Securing Medical Sensor Environments: The CodeBlue Framework Case

Georgios Kambourakis, Eleni Klaoudatou and Stefanos Gritzalis

{gkamb, eklad, sgritz}@aegean.gr

Laboratory of Information and Communication Systems Security
Department of Information and Communication Systems Engineering
University of the Aegean, Karlovassi, GR-83200 Samos, Greece

Tel: +30-22730-82247 Fax: +30-22730-82009

Abstract—**Recently, research on wireless sensor networks targeting to medical environments has gathered a great attention. In this context, the most recent and perhaps the most promising complete scheme is the CodeBlue hardware and software combined platform, developed in the context of the self-titled Harvard's University project. CodeBlue relies on miniature wearable sensors to monitor real-time patients' vital activities and collecting data for further processing. Apart from the essential query interface for medical monitoring, CodeBlue offers protocols for hardware discovery and multihop routing. This paper contributes to the CodeBlue security, which until now is considered as pending or left out for future work by its designers. We identify and describe several security issues and attack incidents that can be directly applied on CodeBlue compromising its trustworthiness. We also discuss possible solutions for both internal and external attacks and the key-management mechanisms that these solutions presume.**

Index Terms— Sensor networks security, Medical sensor networks, CodeBlue prototype.

I. Introduction

A fast growing application for wireless sensor networks involves their use in the medical sector. For example, medical stuff (doctors, nurses, etc) can constantly monitor in real-time mode sensitive health data of their patients given that the latter are equipped with tiny wearable sensors capable of providing vital information. Moreover, in emergency or disaster scenarios, sensors technology would offer medics the opportunity to efficiently provide better services to those who need it. However, in contrast to sensor networks which employ stationary nodes, base stations (infrastructure) and transmit data at relatively low data rates, health monitoring requires higher data rates, reliable communication and multiple mobile receivers (e.g. PDAs carried by caregivers). In addition, as medical data are classified in most cases as private and sensitive, health monitoring must ensure high level security and privacy for the data transmitted or stored in local databases.

While some other projects or solutions employing wearable sensors exist [1-6], the recently developed CodeBlue prototype medical sensor network platform [7] is until now perhaps the most complete proposal in the field. CodeBlue utilize a range of medical sensors integrated with the commonly used Mica2, MicaZ and Telos mote designs. These include a pulse oximeter, two-lead electrocardiogram (EKG), and a specialized motion-analysis sensor board. All sensors are responsible to collect patients' vital data and transmit them either to local medical databases containing medical records or directly to the caregivers' mobile devices. To address this requirement the CodeBlue protocol and middleware framework implemented in TinyOS [8] offers protocols for device discovery, publish/subscribe multihop routing, and a simple query interface which enable medical stuff to request data from groups of patients. CodeBlue also incorporates an RF-based localization system, called Mote-Track [9], to monitor the location of both patients and caregivers.

On the downside, as CodeBlue designers have until now emphasized on the above functionalities, they have left out the security issues for future work. This paper focuses on CodeBlue security identifying and categorizing numerous potential threats and vulnerabilities which can undermine the framework's reliability and robustness.

The rest of this paper is structured as follows: next section gives an overview of the CodeBlue architecture, which is considered essential for the sections to follow. Section III reports on CodeBlue security presenting our analysis on every category of threats and attacks identified. Section IV presents possible countermeasures and remedies against them in the context of the CodeBlue architecture. Finally, Section V offers some concluding thoughts and future directions of this work.

II. General CodeBlue Architecture

This section briefly describes some essential aspects of the CodeBlue architecture including: the way sensors are organized to comprise a network, the communication method between the involved sensors, the way the collected by the sensors data are delivered. Some basic aspects of the CodeBlue architecture are presented in Figure 1. Physical and layer 2 communications between CodeBlue nodes are delivered using the IEEE 802.15.4 [10] specification. On the other hand, for data (packets) routing CodeBlue designers decided to utilize a publish/subscribe protocol namely Adaptive Demand-Driven Multicast Routing (ADMR) [11] because is simple, extensively tested and fits better in medical applications.

According to ADMR, whenever a certain sensor node wishes to transmit data it must advertise its intention in the channel used. Similarly, every device that wishes to receive that data must subscribe itself in order to receive it. Data routing is assisted by other existing network nodes which are arranged by ADMR as forwarders. The latter relay messages coming from certain channels. The procedure of constructing the routing path is as follows: Every CodeBlue node keeps updated a node table indexed by the publisher node ID. Each record in that table contains both the path cost from the publisher to the current node and the previous hop in the best path from the publisher. Thus, each node knows which is the best path originating from every publishing node and ending to itself. In case a subscriber wishes to receive data from a specific channel, it sends a unicast route reply message along the reverse path from itself to the publishing device, using the previous-hop information in the node table. Each intermediate node receiving the route reply acts as a forwarder for the requested channel and will consequently relay received messages for that channel. Best paths from publishers to subscribers must be constantly maintained due to e.g. node movement. This is achieved by periodically propagating a controlled broadcast flood that revise the node tables on all the intermediate nodes.

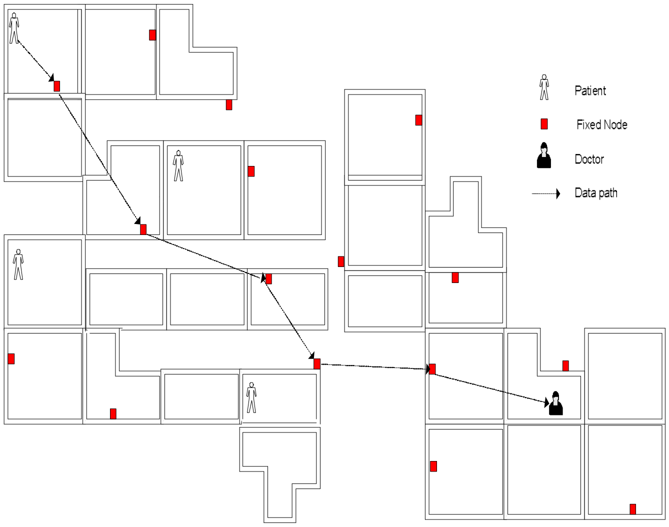
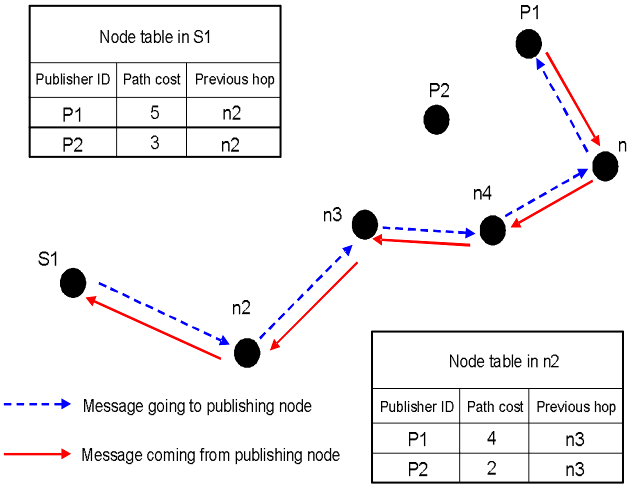


Fig. 1. General CodeBlue architecture

Figure 2 depicts the path construction procedure between publishing node P1 and node S1. At first, S1 send a registration message to P1 through intermediate nodes n2 to n5 which become forwarders. Upon reception P1 will transmit the corresponding data to S1 through the forwarders.

Moreover, ADMR supports a special broadcast channel that employs a simple controlled flooding mechanism to transmit unreliably a message to every node in the network cloud. This broadcast channel is used by every CodeBlue node to periodically publish information about itself, e.g. node ID and supported sensor types. As a result, the receiving devices that wish to learn about other nodes in the network can subscribe to the broadcast channel to receive this information. Fixed nodes (infrastructure) are not mandatory for CodeBlue to operate. CodeBlue devices can also be organized and deliver services in an ad-hoc fashion. However, quality of service in indoor environments (e.g. in hospital premises) can be significantly increased when a backbone of fixed communication nodes is in place.


P1
P2
n5
n4
n3
S1
n2
Message going to publishing node
Message coming from publishing node

Node table in S1

Publisher ID	Path cost	Previous hop
P1	5	n2
P2	3	n2

Node table in n2

Publisher ID	Path cost	Previous hop
P1	4	n3
P2	2	n3

Fig. 2. ADMR routing example

CodeBlue team has created a simple query interface too. As in Directed Diffusion [12] and TinyDB [13], this interface allows CodeBlue devices to receive filtered data by specifying the sensors, data rates, and optional certain filters that should be used for data delivery. More simply put, CodeBlue queries are created by the user equipment (e.g. PDA, laptop) and instruct CodeBlue nodes to publish data that meet the query conditions on a certain ADMR channel. The general structure of a query is as follows: (S, τ, chan, ρ, C, p), where S indicates the set of node IDs that should report data for this query, τ stands for sensor type, chan specifies the channel where the data must be transmitted, ρ is the sampling rate, C the number of samples that the device wishes to receive and p denotes the filter conditions, if any. For example, the query ({3,7}, EKG, 38, 1.0Hz, ∞, {(HR<50) OR (HR>200)}) means that nodes 3 & 7 should transmit electro-cardiogram data in channel 38 every second, when patient's heart pulses is below 50 or above 200.

As already mentioned CodeBlue utilizes a decentralized RF-based tracking system to locate the exact position of patients and caregivers. Location tracking procedure comprises of the following steps: (a) Fixed (beacon) nodes transmit periodically beacon messages using a range of frequencies and transmission power levels. (b) Mobile nodes eavesdrop for these beacons and create a signature consisting of certain parameters, namely average received signal strength (RSSI) for each beacon node, frequency, and power level. (c) After that, the generated signature is compared to a database of all the pre-obtained signatures (corresponding to a number of known locations) to create a 3-Dimentional location. Mapping process can be decentralized when the signature database is mirrored across the set of fixed nodes.

III. On CodeBlue Security

Clearly, medical environments and the associated with them information are considered particularly sensitive. As a result, every informative system deployed in medical premises must comply with the following well known security requirements: Confidentiality, Integrity, Availability, Authentication, Privacy, Non-repudiation, Authorization and Accountability. On the other hand, attacks on wireless sensor networks can be classified in four general categories: Denial of Service (DoS), Snooping, Modification and Masquerading.

This section attempts to analyze CodeBlue from a security point of view, pointing out whether the above requirements are met. To address this issue we identify and discuss all major attacks that enroll in the above categories and can be applied to CodeBlue system. Naturally, many of the discussed threats/attacks are closely related to the ADMR protocol and not to CodeBlue itself. Moreover, by nature, sensor networks are vulnerable to most of the following threats considering the openness of the wireless medium, the anonymous, (semi)uncontrolled terrain between various endpoints and sensors' limitations to processing power that prevent them from employing strong cryptographical methods.

A. *DoS Attacks*

Jamming: This attack applied in the physical layer is targeting to jam certain or all CodeBlue nodes in a given area. Consequently, all jammed nodes are denied communications. To realize this attack, adversaries can utilize specialized jamming equipment or just normal customized devices - possibly using some sort of amplifiers to amplify signal strength or simply create noise - to interfere with CodeBlue sensors in the same frequencies. At worst, this attack can isolate certain patients putting their lives at risk.

Stealth DoS: The attacker tries to exploit the trust that some network nodes show to him in order to fragment the network or to simply isolate some nodes. This situation can be achieved employing numerous methods and techniques. First of all the perpetrator could force some network elements to consume its energy resources. This can be accomplished either by requesting a node e.g. a sensor to continuously transmit data or to continuously route large amounts of data through that target-node. For example, according to the first case the attacker constructs CodeBlue queries with large sampling rate (ρ) against certain publisher IDs, while the latter exhausts specific forwarders by forcing them to relay sensor data constantly.

Routing loops: The target of this attack is to someway disrupt network routing. Malevolent users could modify the address fields in the received packets before they forward it to the next network hop, creating that way infinite routing loops. More specifically, in CodeBlue the attacker can alter the header of the ADMR packets changing one or more of the address fields (senderAddr, destAddr, originAddr, groupAddr) it contains. As a result, the modified packets cannot reach their destination, while on the other hand consume forwarders' resources and network bandwidth aimlessly.

Black or Grey holes: The attacker sends bogus packets announcing that she lies in the shorter path to the network node under attack. Put another way, in CodeBlue the impostor could alter the ADMR header of certain packets e.g. by specifying small hopcount or good link quality indicator, making the adjacent nodes believe and subsequently update their routing tables, that the attacker is located in the shorter path to some destination. After that, the adversary can freely drop every packet he receives, thus creating a black hole, or selectively allowing to some of them to pass. This situation is depicted in Figure 3.

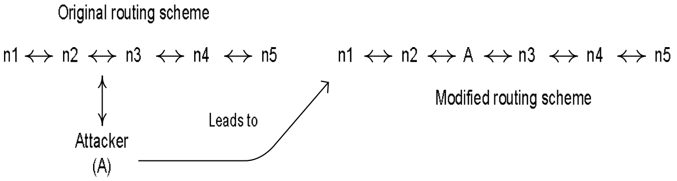


Fig. 3. Black hole style attack example

B. *Snooping Attacks*

Eavesdropping: This attack enables the perpetrator to intercept the data being communicated between e.g. sensor nodes. Latest CodeBlue technical report [7] does not mention whether the framework employs some cryptographical methods in the upper layers. We can even assume that 802.15.4 is used in its default insecure mode, namely without ciphering and authentication services enabled.

Location Attack: Compromising patients/caregivers' location privacy the attacker attempts to find the area where the target is currently located. This attack can be easy implemented in CodeBlue since it is based on the publish/subscribe model meaning that all the sensor nodes are constantly in traceable mode publishing the available data in hand. Consequently, as soon as the target enters within the signal range of the aggressor he can intercept its permanent ID and therefore recognize the patient, caregiver, etc. This inroad can be particularly profitable if the attacker manages to subscribe in the CodeBlue network as a legitimate user (insider). After that, using Mote-Track he is able to simply subscribe and request for the exact position of the node under surveillance.

C. *Modification Attacks*

The primary target of a modification style attack is to somehow forge specific network traffic. Adversaries exploit the fact that the routing protocol in use (ADMR in our case) presumes that all the forwarders are trusted and they do not modify packets passing through them. However, a fraudulent insider could opt to change packets' payload endangering the safety of the patients. For example, one could modify cardiograph sensor data sent to medical stuff showing that the patient has a heart stroke to appear as normal heart beats. A skillful attacker could also falsify medical records of certain patients with erroneous indications.

Stealth traffic hi-jacking: Apart from modifying packets' payload containing sensitive medical data the attacker could falsify routing information aiming to reroute traffic from/to specific network elements through other nodes or paths that he/she controls. Black or grey hole style attacks are classified in this category.

D. *Masquerading Attacks*

Impersonation: The attacker camouflages its device to make the others believe that he/she is someone else. The simplest way to launch this attack is to modify the sender address in ADMR packets which transmits putting the address of another network node.

Sybil attack: During this attack the attacker's device appears to have multiple identities in order to control a significant part of the network. In the CodeBlue context this attack could have serious implications especially when the network operates in ad-hoc mode. As described in Figure 4, the attacker (A) misleadingly persuades nodes B and C to believe that A1, A2 and A3 are their neighborhoods. Consequently, C is bound to select one of them every time he/she wishes to route some traffic. Eventually, all transmitted data will arrive to A who has hi-jacked and subsequently controls all communication traffic around him. In the CodeBlue case this attack could have some other implications as well. For example, the attacking node acting as publisher could advertise through his multiple false identities that he has medical data to send. This situation will probably either fill with bogus entries or overflow the node tables of the subscribing naïve network entities.

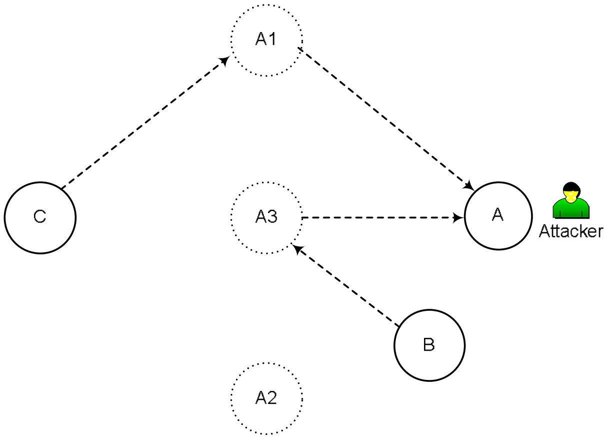


Fig. 4. Sybil attack example

E. *Other Attacks*

It is well known that erratic behaviors in sensors networks seeking physical access to sensor devices are difficult to be repelled due to the anonymous and (semi)uncontrolled terrain in most cases. It is stressed that physical access to CodeBlue sensors could have serious implications for the patients like data alteration, false alarms, disconnections, etc. At best, physical access to a certain sensor enables the aggressor to obtain sensor's secret keys. According to [14] a competent attacker equipped with a laptop is able to retrieve sensor keys in less than a minute given that he/she has physical access to it. The experiments were performed utilizing sensors integrated with the commonly used Mica family designs and especially with Mica2 (which is used in CodeBlue sensors as well). CodeBlue sensors employ the Chipcon CC2420 chip, which as cited in [15] is able to support only two secret keys. Once these keys are compromised the attacker has access to the communications of the whole network. As already mentioned, latest CodeBlue technical report makes no reference to security issues like ciphering algorithms to be used, key administration, security modes, etc. IEEE 802.15.4 specifications per se determine four distinct security modes which are: no security, confidentiality (ciphering) only, authentication only, authentication and confidentiality. However, some of them do have some open security issues too which must be carefully considered [10, 15].

IV. SOLUTIONS ON CODEBLUE SECURITY

All possible attacks could be categorized into two major groups, namely internal and external attacks, based on whether the attack is being provoked by an entity that is part of the network or by an outside entity who has, somehow, gained access to the network. Protection against external attacks could be achieved by securing the data-link layer and by employing techniques such as authentication and data encryption in order to prevent the attacker from gaining access to the network. On the other hand, protection against internal attacks is more difficult to be attained, since every node that is part of the wireless sensor network is presumed to be trusted by all the others. Message encryption and node authentication can also offer some level of protection but there is also a great need for more secure routing protocols. However, as stated in [16], there are some attack categories, such as the Sybil attack, wormhole and sinkhole attacks and HELLO Flood attacks that cannot be easily repelled by the aforementioned mechanisms. Such attacks require stronger mechanisms like the geo-routing protocol, efficient key management schemes, etc.

In the following sections we present possible solutions to protect against major attacks presented in section III. These solutions should be examined in order to better comprehend how they can be integrated within the CodeBlue architecture and offer protection against jamming and routing modifications. A separate section focuses on possible solutions for internal attacks that cannot be easily prevented, such as the Sybil attack, the Wormhole attack and HELLO Flood attacks. In the last section, the most common key management schemes are discussed.

A. *Jamming*

The most common protection against jamming is the employment of the spread-spectrum and frequency hopping communication as in Bluetooth standard. According to those techniques, transmitter and receiver must know the spread code or the hopping sequence in order to be able to distinguish signal from noise. Of course, if the attacker knows or finds out a way to follow the hopping sequence, then he/she is able to jam the network. Direct-sequence spread spectrum is more effective but can be energy consuming since it requires a high-power wideband signal [17]. Since jamming cannot be prevented in many cases, several techniques could be employed in a wireless sensor network in order to detect and throttle the attack only in the unsound area of the network [18, 19]. These solutions employ techniques to detect the jamming and map the affected region. At this point, one can deploy conventional means to remove the attacker or route around the jammed area [18].

Another category of solutions aims to provision an alternative way for data transmission in cases where important data should be sent by a node that is being jammed. Based on the first solution [20] a node that is being jammed could still transmit data to an unaffected node by using higher power for transmission. The unaffected node could then relay the message on its behalf. Of course, care should be taken in energy management since continuous transmission in high-power could lead to power exhaustion. Another solution would be the provision of a backup communication mode to be used in cases of interference. Based on this solution, a node could switch to another supported communication mode, such as acoustic, infrared or optical [21], whenever he has to transmit important data.

B. *Routing Modification Attacks*

As noted in section III, many of the discussed threats/attacks are closely related to routing data modification from attackers. The majority of these attacks can be prevented assuming that all nodes need to be authenticated in order to be able to transmit to and receive messages from the network. Additionally, encryption can offer integrity and confidentiality of the messages exchanged. The combination of authentication and encryption supports protection against attacks that rely on routing modification.

ADMR protocol assumes that each node should be subscribed in a certain channel in order to be able to receive messages from a network entity. Thus, authentication can take place during the subscription phase. ADMR also has a mechanism for periodic updating of routing tables based on the transmission of broadcast messages in a separate channel. Broadcast messages should also be authenticated in order to avoid HELLO Flood Attacks. CodeBlue architecture does not mandates Base Stations (fixed nodes), but in the case where a Base Station exists, adversaries must not be able to spoof broadcast of flooded messages from the corresponding base station. This means that every node should be able to verify messages from the base station but not to alter them. In such cases the μTESLA protocol can be used [22] for authentication of broadcast messages with minimal packet overhead. Other methods that could be examined are SPINS [22] that provides confidentiality via a chaining encryption function and also protection against replay attacks, and authenticated broadcasts based on the μTESLA protocol.

Given that in most cases attacks are difficult to be prevented, several solutions focus on the creation of resistant networks that will be able to operate even if a certain area is being compromised. These solutions provide mechanisms for intrusion tolerant routing. This can be achieved by gathering multiple redundant paths between the nodes and check them for consistency, like in the INSENS protocol [23], or apply multiple disjoint paths [24] for data forwarding. Some routing protocols send packets along multiple, independent paths and verify the consistency among packets received at the destination node [23].

It is stressed that for node authentication and message encryption, it is necessary to specify a key management mechanism and specifically denote the method keys are distributed to the nodes. Several key management mechanisms have been proposed. The main differences between them rely on whether private or public infrastructure is employed as well as if a central authority, such as a Certification Authority (CA) is present. These mechanisms are discussed in subsection D. It should be noted that although several simple key management mechanisms are capable to offer protection against external attacks, they offer very little protection in the case of a node being compromised. Therefore, more advanced techniques are required, always keeping in mind the limited computational and power capabilities of the sensor nodes.

C. *Internal attacks*

Wormhole and sinkhole attack: The combination of those attacks cannot be easily thwarted. Some routing protocols such as Geographic routing protocols [25] seem to be more resistant in these attacks. Based on these protocols, a network topology is constructed using only localized information and each node makes independent forwarding decision based on the location of its neighbours, thus making traffic attraction not easy.

Sybil Attack: Protection against Sybil attacks can be achieved by node authentication in combination with verification [26] techniques that verify location claims. Care must be taken on the way authentication can be employed, since the use of globally shared keys offers no level of protection in cases of node compromise. One possible solution is for each node to have a unique symmetric key with a key server or a base station, as in SPINS [22], which will be used for the authentication and encryption between two neighboring nodes. In order to prevent the insider from establishing shared keys with every node, the base station can limit the number of neighbours a node is allowed to maintain. In this way, a compromised node is restricted to communicate only with its neighbours.

HELLO Flood attacks: The simplest way to prevent HELLO Floods is to use bi-directionality verification of local links before using them. In addition, every node can authenticate each of its neighbours using a trusted base station, as in the Sybil attack, before it tries to forward messages to them.

D. *Key Management mechanisms*

The major constraints of key management in the case of Wireless Sensor Networks derive from the limited computational power of the nodes, the lack of a central trusted entity (CA) and the demands for scalability and frequent topology changes. On the other hand, the main features that a key management mechanism should provide are key pre-distribution, shared key discovery, path-key establishment and re-keying mechanisms.

The use of a network wide shared key, which is the simplest key management mechanism, although offers protection against external attacks, it sustains no protection in cases of internal attacks and node compromise. Using more sophisticated key management mechanisms, a greater level of protection is ensured, but many of these mechanisms lack support for scalability and nodes mobility. For example, in pairwise key management, a pair of nodes can communicate if they have previously agreed on a pair of keys. A network wide secret key can be used for the establishment of the pair of keys. But, this mechanism doesn't scale well, since the support of new nodes requires a great number of keys to be established between nodes. However, as stated in [15], those key-management models are not still well supported by IEEE 802.15.4 specifications.

Recent research focuses on two major approaches. The first is to use probabilistic key management mechanisms [27], in which a pool of keys is shared between nodes. The model guarantees that every two nodes can share one key with a chosen probability. On the downside, the main disadvantage of this mechanism is that if the adversary compromises a great number of nodes, he will be able to discover the pool of keys used and thus gain control of the whole network. Variations of this mechanism propose either the distribution of more than one key (q keys) [28], which makes key discovery more difficult in cases of node compromise, the use of threshold secret sharing [29], or a pseudo-random key pre-deployment approach [30]. These variations improve the functionality of the probabilistic key management model.

The latter examines ways that public key management can be incorporated in wireless sensor networks, by using more efficient public-key management mechanisms. The main idea is to be able to identify the more suitable mechanisms and encryption algorithms, such as elliptic-curve codes, in order to deal with the major constraints mentioned previously. The hardware deployment of public key management is also under consideration.

V. Conclusions and future work

In the near future it is expected that sensor applications will manage to highly penetrate into the medical sector market in a great degree due to their low cost, easy administration, flexibility, etc. At the same time medical data must enjoy maximum privacy and medical premises deserve the highest security level. This paper addresses security issues for CodeBlue prototype, which is perhaps the most complete anticipated framework in this context, combining both hardware and software aspects. Five distinct categories of accustomed attacks in sensor networks were considered to put into question whether CodeBlue foreseeable implementations are secure. The associated threat analysis discloses that CodeBlue designers must cautiously consider security topics in their revised version of the CodeBlue technical report. Node authentication and message encryption can provide a high level of protection. However, it is necessary to study the most expedient approach the available security options can be integrated within the CodeBlue architecture. In every case, their employment becomes unfeasible without the proper employment of a key management mechanism.

Although this study can be seen as preliminary for CodeBlue prototype it has value for every sensor network deployed in medical premises as well. As future work, we would like to expand this work conducting proportional measurements and testing some of the discussed attack scenarios utilizing ADMR protocol and sensors integrated with Mica2 designs.

Acknowledgment

The authors would like to thank Mr. Fragkos Panagiotis and Mr. Ntais Athanasios for their contribution to this work.

References

[1] G.-Z. Yang et al. Body Sensor Network Node, http://www.doc.ic.ac.uk/vip/ubimon/bsn_node/ index.html

[2] D. White et al. AID-N: Advanced Health and Disaster Aid Network, secwww.jhuapl.edu/aidn/.

[3] L. Ohno-Machado et al. SMART: Scalable Medical Alert Response Technology, smart.csail.mit.edu/.

[4] K. V. Laerhoven, B. P. Lo, J. W. Ng, S. Thiemjarus, R. King, S. Kwan, H.-W. Gellersen, M. Sloman, O. Wells, P. Needham, N. Peters, A. Darzi, C. Toumazou, and G.-Z. Yang, Medical healthcare monitoring with wearable and implantable sensors, In Proc. of the Sixth International Conference on Ubiquitous Computing, Tokyo, Japan, September 2004.

[5] E. Shih, V. Bychkovsky, D. Curtis, and J. Guttag. Demo abstract: Continuous, remote medical monitoring, In Proc. of the Second Annual International Conference on Embedded Networked Sensor Systems, November 2004

[6] B. Lo and G. Z. Yang., Key technical challenges and current implementations of body sensor networks, In Proc. of the 2nd International Workshop on Body Sensor Networks (BSN '05), April 2005.

[7] V. Shnayder, B. Chen, K. Lorincz, Thaddeus R. F. Fulford J. and M. Welsh, *Sensor Networks for Medical Care*, Technical Report TR-08-05, Division of Engineering and Applied Sciences, Harvard University, 2005, ftp://ftp.deas.harvard.edu/techreports/tr-2005.html.

[8] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. E. Culler, and K. S. J. Pister, System architecture directions for networked sensors, in Proc. the *9th International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 93–104, Boston, MA, USA, Nov. 2000.

[9] K. Lorincz and M. Welsh. MoteTrack: A Robust, Decentralized Approach to RF-Based Location Tracking, in Proc. of the *International Workshop on Location and Context Awarenes (LoCA '05)* in conjunction with Pervasive Computing 2005, Oberpfaffenhofen, Germany, May 2005.

[10] IEEE *802.15.4—Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)*, Oct. 2003.

[11] J. G. Jetcheva and D. B. Johnson. Adaptive Demand-Driven Multicast Routing in Multi-Hop Wireless Ad Hoc Networks. In proc. of the 2001 *ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '01)*, Oct. 2001.

[12] C. Intanagonwiwat, R. Govindan, and D. Estrin, Directed diffusion: A scalable and robust communication paradigm for sensor networks, in Proc. of the *International Conference on Mobile Computing and Networking*, Aug. 2000.

[13] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, TAG: A Tiny AGgregation Service for Ad-Hoc Sensor Networks, in Proc. the *5th OSDI*, December 2002.

[14] Hartung C., Balasalle J., Han R., *Node Compromise in Sensor Networks: The Need for Secure Systems*, Technical Report CU-CS-990-05, Department of Computer Science University of Colorado, Boulder, Jan. 2005.

[15] Naveen Sastry, David Wagner. Security Considerations for IEEE 802.15.4 Networks. *WiSE'04*, Philadelphia, Pennsylvania, USA Oct. 2004.

[16] C. Karlof and D. Wagner, "Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures", Elsevier's Ad Hoc Network Journal, special issue on sensor network applications and protocols, 2002.

[17] Mika Stahlberg. Radio jamming attacks against two popular mobile networks, 2000. Helsinki University of Technology, Tik-110.501 Seminar on Network Security.

[18] Wood A., Stankovic J., and Son S., JAM: A mapping service for jammed regions in sensor networks. In *Proceedings of the 1st ACM International Conference on Emdedded Networked Sensor Systems (SenSys 2003)* (Los Angeles, Nov. 5-7). ACM Press, New York, 2003, 255-265

[19] K. Chintalapudi and R. Govindan. Localized edge detection in wireless sensor networks. In *Proceedings of the IEEE ICC Workshop on Sensor Network Protocols and Applications (SNPA)*, May 2003.

[20] 3-WS02Anthony D. Wood and John A. Stankovic. Denial of service in sensor networks. *IEEE Computer*, 35(10):54.62, October 2002.

[21] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: A survey .*Computer Networks (Amsterdam, Netherlands: 1999)*, 38(4):393.422, March 2002.

[22] Adrian Perrig, Robert Szewczyk, VictorWen, David Culler, and J. D. Tygar. SPINS: Security protocols for sensor networks. In Proceedings of *the 7th Annual International Conference on Mobile Computing and Networks MOBICOM 2001*, pages 189.199, July 2001.

[23] Deng, R. Han, and S. Mishra, "Insens: Intrusion-tolerant Routing in Wireless Sensor Networks", *23rd IEEE International Conference on Distributed Computing Systems (ICDCS 2003)*, May 2003.

[24] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin., Highly-resilient, energy-efficient multipath routing in wireless sensor networks. *Mobile Computing and Communications Review*, October 2001.

[25] G. G. Finn. Routing and addressing problems in large metropolitan-scale internetworks. Technical Report ISI/RR-87-180, ISI, March 1987.

[26] Naveen Sastry, Umesh Shankar, and David Wagner. Secure verification of location claims, In Proc of the *ACM Workshop on Wireless Security (WiSe 2003)*, September 2003.

[27] L. Eschenauer and. Gligor V, "A Key Management Scheme for Distributed Sensor Networks", In proceedings of the *9th ACM Conference on Computer and Communication Security (Wahiington D.C.)*, ACM Press, New York, 2002, 41-47.

[28] H. Chan, A. Perrig, and D. Song, "Random Key Predistribution Schemes for Sensor Networks", *IEEE Symp. Research in Security and Privacy*, 2003.

[29] S. Zhu et al., "Establishing Pair-wise Keys for Secure Communication in Ad Hoc Networks: A Probabilistic Approach",In proc. of the *11th IEEE International Conference on Network Protocols*, 2003.

[30] R. D. Pietro, L. Mancini, and A. Mci, "Efficient and Resilient key Doscovery based on Pseudo-random Key Pre-deployment", *18th International Conference on Parellel and Distributed Processing Symposium,* Apr. 2004

Second International Conference on Availability, Reliability and Security (ARES'07)
0-7695-2775-2/07 $20.00 © 2007 IEEE