Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2015, Article ID 272869, 13 pages
http://dx.doi.org/10.1155/2015/272869

Research Article

A Cognitive-Radio-Based Method for Improving Availability in Body Sensor Networks

Olga León,[1] Juan Hernández-Serrano,[1] Carles Garrigues,[2] and Helena Rifà-Pous[2]

[1] *Telematics Department, Technical University of Catalonia, 08034 Barcelona, Spain*
[2] *IT, Multimedia and Telecommunications Department, Open University of Catalonia, 08018 Barcelona, Spain*

Correspondence should be addressed to Olga León; olga@entel.upc.edu

Received 11 July 2015; Accepted 20 September 2015

Academic Editor: Gyanendra P. Joshi

Copyright © 2015 Olga León et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

One of the main threats to body sensor networks (BSNs) is Denial of Service attacks that disrupt communications used to transmit patients' health data. The application of cognitive radio (CR) technology into BSNs can mitigate such a threat and improve network availability, by allowing network nodes to cooperatively agree on a new radio channel whenever the quality of the channel being in use decreases. However, the cooperative spectrum sensing mechanisms used by CRs should also be protected to prevent an attacker from predicting the new channel of operation. In this work, we present a lightweight and robust mechanism that appropriately secures the channel selection process while minimizing resources consumption, thus being suited for resource constrained devices such as body sensor nodes. The proposed method has been analyzed in terms of energy consumption and transmission overhead and it has been shown that it outperforms existing cryptographic approaches.

1. Introduction

Sensor and wireless communication technologies are rapidly evolving and spreading to many fields, such as medical services. Body sensor networks (BSNs) [1, 2] are becoming more popular and powerful every day and ongoing efforts, such as the IEEE 802.15.6 standard optimized for low-power BSN devices [3], clearly reflect the increasing importance and potential of these types of networks.

A typical BSN is composed of a number of sensors that are placed at various locations on the body or in body, also known as implantable medical devices (IMDs). As depicted in Figure 1, these sensors forward sensed data to a more computationally powerful device or gateway (e.g., a smartphone) that, in its turn, can transmit the gathered data to a medical center. Therefore, the professionals can constantly monitor the patient's state and take the proper actions according to the observed data. Thus, the use of BSNs can considerably reduce the gap between a medical emergency and the medical response while increasing the autonomy of patients, that is to say, their quality of life.

Body sensors exhibit more constraints regarding size, power, battery availability, and transmission (i.e., the human body is a lossy medium) than those sensors that can be found in conventional wireless sensor networks (WSNs) and, therefore, they require specific solutions. Besides the recent IEEE 802.15.6 standard, already supported by a few commercial devices, several low-power wireless technologies [4–7] suitable for BSNs have emerged during the last years. These technologies define typical transmission rates ranging from several kbps in ANT+ to 6 Mbps in WiFi with the lowest power 802.11b mode.

Lately, there has been increasing concern in incorporating security and privacy mechanisms to medical systems in order to preserve patients' privacy and offer continuous monitoring of their health status. Besides, FDA (Food and Drug Administration) made recently a call for manufacturers to address cybersecurity issues relevant to medical devices for the entire life cycle of the device [8]. Thus, it is expected that these facts will definitely encourage a number of works in this field.

Generally speaking, the following security services should be provided in any medical system.

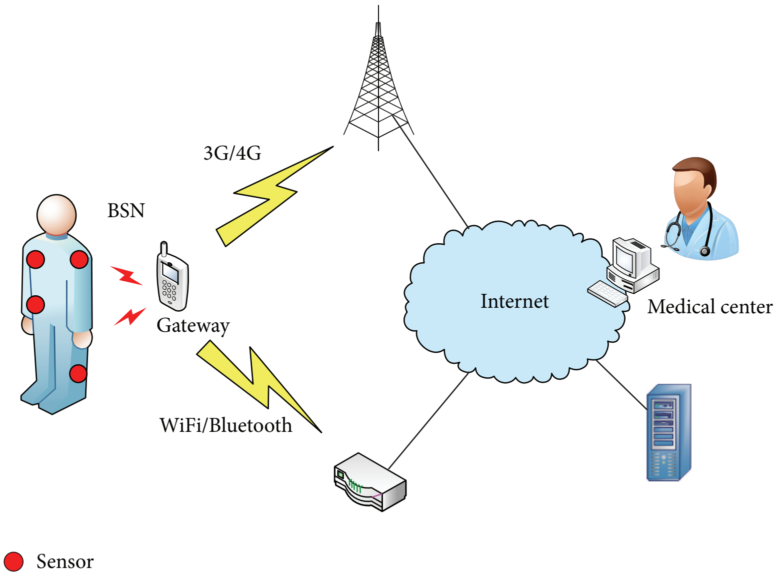


Figure 1: BSN model.

Confidentiality. Data regarding patients' state should be only accessible to authorized entities. In this context, this implies that only the BSN's nodes should be able to interpret the sensed data.

Authentication. The BSN's nodes should be able to verify the source of any received data.

Integrity. Data should not be modified by an unauthorized entity, or at least BSN's nodes should be able to detect that data has been altered.

Availability. Data and device information should be accessible upon request by authorized entities. The human body is a highly dynamic physical environment where wireless channel properties constantly change. Besides, these communications can be severely affected by interferences caused by electronic devices in the proximity of the BSN.

The first three security goals can be easily achieved by means of classical cryptographic tools in conventional networks. However, the limited capabilities of body sensors may prevent from applying them to BSNs. Besides, traditional cryptographic tools cannot prevent disruption of the network services due to interferences, no matter whether they are intentional, for example, a Denial of Service (DoS) attack, or not. Given the relevance of the data sent by body sensors, there is clearly a need for mechanisms to maximize the availability of such networks.

The integration of CR technology [9–11] into BSNs, leading to the concept of cognitive body sensor networks (CBSNs) [12], can significantly improve availability by allowing the nodes to select the best channel at any moment and avoid the harmful effect of interferences. CRs exchange sensed data about channel availability and jointly agree to switch to a new channel when the channel being in use becomes unavailable.

Note that if an attacker manages to eavesdrop channel availability data, it can take advantage of it to perform a new attack on the new channel of operation, thus preventing the network from using an available channel and leading the network to a DoS [13]. Channel switching, if unpredictable, renders DoS attacks more difficult since the attacker must jam every possible transmission channel. Traditional encryption and authentication of exchanged data may help to hide channel switching decisions from external attackers but entail an additional cost that cannot be assumed by heavily constrained devices such as IMDs.

In this paper we present a protocol to protect the process of channel selection in CR-based BSNs. The main goal is to maximize the availability of the network, thus ensuring that patients' data such as blood pressure, heart rate, and temperature will successfully be delivered to a gateway (nonstop monitoring of patients). The protocol makes use of lightweight encryption and authentication primitives specifically suited for constrained devices such as body sensors.

The main contributions of this paper can be summarized as follows.

(i) We apply CR technology into BSNs in order to maximize the availability of services in such networks. Because CRs are able to sense the medium and select the best transmission channel at any moment, the effect of interferences or DoS attacks can be mitigated. In a conventional network, such phenomena would interrupt communications within the BSNs. In a CBSN, the nodes can switch to a new channel whenever the channel in use becomes unavailable.

(ii) We propose a method, suited to constrained devices as body sensors, to secure the exchange of channel availability information and prevent an attacker from eavesdropping such data, thus diminishing the probability of a successful DoS attack.

(iii) We provide a security analysis of the proposed method and derive the time period during which the cryptographic material remains secure.

(iv) The proposed method is compared to other approaches based on traditional cryptographic primitives in terms of energy consumption and CPU usage.

The rest of this document is structured as follows. In Section 2 we review the state of the art on security in BSNs. Section 3 describes the BSN model considered in this work and its potential threats. A lightweight method to secure the process of channel selection in a BSN is presented in Section 4. Sections 5 and 6 present a security analysis of the proposed method and a comparison with existing approaches in terms of resources consumption. Finally, in Section 7 we provide the conclusions of this work.

2. Related Work

To date, research on security in BSNs has mainly focused on protecting data stored at the network nodes from unauthorized access and providing authentication and confidentiality to the communications among the BSN devices. In the following, we provide an overview of the proposals that can be found in the literature.

Many proposed authentication methods are based on biometrics, that is, relying on measurements of physiological values (PVs) [14], such as heart rate, blood pressure, or temperature, in order to establish trust and generate key material. The main idea is ensuring access to sensors only to those devices in physical contact with the patient. The advantage of these methods is that the key source is hard for an attacker to predict without physical access to the patient and also ensures forward-security, because PVs change over time. The main challenge, however, is how to achieve successful authentication among authorized devices when the PV measured by each one is not exactly the same, either due to measurement errors or due to the fact that different devices measure a given PV at different time instants.

Authentication by means of distance-bounding protocols was proposed in several works [15, 16]. This technique provides a very weak mutual authentication between two devices based on measuring the transmission time between them. The rationale behind these protocols is that a legitimate device must be closer than a given distance. As a consequence, they are vulnerable to injection attacks as long as the attacker is close enough to the patient bearing the sensors, for example, by means of a hug.

In [17], the authors presented a protocol based on identity-based encryption (IBE). IBE systems are public key cryptosystems that allow any device to generate a public key from a known identity value such as the sensor ID and require the existence of a trusted third party called the private key generator (PKG) to generate the corresponding private key. To reduce the burden of key generation and encryption/decryption introduced by traditional public key cryptography, the authors proposed to use elliptic curve cryptography (ECC), which provides public key primitives suitable for constrained devices as sensors in BSNs. Despite it, it is still more expensive in terms of resource consumption than approaches based on symmetric cryptography.

In order to preserve user's privacy, a number of works proposed the use of symmetric encryption based on the AES (Advanced Encryption Standard) algorithm [18–20]. Many, such as the one in [19], proposed to use AES with CCM mode of operation, that is to say, AES counter (CTR) mode for data encryption and AES cipher-block-chaining message authentication code (CBC-MAC) for message authentication. The main advantage of this mode is that the same key can be used for authentication and for encryption without compromising security and there is no need for rekeying as long as the number of devices is fixed. As a drawback, the added cost of encryption/decryption and, especially, the costs due to the transmission overheads cannot be neglected in BSNs, where every step forward in resources' saving is of paramount importance. In this line, the authors in [20] presented an in-network mechanism that mimics the AES algorithm and greatly reduces the costs of decryption while they claim achieving the same level of security.

All the above-mentioned proposals approach the problem of protecting patient's data from unauthorized access, modification, or forgery but cannot effectively deal with DoS attacks. Such a protection can be achieved by making use of CR devices that collaboratively switch to another frequency band [11, 21, 22] if the signal-to-noise ratio of the current one is below the required value. Furthermore, it is also necessary to protect the exchanged sensing data in order to prevent an attacker from eavesdropping data and get the next channel to be used in the network. Note that this information may allow an attacker to rapidly perform a DoS attack in the new channel.

In this work, we present a lightweight and secure method that makes use of CR technology for improving the availability of the system, that is, ensuring that the communication between the body sensor nodes will be available even under the presence of unintentional or intentional interferences. The application of CR technology into body sensor networks was already proposed in previous works [23, 24]. However, to the best of our knowledge, none of them addressed security topics.

In [25, 26], several methods for securing spectrum sensing mechanisms were discussed, but they are not suited for heavily constrained devices such as body sensors.

In [27], the authors aimed to improve the availability of a BSN by means of a cross-layer multihop protocol that dealt with routing of data. This scheme, however, can be applied only to multihop BSNs where the path between two given nodes is established according to the connectivity among the nodes. In this approach, nodes make use of several paths but one single channel and thus are more vulnerable to attacks such as jamming than CR-based networks.

3. Network Model and Threats

In this work, we have considered a BSN composed of a set of sensor nodes where all of them can act as sinks, collecting/storing data from other sensors and potentially

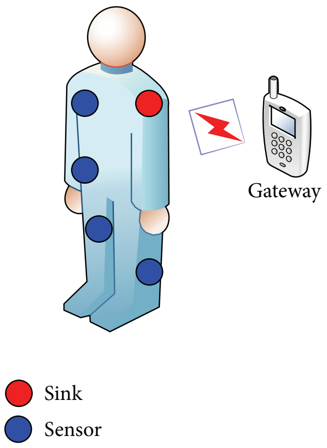


Figure 2: Communication between sensors and the gateway.

transmitting these data to an external gateway if required (see Figure 2). Although this approach introduces some overhead due to the fact that data must be shared among all sensors, it improves network availability and robustness against data loss and fairly distributes energy consumption among all sensors. Also, it makes the process of gathering by the gateway easy, which can connect to any of the BSN nodes to get all the information.

As previously mentioned, we also assume that sensors have cognitive capabilities; that is, they form a CBSN and are able to identify free spectrum bands and adapt their transmission parameters accordingly. Spectrum sensing can be performed by each node on an individual basis or cooperatively. As the latter increases the probability of detection due to space diversity [28], we have adopted such an approach in this work.

In cooperative spectrum sensing, each sensor senses the medium and exchanges its observations with the other members of the network in order to agree on a given channel for data transmission/reception. However, these control data are exposed to many attacks [13], such as packet injection, eavesdropping, or Denial of Service (DoS). Next, we describe the attacker model and the specific attacks that can be executed against CBSNs.

3.1. Attacker Model. In this work we focus on outsiders, that is, external attackers that do not share any cryptographic content with the gateway or the victim's sensor nodes. If the attacker nodes are part of the CBSN, they will have access to the keying material and therefore will be able to successfully eavesdrop and inject data. In any case, the design of a mechanism to counteract this threat is out of the scope of this work.

In the context of CBSNs, we can classify adversaries according to the following criteria:

(i) *Active or passive*: a passive attacker can only eavesdrop data, thus being able to access patient's data and violating his/her privacy. In its turn, an active attacker aims at injecting or modifying data in order to send fake reports on the state of the patient.

(ii) *Type of attack* includes the following:

  (a) eavesdropping: unauthorized access to stored data or to transmitted data among the CBSN devices, thus violating the privacy of the patient,

  (b) modification/injection: an attacker that may alter the content of a packet transmitted by a sensor or impersonate a sensor by forging a packet; these attacks can be executed due to lack of authentication and violate the integrity of the CBSN communications,

  (c) packet replay: an attacker that may capture a packet that was previously sent by a sensor of the network. Regardless of the fact that the CBSN is using authentication mechanisms or not, the packet will be accepted by the networks if antireplay mechanisms are not provided,

  (d) jamming: the adversary that disrupts the CBSN communications by generating interfering signals.

(iii) *Intentional or unintentional*: the adversary can be an external entity willing to cause damage to the communications among sensors and the gateway or can be an entity that unintentionally is causing interferences to those communications. As an example, the patient of interest could be near another patient with wearable sensors, which could inject fake reports if data is not properly authenticated. Examples of unintentional attacks could take place in a situation where two patients bearing body sensors are hugging and unconsciously exchange data. Or the patient could be near a relative who is visiting him/her at the hospital and carries any electronic device that causes interferences to the CBSN.

It is important to remark that, in a CBSN where sensor nodes establish communications using different channels over time, these attacks can be extended to the control data exchange among the devices of the CBSN. As an example, an attacker may forge a report regarding the availability of the channels, thus leading the CBSN to select a channel that is suffering from high interferences or that is currently being used by another service. Note that this attack can lead to a DoS and the failure of the system in monitoring the patient's status. In its turn, eavesdropping of the control channel allows an attacker to have knowledge of the channels to be used by the CBSN. The attacker could take advantage of this situation in order to easily disrupt the communications in the network by performing a new DoS attack every time the CBSN switches to a new channel.

The implementation of security mechanisms in a CBSN [12] to counteract these attacks is specially challenging due to the limited capabilities of CBSN's nodes. In the following section, we describe a simple method to secure the process of channel selection in CBSNs. The proposed mechanism is suited for networks with extremely constrained-resources devices, since it makes use of lightweight cryptographic functions and minimizes the added transmission/reception overhead.

4. Securing Sensing Data and Channel Selection in CBSNs

In the following we present a mechanism for securing the exchange of sensing data and the channel selection process in CBSNs. Section 4.1 outlines the assumptions considered in this work regarding the network model and, in Section 4.2, we describe the protocol operation. For ease of understanding, we present the terminology used along this section as follows:

CTR_M: medium-term session counter (m bits),

CTR_S: short-term session counter (m bits),

D_i^u: data sensed by node u during period i (l bits),

ID^u: link-layer identifier of node u (m bits),

K_i^u: keystream to encrypt and authenticate data for node u during period i (r bits),

KM: keying master,

l: length of the data sensed by a given node during a given period,

m: length of the hash output and all the secrets,

N: number of nodes in the network,

p: number of keystreams K_i^u obtained from a S^u ($p = m/r$), defining the number of sensing periods before updating S^u,

r: length of the keystreams K_i^u, which must be a divisor of m,

S_L: long-term globally shared secret (m-bits),

S_M: medium-term globally shared secret (m-bits),

$S_{s,i}$: long-term secret shared between the KM and node i; it is used to update S_L in case it is compromised,

S^u: short-term shared secret with node u.

4.1. Assumptions.

4.1. Assumptions. Although the proposed protocol is designed to be implemented in heavily constrained devices, we work under the assumption that such devices have at least the following capabilities:

(i) Compute a hash function with an output length of m bits.

(ii) Temporally store in its random access memory at least $m \cdot (N + 3)$ bits, with N the number of nodes in the network. As we detail later in Section 4.2, each node must keep a short-term shared secret for each of the N nodes in the network (including itself) and three more long-term and medium-term secrets, each one with length of m bits.

(iii) Sensor nodes use a synchronization protocol that will be used to share a global short-term session counter and a medium-term session counter among all nodes (see Section 4.2). Given the low transmission rate of sensor networks, existing synchronization schemes [29] provide enough precision for this purpose. We assume that the chosen protocol provides recovery methods upon loss of synchronization. How synchronization is achieved will strongly depend on the chosen protocol, but if the latter requires a master node for providing synchronization, the gateway of the BSN could play this role.

To the best of our knowledge, the former requirement can be assumed even in very constrained devices. As shown in [30], there are several lightweight hash functions that can be integrated into a sensor mote. The latter may not be harder to achieve. As detailed in Section 4.2, during every sensing period each node stores one secret per member of the network, a globally shared secret, and two counters, all of them with the same length m as the hash output. If we consider a typical hash function with an output of 128 or 256 bits and a network with tens to hundreds of sensors, the RAM requirements for sensor nodes are just bounded to a few tens of kilobytes.

4.2. Protocol Operation. Before deploying the CBSN, every sensor node must be preloaded by a keying master (KM) with the following data:

(1) The set of channels that the sensor will have to sense in the cooperative sensing process.

(2) A long-term and globally shared secret S_L of m bits (the hash output length).

(3) A long-term secret $S_{s,i}$ shared between the KM and node i that will be used to update the globally shared secret S_L in case it is compromised.

The KM is an external device, which is not a member of the BSN. Typically, this role is played by the device responsible for gathering data from the sensors or gateway (e.g., a smart phone or a tablet).

Upon deployment of the network, every node derives a medium-term globally shared secret S_M by hashing the XOR of the long-term secret S_L and a counter. The generation process of S_M is clearly depicted in Figure 3. This process is periodically repeated with an updated value of the medium-term counter in order to protect the secret against a potential attacker. Details about how often this process should be carried out and the attacker capabilities are provided in Section 5.

As shown in Figure 4, each node generates a set of random sequences of m bits: one for the node itself and one for each other node in the network. These random sequences, S^u, with u the node identifier, are obtained by hashing the XOR of the link-layer identifier of the node ID^u, the medium-term shared secret S_M, and a short-term session counter CTR_S.

Therefore, our proposal makes use of three types of shared secrets:

(i) A short-term per-node shared secret S^u (one per each node in the network) used for encryption, decryption, and authentication of data.

(ii) A medium-term globally shared secret S_M that is used to derive the short-term per-node shared secrets S^u.

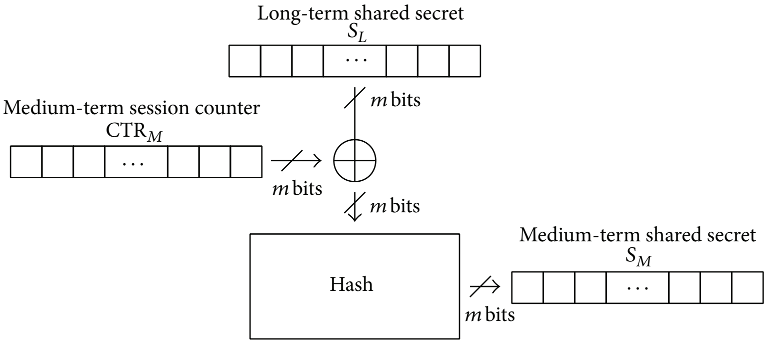


Figure 3: Generation of the medium-term globally shared secret S_M.

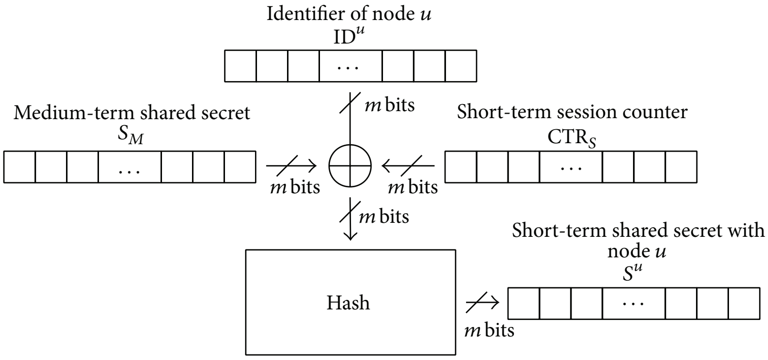


Figure 4: Generation of the short-term shared secret S^u for node u.

(iii) A long-term globally shared secret that is used to derive a new S_M when the current one is about to expire.

As clearly denoted in Figure 5, each sequence S^u is divided into p fragments of r bits, which we will be denoted as K_i^u, each one being used as keystream to encrypt and authenticate data for node u in period i. As per this behavior, a new short-term shared secret must be derived every p sensing period.

When a node performs spectrum sensing, it generates a binary sequence D_i^u of l bits that stores the availability of the different channels. The length of such sequence l will depend on the number of bits used to code the state of each channel and the number of channels. As an example, the simplest way would be to use just a single bit for coding each channel, with value “0” if the channel is occupied and “1” otherwise. If more precise information about the quality of channels is needed (i.e., high, medium, low, and very low quality), more bits can be used to code the channel state.

During a sensing period i, each node must send to its neighbors its own sensing information but also it must process the information received from its neighbors to reach a joint decision.

In order to send its own sensing information, node u will make use of the corresponding keystream K_i^u: the first l bits of the keystream will be used to encrypt channel information D_i^u by means of a XOR addition; the remaining $r-l$ bits are left unchanged and will be used to provide message authentication, as illustrated in Figure 5. The resulting sequence C_i^u will be sent to all the other nodes.

To verify the authenticity and decrypt the content of the packets that have been sent by a given neighbor u, a node will XOR the sequence C_i^u of the received packet with the keystream K_u^i, as depicted in Figure 6. If the last $r-l$ bits of the resulting sequence are not all 0 s, the authentication fails and the entire packet is discarded. Otherwise, channel information can be recovered from the first l bits resulting from the XOR addition.

The above described process is applied for each neighboring node u. Then, the channel reported by a larger number of neighbors will be selected for the operation of the network. Note that because more than one channel may be reported by the same amount of nodes, a tie-break mechanism is needed to guarantee that the process leads to equal results in all nodes. One simple approach that could be used is to select the channel with the highest identifier. However, this would lead to a lower usage of channels with lower identifiers and therefore to providing the attacker with valuable information about channel usage in the CBSN. As a consequence, we propose to use a tie-break method that relies on the format of D_i^u.

Recall that a fundamental characteristic of this protocol is that there is no central entity that is known and trusted by all sensors. This makes the protocol suitable for unattended scenarios, and it also makes it more efficient in terms of data transmitted through the network, because no information is sent regarding which channels have to be sensed or which channel is finally selected. Instead, sensors are deployed with all the information needed to perform the sensing in

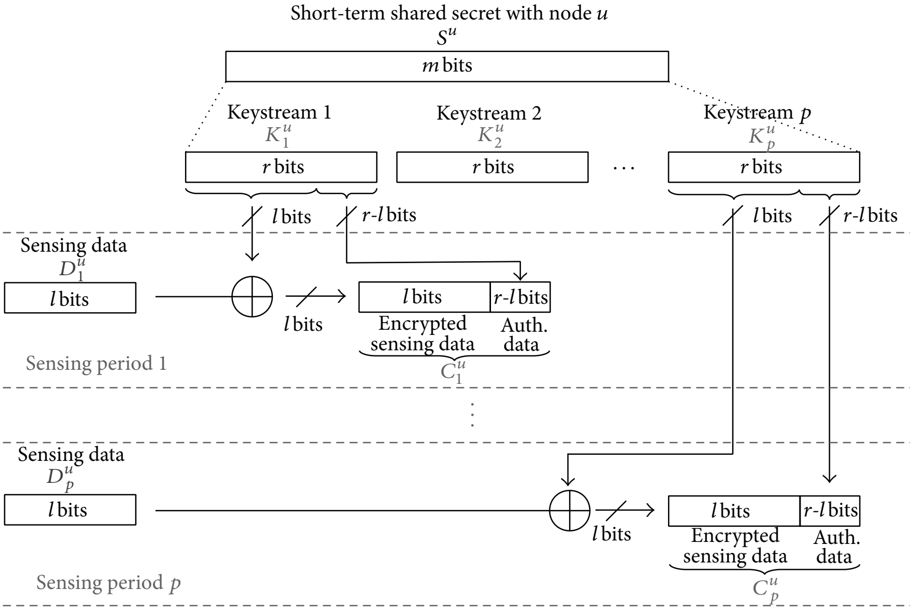


Figure 5: Encrypting and authenticating sensing data.

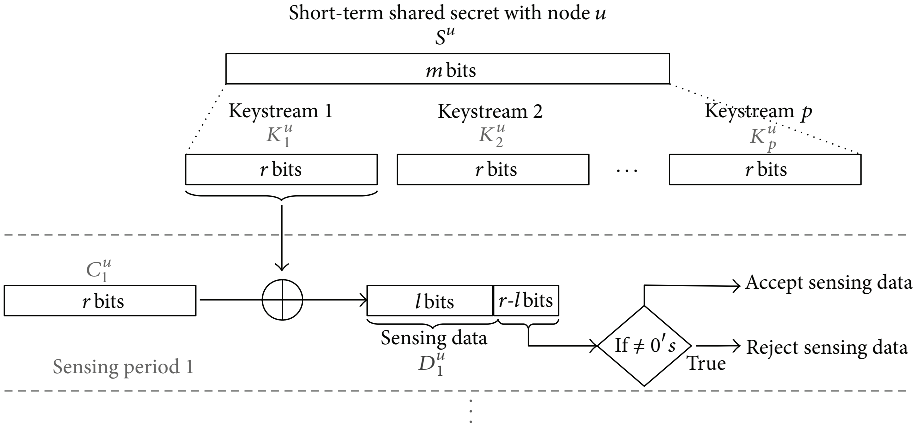


Figure 6: Decrypting sensing data.

a distributed way and make a joint decision autonomously. Thus, there is no need for additional mechanisms to be used when a new node joins the network. In this case, the new node needs to synchronize with the rest of the members to get the proper value of the session counters by making use of the corresponding protocol. However, when a node is expelled from the network because it has been compromised, new cryptographic material must be generated and distributed among the remaining nodes. The KM is responsible for triggering this process and communicates with each sensor node to update the shared long-term secret S_L. Note that because the KM shares a different secret $S_{s,i}$ with each node i, it can securely distribute the new value of S_L. Upon reception of S_L, each node should perform again the initialization process described at the beginning of this section.

5. Security Analysis

The security of the proposed method relies on the shared secrets used to derive the keys and perform encryption and authentication of channel availability data. As long as these secrets are not compromised, data confidentiality can be ensured; that is, an attacker might not be able to get the list of channels to be used in the CBSN. Besides, the method must prevent an attacker from injecting fake data into the system. These issues are discussed as follows. In Section 5.1, we analyze how often the shared secrets should be updated in order to guarantee a proper protection against cryptanalysis; next, in Section 5.2, we evaluate the packet authentication method used in our proposal in terms of probability of bypassing the authentication check.

5.1. Shared Secrets Lifetime. As previously mentioned in Section 3.1, for this analysis, we are assuming that attacks come from external entities and therefore attackers are not able to obtain the cryptographic material that is stored in the body sensors. In the context of this proposal, the lifetime of each of the shared secrets is the interval in which these secrets are considered computationally safe against cryptanalysis, that is, their cryptoperiod.

The cryptoperiod straightly depends on the chosen cryptographic protocols, the length of the secrets themselves, and the amount of times they are used. The more a given secret is used, the shorter its cryptoperiod is, as an attacker gets more information about this secret and therefore the probability of a successful cryptanalysis increases. In fact, cryptoperiod is defined more in terms of the number of times a given secret or key is reused (the amount of ciphertext exposed to an attacker for a given secret/key) than as a given time period, which strongly depends on the transmission rate of the sensor nodes.

Recall the three types of shared secrets used in the proposed method:

(i) *Short-term per-node shared secrets*: one secret S^u per source node that is used to lightweight encrypt, decrypt, and authenticate the channels' sensed data.

(ii) *Medium-term globally shared secrets*: globally shared secret S_M that is used to derive the short-term per-node shared secrets S^u.

(iii) *Long-term globally shared secret*: this being the initially preloaded secret S_L that is used to derive a new medium-term globally shared secret S_M when the current one is about to expire.

As clearly denoted in Figure 5, our approach operates, in some manner, as an additive stream cipher. It is well known that stream ciphers are considered to be secure as long as the key is never reused and, thus, our cipher will be secure if a given value S^u is not repeated. As a result, S^u must be updated every $p = m/r$ sensing period, with p the renewal period, m the length of the shared secret S^u, and r the amount of transmitted bits (sensing data) in a sensing period that are encrypted with S^u.

Recall that, in our proposal, the per-node key used for encryption S^u is generated by means of a hash function. A second requirement is that this function must be cryptographically secure. Note that if the hash function does not accomplish it, an attacker might be able to reverse it, that is, to get the input of the hash function given an output, meaning that, in our proposal, an attacker would be able to recover the value of the medium-term globally shared secret S_M (see Figure 4).

A cryptographically secure hash function with an output of m bits can offer a security level of 2^m operations against preimage attacks and $2^{m/2}$ against collision attacks. Generally speaking, a minimum output of 128 bits is required in order to provide a high level of security for most applications but shorter lengths are accepted if the number of generated messages in a given period is limited, as it is the case of low-rate networks. In Section 6, we propose several lightweight hash candidates with an output of 128 bits; that is to say, we can assume that it is computationally unfeasible for an attacker to invert the hash function and thus to predict the value of S^M as long as it is updated before exceeding its cryptoperiod, which has an upper bound of $2^{m/2} = 2^{64}$ uses.

The long-term globally shared secret S_L is only used to update the current medium-term globally shared secret S_M. Because S_M is not updated very often, it is very unlikely that an attacker manages to obtain several values of S_M to reverse the hash function and recover S_L. As a result, we can assume that the S_L cryptoperiod is long enough and there is no need to update the secret during the nodes' lifetime.

5.2. Authentication. A cryptogram C_i^u of sensing data contains an authentication field of 16 bits that is checked upon reception (see Figure 6). Consequently, an attacker has a chance of 1 in 2^{16} of guessing the next authentication field, which allows it to forge a valid authentication field and inject fake data. Note that this attack can lead the CBSN to wrong decisions about the availability of the spectrum.

If the attacker repeatedly attempts to send valid ciphertexts, it may succeed after 2^{15} attempts, in average. Because the attacker does not know S^u, the authentication field appears to it as a random stream and therefore it must select a fake authentication field at random. Besides, the attacker cannot determine whether a given ciphertext has been accepted or rejected because the receiver does not acknowledge the reception of such packets to the emitter. Otherwise, the attacker could take advantage of this information in order to guess a valid authentication field in a faster way.

In conventional networks, 2^{15} packets may seem an extremely low number but it may provide an adequate level of security in CBSNs. In these networks, the attacker can only send fake packets during the sensing periods, which is in the order of a few milliseconds in most cognitive scenarios [31]. Moreover, as previously stated in Section 1, transmission rates in BSNs are considerably low, with values usually ranging from tens to a few hundred of kilobits per second.

As an example, let us consider a 1 Mbps link, a sensing period of 10 ms, and a packet size of 10 bytes (which is clearly bigger than the typical packet size in sensor networks). Given these parameters, an attacker would only be able to send 125 packets at most in every sensing period. That is to say, the attacker would need an average of 262.144 sensing periods to send a fake packet and pass the authentication check.

6. Cost Evaluation and Comparison with Other Approaches

In this section, we evaluate the cost of our proposal in terms of energy consumption due to transmission overhead and computational cost and compare its performance with the most common approach adopted in sensor networks [32], which is providing authentication and/or encryption of the channel sensing data by means of using standard block ciphers. As is well known, block ciphers have as input the message to be encrypted or authenticated, which is divided into several blocks of fix length and a key. Both the block length and the key length depend on the algorithm being used. Regardless of the algorithm, block ciphers can be used in several modes of operation depending on the service to be provided, that is, encryption only, authentication only, or encryption/authentication. Generally, the following modes of operation are applied.

Authentication. CBC-MAC is a block cipher mode for generating message authentication codes. The message to be authenticated is divided into several blocks of equal size, and each block is encrypted so that the value of a given block depends on the encryption of the previous block. The final output of the cipher, that is, the message authentication code or CBC-MAC, is the result of encrypting the last block of the message. When the input of the cipher is shorter than the block size (as it is usually the case in sensor networks), the CBC-MAC can be obtained by directly encrypting a single block, padded until the block size of the cipher is reached.

Encryption. CTR mode of operation turns a block cipher into a stream cipher, meaning that the resulting ciphertext has the same size as the input or plain text. Thus, it does not force the output length to be a multiple of the block size, as it is the case of other modes such as CBC-MAC. This property makes this mode of operation suitable for encryption in sensor networks where devices usually exchange short-length messages.

Authentication+Encryption. CCM (CTR + CBC-MAC) is a common choice for providing both encryption (CTR) and authentication (CBC-MAC) [19]. A minor variation of CCM, called CCM*, is used in the ZigBee standard [7].

In this work, we assume that Advanced Encryption Standard (AES) in CTR mode is used for encryption and AES CBC-MAC for authentication, as it is the current standard for symmetric cryptography, even in sensor nodes [33]. Currently, there are efficient hardware implementations of AES that are highly affordable.

Regarding hardware platforms, the vast majority of previous works on BSNs have used the 16-bit Texas Instruments' (TI) MSP430 and CC2540 families of microcontrollers [34]. Built around a 16-bit CPU, ultra-low-power MSP430 microcontrollers are designed for low cost and, specifically, low-power consumption embedded applications. As an example, TI's CC2540 family [35] enable robust Bluetooth low energy (BLE) network nodes to be built with low total bill-of-material (BOM) costs. BLE operates in the same spectrum range (2.400 GHz–2.4835 GHz ISM band) as classic Bluetooth technology but uses a different set of channels; instead of 79 1 MHz channels, BLE offers 40 2 MHz channels, 3 for advertising purposes and 37 for data exchange.

6.1. Transmission/Reception Overhead. In this section we analyze the overhead introduced by our proposal in terms of transmission/reception of channel availability data and compare it with the overhead exhibited when conventional approaches are used for data encryption/authentication, as explained above. We assume that all sensors are capable of sensing a given set of channels and report information about their state.

With our proposal, the minimum number of transmitted bits will depend on the number of channels that a given sensor is reporting, the number of bits used to code the state of each channel, and the length of the authentication code. As explained in Section 5.2, a length of 16 bits is enough to secure most applications in WSNs and, thus, we have assumed this value for the authentication field. This leads to a total amount of $\text{Bits}_{tx} = l + 16$ of transmitted bits, where l represents the total number of bits used to code all possible channels, and $\text{Bits}_{tx} = (n - 1)\text{Bits}_{tx}$ received bits, representing the number of bits received by a given sensor from its neighbors.

Aiming to provide a fair comparison, we choose the same key length for block ciphers and for our proposal, that is to say, a 128-bit key. The transmitted bits overhead added by AES CBC-MAC authentication is 128 bits (for a semantically secure implementation also an IV or nonce must be shared between emitter and receiver, so that the overhead can be higher). Regarding encryption, the number of transmitted bits is equal to the number of bits l used to code the state of the channels, but it also requires the use of a nonce, with a length equal to half of the key length, that is, 64 bits per message.

During every sensing period, every node must transmit a packet with sensing information but also must process the packets received from its neighbors. Table 1 and Figure 7, respectively, show the transmission and reception overhead due to the secure sharing of sensing information using both standard block ciphers and our proposal. The values are provided as a function of the number of bits l used to code the state of the channels and the number of nodes N, ranging from 5 to 30. Given that the considered scenario is a body sensor network, this is more than a reasonable value, since a patient wearing more than 30 sensors may be an unlikely situation.

The reader may notice that the overhead introduced by this mechanism increases linearly with the number of nodes for both approaches. However, with the proposed method the transmission/reception overhead is considerably reduced with respect to the use of standard block ciphers while still maintaining an acceptable level of security. In fact, the more the nodes in the CBSN, the bigger the improvement introduced by the former. As we will show later, the transmission/reception savings lead to a huge saving also in energy consumption.

6.2. Computational Cost. In this section, we provide a comparison of the CPU cost in cycles due to the implementation of the cryptographic functions.

If AES is used, the total cryptographic cost per node for securing the exchange of sensing data equals the cost of one encryption and $N - 1$ decryptions. Assuming the Texas Instruments' reference AES-implementation for CC2540 microcontrollers [36], AES encryption needs 6600 cycles/block and AES decryption 8400 cycles/block.

With our proposal, the energy consumption has three components: the computation of the different sequences S_i^u with the hash function before the sensing period begins, the XOR of the keystream K_i^u with the sequence D_i^u that signals the availability of channels, and the XOR of the sequences C_i^u received from neighbors with the precomputed S_i^u. Therefore,

Table 1: Transmission overhead.

	l (bits)	Overhead (bits)
Authentication only (CBC-MAC)	16	128
	32	128
	64	128
Encryption only (CTR)	16	80
	32	96
	64	128
Authentication and encryption (CCM)	16	208
	32	224
	64	256
Authentication and encryption (our proposal)	16	32
	32	48
	64	80

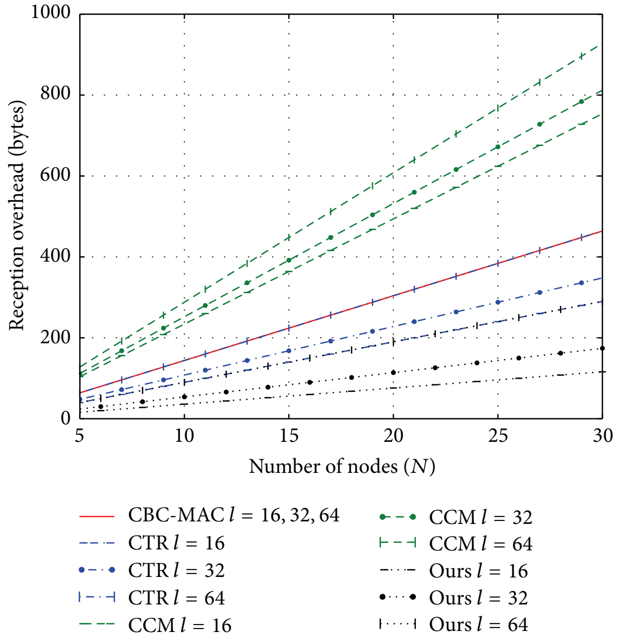


Figure 7: Reception overhead for a varying number of nodes N.

Table 2: Cycles per block for different cryptographic hash functions.

Algorithm	Hash output size	Cycles/block
(Stripped) MAME	128	96
H-PRESENT-128	128	32

a node must compute N hashes and perform N XORs to send channel information and process the reports received from its neighbors.

As per [37] XOR operation with a MSP430 accounts for 4-5 cycles/byte. Assuming a 128-bit hash function and the worst case, every XOR in our proposals accounts for $5 \cdot 16 = 80$ cycles. As previously explained in Section 5.1 we suggest the use of a lightweight cryptographic hash function specifically suited for low-end devices with an output of 128 bits. Table 2 depicts the number of CPU cycles per block needed [38] for two potential candidates. As clearly seen in the table, computing one of these hash functions requires only a few tens of cycles/block. In the following, for this analysis we will consider the (stripped) MAME hash function, as it is a pure hash function that does not rely on a symmetric cipher and requires more CPU cycles (worst case).

Taking into account the previous data, Figure 8 shows the CPU cycles consumed with both our proposal (with a MAME hash function) and standard AES-128 security. It can be clearly seen that our proposal scales much better with the number of nodes while still providing an appropriate level of security. We do not provide values for the process time, that is, the time needed for key generation, encryption, decryption, and data authentication, but the number of CPU cycles required to execute each function. In this way, time values can be obtained for a particular sensor node according to its CPU features.

6.3. Total Energy Cost. Table 3 provides the transmission and reception energy consumption of a TI's CC2540 microcontroller for different modes and values of transmission power. The displayed values have been tested under 25°C. We do not have values for in-body conditions (≈38°C) but we present these data for reference purposes. In any case, this fact only affects the energy consumption for receiving/transmitting. Note that the total cost in terms of energy is much higher, since it should also account for duty cycles, state changes, and other parameters [39].

There is no single specific "energy per CPU cycle" value since the cycle consumption depends on the type of CPU operation. Anyway, in [34], the authors measured that the TI's MSP430F1611 consumes energy at an average of 0.72 nJ per clock cycle, and we have adopted such value in our study.

Figure 9 shows the total average energy consumption, as a function of the number of neighboring sensor nodes and the number l of bits used to code the channels, for the proposed mechanism and standard approaches: CCM (authentication and encryption), CTR (encryption only), and CBC-MAC (authentication only). We have assumed standard reception and short-range transmission of −6 dBm (see Table 3).

As clearly denoted in the figure, our method only requires a few tens of μJ regardless of the number of sensor nodes, while AES-based security requires higher values of energy ranging from 35 μJ to almost 240 μJ when encryption and authentication are provided and for 30 sensor nodes. The more the number of sensor nodes, the bigger the improvement introduced by our proposal.

It must be remarked, however, that the purpose of this figure is to provide reference values to be taken into account for future implementations. Besides, the level of security provided by our method is lower than AES-based methods but still is more than adequate given the features of CBSNs (transmission rate, number of sent messages, etc.) and given the fact that the data we are trying to protect are just a limited number of potential channels to be used for operation of the network.

TABLE 3: Current/energy consumption for RX/TX tested on Texas Instruments CC2540 EM with $T_A = 25^\circ$C with no peripherals active and low MCU activity at 250 kHz.

Test conditions	Current	Energy/byte
RX standard	19.6 mA	58.8 nJ
RX high gain	22.1 mA	66.3 nJ
TX −23 dBm	21.1 mA	63.3 nJ
TX −6 dBm	23.8 mA	71.4 nJ
TX 0 dBm	27 mA	81 nJ
TX 4 dBm	31.6 mA	94.8 nJ

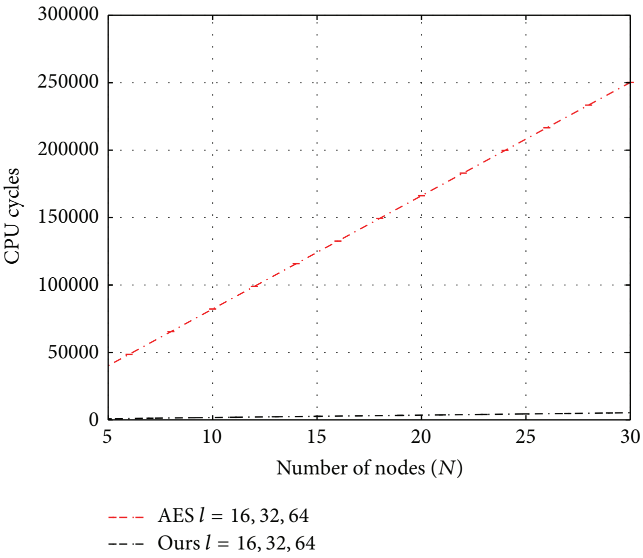


FIGURE 8: Added CPU cycles for a secure exchange of sensing data.

Indeed, this mechanism introduces some overhead due to spectrum sensing and sharing of channel information and, as we pointed out above, the overhead increases linearly with the number of nodes in the network. The synchronization protocol adds some overhead too, but the increase will strongly depend on the chosen protocol. In [29], some protocols with high energy efficiency are referenced, which could be used in our proposal.

Despite it, we claim that it is definitely worth introducing this overhead to increase the availability of the network and make it robust to potential interferences and DoS attacks. Under these undesired situations, the current channel used by the CBSN may become unavailable, but the proposed method allows the nodes to securely agree on a new channel of operation and rapidly resume their transmissions. It must be remarked that securing channel availability information may prevent or at least diminished the effect of further DoS attacks when switching to a new channel.

We do not quantify the benefits of applying our method in terms of attack mitigation because it strongly depends on the capabilities of the attacker (time required for sensing each channel, number of channels that can sense, etc.). As a future work, it would be interesting to estimate the improvement achieved by our approach in terms of throughput of the CBSNs connections and the tradeoff between benefits and overhead.

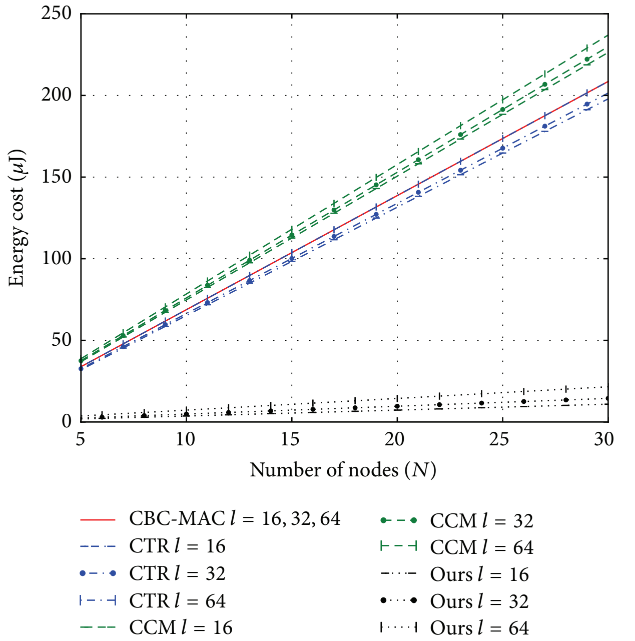


FIGURE 9: Added energy costs for a TI's MSP430F1611 with a TI's CC2540 microcontroller.

7. Conclusions

Body sensor networks (BSNs) emerge as an optimal solution for ensuring constant and remote monitoring of the health status in patients. Recent advances in technology have made it possible to deploy a network of tiny sensors over the human body, and even in body, which can measure vital signs such as temperature, heart rate, or the level of glucose and report these data to medical personnel.

Guaranteeing the availability of such communications is a must as long as connectivity losses during emergency situations may prevent a patient from immediately receiving medical assistance and may end up in catastrophic results.

A new network paradigm, known as cognitive body sensor networks (CBSNs), could mitigate this threat by allowing body sensors to operate in a wide range of frequencies and adapt its transmission parameters according to highly dynamic environment conditions. However, this would come at the expense of implementing cooperative spectrum sensing mechanisms that allow sensors to exchange information about channel quality and availability. As a consequence, CBSNs might become vulnerable to specific attacks that are targeted to these mechanisms.

In this paper we presented a novel and simple method to secure the sensing process in a CBSN and improve its availability. The method relies on cryptographic primitives that require a minimum amount of memory and low energy consumption, thus being more suited for devices with limited resources than traditional approaches. It offers authentication and encryption of control data shared by the sensors in the CBSN to agree on a given channel.

Our proposal was analyzed in terms of security and we showed that although it does not provide the same level of security as AES-based encryption and authentication, it is still sufficient for low packet rate networks such as CBSNs. The provided results also showed that our method outperforms existing approaches in terms of transmission/reception overhead and number of CPU cycles needed, particularly as the number of sensor nodes increases. For typical microcontrollers as CC2540 and MSP430, the improvement in energy consumption clearly justifies the use of the proposed method against AES-based mechanisms in constrained networks such as CBSNs, in which maximizing the network life is extremely important.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This work has been partially supported by the spanish government under grant TEC2011-22746 "TAMESIS" and by the Ministry of Economy and Competitiveness through the projects CO-PRIVACY (TIN2011-27076-C03-02) and SMARTGLACIS (TIN2014-57364-C2-2-R).

References

[1] B. Latré, B. Braem, I. Moerman, C. Blondia, and P. Demeester, "A survey on wireless body area networks," *Wireless Networks*, vol. 17, no. 1, pp. 1–18, 2011.

[2] S. Movassaghi, M. Abolhasan, J. Lipman, D. Smith, and A. Jamalipour, "Wireless body area networks: a survey," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 3, pp. 1658–1686, 2014.

[3] 802.15.6-2012—IEEE Standard for Local and metropolitan area networks—Part 15.6: Wireless Body Area Networks, 2012, http://standards.ieee.org/getieee802/download/802.15.6-2012.pdf.

[4] Bluetooth Low Energy (LE), June 2014, http://www.bluetooth.com/Pages/low-energy-tech-info.aspx.

[5] The ANT+ Alliance, 2014, http://www.thisisant.com.

[6] Wi-Fi Alliance, June 2014, http://www.wi-fi.org/.

[7] Zigbee Alliance, June 2014, http://www.zigbee.org/.

[8] M. Rushanan, A. D. Rubin, D. F. Kune, and C. M. Swanson, "SoK: security and privacy in implantable medical devices and body area networks," in *Proceedings of the 35th IEEE Symposium on Security and Privacy (SP '14)*, pp. 524–539, San Jose, Calif , USA, May 2014.

[9] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, "NeXt generation/dynamic spectrum access/cognitive radio wireless networks: a survey," *Computer Networks*, vol. 50, no. 13, pp. 2127–2159, 2006.

[10] B. Wang and K. J. R. Liu, "Advances in cognitive radio networks: a survey," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 1, pp. 5–23, 2011.

[11] Y. Xu, J. Wang, Q. Wu, A. Anpalagan, and Y.-D. Yao, "Opportunistic spectrum access in unknown dynamic environment: a game-theoretic stochastic learning solution," *IEEE Transactions on Wireless Communications*, vol. 11, no. 4, pp. 1380–1391, 2012.

[12] D. Cavalcanti, S. Das, J. Wang, and K. Challapali, "Cognitive radio based wireless sensor networks," in *Proceedings of the 17th International Conference on Computer Communications and Networks (ICCCN '08)*, pp. 1–6, August 2008.

[13] O. León, J. Hernández-Serrano, and M. Soriano, "Securing cognitive radio networks," *International Journal of Communication Systems*, vol. 23, no. 5, pp. 633–652, 2010.

[14] M. Rostami, A. Juels, and F. Koushanfar, "Heart-to-Heart (H2H): authentication for implanted medical devices," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS '13)*, pp. 1099–1111, ACM, Berlin, Germany, November 2013.

[15] K. B. Rasmussen, C. Castelluccia, T. S. Heydt-Benjamin, and S. Capkun, "Proximity-based access control for implantable medical devices," in *Proceedings of the 16th ACM Conference on Computer and Communications Security (CCS '09)*, pp. 410–419, Chicago, Ill, USA, November 2009.

[16] L. Shi, M. Li, S. Yu, and J. Yuan, "BANA: body area network authentication exploiting channel characteristics," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 9, pp. 1803–1816, 2013.

[17] C. C. Tan, S. Zhong, H. Wang, and Q. Li, "Body sensor network security: an identity-based cryptography approach," in *Proceedings of the 1st ACM Conference on Wireless Network Security (WiSec '08)*, pp. 148–153, April 2008.

[18] O. Garcia-Morchon, T. Falck, T. Heer, and K. Wehrle, "Security for pervasive medical sensor networks," in *Proceedings of the 6th Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous '09)*, pp. 1–10, Toronto, Canada, July 2009.

[19] G. Selimis, L. Huang, F. Massé et al., "A lightweight security scheme for wireless body area networks: design, energy evaluation and proposed microprocessor design," *Journal of Medical Systems*, vol. 35, no. 5, pp. 1289–1298, 2011.

[20] Y. Yan and T. Shu, "Energy-efficient In-network encryption/decryption for wireless body area sensor networks," in *Proceedings of the IEEE Global Communications Conference (GLOBECOM '14)*, pp. 2442–2447, IEEE, Austin, Tex, USA, December 2014.

[21] Y. Xu, A. Anpalagan, Q. Wu, L. Shen, Z. Gao, and J. Wang, "Decision-theoretic distributed channel selection for opportunistic spectrum access: strategies, challenges and solutions," *IEEE Communications Surveys and Tutorials*, vol. 15, no. 4, pp. 1689–1713, 2013.

[22] Y. Xu, J. Wang, Q. Wu, A. Anpalagan, and Y.-D. Yao, "Opportunistic spectrum access in cognitive radio networks: global optimization using local interaction games," *IEEE Journal on Selected Topics in Signal Processing*, vol. 6, no. 2, pp. 180–194, 2012.

[23] R. Chavez-Santiago, K. E. Nolan, O. Holland et al., "Cognitive radio for medical body area networks using ultra wideband," *IEEE Wireless Communications*, vol. 19, no. 4, pp. 74–81, 2012.

[24] A. R. Syed and K.-L. A. Yau, "On cognitive radio-based wireless body area networks for medical applications," in *Prceedings of the 1st IEEE Symposium on Computational Intelligence in Healthcare and e-Health (CICARE '13)*, pp. 51–57, Singapore, April 2013.

[25] R. Chen, J.-M. Park, and K. Bian, "Robust distributed spectrum sensing in cognitive radio networks," in *Proceedings of the 27th IEEE Conference on Computer Communications (INFOCOM '08)*, pp. 1876–1884, IEEE, Phoenix, Ariz, USA, April 2008.

[26] H. Rifà-Pous, M. J. Blasco, and C. Garrigues, "Review of robust cooperative spectrum sensing techniques for cognitive radio networks," *Wireless Personal Communications*, vol. 67, no. 2, pp. 175–198, 2012.

[27] B. Braem, B. Latré, C. Blondia, I. Moerman, and P. Demeester, "Analyzing and improving reliability in multi-hop body sensor networks," *International Journal on Advances in Internet Technology*, vol. 2, no. 1, pp. 152–161, 2009.

[28] I. F. Akyildiz, B. F. Lo, and R. Balakrishnan, "Cooperative spectrum sensing in cognitive radio networks: a survey," *Physical Communication*, vol. 4, no. 1, pp. 40–62, 2011.

[29] B. Sundararaman, U. Buy, and A. D. Kshemkalyani, "Clock synchronization for wireless sensor networks: a survey," *Ad Hoc Networks*, vol. 3, no. 3, pp. 281–323, 2005.

[30] A. Bogdanov, G. Leander, C. Paar, A. Poschmann, M. J. B. Robshaw, and Y. Seurin, "Hash functions and RFID tags: mind the gap," in *Cryptographic Hardware and Embedded Systems—CHES 2008: 10th International Workshop, Washington, D.C., USA, August 10–13, 2008. Proceedings*, vol. 5154 of *Lecture Notes in Computer Science*, pp. 283–299, Springer, Berlin, Germany, 2008.

[31] C. Cordeiro, K. Challapali, D. Birru, and N. Sai Shankar, "IEEE 802.22: the first worldwide wireless standard based on cognitive radios," in *Proceedings of the 1st IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN '05)*, pp. 328–337, Baltimore, Md, USA, November 2005.

[32] S. Ullah, H. Higgins, B. Braem et al., "A comprehensive survey of wireless body area networks on PHY, MAC, and network layers solutions," *Journal of Medical Systems*, vol. 36, no. 3, pp. 1065–1094, 2012.

[33] N. Sastry and D. Wagner, "Security considerations for IEEE 802.15.4 networks," in *Proceedings of the ACM Workshop on Wireless Security (WiSe '04)*, pp. 32–42, Philadelphia, Pa, USA, October 2004.

[34] M. Pajic, Z. Jiang, I. Lee, O. Sokolsky, and R. Mangharam, "From verification to implementation: a model translation tool and a pacemaker case study," in *Proceedings of the 18th IEEE Real Time and Embedded Technology and Applications Symposium (RTAS '12)*, pp. 173–184, Beijing, China, April 2012.

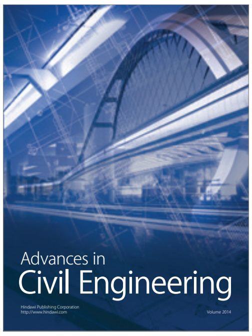
[35] CC2540 2.4-GHz Bluetooth low energy System-on-Chip, 2013, http://www.ti.com/lit/ds/symlink/cc2540.pdf.

[36] U. Kretzschmar, "AES128—a C implementation for encryption and decryption. MSP430 systems. ECCN 5E002 TSPA—technology/software publicly," Application Report SLAA397A, Texas Instruments, Dallas, Tex, USA, 2009, http://www.ti.com.cn/cn/lit/an/slaa397a/slaa397a.pdf.

[37] El Barquero, *Aqueronte: MSP430: Cycles and Instructions*, edited by: Aqueronte, 2011, http://unbarquero.blogspot.com.es/2011/05/msp430-cycles-and-instructions.html.

[38] A. Bogdanov, G. Leander, C. Paar, A. Poschmann, M. J. B. Robshaw, and Y. Seurin, "Hash functions and RFID tags: mind the gap," in *Cryptographic Hardware and Embedded Systems—HES 2008*, vol. 5154 of *Lecture Notes in Computer Science*, pp. 283–299, Springer, Berlin, Germany, 2008.

[39] S. Kamath and J. Lindh, "Measuring bluetooth low energy power consumption," Application Note AN092, Texas Instruments, Dallas, TEx, USA, 2012, Version SWRA347a, http://www.ti.com/lit/an/swra347a/swra347a.pdf.



Hindawi

Submit your manuscripts at
http://www.hindawi.com