity of the transmitter and then forward the message to his children (re-)encrypted using the cluster session key he shares with them. This procedure repeats until the root level is reached.

This mechanism blends with scenario II of our framework, if we consider that every node authenticates its neighbour and the neighbour his next node and so on until they reach the CL. This is useful for our scenario since communication between the CM and the CL might not always be one-hop. Due to the infrastructure-less form

of the WSN in scenario II, the frequent changes of the cluster structure do not introduce significant additional cost, since every node that joins a cluster has only to obtain from the CL the local parameters of the new cluster and the cluster session key. On the other hand, the number of rounds increases along with the number of hierarchical levels.

The protocol described in [20] is a cluster-based GKA protocol based on Joux's tripartite key agreement protocol [22] and comes in two versions, namely contributory and non-contributory. Although in the current version of the protocol nodes are not authenticated during the key agreement phase, authors state that the protocol can easily provide members' authentication. This can be achieved by substituting Joux's tripartite key agreement protocol with an authenticated version of it, like those described in [23] or [24]. For the needs of our study we will examine an authenticated version of this protocol, using the ID-based tripartite authenticated key agreement protocol from [24]. The protocol assumes clusters consisted of either two or three members each. The authenticated version of the protocol also assumes that a Key Generation Center (KGC) exists and participates in the generation of the public/private keys for each member. However, this operation is held once, during the setup phase and thus can be considered as an offline procedure.

The protocol is consisted of 3 main phases following the setup phase. During the setup phase every node has to obtain a long term private key from the KGC. In order to do so, every node sends its long-term public key to the KGC, calculated based on its Identity (ID). The KGC calculates and sends back the private key. During the first phase of the authentication procedure every member computes two elliptic curve points Pi and Ti and sends them towards the other members of its cluster. Nodes in the lower levels belong to only one cluster, but nodes in upper levels belong to two clusters and therefore will have to send those points to a greater number of nodes (actually, this number depends on the size of the clusters the nodes belong to). During the second phase an AuthCreateClusterKey procedure is executed simultaneously in every cluster. Every member first verifies the other members of the same cluster and if verification succeeds, each member calculates the common secret key $K_{cluster}$. By the end of this phase every group member shares a secret key with the nodes of the clusters it belongs to. Finally, in the third phase of the protocol, the root key (K_{root}) is sent downwards to the members of the lower levels, one level at a time, by encrypting the key with the session key of every level. This phase takes as many steps as the height of the tallest branch of the cluster-based structure.

In order to have a comparative evaluation of the two mechanisms we can make the following assumptions. Since both protocols are based on elliptic curve cryptography, we will calculate the number of scalar multiplications and pairings that each entity has to perform. Both protocols assume a tree-based structure where nodes in the lower levels belong to only one cluster and nodes in the upper level (who are actually the CLs of previous levels) belong to at least 2 clusters. Therefore, every node in the lower level will be called a leaf node and their CLs that belong in the upper level structure will be called intermediate nodes.

Table 3 shows a comparative analysis of the two mechanisms discussed, in terms of computational and communication costs. In order to be able to evaluate the protocol in [19] we make the assumption that each cluster is consisted of 4 members. This is an assumption the authors also make while describing the protocol. We have also

Table 3. Comparative Analysis of the Two Mechanisms for Scenario II

Protocol		Scalar multiplications (SM) and Pairings (P)	Messages Sent		Messages Received	
			other rounds	last round (root key)	other rounds	last round (root key)
[19]	intermediate nodes	6 (SM) + 6 (P)	3+1	1	2+1	1
	leaves	3 (SM) + 3 (P)	1	0	1	1
[20]	intermediate nodes	7 (SM) + 10 (P)	4	1	4	1
	leaves	5 (SM) + 5 (P)	2	0	2	1

Table 4. Prerequisites-Keys Stored – Keys Created (Scenario II)

Protocol	Prerequisites	Keys stored	Keys created
[19]	- a TA to calculate the private keys - private key - 1 hash function	master key S	- a pairwise key shared between two members - a cluster session key - a root key
[20]	- a KGC for the setup phase - a public/private key pair for every node - 2 hash functions - a cluster of 2 or 3 members each	public/private key of each member	- a shared cluster key for every cluster - a root Key (K_{root})

calculated the communication cost of the final phase which includes the distribution of the root key in a separate column of the table, in order to keep the calculations as independent from the number of levels as possible. Table 4 presents the prerequisites of every protocol, the number of keys that need to be stored in each node and the number of keys available after the execution of the protocol.

Based on the results of Table 3 we can see that protocol [19] has a lower communication and computation cost per node. However, the protocol in [19] requires a larger number of rounds and more memory in every node for the key storage, i.e., a key must be kept for every pair of nodes which belong to the same cluster. Moreover, it requires the presence of some sort of TA during the setup phase.

5 Conclusion and Future Work

In the previous sections we examined several authenticated GKA protocols that can be custom-tailored to our scenarios. The main issue here is whether these protocols can be implemented by limited-resources devices. In order to deal with this, we must first take a look at the capabilities of the sensors usually employed in medical implementations. Most of the well-known medical sensors implementations use Crossbow's Mica, Mica2, MicaZ and Telos motes and Tmote Sky by Texas Instruments. The Mica2 sensor is build upon ATmega128L processor and Chipcon IEEE 802.15.4 compliant radio interface whereas Tmote Sky is build upon Texas Instrument's

MSP430F1611 microcontroller and Chipcon's CC2420 radio interface. Their memory ranges between 4-10 KB RAM and the maximum data rate of the radio interface is 250 Kbps. Based on the performance measurements presented in [25], a Mica2 mote consumes 30.02mJ for a prime field point multiplication and 423.87mJ for a pairing. The energy consumption for a prime field point multiplication and a pairing performed by Tmote Sky is 7.95mJ and 130.49mA accordingly. We therefore believe that the examined GKA protocols are viable and attractive for medical sensor environments since every node is able to perform ECC scalar multiplications, exponentiations and pairings [26].

For security sensitive environments, as is the case of medical environments, the employment of only a group-key management mechanism is not enough in order to provide data confidentiality and integrity. The fact that data aggregation and filtering procedures are performed at intermediate nodes implies that such nodes access and process data from every patient. This might lead to a privacy threat. Therefore, secure aggregation/filtering mechanisms need to be employed, in order to ensure data integrity and confidentiality and perhaps privacy. For future work we plan to also examine how secure aggregation mechanisms can blend with our framework.

References

1. Dembeyiotis, S., Konnis, G., Koutsouris, D.: A Novel Communications Network for the Provision of Medical Care in Disaster and Emergency Situations. In: 24th EMBS/IEEE, San Francisco (2004)
2. Bottazzi, D., Corradi, A., Montanari, R.: AGAPE: a location-aware group membership middleware for pervasive computing environments. In: 8th IEEE International Symposium on Computers and Communication, pp. 1185–1192. IEEE CS Press, Turkey (2003)
3. Misic, J., Misic, V.B.: Implementation of security policy for clinical information systems over wireless sensor networks. Ad Hoc Networks 5(1), 134–144 (2007)
4. Klaoudatou, E., Konstantinou, E., Kambourakis, G., Gritzalis, S.: Clustering Oriented Architectures in Medical Sensor Environments. In: International Workshop on Security and Privacy in e-Health, March 2008, pp. 929–934. IEEE CS Press, Barcelona (2008)
5. Schwiebert, L., Gupta, S.S., Weinmann, J.: Challenges in Wireless Networks of Biomedical Sensors. In: SIGMOBILE 2001, pp. 151–165 (2001)
6. Hu, F., Kumar, S.: QoS considerations in wireless sensor networks for telemedicine. In: SPIE ITCOM Conference, Orlando, FL (2003)
7. Hu, F., Tillett, J., Ziobro, J., Sharma, N.K.: Secure Tree-Zone-Based Wireless Sensor Networks for Telemedicine Applications. IEEE GLOBECOM, 345–349 (2003)
8. Chen, Y.J., Wang, Y.L., Wu, X.P., Le, P.D.: The Design of Cluster-based Group Key Management System in Wireless Networks. In: International Conference on Communication Technology (2006)
9. Karlof, C., Wagner, D.: Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures. Ad Hoc Network Journal, special issue on sensor network applications and protocols (2002)
10. Raymond, D.R., Midkiff, S.F.: Denial-of-Service in Wireless Sensor Networks: Attacks and Defenses. IEEE Pervasive Computing 7(1), 74–81 (2008)

11. Kambourakis, G., Klaoudatou, E., Gritzalis, S.: Securing Medical Sensor Environments: The Codeblue framework case. In: 2nd International Conference on Availability, Reliability, and Security - 1st International Symposium on Frontiers in Availability, Reliability and Security, April 2007, pp. 637–643. IEEE CS Press, Austria (2007)
12. Shi, H., He, M., Qin, Z.: Authenticated and communication efficient group key agreement for clustered ad hoc networks. In: Pointcheval, D., Mu, Y., Chen, K. (eds.) CANS 2006. LNCS, vol. 4301, pp. 73–89. Springer, Heidelberg (2006)
13. Yao, G., Ren, K., Bao, F., Deng, R.H., Feng, D.: Making the key agreement protocol in mobile ad hoc network more efficient. In: Zhou, J., Yung, M., Han, Y. (eds.) ACNS 2003. LNCS, vol. 2846, pp. 343–356. Springer, Heidelberg (2003)
14. Chen, Y., Zhao, M., Zheng, S., Wang, Z.: An Efficient and Secure Group Key Agreement Using in the Group Communication of Mobile Ad-hoc Networks. In: IEEE CIS 2006, pp. 1136–1142. IEEE CS Press, Los Alamitos (2006)
15. Hietalahti, M.: A clustering-based group key agreement protocol for ad-hoc networks. Electronic Notes in Theoretical Computer Science 192, 43–53 (2008)
16. Teo, J.C.M., Tan, C.H.: Energy-Efficient and Scalable Group Key Agreement for Large Ad Hoc Networks. In: 2nd ACM international workshop on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks, pp. 114–121 (2005)
17. Teo, J.C., Tan, C.H.: Denial-of-service resilience password-based group key agreement for wireless networks. In: 3rd ACM Workshop on QoS and Security For Wireless and Mobile Networks, Crete Island, Greece, pp. 136–143. ACM Press, New York (2007)
18. Lee, J., Chang, C.: Secure communications for cluster-based ad hoc networks using node identities. Journal of Network and Computer Applications 30(4), 1377–1396 (2007)
19. Abdel-Hafez, A., Miri, A., Oronzo-Barbosa, L.: Authenticated Group Key Agreement Protocols for Ad hoc Wireless Networks. International Journal of Network Security 4(1), 90–98 (2007)
20. Konstantinou, E.: Cluster-based Group Key Agreement for Wireless Ad Hoc Networks. In: ARES 2008, pp. 550–557. IEEE Press, Los Alamitos (2008)
21. Burmester, M., Desmedt, Y.G.: A secure and efficient conference key distribution system. In: De Santis, A. (ed.) EUROCRYPT 1994. LNCS, vol. 950, pp. 275–286. Springer, Heidelberg (1995)
22. Joux, A.: A one round protocol for tripartite Diffie-Hellman. In: Bosma, W. (ed.) ANTS 2000. LNCS, vol. 1838, pp. 385–394. Springer, Heidelberg (2000)
23. Al-Riyami, S.S., Paterson, K.G.: Authenticated three party key agreement protocols from pairings. In: Paterson, K.G. (ed.) Cryptography and Coding 2003. LNCS, vol. 2898, pp. 332–359. Springer, Heidelberg (2003)
24. Zhang, F., Liu, S., Kim, K.: D-based one round authenticated tripartite key agreement protocol with pairings (2002), `http://eprint.iacr.org`
25. Szczechowiak, P., Oliveira, L.B., Scott, M., Collier, M., Dahab, R.: NanoECC: Testing the limits of elliptic curve cryptography in sensor networks. In: Verdone, R. (ed.) EWSN 2008. LNCS, vol. 4913, pp. 305–320. Springer, Heidelberg (2008)
26. Roman, R., Alcaraz, C., Lopez, J.: A survey of cryptographic primitives and implementations for hardware-constrained sensor network nodes. Mobile Networks Applications 12(4), 231–244 (2007)