Physiological Value-Based Efficient Usable Security Solutions for Body Sensor Networks

KRISHNA K. VENKATASUBRAMANIAN and SANDEEP K. S. GUPTA
Arizona State University

A Body Sensor Network (BSN) is a network of economically powered, wireless, wearable, and implanted health monitoring sensors, designed to continually collect and communicate health information from the host they are deployed on. Due to the sensitive nature of the data collected, securing BSNs is important for privacy preservation and protecting the host from bodily harm.

In this article, we present Physiological Value-based Security (PVS), a usable and efficient way of securing intersensor communication schemes for BSNs. The PVS scheme distributes the key used for securing a particular message along with the message itself, by hiding it using physiological values. In this way, it not only eliminates the need for any explicit key distribution, but also reduces the number of keys required at each node to meet all its secure communication requirements.

We further demonstrate the use of the PVS scheme in securing *cluster* topology formation in BSNs. Traditional protocols for cluster formation do not consider security and are therefore susceptible to malicious attacks. We present a PVS-based cluster formation protocol which mitigates these attacks. Performance analysis of the protocol shows that compared to cluster formation protocols secured with non-PVS-based key distribution schemes, it performs efficiently.

Categories and Subject Descriptors: C.2.0 [**Computer-Communication Networks**]: General—*Security and protection*; C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless communication; network topology*; C.4. [**Computing Systems Organization**]: Performance of Systems; K.6.5 [**Management of Computing and Information Systems**]: Security and Protection

General Terms: Security, Design

Additional Key Words and Phrases: Body sensor networks, efficiency, physiological values, secure cluster formation, usable security, environmentally-coupled security

ACM Reference Format:

Venkatasubramanian, K. K. and Gupta, S. K. S. 2010. Physiological value-based efficient usable security solutions for body sensor networks. ACM Trans. Sensor Netw. 6, 4, Article 31 (July 2010), 36 pages. DOI = 10.1145/1777406.1777410 http://doi.acm.org/10.1145/1777406.1777410

A preliminary version of this article appeared and won the Best Paper Award in the *Proceedings of the 4th International Conference on Intelligent Sensing and Information Processing*, 197–202.
This research was supported in part by National Science Foundation grant CNS-0617671 and CNS-0831544 and gift from MediServe Information Systems.
Authors' addresses: K. K. Venkatasubramanian and S. K. S. Gupta, IMPACT Lab, Ira A. Fulton School of Engineering, Arizona State University, Tempe, AZ, 85287; email: {kkv, sandeep.gupta}@asu.edu

DOI 10.1145/1777406.1777410 http://doi.acm.org/10.1145/1777406.1777410

1. INTRODUCTION

A *Body Sensor Network (BSN)* (a.k.a. Body Area Network) is a network of wearable and implantable wireless sensors which enables *pervasive, long-term, and real-time* health management for the *host* (patient) it is deployed on [Schwiebert et al. 2001]. It collects, processes, and stores physiological (such as electrocardiogram (EKG), and blood pressure), activity (such as walking, running, and sleeping), and environmental (such as ambient temperature, humidity, and presence of allergens) parameters from the host's body and its immediate surroundings; and can even actuate treatment (such as drug delivery) based on the data collected. BSNs can be very useful in assisting medical professionals to make informed decisions about the course of the patient's treatment by providing them with continuous information about the patient's condition.

Sensors form the essential basis of a BSN and come in different forms including wrist wearable [Lukowicz et al. 2002]; implantable [Laerhoven et al. 2003; Ziaie and Najafi 2001; Schwiebert et al. 2001]; and as a part of ambulatory devices and biomedical smart clothes [Paradiso et al. 2005]. They are *heterogeneous* in terms of capabilities, and are designed to be *unobtrusive* to the host. Consequently individual sensors in a BSN may have a very *limited* form factor, power source, memory, computation, and communication capabilities compared to generic sensor nodes, thus requiring BSNs to employ a large number of nodes in order to collect patient health data in a reliable and fault-tolerant manner. Each BSN has a controlling entity called the *base station* which collects and processes data for the BSN. All the sensors in the BSN communicate the data they collect to the base station at regular intervals through a multihop network. BSNs have many diverse applications including sports health management, home-based healthcare for the elderly, and postoperative care. Some of the prominent health monitoring BSN systems being built today are listed in the Table I.

As a BSN collects personal health data, *securing* it is very important. Lack of security may not only lead to loss of patient privacy, but may also physically harm the host by allowing adversaries to introduce bogus data or modifying/suppressing legitimate ones; thus, potentially inducing erroneous diagnosis and actuation. Therefore, securing a BSN requires preventing adversaries from: (1) joining the network as a legitimate node and introducing bogus health data, (2) preventing eavesdropping of confidential health data exchanged within the network, and (3) preventing health data from being reported or modified. Indeed, protecting the health data is a legal requirement as well. The Health Insurance Portability and Accountability Act (HIPAA) mandates that all personally identifiable health information be protected (http://www.hhs.gov/ocr/privacy/). One of the most vulnerable aspects of BSNs is the use of wireless medium for communication. This allows adversaries to monitor and modify the messages being communicated. Securing all intersensor communication in a BSN is therefore one of the most important aspects of securing the BSN itself.

Secure communication between sensors is well understood and essentially has two phases: *trust establishment* and *data communication* [Adelstein et al.

Table I. Prominent Pervasive Health Monitoring Systems

Project Name	Sensor Types	Purpose
UbiMon[a]	Wearable and Implantable sensors	Capturing transient but life-threatening medical events
HealthGear[b]	Wearable sensors	Detecting sleep apnea
MyHeart[c]	Biomedical smart clothes	Monitoring/detecting cardiovascular diseases
Code-Blue[d]	Wearable sensors	Emergency care, Disaster Response, and Stroke rehabilitation
Lifeshirt[e]	Wearable sensors	Continuous monitoring of vital signs
Ayushman[f]	Wearable sensors	Continuous monitoring of vital signs

[a] http://www.doc.ic.ac.uk/vip/ubimon/home/index.html.

[b] http://research.microsoft.com/~nuria/healthgear/healthgear.htm.

[c] http://www.hitech-projects.com/euprojects/myheart/.

[d] http://www.eecs.harvard.edu/~mdw/proj/codeblue/.

[e] http://www.lifeshirt.com.

[f] http://impact.asu.edu/Ayushman.html.

2005]. Trust establishment is a means of assuring the communicating entities that the other is legitimate. It is usually carried out through the establishment and use of cryptographic keys between the communicating entities. In this article we consider the use of only symmetric keys as a public key-based scheme (for trust establishment) can be very expensive [Perrig et al. 2002]. Once a symmetric key is established between two sensors, secure communication between them can take place during which both the sender and the receiver use the shared symmetric key to maintain data confidentiality and integrity, and to authenticate each other. We classify traditional symmetric key distribution schemes for sensor networks into three generic categories.

—*Predeployment-based*. These techniques require storing large set of keys in each sensor node before deployment. In case the physical deployment details are known, predeployment can be accomplished in a deterministic manner. For example, each node shares a pair-wise key with all its known neighbors, a group key for multicasting to its neighborhood, and a network-wide key to meet its communication requirements [Perrig et al. 2002]. However, given the long-term monitoring requirements of a BSN and static nature of the keys, extended exposure of the predeployed keys to cryptanalytic attacks may also pose a problem in addition to large storage space for the keys. Another predeployment mechanism is to store a set of keys, derived from a common key pool, at each node, such that the probability of finding a common key between any two sensors is high. These nondeterministic schemes, however, have an even larger space requirement than deterministic schemes, and may exhibit drastically poor scalability. Network re organization, that is, node addition, replacement, and redeployment in an alternate location, may be difficult in both cases. Prominent examples of nondeterministic predeployment schemes include Eschenauer and Gligor [2002], Chan et al. [2003], Du et al. [2005], Liu and Ning [2003a], and Pietro et al. [2003].

—*Communication-based*. These techniques require some form of communication between the entities in the network for key distribution. Such schemes either obtain keys from a central entity, such as a base station, or require exchange of information, such as random numbers or node IDs, which along with a predeployed *master key* is used for generating a shared key. In case the key exchange protocol involves a large number of steps, the use of master key may prove to be very expensive for sensors. Space-wise they too require multiple keys to meet the unicast and group communication requirements. Examples include Zhu et al. [2006] and Lai et al. [2002].

—*Public key cryptography-based*. These techniques use a pair of related public and private keys along with mathematically complex and computationally intensive algorithms to distribute symmetric keys between sensors. However, as mentioned before the limited capabilities of sensors makes them prohibitively expensive for key distribution in a BSN. Examples include Malan et al. [2004] and Huang et al. [2003].

It can be seen that each of these protocol classes assumes some form of *predeployment* which itself requires an underlying assumption of trust. This predeployment acts as an initial source of trust and forms the basis of distributing the actual key to be used in secure communication between sensors. However, this assumption makes them unsuitable for BSNs. The main reasons for this stem from the following insecurities and hinderance to usability associated with predeployment.

—If the predeployment takes place at the factory where the sensors are manufactured, then hosts cannot trust the keys unless the entire key distribution chain from the factory to the host is secured [Kuo et al. 2007].

—If the predeployment is to be executed by the host, it would require them to make important decisions about the keys to be used. This might result in poor-quality keys, adversely affecting the security of the system.

—With predeployment adding or moving nodes within the network would require additional host involvement. For example, if a node is added to the network, all the nodes in the neighborhood have to have their keys updated.

—With predeployment, it is very difficult to change the keys in the network which have been compromised.

Over the years, some solutions have been proposed for secure predeployment, such as Message-In-a-Bottle [Kuo et al. 2007] or Resurrecting Duckling [Stajano and Anderson 1999]. These techniques expect the presence of additional equipment (such as a Faraday cage) or extra communication interfaces on the sensors for a side-channel key deployment (e.g., infra-red and ultrasound) and consequently involve considerable user involvement, making them unsuitable for the BSNs. As BSNs become more prevalent and worn by even healthy people with no chronic ailments, for preventive or nonmedical applications (e.g., fitness monitoring), the overhead imposed by predeployment (in terms of security, user involvement, and general network management) will adversely affect the *usability* of the system. We need a forward-looking security solution which follows the “plug-n-play” paradigm, is transparent, and easy to use, which not only meets today’s needs but is also flexible enough to be used in the future.

In this article we present a novel and efficient secure intersensor communication scheme called *Physiological Value-based Security (PVS)*. The PVS scheme utilizes a specific stimulus from the environment of deployment (human body) for distributing keys. The idea is to secure (encrypt and/or integrity protect) the data using an arbitrary key (*key*) and transmit the secured data along with the *key* obfuscated using the *Physiological Value (PV)* at the sender. The receiver uses its version of the PV (which is highly correlated, given the same underlying source) to unhide the *key* and then verify and/or decrypt the data received. The following is a simple usage scenario of PVS: a person may be using an in-vivo or a wearable glucose monitoring sensor as a part of the BSN they wear regularly, for a weight loss program. If the person later develops diabetes they may now require an insulin pump or some other diabetes treatment device. Such a device would need inputs from glucose sensors (potentially from several patient sites) requiring a secure communication between the two. Using the PVS scheme, it would be possible for the sensor and the insulin pump to agree on a symmetric key between themselves as a result of the placement of the pump within the BSN. In summary, some of the advantages that the PVS scheme brings to BSNs include the following.

—It has the potential to completely *eliminate the need for predeployment of keys* in BSNs due to the use of keys generated from physiological stimuli.

—It also eliminates the need for any additional explicit key distribution after the network is set up (e.g., Zhu et al. [2006] and Eschenauer and Gligor [2002]). Keys are distributed during data communication, thereby removing any additional communication steps and improving the *energy efficiency* of BSNs.

—It improves the *space efficiency* of BSNs since a single PV measurement is enough at any given time to uniformly secure all unicast, multicast, and broadcast communication.

—It allows adding, moving, or removing nodes without having to rekey sections of the network, by simple remeasurement of PV. This is achieved by utilizing the dynamic nature of the human body which produces time-variant stimuli [West 2006].

—It makes security provisioning in BSN transparent. Simply deploying the sensors is enough to make them communicate securely. It thus takes a more *plug-n-play* character. This greatly improves the *usability* of security in BSNs.

BSNs can now be easily managed by their hosts and, if necessary, the host’s caregivers. The use of the PVS scheme thus removes an important hurdle in making BSNs as a whole plug-n-play, which in turn improves their chances of being widely accepted.

Applicability of the PVS scheme is not limited to stand-alone intersensor communication security. Interestingly, it can act as a building block for larger, more complex communication protocols as well. To illustrate this point, we present its application in efficiently securing *cluster topology* formation in BSNs [Shankar et al. 2001]. Clusters are one of the most energy-efficient topologies for sensor networks. They are formed by grouping nodes within a BSN into clusters and designating one node within each cluster as a leader. The leader is responsible for aggregating and communicating the data generated by its cluster members to the base station. One of the common ways of forming clusters in a network is based on measuring the signal strength of leader solicitations [Heinzelmann et al. 2000]. This approach is, however, fraught with risks as it can allow malicious entities to form *sinkholes* within the network [Karlof and Wagner 2003]. In this article a secure cluster topology formation protocol is presented, called *Distributed Cluster Formation using Physiological Value-based Security (DCF-PVS)*, which eliminates the problem of sinkhole formation. The protocol actively makes use of the PVS scheme for securing the intersensor communication aspect of its workings. The DCF-PVS scheme has been analyzed in-depth in terms of security it provides. Further, its performance has been analyzed in terms of energy consumption, and compared with alternate versions of secure cluster formation which use traditional key distribution schemes for securing the intersensor communication, instead of the PVS scheme.

The article is organized as follows: Section 2 presents the system model and design goals for developing the PVS scheme. Section 3 presents the PVS scheme in detail focusing on choosing the appropriate PVs, using them for securing intersensor communication, as well as its security and performance analysis. Section 4 presents a PVS-based secure cluster formation protocol, along with security analysis, prototyping results, performance analysis, and comparative results. Section 5 presents the related work followed by Section 6 which concludes the article. In the rest of this work we use the term network to mean a BSN unless specified and the terms adversary, malicious node, and malicious entity are used interchangeably.

2. SYSTEM MODEL

In this section we present our system model and design goals before delving into the details of the PVS scheme.

2.1 Body Sensor Network

A BSN is a health monitoring network of sensor nodes, implanted or worn by a person called the *host*. The sensor nodes in the BSN are *heterogeneous*, possessing the ability to measure multiple stimuli[1] from the host's body. They are assumed to be built to survive extreme conditions such as variation in temperature and presence of water [Paradiso et al. 2005], and are powered using mechanisms such as body movements, body heat production, and biofuels such as blood glucose [Lo and Yang 2005]. We assume the sensor nodes *communicate through the wireless medium*, as wires running between sensors in a BSN will make it obtrusive, especially in the case of implanted sensors or

[1] Already physiological monitoring sensors are able multimodal and are able to sense multiple types of stimuli [Laerhoven and Gellersen 2004; Ouchi et al. 2002].

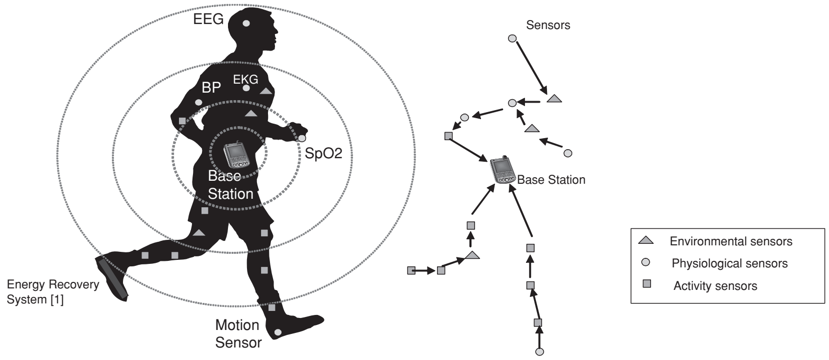

Fig. 1. A health monitoring body sensor network.

when there is a need to reconfigure the placement of sensors on the body. In this article we use the terms nodes, sensors, and sensor nodes interchangeably.

BSNs can have a large number of nodes. It has been suggested in Laerhoven and Gellersen [2004] that a large number of low-quality sensors can perform the task of monitoring as effectively as a few high-quality sensors. Such a network will have multiple sensors measuring the same stimuli providing redundancy and therefore better fault tolerance, while at the same time being *less intrusive*, *energy efficient*, and conducive for regular wearing due to their *lightweight and small form factor*. Networks with sizes of up to 192 and 255 have already been proposed in Kern and Schiele [2003] and Choi et al. [2006], respectively. The BSN is therefore assumed to be a potentially dense network consisting of nodes numbering in the hundreds. A *base station* is used to collect data from the entire BSN, as it has significantly higher computational and communication capabilities compared to the sensors (see Figure 1).

The sensors once deployed are static (no movement) in nature, but can be adjusted as and when required.[2] Moreover, as the sensors are expected to work for long time durations (in some cases the entire lifetime of a patient), such adjustments may be needed from time to time to ensure correct operation of the BSN. This ability to control/manage every aspect of sensor deployment makes them fundamentally different from traditional sensor networks. Nodes in a BSN can be managed at multiple levels unlike traditional sensor networks.

(1) *Node level*. Nodes have to be placed such that they measure various physiological values accurately enough to communicate. For this reason, the physical packaging and placement of nodes and any contact they have with the host's body can be manipulated to minimize noise and measurement artifacts.
(2) *Link level*. For each node, its ID, location of deployment, PVs it can measure, and IDs of all the nodes within its range are known and can be controlled.
(3) *Network level*. Sensors in the BSN may have to undergo *adjustments* from time to time (changing sensor contacts and interface with the host and so

[2] We believe the assumption can be easily relaxed without considerably changing our scheme.

on) along with *reorganization* (changing faulty sensors, adjusting sensors' location, contact, and placement, addition of new improved sensors, and removal of faulty, old sensors) to ensure correct operations over long-term deployments.

Link-level control is provided to some extent by traditional sensor networks, where details about the nodes and their immediate neighborhood are assumed to be known. The latter two, however, are unique to BSNs.

2.2 Threats to Body Sensor Networks

BSNs potentially face many *threats*, due to the sensitive nature of the data they collect and the broadcast nature of the wireless medium they use to communicate. The threats originate from two sources: *active* and *passive adversaries*. *Active adversaries* have the capability to eavesdrop on all traffic within a BSN, inject messages, replay old messages, and spoof nodes in the BSN in order to become part of the network. Active adversaries, if successful, not only can invade a patient's privacy but can also suppress legitimate data or insert a bogus one into the network leading to unwanted actions (drug delivery) or preventing legitimate actions (notifying doctor in case of an emergency). *Passive adversaries*, on the other hand, are only capable of eavesdropping on the message exchanged within the BSN. Additionally, they may potentially be able to perform offline cryptanalytical attacks to access any confidential data being communicated, thereby invading a patient's privacy. They do not try to interfere with the functions of the BSN. Therefore, maintaining confidentiality, integrity, and authenticity of the communicating entities against active and passive adversaries is of paramount importance in a BSN.

In order to address these threats, we assume the following *trust model* for BSNs: The wireless medium is not trusted. The sensors do not accept any message they receive unless they can authenticate the sender. The base station is assumed to be completely trustworthy, nontamperable, and can measure a variety of PVs. Note that in this work we do not address denial-of-service attacks, such as signal jamming, battery depletion, or malicious electronic interference [Wood and Stankovic 2002]. Further, we assume that adversaries are not in contact with the host's body. Sensors in the BAN deployed on the host are assumed to be legitimate and functioning correctly. We do not consider physical compromise of nodes in this article. A physically compromised node would have to be in intimate contact with the host. Although this is possible it is unlikely due to logistical reasons; it is not easy to insert a node inside a human body without the knowledge of the host or the host's surgical team (which is trusted) during an operation and anything worn is mostly under supervision of the host (if the host is not capable, then we assume her to be under the supervision of a caretaker). Even though physical compromise is not considered, the issue of security is still of great significance to BSNs due to threats resulting from the wireless channel. It was recently demonstrated by researchers from the University of Washington and University of Massachusetts, Amherst, that an implanted defibrillator could be hacked to reveal the patient's health data as well as administer an untimely shock [Halperin et al. 2008]. This attack was possible due to the open wireless channel used by the device to communicate with its programmer. Such an attack can be easily extended to BSNs where sensors collect patient data, actuate treatment, and perform wireless communication between one another to relay patient data to the base station. Security is therefore very important in BSNs.

2.3 Design Goals

Any secure communication scheme for BSNs should provide the basic services of confidentiality, integrity protection, and authentication. In addition, they should meet certain goals including the following.

—*Efficiency*. The limited capabilities of sensors make energy efficiency an important aspect of a BSN's design. Cryptographic and protocol requirements of secure communication impose severe load on the sensors of a BSN in terms of energy. Though sensors are assumed to be able to recharge, frequent energy depletion will hinder the continuous monitoring requirement of a BSN. Energy-efficient secure communication is therefore critical for a BSN. Further, BSNs have limited memory available at each sensor. Therefore secure communication schemes for BSNs should be space efficient with respect to the number of keys they need to store.

—*Support for different types of secure communication*. Two types of communication are possible with a BSN: unicast (one-to-one communication) and group communication (one-to-many or many-to-one communication). Security mechanisms for BSNs have to ensure confidentiality, integrity, and authentication for both these communication types.

—*Usability*. Security solutions for BSNs have to be usable. We define usable security solutions as those which activate on deployment, in a plug-n-play manner, with minimal (ideally none) initialization procedures.

Both energy and space efficiency ensure the *scalability* of the BSN. As we will see in the next few sections, PVS-based secure communication meets all these goals.

3. PHYSIOLOGICAL VALUE-BASED SECURITY

Physiological Value-based Security (PVS) is a novel secure intersensor communication scheme for BSNs which utilizes specific Physiological Values (PVs) as a basis for maintaining communication security. Physiological values are stimuli generated from the various functions performed by the human body. Examples of PVs include heart rate, temperature, and blood glucose level. The scheme works in four steps: (1) The communicating entities measure a preagreed PV, successive values of which are first encoded into a binary string; (2) the sender node uses an arbitrary key to secure (encrypt/integrity protect) any data it wants to communicate, hides the key using this binary representation of the PV, and transmits both the hidden key and secured data as a single message; (3) the receiver node, on receiving the message, retrieves the arbitrary key using the local version of PV and retrieves/verifies the data received. Once these steps are executed, any subsequent secure communication between

Table II. Range of Commonly Encountered Physiological Values (albeit unsuitable for PVs).

Physiological Values	Range
Blood Glucose	64-140 mg/dL[a]
Blood Pressure	120-160 mmHg (systolic)[b]
Temperature	97.0-105.0 F[c]
Hemoglobin	12.1-17.2 g/dL[d]
Blood Flow	>0.9 ABI(normal), <0.5 ABI (abnormal)

[a] Varies with activity.
[b] Range is from hypotension to hypertension.
[c] Range across ages and normal and abnormal conditions.
[d] Varies between men and women with respect to age and altitude.

the sensors can take place without explicit PV measurement except in special cases, for example if the network is being reconfigured. An adversary who is not in contact with the host's body will not be able to accurately measure the PV and therefore cannot influence the secure communication taking place between the sensors. Before going further, we present the notations used in the rest of the article.

—$\{d\}_k$ is the encryption of the data d using the key k.

—$MAC(k, d1|d2|d3)$ is the computation of a Message Authentication Code (MAC) on the data $d1$, $d2$, and $d3$ using the key k.

—$\oplus$ is the bit-wise XOR operator.

—$s \rightarrow r : \langle d1, d2 \rangle$ indicates transmission from s to r of a message containing two components $d1$ and $d2$.

—If *msg* represents the message communicated between two nodes and suppose it consists of three elements $msg = \langle e1, e2, e3 \rangle$ then $msg.e1$, $msg.e2$, and $msg.e3$ refer to the first, second, and third element of the message, respectively.

3.1 Physiological Values: Issues and Properties

To be useful, PVs need to posses the following properties: (1) *length and randomness* to prevent any brute-force guessing, (2) *universal measurability* in all people and not just someone with specific conditions, (3) *time variance* to prevent guessing of future values if present value is compromised, and (4) *distinctiveness* in the values of a given PV, for two people, at any given time. Most commonly observed PVs vary within a very small range for most human beings and are therefore not suitable for our purposes (see Table II [Cherukuri et al. 2003]). Identifying PVs which possess the properties specified above is an important step in implementing the PVS scheme.

3.1.1 *PV Choice and Measurement.*

Building upon our initial idea of using physiological values, the use of Inter-Pulse-Interval (IPI) (also known as heart rate variability) as cryptographic keys has been recently proposed in Poon et al. [2006], Bao and Zhang [2005], and Bao et al. [2005]. In Bao et al. [2005] the authors measure IPI from two Photoplethysmogram (PPG) time series obtained from the same person at the same time. The IPI values were computed

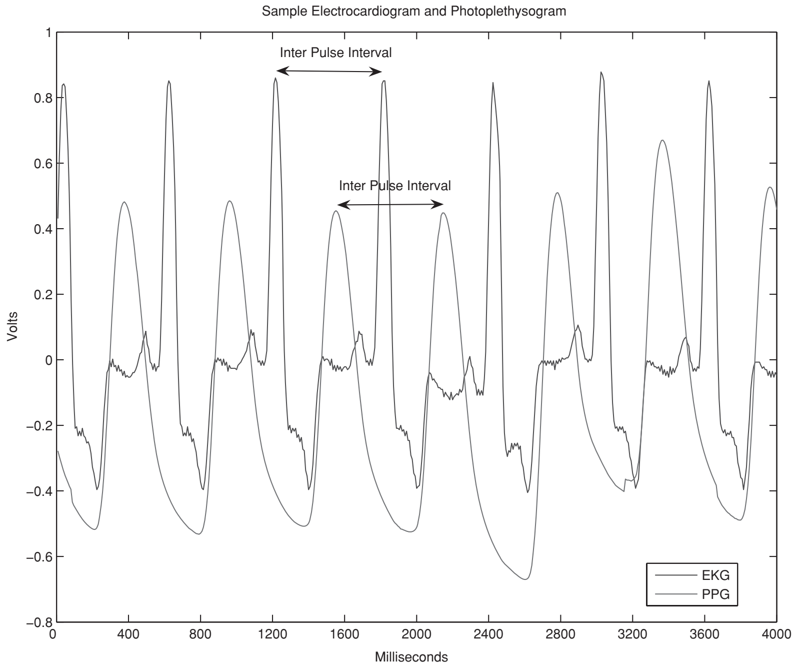

Fig. 2. Sample EKG, PPG signal and inter-pulse-interval.

by measuring the time difference between the peaks in the PPG signal. This series of IPI values was then encoded into binary to form a 128-bit PV. The Hamming distance between the two 128-bit PVs thus obtained was shown to about 90 bits when measured in two different people and around 1-6 bits for the same person in most of the cases, which makes it ideal for our purposes. In Poon et al. [2006] the value of IPI is measured by computing the time difference between the peaks from two sources: EKG and Photoplethysmogram. The resulting samples were also encoded into a 128-bit sequence. It was found that for the IPI measured from people with no ailments, the *false rejection* (when two sequences measured from the same individual do not match) and *false acceptance* (when two keys measured from the different individual match) were about 0.01. Figure 2 illustrates the EKG and PPG signals and the IPI value. An interesting property of IPI as a PV is that it possesses an additional property of being measurable from multiple sources (EKG and PPG). Using PVs like IPI, different sensors can be designed to measure different stimuli while using a common PV to secure their communication.

Finally, note that we cannot expect all sensors in a BSN to be able to measure a common PV. Therefore, there is a considerable need to identify newer physiological values which can be used with the PVS scheme. Identifying appropriate physiological values is an active research problem. Recent work in utilizing EKG [Venkatasubramanian et al. 2008a] and PPG [Venkatasubramanian et al. 2008b] signals directly for agreeing upon keys can be used here. In the rest of this article we assume the PV used by all the sensors in the BSN is the IPI .

3.1.2 *PV Key Strength and Rekeying.* As mentioned earlier, PVs such as IPI can provide about 90 bits of security (deduced from the observation that IPIs between two different people are 90 bits apart) [Bao et al. 2005]. Therefore, once a key has been agreed between a pair of nodes, PV measurements are seldom required for subsequent communication, unless the network configuration is changed, since a 90-bit key would take about 135 years [Schenier 1996] to brute force, rekeying need not be done frequently.

3.1.3 *Topographic Specificity in PVs.* The human body shows a high degree of topographic specificity, that is, a given PV measured at two points on the body tends to produce dissimilar values [McWilliams 2003]. For example, a sensor located at the host's arm will sense a slightly different value for a specific PV compared to a sensor located at the leg or torso. This is one of the reasons why the IPI values, in the previous section were similar but not identical. One of the ways to overcome this problem of small differences in the PVs is to view them as errors introduced during data transmission between communicating sensors [Cherukuri et al. 2003]. A simple Error Correction Function (ECF) can therefore be used to alleviate this problem of slight differences in PVs at the sender and receiver. More details are presented in Section 3.2.1.

3.1.4 *Synchronization Required for PV Measurement.* One of the important properties of the PVs is that they are time variant and vary unpredictably. This prevents malicious entities from guessing the current value of the PVs or knowing future values if the current value is known. But this temporal variation also poses problems for the sensors trying to use the PV to communicate securely. We therefore need to ensure that the communicating sensors see (almost) identical copies of PVs. A very important question in this regard is: what kind of synchronization is required for PVS? This will of course depend upon how the PV is measured. For IPI the values can be derived from the EKG and PPG signal based on the time difference between two adjacent peaks, as shown in Figure 2. To ensure the communicating sensors obtain the same IPI values, the first peak that they see in the EKG and PPG time series should belong to the same beat. This can be easily achieved by considering the PPG peak immediately following an EKG peak. If we compute the average time difference between the observance of the EKG and PPG peaks for a beat then this time difference can be used as a threshold for the level of synchronization required between the sensors.

To estimate the level of synchronization required for PVS scheme, we downloaded EKG and PPG data from the MIT PhysioBank MIMIC database.[3] For the eleven patients whom we considered (based on the availability and completeness of the EKG and PPG data), the average time difference between an EKG peak and the corresponding PPG peak was found to be μ = 250 msec, with standard deviation of $\sigma = \pm$ 6 msec. Therefore we posit that to utilize IPI for securing intersensor communication, the level of synchronization required should be in hundreds of milliseconds. Given the capability of sensor

[3] http://www.physionet.org/physiobank/database/mimicdb/.

synchronization protocols such as Elson et al. [2002] to reach a few microseconds of synchronization, we contend that for measuring IPI only a relatively loose synchronization between the communicating sensors is required.[4]

3.1.5 *PV Measurement.* Now that we know the level of synchronization that is required for the PVS scheme to function, we have to determine when two nodes have to start measuring PVs. With loose synchronization required, this can be achieved in a simple manner.

—When a node (designated sender) wants to communicate with another, it sends a *measurement-sync* (with the ID of the receiver in it) message to the receiver (see Section 3.2.2 for the multicast case) and starts measuring the prechosen PV.

—The propagation delay is considered to be effectively zero, as the signal (electromagnetic radiation) travels at the speed close to *c* (speed of light in free space) and the synch message has to typically reach the distance of about 0.5m on an average. The few nanoseconds the signal takes to cover this distance is well within the margin of error that can be tolerated by our synchronization scheme.

—The receiver, upon receiving the message, starts measuring the PV.

The act of broadcasting a *measurement-sync* message in the open does not cause any security threats because the only consequence of sending the *measurement-sync* message is the resulting PV measurement. No valid message exchange can take place as a result of the synchronization because the adversary has no way of generating a legitimate PV-based key. Additionally, reusing older PV measurements will also not work because of the time variance of PVs. Further, we contend that the *measurement-sync* messages will be used rarely, given the strength of appropriately chosen PVs against brute-force attacks (Section 3.1.2).

It is possible that between the times the *measurement-sync* message is received and the PV measurement begins, the clock (at the receiver) starts to drift. This drift, for a typical crystal oscillator, is usually of the order of 100μsec per second and will not affect the PV synchronization [Elson et al. 2002]. It is also possible that the sensors receiving the *measurement-sync* message do not start at exactly the same time due to reception delays. In Elson et al. [2002] the maximum reception delay in the case of Berkeley motes has been reported to be about 53μsec. Assuming a similar value for BSNs, the misalignment in the start of the PV measurement should not be significant. Similarly, platform-specific delay factors such as interrupt requests and sleep cycles will also not be a cause for concern.

[4] Note that different PVs will have different synchronization requirements. For example, for an EKG-based PVS scheme, the level of synchronization would be limited by the sample rate. The data we utilized was sampled at 125 Hz, that is, one sample was generated every 8 msec. To ensure no data points are missed at either sensors, the sensor clocks have to be synchronized to within 8 msec.

```
PVS_SEND()
if(Data to Send)
    if(PVExists == false)
        send(measurement-synch)
        Measure chosen PV ($PV_s$)
    endif
    Generate RandKey
    mac = MAC(RandKey, Data)
    $\gamma$ = RandKey $\oplus$ $PV_s$
    CERT[Data] = <mac, $\gamma$>
    msg = <Data, CERT[Data]>
    send(msg)
endif
```

```
PVS_RECV()
Packet = receive()
if(Packet == measurement-synch)
    Measure chosen PV ($PV_r$)
if(Packet == msg)
    RandKey''= ECF($PV_r$ $\oplus$ msg.CERT[Data]. $\gamma$)
    mac' = MAC(RandKey'', msg.Data)
    if(msg.CERT[Data].mac == mac')
        Accept msg.Data Received
    else
        Reject msg.Data Received
endif
```

Fig. 3. PVS execution at the sender and the receiver in a BSN.

3.2 Secure Intersensor Communication Using PVS

In this section, we present the PVS protocol for securing intersensor communication. We begin by illustrating the specific case of securing unicast communication between two sensors, followed by a short discussion on how to extend it for secure group communication. We then move on to analyzing its security properties and performance.

3.2.1 *Secure Unicast Communication.* The functions *PVS_SEND* and *PVS_RECV* in Figure 3 show the working of the PVS protocol for secure (integrity protected) unicast communication. Using PVS for encrypted (confidential) communication is a trivial extension. Initially, both the sender and the receiver synchronously measure a preagreed PV (i.e., when *PVExists* is false, sender sends the *measurement-sync* message). In case they have a previously measured PV, they simply skip the measurement part. Now, if the sender has data to send it will: (1) generate an arbitrary key called *RandKey*; (2) compute the physiological certificate on the data (*Data*) to be sent, denoted as *CERT[Data]*, by computing a MAC on *Data* using *RandKey*, hiding *RandKey* by XORing it with its version of PV (PV_s) to form γ, and concatenating the MAC and γ; and (3) transmit the data and the physiological certificate. The receiver upon receiving this message: (1) retrieves the arbitrary key by XORing the received γ with its version of PV (PV_r) and performs error correction on the result to correct for difference; (2) the resulting $RandKey''$ is used to compute the MAC on *Data* received and it is compared with the MAC in *CERT[Data]*; if they match, the data is accepted, else it is rejected. The successful verification of the MAC assures the receiver that the sender is a sensor on the same host as itself. The integrity of the *Data* is also verified by this process. Note that the *CERT* is much larger in length compared with a traditional MAC as it is used to distribute keys as well. For sending confidential information using PVS, the sender computes $C = \{Data\}_{RandKey}$ and $mac = MAC(RandKey,C)$ as a part of the physiological certificate. The receiver unhides *RandKey* and decrypts the *Data*.

It should be noted that once this process is executed, the sender and the receiver can use the measured PV (PV_s and PV_r, respectively) for securing all subsequent communication between them. No repeat measurement is necessary. New PV measurements may, however, be required from time to time if the network is being reconfigured. We end this section with a simple example of PVS executed between two sensors which do not have a common key. This example uses the Hamming code as the error correction function to correct 1-bit errors between the measured PVs. Other error correction functions can also be used as long as the sensors can handle the computational and memory requirements.

Example 1. Here, we demonstrate an example execution of PVS.

—Let the PV measured at the sender and receiver produce PV_s = 0101011 and PV_r = 0111011, respectively. They are one bit apart due to topographic specificity of the body, which we need to correct.

—Assume that the sender chooses a 4-bit key *RandKey* = 1110. The sender computes a MAC on the data it wants to send using this *RandKey*.

—Let f be a Hamming code error correction function which has the ability to correct 1-bit errors. The function f encodes *RandKey* to a value *RandKey* = $\underline{00}1\underline{0}110$, where the underlined digits are the error correcting bits. The encoded key is then XORed with PV_s to produce γ = 0111101, which is sent over to the receiver along with the MAC as *CERT*.

—The receiver uses its version of PV (PV_r), which is a bit different from PV_s and XORs it with γ retrieved from the *CERT* to obtain value *RandKey*$'$ = $\underline{00}0\underline{0}110$.

—The *RandKey*$'$ is then passed through the error correction function (f(*RandKey*$'$)) to generate *RandKey*$''$ = $\underline{00}1\underline{0}110$, which is decoded to 1110; the original *RandKey* chosen by the sender.

3.2.2 *Secure Group Communication.* Performing secure group communication with the PVS scheme is a trivial extension of the secure unicast. For group communication, the sender simply broadcasts a *measurement-sync* message specifying the IDs of all the recipient nodes in it. As noted earlier, the system- and communication-related delays will not impact the PV measurements at the receiving nodes. The rest of the protocol is identical to the *PVS_SEND* and *PVS_RECV* described in Figure 3.

So far the PVS scheme has been described as an efficient means of securing intersensor communication in a BSN. However, this need not be the extent of its capabilities. It can essentially be used to secure communication between any set of entities which can measure a chosen PV. For example, if the base station has the capabilities to measure PVs as well, then we can use the PVS scheme to secure communication between the sensors and the base station along with BSN-wide broadcasts.

3.3 Security Analysis

In this section we analyze the security of the PVS scheme from the vulnerabilities arising from the choice of PVs as well as the ability of remote PV monitoring.

Protection from PV compromise. The security of the PVS scheme depends to a large extent upon the strength of the PV used. Even though *RandKey* is the actual key used to secure the data and may even be varied with each message, it is the PV which is in essence the symmetric key between the communicating sensors and its compromise, irrespective of the *RandKey*, opens up the network to adversaries. Therefore the choice of the PV is very important and has to be done carefully. An important question at this time is *how long can a particular PV measurement be used?* The answer obviously depends upon the strength of the PV and its exposure. As mentioned earlier, the key agreement process using PVs itself was about 90 bits in strength while the key exchanged is 128-bits long [Bao et al. 2005]. If such PVs are being used, then rekeying will be rarely needed. Irrespective of the strength of the PVs, they should be changed during network reconfiguration when nodes are added or removed from the network, to maintain backward secrecy [Schenier 1996]. Note that we do not consider the use of different *RandKey* for each communication as rekeying, since its compromise is dependent on the PV being used. The successful use of the PVS scheme between the communicating entities also enforces authentication by assuring the entities that the others belong to the same BSN. An external malicious entity cannot spoof the identity of a legitimate node or inject bogus messages into the BSN, as it cannot measure the PV and therefore cannot include a valid physiological certificate in any of its messages. An adversary can try to replay past messages (pretending to be the sender) in order to give an impression of unchanging data to the receiver. However, if more than one message is received with the same *RandKey*, the receiver will detect the attack and ignore the messages. The attacker not being aware of the PV being used cannot change the contents of the message, but can only replay them. Another advantage of using the PVS scheme is its temporal variance property: compromise of the current PV does not give any knowledge about any subsequent measurement of the PV. Therefore rekeying in the event of a PV compromise[5] is as simple as a new PV measurement.

Protection from remote PV monitoring. Until now we have assumed that malicious entities cannot measure a PV without being part of the BSN, that is, mounting physical attacks. Recent years have seen progress in radar technologies which allow a patient's heart rate to be measured from a distance within a few meters [DroitCour 2006; Greneker 1997]. These systems work by aiming a radar at the subject's thorax (chest area) and measuring the linear motion of the chest due to the beating of the heart. Using such a device, a malicious entity could potentially measure IPIs (by measuring the time difference between successive beats) from a distance leading to the active participation of a malicious entity in the BSN, without being in contact with the host. The use of a radar-based system presents many issues which makes their use for remote IPI measurement difficult due to: (1) its susceptibility to interference by objects in the environment (e.g., motion of trees and even grass), (2) its

[5] Detecting PV compromise may require the help of additional entities in the network, for example, the base station which may be monitoring network-wide operations.

requirement of directionality (i.e., it should be directed toward a person's thorax), (3) its susceptibility to the movement of the person being monitored, and (4) its requirement that the person being monitored is not in a crowd [DroitCour 2006; Greneker 1997]. In practical terms, we believe these technologies will not require us to change our threat model. This is because they have been developed for remotely measuring the heart rate of a stationary person, from a short distance, in a highly controlled environment. In DroitCour [2006] the author has presented detailed results with respect to the capability of using a radar system for measuring heart rate. The findings show that as the measurement distance increases from 0.5m to 2m, the value of the heart rate measured by a radar system varies from ±7 beats/min to ±20 beats/min from the actual value (measured using an EKG setup). Further, anyone using a radar system at distances of 2m or less would be highly likely to be noticed by the host or someone nearby. Given such a great variation at just 2m distance, we believe that any measurements made over larger distances will produce values which are likely to be even more divergent from the actual value. Further, the presence of artifacts due to motion and orientation of the person being monitored, along with the environment in which the monitoring is taking place (e.g., presence of objects, walls, and people) will also affect the efficacy of the attack. Even if such a technology were to become accurate, utilizing multiple physiological values instead of relying on just the IPI value will ensure that the effect of a PV compromise is limited.

In summary, by making it difficult for malicious nodes to be part of the BSN, using time-variant PVs as cryptographic keys and limiting their exposure, we can successfully use the PVS scheme to preserve the three essential properties of secure communication, namely, confidentiality, message integrity, and authenticity of communicating entities (sensors in the same BAN). Some of the most potent attack vectors on sensor communication such as wormholes, sinkholes, and the Sybil attack can now be prevented as they rely on the lack of the aforementioned properties to succeed [Karlof and Wagner 2003].

3.4 Performance Analysis

It can be seen that the PVS scheme performs the dual task of session key distribution and secure communication in a single step. In this section we analyze the performance of the PVS scheme by comparing it with several prominent key distribution approaches. The criteria of comparison are the design goals presented in Section 2.3. Table III shows the properties of these protocols and that of the PVS scheme.

The *Probabilistic Key Sharing (PKS)* approach predeploys hundreds of keys in each sensor node, such that any two nodes can establish a shared key with some probability [Eschenauer and Gligor 2002; Chan et al. 2003; Du et al. 2005; Liu and Ning 2003a; Pietro et al. 2003]. Whenever two nodes in each other's range do not have a common key they use other nodes in the neighborhood to establish a shared key. Key distribution in a PKS scheme is not deterministic and it only supports pair-wise key distribution (unicast). Using PKS to enable

Table III. Comparison of Key Distribution Schemes with PVS

Approaches	Computation	No. of Comm.	No. of Keys	Deterministic	Comm. Types	Usability
KDC	M+E	≥2	>1	Yes	U/G	No
MKB	M+E	≥2	>2	Yes	U/G	No
PKS	M+E	>1	10–100	No	U	No
PVS	M+X+C	0	1	Yes	U/G	Yes

(M: MAC, E: Encryption, X: XOR, C: Error Correction Function, U: Unicast, G: Group, Comm.: Communication).

secure communication requires storing a large pool of keys at each sensor node. PKS provides both encryption and authentication facilities but as they require explicit predeployment, they lack the plug-n-play character which limits their usability. This approach falls under the predeployment-based key distribution category.

Another approach is the one which uses a *Key Distribution Center (KDC)*. The KDC approach utilizes the base station to distribute keys to sensors in the network. Initially, each sensor shares a unique pair-wise shared key with the base station. The base station uses this key to securely distribute pair-wise keys between any two nodes, which want to communicate securely, in the network [Deng et al. 2003; Undercoffer et al. 2002]. The distribution of keys involves computation of MAC and encryption for securely communicating the shared keys. The scheme is deterministic (as opposed to probabilistic, where two neighbors share a symmetric key only with a certain probability) in establishing pair-wise symmetric keys between two nodes in the network for both secure unicast and group communication. Using the KDC to enable secure communication requires each node to store one key for each node it wants to communicate with, in addition to the pair-wise key with the base station. KDC provides both encryption and authentication facilities, but as they require explicit predeployment, they lack the plug-n-play character which limits their usability. It falls under the communication-based key distribution category.

The *Master-Key-Based (MKB)* approaches use a predeployed master key and some exchanged information to generate pair-wise shared keys between a node with all its neighbors [Zhu et al. 2006; Lai et al. 2002; Perrig et al. 2002]. The MKB approach has the capability to secure both unicast and group communication. It is deterministic in nature and requires computation of a MAC and data encryption for distributing keys. Using an MKB approach to enable secure communication requires each node to store one key for each of its neighbors in addition to the pair-wise key with the base station and a network-wide global key. MKB provides both encryption and authentication facilities, but as they require explicit predeployment, they lack the plug-n-play character which limits their usability. It falls under the communication-based key distribution category.

Physiological Value-based Security (PVS), on the other hand, has modest computation, communication, and storage requirements. In terms of the computational overhead it requires the computation of a MAC for generating *CERT*, computation of bit-wise XOR operations to generate γ, and a simple

ECF for correcting the variations in the measured PVs. Further, by distributing the keys during communication, the PVS scheme completely avoids additional communication overhead (unlike KDC and MKB) making it energy efficient. Additionally, with the PVS scheme each node in the network only needs to store a single PV to secure both unicast and group communication with its neighbors, which makes it highly space efficient with respect to the number of keys that need to be stored at a node for meeting its unicast and group communication requirements. It is deterministic in distributing keys between sensors and provides both encryption and authentication facilities. Additionally, the PVS scheme does not require any form of manual initialization steps; it is completely transparent and thus acquires a plug-n-play character. This efficiency, that is, support for all secure communication requirements, and usability ensures that the PVS scheme meets all our design goals.

Secure intersensor communication is a fundamental requirement for secure BSNs. It has applications in securing many BSN applications such as secure topology formation and routing. In the next section we focus on demonstrating the use of the PVS scheme for secure cluster topology formation. In Karlof and Wagner [2003], the authors demonstrated that traditional cluster formation protocols have several major security flaws if executed without considering security. We remedy this situation by suggesting a secure cluster formation protocol that utilizes the PVS scheme. We also demonstrate how PVS-based secure cluster formation is more efficient compared to alternate implementations which use traditional key distribution approaches.

4. PVS CASE STUDY: SECURE CLUSTER FORMATION

In this section we present a PVS-based secure cluster formation protocol for illustrating its utility in securing large-scale communication requirements of the BSN. The sensors in a BSN forward their data to the base station for further processing. Each sensor transmitting its data directly to the base station can be very expensive, due to the long distance communication to the base station [Heinzelmann et al. 2000]. Multihop communication can help in reducing the energy consumption (at each node) in transmitting the data to the base station [Upadhyayula and Gupta 2007]. Most efficient and commonly used topologies for sensor networks are cluster based [Heinzelmann et al. 2000]. A *cluster* is a group of colocated sensor nodes, with one node designated as a *leader*. The leader collects data from other nodes in the cluster, performs data fusion, and forwards it to the base station.

As an example of the utility of clusters in a BSN consider a set of activity monitoring sensors located on the host's legs. Without being organized into clusters or any other topology, each movement of the leg will result in all the sensors in the region trying to reach the base station with their observed data. This might potentially cause network congestion, provide the base station with redundant data, and result in each node expending large amounts of energy in trying to reach the base station. On the other hand, if the sensors were organized into one or more clusters, the cluster leaders could collect the data from their group, perform data fusion (remove redundant data), and relay only the most important characteristics of the movement to the base station. This results in only a few nodes (leaders) expending energy in reaching the base station.

However, traditional cluster formation algorithms are vulnerable to attacks such as sinkhole formation as they do not consider security [Karlof and Wagner 2003]. In this section we demonstrate the use of PVS in alleviating this problem.

4.1 Cluster Formation

In order to address the vulnerabilities of traditional cluster formation protocols, we have to first understand how they function. Traditional approaches for forming clusters take a distributed approach around a set of elected nodes called the *leader node*. Upon deployment, some of the nodes in the network individually decide to elect themselves as leader nodes for that particular round. Much work has been done in identifying optimal criteria which can be used to efficiently choose leaders around which clusters could be formed, such as Amis and Prakash [2000a, 2000b], Basagni [1999a, 1999b], Baker and Ephremides [1981], Bandyopadhyay and Coyle [2003], Chatterjee et al. [2002], Heinzelmann et al. [2000], and Tang et al. [2005b]. Here we assume the model used by the LEACH protocol for selecting the leaders [Heinzelmann et al. 2000]. Once the leaders have been chosen the clusters can be formed around them. Note, the focus of this work is solely on the cluster formation process after the leaders have been chosen.[6] Let N, L, and $M = N - L$ denote the set of all the sensors in a BSN, the set of elected leader nodes, and the set of nonleader nodes, respectively. The cluster formation process takes place in the following three steps.

—*Step 1: Broadcast Solicitation*. Each leader node $p \in L$ broadcasts a solicitation beacon which contains its ID and other control information. There are a maximum of $|L|$ solicitation beacons broadcasted in this step.

—*Step 2: Cluster Selection*. Each leader node solicitation beacon is received by a subset of nonleader nodes in the network. Each nonleader node $q \in M$, which has received at least one solicitation, decides to join the cluster of a leader $j \in L$ such that, out of all the beacons received by q, $s_j = max(s_1, s_2, \ldots, s_h)$, where s_k is the signal strength of the solicitation beacon from leader node k, and $h \leq |L|$.

—*Step 3: Transmitting Reply*. Each sensor $q \in N$ now transmits a reply message to the leader node whose cluster it decides to join.

4.2 Reclustering

As the cluster leaders perform long distance communication to the base station, they lose energy at a much higher rate than other nodes. Therefore the clusters in a network have to be reorganized from time to time, by designating specific nonleader nodes (based on protocols mentioned earlier) in the network as leaders and repeating the protocol to form clusters around them. Energy may

[6] We only consider single-level clusters in this article. Forming multilevel hierarchical clusters is a simple extension [Bandyopadhyay and Coyle 2003].

not be the only criterion for repeating cluster formation. For example, cluster formation may also be required after sensors in the network are reconfigured for improving the network performance, for example, data latency.

4.3 Security Issues with Cluster Formation

There are a few inherent problems with the aforesaid signal strength-based unsecured cluster formation protocol.

—*Implicit Trust on Leader Nodes*. In step 1, the nonleader nodes assume that only the elected leader nodes broadcast the solicitation beacon and that each elected leader node is trustworthy. Therefore when they decide to join its cluster in step 2, they do not know if they are joining the cluster of a legitimate leader node or a malicious entity posing as legitimate leader node. This can allow a malicious entity to broadcast a much stronger solicitation signal (than all legitimate leader nodes) in step 1 and fool the nonleader nodes into designating it as their leader. In Karlof and Wagner [2003] this attack is referred to as the HELLO Flood attack. A malicious entity successfully mounting this attack forms what are called *sinkholes* for all the sensors which designated it as their leader. Once a sinkhole is formed, it can easily manipulate the data passing through it. Some of the principal consequences of sinkholes include: data integrity loss and missing data packets due to selective data forwarding [Karlof and Wagner 2003]. The problem was originally presented in the context of traditional sensor networks, but the same applies for BSNs as well.

—*Implicit Trust on Nonleader Nodes*. A leader node similarly assumes that the reply it received in step 3 was from a trustworthy nonleader node thereby allowing a malicious entity to join a cluster and potentially generate bogus data.

Therefore, a secure cluster formation protocol is required that ensures the clusters formed within a BSN do not contain any sinkholes and that all nodes in all clusters are legitimate. It is worth noting that although we have used signal strength as the cluster formation criteria, the aforesaid insecurities would exist even when other cluster formation criteria are used.

4.4 Secure Cluster Formation Protocol

The main reason traditional cluster formation protocols suffer from sinkhole formation or malicious nodes joining the cluster is because of the lack of authentication in intersensor communication. This allows any arbitrary entity to pose as a legitimate node. In this section we present a secure cluster formation protocol called *Distributed Cluster Formation using Physiological Value-based Security (DCF-PVS)* which alleviates these problems. The protocol uses the signal strength as a cluster formation criteria and each sensor communication during the cluster formation is authenticated using the PVS scheme. As cluster formation is a network-wide operation, and the required PV measurement synchronization is carried out by the BS which broadcasts the *measurement-synch* message before each cluster formation.

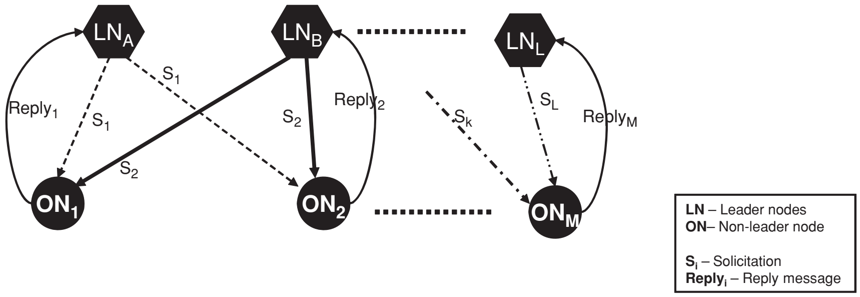

Fig. 4. Distributed cluster formation protocol.

The *DCF-PVS* protocol extends the traditional cluster formation scheme by using PVS for securing the solicitations and replies. The protocol has two steps, as illustrated in Figure 4.

—*Step 1*. The leader nodes (LNs) broadcast a solicitation, with their identity information. The solicitation includes a *CERT* to authenticate the leaders to the nonleader nodes. The messages S_1, $S_2, \ldots, S_L$ in Figure 4 are the solicitations.

—*Step 2*. The nonleader nodes (ONs) receive the solicitations and verify the *CERT*. If successful, they send a reply back to the leader node whose solicitation was received at the highest signal strength. A *CERT* is again included for authentication purposes. The leader nodes verify the *CERT* in the reply and accept the nonleader node in their cluster. The messages $Reply_1$, $Reply_2, \ldots, Reply_M$ in Figure 4 are the reply messages.

The steps of the protocol are summarized next, where p is the leader node which solicits, k is the nonleader node which receives the solicitation, and d is the chosen leader. The other symbols have their usual meanings.

$$\textit{Step 1} : p \rightarrow * :< p, CERT[p] >$$
$$\textit{Step 2} : k \rightarrow d :< k, d, CERT[d, k] >$$

Here $CERT[x] = < MAC(RandKey, x), RandKey \oplus PV_u >$, $RandKey$ is the key being established between the two sensors, and u is either the leader or nonleader node. We have prototyped the *DCF-PVS* protocol on Crossbow Mica2 motes (http://www.xbow.com) using the assumed PV values. Our purpose was to evaluate its performance solely in terms of the communication and cryptographic overhead imposed by PVS. The size of the binary files uploaded to the motes was 12.8KB for the leader node and 13.9KB for the nonleader node. It can be seen that the protocol implementations are extremely lightweight in terms of code size. As a comparison, an implementation of the Elliptic Curve Cryptography (ECC) using 168-bit key on motes takes 35KB of memory, which further increases with larger key sizes and protocol steps [Malan et al. 2004].

4.5 Security Analysis

In this section we analyze the security provided by the *DCF-PVS* scheme. As it prominently uses the PVS scheme, the associated security properties hold true here as well and we will not repeat them.

The DCF-PVS scheme does not suffer from the ailments of traditional cluster formation protocols. Attacks such as the HELLO Flood attack or the sinkhole formation are avoided. Therefore, if a malicious entity tries to broadcast solicitations at a very high signal strength (to potentially mount a HELLO Flood attack), the nonleader nodes will not designate it as their leader because the solicitation will lack a valid *CERT*, thus avoiding sinkhole formation. Attempts by malicious nodes to join a BSN by posing as a nonleader node (to join the BSN and subsequently introduce bogus messages into the network) will also fail for the same reason. If a malicious entity tries to join the network by replaying previously communicated solicitations or reply messages, the recipients will not be able to successfully unlock the *RandKey*. This is because the *CERT* in the replayed messages would be computed using a "stale" measurement of a temporally varying PV and therefore it would be rejected.

Finally, the leader selection process itself does not face any security threats. Most leader selection schemes, including the LEACH protocol, require self-nomination based on various criteria such as energy available and whether the node was a cluster leader in the previous round [Heinzelmann et al. 2000]. Once nodes have nominated themselves as cluster leaders, the cluster formation process can start and it can be secured using the PVS scheme.

4.6 Performance Analysis

We performed numerical analysis of the DCF-PVS scheme to determine the overhead it imposes in terms of energy consumption. We then compare its performance with alternate secure cluster formation protocols which use the KDC and the LEAP protocol [Zhu et al. 2006] as a basis of securing intersensor communication, instead of the PVS scheme. The reason for such a comparison is to demonstrate that using PVS as a basis for secure cluster formation is more energy efficient.

4.6.1 *Energy Model.* We assume a first-order energy model for reasoning about energy overhead imposed by the need for security. As the overhead associated with computation and sensing is much smaller compared to communication [Lo and Yang 2005], we assume their contribution to energy consumption to be negligible. The model, presented in Heinzelmann et al. [2000], assumes that the amount of energy consumed by a sensor for transmitting a k-bit message, to a maximum distance d is $k \times E_t$ Joules, where $E_t = \alpha + \beta d^p$. Here α, β, and p are constants which specify the energy dissipated by the transmitter/receiver circuitry, the energy consumed for signal amplification to maintain an acceptable Signal-to-Noise (SNR) ratio, and the path-loss coefficient ($p = 2$), respectively. Similarly, the receiver node consumes $k \times E_r$ Joules for receiving a k-bit message, where $E_r = \alpha$. Under this model, the value of α and β used are $50nJ/bit$ and $100pJ/bit/m^2$, respectively [Heinzelmann et al. 2000].

Table IV. Parameters Used in Analysis of the Proposed Distributed Cluster Formation Protocol

Node ID	16 bits
Key	128 bits
Signal Strength (SS)	16 bits
Maximum Distance Transmitted (d)	0.5 m
MAC / Nonce	128 bits
Number of Nodes (N)	10–500
% of LNs (L)	10%
Avg. % of Non-leader nodes which receive LNs message (Q)	10% of N

4.6.2 *Formulations.* Our principal aim with this analysis is not to consider absolute value of energy consumed, but to show the overall trend in energy consumption. We therefore present normalized results for all our experiments, that is, set value of one of the schemes to 1 and show how expensive or inexpensive the other schemes are in comparison. Table IV shows the value of the parameters used in the analysis. Table V shows our energy consumption formulations based on the assumed energy model. Before we present our results, we explain the table using the first row (DCF-PVS) as an example.

—The first column titled Scheme specifies the protocols being considered for example, DCF-PVS.

—The second column titled Message Content specifies the number of steps in the protocol and its content. For example the DCF-PVS scheme consists of two steps: the first step is the solicitation and requires the transmission of a node-ID and a physiological certificate, while the second is the reply message which is also identical to solicitation in terms of content.

—The third column titled Message Size specifies the size of the message being sent in bits. For example, the symbol s specifies the number of bits there are in the solicitation message which is 16+256=272 bits (from Table IV 16 bits for ID and 256 bits for *CERT*, as the length of MAC is 128 bits and that of γ is same as *RandKey*, that is, 128 bits. The number of bits for the reply is identical to s. The notation $|x|$ denotes the length of x in bits.

—The fourth column titled Network Energy Consumption Formulation specifies the formulation based on the assumed energy model, to compute the energy consumed by the entire BSN during the execution of the protocol. For DCF-PVS this can be computed in two steps.

(1) There are L solicitations (each of size s bits) broadcasted in the first step, which results in sLE_t Joules of energy consumption for the BSN, where $E_t = \alpha + \beta d^p$ from the energy model in Section 4.6.1 and $p = 2$. If each of the L solicitations is received on an average by Q nonleader nodes, the BSN energy consumption will be: $sQLE_r$ Joules, where $E_r = \alpha$ from the energy model in Section 4.6.1.

(2) The nonleader nodes depending upon the signal strength of the solicitation send a reply back with y bits resulting in the consumption of $yE_t(N - L)$ Joules for transmission within the BSN. The leader nodes

Table V. Energy Formulation for Distributed Cluster Formation (DCF) Protocols

Scheme	Message Content	Message Size	Total Energy
DCF-PVS	**Solicitation:**<ID, CERT> **Reply:**<ID, CERT>	$s = \lvert ID\rvert + \lvert Cert\rvert$ $y = s$	$T = sLE_t + sQLE_r$ $+ yE_t(N-L) + yE_r(N-L)$
NSP	**Solicitation:**<ID, CERT> **Reply:**<ID, CERT>	$t = \lvert ID\rvert + \lvert Cert\rvert$ $l = t$	$T = tLE_t + tQLE_r$ $+ lE_t(N-L) + lE_r(N-L)$
DCF-LEAP	**Key Distribution** **Discovery:**<ID,MAC> **Ack:**<ID,MAC> **Group-Key-Dist:**<Key> **DCF Execution** **Solicitation:**<ID,Nonce,MAC> **Reply:**<ID,MAC>	$f = \lvert ID\rvert + \lvert MAC\rvert$ $a = f$ $g = \lvert Key\rvert$ $n = \lvert ID\rvert + \lvert Nonce\rvert + \lvert MAC\rvert$ $l = n - \lvert Nonce\rvert$	$E_{dleap} = fLE_t + fQLE_r$ $+aQLE_t + aQLE_r$ $+gLE_t + gQLE_r$ $E_{dprot} = nLE_t + nQLE_r$ $+ lE_t(N-L) + lE_r(N-L)$ $T = E_{dleap} + E_{dprot}$
DCF-KDC	**Key Distribution** **Request:**<ID,Q(ID),MAC> **Reply:**<ID,Key> **DCF Execution** Same as DCF-LEAP	$r = \lvert ID\rvert + Q\lvert ID\rvert + \lvert MAC\rvert$ $y = \lvert ID\rvert + \lvert Key\rvert$	$T = rLE_t + y(Q+1)LE_r$ $+ E_{dprot}$
DCF-MLP	**Discovery:**<ID,MAC> **Ack:**<ID,MAC>	$d = \lvert ID\rvert + \lvert MAC\rvert$ $a = d$	$T = dLE_t + dQLE_r$ $+aE_t(N-L) + aE_r(N-L)$

(DCF-PVS: PVS-based cluster formation, NSP: Non-secure cluster formation, DCF-LEAP: LEAP protocol-based cluster formation, DCF-KDC: KDC-based cluster formation, DCF-MLP: Modified LEAP protocol-based cluster formation).

which receive these solicitations result in the consumption of $yE_r(N-L)$ Joules for the network.

The two steps of DCF-PVS therefore yield a total of $T = [sLE_t + sQLE_r] + [yE_t(N-L) + yE_r(N-L)]$ Joules during their execution, where T the energy consumed by the scheme. Similar formulations have been presented for other protocols considered in this analysis.

We now present the comparative study between the DCF-PVS scheme and implementation of distributed cluster formation protocol using various key distribution schemes and then move on to the analysis of the the DCF-PVS scheme itself.

4.6.3 *Comparison of the DCF-PVS with Alternate Versions.* The DCF-PVS scheme utilizes PVS for securing its intersensor communication requirements. If keys are explicitly distributed between the leader nodes and nonleader nodes

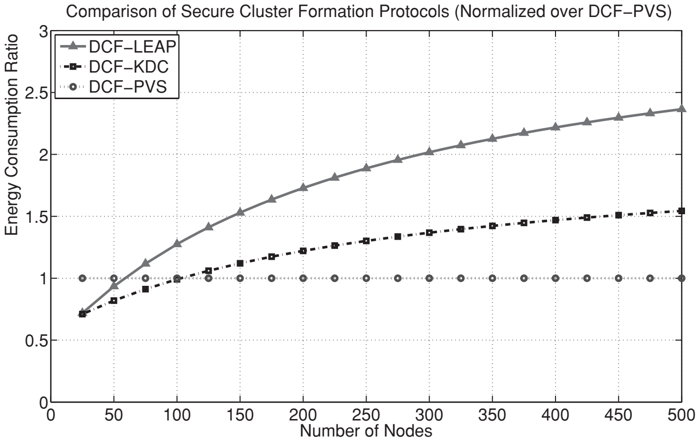

Fig. 5. Energy consumption trends for DCF-PVS with alternate versions based on KDC and LEAP with respect to network size.

in the network, then secure cluster formation can also be executed without using PVS.

First we compare the the DCF-PVS scheme with its alternate version which uses a generic KDC-based protocol and the MKB-based protocol (LEAP [Zhu et al. 2006]) for distributing keys between the leaders and nonleaders to facilitate secure cluster formation. We name the secure cluster formation protocol based on KDC *DCF-KDC* and the one utilizing LEAP as *DCF-LEAP*. We chose KDC and LEAP primarily because they have minimal predeployment requirements, similar to the PVS scheme. The parameters used in our comparison and the energy model utilized are the same as in the preceding section. Rows 3 and 4 in Table V present the formulations used for each of these protocols. To implement the distributed secure cluster formation protocol using KDC and LEAP we first use them to distribute symmetric keys between the leader and nonleader nodes in the network appropriately. Therefore for secure cluster formation to take place, key distribution has to be done between each leader and all the nonleaders in its range. Once the keys are distributed between sensors, the protocol can execute as described in Section 4, however, as keys are present between the leaders and nonleaders, a simple *MAC* is used instead of *CERT*, and a *Nonce* is used for ensuring transaction freshness.

Figure 5 shows the results of the comparison between DCF-PVS, DCF-KDC, and DCF-LEAP. The results are normalized with respect to DCF-PVS. For smaller network sizes, DCF-KDC or DCF-LEAP are less expensive, as the overhead from distributing keys is offset by the presence of the longer *CERT* in each message of DCF-PVS. For larger network sizes, however, we see that DCF-PVS is least expensive, as the communication overhead for distributing keys in DCF-KDC or DCF-LEAP increases above that of using *CERT*. The main reason for DCF-PVS being less expenxive, is because it can distribute keys during data communication, while the other protocols require key distribution followed by secure cluster formation. One cannot do secure cluster

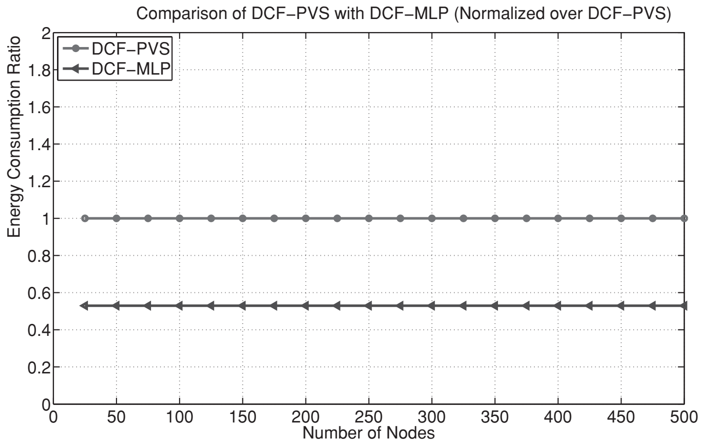

Fig. 6. Energy consumption trends for DCF-PVS with modified LEAP protocol (DCF-MLP) with respect to network size.

formation directly as the sensors do not share a key with each other to begin with. As BSNs get progressively larger the number of neighbors for the node increases and this also increases the key distribution overhead before secure cluster formation can take place. As the BSNs are getting progressively larger the energy-efficient nature of DCF-PVS and its minimal storage requirements make it highly scalable, efficient, and suitable for secure cluster formation in BSNs.

An interesting property of the LEAP protocol is that it itself can be modified to form secure clusters in a distributed manner. We call this the *Modified LEAP-based Distributed Cluster Formation Protocol* (*DCF-MLP*). DCF-MLP works exactly like LEAP except it distributes pair-wise keys only between leader and the nonleader nodes in its range, the key distribution is initiated by the leader nodes by broadcasting a discovery message, and each nonleader node shares a pair-wise key with only one leader node, whose discovery message was received at the highest signal strength compared to all the other leader nodes in its range. The last row in Table V shows the formulation for DCF-MLP. This is unlike using LEAP for key distribution for DCF-LEAP where each leader node has a pair-wise and group key with all the nonleader nodes in its range.

We compared the energy consumption of the DCF-MLP protocol with DCF-PVS and the results, normalized with respect to DCF-MLP, are given in Figure 6. We found that DCF-MLP can lead to more efficient cluster formation compared DCF-PVS. This difference is due to the larger length of the *CERT* used in distributed protocol solicitation and reply, compared to the simple MAC utilized by DCF-MLP. The *CERT* in DCF-PVS is longer because it contains the hidden key (as γ) required for the receiver to verify the message sent. DCF-MLP does not face this issue because of the predeployment of keys. However, implementing DCF-MLP requires a node to store a pair-wise key with the base station, a pair-wise key with certain neighbors (if the node is a leader then it shares one key with all the nonleaders which joined its cluster

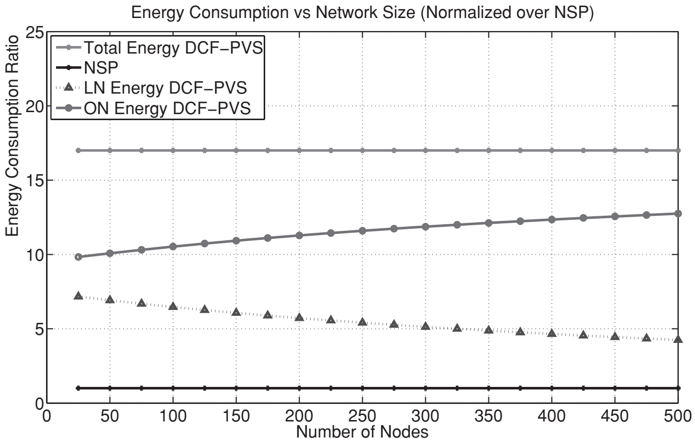

Fig. 7. Energy consumption trends for the DCF-PVS, leader nodes (LN) in DCF-PVS, nonleader nodes (ON) in DCF-PVS, and nonsecure cluster formation protocol with respect to network size.

and if the node is a nonleader it shares one key with the leader whose cluster it joins), and a global key for securing its discovery message. The PVS scheme can achieve the same result by simply storing a single PV. On the whole, there is a trade-off between energy and storage space here and the scheme which is most appropriate for a given BSN should be used.

4.6.4 *Comparing DCF-PVS with Nonsecure Cluster Formation.* We begin by comparing the energy consumed by the BSN for DCF-PVS, and nonsecure cluster formation protocol (*NSP*) for different network sizes. Figure 7 shows the plot normalized with respect to NSP. The formulations used in this section are presented in first two rows of the Table V. For DCF-PVS and NSP we assumed that each solicitation will be received by Q nonleader nodes. We varied the number of nodes in the network from 10 to 500 and as expected the DCF-PVS protocol is more expensive than NSP but at a constant level determined by the overhead imposed in communicating the *CERT* with each message. The figure also shows the breakup of energy consumed by leader nodes and nonleader nodes for DCF-PVS normalized over NSP. As the total number of nodes increases, the energy consumption of the network due to the leader nodes decreases compared to the nonleader nodes. This is because the difference between the number of nonleader and leader nodes increases in absolute terms.

5. RELATED WORK

Key distribution protocols. Many key distribution protocols for sensor networks have been proposed in the literature. These protocols can be grouped into three categories based on the type of communication they secure: pairwise key distribution for securing unicast communication, such as Chan et al. [2003], Liu and Ning [2003a, 2003b], Lai et al. [2002], Eschenauer and Gligor [2002], Perrig et al. [2002], Du et al. [2005], Zhu et al. [2006], and Pietro et al. [2003]; group-wise key distribution for securing multicast, such as

Burmester and Desmedt [1994], Carman et al. [2002], and Zhu et al. [2006]; and network-wise key distribution for secure broadcast such as Zhu et al. [2006], Perrig et al. [2002], and Slijepcevic et al. [2002]. It can be seen that most of these protocols except Zhu et al. [2006] and Perrig et al. [2002] are designed to specifically secure a single type of sensor communication (the most popular being unicast). Further, each of these protocols requires individual sensors to store a variety of keys to meet their security requirements and therefore assume that sensors have large key storage capacities. Given the limited memory space available within the sensors in a BSN, these key distribution protocols may not be practical.

Unlike the preceding schemes which assume the predeployment of keys, work has also been done on techniques which could be used for secure predeployment of keys. Some of the prominent protocols proposed in this area include: Talking to Strangers [Balfanz et al. 2002], Key Infection [Chan et al. 2004], Message-in-a-Bottle [Kuo et al. 2007], and Resurrecting Duckling [Stajano and Anderson 1999]. Here we analyze the applicability of the latter two to BSNs because Balfanz et al. [2002] assumes the usage of asymmetric key cryptography, while Chan et al. [2004] simply broadcasts the keys in clear assuming that the adversary reaches the location later.

The Message-in-a-Bottle (MIB) [Kuo et al. 2007] scheme establishes a predeployed pair-wise key between a node and the base station. The scheme requires the sensor network administrator to place a new node (to be deployed) inside a Faraday cage with a *keying* node which provides it with its key in small segments. To prevent any leak of this keying message (due to an opening in the cage), an additional node, called the *beacon* node, jams the environment outside. We can use MIB here, but we believe it is inferior to the PVS scheme in the context of BSN because: (1) It needs a Faraday cage to allow sensors to be able to share a key; (2) it requires active involvement by the host or administrator (physical action of putting the nodes in the cage), which makes it less usable if the network to be deployed is large; (3) it needs multiple entities (beacon node, keying node) to be able to distribute the shared key, all of which may not be available when the patient is not in a controlled environment; (4) the key distribution occurs between base station and a node and not between two nodes which is not sufficient for use in BSNs we envision where multihop secure communication may be required. The PVS scheme, on the other hand, does not require any extra hardware, additional entities, or host involvement to distribute keys. Secure communication can be executed in a manner that is transparent to the host.

The Resurrecting Duckling [Stajano and Anderson 1999] scheme utilizes a side-channel for key deployment. The work was done in the context of ad hoc networks in a home environment, where devices are made to share a common key by the physical act of touching one against another. Some of the primary issues here are the following.

(1) the need for a side-channel to be able to distribute keys requiring newer interfaces such as infrared and manual action such as physical touching of devices;

(2) the sensors on the body to which a node communicates might be implanted which makes physical contact impossible. The PVS scheme, on the other hand, does not require additional interfaces built into the devices nor does it require any physical involvement of host or administrator, which makes it more efficient and usable.

Interestingly, even though the PVS scheme is designed for enabling secure communication without any need for key predeployment, it can also be used as an efficient technique for key predeployment itself. Traditional key distribution protocols can then be executed using the key provided by the PVS scheme. This property could be useful if, for some reason, the PVs that the sensors measure are are particularly strong, for long-term usage.

Environmentally-coupled key generation. The idea of using PVs for securing intersensor communication was first introduced in our previous work [Cherukuri et al. 2003]. It assumed a network of implanted sensors and used PVs for securing the sensor communication. It did not provide sufficient detail on the PVs that could be used, the properties they need to posses, or how to synchronize the PV measurements. But it did address the problem of removing the slight difference in PVs measured at different points in the body using an error correction (majority decoding) approach, which was itself based on the work done in Juels and Wattenberg [1999] for correcting differences in biometrics. A preliminary version of this work was presented in Venkatasubramanian and Gupta [2006]. The paper solely focuses on the use of PVs for secure cluster formation, without providing significant details on their measurement and usage, or detailed performance analysis DCF-PVS. In Venkatasubramanian et al. [2008b], the authors demonstrate the use of Fast Fourier Transform (FFT)-based features from PPG signals for facilitating key agreement between nodes in a BSN, called Physiological Signal-based Key Agreement (PSKA). With PSKA, unlike the PVS scheme, the agreed upon key is used for future communication and the signal features are discarded. The scheme can potentially reduce the topographic specificity associated with PVs and requires lesser signal samples than IPI to function. However, it can provide only pair-wise key agreement and is much more expensive as it uses frequency domain features.

In Mayrhofer [2007] and Mayrhofer and Gellersen [2007], the authors present a Candidate Key Protocol (CKP) for generating pair-wise cryptographic keys from feature vectors generated from accelerometer data for hand-held devices. Their schemes require the involvement of the users, in terms of physically shaking the two devices and pressing a button (called "authenticate now") on the devices to synchronize the accelerometer signal measurement process and execute the protocol. Though not directly useful in BSN environments due to its requirement of shaking, the CKP's underlying technique of using frequency domain features might be useful in the domain of BSNs to reduce the topographic specificity of PVs, but increase the complexity of using it. On the other hand, applying the PVS scheme's feature generation, quantization, and error correction might not work for CKP as we suspect

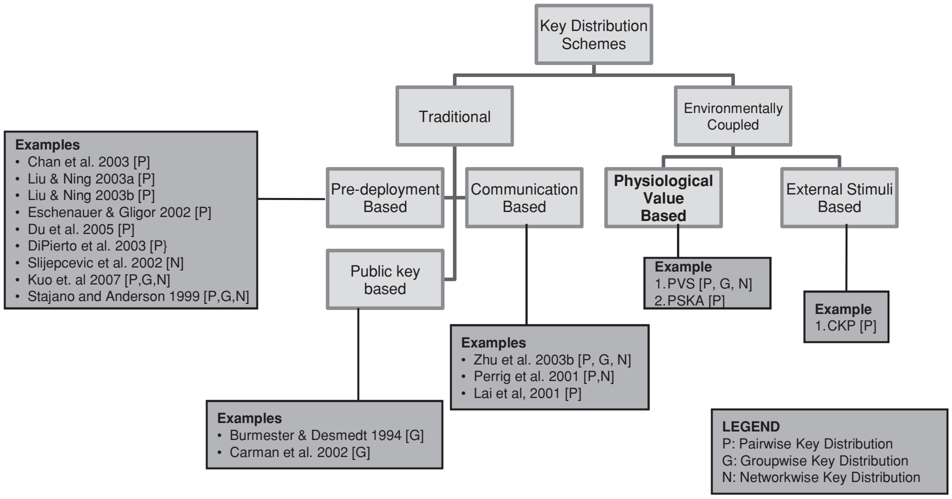

Fig. 8. Classification of traditional key distribution schemes for body sensor networks.

the errors between two quantized streams of acceleration values might be potentially large, preventing the use of simple error correction. Figure 8 presents a high-level classification of these key distribution protocols.

Cluster topology formation. With regards to cluster formation, a variety of schemes are presented in the literature. In each of these cases considerable effort is placed on the development of algorithms for optimally selecting the cluster leaders based on metrics such as: node IDs [Baker and Ephremides 1981], node mobility [Basagni 1999a, 1999b; Chatterjee et al. 2002], residual energy [Bandyopadhyay and Coyle 2003; Heinzelmann et al. 2000], load balancing [Amis and Prakash 2000a], and choosing leader nodes based on finding Minimum Dominating Set [Amis and Prakash 2000b]. However, none of these protocols was designed with security in mind and they are vulnerable to sinkhole formation. In Heinzelmann [2000], a centralized cluster formation protocol which uses the base station to decide the cluster for each node has been described. As it does not consider security, it is susceptible to sinkhole formation as well. The signal-strength-based protocols described earlier are called *leader-first* cluster formation protocols. Apart from the leader-first approach solutions have been proposed for a *cluster-first* approach to cluster formation [Krishna et al. 1997; Xu et al. 2001; Lin and Gerla 1997]. They view the cluster formation problem as a clique formation problem and try to form groups of nodes in a network without any leader node. The cluster-first approaches proposed have been for ad hoc networks or remote sensor networks with little infrastructure support. In the case of BSNs we argue that such approaches are complex given the limited capabilities of sensors and the presence of a powerful base station to control the whole network. We therefore do not consider them in this work. A secure cluster-first approach to cluster topology formation has also been proposed in Sun et al. [2006], however, it makes use of μTelsa which requires predeployment, along with public key cryptography, to achieve its goal of securely forming cliques within the network, making it inefficient for BSNs.

Near-body wireless communication. Finally, as the sensors in a BSN are placed very close to the body or even implanted, the wireless signal might use the human body to communicate. This has the potential for improving the security of the wireless channel, which is the primary cause for many of the issues discussed in this work. Therefore, for sensors to be used in the BSN many physical communication and manufacturing issues have to be considered. Recent years have seen considerable work in this domain such as Prakash et al. [2003] in modeling the effects of signal propagation through the body, Prakash and Gupta [2003] and Tang et al. [2005b] in developing energy-efficient coding and modulation techniques, and Tang et al. [2005a] in minimizing heat dissipation. Such models and schemes have to be considered in developing the sensors so as to achieve efficient intrabody wireless communication.

6. CONCLUSIONS

With a rapidly aging population the next few decades will require a proportional increase in the number of caregivers to sustain today's standards of care without increasing medical errors and health-care costs [McGinnis and Moore 2006]. BSNs can play a major role in addressing this issue by facilitating long-term, continuous, and real-time monitoring of health information. BSNs are part of a new class of environment-coupled systems called *Cyber-Physical Systems* which can not only monitor their environment (human body) but can also induce/actuate change in it (delivery drugs) if need be [Venkatasubramanian et al. 2009]. In this article we presented a novel, usable, and efficient scheme for securing intersensor communication in BSNs utilizing its environment-coupled nature (i.e., physiological values from the host's body), called *Physiological Value-based Security* (PVS). We used the PVS scheme in a protocol for secure cluster topology formation in a BSN and analyzed its security and performance. Performance analysis of the PVS-based secure cluster formation protocol and their alternate versions (based on traditional key distribution approaches) showed that the former is more energy efficient for larger network sizes. In the future, we will work toward identifying other physiological signals apart from IPI as PVs. This is essential because not all sensors can be expected to measure the same PVs and the latency associated with using the IPI is about 30 seconds, which is considerable in BSN systems. Additionally, we plan to investigate techniques for eliminating the effects of topographic specificity, which has the potential to eliminate key distribution completely. Other practical issues, such as dealing with the various artifacts in the measurement of physiological signals, will be studied to make the PVS scheme more viable in real-life settings.

ACKNOWLEDGMENTS

We gratefully acknowledge the intellectual contributions of Ayan Banerjee, Guofeng Deng, Gianni Giorgetti, and Tridib Mukherjee of the IMPACT Lab, Arizona State University. We would like to thank Georgios Varsamopoulos of the IMPACT Lab, Arizona State University for proof-reading the article. We would also like to thank the Area Editor Adrian Perrig and the anonymous reviewers who helped improve this article immensely.

REFERENCES

ADELSTEIN, F., GUPTA, S. K. S., RICHARD, G. G., AND SCHWIEBERT, L. 2005. *Fundamentals of Mobile and Pervasive Computing*. McGraw Hill.

AMIS, A. D. AND PRAKASH, R. 2000a. Load balancing clusters in wireless ad hoc networks. In *Proceedings of the 3rd Symposium on Application-Specific Systems and Software Engineering Technology*. IEEE, 25–32.

AMIS, A. D. AND PRAKASH, R. 2000b. Max-Min D-cluster formation in wireless ad hoc networks. In *Proceedings of 19th Conference on Computer Communications*. IEEE, 32–41.

BAKER, D. J. AND EPHREMIDES, A. 1981. The architectural organization of a mobile radio via a distributed algorithm. *IEEE Trans. Comm. 29*, 11, 1694–1701.

BALFANZ, D., SMETTERS, D., STEWART, P., AND WONG, H. C. 2002. Talking to strangers: Authentication in ad-hoc wireless networks. In *Proceedings of the Symposium on Network and Distributed Systems Security*. Internet Society.

BANDYOPADHYAY, S. AND COYLE, E. J. 2003. An energy efficient hierarchical clustering algorithm for wireless sensor networks. In *Proceedings of the 22nd Conference on Computer Communications*. IEEE, 1713–1723.

BAO, S.-D. AND ZHANG, Y. T. 2005. A new symmetric cryptosystem of body area sensor networks for telemedicine. In *Proceedings of the 6th Asian-Pacific Conference on Medical and Biological Engineering*.

BAO, S.-D., ZHANG, Y. T., AND ZHANG, Y.-T. 2005. Physiological signal based entity authentication for body area sensor networks and mobile healthcare systems. In *Proceedings of the 27th Conference on Engineering in Medicine and Biology*. IEEE, 2455–2458.

BASAGNI, S. 1999a. Distributed and mobility-adaptive clustering for multimedia support in multi-hop wireless networks. In *Proceedings of the 50th International Vehicular Technology Conference*. Vol. 2. IEEE, 889–893.

BASAGNI, S. 1999b. Distributed clustering for ad-hoc networks. In *Proceedings of the International Symposium on Parallel Architectures Algorithms and Networks*. 310–315.

BURMESTER, M. AND DESMEDT, Y. 1994. A secure and efficient conference key distribution system. In *Proceedings of Workshop on the Theory and Applications of Cryptographic Techniques*. Eurocrypt, 275–286.

CARMAN, D., MATT, B., AND CIRINCIONE, G. 2002. Energy-Efficient and low-latency key management for sensor networks. In *Proceedings of 23rd Army Science Conference*.

CHAN, H., ANDERSON, R., AND PERRIG, A. 2004. Key infection: Smart trust for smart dust. *Proceedings of the International Conference on Network Protocols*. IEEE, 206–215.

CHAN, H., PERRIG, A., AND SONG, D. 2003. Random key predistribution schemes for sensor network. In *Proceedings of the Symposium of Research in Security and Privacy*. IEEE, 197–213.

CHATTERJEE, M., DAS, S. K., AND TURGUT, D. 2002. WCA: A weighted clustering algorithm for mobile ad hoc networks. *J. Cluster Comput. Special Issue on Mobile Ad hoc Netw. 5*, 193–204.

CHERUKURI, S., VENKATASUBRAMANIAN, K., AND GUPTA, S. K. S. 2003. BioSec: A biometric based approach for securing communication in wireless networks of biosensors implanted in the human body. In *Proceedings of the 1st Workshop Wireless Security and Privacy*. ACM, 432–439.

CHOI, S., SONG, S.-J., SOHN, K., KIM, H., KIM, J., YOO, J., AND YOO, H.-J. 2006. A low-power star-topology body area network controller for periodic data monitoring around and inside the human body. In *Proceedings of the 10th Annual International Symposium for Wearable Computing*. IEEE, 139–140.

DENG, J., HAN, R., AND MISHRA, S. 2003. A performance evaluation of intrusion-tolerant routing in wireless sensor networks. In *Proceedings of 2nd International Workshop on Information Processing in Sensor Networks*. IEEE, 349–364.

DroitCour, A. 2006. Non-Contact measurement of heart and respiration rates with a single-chip microwave doppler radar. Ph.D. thesis, Stanford University.

Du, W., Deng, J., Han, Y. S., Varshney, P. K., Katz, J., and Khalili, A. 2005. A pairwise key predistribution scheme for wireless sensor networks. *ACM Trans. Inf. Syst. Secur. 8*, 2, 228–258.

Elson, J., Girod, L., and Estrin, D. 2002. Fine-Grained network time synchronization using reference broadcasts. In *Proceedings of the 5th Symposium on Operating Systems Design and Implementation*. Vol. 36. ACM, 147–163.

Eschenauer, L. and Gligor, V. D. 2002. A key-management scheme for distributed sensor networks. In *Proceedings of the 9th Conference on Computer and Communications Security*. ACM, 41–47.

Greneker, E. F. 1997. Radar sensing of heartbeat and respiration at a distance with applications of the technology. *IEE Radar 449*, 150–154.

Halperin, D., Heydt-Benjamin, T., Ransford, B., Clark, S., Defend, B., Morgan, W., Fu, K., Kohno, T., and Maisel, W. 2008. Pacemakers and implantable cardiac defibrillators: Software radio attacks and zero-power defenses. In *Proceedings of the Symposium on Security and Privacy*. IEEE, 129–142.

Heinzelmann, W. R. 2000. Application specific protocol architectures for wireless networks. Ph.D. thesis, MIT.

Heinzelmann, W. R., Chandrakasan, A., and Balakrishnan, H. 2000. Energy-Efficient communication protocols for wireless microsensor networks. In *Proceedings of the Hawaii International Conference on System Sciences*. IEEE, 8020–8030.

Huang, Q., Cukier, J., Kobayashi, H., Liu, B., and Zhang, J. 2003. Fast authenticated key establishment protocols for self-organizing sensor networks. In *Proceedings of the 2nd International Conference on Wireless Sensor Networks and Applications*. ACM, 141–150.

Juels, A. and Wattenberg, M. 1999. A fuzzy commitment scheme. In *Proceedings of the 9th Conference on Computer and Communications Security*. ACM, 28–36.

Karlof, C. and Wagner, D. 2003. Secure routing in wireless sensor neworks: Attacks and countermeasures. In *Proceedings of 38th International Conference on Communication*. IEEE, 113–127.

Kern, N. and Schiele, B. 2003. Multi-Sensor activity context detection for wearable computing. In *Proceedings of the European Symposium on Ambient Intelligence*.

Krishna, P., Vaidya, N. H., Chatterjee, M., and Pradhan, D. K. 1997. A cluster-based approach for routing in dynamic networks. *SIGCOMM - Comput. Commun. Rev. 27*, 2, 49–64.

Kuo, C., Luk, M., Negi, R., and Perrig, A. 2007. Message-In-a-Bottle: User-Friendly and secure key deployment for sensor nodes. In *Proceedings of the 5th Conference on Embedded Networked Sensor System*. ACM.

Laerhoven, K. V. and Gellersen, H. W. 2004. Spine versus porcupine: A study in distributed wearable activity recognition. In *Proceeding of the 8th International Symposium on Wearable Computers*. 142–149.

Laerhoven, K. V., Villar, N., and Gellersen., H. W. 2003. A layered approach to wearable textile networks. In *Proceedings of the Eurowearable Conference*. IEE, 62–67.

Lai, B., Kim, S., and Verbauwhede, I. 2002. Scalable session key construction protocol for wireless sensor networks. In *Proceedings of the Workshop on Large Scale RealTime and Embedded Systems*. IEEE.

Lin, C. R. and Gerla, M. 1997. Adaptive clustering for mobile wireless networks. *J. Select. Areas Commun. 15*, 7, 1265–1275.

Liu, D. and Ning, P. 2003a. Establishing pairwise keys in distributed sensor networks. In *Proceedings of the 10th Conference on Computer and Communications Security*. ACM, 52–61.

Liu, D. and Ning, P. 2003b. Location-Based pairwise key establishment for static sensor networks. In *Proceedings of the 1st Workshop on Ad-hoc and Sensor Networks*. ACM, 72–82.

Lo, B. and Yang, G.-Z. 2005. Key technical challenges and current implementations of body sensor networks. In *Proceedings of the International Workshop on Wearable and Implantable Body Sensor Networks*.

Lukowicz, P., Anliker, U., Ward, J., Trster, G., Hirt, E., and Neufelt, C. 2002. Amon: A wearable medical computer for high risk patients. In *Proceedings of the 6th International Symposium on Wearable Computers*. IEEE, 133–134.

MALAN, D. J., WELSH, M., AND SMITH, M. D. 2004. A public-key infrastructure for key distribution in TinyOS based on elliptic curve cryptography. In *Proceedings of the 2nd International Conference on Sensor and Ad Hoc Communications and Networks*. IEEE, 71–80.

MAYRHOFER, R. 2007. The candidate key protocol for generating secret shared keys from similar sensor data streams. Lecture Notes in Computer Science, vol. 4572. Springer, 1–15.

MAYRHOFER, R. AND GELLERSEN, H. 2007. Shake well before use: Authentication based on accelerometer data. Lecture Notes in Computer Science, vol. 4880. Springer, 144–161.

MCGINNIS, S. L. AND MOORE, J. 2006. The impact of the aging population on the health workforce in the United States: Summary of key findings. *Cahier de Sociologie et de Démographie Médicale 46*, 2, 193–220.

MCWILLIAMS, C. 2003. The biophysical properties of the transdermal measurement. International Society of Electrodermologists. http://www.electrodermology.com/biophysprop.htm.

OUCHI, K., SUZUKI, T., AND DOI, M. 2002. Lifeminder: A wearable healthcare support system using user's context. In *Proceedings of the 22nd International Conference on Distributed Computing Systems Workshops*. IEEE, 791–792.

PARADISO, R., LORIGA, G., AND TACCINI, N. 2005. A wearable health care system based on knitted integrated sensors. *IEEE Trans. Inform. Technol. Biomed. 9*, 3, 337–344.

PERRIG, A., SZEWCZYK, R., WEN, V., CULLER, D., AND TYGAR, D. 2002. SPINS: Security protocol for sensor networks. *Wirel. Netw. 8*, 5, 521–534.

PIETRO, R. D., MANCINI, L. V., AND MEI, A. 2003. Random key assignment for secure wireless sensor networks. In *Proceedings of the Workshop on Security of Ad hoc and Sensor Networks*. ACM, 62–71.

POON, C. C. Y., ZHANG, Y.-T., AND BAO, S.-D. 2006. A novel biometrics method to secure wireless body area sensor networks for telemedicine and m-health. *IEEE Comm. Magaz. 44*, 4, 73–81.

PRAKASH, Y. AND GUPTA, S. K. S. 2003. Energy efficient source coding and modulation for wireless applications. In *Proceedings of the Wireless Communications and Networking Conference*. IEEE, 212–217.

PRAKASH, Y., LALWANI, S., GUPTA, S. K. S., ELSHARAWY, E., AND SCHWIEBERT, L. 2003. Towards a propagation model for wireless biomedical applications. In *Proceedings of the International Conference on Communications*. IEEE, 1993–1997.

SCHENIER, B. 1996. *Applied Crytpography*, 2nd ed. John Wiley and Sons.

SCHWIEBERT, L., GUPTA, S. K. S., AND WEINMANN, J. 2001. Research challenges in wireless networks of biomedical sensors. In *Proceedings of the 7th International Conference on Mobile Computing and Networking*. ACM/IEEE, 151–165.

SHANKAR, V., NATARAJAN, A., GUPTA, S. K. S., AND SCHWIEBERT, L. 2001. Energy-Efficient protocols for wireless communication in biosensor networks. In *Proceedings of the 12th International Symposium on Personal, Indoor and Mobile Radio Communications*. IEEE, 114–118.

SLIJEPCEVIC, S., POTKONJAK, M., TSIATSIS, V., ZIMBECK, S., AND SRIVASTAVA, M. B. 2002. On communication security in wireless ad-hoc sensor networks. In *Proceedings of the 11th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*. IEEE, 139–144.

STAJANO, F. AND ANDERSON, R. 1999. The resurrecting duckling: Security issues for ad-hoc wireless networks. In *Proceedings of the 7th International Workshop on Security Protocols*. Springer Verlag, 172–194.

SUN, K., PENG, P., AND NING, P. 2006. Secure distributed cluster formation in wireless sensor networks. In *Proceedings of the 22nd Annual Computer Security Applications Conference*. 131–140.

TANG, Q., GUPTA, S. K. S., AND SCHWIEBERT, L. 2005a. BER performance analysis of an on-off keying based minimum energy coding for energy constrained wireless sensor application. In *Proceedings of the International Conference on Communications*. IEEE, 2734–2738.

TANG, Q., TUMMALA, N., GUPTA, S. K. S., AND SCHWIEBERT, L. 2005b. Communication scheduling to minimize thermal effects of implanted biosensor networks in homogeneous tissue. *IEEE Trans. Biomed. Engin. 52*, 7, 1285–1294.

UNDERCOFFER, J., AVANCHA, S., JOSHI, A., AND PINKSTON, J. 2002. Security for sensor networks. In *Proceedings of the CADIP Research Symposium*.

Upadhyayula, S. and Gupta, S. K. S. 2007. Spanning tree based algorithms for low latency and energy efficient data aggregation enhanced convergecast (DAC) in wireless sensor networks. *Ad Hoc Netw. 5*, 5, 626–648.

Venkatasubramanian, K., Banerjee, A., and Gupta, S. K. S. 2008a. EKG-Based key agreement in body sensor networks. In *Proceedings of the 2nd Workshop on Mission Critical Networks*. IEEE, 1–6.

Venkatasubramanian, K., Banerjee, A., and Gupta, S. K. S. 2008b. Plethysmogram-Based secure inter-sensor communication in body area networks. In *Proceedings of the Military Communications Conference*. IEEE, 1–7.

Venkatasubramanian, K., Banerjee, A., and Gupta, S. K. S. 2009. Green and sustainable cyber physical security solutions for body area networks. In *Proceedings of the International Workshop on Body Sensor Networks*.

Venkatasubramanian, K. and Gupta, S. K. S. 2006. Security for pervasive health monitoring sensor applications. In *Proceedings of the 4th International Conference on Intelligent Sensing and Information Processing*. 197–202.

West, B. J. 2006. *Where Medicine Went Wrong: Rediscovering the Path to Complexity*. Studies of Non Linear Phenomena in Life Sciences, vol. 11. World Scientific.

Wood, A. D. and Stankovic, J. A. 2002. Denial of service in sensor networks. *IEEE Comput. 35*, 10, 54–62.

Xu, S., Heidemann, J., and Estrin, D. 2001. Geography-Informed energy conservation for ad-hoc routing. In *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*. ACM, 70–84.

Zhu, S., Setia, S., and Jajodia, S. 2006. LEAP+: Efficient security mechanisms for large-scale distributed sensor networks. *ACM Trans. Sens. Netw. 2*, 4, 500–528.

Ziaie, B. and Najafi, K. 2001. An implantable microsystem for tonometric blood pressure measurement. *Biomed. Microdevices 3*, 8, 285–292.

Received May 2007; revised September 2009; accepted October 2009