

Efficient Group Key Management and Authentication for Body Sensor Networks

Sye Loong Keoh

Dept. of Info. & Systems Security, Philips Research
High Tech Campus 34, 5656 AE, Eindhoven, The Netherlands
Email: sye.loong.keoh@philips.com

Abstract—Wireless body sensor networks (BSN) are being used to continuously monitor the patient’s conditions and recovery progress. It is very important to secure the confidentiality, integrity and authenticity of the patient’s health record in such applications. In this paper, we propose a novel key distribution and management scheme that uses keychains to establish group keys for body sensor networks. This scheme caters for efficient group key update and re-keying in order to adapt to membership changes. We also present a lightweight approach to enable sensor-to-sensor authentication in the BSNs. Sensors authenticate each other by computing a Elliptic Curve Diffie-Hellman (ECDH) key between each other based on an authenticated membership broadcast received from the patient’s device. Both protocols have been implemented on Tmote Sky platform, analysed and evaluated to demonstrate their feasibility. We also shows the security analysis of the protocols using BAN Logic.

I. INTRODUCTION

Ubiquitous computing is a vision of embedding computation into the environment to provide seamless adaptability to our everyday activities. Many sensors have been deployed into the environment to gather context information from various sources in order to accurately detect user’s activities and the ambient of the environment. This significantly improves the users’ experience as the system can act intelligently according to the user’s context. In the medical domain, wearable and implementable sensors are being used to continuously monitor the patient’s conditions and recovery progress, thus forming a body sensor networks (BSN) that can interact with each other, communicate with portable processing units (i.e., PDAs) and the fixed network infrastructure. Consequently, this enables early release of patients from hospital as their conditions can be monitored at home. Healthcare personnel can also be automatically alerted to obtain assistance if the patient’s condition deteriorates.

In medical applications, it is very important to secure the confidentiality, integrity and authenticity of the patient’s health record. Physiological data gathered from the sensors must be encrypted during transmission to prevent passive eavesdropping and the data integrity must be preserved to prevent malicious tampering. Furthermore, data authenticity is needed to secure the provenance, e.g., when the glucose sensor detects that the patient’s glucose level has exceeded

the monitoring threshold, it sends an instruction to the insulin pump to adjust the insulin dosage. The insulin pump must ensure the authenticity of the instruction source. However, providing confidentiality, integrity and data authenticity to BSNs in medical applications is a challenging task as sensors and mobile devices have limited computational resources, they are not capable of performing complex cryptographic operations [1].

The security goals for a medical body sensor network establishment are first to preserve the confidentiality and integrity of the medical data that is private and confidential. As medical data is being transmitted over the wireless interface, it is susceptible to eavesdropping, and thus sensor readings must be encrypted. Given the sensors’ limited computational capabilities, the use of a shared group key between sensors and other mobile devices to perform symmetric-key encryption is desirable. However, a light weight and efficient key distribution and management scheme must be designed to ensure that sensors incur low computational and power consumption overheads by minimising the number of messages exchanged and using less computational intensive cryptographic operations.

The second security goal is to enable sensors and devices to authenticate each other directly in the BSN efficiently as the use of group keys does not provide a message’s source authenticity. In some circumstances, an actuator must authenticate the message source before it executes the action encapsulated in the message. Although devices and sensors can authenticate each other with the patient’s device acting as the trusted third party to mediate their communication, e.g., similar to the Needham-Schroeder Protocol [2]. This is not efficient as it incurs more messages to be relayed, thus consuming more power.

In this paper, we propose an efficient key distribution and key management scheme using keychains to establish a shared secret group key for BSNs. Our scheme guarantees the authenticity of keying materials for generating group keys without using public-key cryptography. Security protocols have been designed to provide membership management in conjunction with the key management. A lightweight sensor-to-sensor authentication scheme is also proposed, sensors authenticate each other using symmetric-key cryptography by proving knowledge of a derived Diffie-Hellman (DH) secret key only known between the two communicating parties.

The paper is organised as follows: Section II presents related

Part of this research was done at Dept. of Computing, Imperial College London, supported by the UK EPSRC BiosensorNet (Grant EP/C547586/1). The author would like to thank Emil Lupu and Morris Sloman for their comments and constructive feedback.

work and discusses how they do not meet the security goals of BSNs. Section III introduces the assumptions of this research, as well as some previous work on secure discovery protocol. In Section IV, we describe the novel key distribution and management scheme, while section V presents the sensor-to-sensor authentication. We describe our implementation and security analysis in Section VI. Finally, we conclude the paper with possible future work in Section VII.

II. RELATED WORK

Blinking Led Indicated Grouping (BLIG) [3] uses LEDs to group a set of sensors on a patient together, and when the identity of the patient is established, a mapping between the patient's true id and the sensor group is made. The approach uses short range communication hardware, i.e., (<0.5m) to communicate with a new sensor during discovery. A healthcare worker is needed to verify the grouping of the sensors based on a synchronised blinking behaviour using a number of LEDs. We extend the protocol to cater for key distribution and management in body sensor networks to preserve the confidentiality and integrity of the medical data in the medical domain.

The use of pre-shared symmetric key for wireless sensor networks [4], [5], [6] loads a secret-key share into each sensor node. Through some cryptographic properties, sensors can derive a secret-key among themselves using their respective key share. This approach is typically used for large scale sensor deployment for environmental monitoring and intensive computation is needed to update group key whenever there is a change in the membership.

Jiang [7] proposed a scheme that is based on self-certified keys (SCK) and Elliptic Curve Cryptography (ECC) [8], [9] to establish pair-wise keys to achieve authentication. Each sensor establishes a secret with the user based on the secret information pre-loaded by a key distribution centre (KDC). Authentication is achieved if the user is able to demonstrate knowledge of the shared secret-key with at least t sensor. It assumes that each patient's body sensor network must use a different ECC curve parameters as each network is perceived as a domain. A more lightweight scheme is needed for group communication and sensor-to-sensor authentication after the initial sensor discovery in order to reduce the computational requirements. There are various other research works that proposed security solutions based on ECC [10], [11].

SNAP [12] also proposed a scheme to establish a pair-wise key between each sensor and the base station using ECC. It requires each sensor to be attached with a biometric device to authenticate the patient and uses the shared secret to communicate with the base station. However, it does not seek to establish a group key in their architecture, and authentication between sensors is via the base station.

The group key management scheme that is based on clustering of sensor nodes [13] and uses a hierarchical scheme requires a key share to be contributed by each member of the cluster to generate a group key. This approach potentially incurs excessive overheads during key updates.

III. ASSUMPTIONS AND PREVIOUS WORK

In the medical domain, many sensors can be worn and implanted into the patient's body, e.g., an ECG sensor is used to keep track of the patient's heart condition, a SpO_2 sensor monitors oxygen saturation, and a few other medical sensors are used to observe blood pressure and body temperature. Additionally, a patient controlling device such as a gumstix¹, Personal Digital Assistant (PDA) or mobile phone is used as the portable processing unit to coordinate the communication between the wireless medical sensors. This device together with the sensors form a Body Sensor Network (BSN) that continuously monitors the patient's physiological parameters. Therefore, it is assumed that there are only a limited number of sensors attached to the patient's body, hence a BSN is a variant of relatively small wireless sensor network.

In our previous work [14], we have designed a secure discovery protocol to securely establish an association of sensors with the patient. It is based on the synchronised LED blinking pattern between the sensor and the patient's device during the association process. The healthcare worker authorises the sensor-to-patient association through observation of the synchronous LED blinking pattern. With this, it ensures that only the designated sensors are associated with the patient and only by an authorised party. For example, only nurses (or other medical staff) can attach an approved ECG sensor to monitor the heart-rate of a post-operative patient and only the associated sensor can exchange information with the patient's BSN. This prevents arbitrary ECG sensors from being associated with the patient, and ensures that other ECG sensors in the vicinity (e.g., sensors attached to other patients in the same hospital ward) cannot be mistakenly or maliciously interfere with that patient's BSN.

In addition to the sensor association, the discovery protocol establishes a shared secret-key K_{ps_i} between $sensor_i$ and patient's controller. Furthermore, the sensor also sends a DH key share, g^{s_i} to the patient's controller. In this paper, we exploit this shared secret key, K_{ps_i} to facilitate group key distribution using symmetric key cryptography (c.f. Section IV) and the sensor's DH key share, g^{s_i} to facilitate sensor-to-sensor authentication (c.f. Section V).

IV. GROUP KEY DISTRIBUTION AND KEY MANAGEMENT

Communication in the BSN must be secured, ensuring the integrity and confidentiality of messages encapsulating sensor readings, events and commands. Communication between the $sensor_i$ and the patient's controller can be encrypted using a shared secret key, K_{ps_i} which has been established in the discovery protocol. However, no keying material is shared among the sensors and routing all communication via the patient's controller would be inefficient. Furthermore, notifications of medical events are usually sent to many interested parties and network configuration changes need to be conveyed to all BSN nodes. Point-to-point notification incurs redundancy as the same message must be encrypted n times and then sent

¹www.gumstix.org

to n parties in the network. Encrypted broadcast provides an effective scheme for this constrained environment that consists of mostly sensors with scarce resources. A shared group key, G can be established to enable all parties in the BSN to communicate with each other by encrypting the readings, events and actions. This is based on the assumption that the sensors are well behaved and do not impersonate other sensors in the same BSN. This assumption is reasonable as only authenticated sensors approved by the hospital have been included in the BSN.

The key distribution and management scheme we propose has the advantage of only using symmetric-key cryptography and computation of hashes, and this reduces significantly the computational requirement on the sensor as it does not rely on public-key cryptography to distribute keying materials.

Multiple one-way hash chains can be used to generate shared group keys. One key from each hash chain is concatenated and the result is hashed to produce the group key. The group key can be renewed by advancing all keychains backward to obtain the previous key from the corresponding hash chain. The next group key is then similarly generated by hashing the concatenated keys from the chains. It is advocated that at least two hash chains are used because the group keys become vulnerable if only one hash chain is used.

This scheme provides forward secrecy as it does not reveal any information about the hash chains and the keying materials for generating the group key if the group key is compromised.

A. Establishing a Group Key, G_i

The patient's controller in the BSN is responsible for the key distribution and management as it typically has higher computational capability and already shares secret keys with each sensor. It generates k hash chains [15] (X^a, X^b, \dots, X^k), each consisting of n keys using a one-way hash function, e.g., SHA-1. A random number is generated as the initial key and the hash function is applied to the initial key to generate the next key. The next key is then hashed repeatedly for $n-1$ times to produce the hash chain.

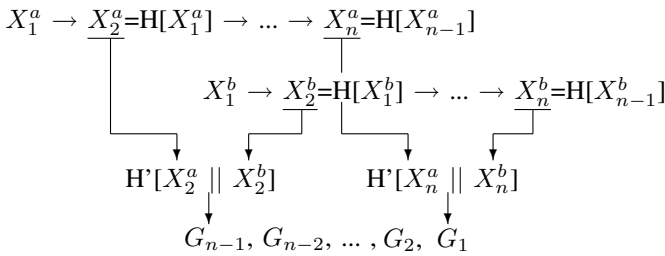


Fig. 1. Generating group keys.

All hash chains are used in reverse order where the initial key X_n^a, \dots, X_n^k are sent to the sensor. They are encrypted using K_{ps_i} , the secret-key shared between the patient's controller and the sensor. On receipt, each sensor computes the group secret key, G_i by taking the hash of the concatenation of all initial keys. This key may be changed whenever a new sensor

or device is added, but it must be renewed when an existing sensor is removed from the BSN or when the group key has been used for an extended period of time (c.f. Section IV-B). Our key management scheme provides forward secrecy in that compromising any of the G_i keys does not reveal sufficient information to compute previous/future group keys.

Figure 1 shows an example where two hash chains are used. The first chain, X^a contains keys $X_1^a, X_2^a, \dots, X_n^a$ and the second chain X^b contains $X_1^b, X_2^b, \dots, X_n^b$. Both X_n^a and X_n^b are sent to all sensors, they are concatenated and then hashed to produce the first group key, G_1 .

B. Re-Keying or Key Update

When a new sensor is discovered, the patient's controller conveys the current key of each hash chain (X_i^a, \dots, X_i^k) encrypted with the shared secret-key, K_{ps_i} to the new sensor, thus enabling the sensor to compute the current group key, G_i .

When a sensor leaves the BSN, the group key must be renewed. The patient's controller advances each hash chain backward to obtain the previous key on the chain, i.e., $X_{i-1}^a, \dots, X_{i-1}^k$. As shown in Figure 2, there are two hash chains. X_{i-1}^a is encrypted with the current X_i^a , while X_{i-1}^b is encrypted with X_i^b , they are then broadcast to all sensors/devices in a key update message. Upon receipt, the sensors compute $H[X_{i-1}^a]$ and $H[X_{i-1}^b]$ and ensure that the hash values match the current X_i^a and X_i^b respectively. This authenticates the source and contents of the key update message as only the patient's controller knows the hash chains. The sensors then derive the new group key, G_{i-1} using $H^*[X_{i-1}^a || X_{i-1}^b]$.

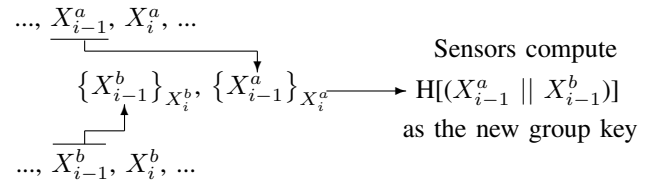


Fig. 2. Renewing the group key.

If a sensor is detected to have been compromised, a new group key must be derived. Assuming that two hash chains are used, the patient's controller broadcasts the following:

$$\{s_1, \{X_{i-1}^a, X_{i-1}^b, N_p\}_{K_{ps_1}}\}, \quad \{s_2, \{X_{i-1}^a, X_{i-1}^b, N_p\}_{K_{ps_2}}\}, \dots, \{s_n, \{X_{i-1}^a, X_{i-1}^b, N_p\}_{K_{ps_n}}\}$$

The compromised sensor's key is not used, so it will not be able to obtain X_{i-1}^a and X_{i-1}^b , but other sensors can use their respective secret-key to decrypt the message and compute the new group key. N_p is used to detect replay of the message. Detecting that a node has been compromised, is very difficult, usually based on anomalous behavior and is not covered here.

When all keys in the hash chains have been used up, the patient's controller generates new hash chains and conveys the first key of each hash chain ($X_n^{a'}, \dots, X_n^{k'}$) to all sensors. These initial keys are encrypted with each individual sensor's secret-key to ensure that they are from the patient's controller as encrypting them with the group key does not prove source authenticity.

Unlike TESLA [16] which uses hash chains for authenticated broadcast, we use hash chains to generate group keys

for encryption. This has the advantage of efficient and effective authentication of the source of key update messages when re-keying or distributing key updates without relying on public-key cryptography.

V. SENSOR-TO-SENSOR AUTHENTICATION

The use of group key is insufficient in that authentication is difficult because it is not possible to distinguish the sender of messages encrypted using the group key. Hence, it does not guarantee non-repudiation, which implies that anyone who possesses the group key can modify the message content. As a result, the group key can only be used to ensure message confidentiality in BSNs for broadcast messages such as event notifications to all sensors.

In order to enable sensors to authenticate each other, we propose a lightweight approach that requires the patient's controller to broadcast an authenticated membership list to all sensors in the BSN. With the assumption that the patient's controller is trusted, the sensor can use the information in the membership list to derive a one-time DH secret key with the sensor to which it wants to communicate with. Only both communicating parties can compute this key and no one else. Authentication is then achieved using a challenge-response protocol based on symmetric key cryptography, by requiring both parties to demonstrate knowledge of the shared DH secret key between them.

A. Authenticated Membership Broadcast

Since the patient's controller shares a secret key, K_{ps} with each sensor in the BSN, it can formulate an authenticated membership list that indicates the mapping of *sensor id* with the sensor's DH key share, g^{s_i} . Note that the patient's controller has previously obtained the DH key share, g^{s_i} of all sensors during the discovery process when they join the BSN.

$$MembershipList = \begin{pmatrix} s_1 & s_2 & \dots & s_n \\ g^{s_1} & g^{s_2} & \dots & g^{s_n} \end{pmatrix}$$

A Message Authentication Code (MAC) of the membership list is computed for each sensor using the secret key shared with the patient controller, K_{ps_i} . The MACs are then broadcast to all members of the BSN. After that, the membership list and the nonce is broadcast in clear to all sensors. This enables instant authentication of the membership list, as only the sensors that have the knowledge of their respective secret key, K_{ps_i} can verify the MAC of the membership list.

$$\{S_1, MAC(K_{ps_1}, N_p+List)\}, \{S_2, MAC(K_{ps_2}, N_p+List)\}, \dots, \{S_3, MAC(K_{ps_n}, N_p+List)\}$$

When compared this authenticated broadcast scheme with TESLA, it has the advantage of determining the message authenticity instantly and does not need to rely on time synchronisation. In addition, the sensors do not need extra storage or buffering for storing broadcast messages as in TESLA, hence distributed denial-of-service (DDoS) attack as reported as a vulnerability in TESLA can be avoided.

B. Establishing Authentication Key

With the authenticated membership list, a sensor can derive a one-time DH secret key with the other party that it wants to communicate with. For example, when S_1 wants to communicate with S_2 , it obtains g^{s_2} from the membership list and computes $g^{s_1 s_2}$ as the DH shared key with S_2 , while S_2 obtains g^{s_1} from the membership list and computes $g^{s_2 s_1}$. Note that, knowing g^{s_1} and g^{s_2} from the membership list, an attacker cannot compute $g^{s_2 s_1}$. The DH secret key is only computed once between two communicating parties, it is then used for their successive interaction. Similarly, S_1 computes $g^{s_1 s_3}$ when it wants to communicate with S_3 , etc.

Authentication between two sensors is performed using the standard challenge-response protocol, i.e., symmetric key cryptography. S_1 sends a challenge, typically a nonce encrypted with the DH secret-key, $g^{s_1 s_2}$ to S_2 . If S_2 is able to respond with the correct nonce+1, it proves that S_2 indeed has knowledge of $g^{s_1 s_2}$ and hence authenticating S_2 .

VI. IMPLEMENTATION AND SECURITY ANALYSIS

A. Implementation

We have implemented the key management and authentication protocols on Tmote Sky².

The Diffie-Hellman (DH) key exchange has been implemented using Elliptic Curve Cryptography (ECC), TinyECC version 0.3 [8]. We used the recommended 160-bit Elliptic Curve domain parameters over F_p associated with verifiably random parameters, i.e., *secp160r1*. A wide range of parameters can be selected from [17] and a base point G is chosen.

Symmetric key cryptography is realized using Skipjack algorithm. We modified the MicaZ specific Skipjack algorithm implemented in TinySec [18] for use in Tmote Sky. The hash function SHA-1 produces 160-bit output which fits nicely into the key size of Skipjack. We used the Skipjack Cipher Block Chaining (CBC) mode with a block size of 8 bytes and non-repeating Initialisation Vector (IV). The accelerometer reading of the mote has been used as the seed to generate the initial IV to produce different ciphertext from the same plaintext.

TABLE I
LATENCY FOR VARIOUS SECURITY OPERATIONS ON TMOTE SKY

Security Operations	Time
SHA-1	0.015 s
Skipjack Encryption	150 μ s
Skipjack Decryption	90 μ s
Diffie-Hellman Key Generation	5.97 s

Table I shows the execution time of various security operations on Tmote Sky. The SHA-1 takes 0.015 s and as we know, symmetric-key encryption is significantly faster than public-key encryption, the Skipjack encryption takes 150 μ s and decryption takes 90 μ s. The key management scheme also uses Skipjack for key distribution, key renewal and key

²Runs TinyOS and has 16-bit, 8 MHz Texas Instruments MSP430 processor with 48 KB of ROM and 10 KB of RAM

updates. The DH key generation is the most computationally intensive operation used in sensor-to-sensor authentication and it takes 5.97 s. In terms of the code size, the hash function SHA-1 uses 2,442 bytes [8] and Diffie-Hellman uses 6.92 Kb of ROM and 1.12 Kb of RAM.

B. Security Analysis

The security goal of key management is to ensure the source authenticity of keying materials, thus guaranteeing the authenticity of group keys. As keying materials are encrypted, attackers who do not have access to the hash chains (e.g., X^a and X^b) cannot decrypt the key update messages. Using BAN Logic [19], we prove that:

- Sensors believe that the initial or current keying materials (e.g., X_i^a , X_i^b , ..., X_i^k) from the patient's controller are authentic when joining the BSN.
- When performing key update, existing sensors in the BSN believe that the keying materials broadcast to all sensors are authentic to generate the new group key. Hence believing in the new group key, G_i .

Similarly, we used BAN Logic to show that the DH secret-key established between two communicating sensors are authentic and they mutually believe in their DH shared key.

1) *Belief in the Initial/Current Keying Materials:* When a new sensor or actuator is added to the BSN, the patient's controller sends a nonce together with the initial or current keying materials (X_i^a , X_i^b , ..., X_i^k) to the sensor using the secret key, K_{ps} shared between them. Assuming that two hash chains (X^a , X^b) are used, the idealised protocol is as follows:

$$P \rightarrow S_i: \left\{ N_p, (P \stackrel{X_i^a}{\equiv} S_i), (P \stackrel{X_i^b}{\equiv} S_i) \right\}_{K_{ps_i}}$$

Since it's the new sensor S_i , that generated the secret key, K_{ps_i} and shared it with the patient's controller P , we assume that S_i believes in K_{ps_i} , hence it can decrypts the message sent by P . Using the *message meaning rule*, we derive that S_i believes that P once said the keying materials, X_i^a and X_i^b .

$$\frac{S_i \equiv P \xleftrightarrow{K_{ps_i}} S_i, S_i \triangleleft \left\{ (P \stackrel{X_i^a}{\equiv} S_i), (P \stackrel{X_i^b}{\equiv} S_i) \right\}_{K_{ps_i}}}{S_i \equiv P \sim \{ (P \stackrel{X_i^a}{\equiv} S_i), (P \stackrel{X_i^b}{\equiv} S_i) \}}$$

Further, using the *nonce verification rule* that the message received by S_i is fresh, we derive that S_i believes that P believes in X_i^a and X_i^b .

$$\frac{S_i \equiv \{ \#(P \stackrel{X_i^a}{\equiv} S_i), \#(P \stackrel{X_i^b}{\equiv} S_i) \}, S_i \equiv P \sim \{ (P \stackrel{X_i^a}{\equiv} S_i), (P \stackrel{X_i^b}{\equiv} S_i) \}}{S_i \equiv P \equiv \{ (P \stackrel{X_i^a}{\equiv} S_i), (P \stackrel{X_i^b}{\equiv} S_i) \}}$$

With the assumption that S_i believes that P has *jurisdiction* over the keying materials, we conclude that S_i believes in X_i^a and X_i^b .

$$\frac{S_i \equiv P \equiv \{ (P \stackrel{X_i^a}{\equiv} S_i), (P \stackrel{X_i^b}{\equiv} S_i) \}}{S_i \equiv \{ (P \stackrel{X_i^a}{\equiv} S_i), (P \stackrel{X_i^b}{\equiv} S_i) \}}$$

The sensor S_i which believes in X_i^a and X_i^b computes the group key, $G_i = \text{H}[(X_i^a \parallel X_i^b)]$, hence S_i believes in G_i too, $S_i \equiv G_i$.

2) *Belief in the New Group Key during Key Update:* The group key is renewed during the key update. The patient's controller P broadcasts the new keying materials encrypted with the current key of each hash chain to all sensors. Assuming that two key chains (X^a , X^b) are used, the idealised protocol is as follows:

$$P \rightarrow S: \left\{ N_p, (P \stackrel{X_{i-1}^a}{\equiv} S) \right\}_{X_i^a}, \left\{ N_p, (P \stackrel{X_{i-1}^b}{\equiv} S) \right\}_{X_i^b}$$

Although the keying materials are encrypted using X_i^a and X_i^b , which means that any member of the BSN could have produced this message, the sensor has a strong belief that X_{i-1}^a and X_{i-1}^b are originated from the patient's controller P . This is because only P can produce the correct X_{i-1}^a and X_{i-1}^b , when hashed matches X_i^a and X_i^b respectively. Therefore, S believes that P once said X_{i-1}^a and X_{i-1}^b .

Using the *nonce verification rule*, the key update message received by S is fresh, we derive that S believes that P believes in X_{i-1}^a and X_{i-1}^b . Using the *jurisdiction rule*, we derive that S believes in X_{i-1}^a and X_{i-1}^b and hence can compute the new group key, G_{i-1} . The corresponding BAN Logic derivations are illustrated in Figure 3.

3) *Sensor-to-sensor Authentication:* In this section, we show that the sensor believes in the membership list broadcast by the patient's controller. We also prove that a shared secret key can be securely computed between two communicating sensors to perform authentication. The idealised protocol of the authenticated membership list broadcast is as follows, where the MAC of the membership list and nonce is computed using the patient-sensor secret-key, K_{ps_i} :

$$P \rightarrow S_i: \{ N_p, MembershipList \}_{K_{ps_i}}$$

As the sensor shares a secret key, K_{ps_i} with the patient's controller, and using the *message meaning rule*, we derive that sensor S_i believes that P once said the membership list.

$$\frac{S_i \equiv P \xleftrightarrow{K_{ps_i}} S_i, S_i \triangleleft \{ MembershipList \}_{K_{ps_i}}}{S_i \equiv P \sim MembershipList}$$

Further, using the *nonce verification rule*, the membership list received by S_i is fresh, we derive that S_i believes that P believes in the membership list.

$$\frac{S_i \equiv \#(MembershipList), S_i \equiv P \sim MembershipList}{S_i \equiv P \equiv MembershipList}$$

As S_i believes that P has *jurisdiction* over the membership list, we conclude that S_i believes in the membership list.

The following shows the use of *Nonce Verification Rule* to derive that S believes that P believes in X_{i-1}^a and X_{i-1}^b .

$$\frac{S \models \{\#(P \stackrel{X_{i-1}^a}{\equiv} S), \#(P \stackrel{X_{i-1}^b}{\equiv} S)\}, S \models P \vdash \{(P \stackrel{X_{i-1}^a}{\equiv} S), (P \stackrel{X_{i-1}^b}{\equiv} S)\}}{S \models P \models \{(P \stackrel{X_{i-1}^a}{\equiv} S), (P \stackrel{X_{i-1}^b}{\equiv} S)\}}$$

The following shows the use of *Jurisdiction Rule* to derive that S believes in X_{i-1}^a and X_{i-1}^b .

$$\frac{S \models P \Rightarrow \{(P \stackrel{X_{i-1}^a}{\equiv} S), (P \stackrel{X_{i-1}^b}{\equiv} S)\}, S \models P \models \{(P \stackrel{X_{i-1}^a}{\equiv} S), (P \stackrel{X_{i-1}^b}{\equiv} S)\}}{S \models \{(P \stackrel{X_{i-1}^a}{\equiv} S), (P \stackrel{X_{i-1}^b}{\equiv} S)\}}$$

Fig. 3. Belief in the new group key during key update.

$$\frac{S_i \models P \Rightarrow \text{MembershipList}, S_i \models P \models \text{MembershipList}}{S_i \models \text{MembershipList}}$$

Finally, through the belief in the membership list that is originated from a trusted source, the sensor also believes in the DH key shares of all the members in BSN. Therefore, they can compute a DH secret key with the party that it wants to communicate with. In this paper, we do not reproduce the proof of challenge-response protocol using BAN Logic.

VII. CONCLUSIONS AND FUTURE WORK

We have presented a novel key distribution and management scheme based solely on hash chains, hash functions and symmetric-key encryption; it is thus very efficient and has low computational overheads. Distribution of group keys is simplified and key update messages can be authenticated easily thus facilitating efficient renewal of group keys to cater for membership changes.

We've also identified that the use of group keys is not sufficient to cater for sensor-to-sensor authentication, hence posing difficulty in determining the message source authenticity. We have devised a simple and lightweight authentication scheme that is based on authenticated broadcast of sensors' DH keyshares by the patient's device. This enables two communicating sensors to derive a DH secret key between themselves. Authentication between them is based on the ability to demonstrate knowledge of the DH secret-key.

Further performance analysis work are required to explore protocol trade-offs. Mitigating DoS attacks aimed at depleting resources and investigating mechanisms to detect anomalous behaviour is also required. Finally, we would like to extent our work to enable access control, in particular the ability to deploy and enforce access control policies.

REFERENCES

- [1] H. S. Ng, M. L. Sim, and C. M. Tan, "Security issues of wireless sensor networks in healthcare applications," *BT Technology Journal*, vol. 24, pp. 138–144, April 2006.
- [2] R. Needham and M. Schroeder, "Using encryption for authentication in large networks of computers," *Commun. ACM*, vol. 21, no. 12, pp. 993–999, 1978.
- [3] J. Andersen and J. E. Bardram, "BLIG: A New Approach for Sensor Identification, Grouping, and Authorisation in Body Sensor Networks," in *Proc. 4th Int. Workshop on Wearable and Implantable Body Sensor Networks, Aachen, Germany, March 26 - 28, 2007*.
- [4] D. Liu, P. Ning, and R. Li, "Establishing Pairwise Keys in Distributed Sensor Networks," *ACM Trans. on Information and System Security*, vol. 8, no. 1, pp. 41 – 77, Feb 2005.
- [5] D. Liu, P. Ning, and W. Du, "Group-Based Key Pre-Distribution in Wireless Sensor Networks," in *Proc. ACM Workshop on Wireless Security (WiSe 2005)*, 2005, pp. 11–20.
- [6] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in *Proc. 9th ACM Conf. on Computer and communications security*. Washington, DC, USA: ACM, 2002.
- [7] C. Jiang, B. Li, and H. Xu, "An Efficient Scheme for User Authentication in Wireless Sensor Networks," in *Proc. 21st Int. Conf. on Advanced Information Networking and Applications Workshops*. Washington, DC, USA: IEEE Computer Society, 2007.
- [8] A. Liu, P. Kampanakis, and P. Ning, "TinyECC: Elliptic Curve Cryptography for Sensor Networks (Version 0.3)," Feb 2007. [Online]. Available: <http://discovery.csc.ncsu.edu/software/TinyECC/>
- [9] H. Wang, B. Sheng, and Q. Li, "TelosB Implementation of Elliptic Curve Cryptography over Primary Field," Dept. of Computer Science, College of William and Mary, Tech. Rep. WM-CS-2005-12, October 2005.
- [10] C. C. Tan, H. Wang, S. Zhong, and Q. Li, "Body sensor network security: an identity-based cryptography approach," in *Proc. of 1st ACM conference on Wireless network security*. New York, NY, USA: ACM, 2008, pp. 148–153.
- [11] R. Watro, D. Kong, S. fen Cuti, C. Gardiner, C. Lynn, and P. Kruus, "TinyPK: securing sensor networks with public key technology," in *Proc. 2nd ACM workshop on Security of ad hoc and sensor networks*. Washington DC, USA: ACM, 2004, pp. 59–64.
- [12] K. Malasri and L. Wang, "Addressing security in medical sensor networks," in *Proc. 1st ACM Int. Workshop on Systems and Networking Support for Healthcare and Assisted Living Environments*. San Juan, Puerto Rico: ACM, 2007.
- [13] E. Klaoudatou, E. Konstantinou, G. Kambourakis, and S. Gritzalis, "Clustering Oriented Architectures in Medical Sensor Environments," in *Proc. 3rd International Conference on Availability, Reliability and Security (ARES), 4-7 March*, 2008, pp. 929 – 934.
- [14] S. Keoh, "Security proof of secure discovery protocol for body sensor networks," Aug 2008. [Online]. Available: <http://www.doc.ic.ac.uk/slk/files/bsn-proof.pdf>
- [15] L. Lamport, "Password Authentication with Insecure Communication," *Communications of the ACM*, vol. 24, no. 11, pp. 770–772, 1981.
- [16] A. Perrig, R. Canetti, J. Tygar, and D. Song, "The TESLA Broadcast Authentication Protocol," *RSA Cryptobytes*, 2002.
- [17] "SEC 2: Recommended Elliptic Curve Domain Parameters," Standards for Efficient Cryptography (SEC), Certicom Research, Tech. Rep., September 2000. [Online]. Available: http://www.secg.org/collateral/sec2_final.pdf
- [18] C. Karlof, N. Sastry, and D. Wagner, "TinySec: A Link Layer Security Architecture for Wireless Sensor Networks," in *Proc. 2nd Int. Conf. on Embedded Networked Sensor Systems*. Baltimore, MD, USA: ACM, 2004, pp. 162–175.
- [19] M. Burrows, M. Abadi, and R. Needham, "A logic of authentication," *ACM Trans. Comput. Syst.*, vol. 8, no. 1, pp. 18–36, 1990.