Lightweight and Confidential Data Discovery and Dissemination for Wireless Body Area Networks

Daojing He, *Member, IEEE*, Sammy Chan, *Member, IEEE*, Yan Zhang, *Senior Member, IEEE*, and Haomiao Yang, *Member, IEEE*

***Abstract*—As a special sensor network, a wireless body area network (WBAN) provides an economical solution to real-time monitoring and reporting of patients' physiological data. After a WBAN is deployed, it is sometimes necessary to disseminate data into the network through wireless links to adjust configuration parameters of body sensors or distribute management commands and queries to sensors. A number of such protocols have been proposed recently, but they all focus on how to ensure reliability and overlook security vulnerabilities. Taking into account the unique features and application requirements of a WBAN, this paper presents the design, implementation, and evaluation of a secure, lightweight, confidential, and denial-of-service-resistant data discovery and dissemination protocol for WBANs to ensure the data items disseminated are not altered or tampered. Based on multiple one-way key hash chains, our protocol provides instantaneous authentication and can tolerate node compromise. Besides the theoretical analysis that demonstrates the security and performance of the proposed protocol, this paper also reports the experimental evaluation of our protocol in a network of resource-limited sensor nodes, which shows its efficiency in practice. In particular, extensive security analysis shows that our protocol is provably secure.**

***Index Terms*—Data discovery and dissemination, efficiency, security, wireless body area networks.**

I. Introduction

ADVANCES in wearable and implantable body sensors, wireless communication technologies, and embedded computing enable the development and deployment of wireless body area networks (WBANs). A WBAN is a wireless network of small and intelligent sensors which are attached to or implanted in a patient to monitor his physiological activities and actions. It is an effective way to collect and provide continuous and real-time health and movement related information of patients to medical staff.

Compared to the traditional wireless sensor networks (WSNs) and mobile ad hoc networks (MANETs), security is a more critical issue in WBANs. This is because the medical data collected by WBANs are very sensitive, and the use of such data should conform with the regulations stipulated by healthcare alliances such as HITRUST [1] and legal directives such as those adopted in the U.S. [2] and Europe [3]. If not securely protected, for example, the data may be eavesdropped by adversaries who may sell them to interested parties such as insurance companies as the data could reveal the health condition of patients.

In an operational WSN (including WBAN), some common variables may be stored in each node of the network. The data discovery and dissemination protocols add, delete, and modify such variables by requesting each node to exchange packets so that they eventually become consistent across the network. In the literature, a number of data discovery and dissemination protocols have been proposed [4]–[6]. The most well-known ones are Drip [4], DIP [5], and DHV [6] as they have been included in TinyOS distributions. Unfortunately, to the best of our knowledge, these proposed protocols focus on reliable data dissemination but ignore the security aspect. Thus, ensuring security in these protocols is an unresolved issue. This is particularly important for WBANs since, as described previously, they convey sensitive patients' information.

When we address the problem of secure data discovery and dissemination for WBANs, we need to take into account the distinct features of WBANs. Besides the sensitive nature of the data, WBANs are also different from MANETs and other types of WSNs in terms of available resources of body sensors, the mobility of body sensors, and the proximity to adversaries. Due to these differences, security mechanisms developed for WSNs and MANETs may not be applicable to WBANs. For example, all existing security solutions which involve cryptographically strong protocols incur too much computation and communications cost for body sensors. This will be discussed in more detail in Section III-B.

Motivated by the aforementioned observations, this paper has the following main contributions:

1) We first investigate the security issues in data discovery and dissemination protocols for WBANs and point out that there is a need of authenticating the disseminated data.
2) Taking into account the unique features and application requirements of a WBAN, we extend Drip to become

Manuscript received June 30, 2013; revised October 14, 2013; accepted November 21, 2013. Date of publication December 3, 2013; date of current version March 3, 2014. This work was supported in part by the European Commission FP7 Project EVANS under Grant 2010-269323, the SmartGrids ERA-Net project PRO-NET funded through Research Council of Norway (Project 217006), a strategic research grant from City University of Hong Kong (Project 7004054), the Fundamental Research Funds for the Central Universities, and the Specialized Research Fund for the Doctoral Program of Higher Education. D. He is the corresponding author.

D. He is with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China (e-mail: hedaojinghit@gmail.com).

S. Chan is with the Department of Electronic Engineering, City University of Hong Kong, Hong Kong (e-mail: eeschan@cityu.edu.hk).

Y. Zhang is with the Simula Research Laboratory, 1364 Fornebu, Norway (e-mail: yanzhang@ieee.org).

H. Yang is with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 610051, China (e-mail: haomyang@uestc.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JBHI.2013.2293620

a secure, lightweight, confidential, and denial-of-service (DoS)-resistant data discovery and dissemination protocol for WBANs based on the use of multiple one-way key hash chains. We demonstrate that our protocol incurs lower computational, energy and communication costs while securing the multihop dissemination of data items. Also, we apply the provable security technique to formally prove the authenticity and integrity of the disseminated data items in our protocol.

3) We implement our proposed protocol in testbeds of MicaZ and TelosB motes, respectively. Measurements from experiments verifies its high efficiency in practice.

The rest of this paper is organized as follows. Section II analyzes security vulnerabilities in data discovery and dissemination in the context of WBANs. The unique features of WBANs, related work, and design consideration of securing such protocols are given in Section III. The assumptions, network model, threat model, and the security goals are discussed in Section IV. Then, our protocol is presented in Section V in detail. Section VI provides theoretical analysis of the security of our protocol. Section VII presents the implementation and experimental evaluation of our protocol. Section VIII concludes this paper.

II. Security Vulnerabilities in Data Discovery and Dissemination of WBANs

A. Review on Data Discovery and Dissemination of WBANs

Here, based on Drip, we illustrate the security vulnerabilities in data discovery and dissemination protocols for WBANs. However, note that for other protocols in the literature sharing the same underlying algorithm, Trickle [7], they would also bear the same vulnerabilities. According to Trickle, a node needs to periodically broadcast a summary of its stored data, unless the same summary has been received recently. If a node receives an older summary from its neighboring node, it will update the neighboring node with the latest summary. To save energy, when the data in all nodes are consistent, the broadcast interval is increased exponentially. However, when a node has new data, it will report more quickly. Therefore, when new data are injected by the base station, Trickle will disseminate them quickly. For Drip, it establishes an independent Trickle session for each data item.

Each data item contains a unique key to identify the variable (i.e., parameter or command) that it aims to update, and a value to reflect its freshness. In Drip, each data item is formatted as a three-tuple (*key, version, data*), in which *key* identifies uniquely the concerned variable, *version* indicates whether the data item is new (a larger *version* means a newer data), and *data* denotes the disseminated value for the concerned variable.

B. Security Vulnerabilities in Data Discovery and Dissemination of WBANs

All existing data discovery and dissemination protocols assume the networks to be operated in a benign environment. However, in practice, WBANs may be subject to malicious attacks from external attackers. By placing an intruder node or compromising a node of the WBAN, an adversary could possibly modify or replace the legitimate data being propagated in the WBAN. Furthermore, an adversary can reboot the whole network with wrong data *data** by injecting a fake data item (*key, version, data**) to the network where *version* is larger than all version numbers of the concerned variable stored on the body sensor nodes. Alternatively, the adversary can even erase an important variable identified by *key* from all sensor nodes by sending the data item (*key, version, 0*) using a data discovery and dissemination protocol, where *version* is a large enough number.

Recall that, under Trickle, each node needs to broadcast new data. Adversaries can easily exploit this feature by launching DoS attacks to drain the resources of body sensor nodes from their intended functions. For example, an adversary can inject large volume of bogus data items to the WBAN, effectively exhausting the energy of body sensor nodes to process the bogus data items.

Clearly, without any security protection, the disseminated data items are easily intercepted by eavesdroppers. Leakage of important information may lead to serious consequence. For example, if an adversary eavesdrops a small program update, it can attack the WBAN by exploiting potential bugs in the program update.

III. Unique Features of WBANs and Related Work

A. Unique Features and Application Requirements of WBSNs

In this section, some differences between WBANs and MANETs (or traditional WSNs) are listed as follows.

1) *Communication:* In most cases, the nodes of a WBAN directly communicate with the base stations, but do not communicate among themselves.

2) *Resource:* Body motes have much lower processing speed, memory, and transmission speed than traditional sensor nodes, because they have a very small form factor (often less than 1 cm^3). Also, energy is a scarce resource. It is desirable that body motes can operate for a long period of time (up to tens of years for implanted motes) and yet it is difficult, if not impossible, to change or recharge their batteries.

3) *Physical compromise of body sensor nodes:* WBANs are more vulnerable to physical node compromise attacks because a) mobility: body sensors can sometimes be lost easily when patients move around and b) accessibility: body sensors carried by patients can be easily identified by adversaries.

4) *Mobility:* BAN users may move around. Therefore, WBAN nodes share the same mobility pattern, unlike WSN nodes that are usually considered stationary.

B. Related Work

In the literature, a large number of security solutions [8]–[11] such as secure reprogramming, secure routing, key management have been proposed for traditional WSNs. However, because of the unique characteristics of a WBAN, current security protocols designed for these networks are not always well suited to support resource-constrained WBANs. For example, a commonly used solution [8] is to require the base station to digitally sign each disseminated packet, which will later be verified at

TABLE I
PROS AND CONS OF ALL EXISTING PROTOCOLS

Protocol	Advantages	Disadvantages
A commonly used solution [8]	Low storage requirement	High computational and communication costs
TESLA and its extensions [9]–[11]	Low computational cost	Vulnerable to a DoS attack
Both hash chain and hash tree methods [12]	Moderate computational cost	All data items must be available beforehand
Symmetric cryptosystems [13]	Low computational cost	Same type of physiological parameters need to be measured by each biosensor
Public key cryptography based methods [14], [15]	Communication effecient	Computationally inefficient

each node before any processing. Unfortunately, such a solution is not suitable for WBANs due to its high computational and communication costs. TESLA and its extensions [9]–[11] have been proposed to address this issue. In these schemes, the base station reveals the key to the nodes at a later time, when nodes can verify that the key is valid. However, these schemes require time synchronization among the nodes and are vulnerable to a DoS attack because of the authentication delay. For example, if each node is flooded by forged data items, they are buffered in the node until the key is disclosed by the base station. In the literature, to solve the above challenge, both hash chain and hash tree methods [12] have been proposed to achieve broadcast authentication of multiple packets. However, besides the high computation overhead due to signature verification on each node, another disadvantage of these two methods is that all data items to be broadcast must be available beforehand. They are needed by the base station to generate the digital signature for bootstrapping the authentication process. In contrast, a good scheme should not require prior knowledge of all data items to be broadcast. Preconfiguring a shared key in each node does not offer enough protection on data discovery and dissemination against physical compromise of body sensor nodes.

Recently, many security solutions [13]–[15] have been designed for WBANs, but they cannot be directly employed to ensure the data discovery and dissemination security. For example, some security schemes for WBANs using symmetric cryptosystem have been proposed [13], where the body's physiological signals are used for secure key distribution. However, it is assumed that the same type of physiological parameters can be measured by each biosensor. Such an assumption limits the application scope of this approach. Moreover, all these symmetric key cryptographic mechanisms [13] cannot be employed to provide security protection on data discovery and dissemination, since this design is exposed to single-node-compromise attack. Security mechanisms for WBANs based on public key cryptography are studied in [14] and [15]. In [14], ECC is used to establish symmetric keys between the base station and sensor nodes. In [15], by encrypting medical data using identity-based public key, the balance between security and privacy with accessibility is maintained. However, their computational load is rather heavy for resource-limited biosensors. For example, [14] reports that, for a Tmote Sky mote, the ECC key agreement takes 7.198 s.

We summarize the pros and cons of all existing protocols in Table I.

IV. ASSUMPTIONS, NETWORK MODEL, THREAT MODEL, AND SECURITY GOAL

A. Assumptions

Our protocol is based on the following assumptions.

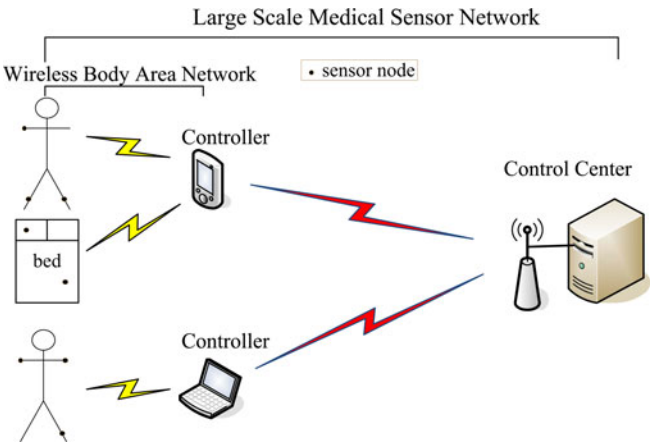


Fig. 1. Architecture of a typical large-scale MSN.

1) Being the source of dissemination, the base station is assumed to be noncompromisable and trustworthy. This assumption is reasonable because if the source of dissemination is not trustworthy, nothing can be done by the security measures in the data discovery and dissemination. Note that physically compromising the base station is harder because they can be protected in secure locations.
2) The body sensor nodes are resource constrained in terms of memory space, computation power, bandwidth, and energy. Due to the constrained resources, computationally expensive and energy intensive operations such as public key cryptography are not favorable for such nodes.

B. Network Model

As shown in Fig. 1, a large-scale medical sensor network (MSN) deployed in a medical center (e.g., home, hospital, or large medical facilities) accommodates some WBANs. For each WBAN, body sensors are used to automatically monitor physiological readings, which can be forwarded to a nearby processing device, referred to as the *controller* or *base station*, such as a smart phone, a wrist watch, a tablet PC, a laptop PC, or a robot, based on the application needs. Packets broadcast by the base station can be received by each body sensor through one or two hops. This two-tier architecture conforms the features of a WBAN such as communication, resource, deployment, density, and mobility characteristics. Moreover, such a two-tier architecture is indispensable for increasing overall network capacity and scalability, reducing system complexity, prolonging network lifetime, and ensuring the security and privacy. With respect to disseminating data items, sensors of practical WBANs are at most three hops away from the base station.

C. Threat Model

We assume that the adversary has access to computationally resourceful entities such as desktop PCs. It may launch either external or insider attacks. For external attacks, an adversary can eavesdrop all the traffic within an MSN, inject arbitrary messages, replay old messages, spoof node identities, and jam the communication channel. However, it is assumed that the adversary cannot jam the channel for a long period of time without being detected and antijammed. Also, an adversary can launch DoS attacks to consume the limited resources (e.g., battery power, memory) on selected nodes. For insider attacks, an adversary can physically compromise some nodes, extract the cryptographic keys from them, and then launch attacks on the network. We make no assumption about the number of adversaries, their localizations, their radio range, or the degree of collusion among them.

D. Security Goal

Our overall objective is to build a secure data discovery and dissemination protocol for WBANs using limited-resourced wireless body motes. Desirable properties include data authenticity and integrity, data freshness, dynamic data, semantic security, energy efficiency (with respect to both communication and computation), low storage overhead, data confidentiality, delay tolerance and immediate authentication, and resistance against malicious attacks such as node compromise and DoS attacks. Dynamic data means that the sequence of broadcast data items need not be known beforehand in its entirety by the base station.

V. Proposed Protocol

Before giving the detailed description of our protocol, we first provide an overview of our protocol.

A. Overview of the Proposed Protocol

In our protocol, symmetric key cryptography is used to achieve data confidentiality because they are computationally efficient even for resource constrained body sensor nodes. To prevent the untrusted nodes to forge the broadcast data items, the symmetric keys are changed on a per-packet basis. The symmetric keys are derived from a one-way hash chain, where a key can only be verified, but cannot be forged, from the previous key. Therefore, the authenticity of the broadcast data items is achieved. In addition, the proposed protocol can encrypt each data item independently, without the need to know the entire sequence of data items *a priori*. The aforementioned features achieve our design goals for the nodes which are one hop away from the base station. However, for nodes which are further away, they can disentangle the keys from the data items, and then use the keys to encrypt malicious data for subsequent forwarding. To resist such an impersonation attack, our protocol generates different key chains for nodes in different hop groups. More specifically, the base station establishes multiple distinct one-way key chains, each of which corresponds to the nodes with the same hop distance from the base station. This is motivated by some features of the WBANs. For example, as described previously, the maximum hop numbers from the base station to all nodes is very small, i.e., two or three hops. Thus, the communication overhead incurred by the proposed approach is small. Another feature is that, for WBANs, often the hop distance of each node from the base station remains unchanged.

TABLE II
Notations

Notation	Description
$E(X, K)$	encrypting message X with a symmetric key K
$D(X, K)$	decrypting cipher text X with a symmetric key K
, or $\|$	concatenation operator of the two bit streams
$H(.)$	public one-way cryptographic hash function (e.g., SHA-1)
$H(M)$	the hash value of message M

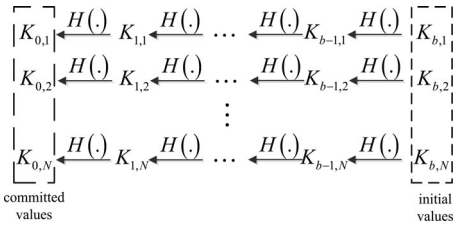


Fig. 2. Construction of multiple one-way key hash chains.

In our proposed protocol, different from most of broadcast authentication methods, a digital signature is not required to bootstrap the authentication process. Instead, multiple distinct one-way key chains are distributed to the nodes based on their hop distances to the base station for packet authentication. This obviates the need for digital signature, resulting in efficient use of the limited computational power of each node.

The proposed protocol consists of three phases: system initialization, packet preprocessing, and packet verification. The system initialization phase is carried out before network deployment. In this phase, the base station creates multiple one-way key hash chains, and loads the committed value of the hash chain on each node corresponding to the hop distance. Then, before disseminating data items, the base station executes the packet preprocessing phase in which data packets are constructed. Finally, in the packet verification phase, a node verifies each received packet. If the result is positive, it updates the data according to the received packet. In the following, each phase is described in detail. The notations used in the description are listed in Table II.

B. System Initialization Phase

Our protocol employs multiple one-way hash chains to secure the Drip protocol. Hash chains are based on a function $H(.)$ with the property that its computation is easy, whereas its inverse $H^{-1}(.)$ is extremely difficult to compute. A hash chain with length b is generated by applying $H(.)$ to an initial element repeatedly for b times (i.e., as shown in Fig. 2). The last value after $H(.)$ has been applied b times is called the committed value of the hash chain (see Fig. 2).

The nodes are divided into N groups according to their hop distance from the base station. Before the nodes are deployed,

the base station constructs N hash chains as follows. It generates N distinct random seed numbers and computes a one-way hash chain with the length b starting from each seed, as shown in Fig. 2 (the $(b-i)$th output of hash function derived from the jth random seed number (i.e., $K_{b,j}$) is denoted as $K_{i,j}$). Here, the length b of each chain can be arbitrary but no less than the number of data items that the base station wants to disseminate in the lifetime of the network.

The committed value of the jth key chain, $K_{0,j}$, corresponding to hop distance j, is predistributed to the nodes in the jth hop group before they are deployed. A secure method suitable for key distribution before node deployment is the Message-In-Bottle protocol [16].

In our protocol, we extend the three-tuple *(key, version, data)* of Drip into a four-tuple *(order, key, version, data)* to represent a data item, where *order* refers to the dissemination order of this data item (the higher the order, the newer the data dissemination), and the other three elements bear the same meaning as that in Drip. Same as the Drip implementation, *key* and *version* are 2 and 4 bytes long, respectively. For the *order* field, it can be just as short as 4 bits because we can allow a wrap around in the number space to take place. This is possible based on two characteristics of the dissemination process. First, the configuration of a WBAN is not expected to change frequently, and hence, the dissemination rate would be low. Second, only a small amount of data is disseminated in each round, so the time required to complete one round of dissemination should be very short. As a result, each sensor node would not experience any ambiguity in determining which order number is the latest even if there is a wrap around in an order number.

C. Packet Preprocessing Phase

After system initialization, if the base station wants to disseminate a data item $d = \{$*order, key, version,data*$\}$, it creates a packet through concatenating the data item d and the successor keys, and encrypts the data item with a symmetric encryption technique using the successor keys. More specifically, the packet preprocessing result (P_i) for the ith data item ($d_i = \{order_i, key_i, version_i, data_i\}$) propagated to N hops is

$$P_i = E(\{d_i, K_{i,1}\}, K_{i-1,1}) \| \ldots \| E(\{d_i, K_{i,N}\}, K_{i-1,N})$$

where $1 \leq i \leq b$ and $order_i = i$. Note that different from DIP and DHV, in Drip, each data item is independently advertised and disseminated. Therefore, with Drip, the base station can disseminate data items either to all nodes through broadcast mode, or just some specific nodes through multicast mode. For the latter case, the identity information of the target nodes can be included in the packet header, i.e., the *Destination* field of the packet. In this case, as described previously, each cryptographic hash $H(.)$ calculated over the *Destination* field ensures its authority and integrity.

D. Packet Verification Phase

In this section, the packet verification for a packet destined to the first hop group will be described. The verification of a packet in the other hop groups uses the same procedure with keys corresponding to those that were used in the packet preprocessing phase.

Upon receiving a packet (denoted by P_i) (from any one-hop neighboring node or the base station), each sensor node, say S_k, retrieves the correct group information from P_i (i.e., the node parses the right field $E(\{d_i, K_{i,1}\}, K_{i-1,1})$ and then uses the key $K_{i-1,1}$ to perform $D(E(\{d_i, K_{i,1}\}, K_{i-1,1}), K_{i-1,1}) = \{d_i, K_{i,1}\}$ to decrypt the cipher text. Then, node S_k runs the following operations.

1) If this is a new data item (i.e., the $order_i$ included in this packet is newer than that of its stored $\langle order_{i-1}, K_{i-1,1}\rangle$), node S_k checks whether the received key $K_{i,1}$ hashes to the stored $K_{i-1,1}$. If the result is positive, the authenticity and integrity of the packet is assured (by use of an appropriate block cipher mode) and the packet is accepted; otherwise, the packet is rejected. The stored $K_{i-1,1}$ is replaced by $K_{i,1}$. If this is a new version, node S_k updates the data according to the *key* of this packet; otherwise, node S_k simply discards this packet.
2) If node S_k has recently heard an identical data packet (i.e., the $order_i$ included in this packet is same as that of its stored $\langle order_{i-1}, K_{i-1,1}\rangle$), it increases the broadcast interval of this packet through the Trickle algorithm, thereby limiting energy costs when a network is consistent.
3) If this is an old data packet (i.e., the $order_i$ included in this packet is older than that of its stored $\langle order_{i-1}, K_{i-1,1}\rangle$. That is, the data packet distributed by its one-hop neighboring node is old), node S_k broadcasts its stored data packet.

VI. Security and Efficiency Analysis

A. Security Analysis

In this section, we analyze the security of our protocol to verify that the security requirements mentioned in Section IV-D are satisfied.

1) Data Authenticity and Integrity: Each packet contains the successor keys of multiple one-way key hash chains which can only be generated by the base station. An adversary cannot obtain any successor key by compromising any node. Thus, the proposed mechanism can ensure that those data items from the base station were not altered in transit.

2) Ensurance of Freshness: The *order* field of a data item is used to indicate not only which key of the key hash chain is used but also the sequence of the data item. However, the *version* field is used to represent freshness of a data identified by the *key* field. Also, as described previously, the authenticity and integrity of these two fields (i.e., *order* and *key*) included in the data items is ensured.

3) Resistance Against Node Compromise: As described in Section V-B, only the committed value of a one-way key hash chain is preloaded in each sensor node corresponding to its hop distance with the base station, which is only used to verify the authority and integrity of a received data item. During the protocol run, only the previously used keys of the key hash chain are stored on the nodes. Thus, even if an adversary compromises any

sensor nodes, the adversary just obtains the committed values and the previously used keys. Thus, regardless of the number of compromised nodes, an adversary cannot launch any security attacks.

4) Resistance Against DoS Attack: Packet verification is a very efficient process based on a symmetric decryption and a hash function operations. Even if an adversary floods the nodes with a large number of packets, not much computation resource is drained to authenticate the packets. Thus, our protocol can resist the DoS attack on authentication delay. Also, authentication of the disseminated data items are carried out in each node. Therefore, each node can reject/filter any bogus data item from the adversary. As a result, the proposed protocol can prevent from the DoS attacks exploiting the Trickle algorithm.

Also, as described in Section V, it is clear that our protocol can ensure dynamic data, data confidentiality, delay tolerance, and immediate authentication.

The data confidentiality of the proposed scheme can be achieved directly through the semantic security of the block cipher. Therefore, in the following, we only give the formal proof of the authenticity and integrity of the disseminated data items in the proposed scheme based on the three assumptions below:

Assumption 1: The block cipher is a secure pseudorandom function (PRF) family. Note that our scheme uses a block cipher with cipher block chaining (CBC) mode. According to [17], Theorem 17], CBC can achieve security by using PRF.

Assumption 2: There exist families of universal one-way hash functions (UOWHFs) [18]. $\mathbb{U}$ is called a family of UOWHFs, if for all probabilistic-polynomial-time (PPT) adversaries, say $\mathcal{A}$, $\mathcal{A}$ have negligible probability in winning the following game: $\mathcal{A}$ first choose a message m, and then $\mathcal{A}$ are given a random function $H(.) \in \mathbb{U}$. To win, $\mathcal{A}$ must output $m' \neq m$ such that $H(m') = H(m)$. Note that in our scheme, $H(.)$ can be implemented by the common hash functions, such as SHA-1.

Assumption 3: The block cipher is semantically secure. Note that in our scheme the block cipher can be implemented by advanced encryption standard (AES) with the CBC mode.

Theorem 1: The proposed scheme achieves the authenticity and integrity of data items, assuming that the block cipher is a secure PRF family, and assuming that $H(.)$ belongs to UOWHFs and the block cipher is semantically secure.

Proof: Our theorem follows from the idea in [19], and thus, here, we only give a proof sketch briefly. The main idea is that the success probability of forging a data item depends on the security of the block cipher and the universal property of the hash function. Here, we prove through a contradiction argument. Suppose that there exists a PPT adversary $\mathcal{A}$, which can defeat the authenticity and integrity of data items in the proposed scheme. This means that $\mathcal{A}$ controls the communication links and manages, with nonnegligible probability, to forge a data item. Then, there is a PPT distinguisher $\mathcal{D}$ which can distinguish between $E(\cdot, K)$ (block cipher family) or *Rand* (true random function) with nonnegligible probability.

To this end, the distinguisher $\mathcal{D}$ is designed to use a function either from $E(\cdot, K)$ or from *Rand* as encryption oracle $\mathcal{O}$. And $\mathcal{B}$ simulates the forger $\mathcal{A}$'s environment to interact with $\mathcal{A}$. We argue that $\mathcal{D}$ succeeds with nonnegligible probability as follows.

First, we denote the advantage of $\mathcal{D}$ by $Adv^{\mathrm{PRF}}(\mathcal{D})$. Namely, $Adv^{\mathrm{PRF}}(\mathcal{D}) = |Pr[\mathcal{O} \leftarrow E(\cdot, K) : \mathcal{D}^{\mathcal{O}} = 1] - Pr[\mathcal{O} \leftarrow Rand : \mathcal{D}^{\mathcal{O}} = 1]|$, where $|\cdot|$ represents the absolute value, $Pr[\cdot]$ represents the probability, and for a set Y, we write $y \leftarrow Y$ meaning that an element sampled from the uniform distribution on Y is assigned to the variable y. If the encryption oracle $\mathcal{O} \leftarrow E(\cdot, K)$, then $\mathcal{D}$ provides $\mathcal{A}$ with forging environment. So, we have $Adv^{\mathrm{PRF}}(\mathcal{D}) = |Pr[\mathcal{A} = 1] - Pr[\mathcal{O} \leftarrow Rand : \mathcal{D}^{\mathcal{O}} = 1]|$. While $\mathcal{O} \leftarrow Rand$, the distinguisher $\mathcal{D}$ outputs 1 only if hashes are equals, for example, considering one-hop nodes as message receivers, this event only occurs when we take the same input $K_{i,1}$ or when we have a collision $H(K_{i,1}{}') = H(K_{i,1}) = K_{i-1,1}$. However, according to Assumption 2, $H(.)$ belongs to UOWHFs. Moreover, since the data item d_i is new and the block cipher $E(\cdot, K)$ is semantically secure, according to Assumption 3, $\mathcal{A}$ cannot decrypt and recover the data item d_i and $K_{i,1}$. Therefore, the aforementioned contradictions ensure the authenticity and integrity of data items in the proposed scheme. ■

B. Efficiency Analysis

The computation complexity of the proposed system on each node only involves one symmetric decryption and one hash function for each received data packet, independent of how many nodes are contained in a WBAN or how big the sizes of data packets are. As will be shown in the next section, the computational complexities of a hash function and an AES symmetric decryption are in the same level (i.e., several milliseconds) when implemented on resource-limited sensor nodes such as TelosB and MicaZ motes; thus, these operations are very fast and can be omitted. Thus, our protocol incurs very little overhead of computing. Also, a node only needs to save a two-tuple $\langle order_{i-1}, K_{i-1,1} \rangle$ in RAM to verify the integrity of each packet. So, it is very memory-efficient. The communication overhead (i.e, the radio power consumption) is proportional to not only the length of each data item but also the number of hops of data dissemination. Because the data item is very short (e.g., several bytes) and the sensors of practical WBANs are at most three hops away from the base station, our protocol is communication-efficient.

VII. Implementation and Performance Evaluation

We evaluate the proposed protocol by implementing all its components in an experimental test-bed. Also, we choose Drip for performance comparison.

A. Implementation and Experimental Setup

We have implemented the base station side programs in C with the OpenSSL tookit [20], and the sensor node side programs in nesC, respectively. In our experiments, the base station side programs run on laptop PCs with 2-GB RAM, operating under the Ubuntu 11.04 environment. The sensor node side programs run on MicaZ and TelosB motes, which are as resource constrained as body sensor motes. The MicaZ mote has an 8-bit 8-MHz

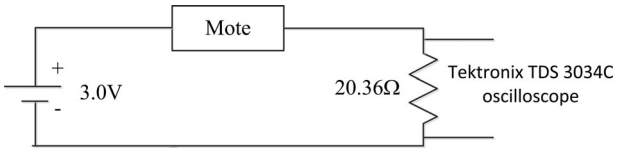


Fig. 3. Experimental setup for investigating the energy consumption.

Atmel microcontroller with 4-kB RAM and 128-kB ROM, while the TelosB mote is equipped with an 8-MHz MSP430 microcontroller, 10-kB RAM, 48-kB ROM, and 1-MB flash memory. Our motes run TinyOS 2.x. RC5, data encryption standard (DES), Triple-DES, and AES are all symmetric-key encryption/decryption algorithms. It is commonly known that AES is more efficient than all other algorithms for a given level of security. We have thus implemented software AES decryption [21] on MicaZ and TelosB motes. Unless otherwise stated, all reported results for PCs (respectively, sensor nodes) are average results of one million (respectively, 1000) repeated experiments.

To implement the proposed protocol, we add the following functionalities in the C tools on the base station side: construction of multiple one-way key hash chains, generation of all data packets. To obtain *version* of each data item, we modify *DisseminatorC* and *DisseminatorP* modules of Drip nesC library to provide an interface named *DisseminatorVersion*.

Following the design of our protocol described previously, we employ the *SHA-1 hash* operation of TinyECC 2.0 library [22] and software AES decryption operation [21] to add the verification function of data packets into the Drip nesC library. In our implementation, the base station (i.e., a laptop PC) first sends the data packets through a serial port to a particular sensor node which is referred to as *retransmitter*. Then, the retransmitter disseminates these packets by Drip or our protocol on behalf of the base station.

Adopting the approach in [23], we have built a simple circuit, shown in Fig. 3, for measuring the power consumption in the motes due to the involved cryptographic operations. First, we can obtain the voltage across the mote V_m by subtracting the voltage across the resistor V_r from the voltage of the battery V_b. As shown in Fig. 3, V_r can be accurately measured by the Tektronix TDS 3034C digital oscilloscope, and $V_b = 3$ V. The current through the circuit I is given by Ohm's law, i.e., $I = V_r/20.36$. The consumed power in the mote is therefore $V_m \times I$. In addition, once we have measured the execution time of the cryptographic operation, the corresponding energy consumption is given by the product of execution time and consumed power.

B. Evaluation Results

Four metrics are used to evaluate our proposed protocol: execution time, memory overhead, propagation delay, and energy overhead. The execution time measures the time duration for the deployed cryptographic operations. The memory overhead measures the amount of data space consumed by the real implementation. Propagation delay is defined as the time required for a round of disseminated data items to reach all nodes in the network.

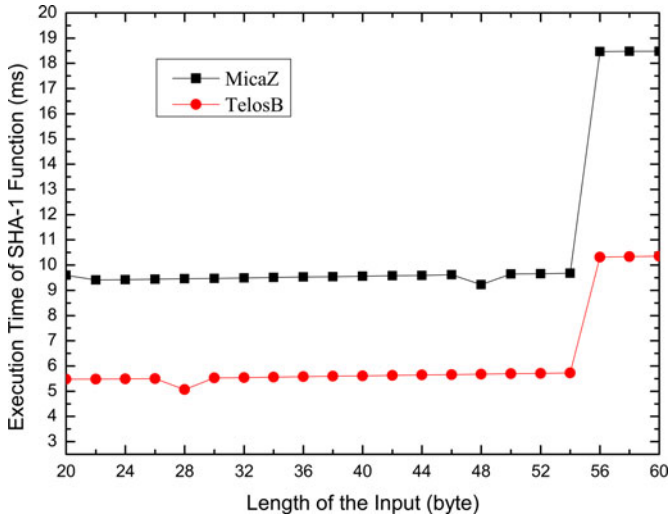


Fig. 4. Execution times of SHA-1 hash function on MicaZ and TelosB motes.

Fig. 4 shows the execution times of SHA-1 hash function (extracted from TinyECC 2.0 [22]) on MicaZ and TelosB motes. The inputs to the hash function are randomly generated numbers with length varying from 20 to 60 bytes in increments of 2 bytes. We perform the same experiment 10 000 times and take an average over them. For example, the execution times on a MicaZ mote for inputs of 54, 56, and 60 bytes are 9.6788, 18.4678, and 18.4775 ms, respectively. Also, the execution times on a TelosB mote for inputs of 54, 56, and 60 bytes are 5.7263, 10.3161, and 10.3571 ms, respectively. From Fig. 4, it can be seen that the execution time remains very stable when the byte length of the input falls in the interval $[0, 55]$.

Tables III and IV show the execution time of software AES encryption/decryption for MicaZ and TelosB motes with the plaintext (from P_i (e.g., the field $E(\{d_i, K_{i,1}\}, K_{i-1,1})$ of packet P_i) of different lengths, respectively. As expected, it can be seen from Tables III and IV that the time consumed by software AES implementation is extremely small. For example, even with 144-byte ciphertext as input, the decryption procedure takes 11.983 and 19.510 ms on MicaZ and TelosB motes, respectively.

To compare with the computation time of public key cryptography, we have also implemented the 160-bit ECC algorithm from the TinyECC 2.0 library [22] on MicaZ and TelosB motes. The signature verification time (using a random 20-byte number as the output) are measured to be 2.436 and 3.968 s, respectively. They are 914 and 909 times longer than software AES decryption operation with a 32-byte random ciphertext as input on MicaZ and TelosB motes, respectively. It is demonstrated that packet authentication based on the use of AES symmetric algorithm is much more efficient than that based on verifying the digital signature. Thus, our protocol is suitable for resource-limited sensor nodes.

Next, the energy consumption of ECC verification and SHA-1 hash function operations are investigated (when the radio of the mote is OFF). When a MicaZ mote is used in the circuit, $V_r = 138$ mV, $I = 6.7779$ mA, $V_m = 2.8620$ V, $P = 19.3983$ mW.

TABLE III
EXECUTION TIME OF SOFTWARE AES FOR A MICAZ MOTE WITH THE PLAINTEXT OF DIFFERENT LENGTHS

Plaintext length (bytes)	16	32	48	64	80	96	112	128	144	160
Encryption time (ms)	1.147	2.283	3.355	4.555	5.691	6.829	7.964	9.106	10.238	11.367
Decryption time (ms)	1.337	2.663	4.001	5.326	6.66	7.99	9.319	10.644	11.983	13.314

TABLE IV
EXECUTION TIME OF SOFTWARE AES FOR A TELOSB MOTE WITH THE PLAINTEXT OF DIFFERENT LENGTHS

Plaintext length (bytes)	16	32	48	64	80	96	112	128	144	160
Encryption time (ms)	1.838	3.645	5.46	7.254	9.061	10.864	12.676	14.491	16.287	18.079
Decryption time (ms)	2.193	4.359	6.517	8.675	10.843	13.006	15.171	17.343	19.510	21.673

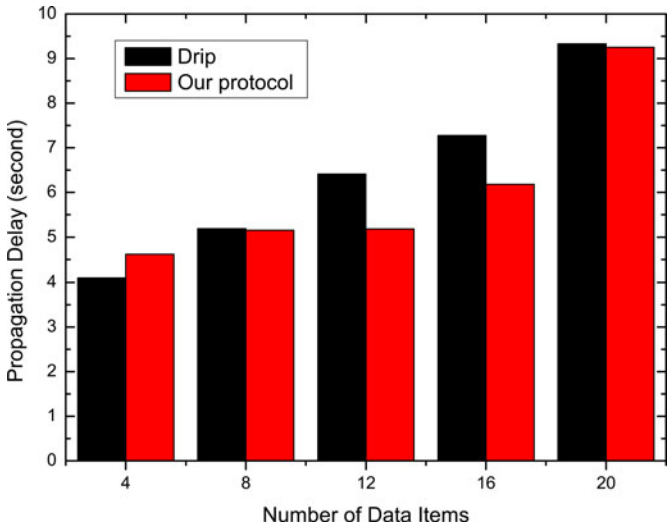


Fig. 5. Propagation delay comparison of two protocols.

When a TelosB mote is used, $V_r = 38$ mV, $I = 1.8664$ mA, $V_m = 2.9620$ V, $P = 5.5283$ mW. By multiplying the power with the execution time obtained from Fig. 4, we can determine the total energy consumption of SHA-1 operation on the motes. For example, the energy consumption of SHA-1 operation with a random 56-byte number as input on MicaZ and TelosB motes are 0.3582 and 0.0317 mJ, respectively. Also, the energy consumption of ECC signature verification operation on MicaZ and TelosB motes are 2835.2555 and 1316.1777 mJ, respectively.

Next, we investigate the impact of security functions on the propagation delay by carrying out experiments in a testbed, comprising 24 TelosB motes arranged in a 4×6 grid. The space between each node is about 35 cm and the transmission power of each node is adjusted to the level such that only one-hop neighbors are within the transmission range. The role of retransmitter is played by the node at the vertex of the grid. Also, in order to investigate the mobility of biosensor nodes, the blackboard moves round.

In this experiment, the base station disseminates data items to the TelosB motes in the grid. The packet delivery rate at the base station is 5 packets/s. Fig. 5 shows the propagation delays of Drip and our protocol measured from the experiment. The propagation delay of our protocol is the time from the construction of multiple one-way key hash chains, until the corresponding variables on all nodes are updated. In our implementation, the lengths of *order* and $data$ in each data item are 4 bits and 2 bytes, respectively. In order to obtain accurate average results, for each experiment, we have executed each dissemination operation for 20 times and taken an average over them. As the number of data items increases, the propagation delays of all protocols increase almost linearly. For example, when the number of data items is 4 (respectively, 12), the propagation delays of Drip and our protocol are 4.099 s (respectively, 4.615 s) and 6.413 s (respectively, 5.181 s), respectively. From Fig. 5, it is concluded that the packet authentication by sensor nodes in our protocol has very low impact on the propagation delay of dissemination.

TABLE V
CODE SIZES (BYTES) ON MICAZ AND TELOSB MOTES

		Drip	Our protocol	Software AES decryption in our protocol
MicaZ	ROM	14,756	22,060	3,642
	RAM	423	2,681	1,888
TelosB	ROM	14,808	20,232	4,140
	RAM	453	2,711	1,888

Table V shows the ROM and RAM usage of our protocol on MicaZ and TelosB motes. The code sizes of Drip (with four data items) and the software AES decryption function are also included for reference purposes. For example, the implementation of our protocol on a TelosB mote uses 2711 bytes of RAM and 20 232 bytes of ROM, respectively. The resulting size of our implementation corresponds to 26.47% and 41.16% of the RAM and ROM capacities of TelosB, respectively.

VIII. CONCLUSION

In this paper, we have identified the security vulnerabilities in data discovery and dissemination of WBANs. Based on the unique features and application requirements of a WBAN, we then proposed a lightweight and confidential data discovery and dissemination protocol. It uses low-complexity symmetric cryptographic techniques for maintaining confidentiality. Also, it changes the encryption key on a per-packet basis to prevent the intermediate nodes from forging the keys, ensuring authenticity of the broadcast data items. Our solution conforms with the resource limitations of a WBAN. In addition to analyzing the security of our protocol, this paper has also reported the evaluation results of our protocol in an experimental network of resource-limited sensor nodes, which show that our protocol is efficient and feasible in practice. We have also given a formal proof of the authenticity and integrity of the disseminated data items in our protocol.

REFERENCES

[1] The Health Information Trust Alliance (HITRUST), [Online]. Available: http://www.hitrustalliance.org

[2] The US Congress. (1996) Health Insurance Portability and Accountability Act. Washington D.C., USA, [Online]. Available: http://www.hhs.gov/ocr/privacy/

[3] European Union. (1995 Oct. 24) Directive 95/46/EC of the European Parliament and of the Council on the Protection of Individuals with Regard to the Processing of Personal Data and on the Free Movement of Such Data. [Online]. Available: http://www.refworld.org/docid/3ddcc1c74.html

[4] G. Tolle and D. Culler, "Design of an application-cooperative management system for wireless sensor networks," in *Proc. Eur. Conf. Wireless Sensor Netw.*, 2005, pp. 121–132.

[5] K. Lin and P. Levis, "Data discovery and dissemination with DIP," in *Proc. ACM/IEEE Int. Conf. Inform. Process. Sensor Netw.*, 2008, pp. 433–444.

[6] T. Dang, N. Bulusu, W. Feng, and S. Park, "DHV: A code consistency maintenance protocol for multi-hop wireless sensor networks," in *Proc. Eur. Conf. Wireless Sensor Netw.*, 2009, pp. 327–342.

[7] P. Levis, N. Patel, D. Culler, and S. Shenker, "Trickle: A self-regulating algorithm for code maintenance and propagation in wireless sensor networks," in *Proc. Netw. Syst. Design Implementation*, 2004, pp. 15–28.

[8] D. He, J. Bu, S. Zhu, S. Chan, and C. Chen, "Distributed access control with privacy support in wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 10, no. 10, pp. 3472–3481, Oct. 2011.

[9] A. Perrig, R. Canetti, D. Song, and J. Tygar, "Effcient and secure source authentication for multicast," in *Proc. Netw. Distributed Syst. Security Symp.*, 2001, pp. 35–46.

[10] D. Liu and P. Ning, "Multi-level utesla: Broadcast authentication for distributed sensor networks," *ACM Trans. Embedded Comput. Syst.*, vol. 3, no. 4, pp. 800–836, 2004.

[11] Y. Chen, I. Lin, C. Lei, and Y. Liao, "Broadcast authentication in sensor networks using compressed bloom filters," in *Proc. IEEE Distributed Comput. Sensor Syst.*, 2008, pp. 99–111.

[12] D. He, S. Chan, S. Tang, and M. Guizani, "Secure data discovery and dissemination based on hash tree for wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 12, no. 9, pp. 4638–4646, Sep. 2013.

[13] Z. Zhang, H. Wang, A. Vasilakos, and H. Fang, "ECG-cryptography and authentication in body area networks," *IEEE Trans. Inf. Technol. Biomed.*, vol. 16, no. 6, pp. 1070–1078, Nov. 2012.

[14] K. Malasri and L Wang, "Design and implementation of a secure wireless mote-based medical sensor network," *Sensors*, vol. 9, no. 8, pp. 6273–6297, 2009.

[15] C. C. Tan, H. Wang, S. Zhong, and Q. Li, "IBE-Lite: A lightweight identity-based cryptography for body sensor networks," *IEEE Trans. Inf. Technol. Biomed.*, vol. 13, no. 6, pp. 926–932, Nov. 2009.

[16] C. Kuo, M. Luk, R. Negi, and A. Perrig, "Message-in-a-bottle: User-friendly and secure key deployment for sensor nodes," in *Proc. ACM SenSys*, 2007, pp. 233–246.

[17] M. Bellare, A. Desai, E. Jokipii, and P. Rogaway, "A concrete security treatment of symmetric encryption," in *Proc. ACM Foundations Comput. Sci.*, 1997, pp. 394–403.

[18] M. Naor and M. Yung, "Universal one-way hash functions and their cryptographic applications," in *Proc. ACM Symp. Theory Comput.*, 1989, pp. 33–43.

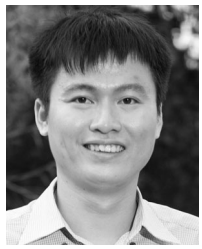
[19] C. Karlof, N. Sastry, and D. Wagner, "TinySec: a link layer security architecture for wireless sensor networks," in *Proc. ACM SenSys*, 2004, pp. 162–175.

[20] OpenSSL. [Online]. Available: http://www.openssl.org

[21] [Online]. Available: http://tinyos.cvs.sourceforge.net/viewvc/tinyos/tinyos-2.x-contrib/crypto/index.html

[22] A. Liu and P. Ning, "TinyECC: A configurable library for elliptic curve cryptography in wireless sensor networks," in *Proc. ACM/IEEE Int. Conf. Inform. Process. Sensor Netw.*, 2008, pp. 245–256.

[23] J. Lee, K. Kapitanova, and S. Son, "The price of security in wireless sensor networks," *Comput. Netw.*, vol. 54, no. 17, pp. 2967–2978, Dec. 2010.



Daojing He (S'07–M'13) received the B.Eng. and M. Eng. degrees from the Harbin Institute of Technology, Harbin, China in 2007 and 2009, respectively, and the Ph.D. degree from Zhejiang University, Hangzhou, China, in 2012, all in computer science.

He is currently an Associate Professor in the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China. His research interests include network and systems security.

Dr. He is an Associate Editor or on the editorial board of some international journals such as *IEEE Communications Magazine*, *Springer Journal of Wireless Networks*, *Wiley's Wireless Communications and Mobile Computing Journal*, *Wiley's Security and Communication Networks Journal*, and *KSII Transactions on Internet and Information Systems*. He has been serving as a TPC member for leading conferences including the IEEE Wireless Communications and Networking Conference, the IEEE Global Communications Conference, and the IEEE International Conference on Communications.

Sammy Chan (S'87–M'89) received the B.E. and M.Eng.Sc. degrees in electrical engineering from the University of Melbourne, Parkville, Vic., Australia, in 1988 and 1990, respectively, and the Ph.D. degree in communication engineering from the Royal Melbourne Institute of Technology, Melbourne, Vic., Australia, in 1995.

From 1989 to 1994, he was with Telecom Australia Research Laboratories, first as a Research Engineer, and between 1992 and 1994 as a Senior Research Engineer and a Project Leader. Since December 1994, he has been with the Department of Electronic Engineering, City University of Hong Kong, Hong Kong, where he is currently an Associate Professor.

Yan Zhang (M'05–SM'10) received the Ph.D. degree from Nanyang Technological University, Singapore.

He is working with Simula Research Laboratory, Norway, and he is an adjunct Associate Professor at the University of Oslo, Oslo, Norway. His research interests include resource, mobility, spectrum, energy, and data management in communication networks.

Haomiao Yang (M'12) received the M.S. and Ph.D. degrees in computer applied technology from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2004 and 2008, respectively.

From 2012 to 2013, he was a Postdoctoral Fellow at Kyungil University, Gyeongsangbuk-do, Korea. He is currently an Associate Professor at the School of Computer Science and Engineering, UESTC, China. His research interests include cryptography, cloud security, and big data security.