Exploiting Prediction to Enable Secure and Reliable Routing in Wireless Body Area Networks

Xiaohui Liang†, Xu Li‡, Qinghua Shen†, Rongxing Lu†, Xiaodong Lin*,
Xuemin (Sherman) Shen†, and Weihua Zhuang†
†Department of Electrical and Computer Engineering, University of Waterloo, Canada
‡ INRIA Lille - Nord Europe, Univ Lille Nord de France, USTL, CNRS UMR 8022, LIFL, France
*Faculty of Business and Information Technology, University of Ontario Institute of Technology, Canada
Email: {x27liang, q2shen, rxlu, xshen, wzhuang}@bbcr.uwaterloo.ca; xu.li@inria.fr; xiaodong.lin@uoit.ca

***Abstract*—In this paper, we propose a distributed Prediction-based Secure and Reliable routing framework (PSR) for emerging Wireless Body Area Networks (WBANs). It can be integrated with a specific routing protocol to improve the latter's reliability and prevent data injection attacks during data communication. In PSR, using past link quality measurements, each node predicts the quality of every incidental link, and thus any change in the neighbor set as well, for the immediate future. When there are multiple possible next hops for packet forwarding (according to the routing protocol used), PSR selects the one with the highest predicted link quality among them. Specially-tailored lightweight source and data authentication methods are employed by nodes to secure data communication. Further, each node adaptively enables or disables source authentication according to predicted neighbor set change and prediction accuracy so as to quickly filter false source authentication requests. We demonstrate that PSR significantly increases routing reliability and effectively resists data injection attacks through in-depth security analysis and extensive simulation study.**

***Keywords*—Wireless body area networks; routing; prediction; reliability; security; data injection attacks**

I. Introduction

Recent advances in microcircuits and medical sensing have made it possible to deploy battery-powered miniaturized sensors on, in or around the human body for long-term healthcare monitoring [1]. These body sensors report their sensory data to a data sink via wireless communication channels. The data sink is a pre-defined portable device, such as a PDA or a cell phone worn on the human body. It may be linked to a remote healthcare agency through the cellular network and/or the Internet, for example. The body sensors and the sink together constitute a small-scale wireless sensor network (WSN), called wireless body area network (WBAN). It enables real-time health-related information to be provided to emergency medical specialists, who are then enabled to deliver appropriate and timely medical treatment to the monitored patients. WBANs are particularly suitable for monitoring people having chronic diseases or working and living under extreme conditions.

Although WBANs are deployed in a compact spatial region (along the human body), multi-hop communication rather than single-hop is their main communication pattern. Previous research [2]–[4] indicated that, due to the energy absorption of the human body, the physical channels of WBANs have much higher path loss than those in free space propagation especially when the communication is non-line of sight (NLOS); this communication, for example, occurs when the sender is placed on the back and the receiver on the chest. Alternatively, high power radio frequency (RF) enforced throughout a large coverage area cannot be used in WBANs because RF energy waves may heat and damage body tissue by energy absorption. This consideration implies that in WBANs, multi-hop communication has advantages and sometimes is an absolute requirement. The experimental study in [5] further confirms that multi-hop communication is most reliable in WBANs.

It is rather straightforward to perform multi-hop routing in a small-scale static WSN environment. WBANs are small in size, but they are composed of nodes that move along body gestures. Node mobility leads to dynamically changing network topology and significantly varying RF energy absorption (thus link quality), rendering routing a challenging task. The openness of the wireless media makes it easy for a malicious adversary to launch various security attacks and violate the basic security requirements, i.e., data confidentiality, authenticity, integrity and non-repudiation. This problem exists in any wireless network, but it is more serious in WBANs because the network traffic is health-related, highly personal and user-sensitive [6]–[8]. Pure cryptographic security solutions are often computation-intensive and vulnerable to *data injection attacks* given that body sensors have restricted resources.

A data injection attack aims to consume the resources of a target network node by sending false data to it. For example, the attacker may eavesdrop the communication transactions of the target node, retrieve useful authentication information and use it to send false packets to the target node. Without precaution, the target node will put intensive efforts to respond to the false packets; even worse, it may retransmit them to other nodes. In energy-constrained WBANs, data injection attacks can exhaust sensor battery power quickly and reduce network lifetime. Sensors should be intelligent enough to recognize and reject false data at minimal cost. Cryptography alone is not sufficient to solve this security problem.

To ensure secure and reliable routing (toward the data sink) in WBANs, the following two requirements must be satisfied, in addition to the aforementioned basic security requirements:

- Localized reliable data forwarding: A node should be able to select an incidental link to forward data packets, which

is likely to have high quality in the immediate future.
- Resilience against data injection attacks: A node should be able to avoid processing false and/or irrelevant data injected into the network during a short period of time.

In this paper, we address the two requirements by proposing a novel distributed Prediction-based Secure and Reliable routing framework (PSR), and thus, persistent data injection attacks, regarded as notorious Denial-of-Service (DoS) attacks, and other robust and exhaustive adversaries are beyond our scope.

It is observed that body sensors may exhibit regular mobility when a user's physical activity (e.g., swimming and jogging) contains repeated motions, and as a result, link quality and a neighbor sensor set often present periodic changes. This observation serves as the foundation of our proposed PSR routing framework, which can be combined with any specific WBAN routing protocol to increase the latter's security and reliability performance. By employing PSR, each node maintains an autoregressive (AR) model [9] for every neighbor, based on the link quality measurements (characterized by the received signal power at the other side of the links) between them. Using this model, it predicts the quality of its incidental links as well as the change of its neighbor set.

By the underlaying routing protocol, a node selects a subset of incidental links that can be used to forward packets to the data sink; among these links, it chooses the one that has the highest predicted quality as routing next hop. Each node is equipped with two novel authentication mechanisms specifically devised for source authentication and data authentication. It performs lightweight hash-based data authentication for every received data packet; it disables relatively computation-intensive source authentication if its neighbor set is not changing according to the prediction results in order to save computational resources, or enable source authentication otherwise. The logic is that, if the neighbor set is not changing, source authentication will not be necessary since the existing neighbors have already been authenticated. To the best of our knowledge, this prediction-based security technique is proposed for the first time in this paper.

Through in-depth analysis, we demonstrate that PSR is resilient against data injection attacks. We evaluate the performance of PSR through extensive simulation. In the simulation, PSR is implemented by being integrated with a hop-count based greedy routing procedure and compared with a previous static tree-based routing protocol [10]. Our simulation results indicate that PSR has significantly (up to 80%) lower packet dropping rate and (up to 50%) shorter routing delay, and that it is able to quickly identify and filter over 70% of the false source authentication requests, effectively resisting data injection attacks.

The remainder of this paper is organized as follows. Section II reviews some related work, and Sec. III defines frequently-used notations, the network model and the security model. Section IV introduces a prediction model and security initialization. PSR is proposed in Sec. V, and its security analysis and performance evaluation are presented respectively in Sec. VI and VII. The closing remarks are given in Section VIII.

II. Related work

Research has been carried out for efficient data communications in accordance with the multi-hop architecture of WBANs. For example, Latre et al. [10] aim to improve energy efficiency and communication reliability by activity scheduling. In their solution, a static routing tree is built in advance, and time is slotted. Time slots are assigned to sensors according to the routing tree, which then transmit their data only during their assigned time slots. The assignment is carefully done in a hierarchical and distributed manner to reduce idle listening and minimize signal interference. Quwaider et al. [11] propose a dynamic power assignment, that is, to determine the necessary transmission power for each wireless link. In their solution, a sender transmits every packet together with the information about the transmission power used. A receiver node measures the received signal strength and decides whether the transmission power is too large or too small, and it informs the sender to adjust the transmission power accordingly, improving communication reliability.

A few researchers focus on efficient routing algorithms, which are the foundation of data communication. They exploit the delay tolerant network (DTN) concept and postural information [4], [12], [13]. In [12], the authors aim to minimize end-to-end delay by avoid using nodes that have a high storage/buffering delay due to topological disconnections. They develop a probabilistic distance-vector packet based routing algorithm. This algorithm uses a stochastic link cost formulation to capture multi-scale topological localities in human postural movements. It assumes that, if a link is connected in current time slot, the probability that the link will remain connected in the next time slot increases at a fixed rate. This assumption may not hold however in reality. In [13], a few variants of DTN routing are presented in the WBAN context. They implicitly assume that a link has constant quality (by ignoring link quality different in routing), which as we previously discussed is not reasonable due to node mobility and RF energy absorption, and they require each node to have possibly unrealistic restricted mobility, intermittently coming within up to 2-hop distance from the sink.

In this paper, we propose a novel routing framework PSR for WBANs. Unlike the aforementioned previous work, PSR does not rely on any of these strong assumptions and adds link quality as a routing metric. It uses a well-established prediction model to predict link quality. The prediction result is exploited in relay node selection to improve routing reliability and in data transmission to resist data injection attacks. We will show that PSR is able to effectively resist these attacks through security analysis in Sec. VI and performance evaluation in Sec. VII.

III. Notations and models

Before proceeding further, we define the network model and the security model that PSR is to be developed upon. A non-exhaustive list of notations to be used throughout the rest of the paper can be found in Table I.

TABLE I
FREQUENTLY USED NOTATIONS

S	a set of s body sensor nodes $\{n_1, n_2, \cdots, n_s\}$
T_c	current time slot
λ	the length of a single time slot
$\mathcal{H}$	$\{(i, h_i)_{1 \le i \le s}\}$ represent (index i, hop count h_i)
(i, k_i)	node index and the corresponding secret key
$S_{i,j}$	a secret key shared by nodes n_i and n_j
$\mathcal{M}_i$	a matrix containing link quality measurements
H_1, H_2, H_a, H_b	four cryptographic secure hash functions
(d, m)	a hash seed and a positive even integer
$\mathcal{N}_j^c$	a real neighbor set at the end of T_c
$\hat{\mathcal{N}}_j^c$	a predicted neighbor set at the beginning of T_c

A. Network model

Consider a WBAN composed of s body sensors. We denote by $\mathcal{S} = \{n_1, n_2, \cdots, n_s\}$ the sensor set and n_0 the sink. Every sensor is associated with a unique identifier or index such as MAC address or manufacturer serial number by which it can be distinguished from others. Two nodes are neighbors if they are within each other's communication range. Each node has some fixed neighbors to which it has a constant distance along the surface of human body in spite of body gestures. For example, a node placed on a wrist may be a fixed neighbor of a node placed on the elbow of the same arm, and vice versa. A communication link between two fixed neighbors is called *backbone link*. The backbone links alone connect all the nodes together. Considering the critical nature of WBAN applications, these backbone links are necessary in order to guarantee connectivity. Also, it is feasible to establish these links since a WBAN is usually deployed manually.

A shortest path tree rooted at the data sink n_0 is constructed using backbone links, as shown in Fig. 1. Along this tree, the hop count information $\mathcal{H} = \{(i, h_i)_{1 \le i \le s}\}$ is obtained, where i is node index and h_i the hop count from n_i to n_0, and distributed to each sensor node. Although any existing WBAN routing protocol may be applied on individual nodes for identifying routing next hop candidates, for simplicity we use a greedy routing protocol based on the established hop count information to present and evaluate PSR. We do not use real-time hop count information for two reasons: i) maintaining such information is costly when the network topology is changing, and ii) delay induced by using non-shortest paths is not a major concern in such a small-scale network. The logic of greedy forwarding is to move a packet to a node closer to n_0 than the node currently holding it.

Time is locally slotted by nodes with equal length λ, which is a positive real number, the same for all the nodes. At the beginning of each time slot, n_i chooses appropriate routing next hop and authentication policies to follow for the current time slot; at the end of each time slot, it adjusts a few system parameters for better decision making in the next time slot.

B. Security model

WBANs are subject to both internal attacks and external attacks. Here we consider only external attacks, which are launched by adversaries outside the network. DoS attacks are out of the scope of this work. We do not consider lower-layer jamming attacks that block the traffic between two neighboring nodes. Encryption (e.g. symmetric approaches) and hashing prevent eavesdropping attacks and data modification attacks at low cost. Signature approaches realize authenticity and non-repudiation, but with large computation overhead.

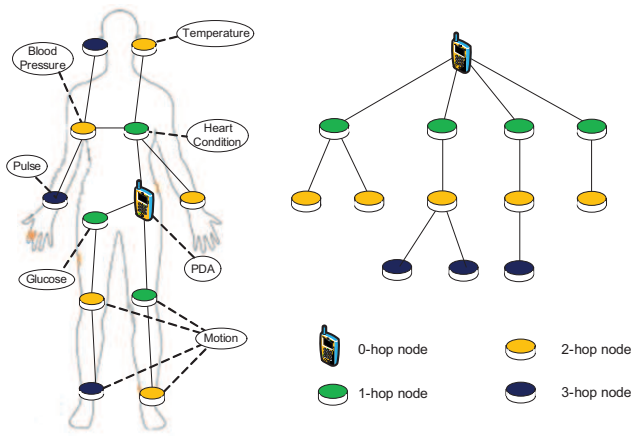


Fig. 1. Shortest path tree based on backbone links

Thus we focus on network-layer data injection attacks that make use of the weakness of signature-based authentication to exhaust sensors' computational resources including CPU cycle and battery power. If nodes are unable to resist such attacks, the network will be paralyzed, and the network lifetime will be shortened. Data injection attacks can be launched in the following three forms:

- Exhaustive source authentication attacks, where the attacker repeatedly sends false authentication requests;
- Exhaustive data authentication attacks, where the attacher continuously sends false data packets to a node;
- Data replay attacks, where the attacker uses eavesdropped security materials to inject forged data packets.

IV. PRIMITIVES

In this section, we introduce autoregression and a security initialization that will be used in PSR.

A. Autoregression

The autoregressive (AR) model is a tool for understanding and predicting a time series of data [14]. It can be used to estimate the current term z_k of the series by a linear weighted sum of previous p terms (i.e., observations) in the series. The model order p is generally less than the length of the series. Formally, $\mathsf{AR}(p)$ is defined as

$$z_k = c + \sum_{i=1}^{p} \phi_o z_{k-i} + \epsilon_k,$$

where c is a constant standing for the mean of the series, ϕ_i autoregression coefficients, and ϵ_k the zero-mean Gaussian white noise error term. For simplicity, c is often omitted.

The derivation of $\mathsf{AR}(p)$ involves determining the coefficients ϕ_i for $i \in [1 \cdots p]$ that give a good prediction. The model can be updated continuously as new samples arrive so as to ensure accuracy, or it may be recomputed when the

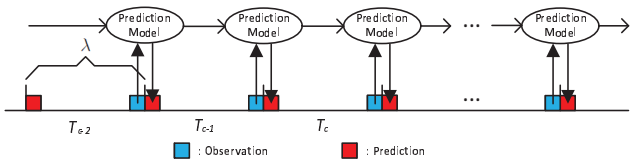


Fig. 2. Prediction model update along time axis

prediction error, i.e., the difference between the predicted value and the true measurement, is very large. In [9], a simplified AR model is presented and used for neighborhood prediction. This model can be updated through trivial calculus, greatly reducing the requirement on the computational power of the nodes that implement it. Thus, it becomes embeddable on tiny and computationally weak body sensors, and we adopt the model later in PSR for link quality prediction.

B. Security initialization

Initially, system parameters are configured and embedded in every node as follows. At the first step, given a security parameter $k \in \mathbb{Z}^+$, the administrator runs a bilinear pairing generator [15] on input k to generate a prime q, two groups $\mathbb{G}_1, \mathbb{G}_2$ of order q, and an admissible bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$. The administrator then chooses a random generator $P \in \mathbb{G}_1$, a random $s \in \mathbb{Z}_q^*$, and four cryptographic hash function $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1^*$, $H_2 : \{0,1\}^* \rightarrow \mathbb{Z}_q^*$, and $H_a, H_b : \{0,1\}^* \rightarrow \{0,1\}^*$. The administrator computes $P_{pub} = sP$ and sends PP $= \langle q, \mathbb{G}_1, \mathbb{G}_2, e, P, P_{pub}, H_1 \rangle$ to the nodes and the sink. At the second step, for n_i, the administrator computes $Q_i = H_1(n_i) \in \mathbb{G}_1^*$, and sets its private key $k_i = sQ_i$. For the sink node n_0, it also computes $Q_0 = H_1(n_0)$ and a private key $k_0 = sQ_0$. All the nodes keep their private keys secretly. At the third step, the administrator sends hop count information $\mathcal{H}$ to every node.

The session key $S_{i,j}$ between nodes n_i and n_j can be non-interactively calculated as $S_{i,j} = e(H_1(n_i), H_1(n_j))^s = e(k_i, H_1(n_j)) = e(H_1(n_i), k_j)$ by using bilinear pairing property. Then, if n_i uses a symmetric key encryption scheme E to encrypt data with $S_{i,j}$ and sends the ciphertext $C = E(S_{i,j}, data)$ to n_j, n_j can decrypt C by a symmetric key decryption scheme D and obtains $data = D(S_{i,j}, C)$. Since $S_{i,j}$ is known only to nodes n_i and n_j, n_j is able to secretly obtain $data$ and check if the ciphertext C is generated by n_i. Note that we do not adopt a simple setting in which all the nodes share the same key. The reason is as follows. Nodes could be compromised (such situation is not considered here though) and reveal the key to attackers, putting the entire network at risk [16].

V. Prediction Based Secure and Reliable Routing

In this section, we propose a novel distributed Prediction-based Secure and Reliable routing framework (PSR) for WBANs. PSR can be integrated with any routing protocol to improve the latter's reliability and security performance. It is composed of two sub-algorithms *Next-hop selection* and *Data transmission*, both of which employ prediction-based techniques to help nodes make decisions on routing and data processing. As shown in Fig. 2, at the beginning of each time slot, nodes use the link quality measurements collected in the past time slots to predict neighborhood conditions (link quality and neighbor set) in the current time slot and run the two algorithms with respect to the prediction results, and at the end of each time slot, they use real conditions measured during the time slot to update the prediction model.

$\mathcal{M}_i$ at the beginning of time slot T_{c+1}

	T_{c-p}	T_{c-p+1}	$\cdots$	T_{c-1}	T_c
n_0	$q_{i,0}(c-p)$	$q_{i,0}(c-p+1)$	$\cdots$	$q_{i,0}(c-1)$	$q_{i,0}(c)$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
n_{i-1}	$q_{i,i-1}(c-p)$	$q_{i,i-1}(c-p+1)$	$\cdots$	$q_{i,i-1}(c-1)$	$q_{i,i-1}(c)$
n_{i+1}	$q_{i,i+1}(c-p)$	$q_{i,i+1}(c-p+1)$	$\cdots$	$q_{i,i+1}(c-1)$	$q_{i,i+1}(c)$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
n_s	$q_{i,s}(c-p)$	$q_{i,s}(c-p+1)$	$\cdots$	$q_{i,s}(c-1)$	$q_{i,s}(c)$

$\mathcal{M}_i$ at the beginning of time slot T_c

Fig. 3. Link quality matrix $\mathcal{M}_i$ of node n_i

Here, we present PSR with autoregressive model [9] being used for prediction due to its simplicity. But nevertheless, it can be replaced with any other prediction model as needed. Below we elaborate the two sub-algorithms with respect to an arbitrary sensor node n_i and an arbitrary time slot T_c.

A. Next hop selection

Node n_i maintains matrix $\mathcal{M}_i(s \times p)$ that stores the link quality measurements between itself and every other node in the network for the immediate past p time slots. Here p is a pre-defined system parameter. Link quality is characterized by the received signal power at the receiver side. In this matrix, each row corresponds to a unique node; the k-th column indicates the link quality between n_i and the other s nodes in $T_{c-p+k-1}$ (the current time slot is T_c). The matrix is initially empty. It is possible that some rows remain to have only 0 values since the corresponding node may have never been neighboring with n_i during the p time slots. Because a WBAN is a small-scale network of only a few nodes, it is feasible that each node maintains such a link quality matrix. Figure 3 comparatively shows $\mathcal{M}_i$ at the beginning of T_c and T_{c+1}.

Based on this link quality matrix, n_i builds an order-p autoregressive model. At the beginning of T_c, using this model n_i predicts the link quality with every other node, and it chooses a neighbor that has the best predicted link quality among those closer to the sink than itself as next hop (greedy forwarding). If the prediction model is not established yet, the backbone-link based shortest path tree will be used conservatively for packet forwarding as the backbone links have relatively stable quality. In the sequel, n_i transmits every data packet with the selected next hop as a designated receiver.

All the neighbors hear the data transmission of n_i and measure the received signal power (i.e., link quality). They then reply n_i with an acknowledgement (ACK) carrying the measurements whether they are the intended receiver or not.

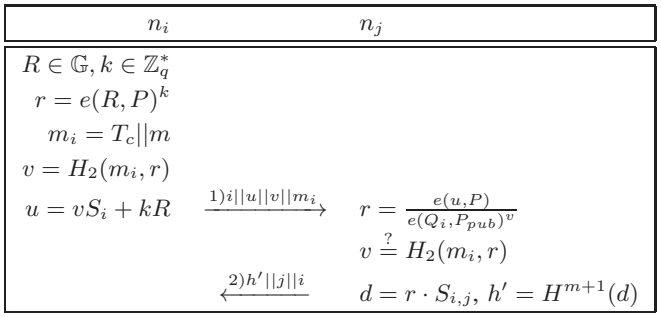


Fig. 4. Source authentication

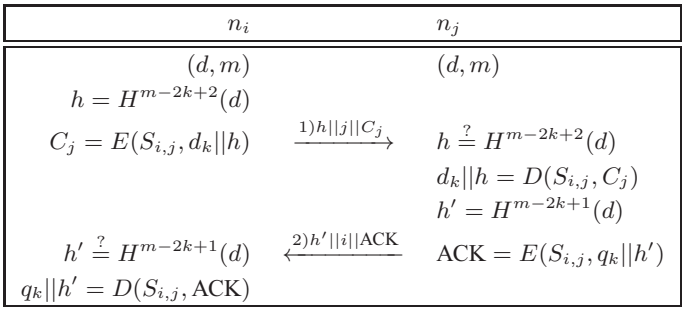


Fig. 5. Data authentication

A detailed description of data transmission and acknowledging is presented in the next subsection. By receiving ACKs from neighboring nodes, n_i knows the average quality of the incidental links to them during T_c and updates $\mathcal{M}_i$ with the average results at the end of T_c. Note that, if an expected ACK does not arrive from a node, n_i will consider the corresponding link quality measurement to be $-\infty$.

B. Data transmission

Node n_i shares with another node n_j a set of secret tokens if they have successfully authenticated each other. For each data packet to be sent, n_i checks if it has a valid token with every n_j in the network. Having a valid token with n_j means being recognized by it. Thus, n_i starts the data authentication immediately if the check results are all positive. Otherwise, it has to first start source authentication with the n_js for which the check results are negative. To tolerate occasional transmission failure, n_i initiates source authentication up to the maximum number of times. After all the authentication retrials or after having a valid token with every other node, n_i proceeds with the data transmission.

The set of tokens shared between n_i and n_j are a sequence of hash values, like $\langle H^m(d), H^{m-1}(d), \cdots, H(d)\rangle$, where H is a function defined as $H^{2k}(d) = H_a^k(d)$ and $H^{2k-1}(d) = H_b^k(d)$ for any integer $k \geq 1$ (refer to Sec. IV-B for the definition of functions H_a and H_b). The token set is therefore partitioned evenly into two disjoint portions used respectively by n_i and n_j for authenticating packets. In each data transmission, n_i attaches a single token from its portion to the data packet. The token is placed at the beginning of the packet if n_j is the next hop, or at the end otherwise. Tokens are used one by one in a pre-defined order; once used, they are no longer secrets and become invalid for future use.

Every data packet sent by n_i contains a token for every neighbor n_j, which is therefore able to authenticate the packet. This is because that the valid tokens are secrets shared only between n_i and n_j, and outside attackers can obtain valid tokens only if data transmission failure happens (the analysis can be found in Sec. VI). For each authenticated data packet from n_i, n_j identifies whether or not it is the intended receiver (i.e., the next hop) and responsible for packet forwarding by checking the token's position in the packet, and it also replies n_i with an ACK packet, enabling n_i to measure the quality of the link between them. The ACK packet is authenticated similarly using a token from n_j's portion of the token set.

1) Source authentication: The center of data transmission is obviously the processes of source authentication and data authentication. We first elaborate source authentication, which enables two neighboring nodes n_i and n_j to authenticate each other. Figure 4 shows a source authentication process between these two nodes in T_c. It consists of two steps. At the first step, n_i broadcasts $i||u||v||m_i$ as a source authentication request to all the neighboring nodes. Here, m_i contains the current time slot T_c and the number m of tokens to be generated, and the pair (u, v) is an identity based signature on message m_i using the signature scheme [17]. Each neighbor n_j then verifies the signature by computing $r = \frac{e(u,P)}{e(Q_i,P_{pub})^v}$ and checking if $v{=}H_2(m_i, r)$. If the equality holds, n_j accepts the signature and replies n_i with $h'||j||i$ at the second step; otherwise, it stops the authentication process. Here, $h' = H^{m+1}(d)$ is a hash value, where $d = r \cdot S_{i,j}$. From security initialization (see Sec. IV-B), since $S_{i,j}$ is only known by n_i and n_j, they are able to calculate d. In the source authentication request, if n_i receives h', n_i knows that n_j must have received the request and then reveals h' in T_c. Notably, r is a random value and d will be generated independently for different source authentications. They will not use d in data transmissions and therefore others will not be able to calculate $S_{i,j} = d/r$. After a successful source authentication, nodes n_i and n_j agree upon the use of (d, m) for data authentications.

2) Data authentication: It is carried out for every packet and enables receiver n_j to ascertain that a packet is indeed from sender n_i as it claims to be. By source authentication, (d, m) are established and recorded by both nodes n_i and n_j. After the source authentication, it is required that n_i use token $H^{m-2k+2}(d)$ for sending the k-th data packet d_k to n_j and n_j then uses token $H^{m-2k+1}(d)$ for sending back the corresponding ACK to n_i. The sequence information k is contained in d_k. This data authentication process is illustrated in Fig. 5. It consists of two steps. At the first step, n_i sends $h||j||C_j$ to n_j, where $h = H^{m-2k+2}(d)$ is a valid token and C_j is a ciphertext of the combination of d_k and h. At the second step, n_j checks whether the embedded token is used for the first time. It is able to do the check because it knows all the used tokens. If the token is indeed used for the first time, n_j proceeds to decrypt C_j. If the h obtained by the decryption equals to the one outside C_j, then n_j believes in the integrity of the data packet and replies n_i with $h'||i||$ACK, where $h' = H^{m-2k+1}(d)$ is a valid token and ACK indicates

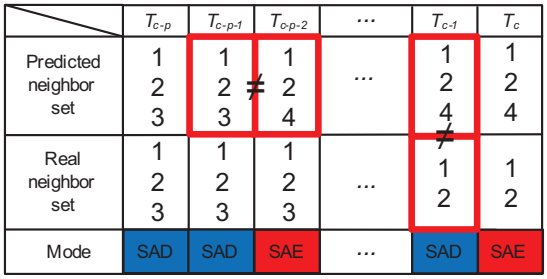


Fig. 6. Neighbor set

the successful recipient of d_k and reports the link quality q_k. If any of the above checks fails, n_j stops the process and ignores the packet. After receiving the ACK, n_i performs similar checks and retrieves q_k. Note that tokens h and h' can be only used for the k-th data packet of n_i after the last source authentication. If $m-2k+2 \leq 0$, n_i has to start a new source authentication process with n_j for a new tuple (d, m) in T_{c+1}. Further, if n_i does not receive any ACK with valid tokens from n_j within T_c, it marks all the unused tokens with n_j invalid, and a new source authentication is needed in T_{c+1}.

The data authentication process between n_i and n_j indicates that each data authentication consumes a token pair (h, h') in the shared token set between n_i and n_j. In fact, each data authentication has to consume a token pair between n_i and its every neighbor in order for n_i to be able to measure the link quality with them. Suppose that n_i has k neighbors $\{n_i^1, n_i^2, \cdots, n_i^k\}$ in addition to the next hop n_j. Let (h_l, h_l') be the token pair between n_i and neighbor $n_i^l, 1 \leq l \leq k$, which are respectively from the token sets that n_i shares with those neighbors. At the first step of data authentication, n_i attaches tokens $h_1, h_2, \cdots, h_k$ to the end of a data packet, i.e., $h||j||C_j||h_1||\cdots||h_k$; at the second step, neighbor $n_i^l, 1 \leq l \leq k$ responds with an ACK carrying h_l', i.e., $h_l'||i||$ACK. Note that n_i^l only verifies the tokens in the data packet without putting any effort on processing C_j, since its token does not appear at the beginning of the packet (i.e., it is not the intended receiver) and C_j can be decrypted only by the indented receiver.

C. Disabling source authentication

Source authentication is much more costly than data authentication as it requires decryption operations while data authentication only involves equality checks. If there are many false source authentication requests, as a receiver n_i will waste significant resources on processing them. To deal with this problem, n_i may adaptively enable or disable its source authentication function in T_c according to predicted neighborhood change and prediction accuracy.

Specifically, n_i chooses the set $\hat{\mathcal{N}}_i^c$ of possible neighbors at the beginning of T_c by checking the link quality prediction results (see Sec. V-A): a node is a possible neighbor if the corresponding link quality is predicted to have a value beyond certain threshold (a.k.a. receiver sensitivity). At the end of T_c, n_i computes the real neighbor set $\mathcal{N}_i^c$ in T_c based on the received ACKs during the time slot. Then it decides whether to disable source authentication for T_{c+1}, based on $\mathcal{N}_i^c$, $\hat{\mathcal{N}}_i^c$ and $\hat{\mathcal{N}}_i^{c-1}$. If $\mathcal{N}_i^c = \hat{\mathcal{N}}_i^c$ and $\hat{\mathcal{N}}_i^c = \hat{\mathcal{N}}_i^{c-1}$, n_i is in source authentication disabled mode (SAD) (or source authentication enabled (SAE) mode otherwise) as shown in Fig. 6. This condition implies that the prediction is accurate and the neighbor set is not expected to change; thus it is not necessary to perform source authentication. If the link quality prediction model is not established yet, $\hat{\mathcal{N}}_i^c$ is not available. In this case, source authentication has to be enabled by default. Source authentication is also periodically opened in order to accommodate unexpected legitimate neighbors.

VI. Security Analysis

In this section, we analyze the security properties of the PSR framework. Specifically, following the security model discussed in Sec. III-B, our analysis focuses on the resilience of PSR against data injection attacks including exhaustive source authentication attacks, exhaustive data authentication attacks and data replay attacks.

A. Resilience to exhaustive source authentication attacks

Fig. 4 shows the source authentication process. The sender node n_i computes an identity based signature (u, v) on message m_i and sends $u||v||m_i$ as authentication request to a node n_j at the first step. By checking $v{=}H_2(m_i, r)$ where $r = \frac{e(u,P)}{e(Q_i, P_{pub})^v}$, n_j knows whether the request is made by n_i or not. Specifically, if $r = e(R, P)^x$, n_j will be able to calculate $u = xR + vsQ_i$. Since $vsQ_i = vk_i$ can be only generated by n_i using its private key k_i, n_j is able to confirm that the signature (u, v) on m_i is indeed generated by n_i. This confirmation guarantees that n_j detects false source authentication requests. This signature-based approach consumes relatively intensive computational resources (compared with hash-based data authentication).

However, n_j does not always respond to source authentication requests. It records the real neighbor sets in the past and uses a prediction model to estimate the future neighbor sets (one time slot ahead). Such information assists it in making a wise decision: to disable the source authentication function when it is not necessary, i.e., when neighbor set is not changing and current neighbors have already been authenticated. In this way, most false source authentication requests can be directly ignored. Such an attack can still consume some computational resources of a receiver node when the node periodically enables source authentication for accepting new neighboring nodes. But the attack capability is significantly reduced.

B. Resilience to exhaustive data authentication attacks

A receiver node accepts only data packets that contain valid tokens. Recall that the tokens are created in a reverse order of hash values and initially known only to the sender and receiver nodes. Therefore, attackers cannot obtain the tokens in advance of the transmissions. Any false data packet will be rejected directly by the receiver node if attached the token is invalid (either unrecognized or already used).

C. Resilience to data replay attacks

If a data transmission fails at the receiver node, a data replay attacker may use the intercepted tokens and inject forged information into the network. In this case, the receiver node has to consume additional computation power to detect these forged data packets by decryption operations. We show that such attack capability can be limited in terms of attacking period and numbers of valid tokens. By adopting the hash chain technique, if $H^{m-2x+2}(d)$ is received and the x-th data packet is checked by the receiver node, the data packet with tokens $H^{m-2y+2}(d)$ for $1 \leq y \leq x$ will not be accepted anymore. This is because the y-th data packet cannot arrive later than the x-th data packet (packets are transmitted sequentially along a single hop). The possible attacking period is therefore largely reduced (see Theorem 1 below). Theorem 2 further indicates that the attacker can only obtain a limited number of valid tokens and thus attack the network a limited number of times. We denote the receive probability of n_j on a single transaction by ρ.

Theorem 1: If n_i consumes k tokens from a token set for data authentication in every time slot, then a data replay attacker $\mathcal{A}$ has an average attacking period $P_A = \frac{(1-\rho)\cdot\lambda}{\rho\cdot k}$ available for each eavesdropped token h.

Proof: If $\mathcal{A}$ eavesdrops a token h from n_i's transaction in T_c, the token may be already received by n_j. Thus $\mathcal{A}$ can replay token h to exhaust n_j's computational resources with probability $1-\rho$ during time period λ/k. In addition, if the next transaction of n_i fails, $\mathcal{A}$ can use h to attack for an additional time period λ/k, totally $2\lambda/k$. Thus, we are able to obtain the average attacking period for token h as follows: $P_A = \Sigma_{x=1}^{+\infty}\rho(1-\rho)^x \cdot \frac{x\lambda}{k} = \frac{(1-\rho)\cdot\lambda}{\rho\cdot k}$. ■

Theorem 2: If n_i consumes k tokens from a token set for data authentication in every time slot, then a data replay attacker $\mathcal{A}$ can obtain $2k-1$ valid tokens at most.

Proof: If no less than $2k$ valid tokens are obtained by $\mathcal{A}$, the data authentications by n_i must fail in two or three successive time slots. However, this is impossible as we show below. If data authentications fail in two successive time slots, n_i will not receive any ACKs from n_j in the first time slot and stop using the rest of the tokens. In this case, $\mathcal{A}$ can obtain at most k tokens. If data authentication fail in three successive time slots, n_i will not receive any ACKs from n_j in the second time slot and stop using tokens in the third time slot. In this case, $\mathcal{A}$ can obtain at most $2k-1$ tokens. Thus, $\mathcal{A}$ obtains $2k-1$ valid tokens at most. ■

VII. Performance Evaluation

In this section, we evaluate PSR through an extensive set of simulations. According to Sec. II, there are only a few multi-hop routing protocols designed for WBANs, none of which use link quality as routing metric. We choose to compare PSR (the version described in Sec. V) with a static tree-based routing protocol [10], referred to as Backbone, where sensors route data packets toward the data sink along a shortest path tree constructed using backbone links (see Sec. III-A). The Backbone protocol is reliable compared with other existing protocols, because the tree is a fixed structure with relatively stable links in the presence of postural mobility. It can thus be a good benchmark algorithm. As we will see, PSR outperforms Backbone in reliability and has desired security performance.

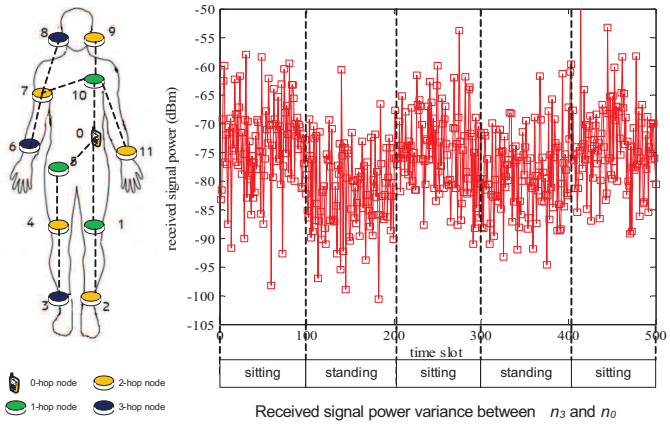


Fig. 7. Link quality varying with body movements

A. Simulation setup

We consider a WBAN deployed on the body of a person with height 1.7m. The network is composed of 12 nodes. As shown in Fig. 7, the data sink n_0 is placed on the waist; the others are placed on knees, ankles, shoulders, wrists and head. A shortest path tree rooted at n_0 is built using backbone links (see Sec. III-A) and shown by dotted lines. Similar WBAN settings can be found in [3], [10]. A well-defined and simplified channel model given by IEEE 802.15 task group 6 [18] is adopted in our simulation. The path loss between any two sensors deployed above body surface is given by:

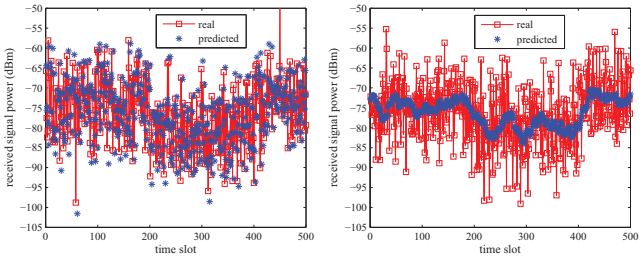
$$PL(d)[dB] = a \times log_{10}(d) + b + N$$

where a and b are coefficients of linear fitting, d is the direct distance between nodes n_i and n_j, N is a random variable of zero-mean normal distribution with standard deviation σ_N. We choose one of the suggested values by IEEE 802.15 task group 6 [18] under the frequency band 2.4GHz outdoors ($a = 29.3, b = -16.8, \sigma_N = 6.89$). Given the direct distance between n_i and n_j, the path loss can be calculated. Furthermore, we consider a noise model where the received signal power is given by:

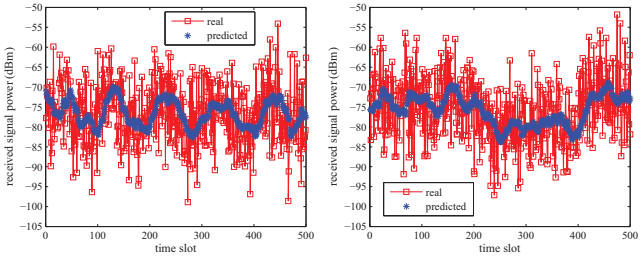
$$P_r(d)[dBm] = P_s - PL(d) - N_0$$

where P_s represents the transmission power, P_r the received signal power, and N_0 the noise power.

On one hand, the transmission power of body sensors must be kept less than an upper bound (13.98 dBm [18]) in order not to produce any harm to tissues. On the other hand, it must be strong enough to ensure the successful transmission, i.e., to maintain P_r at a certain level so that the receiver is able to filter data from noise. Under these circumstances, we define the minimum requirement of successful delivery with a power margin, and consider that a packet can be decoded correctly if

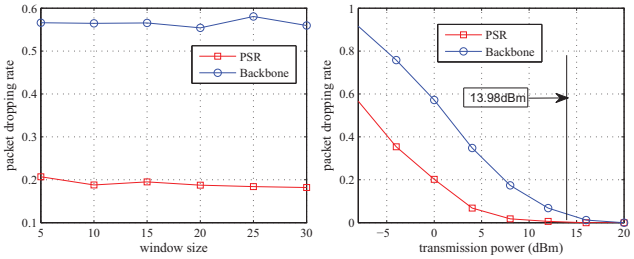


(a) Window size = 1 (b) Window size = 20

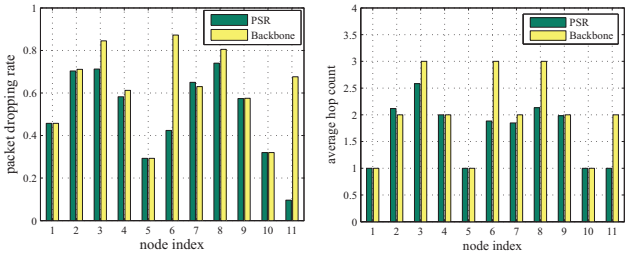


(c) Posture period = 50 (d) Posture period = 200

Fig. 8. Prediction accuracy



(a) Dropping rate per window size (b) Dropping rate per TRX power



(c) Dropping rate per node (d) Hop count per node

Fig. 9. Reliability performance

and only if the ratio of received signal power to noise power is larger than the power margin. In our simulation, the power margin is 10 dB and the receiver sensitivity -90 dBm [19].

We repeatedly alternate the body posture between sitting and standing, each of which lasts a fixed period of time (i.e., 50 or 200 simulated time slots). The postural mobility has direct impact on link quality. For example, from Fig. 7 we can see that the quality of the link from ankle-mounted sensor n_3 to data sink n_0 (received signal power at n_0) in sitting status is higher than that in standing status.

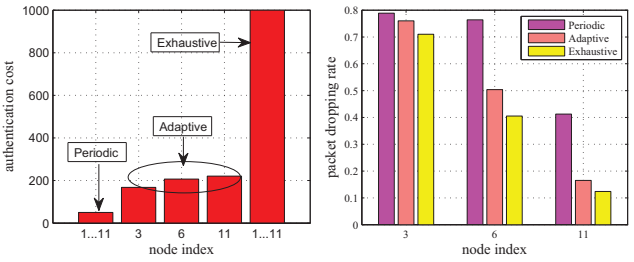
B. Simulation results

1) Prediction accuracy: We set the AR model order $p = 20$, posture period $= 200$ time slots, and employ a *sliding-window technique* for smoothing the noisy link quality measurements (i.e., received signal power). Specifically, we slide a window of certain size w (in time slot) along the time series, compute the average of the measurements within the window, and input the results into the AR model for prediction. Figures 8(a) and 8(b) show the predicted values and the true values between nodes n_3 and n_0, respectively with $w = 1$ and $w = 20$. We observe that when $w = 1$ the predicted link quality varies significantly along with the real values. The big variation is due to random channel noise. It hides the regularity of link quality brought by periodic postural mobility and renders the prediction results useless. In the case of $w = 20$, the regularity can be easily observed. Thus, we choose $w = 20$ in the rest of our simulation. Figures 8(c) and 8(d) show the influence of posture period on link quality prediction with $p = 20$ and $w = 20$. It can be seen that the trends of predicted values well matches that of the real values for the link from n_3 to n_0 whether posture period is set to 50 time slots or 200 time slots. The above results indicate that link quality can be predicted, and the prediction can be exploited to enable better link selection to improve routing reliability.

2) Reliability performance: Figure 9(a) shows that n_3 is able to find a better link by PSR than by Backbone. We observe that the packet dropping rate for the single hop from n_3 toward n_0 is reduced from 0.6 to 0.2. This per-hop reliability gain helps nodes improve end-to-end routing reliability. We also find out that the gain slightly changes over different window sizes. The reason is that the channel condition is extremely unstable, and the random channel noise and vibrating path loss diminish the difference of the results. From Fig. 9(b), it is observed that as transmission power increases, per hope packet dropping rate in both PSR and Backbone decreases, and PSR slowly loses its advantage over Backbone. This is because high transmission power increases link quality in general and diminishes the reliability difference due to algorithm design.

Figures 9(c) and 9(d) show end-to-end packet dropping rate and average hop count between different sensors and the data sink n_0. It can be observed that PSR outperforms Backbone in both aspects. The reason is that nodes when adopting PSR are able to find a better relay path by referring to link quality prediction results. If two nodes become each other's neighbor due to the body movement, the node with smaller hop count may be selected as a relay for the node with larger hop count (subject to link quality check) in PSR. Such opportunistic routing enables nodes to save more energy by reducing the number of relaying. For instance, according to Fig. 7, n_3 may directly transmit a data packet to n_5 for sitting status, rather than going through n_4, and the hop count to n_0 is reduced to 2 from 3 (in Backbone).

3) Security performance: Data authentication is realized by simple equality check. Hence, we focus on source authentication cost. We examine three source authentication policies in the context of PSR: exhaustive authentication, periodic authentication, and adaptive authentication. The exhaustive authentication policy requires each node to check every source authentication request in any time slot; periodic authentication



(a) Authentication cost (b) Packet dropping rate

Fig. 10. Security performance

policy requires that each node periodically checks source authentication requests at regular intervals (set to 20 time slots in our simulation); adaptive authentication policy inherits the periodic authentication policy, and it additionally requires each node to adaptively disable or enable source authentication (see Sec. V-C). We define authentication cost as the number of false source authentication requests that a node responds to. In our simulation, an attacker sends every node 1000 false source authentication requests, one per time slot. Figure 10 shows authentication cost and end-to-end packet dropping rate of three nodes n_3, n_6, n_{11} during 1000 time slots with the three authentication policies being applied.

Among the three policies, we observe that the exhaustive one achieves the lowest packet dropping rate. This is because nodes do not miss any neighbor and are always able to find the best link (i.e., with highest received signal power at the other side) as next hop. But this policy has the highest authentication cost due to its exhaustive nature. The periodic policy leads to the opposite performance: highest packet dropping rate and lowest authentication cost. It is because nodes are often unable to discover and use quality links as source authentication is blindly closed at fixed intervals. The performance of adaptive policy as expected is in between. It achieves low packet dropping rate (comparable to the exhaustive policy's) at small authentication cost (comparable to the periodic policy's) due to the intelligent source authentication enabling/disabling. In particular, the resultant authentication cost is less than 300, meaning that over 70% false requests are directly filtered.

VIII. Conclusion

In this paper, we proposed a prediction-based secure and reliable routing framework (PSR) for WBANs. This framework requires each sensor node to locally maintain a prediction model and obtain the neighborhood conditions in the immediate future. With the prediction results, the nodes can choose the incidental links of the best quality for packet relay to improve routing reliability and adaptively enable/disable source authentication function to resist data injection attacks. Through both analysis and simulation, we demonstrated that PSR indeed enables secure and reliable routing. Currently, PSR uses an exhaustive ACK technique to measure link quality: a receiver node acknowledges every authenticate data packet. Although an ACK packet is much smaller than a data packet in size and its transmission cost is considered negligible, there may possibly be a large number of ACKs transmitted and consumes a lot of network resources as a whole. To reduce ACK transmission overhead, we may instead use a group ACK scheme. The idea is to let a receiver transmit a single ACK for each distinct group of successive data packets from a sender, and each ACK contains the number of data packets it has successfully received in the current time slot and the aggregate received signal power of these packets. By reading ACKs from neighboring nodes, the sender is able to compute the average quality of each incidental link.

References

[1] Z. Ren, G. Zhou, A. Pyles, M. Keally, W. Mao, and H. Wang, "Bodyt2: Throughput and time delay performance assurance for heterogeneous bsns," in *Proc. IEEE INFOCOM*, 2011, pp. 2750–2758.

[2] A. Natarajan, B. de Silva, K.-K. Yap, and M. Motani, "Link layer behavior of body area networks at 2.4 ghz," in *MOBICOM*, 2009, pp. 241–252.

[3] S. Ullah, H. Higgins, B. Braem, B. Latre, C. Blondia, I. Moerman, S. Saleem, Z. Rahman, and K. Kwak, "A comprehensive survey of wireless body area networks," *Journal of Medical Systems*, pp. 1–30, 10.1007/s10916-010-9571-3.

[4] B. Latre, B. Braem, I. Moerman, C. Blondia, and P. Demeester, "A survey on wireless body area networks," *Journal of Wireless Networks*, vol. 17, pp. 1–18, 2011.

[5] A. Natarajan, M. Motani, B. de Silva, K.-K. Yap, and K. C. Chua, "Investigating network architectures for body sensor networks," in *HealthNet*, 2007, pp. 19–24.

[6] X. Lin, R. Lu, X. Shen, Y. Nemoto, and N. Kato, "Sage: A strong privacy-preserving scheme against global eavesdropping for ehealth systems," *IEEE Journal on Selected Areas in Communication*, vol. 27, no. 4, pp. 365–378, 2009.

[7] X. Liang, L. Chen, R. Lu, X. Lin, and X. Shen, "Pec: A privacy-preserving emergency call scheme for mobile healthcare social networks," *IEEE/KICS J. Commu. and Networks (JCN)*, vol. 13, no. 2, pp. 102–112, 2011.

[8] R. Lu, X. Lin, H. Luan, X. Liang, and X. Shen, "Pseudonym changing at social spots: An effective strategy for location privacy in vanets," *IEEE Transactions on Vehicular Technology*, 2011, preprint.

[9] X. Li, N. Mitton, and D. Simplot-Ryl, "Mobility prediction based neighborhood discovery in mobile ad hoc networks," in *Networking (1)*, 2011, pp. 241–253.

[10] B. Latre, B. Braem, I. Moerman, C. Blondia, E. Reusens, W. Joseph, and P. Demeester, "A low-delay protocol for multihop wireless body area networks," in *Proc. of 4th International Conference on Mobile and Ubiquitous Systems: Networking and Services*, 2007, pp. 1–8.

[11] M. Quwaider, J. Rao, and S. Biswas, "Transmission power assignment with postural position inference for on-body wireless communication links," *ACM Trans. Embedded Comput. Syst.*, vol. 10, no. 1, 2010.

[12] M. Quwaider and S. Biswas, "Dtn routing in body sensor networks with dynamic postural partitioning," *Ad Hoc Networks*, vol. 8, no. 8, pp. 824–841, 2010.

[13] M. Quwaider, M. Taghizadeh, and S. Biswas, "Modeling on-body dtn packet routing delay in the presence of postural disconnections," *EURASIP Journal on Wireless Communications and Networking*, vol. 2011, no. 280324, pp. 1–19, 2010.

[14] G. Box, G. M. Jenkins, and G. C. Reinsel, *Time Series Analysis: Forecasting and Control*, 4th ed. Wiley, 2008.

[15] D. Boneh and M. K. Franklin, "Identity-based encryption from the weil pairing," in *CRYPTO*, 2001, pp. 213–229.

[16] M. Li, S. Yu, W. Lou, and K. Ren, "Group device pairing based secure sensor association and key management for body area networks," in *Proc. IEEE INFOCOM*, 2010, pp. 2651–2659.

[17] F. Hess, "Efficient identity based signature schemes based on pairings," in *Selected Areas in Cryptography*, 2002, pp. 310–324.

[18] T. Aoyagi, J. Takada, K. Takizawa, N. Katayama, T. Kobayashi, K. Y. Yazdandoost, H. Li, and R. Kohno, "Channel model for wearable and implantable wbans," *IEEE 802.15-08-0416-04-0006*, 2008.

[19] S. J. Marinkovic, E. M. Popovici, C. Spagnol, S. Faul, and W. P. Marnane, "Energy-efficient low duty cycle mac protocol for wireless body area networks," *IEEE Transactions on Information Technology in Biomedicine*, vol. 13, no. 6, pp. 915–925, 2009.