Energy Efficient Algorithm Using Data Integrity In Wired Networks

[1]T.Balasathuragiri, [2]Prof Mr.T.Anand
[1]B.E (M.E) [2]M.C.A.,Mpil.,M.E.,
[1, 2]Dept. of Computer Science &Engineering
Madha Engineering College
Anna University of Technology
Chennai, India

Abstract
Designing cost-efficient, secure network protocols for **W**ireless **S**ensor **N**etworks (WSNs) are a challenging problem because sensors are resource-limited wireless devices. Since the communication cost is the most dominant factor in a sensor's energy consumption, we introduce an energy-efficient **V**irtual **E**nergy-**B**ased **E**ncryption and **K**eying (VEBEK) scheme for WSNs that significantly reduces the number of transmissions needed for rekeying to avoid stale keys. In addition to the goal of saving energy, minimal transmission is imperative for some military applications of WSNs where an adversary could be monitoring the wireless spectrum. 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. VEBEK is able to efficiently detect and filter false data injected into the network by malicious outsiders.

Index Terms—Security, WSN security, VEBEK, virtual energy-based keying, resource-constrained devices.

1 Introduction
WSN technology is no longer nascent and will be used in a variety of application scenarios. Typical application areas include environmental, military, and commercial enterprises. Future improvements in technology will bring more sensor applications into our daily lives and the use of sensors will also evolve from merely capturing data to a system that can be used for real-time compound event alerting. Securing sensor networks poses unique challenges to protocol builders because these tiny wireless devices are deployed in large numbers, usually in unattended environments, and are severely limited in their capabilities and resources (e.g., power, computational capacity, and memory). For instance, a typical sensor operates at the frequency of 2.4 GHz, has a data rate of 250 Kbps, 128 KB of program flash memory, 512 KB of memory for measurements, transmit power between 100 _W and 1 mW, and a communications range of 30 to 100 m. There are two fundamental key management schemes for WSNs: static and dynamic. In static key management schemes, key management functions (i.e., key generation and distribution) are handled statically. On the other hand, dynamic key management schemes perform keying functions (rekeying) either periodically or on demand as needed by the network. The purpose of this paper is to develop an efficient and secure communication framework for WSN applications. Specifically, in this paper, we introduce **V**irtual **E**nergy-**B**ased **E**ncryption and **K**eying (VEBEK) for WSNs, which is primarily inspired by our previous work. VEBEK's secure communication framework provides a technique to verify data in line and drop false packets from malicious nodes, thus maintaining the health of the sensor network. VEBEK dynamically updates keys without exchanging messages for key renewals and embeds integrity into packets as opposed to enlarging the packet by appending **M**essage **A**uthentication **C**odes (MACs). The contributions of this paper are as follows:

1. A dynamic en route filtering mechanism 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 medium access control layers.

2 Backgrounds And Motivation
One significant aspect of confidentiality research in WSNs entails designing efficient key management schemes. This is because regardless of the encryption mechanism chosen for WSNs, the keys must be made available to the communicating nodes (e.g., sources and sink(s)). The former is static key management and the latter is dynamic key management. There are myriads of variations of these basic schemes in the literature. In this work, we only consider dynamic keying mechanisms in our analysis

since VEBEK uses the dynamic keying paradigm. Dynamic keying schemes go through the phase of rekeying either periodically or on demand as needed by the network to refresh the security of the system.

EDyn =(EKdisc +Ecomp)* E[rh]*X/T

Where X is the number of packets in a message is the key refresh rate in packets per key; EKdisc is the cost of shared key discovery with the next hop sensor after initial deployment, and E[rh] is the expected number of hops. In the dynamic key-based schemes, T may change periodically, on demand, or after a node-compromise. A good analytical lower bound for E[rh] is given in as

E[rh]=D _ tr/E[dh]+ 1;

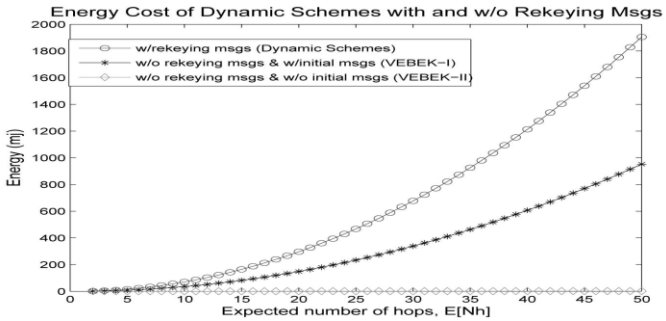

Fig. 1. Keying cost of dynamic key-based schemes based on E[nh] versus VEBEK.

Where D is the end-to-end distance (m) between the sink and the source sensor node, tr is the approximated transmission range (m), and E[dh] is the expected hop

Enode =Etx + Erx + Ecomp

Where Enode is the approximate cost per node for key
generation and transmission, E[Ne] is the expected number of neighbors for a given sensor, M is the number of key establishment messages between two nodes, and Etx and Erx are the energy cost of transmission and reception, respectively. Given the transmission range of sensors (assuming bidirectional communication links for simplicity), tr, total deployment area, A, total number of sensors deployed, N, E=[Ne] can be computed as

E[Ne]=N*:╥* t2r/A

VEBEK does rekeying without messages. There are two operational modes of VEBEK (VEBEK-I and VEBEK-II). The details of these modes are given in. However, for now it suffices to know that VEBEK-I is representative of a dynamic system without rekeying messages, but with some initial neighborhood info exchange whereas VEBEK-II is a dynamic system without rekeying messages and without any initial neighborhood info exchange.

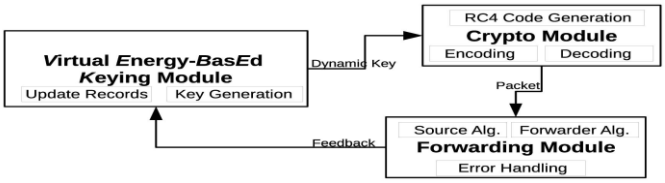

Fig. 2. Modular structure of VEBEK framework.

3 Semantics Of Vebek

The VEBEK framework is comprised of three modules: Virtual Energy-Based Keying, Crypto, and Forwarding. The virtual energy-based keying process involves the creation of dynamic keys. Contrary to other dynamic keying schemes, it does not exchange extra messages to establish keys. A sensor node computes keys based on its residual virtual energy of the sensor. The key is then fed into the crypto module. The crypto module in VEBEK employs a simple encoding process, which is essentially the process of permutation of the bits in the packet according to the dynamically created permutation code generated via RC4. The encoding is a simple encryption mechanism adopted for VEBEK. However, VEBEK's flexible architecture allows for

adoption of stronger encryption mechanisms in lieu of encoding. Last, the forwarding module handles the process of sending or receiving of encoded packets along the path to the sink.

TABLE 1
Notations Used

E_{tx}	Tx energy	E_{sens}	Sensing energy
E_{rx}	Rx energy	E_{sa}	Staying alive energy
E_{comp}	Computation energy	E_{vc}	Virtual cost
E_{enc}	Encoding energy	E_p	Perceived energy
E_{dec}	Decoding energy	E_b	Bridge energy

A high-level view of the VEBEK framework and its underlying modules are shown in Fig. 2. These modules are explained in further detail below. Important notations used are given in Table 1, Table 2.

TABLE 2
Notations Used

E_{Fw}	Forwarding energy	P_{drop}	Drop probability
E_{Kdisc}	Key discovery energy	φ	Synch ratio
E_{Dyn}	Dynamic keying cost	l	packet size
E_{So}	Source node energy	N	# of nodes
$E[\eta_h]$	Expected # of hops	r	# of watched nodes

3.1 Virtual Energy-Based Keying Module

The virtual energy-based keying module of the VEBEK Frame work is one of the primary contributions of this paper. It is essentially the method used for handling the keying process. It produces a dynamic key that is then fed into the crypto module. In VEBEK, each sensor node has a certain virtual energy value when it is first deployed in the network. The rationale for using virtual energy as opposed to real battery levels as in our earlier work, DEEF, is that in reality battery levels may fluctuate and the differences in battery levels across nodes may spur synchronization problems, which can cause packet drops. 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. VEBEK includes packet reception (Erx), packet transmission (Etx), packet encoding (Eenc), packet decoding (Edec) energies, and the energy required to keep a node alive in the idle state (Ea).3 specifically, the transient value of the virtual energy, Ev, is computed by decrementing the total of these predefined associated costs, Evc, from the previous virtual energy value.

Evc = l * (etx + eenc)+ t * ea+ Esync.

Thus, assuming that the receiving node has the initial virtual energy value of the sending node and that the packet is successfully received and decoded associated with a given source sensor, k, the virtual cost of the perceived energy is computed as follows:

Ekp= l*(erx + edec+ etx+ eenc)+ t* 2* ea.

Where in both the equations, the small es refer to the one bit energy costs of the associated parameter. However, Esynch in refers to a value to synchronize the source with the watcher-forwarders toward the sink as watcher-forwarder nodes spend more virtual energy due to packet reception and decoding operations, which are not present in source nodes.

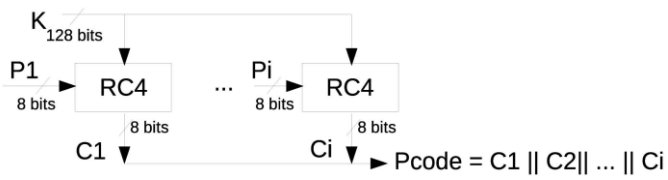

Fig.3. an illustration of the use of RC4 encryption mechanism in VEBEK.

3.2 Crypto Module

Due to the resource constraints of WSNs, traditional digital signatures or encryption mechanisms requiring expensive cryptography is not viable. The scheme must be simple, yet effective. Thus, in this section, we introduce a simple encoding

operation similar to that used. 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.

The benefits of this simple encoding scheme are: 1) since there is no hash code or message digest to transmit, the packet size does not grow, avoiding bandwidth overhead on an already resource-constrained network, thus increasing the network lifetime, 2) the technique is simple, thus ideal for devices with limited resources (e.g., PDAs).
3) The input to the RC4 encryption mechanism, namely, the key, changes dynamically without sending control messages to rekey.

3.3 Forwarding Module

The final module in the VEBEK communication architecture is the forwarding module. The forwarding module is responsible for the sending of packets (reports) initiated at the current node (source node) or received packets from other sensors (forwarding nodes) along the path to the sink. The reports traverse the network through forwarding nodes and finally reach the terminating node, the sink. The operations of the forwarding module are explained in this section.

3.3.1 Source Node Algorithm

When an event is detected by a source node, the next step is for the report to be secured. The source node uses the local virtual energy value and an IV (or previous key value if not the first transmission) to construct the next key. As discussed earlier, this dynamic key generation process is primarily handled by the VEBEK module. The source sensor fetches the current value of the virtual energy from the VEBEK module. Then, the key is used as input into the RC4 algorithm inside the crypto module to create a permutation code for encoding the hIDjtypejdatai message. The encoded message and the cleartext ID of the originating node are transmitted to the next hop (forwarding node or sink) using the following format: ID; fID; type; datagPc_, where fxgPc constitutes encoding x with permutation code Pc. The local virtual energy value is updated and stored for use with the transmission of the next report.

3.3.2 Forwarder Node Algorithm

Once the forwarding node receives the packet it will first check its watch-list to determine if the packet came from a node it is watching. If the node is not being watched by the current node, the packet is forwarded without modification or authentication. Although this node performed actions on the packet (received and forwarded the packet), its local virtual perceived energy value is not updated. This is done to maintain synchronization with nodes watching it further up the route. If the node is being watched by the current node, the forwarding node checks the associated current virtual energy record stored for the sending node and extracts the energy value to derive the key.

4 Operational Modes Of Vebek

The VEBEK protocol provides three security services: Authentication, integrity, and non repudiation. The VEBEK framework also considers this need for flexibility and thus, supports two operational modes: VEBEK-I and VEBEK-II.

4.1 VEBEK-I

VEBEK-I operational mode, all nodes watch their neighbors; whenever a packet is received from a neighbor 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. 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**.**

4.2 VEBEK-II

VEBEK-II operational modes, 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.

5. Comparison Of VEBEK-II With Other Statistical Schemes

In this section, we evaluate the energy performance of VEBEK-II with other "en-route dynamic filtering" works in the literature. We focus on statistical schemes because they have received a lot of attention in recent years. Specifically, we compare the expected energy costs of DEF, SEF, and STEF with that of VEBEK-II because VEBEK-II is the statistical mode of the VEBEK framework. First, we briefly summarize each protocol and discuss their drawbacks. Then, the comparison results are presented.

An illustration of each protocol is given in Fig. 11. In the **D**ynamic **E**n-route **F**iltering (DEF) scheme by Yu and Guan, a legitimate report is endorsed by multiple sensing nodes using their own authentication keys. Before deployment, each node is preloaded with a seed authentication key and l+1 secret key randomly chosen from a global key pool.

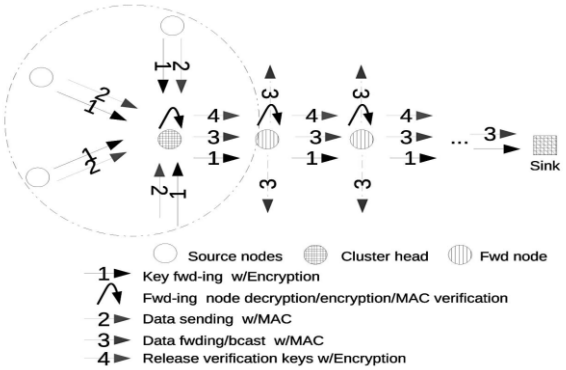

Fig.04 Dynamic En-route Filtering

Ye et al., proposed **S**tatistical **E**n-route **F**iltering (SEF). In SEF, each sensing report is validated by multiple keyed message authentication codes. Specifically, each node is equipped with some number of keys that are drawn randomly from the global key pool. First, a center of stimulus is selected among the source sensor nodes in the event region. Then, once a report is generated by a source node, a MAC is appended to the report. Next, another upstream node that has the same key as the source can verify the validity of the MAC and filters the packet if the MAC is invalid. However, the downside of SEF is that the nodes must store keys and packets are enlarged by MACs. Although the authors suggest the use of bloom-filters to decrease the MAC overhead, SEF is a static key-based scheme and it inherits all the downsides of static key management schemes. DEF and SEF are probabilistic schemes; a comparison of each scheme with VEBEK-II in terms of their energy consumption is presented in Fig. 12. The results are generated for one round of communication from a source node to the sink, which is assumed to be located n hops away from the source node. The x-axis represents the hop count and is varied, while the y-axis is the energy. To simplify the comparisons, we assumed that all the nodes in DEF, SEF, and VEBEK-II would have the necessary keying material with 0.7 probabilities to do the desired security features imposed by the specific protocol in a benign environment (no malicious nodes).

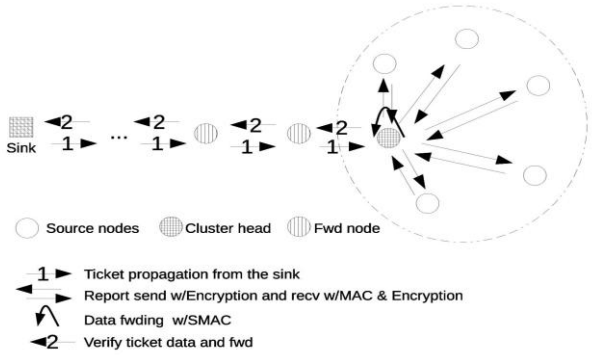

Fig.05 Secure Ticket-Based En-route Filtering

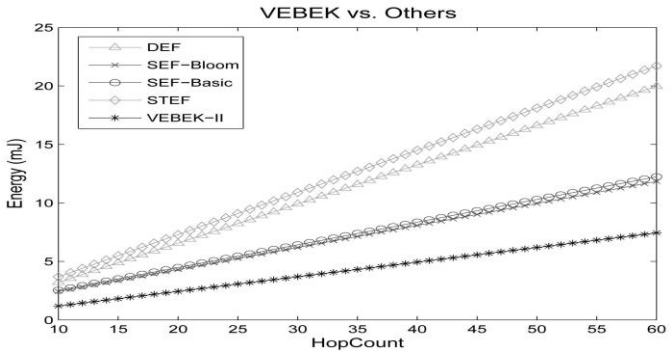

Fig. 06. Comparison of VEBEK, DEF, SEF, and STEF

6 Related Work

En route dynamic filtering of malicious packets has been the focus of several studies, including DEF by Yu and Guan, SEF, and STEF . As the details are given in the performance evaluation section. Where they were compared with the VEBEK framework, the reader is referred to that section for further details as not to replicate the same information here. Moreover, Ma's work applies the same filtering concept at the sink and utilizes packets with multiple MACs appended. A work proposed by Hyun and Kim uses relative location information to make the compromised data meaningless and to protect the data without cryptographic methods. In using static pair wise keys and two MACs appended to the sensor reports, "an interleaved hop-by-hop authentication scheme for filtering of injected false data" was proposed by Zhu et al. to address both the insider and outsider threats. Another crucial idea of this paper is the notion of sharing a dynamic cryptic credential (i.e., virtual energy) among the sensors. A similar approach was suggested inside the SPINS study [24] via the SNEP protocol. In particular, nodes share a secret counter when generating keys and it is updated for every new key.

7 Conclusions and Future Work

Communication is very costly for **W**ireless **S**ensor **N**etworks (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. In comparison with other key management schemes, VEBEK has the following benefits:

1) It does not exchange control messages for key renewals and is therefore able to save more energy and is less chatty.

2) 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.,

3) 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.

We have evaluated VEBEK's feasibility and performance through both theoretical analysis and simulations. Our results show that different operational modes of VEBEK (I and II) can be configured to provide optimal performance in a variety of network configurations depending largely on the application of the sensor network. We also compared the energy performance of our framework with other en route malicious data filtering schemes. Our results show that VEBEK performs better (in the worst case between 60-100 percent improvements in energy savings) than others while providing support for communication error handling, which was not the focus of earlier studies. Our future work will address insider threats and dynamic paths.

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] Crossbow Technology, http://www.xbow.com, 2008.

[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.