Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2008, Article ID 529879, 16 pages
doi:10.1155/2008/529879

Research Article

Biometric Methods for Secure Communications in Body Sensor Networks: Resource-Efficient Key Management and Signal-Level Data Scrambling

Francis Minhthang Bui and Dimitrios Hatzinakos

*The Edward S. Rogers Sr. Department of Electrical and Computer Engineering, University of Toronto,
10 King's College Road, Toronto, Ontario, Canada M5S 3G4*

Correspondence should be addressed to Dimitrios Hatzinakos, dimitris@comm.utoronto.ca

Received 1 June 2007; Revised 28 September 2007; Accepted 21 December 2007

Recommended by Juwei Lu

As electronic communications become more prevalent, mobile and universal, the threats of data compromises also accordingly loom larger. In the context of a body sensor network (BSN), which permits pervasive monitoring of potentially sensitive medical data, security and privacy concerns are particularly important. It is a challenge to implement traditional security infrastructures in these types of lightweight networks since they are by design limited in both computational and communication resources. A key enabling technology for secure communications in BSN's has emerged to be biometrics. In this work, we present two complementary approaches which exploit physiological signals to address security issues: (1) a resource-efficient key management system for generating and distributing cryptographic keys to constituent sensors in a BSN; (2) a novel data scrambling method, based on interpolation and random sampling, that is envisioned as a potential alternative to conventional symmetric encryption algorithms for certain types of data. The former targets the resource constraints in BSN's, while the latter addresses the fuzzy variability of biometric signals, which has largely precluded the direct application of conventional encryption. Using electrocardiogram (ECG) signals as biometrics, the resulting computer simulations demonstrate the feasibility and efficacy of these methods for delivering secure communications in BSN's.

1. INTRODUCTION

Security is a prime concern of the modern society. From a local house-hold setting to a more global scope, ensuring a safe and secure environment is a critical goal in today's increasingly interconnected world. However, there are still outstanding obstacles that have prevented the realization of this objective in practical scenarios, despite many technological advances. Recently, body sensor networks (BSNs) have shown the potential to deliver promising security applications [1–3]. Representing a fast-growing convergence of technologies in medical instrumentation, wireless communications, and network security, these types of networks are composed of small sensors placed on various body locations. Among the numerous advantages, this BSN approach permits round-the-clock measurement and recording of various medical data, which are beneficial compared to less frequent visits to hospitals for checkup. Not only there is convenience for an individual, but also more data can be collected to subsequently aid reliable diagnoses. In other words, a BSN helps bridge the spatio-temporal limitations in pervasive medical monitoring [4, 5].

Aside from medical applications, analogous scenarios may be considered with a general network of wearable devices, including cell phones, headsets, handheld computers, and other multimedia devices. However, the incentive and urgency for inter-networking such multimedia devices may be less obvious and imminent (more on the convenience side), compared to those in medical scenarios (more on the necessity side).

The objectives of this work are to: (1) examine the various nascent BSN structures and associated challenges, (2) establish a flexible high-level model, encompassing these assumptions and characteristics, that is conducive to

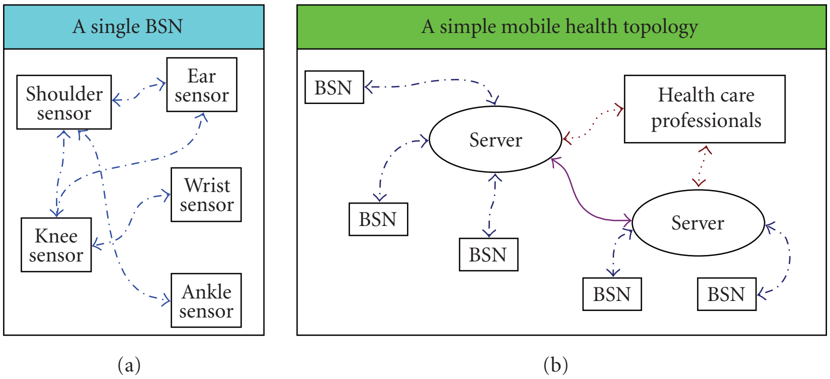


Figure 1: Model of a mobile health network, consisting of various body sensor networks.

future research from a signal-processing perspective, (3) propose signal processing methods and protocols, in the context of a high-level model, that improve upon existing schemes for providing security in BSNs. More specifically, the last objective (3) is two-fold: (a) we construct a secure key distribution system that is shown to be more resource-efficient than the current scheme based on fuzzy commitment; (b) we propose and study a data scrambling method that has the potential to supplant conventional encryption, in securing certain types of data using biometrics [3].

The remainder of this paper is organized as follows. In Section 2, we provide a survey of the existing research on BSNs, highlighting the salient features and assumptions. This is followed by a high-level summary of our methodologies and objectives of research on BSNs in Section 3. Detailed descriptions are next given for a resource-efficient key management system, including key generation and distribution, in Section 4. Then, we present the INTRAS framework for data scrambling in Section 5. And, in order to evaluate the system performance, simulation results are summarized in Section 6. Lastly, concluding remarks for future directions are given in Section 7.

2. LITERATURE SURVEY

2.1. BSN structure and assumptions

Even though BSN is a comparatively new technology, it has garnered tremendous interest and momentum from the research community. This phenomenon is easy to understand when one remarks that a BSN is essentially a sensor network, or to a broader extent an ad hoc network [6, 7], with characteristics peculiar to mobile health applications.

So far, the current trend in BSN research has focused mainly on medical settings [4]. As an ad hoc network, a typical BSN consists of small sensor devices, usually destined to report medical data at varying intervals of time. Figure 1(a) shows a typical high-level BSN organization. Each BSN consists of a number of sensors, dedicated to monitoring medical data of the wearer. As noted in [1, 4], for implanted sensors, wireless communication is by far the preferred solution since wired networking would necessitate laying wires within the human body; and for wearable devices, wireless networking is also desirable due to user convenience.

There are many possible variations on the BSN structure, especially with respect to the network topologies formed from various BSNs. A very simple topology is given in Figure 1(b), depicting a mobile-health network and organizing several BSNs under one server. As explored in [5], a more sophisticated organization can involve elected leader nodes within a BSN, which allow for more specialized communication requirements. For instance, certain nodes have higher computational capabilities than others in order to perform more sophisticated tasks. This hierarchical organization is needed for a scalable system, especially with a fixed amount of resources.

2.2. Resource constraints in BSNs

As in a typical ad hoc network, there is a large range of variations in resource constraints. From the proposed prototypes and test beds found in the existing literature, the computational and bandwidth limitations in BSNs are on par with those found in the so-called microsensor networks [6, 7]. While relatively powerful sensors can be found in a BSN, the smaller devices are destined to transmit infrequent summary data, for example, temperature or pressure reported every 30 minutes, which translates to transmissions of small bursts of data on the order of only several hundred, or possibly thousand, bits.

The computational and storage capabilities of these networks have been prototyped using UC Berkeley MICA2 motes [5], each of which provides an 8-MHz ATMega-128 L microcontroller with 128 KB of programmable flash, and 4-Kbytes of RAM. In fact, these motes may exceed the resources found in smaller BSN sensors. As such, to be safe, a proposed design should not overstep the capabilities offered by these prototype devices.

With energy at a premium, a study of the source of energy consumption in a BSN has been performed by evaluating the amount of energy dispensed per bit of information, similar to the analysis in [6]. The conclusion is that [1, 2, 4, 8], while computational and communication resources are both constrained in a BSN, the most expensive one is the communication operation. The computational costs are typically smaller so much that they are almost negligible compared to the cost of communication. Moreover, recall that the payload data for a scheduled transmission session in a BSN are on the order of a few hundred bits, which means that even a typical 128-bit key employed for encryption would be substantial by comparison. As such, only information bits that are truly necessary should be sent over the channel. This guideline has profound repercussions for the security protocols to be adopted in a BSN.

2.3. Security and biometrics in BSNs

While the communication rate specifications in BSN are typically low, the security requirements are stringent, especially when sensitive medical data are exchanged. It should not be possible for sensors in other BSNs to gain access to data privy to a particular BSN. These requirements are difficult to guarantee due to the wireless broadcasting nature of a BSN, making the system susceptible to eavesdroppers and intruders.

In the BSN settings evaluated by [1, 4, 5, 8], the prototypes show that traditional security paradigms designed for conventional wireless networks [9] are in general not suitable. Indeed, while many popular key distribution schemes are asymmetric or public-key- based systems, these operations are very costly in the context of a BSN. For instance, it was reported that to establish a 128-bit key using a Diffie-Hellman system would require 15.9-mJ, while symmetric encryption of the same bit length would consume merely 0.00115-mJ [1]. Therefore, while key distribution is certainly important for security, the process will require significant modifications in a BSN.

By incorporating the body itself and the various physiological signal pathways as secure channels for efficiently distributing the derived biometrics, security can be feasibly implemented for BSN [1, 2]. For instance, a key distribution scheme based on fuzzy commitment is appropriate [1, 10]. A biometric is utilized for committing, or securely binding, a cryptographic key for secure transmission over an insecure channel. More detailed descriptions of this scheme will be given in Section 2.5. Essentially, for this construction, the biometric merely serves as a witness. The actual cryptographic key, for symmetric encryption [9], is externally generated, (i.e., independent from the physiological signals). This is the conventional view of biometric encryption [11]. The reasons are two-fold: (1) good cryptographic keys need to be random, and methods for realizing an external random source are quite reliable [9]; moreover, (2) the degree of variations in biometrics signals is such that two keys derived from the same physiological traits typically do not match exactly. And, as such, biometrically generated keys would not be usable in conventional cryptographic schemes, which by design do not tolerate even a single-bit error [9, 11].

2.4. The ECG as a biometric

While many physiological features can be utilized as biometrics, the ECG has been found to specifically exhibit desirable characteristics for BSN applications. First, it should be noted that for the methods to be examined, the full-fledged ECG signals are not required. Rather, it is sufficient to record only the sequence of *R*-*R* wave intervals, referred to as the interpulse interval (IPI) sequence [4]. As a result, the methods are also valid for other cardiovascular signals, including phonocardiogram (PCG), and photoplethysmogram (PPG). What is more, as reported in [1, 4, 5], there are existing sensor devices for medical applications, manufactured with reasonable costs, that can record these IPI sequences effectively. That is, the system requirements for extracting the IPI sequences can be essentially considered negligible.

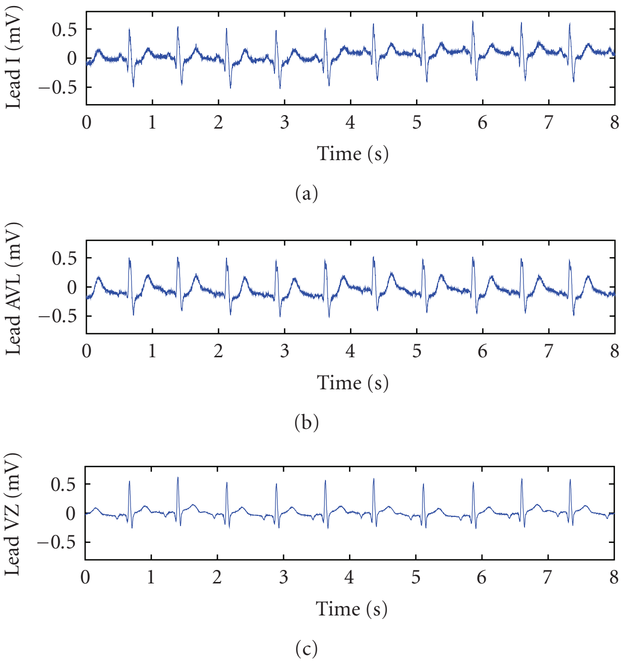
2.4.1. Time-variance and key randomness

At this point, it behooves us to distinguish between time-invariant and time-variant biometrics. In most conventional systems, biometrics are understood and required to be time-invariant, for example, fingerprints or irises, which do not depend on the time measured. This is so that, based on the recorded biometric, an authority can uniquely identify or authenticate an individual in, respectively, a one-to-many and one-to-one scenario [11]. By contrast, ECG-based biometrics are time-variant, which is a reason why they have not found much prominence in traditional biometric applications. Fortunately, for a BSN setting, it is precisely the time-varying nature of the ECG that makes it a prime candidate for good security. As already mentioned, good cryptographic keys need a high degree of randomness, and keys derived from random time-varying signals have higher security, since an intruder cannot reliably predict the true key. This is especially the case with ECG, since it is time-varying, changing with various physiological activities [12]. More precisely, as previously reported in [13], heart rate variability is characterized by a (bounded) random process.

2.4.2. Timing synchronization and key recoverability

Of course, key randomness is only part of the security problem. An ECG biometric would not be of great value unless the authorized party can successfully recover the intended cryptographic key from it. In other words, the second requirement is that the ECG-generated key should be reproducible with high fidelity at various sensor nodes in the same BSN.

To expose the feasibility of accurate biometric reproducibility at various sensors, let us consider typical ECG signals from the PhysioBank [14], as shown in Figure 2. For the present paper, it suffices to focus on the so-called QRS-complexes, particularly the *R*-waves, which represent usually the highest peaks in an ECG signal [12, 15]. The sequence of *R*-*R* intervals is termed the interpulse interval (IPI) sequence [4] and essentially represents the time intervals between successive pulses. In this case, three different ECG signals are measured simultaneously from three different electrode or lead placements (I, AVL, VZ [12, 14]). What is noteworthy is that, while the shapes of specific QRS complexes are different for each signal, the sequences of IPI for the three signals, with proper timing synchronization, are remarkably identical. Physiologically, this is because the three



(a)

(b)

(c)

Figure 2: ECG signals simultaneously recorded from three different leads. (Taken from the PhysioBank [14].)

leads measure three representations of the same cardiovascular phenomenon, which originates from the same heart [12]. In particular, the IPI sequences capture the heart rate variations, which should be the same regardless of the measurement site.

Therefore, in order to recover identical IPI sequences at various sensors, accurate timing synchronization is a key requirement. While the mechanism of timing synchronization is not directly addressed in this paper, one possible solution is to treat this issue from a network broadcast level [1, 4, 5]. Briefly stated, in order that all sensors will ultimately produce the same IPI, they should all listen to an external broadcast command that serves to reinitialize, at some scheduled time instant, the ECG recording and IPI extraction process. This scheduling coordination also has a dual function of implementing key refreshing [4, 5, 9]. Since a fresh key is established in the BSN with each broadcast command for re-initialization, the system can enforce key renewal as frequently as needed to satisfy the security demand of the envisioned application: more refreshing ensures higher security, at the cost of increased system complexity.

2.5. *Single-point fuzzy key management with ECG*

So far, various strategies in the literature have exploited ECG biometrics to bind an externally generated cryptographic key and distribute it to other sensors via fuzzy commitment [1, 2, 5, 16]. The cryptographic key intended for the entire BSN is generated at a single point, and then distributed to the remaining sensors. In addition, the key is generated independently from the biometric signals, which merely act as witnesses. For these reasons, we will henceforth refer to this scheme as single-point fuzzy commitment.

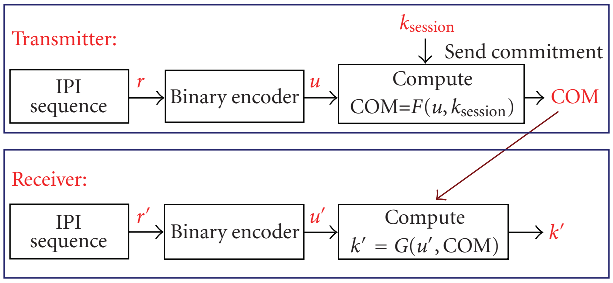


Figure 3: Single-point fuzzy key management.

Figure 3 summarizes the general configuration of the single-point key management. The data structures of the signals at various stages are as follows:

(i) r: the sequence of IPI derived from the heart, represented by a sequence of numbers, the range and resolution of which are dependent on the sensor devices used.

(ii) u: obtained by uniform quantization of r, followed by conversion to binary, using a PCM code [17].

(iii) r', u': the corresponding quantities to the nonprime versions, which are derived from the receiver side.

(iv) k_{session}: an externally generated random key to be used for symmetric encryption in the BSN. It needs to be an error correction code, as explained in the sequel.

(v) k': the recovered key, with the same specifications as k_{session}.

(vi) COM: the commitment signal, generated using a commitment function F defined as

$$\text{COM} = F(u, k_{\text{session}}) = (\underbrace{h(k_{\text{session}})}_{a}, \underbrace{u \oplus k_{\text{session}}}_{d}), \tag{1}$$

where $h(\cdot)$ is a one-way hash function [9], and $\oplus$ is the XOR operator.

Therefore, the commitment signal to be transmitted is a concatenation of the hashed value of the key and an XOR-bound version of the key. With the requirement of k_{session} being a codeword of an error correcting code, with decoder function $f(\cdot)$, the receiver produces a recovered key k', using a fuzzy knowledge of u', as

$$k' = G(u', \text{COM}) = G(u', a, d) = f(u' \oplus d). \tag{2}$$

If $f(\cdot)$ is a t-bit error-correcting decoder (i.e., can correct errors with a Hamming distance of up to t), then

$$f(u' \oplus d) = f(k_{\text{session}} + (u' \oplus u)) = f(k_{\text{session}} + e). \tag{3}$$

Hence, as long as r and r' are sufficiently similar, so that $|e| \leq t$, the key distribution should be successful. This can be verified using the included check-code $a = h(k_{\text{session}})$: checking whether $h(k') = a = h(k_{\text{session}})$. However, if the check-code is also corrupted, a false verification failure may occur.

3. OUR CONTRIBUTIONS

The existing research in BSN using ECG biometric can be classified into two major categories: network topology (via clustering formation), and key distribution (via fuzzy commitment). We will not address the first topic in this paper (the interested reader can refer to [5] and the references therein). However, in the previous section, we have reviewed in some detail the second challenge of key distribution, since one part of our contribution will focus on extending this approach. Furthermore, we also see the need for a third area of research: the data encryption stage, which is of course the raison d'être for secure key distribution in the first place.

In the BSN context, the use of conventional encryption is hampered by the key variability inherent in biometric systems. Biometric signals are typically noisy, which inevitably lead to variations, however minute, in the recovered cryptographic keys. The problem is that, however minute the variation, a single-bit error is sufficient to engender a decryption debacle with conventional cryptography. It is possible to employ extremely powerful error-correcting coders and generous request-resend protocols to counteract these difficulties. Of course, the amount of accrued energy consumption and system complexity would then defeat the promise of efficient designs using biometrics.

A more practical alternative would be to employ an encryption scheme that is inherently designed to rectify the inevitable key variations. One such alternative is the fuzzy vault method [11], the security of which is based on the intractable polynomial root finding problem. However, this choice may not be practical, since the scheme requires high computational demands, which can defy even conventional communication devices, let alone the more resource-scarce BSN sensors.

With the above challenges in mind, we propose two flexible methodologies for improving resource consumption in BSN. First, we present a key management scheme that consumes less communication resources compared to the existing single-point fuzzy key method, by trading off processing delay and computational complexity for spectral efficiency, which is the effective data rate transmitted per available bandwidth [17]. This represents more efficient use of bandwidth and power resources.

Second, to accommodate the key mismatch problem of conventional encryption, we propose a data scrambling framework known as INTRAS, being based on interpolation and random sampling. This framework is attractive not only for its convenient and low-complexity implementation, but also for its more graceful degradations in case of minor key variations. These characteristics accommodate the limited processing capabilities of the BSN devices and reinforce INTRAS as a viable alternative candidate for ensuring security in BSN based on physiological signals.

In order to be feasibly implementable in a BSN context, a design should not impose heavy resource demands. To ensure this is the case, we will adhere to the precedents set by the existing research. Only methods and modules which have been deemed appropriate for the existing prototypes would be utilized. In this sense, our contributions are not in the instrumentation or acquisition stages, rather we propose modifications in the signal processing arena, with new and improved methodologies and protocols that are nonetheless compatible with the existing hardware infrastructure.

4. MULTIPOINT FUZZY KEY MANAGEMENT

As discussed above, only information bits that are truly essential should be transmitted in a BSN. But, by design, the minimum number of bits, required by the COM sequence, in single-point key management scheme is the length of the cryptographic key (no check-code transmitted). Motivated by this design limitation, we seek a more flexible and efficient alternative. The basic idea is to send only the check-code and not a modified version of the key itself over the channel. At each sensoring point in a BSN, the cryptographic key is regenerated from the commonly available biometrics. As such, this scheme is referred to as multipoint fuzzy key management.

With respect to key generation, the possibility of constructing k_{session} from the biometric signal r has been explored in [4, 16], with the conclusion that the ECG signals have enough entropy to generate good cryptographic keys. But note that this generation is only performed at a single point. In other words, the only change in Figure 3 is that k_{session} itself is now some mapped version of u.

However, because of the particular design of BSN, other sensor nodes also have access to similar versions of u. As explained above, the generated biometrics sequences from sensors within the same BSN are remarkably similar. For instance, it has been reported that for a 128-bit u sequence captured at a particular time instant, sensors within the same BSN have Hamming distances less than 22; by contrast, sensors outside the BSN typically result in Hamming distances of 80 or higher [18]. Then, loosely speaking, it should be possible to reliably extract an identical sequence of some length less than 106 bits from all sensors within a BSN.

It should be noted that these findings are obtained for a normal healthy ECG. Under certain conditions, the amount of reliable bits recovered may deviate significantly from the nominal value. But note that these cited values are for any independent time segments corresponding to 128 raw bits derived from the continually varying IPI sequence. In other words, even if the recoverability rate is less, it is possible to reliably obtain an arbitrary finite-length key, by simply extracting enough bits from a finite number of nonoverlapping 128-bit snapshots derived from the IPI sequences. This possibility is not available with a time-invariant biometric, for example, a fingerprint biometric, where the information content or entropy is more or less fixed.

In a multipoint scheme, a full XOR-ed version of the key no longer needs to be sent over the channel. Instead, only the check-code needs to be transmitted for verification. Furthermore, the amount of check-code to be sent can be varied for bandwidth efficiency, depending on the quality of verification desired.

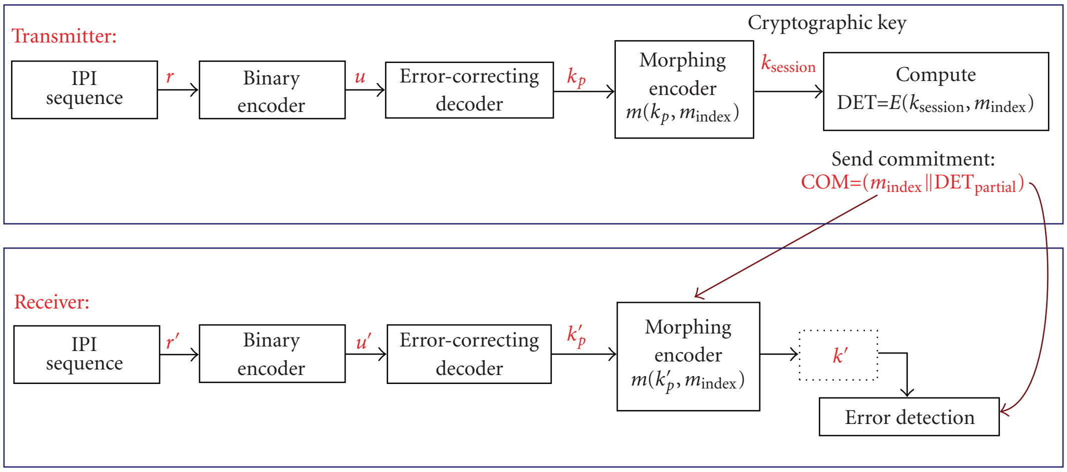


FIGURE 4: Multipoint fuzzy key management scheme.

4.1. Multipoint system modules

The basic hardware units supporting the following modules are already present in a single-point system. Thus, the innovation is in the design of the roles that these blocks take at various points in the transmission protocol. A high-level summary of the proposed multipoint scheme is depicted in Figure 4.

4.1.1. Binary encoder

Similar to a single-point key management, the first step involves signal conditioning by binary encoding (i.e., quantization and symbol mapping).

4.1.2. Error-correcting decoder

The next step seeks to remove just enough (dissimilar) features from a signal so that, for two sufficiently similar input signals, a common identical signal is produced. This goal is identical to that of an error-correcting decoder, if we treat the signals u and u' as if they were two corrupted codewords, derived from a common clean codeword, of some hypothetical error-correcting code.

For an error-correcting encoder with n-bit codewords, any n-bit binary sequence can be considered as a codeword plus some channel distortions. This concept is made more explicit in Figure 5. Here, we have conceptually modeled the ECG signal-generation process to include a hypothetical channel encoder and a virtual distorting channel. In an analogous formulation, many relevant similarities are found in the concept of the so-called superchannel [19]. A superchannel is used to model the equivalent effect of all distortions, not just the fading channel typical of the physical layer, but also other nonlinearities in other communication layers, with the assumption of cross-layer interactions.

An analogous study of the various types of codes and suitable channel models, in the BSN context, would be beyond the scope of this paper. Instead, the goal of the present paper is to establish the general framework for this approach.

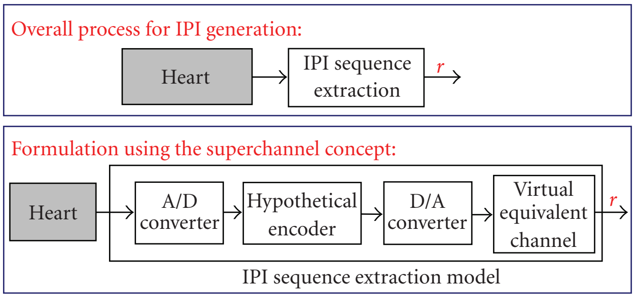

Overall process for IPI generation:
Heart
IPI sequence extraction
r
Formulation using the superchannel concept:
Heart
A/D converter
Hypothetical encoder
D/A converter
Virtual equivalent channel
r
IPI sequence extraction model

FIGURE 5: Equivalent superchannel formulation of ECG generation process.

In addition, while the optimal coding scheme for a BSN may not be a conventional error-correcting code [17, 19], we will limit our attention to a conventional BCH code family, to evaluate the feasibility of this superchannel formulation.

In practical terms, for Figure 4, a conventional BCH error-correcting decoder is used to *encode* a raw binary sequence, treated as a corrupted codeword of a corresponding hypothetical BCH encoder. This means that the error-correcting decoder in Figure 4 is used to reverse this hypothetical encoding process, generating hopefully similar copies of the pre-key k_P at various sensors, even though the various u-sequences may be different. In essence, the key idea of this error-correction decoder module is to correct the errors caused by the physiological pathways. The equivalent communication channels consist of the nonidealities and distortions existing between the heart and the sensor nodes.

In the following, we analyze the practical consequences, in terms of the required error-correcting specification, of the above conceptual model. Let us assume that ideal access to the undistorted IPI sequence R_I originates directly from the heart. Then, each sensor receives a (possibly) distorted copy of R_I. For example, consider sensors $i = 1, 2, \ldots, N$ with copies:

$$r_1 = c_1(R_I), r_2 = c_2(R_I), \ldots, r_N = c_N(R_I), \tag{4}$$

where $c_i(\cdot)$ represents the distorting channel from the heart to each sensor i.

Next, approximating the binary-equivalent channels as additive-noise channels [17], we can write

$$u_1 = u_I + e_1, u_2 = u_I + e_2, \dots, u_N = u_I + e_N, \tag{5}$$

where u_I is the binary-encoded sequence of R_I, and e_i represents the equivalent binary channel noise between the heart and sensor i.

Furthermore, consider an error-correcting code C with parameters (n, k, t), where n is the bit-length of a codeword, k is the bit-length of a message symbol, and t is the number of correctable bit errors. Let the encoder and decoder functions of C be $e_C(\cdot)$ and $d_C(\cdot)$, respectively. Define the demapping operation as the composite function $f_C(\cdot) = e_C(d_C(\cdot))$. In other words, for a particular n-bit sequence x, the operation $\widehat{x} = f_C(x)$ should demap x to the closest n-bit codeword $\widehat{x}$.

Then, suppose the bit-length of u_I is n and apply the demapper to obtain: $\widehat{u_I} = f_C(u_I) = u_I + E$, where $|E| \leq t$ is the Hamming distance from u_I to the nearest codeword $\widehat{u_I}$. Similarly, after demapping the other sensor sequences:

$$\begin{aligned} \widehat{u_1} &= f_C(u_1) = f_C(u_I + e_1) = f_C(\widehat{u_I} - E + e_1), \\ &\vdots \\ \widehat{u_N} &= f_C(u_N) = f_C(u_I + e_N) = f_C(\widehat{u_I} - E + e_N). \end{aligned} \tag{6}$$

The preceding relations imply that correct decoding is possible if $|e_1 - E| \leq t, \dots, |e_N - E| \leq t$. Moreover, the correct demapped codeword sequence is $\widehat{u_I}$, which is due to the original ideal sequence u_I directly from the heart. If error-correction is successful at all nodes according to the above condition, then the same pre-key sequence, $k_P = d_C(u_I) = d_C(\widehat{u_I})$, will be available at all sensors.

The above assessment is actually pessimistic. Indeed, it is accurate for the case where the channels c_i's have not distorted the sensor signals too far away from the ideal sequence u_I. However, when *all* the sensor channels carry the signals further away from the ideal case, the same code sequence can still be obtained from all sensors. But in this case, the decoded sequence will no longer be $\widehat{u_I}$, as examined next.

Let the codeword closest to *all* sequences $u_1, u_2, \dots, u_N$ be u_C. The condition that all signals have moved far away from the ideal case is more precisely defined by requiring the Hamming distance between u_C and u_I to be strictly greater than t (sensor sequences no longer correctable to u_I by the error-correcting decoder). Let

$$u_1 = u_C + \epsilon_1, u_2 = u_C + \epsilon_2, \dots, u_N = u_C + \epsilon_N, \tag{7}$$

where ϵ_i represents the respective Hamming distance. Then, the same key sequence, namely $k_P = d_C(u_C)$, is recoverable at all sensors provided that $\epsilon_1 \leq t, \dots, \epsilon_N \leq t$. In other words, the signals may depart significantly from the ideal case but will still be suitable for key generation, provided that they are all close enough to some codeword u_C.

4.1.3. Morphing encoder and random set optimization

The relevant data structures for this module are:

(i) k_p, k'_p: pre-key sequences, with similar structures as the session keys in the single-point scheme.

(ii) $m(\cdot)$, m_{index}: respectively, the morphing function and a morphing index, which is a short input sequence, for example, 2 to 4 bits. Here, we use the cryptographic hash function SHA-1 [9] for the morphing function $m(\cdot)$.

(iii) k_{session}, k': morphed versions of the pre-key sequences to accommodate privacy issues. Since the output of the SHA-1 function is a 160-bit sequence, for an intended 128-bit key, one can either use the starting or the ending 128-bit segment.

From a cryptographic perspective, the generated pre-key k_P is already suitable for a symmetric encryption scheme; as such, this morphing block can be considered optional. However, one of the stated goals is to ensure user privacy and confidentiality. As noted in [11], for privacy reasons, any signals, including biometrics, generated from physiological data should not be retraceable to the original data. The reason is because the original data may reveal sensitive medical conditions of the user, which is the case for the ECG. Therefore, a morphing block serves to confidently remove obvious correlations between the generated key and the original medical data.

In addition, due to the introduction of a morphing block, there is an added advantage that ensues, especially for the INTRAS framework to be presented in Section 5. First, suppose that we can associate a security metric (SM) to a pair of input data x and its encrypted version x_d, which measures in some sense the dissimilarity as $\text{SM}(x, x_d)$. Then, we can optimize the level of security by picking an appropriate key sequence. Deferring the details of INTRAS to the next section, we examine this idea as follows. Let x be a sequence of data to be scrambled, using a key sequence d. The scrambled output is

$$x_d = \text{INTRAS}(x, d). \tag{8}$$

Then, for the sequence x, the best key d_{opt} should be

$$d_{\text{opt}} = \arg\max_{d} \text{SM}(x, x_d). \tag{9}$$

In other words, $d = d_{\text{opt}}$ is a data-dependent sequence that maximizes the dissimilarity between x and the scrambled version x_d. Of course, implementing this kind of "optimal" security may not be practical. First, solving for d_{opt} can be difficult, especially with nonlinear interpolators. In addition, since the optimal key is data-dependent, the transmitter would then need to securely exchange this key with the receiver, which defeats the whole purpose of key management.

A more suitable alternative is to consider the technique of random set optimization. Essentially, for difficult optimization problems, one can perform an (exhaustive) search over some limited random set from the feasible space. If the set is sufficiently random, then the constrained solution can be a good estimate of the optimal solution.

Combining the above two goals of data hiding and key optimization, a morphing block, denoted by $m(\cdot)$, can be suitably implemented using a keyed hash function [9]. With this selection, the first goal is trivially satisfied. Furthermore, a property of a hash function is that small changes in the input results in significant changes in the output (i.e., the avalanche effect [9]). In other words, it is possible to generate a pseudorandom set using simple indexing changes in a morphing function, starting from a pre-key k_p. Specifically, consider the generation of the key sequence d for INTRAS:

$$d = m([k_p, m_{\text{index}}]), \quad m_{\text{index}} \in \mathcal{M}, \tag{10}$$

with $\mathcal{M}$ being the available index set for the morphing index m_{index}. The cardinality of $\mathcal{M}$ should be small enough that m_{index} (e.g., a short sequence of 2 to 4 bits) can be sent as side information in COM. The input to the morphing function is the concatenation of k_p and the morphing index m_{index}. Due to the avalanche effect, even small changes due to the short morphing index would be sufficient to generate large variations in the output sequence d.

Then, corresponding to Figure 4, the appropriate k_{session} is the one generated from k_p using $m_{\text{index opt}}$, where

$$m_{\text{index opt}} = \underset{m_{\text{index}} \in \mathcal{M}}{\arg\max}\, \text{SM}(x, \text{INTRAS}(x, d)). \tag{11}$$

In the above equation, d is defined as in (10). This optimization can be exhaustively solved, since the cardinality of $\mathcal{M}$ is small. As shown in Figure 4, m_{index} can be transmitted as plain-text side-information as part of COM, that is, without encryption. This is plausible because, without knowing k_p, knowing m_{index} does not reveal information about k_{session}.

It should also be noted that only the transmitting node needs to perform the key optimization. Therefore, if computational resource needs to be conserved, this step can be simplified greatly (e.g., selecting a random index for transmission) without affecting the overall protocol.

The selection of an appropriate SM is an open research topic, which needs to take into account various operating issues, such as implementation requirements as well as the statistical nature of the data to be encrypted. For the present paper, we will use as an illustrative example the mean-squared error (MSE) criterion for the SM. In general, the MSE is not a good SM, since there exist deterministically invertible transforms that result in high MSE. However, the utility of the MSE, especially for multimedia data, is that it can provide a reasonable illustration of the amount of (gradual) distortions caused by typical lossy compression methods. An important argument to be made in Section 5 is that, in the presence of noise and key variations, the recovered data suffer a similar gradual degradation. Therefore, the use of the MSE to assess the difference between the original and recovered images is especially informative. In other words, there is a dual goal of investigating the robustness of the INTRAS inverse, or recovery process.

4.1.4. Transmission and error detection

(i) DET and $E(\cdot)$: the error-detection bits, and the function used to generate these bits, respectively. For simplicity, the same hash function SHA-1 is used for $E(\cdot)$.

(ii) COM: the commitment signal actually transmitted over the channel.

Note that COM is the concatenation of the morphing index and part of DET. Being the output of SHA-1, DET is a 160-bit sequence. However, since error detection—as opposed to correction in the single-point scheme—is performed, it is not necessary to use the entire sequence. Therefore, depending on the bandwidth constraint or the desired security performance, only some segment of the sequence is partially transmitted, for example, the first 32 or 64 bits as done in the simulation results. The length of this partial sequence determines the confidence of verification and can be adapted according to the envisioned application.

The receiver should already have all the information needed to regenerate the pre-key k_p. Possible key mismatches are detected based on the partial DET bits transmitted. If verification fails, a request for retransmission needs to be sent, for example, using an ARQ-type protocol.

4.2. Performance and efficiency

The previous sections show that the most significant advantage of a multipoint scheme, in a BSN context, involves the efficient allocation of the scarce communication spectrum. With respect to spectral efficiency, the number of COM bits required for the original single-point scheme is at least the length of the cryptographic key. By contrast, since the proposed system only requires the transmitted bits for error detection, the number can be made variable. Therefore, depending on the targeted amount of confidence, the number of transmitted bits can be accordingly allocated for spectral efficiency.

However, this resource conservation is achieved at the expense of other performance factors. First, as in the single-point key management scheme, the success of the proposed multipoint construction relies on the similarities of the physiological signals at the various sensors. Although the requirements in terms of the Hamming distance conditions are similar, there are some notable differences. For the single-point management, from (3), the tolerable bit difference is quantifiable completely in terms of the pair of binary features u and u'. By contrast, for the multipoint management, from (6), the tolerable bit difference is also dependent on the distance of the uncorrupted binary IPI sequence u_I from the closest codeword. In other words, the closer the IPI sequence is from a valid codeword, the less sensitive it is from variations in multiple biometric acquisitions.

This preceding observation provides possible directions to reinforce the robustness and improve the performance of the multipoint approach. For instance, in order to reduce the potential large variations in Hamming distances, Gray coding can be utilized in the binary encoder. This allows for incremental changes in the input signals to be reflected as the smallest possible Hamming distances [17].

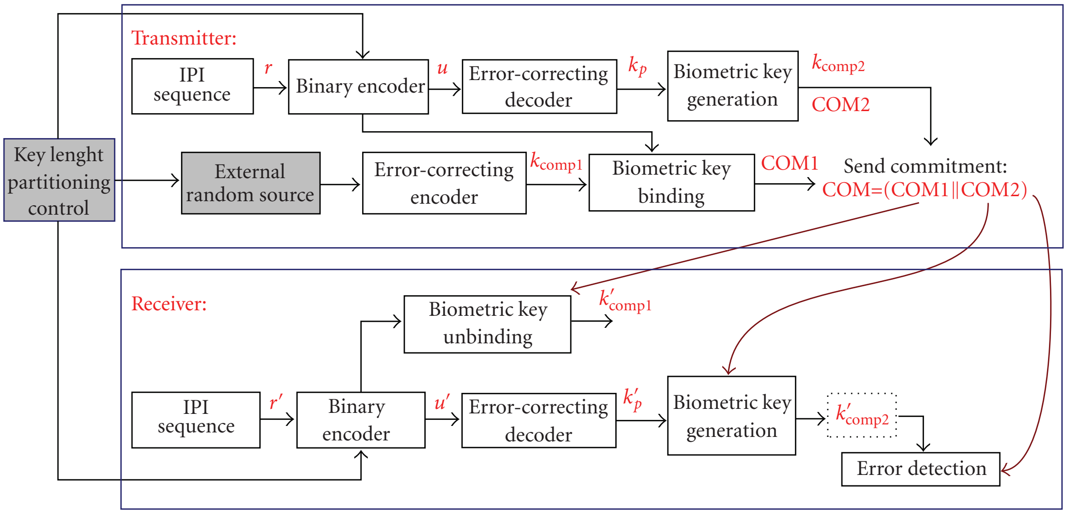


Figure 6: Multipoint management with key fusion.

Moreover, in order to improve the distances between the obtained IPI sequences and the codewords, an error-correcting code that takes into account some prior knowledge regarding the signal constellation is preferred. In other words, this is a superchannel approach, that seeks an optimal code that is most closely matched to the signal space. Of course, additional statistical knowledge regarding the underlying physiological processes would be needed.

Therefore, in the present paper, the performance results without these possible modifications will be evaluated, delivering the lower-bound benchmark upon which future designs can be assessed. It is expected that the false-rejection rates will demonstrate more significant gains. This is because, by design, the multipoint scheme can detect variations and errors with good accuracy (i.e., providing good false-acceptance rates). However, it is less robust in correcting the errors due to coding mismatches. And it is in this latter aspect that future improvements can be made.

In either scheme, there is also an implicit requirement of a buffer to store the IPI sequences prior to encoding. Consider the distribution of a 128-bit cryptographic key in a BSN, obtained from multiple time segments of nonoverlapping IPI sequences with the BCH code (63, 16, 11). Then, the number of IPI raw input bits to be stored in the buffer would be $(128/16) \times 63 = 504$ bits.

To assess the corresponding time delay, consider a typical heart rate of 70 beats per minute [15]. Also, each IPI value is used to generate 8 bits. Then, the time required to collect the 504 bits is approximately $(504/8) \times (60/70) = 54$ seconds. In fact, this value should be considered a bare minimum. First, additional computational delays would be incurred in a real application. Furthermore, the system may also need to wait longer, for the recorded physiological signal to generate sufficient randomness and reliability for the key generation. While the heart rate variations are a bounded random process [13], the rate of change may not be fast enough for a user's preference. In other words, a 504-bit sequence obtained in 54 seconds may not be sufficiently random. To address this inherent limitation, in trading off the time delay for less bandwidth consumption, a compromise is made in the next section.

4.3. Multipoint management with key fusion extension

In the system considered so far, the sole random source for key generation is the ECG. Without requiring an external random source, a multipoint strategy has enabled a BSN to be more efficient with respect to the communication resources, at the expense of computational complexity and processing delay. As discussed in Section 2.2, this is generally a desirable setup for a BSN [1, 2]. However, in operating scenarios where the longer delays and higher computational complexity become prohibitive, it is possible to resort to an intermediate case.

Suppose the security requirements dictate a certain key length. Then, the key can be partitioned into two components: the first constructed by an external random source, while the second derived from the ECG. The total number of bits generated equals the required key length. Evidently, for a system with severe bandwidth restriction, most of the key bits should be derived from the ECG. Conversely, when transmission delay is a problem, more bits should be generated by an external source.

A high-level summary of a possible key fusion approach is depicted in Figure 6. The key k_{session} is a concatenation of two components, that is, $(k_{\text{comp1}}, k_{\text{comp2}})$. The first component k_{comp1} is distributed using fuzzy commitment, while the second k_{comp2} is sent using the multipoint scheme.

In order to ensure that the overall cryptographic key is secured using mutually exclusive information, it is necessary to partition the output from the binary encoder properly. As a concrete example, let us consider generating a 128-bit key, half from a fuzzy commitment and half from a multipoint distribution, using a BCH (63, 16, 11) code. Then, the first $128/2 = 64$ bits from the raw binary output are used to bind the externally generated 64-bit sequence. The remaining 64 bits need to be generated from the next $(64/16) \times 63 = 252$ raw input bits. In other words, this scheme requires waiting for $64 + 252 = 316$ bits to be recorded, as opposed to 504 bits in the nonfusion multipoint case.

Therefore, from an implementation perspective, this fusion system allows a BSN to adaptively modify its key construction, depending on the delay requirements. But the disadvantage is the sensors need to be sufficiently complicated to carry out the adaptation in the first place. For instance, additional information needs to be transmitted for proper transceiver synchronization in the key construction. Furthermore, some form of feedback is needed to adjust the key length for true resource adaption. These requirements are conceptually represented by the key length partitioning control block in Figure 6. It can be practically implemented by embedding additional control data bits into the transmitted COM sequence to coordinate the receiver. As with most practical feedback methods, there is some inevitable delay in the system adaptive response.

Nonetheless, whenever implementable, a key fusion approach is the most general one, encompassing both the single-point and multipoint schemes as special cases, in addition to other intermediate possibilities.

5. INTRAS DATA SCRAMBLING

In the previous section, the general infrastructure and several approaches for generating and establishing common keys at various nodes in a secure manner have been described. The next straightforward strategy would be to utilize these keys in some traditional symmetric encryption scheme [9]. However, in the context of a BSN, this approach has several shortcomings. First, since conventional encryption schemes are not conceived with considerations of resource limitations in BSN, a direct application of these schemes typically implies resource inefficiency or performance loss in security. Second, operating at the bit-level, conventional encryption schemes are also highly sensitive to mismatching of the encryption/decryption keys: even a single-bit error, by design, results in a nonsense output.

Addressing the above limitations of conventional encryption in the context of a BSN, we propose an alternative method that operates at the signal-sample level. The method is referred to as INTRAS, being effectively a combination of interpolation and random sampling, which is inspired by [20, 21]. The idea is to modify the signal after sampling, but before binary encoding.

5.1. *Envisioned domain of applicability*

The proposed method is suitable for input data at the signal-level (nonbinary) form, which is typical of the raw data transmitted in a BSN. There are two fundamental reasons for this constraint.

First, for good performance in terms of security with this scheme, the input needs to have a sufficiently large dynamic range. Consider the interpolation process (explained in more detail in the next section): binary inputs would produce interpolated outputs that have either insufficient variations (e.g., consider linear interpolation between 1 and 1, or 0 and 0) or result in output symbols that are not in the original binary alphabet (e.g., consider linear interpolation between 1 and 0). More seriously, for a brute force attack, the FIR process (see (14)) can be modeled as a finite-state machine (assuming a finite discrete alphabet). Then, in constructing a trellis diagram [17], the comparison of a binary alphabet versus a 16-bit alphabet translates to 2^1 branches versus (potentially) 2^{16} branches in each trellis state. Therefore, working at a binary level would compromise the system performance. In other words, we are designing a symbol recoder. As such the method draws upon the literature in nonuniform random sampling [21].

Second, the scheme is meant to tolerate small key variations (a problem for conventional encryption), as well as to deliver a low-complexity implementation (a problem for fuzzy vault). However, the cost to be paid is a possibly imperfect recovery, due to interpolation diffusion errors with an imperfect key sequence. It will be seen that in the presence of key variations, the resulting distortions are similar to gradual degradations found in lossy compression algorithms, as opposed to the all-or-none abrupt recovery failure exhibited by conventional encryption. Therefore, similar to the lossy compression schemes, the intended input should also be the raw signal-level data.

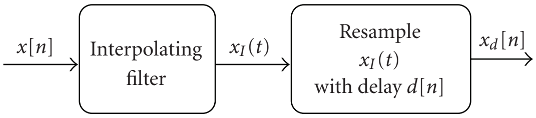


Figure 7: Interpolation and random sampling (INTRAS) structure.

5.2. *INTRAS high-level structure*

The general structure of an INTRAS scrambler is shown in Figure 7, with an input sequence $x[n]$. At each instant n, the resampling block simply re samples the interpolated signal $x_I(t)$ using a delay $d[n]$ to produce the scrambled output $x_d[n]$. Security here is obtained from the fact that, by properly designing the interpolating filter, the input cannot be recovered from the scrambled output $x_d[n]$, without knowledge of the delay sequence $d[n]$.

In a BSN context, the available (binary) encryption key k_{session} is used to generate a set of sampling instants $d[n]$, by multilevel symbol-coding of k_{session} [17]. This set of sampling instants is then used to resample the interpolated data sequence. Note that, when properly generated, k_{session} is a random key, and that the derived $d[n]$ inherits this randomness. In other words, the resampling process corresponds effectively to random sampling of the original data sequence. Without knowledge of the key sequence, the unauthorized recovery of the original data sequence, for example, by brute-force attack, from the resampled signal is computationally impractical. By contrast, with knowledge of $d[n]$, the recovery of the original data is efficiently performed; in some cases, an iterative solution is possible. Therefore, the

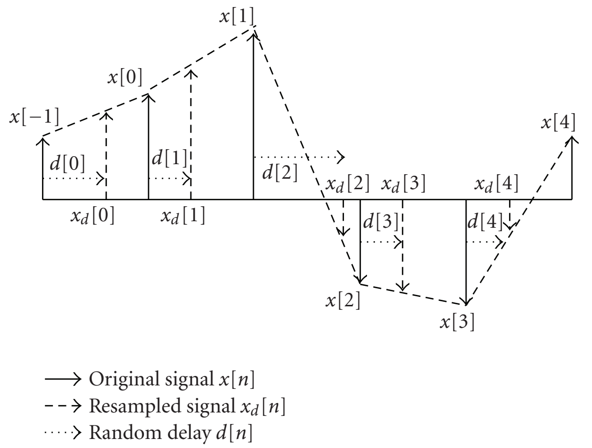


Figure 8: Graphical illustration of linear interpolation followed by random sampling.

proposed scheme satisfies the main characteristics of a practical cryptographic system. More importantly, it not only requires less computational resources for implementation, but also is more robust to small mismatching of the encryption and decryption keys, which is often the case in biometrics systems.

5.3. INTRAS with linear Interpolators

While Figure 7 shows an intermediate interpolated analog signal, $x_I(t)$, this is more or less a convenient abstraction only. Depending on the filter used and the method of resampling, we can in fact bypass the continuous-time processing completely.

First, the window size or memory length M needs to be selected, determining the range of time instants over which the resampling can occur. For a causal definition, the window needs to span only the previous data symbols. Then, the current output symbol is obtained as a linear combination of the previous symbols.

Consider a simple linear interpolator with $M = 1$, so that the window size is two symbols, consisting of the current symbol and one previous one. Then, the resampled signal $x_d[n]$ can be obtained in discrete-time form as

$$\begin{aligned} x_d[n] &= a_0[n] \cdot x[n] + a_1[n] \cdot x[n-1] \\ &= d[n] \cdot x[n] + (1 - d[n]) \cdot x[n-1], \end{aligned} \tag{12}$$

where $0 \leq d[n] \leq 1$. The rationale for this definition is illustrated in Figure 8. When $d = 0$, the output is the previous symbol. When $d = 1$, it is the current symbol. And for $0 < d < 1$, the filter interpolates between these values. This is precisely what a linear interpolator does but implemented entirely in discrete-time. The iterative definition (12) needs initialization to be complete: a virtual pre-symbol can be defined with an arbitrary value $x[-1] = A$.

Also, observe that computing $x_d[n]$ actually corresponds to computing a convex combination of two consecutive symbols $x[n]$ and $x[n-1]$, that is, weighting coefficients a_0, a_1 satisfy

$$a_0 + a_1 = 1, \quad a_0 \geq 0,\ a_1 \geq 0, \tag{13}$$

for each n. A convex combination is sufficient to maintain the full dynamic range (in fact, a more generalized linear combination is redundant, since it leads to unbounded output value).

The INTRAS structure is a scrambler because, depending on the random sequence $d[n]$, the output signal can differ significantly from the input. However, it is not encryption in the conventional sense, since knowing the input data and encrypted output is equivalent to knowing the key. Moreover, small mismatches in the decryption key do not lead immediately to nonsense output but rather represent a more graceful degradation, characterized by an increasing mean-squared error (MSE). This is in stark contrast to the all-or-none criterion of conventional encryption and is thus more suitable for biometric systems.

As the memory length M is increased, a number of possibilities can be applied in interpolation. For example, (i) the simplest approach is to simply interpolate between every two successive samples (graphically, joining a straight line). Then, the sampling delay determines which line should be used to pick the scrambled output. Or, (ii) linear regression can be first performed over the symbols spanning the window of interest [22]. Then, the sampling delay is applied to the best-fit regression line to produce the output. Alternatively, (iii) by revisiting the form of (12), which recasts interpolation as a convex combination, we can expand the formulation to incorporate a multiple-symbol combination as follows:

$$\begin{aligned} x_d[n] &= a_0[n]x[n] + a_1[n]x[n-1] + \cdots + a_M[n]x[n-M] \\ &= \sum_{i=0}^{M} a_i[n]x[n-i], \end{aligned} \tag{14}$$

where the convex combination condition, for a proper output dynamic range, requires that

$$\begin{gathered} \sum_{i=0}^{M} a_i[n] = 1, \\ a_0 \geq 0, \qquad a_1 \geq 0, \ldots, a_M \geq 0. \end{gathered} \tag{15}$$

Therefore, the cryptographic key k_{session} is used to encode $M + 1$ sequences of random coefficients. (Actually, because of the convex-combination requirement, there is a loss of degree of freedom, and only M sequences of this set are independent). Equivalently, the operation corresponds to a time-varying FIR filter [17] (with random coefficients).

This previous construction can be recast as a special case of the classical Hill cipher [9] as follows. Consider an input sequence $x[n] = \{x[0], x[1], \ldots, x[N-1]\}$ of length N. For the purpose of illustration, let us select $M = 2$, which implies that we need to initialize the first 2 virtual pre-symbols, $\{x[-2], x[-1]\}$ with assumed secret values, known to the intended receiver. One straight-forward approach would be to

Table 1: Matrix form of the convex-combination approach to linear interpolation.

$$\begin{bmatrix} A[N-1] & B[N-1] & C[N-1] & 0 & \cdots & 0 & 0 & 0 \\ 0 & A[N-2] & B[N-2] & C[N-2] & \cdots & 0 & 0 & 0 \\ \vdots & & & & & & & \\ 0 & 0 & 0 & 0 & \cdots & A[0] & B[0] & C[0] \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x[N-1] \\ x[N-2] \\ \vdots \\ x[0] \\ x[-1] \\ x[-2] \end{bmatrix} = \begin{bmatrix} x_d[N-1] \\ x_d[N-2] \\ \vdots \\ x_d[0] \\ x_d[-1] \\ x_d[-2] \end{bmatrix}$$

generate these symbols from the available cryptographic key k_{session}.

For notational simplicity, let us denote the coefficient sequences as $A[n] = a_0[n], B[n] = a_1[n], C[n] = a_2[n]$. Then, the remaining scrambled output symbols are computed for $n = 0, 1, \ldots, N-1$ as

$$\begin{aligned} x_d[0] &= A[0]x[0] + B[0]x[-1] + C[0]x[-2], \\ x_d[1] &= A[1]x[1] + B[1]x[0] + C[1]x[-1], \\ &\vdots \\ x_d[N-1] &= A[N-1]x[N-1] + B[N-1]x[N-2] \\ &\quad + C[N-1]x[N-3], \end{aligned} \tag{16}$$

which can be expressed in a matrix form:

$$\mathbf{A}\mathbf{x} = \mathbf{x_d}. \tag{17}$$

An expanded version of this equation is shown in Table 1. Here, we have rearranged the equations and augmented the last two rows with the virtual pre-symbols, so that the final form is explicitly recognized as a row-echelon matrix [22].

The obtained linear matrix representation is reminiscent of the Hill cipher, which is also a linear map modulo 26 (for 26 letters in the alphabet). However, there are some basic differences to be remarked here. First, the Hill cipher does not restrict the form of **A**, which can consist of any numbers. This means that the dimension of the mapping matrix needs to be small, otherwise matrix inversion would be prohibitively expensive. However, keeping the dimension small is equivalent to low security. Moreover, the Hill cipher is also unusable whenever **A** is singular.

In our proposed scheme, the above disadvantages are largely avoided. From the row-echelon form in Table 1, as long as $A[n] \neq 0$, for all n, then **A** has full-rank. Thus, the matrix equation will always have a solution, which is also unique. This shows that during the generation of random coefficients, the coefficient sequence $A[n]$ should be kept nonzero. In addition, the dimension of **A** is $(N + M) \times (N + M)$. For a typical signal sequence, this represents a large matrix, which would not be practical with a standard Hill cipher. But in this case, direct matrix inversion does not need to be performed. Instead an iterative solution can be obtained. Starting from the first symbol, we solve for $x[n]$, given $x_d[n]$, and the knowledge of the coefficient sequences and virtual pre-symbols. For $M = 2$, we start with

$$x[0] = \frac{x_d[0] - a_1[0] \cdot x[-1] - a_2[0] \cdot x[-2]}{a_0[0]}. \tag{18}$$

More generally, we have

$$x[n] = \frac{x_d[n] - \sum_{i=1}^{M} a_i[n]x[n-1]}{a_0[n]}. \tag{19}$$

Therefore, with the knowledge of the coefficient sequences and the virtual pre-symbols, the signal can be descrambled efficiently in an iterative manner.

Furthermore, the row-echelon representation also shows that without complete knowledge of the coefficient sequences, or the virtual pre-symbols, the original data sequence **x** cannot be uniquely solved. Indeed, a linear system either has: (1) a unique solution, (2) no solution, or (3) infinitely many solutions [22]. Missing any of the coefficients is tantamount to incomplete knowledge of a row of the echelon matrix, which then implies either case (2) or (3) only. And assuming that the echelon matrix **A** is properly constructed with $A[n] \neq 0$, for all n, then the incomplete **A** (with a deleted row whenever the corresponding delay symbol is unknown) still has full rank [22]. This then implies that case (3) is true: an intruder without knowledge of the delay sequences would need to guess from infinitely many possible choices in the solution space.

However, there is a catastrophic case in the current form (17): when **x** contains long runs of constant values, then the corresponding segment in $\mathbf{x_d}$ in fact does not change at all. This is because each row of **A** creates a convex combination. A simple fix involves using the bits from k_{session} to create a premasking vector that randomly flips the signs of elements in **x**. This is achieved by directly remapping the sequence of 0 and 1 in k_{session} to -1 and 1, which is called the sequence $\widetilde{\mathbf{s_R}}$. Since the goal here is to simply prevent long runs of a single constant value, rather than true randomness, it is permissible to stack together a number of $\widetilde{\mathbf{s_R}}$ sequences to create a longer (column) vector as follows:

$$\mathbf{s_R} = \begin{bmatrix} \widetilde{\mathbf{s_R}}^T \\ \widetilde{\mathbf{s_R}}^T \\ \vdots \\ \widetilde{\mathbf{s_R}}^T \end{bmatrix}. \tag{20}$$

Enough of the $\widetilde{\mathbf{s_R}}$ sequences should be concatenated (with a possible truncation of the last sequence) to make the dimension of $\mathbf{s_R}$ exactly $N \times 1$ (an N-element column vector, with elements being either -1 or 1). Then, a sign-perturbed input sequence is computed as

$$\widehat{\mathbf{x}} = \mathbf{x} \otimes \begin{bmatrix} \mathbf{s_R} \\ 1 \\ \vdots \\ 1 \end{bmatrix}, \tag{21}$$

where the last vector is augmented to account for virtual pre-symbols (which should not have long runs of constant values, being created from a random key), and $\otimes$ denotes element-wise multiplication. Then, the modified scrambling operation

$$\mathbf{x_d} = \mathbf{A}\widehat{\mathbf{x}} \tag{22}$$

is no longer limited by the above catastrophic case, since there are now deliberate signal perturbations even when the original input is static.

5.4. INTRAS with higher-order interpolators

As in conventional cryptography, the security of the system can be improved simply by assigning more bits to the key. However, this implies further resource consumption. An alternative, which offers a tradeoff with computational requirements, is to employ higher-order interpolators. This approach can be connected to Shamir's polynomial secret sharing scheme [23].

From the basic idea in Figure 7, the interpolating filter used is of a higher-order family. For illustration, we focus specifically on the class of Lagrange interpolators [21]. Note that such an approach has been previously applied for security, for example, in Shamir's scheme. However, there are a number of differences. First, in Shamir's approach, the secret is hidden as a particular polynomial coefficient, with the remaining coefficients being random. Moreover, there is no implication of a sliding-window type of interpolation. By contrast, the secret in the present paper is derived from a random sampled value, once the complete polynomial has been constructed. Second, the interpolation is applied sequentially over a limited sliding-window of values. These two characteristics make the implementation simpler and more appropriate for a BSN.

A Lagrange interpolator essentially constructs a polynomial $P_N(x)$ of degree N that passes through $N+1$ points of the form (x_k, y_k) and can be expressed as a linear combination of the Lagrange basis polynomials:

$$P_N(x) = \sum_{k=0}^{N} y_k L_{N,k}(x), \tag{23}$$

where the basis polynomials are computed as

$$L_{N,k}(x) = \frac{\prod_{\substack{j=0 \\ j \neq k}}^{N} (x - x_j)}{\prod_{\substack{j=0 \\ j \neq k}}^{N} (x_k - x_j)}. \tag{24}$$

For a set of $N+1$ points, it can be shown from the fundamental theorem of algebra that $P_N(x)$ is unique. Therefore, the degree N of the interpolator used in secret sharing needs to be one less than the number of available shares in order for a secret to be properly concealed. This can be explained alternatively by Blakley's related secret sharing scheme, which essentially states that n hyperplanes in an n-dimensional space intersect at a specific point. And therefore, a secret may be encoded as any single coordinate of the point of intersection.

The construction for the present BSN context is as follows. For clarity, we illustrate the construction for a system with memory length $M = 3$.

(1) In creating a new scrambled symbol, the focus is on the values within a limited window including 3 previous values. In other words, there are 4 values of interest at the current sampling index n, $(t[n], x[n])$, $(t[n-1], x[n-1])$, $(t[n-2], x[n-2])$, $(t[n-3], x[n-3])$, where $t[n]$ denotes the time value corresponding to sampling index n.

(2) These values constitute the available shares and are pooled together to construct a third-degree polynomial, that is, $P_3(t)$ for the current window.

(3) A new secret share is created corresponding to a random time value of $t_R \in T_W$, where T_W represents the current range of the window. The current share $(t[n], x[n])$ is replaced with the new share $(t_R, P_3(t_R))$ in the output signal.

(4) The construction moves to the next point similarly, until the whole sequence has been processed.

The descrambling operation by a receiver proceeds sequentially in the reverse manner.

(1) Due to the particular design of INTRAS, at each instant n, a total of M previous shares are available (in the initial step, these are the virtual pre-symbols).

(2) Therefore, with an incoming new share, $(t_R, x_d[n])$, and knowledge of t_R provided from the biometrics, the polynomial $P_3(t)$ for the current window can be completely reconstructed by the receiver.

(3) The original symbol or share $(t[n], x[n])$ can then be recovered.

(4) This recovered share then participates in the next sliding window. The process can thus be repeated until the entire sequence has been recovered.

Due to the similar construction based on Lagrange interpolators, the security of INTRAS is at least as good as Shamir's scheme for each sliding window. Furthermore, note that a new random delay for a new secret share needs to recovered, from the biometrics, for the next sliding window. Therefore, suppose a previous window was compromised, an intruder would still need to repeat the process for the next iteration, albeit the process is now easier, since at least M shares of the required $M+1$ shares have been previously compromised.

While the application of INTRAS with higher-order interpolation delivers improved security and flexibility, the disadvantage is a large increase in computational complexity, especially when the size of the memory M is substantial. Therefore, a sensible strategy would be to apply linear interpolation

Table 2: Performance of key generation and distribution at various coding conditions.

Parameters			Without key fusion		With key fusion	
No. of subjects	BCH Code	No. of DET bits	FRR (%)	FAR (%)	FRR (%)	FAR (%)
24	(63, 45, 3)	64	15.6	0.02	14.7	0.02
24	(63, 16, 11)	64	4.5	0.02	4.2	0.02
24	(63, 16, 11)	32	4.7	0.03	4.4	0.02
40	(63, 45, 3)	64	17.1	0.03	16.6	0.03
40	(63, 16, 11)	64	5.1	0.03	4.8	0.03
40	(63, 16, 11)	32	5.3	0.04	5.0	0.03

for the links between weaker sensors, whereas higher-order interpolation would be used for more capable sensors.

6. SIMULATION RESULTS

Even though the proposed methods should be applicable to other types of cardiovascular biometrics, ECG-based biometrics are the focus of performance assessment, since ECG data are widely available in various public databases. In the simulations, the ECG data, with R-R annotations, archived at the publicly available PhysioBank database are used [14]. These signals are sampled at 1 KHz with 16-bit resolution. In order to simulate the placements of various sensors in a BSN, ECG records that include multichannel signals, recorded by placing leads at various body locations, are specifically selected. Since these leads are simultaneously recorded, timing synchronization is implicitly guaranteed.

6.1. Key generation and distribution

Several key distribution scenarios, which are meant to illustrate the possible improvement in terms of communication resources, as measured by the corresponding spectral efficiency, are demonstrated. Table 2 summarizes the simulation parameters and resulting findings for a targeted 128-bit cryptographic key.

The coding rate for error-correcting coding as well as the number of bits used for channel error detection is varied. Note that, compared to the single-point scheme, the amount of information actually transmitted over the channel for key distribution is lower. The results illustrate that the error-correcting stage is crucial. If key regeneration fails at the receiver, then no amount of additional transmitted bits can make a difference, since no error correction is performed. On the other hand, if key regeneration is successful, then a smaller number of bits only negligibly degrade the key verification. The results without and with key fusion are shown for comparisons. Here, half of the key bits are derived from the biometrics, while the remaining from an external source.

The performance metrics utilized for comparison are the standard false rejection rate (FRR) and the false acceptance rate (FAR) [4, 11]. In each case, we experimentally optimize (by a numerical search) the Hamming distance threshold of the DET bit sequence in order to give the smallest FAR, and recorded the corresponding FRR. In other words, a minimum FAR is the objective, at the expense of a higher FRR. Note that this goal is not always appropriate; depending on the envisioned application a different, more balanced operating point, may be more suitable. In this case, the relevant operating point is contrived instead for a particular application: to supply the cryptographic key for a conventional encryption method. Evidently, for this scenario, if accepted as a positive match, the receiver-generated cryptographic key needs to be an exact duplicate of the original key. Otherwise, the conventional encryption and decryption procedure, which is mostly an all-or-none process, will fail even for a single-bit mismatch in the cryptographic key. This disastrous case is prevented by imposing a very small FAR. Therefore, the reported results show what can be correspondingly expected for the FRR. A more tolerant alternative to data scrambling is examined in the next section, where the feasibility of INTRAS is assessed.

The results for the key fusion scheme show only a minor changes compared to key distribution from only the biometrics. This is an indication that the biometrics are already providing a good degree of randomness for key generation. If this was not the case, the external random source (which is forced to generate statistically reliable random keys) would have resulted in significant improvement, since it would provide a much improved source of randomness for the key. But according to the obtained results, only slight changes are observed in the FAR.

6.2. Data scrambling

Using the MSE as a performance metric, Figure 9 shows the results for INTRAS that combines two consecutive symbols ($M = 1$) and a key sequence $d[n]$ constructed from a 128-bit key. In this case, the input symbols are simulated as an i.i.d. sequence of integers, ranging from -10 to 10. The distortions are modeled using a simple additive white Gaussian noise (AWGN) channel. Recall that, without any channel distortion, the INTRAS scheme can be summarized as follows. The scrambling step is

$$x_d[n] = \text{INTRAS}(x[n], d[n]), \tag{25}$$

with input $x[n]$ and key sequence $d[n]$. The corresponding descrambling step for ideal recovery of the original signal is

$$x[n] = \text{INTRAS}^{-1}(x_d[n], d[n]). \tag{26}$$

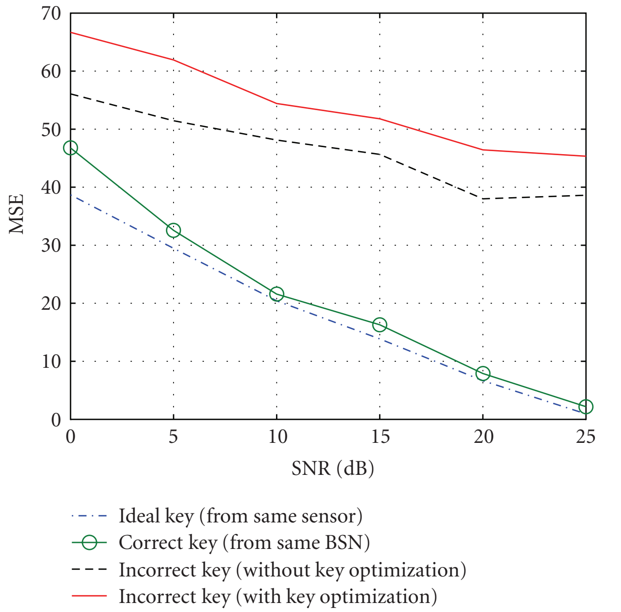


Figure 9: INTRAS data scrambling, with memory length $M = 1$.

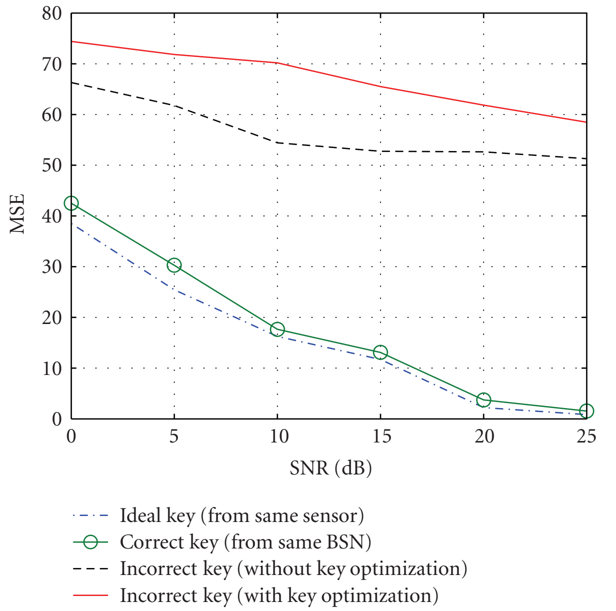


Figure 10: INTRAS data scrambling, with memory length $M = 3$ using Lagrange interpolation.

To account for the channel distortion, the signal seen at the input to the descrambler or receiver side is

$$\widehat{x_d[n]} = x_d[n] + v[n] = \text{INTRAS}(x[n], d[n]) + v[n], \tag{27}$$

where $v[n]$ is the AWGN. The associated channel signal-to-noise ratio (SNR) is computed as

$$\text{SNR} = \frac{\mathcal{E}\{|x_d[n]|^2\}}{\mathcal{E}\{|v[n]|^2\}}, \tag{28}$$

where $\mathcal{E}\{\cdot\}$ represents the statistical expectation operator.

Depending on the key used for scrambling, there are 4 recovery strategies shown in the results. Let $d[n]$, $d_{\text{BSN}}[n]$, $d_{\text{non-BSN}}[n]$, $d_{\text{non-BSN-opt}}$ be, respectively, the original key sequence used for scrambling, a key sequence from a device in the same BSN, a key sequence from an intruder outside of the intended BSN, without and with key optimization. Then, the four corresponding MSE performances, between the original signal and the signal recovered using one of these key sequences, can be computed. For example, when the original key is known as

$$\text{MSE}_{\text{Ideal}} = \text{MSE}(x[n], \text{INTRAS}^{-1}(\widehat{x_d[n]}, d[n])). \tag{29}$$

As shown in Figure 9, without knowledge of the key, the signal recovered by an intruder differs significantly from the genuine signal. Moreover, an increase in the signal-to-noise ratio does not lead to a significant improvement with an incorrect key. By contrast, with the correct key, the receiver performance improves as expected with better operating environments. The gradual change in MSE is analogous to the effect caused by varying the degree of compression in a lossy compression scheme.

With respect to key optimization, there appear to be only insignificant changes for the case with a correct key. However, with an incorrect key, there is a pronounced difference. This is an indication of improved security. An intruder would have more difficulty compromising a system with key optimization.

Next, in order to further improve security (for the same key length), additional processing cost is added by combining 4 symbols (with memory length = 3). As shown in Figure 10, the additional processing not only helps further separate the distinction of sensors inside and outside the BSN, it also improves the performance at high-noise situation for the authorized receiver. This is because each input symbol is now contained in a wider window of output symbols, so that the advantage of diversity is achieved.

7. CONCLUDING REMARKS

In this paper, methods using biometrics for efficiently providing security in BSNs have been proposed. Two complementary approaches addressing, respectively, the key management issues and the fuzzy variability of biometric signals are examined. One of the goals has been to allow for flexibility in each method. Depending on the actual application, a system can be accordingly reconfigured to be best suited for the required resource constraints. To this end, the proposed methods have built-in adjustable parameters that allow for varying degrees of robustness versus complexity.

While the proposed multipoint key management strategy and the INTRAS framework have specifically targeted relevant issues for a BSN, there remain important considerations that need to be addressed for practical deployment. Since a BSN is envisioned as a wireless network, the effects of channel fading and distortions should be considered. The robustness of the system needs to be evaluated in these practical scenarios. Furthermore, while the ECG and related signals have been touted as the most appropriate biometrics for a BSN, there is of course a wide range of sensors and devices that do not have access to the body's cardiovascular networks. Therefore, methods that allow for some form of interactions and management of these devices need to be considered for a BSN. In this manner, a BSN would be integrated more easily into other existing network systems without severe security compromises.

ACKNOWLEDGMENTS

This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC). Some of the material in this paper appears in the proceedings of Biometrics Symposium 2007, Baltimore, Maryland, USA.

REFERENCES

[1] S. Cherukuri, K. K. Venkatasubramanian, and S. K. S. Gupta, "Biosec: a biometric based approach for securing communication in wireless networks of biosensors implanted in the human body," in *Proceedings of the 32nd International Conference on Parallel Processing (ICPP '03)*, pp. 432–439, Kaohsiung, Taiwan, October 2003.

[2] S.-D. Bao, L.-F. Shen, and Y.-T. Zhang, "A novel key distribution of body area networks for telemedicine," in *Proceedings of IEEE the International Workshop on Biomedical Circuits and Systems*, pp. 1–20, Singapore, December 2004.

[3] F. M. Bui and D. Hatzinakos, "Resource allocation strategies for secure and efficient communications in biometrics-based body sensor networks," in *Proceedings of the Biometrics Symposium (BSYM '07)*, Baltimore, Md, USA, September 2007.

[4] C. C. Y. Poon, Y.-T. Zhang, and S.-D. Bao, "A novel biometrics method to secure wireless body area sensor networks for telemedicine and m-health," *IEEE Communications Magazine*, vol. 44, no. 4, pp. 73–81, 2006.

[5] K. K. Venkatasubramanian and S. K. S. Gupta, "Security for pervasive health monitoring sensor applications," in *Proceedings of the 4th International Conference on Intelligent Sensing and Information Processing (ICISIP '06)*, pp. 197–202, Bangalore, India, December 2006.

[6] W. R. Heinzelman, A. Chandrakansan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences (HICSS '00)*, vol. 2, pp. 3005–3014, Maui, Hawaii, USA, January 2000.

[7] M. Ilyas, Ed., *The Handbook of Ad Hoc Wireless Networks*, CRC Press, Boca Raton, Fla, USA, 2003.

[8] V. Shankar, A. Natarajan, S. K. S. Guptar, and L. Schwiebert, "Energy-efficient protocols for wireless communication in biosensor networks," in *Proceedings of the 12th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC '01)*, vol. 1, pp. 114–118, San Diego, Calif, USA, September 2001.

[9] W. Stallings, *Cryptography and Network Security: Principles and Practice*, Prenticall Hall, Upper Saddle River, NJ, USA, 4th edition, 2006.

[10] A. Juels and M. Wattenberg, "A fuzzy commitment scheme," in *Proceedings of the 6th ACM Conference on Computer and Communications Security (CCS '99)*, pp. 28–36, Singapore, November 1999.

[11] A. Cavoukian and A. Stoianov, "Biometric encryption: a positive-sum technology that achieves strong authentication, security and privacy," Information and Privacy Commissioner/Ontario, March 2007.

[12] J. Malmivuo and R. Plonsey, *Bioelectromagnetism: Principles and Applications of Bioelectric and Biomagnetic Fields*, Oxford University Press, New York, NY, USA, 1995.

[13] S. Lu, J. Kanters, and K. H. Chon, "A new stochastic model to interpret heart rate variability," in *Proceedings of the 25th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS '03)*, vol. 3, pp. 2303–2306, Cancun, Mexico, September 2003.

[14] A. L. Goldberger, L. A. N. Amaral, L. Glass, et al., "PhysioBank, physioToolkit, and physioNet: components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. 215–220, 2000.

[15] D. Bruce Foster, *Twelve-Lead Electrocardiography: Theory and Interpretation*, Springer, New York, NY, USA, 2nd edition, 2007.

[16] S. D. Bao, Y. T. Zhang, and L. F. Shen, "A new symmetric cryptosystem of body area sensor networks for telemedicine," in *Proceeding of the 6th Asian-Pacific Conference on Medical and Biological Engineering (APCMBE '05)*, Tsukuba, Japan, April 2005.

[17] J. G. Proakis, *Digital Communications*, McGraw Hill, New York, NY, USA, 4th edition, 2001.

[18] S.-D. Bao, Y.-T. Zhang, and L.-F. Shen, "Physiological signal based entity authentication for body area sensor networks and mobile healthcare systems," in *Proceedings of the 27th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS '05)*, pp. 2455–2458, Shanghai, China, September 2005.

[19] G. Kabatiansky, E. Krouk, and S. Semenov, *Error Correcting Coding and Security for Data Networks: Analysis of the Superchannel Concept*, John Wiley & Sons, New York, NY, USA, 2005.

[20] V. Valimaki, T. Tolonen, and M. Marjalainen, "Signaldependent nonlinearities for physical models using time-varying fractional delay filters," in *Proceedings of the International Computer Music Conference (ICMC '98)*, pp. 264–267, Ann Arbor, Mich, USA, October 1998.

[21] F. Marvasti, *Nonuniform Sampling: Theory and Practice*, Springer, New York, NY, USA, 2001.

[22] B. Noble and J. Daniel, *Applied Linear Algebra*, Prentice Hall, Englewood Cliffs, NJ, USA, 3rd edition, 1987.

[23] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.