IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 9, Issue 3 (Mar. - Apr. 2013), PP 34-43
www.iosrjournals.org

Virtual Energy Based Encryption & Keying on Wireless Sensor Network

Aditya Shukla[1], Anurag Pandey[2], Saurabh Srivastava[3]
[1,2,3](Dept of C.S, Institute of Tech. & Mgmt / G.B.T.U, India)

Abstract : *Since the secure communication for Wireless Sensor Network (WSNs) is a challenging problem because sensors are resources limited and cost is the most dominant factor in a energy consumption, for this we introduce an energy-efficient Virtual Energy-Based Encryption and Keying(VEBEK) scheme for WSNs that reduces the number of transmission needed for rekeying the packets.*

VEBEK is a secure communication framework where sensed data is encoded by a RC4 encryption mechanism based on a permutation code generator. In the RC4 encryption mechanism keys changes dynamically as a function of the residual virtual energy of the sensor. Thus, one-time dynamic key is employed for one packet only and different keys for different packets. VEBEK unbundles key generation from other security services, namely authentication, integrity, non-repudiation. VEBEK is able to efficiently detect & filter false data injected by malicious outsiders. The VEBEK framework consists of two operational modes (VEBEK-1 and VEBEK-2).

Our results show that VEBEK, without incurring transmission overhead is able to eliminate malicious data from the network in an energy efficient manner.

Keywords- *Authentication, Integrity, Non-Repudiation, RC4, Rekeying, VEBEK, WSNs.*

I. Introduction

Recent advances in micro-electro-mechanical systems (MEMS) technology, wireless communications, and digital electronics have enabled the development of low-cost, low-power, multifunctional sensor nodes that are small in size and communicate unmetered in short distances. These tiny sensor nodes, which consist of sensing, data processing, and communicating components, leverage the idea of sensor networks based on collaborative effort of a large number of nodes. Sensor networks represent a significant improvement over traditional sensors, which are deployed in the following two ways:

- Sensors can be positioned far from the actual phenomenon, i.e., something known by sense perception. In this approach, large sensors that use some complex techniques to distinguish the targets from environmental noise are required.
- Several sensors that perform only sensing can be deployed. The positions of the sensors and communications topology are carefully engineered. They transmit time series of the sensed phenomenon to the central nodes where computations are performed and data are fused. But sensor networks also introduce severe resource constraints due to their lack of data storage and power.

Both of these represent major obstacles to the implementation of traditional computer security techniques in a wireless sensor network. The unreliable communication channel and unattended operation make the security defences even harder. Indeed, as pointed out in, wireless sensors often have the processing characteristics of machines that are decades old (or longer), and the industrial trend is to reduce the cost of wireless sensors while maintaining similar computing power. With that in mind, many researchers have begun to address the challenges of maximizing the processing capabilities and energy reserves of wireless sensor nodes while also securing them against attackers. All aspects of the wireless sensor network are being examined including secure and efficient routing, data aggregation, group formation, and so on.

In addition to those traditional security issues, we observe that many general-purpose sensor network techniques (particularly the early research) assumed that all nodes are cooperative and trustworthy. Researchers therefore began focusing on building a sensor trust model to solve the problems beyond the capability of cryptographic security. The development of wireless sensor network was originally motivated by military applications like battlefield surveillance. However, WSNs are now used in many civilian application areas including the environment and habitat monitoring due to various limitations arising from their inexpensive nature, limited size, weight and ad hoc method of deployment; each sensor has limited energy. Moreover, it could be inconvenient to recharge the battery, because nodes may be deployed in a hostile or impractical environment. At the network layer, the intention is to find ways for energy efficient route setup and reliable relaying of data from the sensor nodes to the sink, in order to maximize the lifetime of the network. The major differences between the wireless sensor network and the traditional wireless network sensors are very sensitive to energy consumption.

1.1 Wireless Sensor Network

A wireless sensor network is, roughly speaking, a group of highly-constrained hardware platforms called sensor nodes that collaborate towards a set of common goals. More specially, those goals are monitoring (continuously monitor certain features of their surroundings), alerting (check whether certain physical circumstances are occurring, triggering an alarm), and provisioning of information on-demand (answer to a certain query regarding the properties of the environment or the network itself). Most of the functionality of a sensor network is data-driven, although it is also possible to use it as a distributed computing platform under special circumstances.

All the functionality of the sensor network is provided thanks to the individual capabilities of the sensor nodes. A single sensor node has built-in sensors, limited computational capabilities, and communicates through a wireless channel. Therefore, they are able to get the physical information of their surroundings, process that raw information, and communicate with other nodes in its neighbourhood. Nodes are also small in size, and can unobtrusively provide the physical information of any entity. Moreover, nodes are battery-powered, thus they can act independently and operate autonomously if required.

WSN technology will be used in a variety of application areas such as environmental, military, and commercial enterprises. For example, sensor nodes forming a network under water could be used for oceanographic data collection, pollution monitoring, assisted navigation, military surveillance, and mine reconnaissance operations.

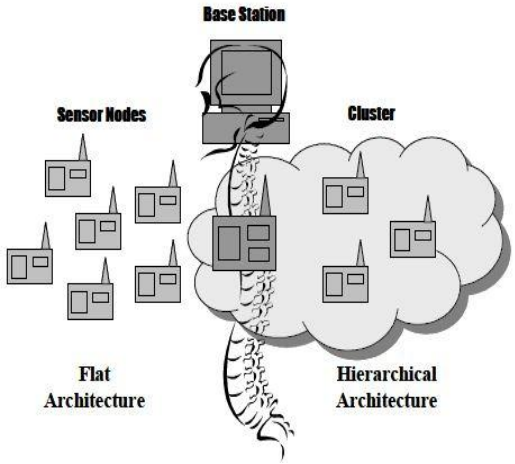

Fig 1: Overview of the Architecture of WSN

1.1.1 Routing Protocols in WSN

Routing in wireless sensor networks differs from conventional routing in fixed networks in various ways. There is no infrastructure, wireless links are unreliable, sensor nodes may fail, and routing protocols have to meet strict energy saving requirements. Many routing algorithms were developed for wireless networks in general. All major routing protocols proposed for WSNs may be divided into seven categories as shown in Table 1. We review sample routing protocols in each of the categories in preceding sub-sections.

Table 1: Routing Protocols for WSNs

Representative Protocols	Category
1) Location based Protocols	MECN, SMECN, GAF, GEAR, Span, TBF, BVGF, GeRaF
2) Data-centric Protocols	SPIN, Directed Diffusion, Rumor Routing, COUGAR, ACQUIRE, EAD, Information Directed Routing, Gradient-Based Routing, Energy-aware Routing, Information Directed Routing, Quorum Based Information Dissemination, Home Agent Based Information Dissemination
3) Hierarchical Protocols	LEACH, PEGASIS, HEED, TEEN, APTEEN
4) Mobility-based Protocols	SEAD, TTDD, Joint Mobility and Routing, Data MULES, Dynamic Proxy Tree-Base Data Dissemination
5) Multipath-based Protocols	Sensor-Disjoint Multipath, Braided Multipath, N-to-1
6) Heterogeneity-based Protocols	IDSQ, CADR, CHR
7) QoS-based protocols	SAR, SPEED, Energy-aware routing

1.2 VEBEK

VEBEK dynamically updates keys without exchanging messages for key renewals and embeds integrity into packets as opposed to enlarging the packet by appending message authentication codes (MACs). Specifically, each sensed data is protected using a simple encoding scheme based on a permutation code generated with the RC4 encryption scheme and sent toward the sink. The key to the encryption scheme dynamically changes as a function of the residual virtual energy of the sensor, thus requiring no need for rekeying. Therefore, a one-time dynamic key is used for one message generated by the source sensor and different keys are used for the successive packets of the stream. The nodes forwarding the data along the path to the sink are able to verify the authenticity and integrity of the data and to provide non-repudiation. The protocol is able to continue its operations under dire communication cases as it may be operating in a high-error-prone deployment area like under water. VEBEK unbundles key generation from other security services, namely authentication, integrity, and non-repudiation; thus, its flexible modular architecture allows for adoption of other encryption mechanisms if desired.

The contributions of this paper are as follows:

(1) A dynamic en-route filtering mechanism without that does not exchange explicit control messages for rekeying.

(2) Provision of one time keys for each packet transmitted to avoid stale keys

(3) A modular and flexible security architecture with a simple technique for ensuring authenticity, integrity and non-repudiation of data without enlarging packets with MACs

(4) A robust secure communication framework that is operational in dire communication situations and over unreliable MACs.

In comparison with other key management schemes, VEBEK has the following benefits:

- It does not exchange control messages for key renewals and is therefore able to save more energy and is less chatty.
- It uses one key per message so successive packets of the stream use different keys—making VEBEK more resilient to certain attacks (e.g., replay attacks, brute-force attacks, and masquerade attacks),
- It unbundles key generation from security services, providing a flexible modular architecture that allows for an easy adoption of different key-based encryption or hashing schemes.

II. Middleware

Middleware refers to software and tools that can help hide the complexity and heterogeneity of the underlying hardware and network platforms, ease the management of system resources, and increase the predictability of application executions. WSN middleware is a kind of middleware providing the desired services for sensing-based pervasive computing applications that make use of a wireless sensor network and the related embedded operating system of the sensor nodes.

The motivation behind the research on WSN middleware derives from the gap between the high-level requirements from pervasive computing applications and the complexity of the operations in the underlying WSNs.

The application requirements include high edibility, re-usability, and reliability. The complexity of the operations with a WSN is characterized by constrained resources, dynamic network topology, and low level embedded OS APIs. WSN middleware provides a potential solution to bridging the gap and removing the impediments. WSN middleware to help identify the key services, challenging issues, and important techniques.

Compared with the existing surveys, this paper makes the following distinct contributions. First, it proposes a reference model for analyzing the functionalities and key services of WSN-middleware. Second, it provides a detailed review of the existing work on the most important aspects in developing WSN middleware, covering the major approaches to and corresponding techniques of implementing the services. Third, the paper proposes a feature tree-based taxonomy that organizes WSN-middleware features and their relationships into a framework to help understand and classify the existing work

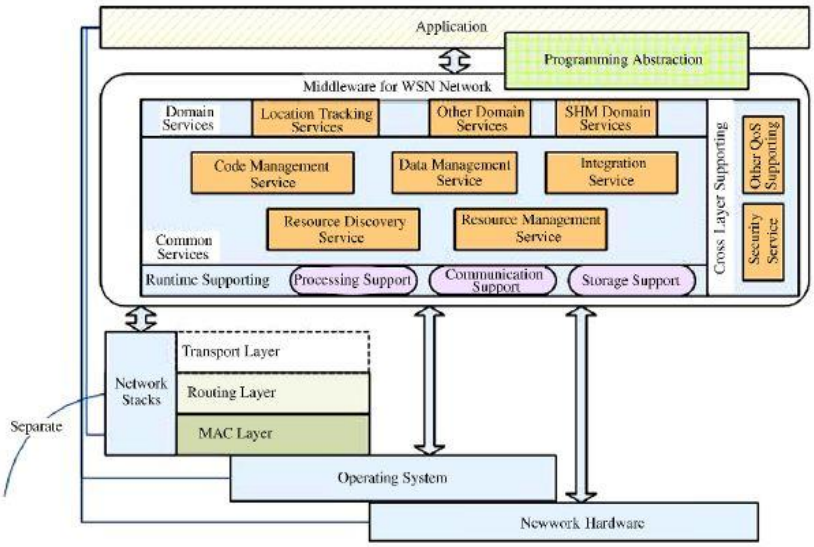

Fig 2. Reference Model of WSN Middleware

III. Operational Module

The protocol provides three security services: Authentication, integrity, and non-repudiation. The fundamental notion behind providing these services is the watching mechanism described before. The watching mechanism requires nodes to store one or more records (i.e., current virtual energy level, virtual bridge energy values, and Node-Id) to be able to compute the dynamic keys used by the source sensor nodes, to decode packets, and to catch erroneous packets either due to communication problems or potential attacks. However, there are costs (communication, computation, and storage) associated with providing these services. In reality, applications may have different security requirements. For instance, the security need of a military WSN application (e.g., surveiling a portion of a combat zone) may be higher than that of a civilian application (e.g., collecting temperature data from a national park). The VEBEK framework also considers this need for flexibility and thus, supports two operational modes: VEBEK-I and VEBEK-II. The operational mode of VEBEK determines the number of nodes a particular sensor node must watch.

Depending on the vigilance required inside the network, either of the operational modes can be configured for WSN applications. Different modes and the range of associated costs of each mode are given in . The details of both operational modes are given below.

3.1 VEBEK-I

In the VEBEK-I operational mode, all nodes watch their neighbours; whenever a packet is received from a neighbour sensor node, it is decoded and its authenticity and integrity are verified. Only legitimate packets are forwarded toward the sink. In this mode, we assume there exists a short window of time at initial deployment that an adversary is not able to compromise the network, because it takes time for an attacker to capture a node or get keys. During this period, route initialization information may be used by each node to decide which node to watch and a record r is stored for each of its 1-hop neighbours in its watch-list. To obtain a neighbour's initial energy value, a network-wise master key can be used to transmit this value during this period similar to the shared-key discovery phase of other dynamic key management schemes. Alternatively, sensors can be pre-loaded with the initial energy value. When an event occurs and a report is generated, it is encoded as a function of a dynamic key based on the virtual energy of the originating node, and transmitted. When the packet arrives at the next-hop node, the forwarding node extracts the key of the sending node (this could be the originating node or another forwarding node) from its record (the virtual perceived energy value associated with the sending node and decodes the packet). After the packet is decoded successfully, the plaintext ID is compared with the decoded ID. In this process, if the forwarding node is not able to extract the key successfully, it will decrement the predefined virtual energy value from the current perceived energy and tries another key before classifying the packet as malicious (because packet drops may have occurred due to communication errors). This process is repeated several times; however, the total number of trials that are needed to classify a packet as malicious is actually governed by the value of Virtual Key Search Threshold. If the packet is authentic, and this hop is not the final hop, the packet is re-encoded by the forwarding node with its own key derived from its current virtual bridge energy level. If the packet is illegitimate, the packet is discarded. This process continues until the packet reaches the sink. Accordingly, illegitimate traffic is filtered before it enters the network. Re-

encoding at every hop refreshes the strength of the encoding. Recall that the general packet structure is [ID, {ID, type, data} k]. To accommodate this scheme, the ID will always be the ID of the current node and the key is derived from the current node's local virtual bridge energy value. If the location of the originating node that generated the report is desired, the packet structure can be modified to retain the ID of the originating node and the ID of the forwarding node. VEBEK-I reduces the transmission overhead as it will be able to catch malicious packets in the next hop, but increases processing overhead because of the decode/encode that occurs at each hop.

3.2 VEBEK-II

In the VEBEK-II operational mode, nodes in the network are configured to only watch some of the nodes in the network. Each node randomly picks r nodes to monitor and stores the corresponding state before deployment. As a packet leaves the source node (originating node or forwarding node) it passes through node(s) that watch it probabilistically. Thus, VEBEK-II is a statistical filtering approach like SEF and DEF. If the current node is not watching the node that generated the packet, the packet is forwarded. If the node that generated the packet is being watched by the current node, the packet is decoded and the plaintext ID is compared with the decoded ID. Similar to VEBEK-I, if the watcher-forwarder node cannot find the key successfully, it will try as many keys as the value of Virtual Key Search Threshold before actually classifying the packet as malicious. If the packet is authentic, and this hop is not the final destination, the original packet is forwarded unless the node is currently bridging the network. In the bridging case, the original packet is re-encoded with the virtual bridge energy and forwarded. Since this node is bridging the network, both virtual and perceived energy values are decremented accordingly. If the packet is illegitimate, which is classified as such after exhausting all the virtual perceived energy values within the Virtual Key Search Threshold window, the packet is discarded. This process continues until the packet reaches the sink. This operational mode has more transmission overhead because packets from a malicious node may or may not be caught by a watcher node and they may reach the sink (where it is detected). However, in contrast to the VEBEK-I mode, it reduces the processing overhead (because less re-encoding is performed and decoding is not performed at every hop). The trade-off is that an illegitimate packet may traverse several hops before being dropped. The effectiveness of this scheme depends primarily on the value r, the number of nodes that each node watches. Note that in this scheme, re-encoding is not done at forwarding nodes unless they are bridging the network.

IV. Security Problems

Among all the open problems that sensor networks as a paradigm has to face, security is one of the most important. The sensor nodes, the environment, or the communication channel can be manipulated by any malicious adversary for its own benefit. The first cause of this problem is the hardware constraints inherent to the nodes: due to their small size, their energy consumption requirements, and their limited computational capabilities, it is very difficult to incorporate the mechanisms used as a foundation for secure protocols. The second cause is the public nature of both the wireless channel and the sensor nodes: any device can listen to the communication and the nodes can be accessed and tampered by any external entities. Finally, the third cause is the distributed nature of the sensor network: all protocols have to cooperate for pursuing a common goal, and any internal or external problem may hinder the provision of the network services.

Since sensor networks are very vulnerable against attacks, it is necessary to have certain mechanisms that can protect the network before, during, and after any kind of attack. One of the most important tools for ensuring the security of the network and its services are the security primitives. As mentioned in the introduction, we will consider that those primitives are Symmetric Key Encryption (SKE), Public Key Cryptography (PKC), and Hash functions. Hash and SKE primitives are the building blocks to a basic protection of the information assuring the congeniality and integrity of the channel. Moreover, PKC allow the authentication of the peers involved in any information exchange, thus protecting from the participation of external entities and eliminating the problem of a malicious insider trying to use more than one identity .These primitives are not sufficient by themselves for guaranteeing the overall security of the whole network, since any malicious insider located inside the network can disrupt its behaviour regardless of the protection provided by those primitives. Nevertheless, they are essential for providing basic security to the core protocols of the network, that is to say, the minimal set of protocols required to provide services, such as routing, data aggregation and time synchronization. These core protocols provide packet transmission from one node to another node, grouping a set of sensor readings into one single piece of data, and synchronizing the clocks of the network, respectively.

Moreover, it is possible to create better network services based on the primitives. For example, if the authenticity of a code that is being uploaded to the node using the wireless channel can be assured, it will be possible to update the behaviour of the node or to execute a mobile agent. Also, in most services, sensor nodes have to exchange certain information in order to obtain a global perspective of a situation from local

information. Authenticating the source of such information and assuring its integrity can lead to the creation of better and more efficient trust management algorithms, intrusion detection systems, location procedures, and so on. Another note regarding symmetric security primitives is the need of having a key management system (KMS) for constructing a secure key infrastructure. In most cases, it is not possible to know beforehand where the nodes are going to be located inside the network, but a single sensor node needs to know the keys it shares with its neighbours in order to open a secure channel. This is a well-researched topic, with many types of protocols that try or the most adequate properties for a certain context .

4.1 Obstacles of Sensor Security

A wireless sensor network is a special network which has many constraints compared to a traditional computer network. Due to these constraints it is difficult to directly employ the existing security approaches to the area of wireless sensor networks. Therefore, to develop useful security mechanisms while borrowing the ideas from the current security techniques, it is necessary to know and understand these constraints first .

4.1.1 Very Limited Resources

All security approaches require a certain amount of resources for the implementation, including data memory, code space, and energy to power the sensor. However, currently these resources are very limited in a tiny wireless sensor.

- Power Limitation
- Limited Memory and Storage Space

4.1.2 Unreliable Communication

Certainly, unreliable communication is another threat to sensor security. The security of the network relies heavily on a defined protocol, which in turn depends on communication.

- Unreliable Transfer
- Conflicts
- Latency

4.1.3 Unattended Operation

Depending on the function of the particular sensor network, the sensor nodes may be left unattended for long periods of time. There are three main caveats to unattended sensor nodes:

- Exposure to Physical Attacks
- Managed Remotely
- No Central Management Point

V. Design Module

5.1 Keying Module

It is essentially the method used for handling the keying process. It produces a dynamic key that is then fed into the crypting module. In SVE, each sensor node has a certain virtual energy value when it is first deployed in the network. After deployment, sensor nodes traverse several functional states. The states mainly include node-stay-alive, packet reception, transmission, encoding, and decoding. As each of these actions occurs, the virtual energy in a sensor node is depleted. The current value of the virtual energy, Evc, in the node is used as the key to the key generation function, F. During the initial deployment, each sensor node
will have the same energy level Eini, therefore, the initial key, K1, is a function of the initial virtual energy value and an initialization vector (IV). Subsequent keys, Kj, are a function of the current virtual energy, Evc, and the previous key Kj-1. SVEs virtual energy-based keying module ensures that each detected packet is associated with a new unique key generated based on the transient value of the virtual energy.

5.2 Crypting Module

Due to the resource constraints of WSNs, traditional digital signatures or encryption mechanisms requiring expensive cryptography is not viable. The encoding operation is essentially the process of permutation of the bits in the packet, according to the dynamically created permutation code via the RC4 encryption mechanism. The key to RC4 is created by the previous module (keying module).The purpose of the crypting module is to provide simple confidentiality of the packet header and payload while ensuring the authenticity and integrity of sensed data without incurring transmission overhead of traditional schemes.
However, since the key generation and handling process is done in another module, SVEs flexible architecture allows for adoption of stronger encryption mechanisms in lieu of encoding.

5.2.1 Description of RC4

RC4 consists of two parts. The key scheduling phase will generate the initial permutation from a (random) key of length l bytes. Typically l will be in the range between 5 and 64. The maximal key length is l = 256. The main part of the algorithm is a pseudo random generator that produces one byte output in each step. The encryption will be an XOR of the pseudo random sequence with the message, as usual for stream ciphers.
For the analysis of RC4 it is convenient to replace the original algorithm that works on bytes (Z/256Z) by a generalization that works on Z/nZ for some n ∈ N. For n = 256 we obtain the original algorithm.

Algorithm 1 RC4 key scheduling

```
1: {initialization}
2: for i from 0 to n − 1 do
3: S[i] := i
4: end for
5: j := 0
6: {generate a random permutation}
7: for i from 0 to n − 1 do
8: j := (j + S[i] + K[i mod l]) mod n
9: Swap S[i] and S[j]
10: end for
```

Algorithm 2 RC4 pseudo random generator

```
1: {initialization}
2: i := 0
3: j := 0
4: {generate pseudo random sequence}
5: loop
6: i := (i + 1) mod n
7: j := (j + S[i]) mod n
8: Swap S[i] and S[j]
9: k := (S[i] + S[j]) mod n
10: print S[k]
11: end loop
```

We will call n successive outputs of the RC4 pseudo random generator a round, i.e. the first round will produce the output bytes 1 to n, the second round the bytes n + 1 to 2n and so on. If an attack only uses bytes from the i-th round or later we will call it an i-th round attack.

5.3 Forwarding Module

The node after receiving the packet needs to follow the following steps:

Step1: check for data received
Step2: if yes Get Node Id
Step 3: if received node id =Check watched node
Step 4: send data to next node go to step 1
Step 5: decrypt data, check authenticity if authentic go to step 7
Else go to step 8
Step 6: Get current (my) Key value Encrypt data with My key value
Step 7: send data
Step 8: go to step 1

The topology is taken with multiple clusters .All the sensor nodes communicate to their cluster heads which in turn sends message to the sink node or the base station.

VI. Methodology

6.1 Existing System

An existing Dynamic Energy-based Encoding and Filtering framework to detect the injection of false data into a sensor network. Dynamic Energy-based that each sensed event report be encoded using a simple encoding scheme based on a keyed hash. The key to the hashing function dynamically changes as a function of the transient energy of the sensor, thus requiring no need for re-keying. Depending on the cost of transmission vs. computational cost of encoding, it may be important to remove data as quickly as possible. Accordingly, DEEF can provide authentication at the edge of the network or authentication inside of the sensor network. Depending on the optimal configuration, as the report is forwarded, each node along the way verifies the

correctness of the encoding probabilistically and drops those that are invalid. We have evaluated DEEF's feasibility and performance through analysis our results show that DEEF, without incurring transmission overhead.

6.2 Proposed System

VEBEK is a secure communication framework where sensed data is encoded using a scheme based on a permutation code generated via the RC4 encryption mechanism. The key to the RC4 encryption mechanism dynamically changes as a function of the residual virtual energy of the sensor. Thus, a one-time dynamic key is employed for one packet only and different keys are used for the successive packets of the stream. The intermediate nodes along the path to the sink are able to verify the authenticity and integrity of the incoming packets using a predicted value of the key generated by the sender's virtual energy, thus requiring no need for specific rekeying messages. Our results show that VEBEK, without incurring transmission overhead (increasing packet size or sending control messages for rekeying), is able to eliminate malicious data from the network in an energy efficient manner. The encoding operation is essentially the process of permutation of the bits in the packet, according to the dynamically created permutation code via the RC4 encryption mechanism. The key to RC4 is created by the previous module (virtual energy-based keying module). The purpose of the crypto module is to provide simple confidentiality of the packet header and payload while ensuring the authenticity and integrity of sensed data without incurring transmission overhead of traditional schemes. However, since the key generation and handling process is done in another module, VEBEK's flexible architecture allows for adoption of stronger encryption mechanisms in lieu of encoding. We also show that our framework performs better than other comparable schemes in the literature with an overall 60-100 percent improvement in energy savings without the assumption of a reliable medium access control layer.

VII. Application

Sensor networks may consist of many different types of sensors such as seismic, low sampling rate magnetic, thermal, visual, infrared, acoustic and radar, which are able to monitor a wide variety of ambient conditions that include the following -:

- temperature,
- humidity,
- vehicular movement,
- lightning condition,
- pressure,
- soil makeup,
- noise levels,
- the presence or absence of certain kinds of objects,
- mechanical stress levels on attached objects, and
- the current characteristics such as speed, direction,

and size of an object.

Sensor nodes can be used for continuous sensing, event detection, event ID, location sensing, and local control of actuators. The concept of micro-sensing and wireless connection of these nodes promise many new application areas. We categorize the applications into military, environment, health, home and other commercial areas. It is possible to expand this classification with more categories such as space exploration, chemical processing and disaster relief.

7.1. Military Applications

Wireless sensor networks can be an integral part of military command, control, communications, computing, intelligence, surveillance, reconnaissance and targeting (C4ISRT) systems. The rapid deployment, self-organization and fault tolerance characteristics of sensor networks make them a very promising sensing technique for military C4ISRT.

Sensor networks can be incorporated into guidance systems of the intelligent ammunition.

Battle damage assessment-: Just before or after attacks, sensor networks can be deployed in the target area to gather the battle damage assessment data. Nuclear, biological and chemical attack detection and reconnaissance: In chemical and biological warfare, being close to ground zero is important for timely and accurate detection of the agents. Sensor networks deployed in the friendly region and used as a chemical or biological warning system can provide the friendly forces with critical reaction time, which drops casualties drastically.

7.2 Environmental Applications

Some environmental applications of sensor networks include tracking the movements of birds, small animals, and insects; monitoring environmental conditions that affect crops and livestock; irrigation; macro instruments for large-scale Earth monitoring and planetary exploration; chemical/ biological detection; precision agriculture; biological, Earth, and environmental monitoring in marine, soil, and atmospheric contexts; forest fire

Forest fire detection-: Since sensor nodes may be strategically, randomly, and densely deployed in a forest, sensor nodes can relay the exact origin of the fire to the end users before the fire is spread uncontrollable.
Flood detection -: An example of a flood detection is the ALERT system deployed in the
US. Several types of sensors deployed in the ALERT system are rainfall, water level and weather sensors. These sensors supply information to the centralized database system in a pre-defined way.
Precision Agriculture-: Some of the benefits is the ability to monitor the pesticides level in the drinking water, the level of soil erosion, and the level of air pollution in real time.

7.3 Health Applications

Some of the health applications for sensor networks are providing interfaces for the disabled; integrated patient monitoring; diagnostics; drug administration in hospitals; monitoring the movements and internal processes of insects or other small animals; telemonitoring of human physiological data; and tracking and monitoring doctors and patients inside a hospital. Tracking and monitoring doctors and patients inside a hospital: Each patient has small and light weight sensor nodes attached to them. Each sensor node has its specific task. For example, one sensor node may be detecting the heart rate while another is detecting the blood pressure. Doctors may also carry a sensor node, which allows other doctors to locate them within the hospital.
Drug administration in hospitals-: If sensor nodes can be attached to medications, the chance of getting and prescribing the wrong medication to patients can be minimized. Because, patients will have sensor nodes that identify their allergies and required medications. Computerized systems as described in have shown that they can help minimize adverse drug events.

7.4 Other Commercial Applications

Some of the commercial applications are monitoring material fatigue; building virtual keyboards; managing inventory; monitoring product quality; constructing smart office spaces; environmental control in office buildings; robot control and guidance in automatic manufacturing environments; interactive toys; interactive museums; factory process control and automation; monitoring disaster area; smart structures with sensor nodes embedded inside; machine diagnosis; transportation; factory instrumentation; local control of actuators; detecting and monitoring car thefts; vehicle tracking and detection; and instrumentation of semiconductor processing chambers, rotating machinery, wind tunnels, and anechoic
chambers

VIII. Conclusion

Communication is very costly for wireless sensor networks (WSNs) and for certain WSN applications. Independent of the goal of saving energy, it may be very important to minimize the exchange of messages (e.g., military scenarios). To address these concerns, we presented a secure communication framework for WSNs called Virtual Energy- Based Encryption and Keying.

The possible use of wireless sensor motes and networks extends over a vast area of human activity. Although, most of the applications are still under research and few completed products or services have become available for public use, there is remarkable effort and progress. New scientific fields like pervasive computing have, already, appeared. As most of the applications are focused on monitoring, the distributed sensing seems to enable the parameterization of the physical environment and the integration of it to established forms ofss information propagation (like the internet). Apart from these, adding the parameter “mobility” creates another dimension to the information system.

References

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless Sensor Networks: A Survey,” Computer Networks, vol. 38, no. 4, pp. 393-422, Mar. 2002.
[2] C. Vu, R. Beyah, and Y. Li, “A Composite Event Detection in Wireless Sensor Networks,” Proc. IEEE Int’l Performance, Computing, and Comm. Conf. (IPCCC ’07), Apr. 2007.
[3] S. Uluagac, C. Lee, R. Beyah, and J. Copeland, “Designing Secure Protocols for Wireless Sensor Networks,” Wireless Algorithms, Systems, and Applications, vol. 5258, pp. 503-514, Springer, 2008.
[4] F. Stajano and R. Anderson. The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks wireless networks. In 7th International Workshop on Security Protocols, 1999.
[5] G.J. Pottie and W.J. Kaiser, “Wireless Integrated Network Sensors,” Comm. ACM, vol. 43, no. 5, pp. 51-58, 2000.

[6] R. Roman, C. Alcaraz, and J. Lopez, "A Survey of Cryptographic Primitives and Implementations for Hardware-Constrained Sensor Network Nodes," Mobile Networks and Applications, vol. 12, no. 4, pp. 231-244, Aug. 2007.
[7] H. Hou, C. Corbett, Y. Li, and R. Beyah, "Dynamic Energy-Based Encoding and Filtering in Sensor Networks," Proc. IEEE Military Comm. Conf. (MILCOM '07), Oct. 2007.
[8] L. Eschenauer and V.D. Gligor, "A Key-Management Scheme for Distributed Sensor Networks," Proc. Ninth ACM Conf. Computer and Comm. Security, pp. 41-4, 2002.
[9] M. Eltoweissy, M. Moharrum, and R. Mukkamala, "Dynamic Key Management in Sensor Networks," IEEE Comm. Magazine, vol. 44, no. 4, pp. 122-130, Apr. 2006.
[10] M. Zorzi and R. Rao, "Geographic Random Forwarding (GeRaF) for Ad Hoc and Sensor Networks: Multihop Performance," IEEE Trans. Mobile Computing, vol. 2, no. 4, pp. 337-348, Oct.-Dec. 2003.
[11] I.F. Akyildiz, Weilian Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks, IEEE Communications Magazine, Aug. 2002.
[12] S. Tilak, N.B. Abu-Ghazaleh, and W. Heinzelman. A Taxonomy of Wireless Micro-Sensor Network,Models. ACM SIGMOBILE Mobile Computing and Communications Review, April 2002.
[13] A. Bharathidasan and V. Ponduru. Sensor Networks: An Overview. Technical report, University of California, Davis.
[14] P. Rentala, R. Musunnuri, S. Gandham, and U. Saxena. Survey on Sensor Networks. Technical report, University of Texas at Dallas.
[15] J. Weatherall and A. Jones. Ubiquitous networks and their applications. IEEE Wireless Communications, Feb. 2002.
[16] D. Estrin, D. Culler, K. Pister, and G. Sukhatme. Connecting the physical world with pervasive networks. IEEE Pervasive Computing, Jan.-March 2002.