International Journal on Applications in Information and Communication Engineering
Volume 1 : Issue 4 : April 2015, pp 60 – 65 www.aetsjournal.com ISSN (Online) : 2394 - 6237

SHELTERED WIRELESS ANTENNA GRID USING DYNAMISM BASED ENCRYPTION

Dr. SIVAKUMAR K , Mr. D. KALEESWARAN

Abstract — Dynamic — In basic sensor organizations it is fundamental to frame sure the credibility and uprightness of detected information. Further, one must affirm that bogus information infused into the system by malevolent hub isn't seen as precise information. Sensors are asset restricted remote gadgets. Since correspondence cost is that the premier predominant accept a sensor's vitality utilization, we presents a vitality effective Virtual Energy-Based Encryption and Keying (VEBEK) plot for WSNs that essentially diminishes the amount of transmissions required for re-keying to maintain a strategic distance from stale keys. In extra to the objective of sparing vitality, negligible transmission is basic. VEBEK could likewise be a safe correspondence structure where detected information is encoded utilizing a plan upheld a change code created by means of the RC4 encryption instrument. The way in to the RC4 encryption system progressively changes as an element of the leftover virtual vitality of the sensor. In this way, a one-time powerful key's utilized for one bundle just and various keys are utilized for the progressive parcels of the stream. VEBEK is during a situation to proficiently identify and channel bogus information infused into the system by pernicious pariahs. The VEBEK structure comprises of two operational modes VEBEK-I and VEBEK-II. VEBEK-I: Each hub screens its one-jump neighbors and VEBEK-II: It screens downstream hubs. VEBEK, without bringing about transmission overhead (expanding bundle size or sending control messages for rekeying), is during a situation to take out malignant information from the system in a vitality productive manner.Index Terms – Security, VEBEK, Virtual energy-based keying, resource-limited devices

Keywords— Sensors, Security, VEBEK-II, VEBEK-I, RC4 encryption, etc.

I. INTRODUCTION

Sensor network technology has rapidly developed in recent years and will be used in a variety of application scenarios. Typical application includes environmental, military, and commercial enterprises. Sensors may be used to detect the location of enemy sniper fire or to detect harmful chemical agents before they reach troops. It is important to provide authentic and accurate data to surrounding sensor nodes and to the sink to trigger time-critical responses (e.g., troop movement, evacuation, and first response deployment). Protocols should be resilient against false data injected into the network by malicious nodes. Otherwise, consequences for propagating false data or redundant data are costly, depleting limited network resources and wasting response efforts.

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). The protocol builders must be cautious about utilizing the limited resources of the sensors efficiently.

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 statistically. That is, sensors have fixed number of keys loaded either prior to or shortly after network deployment. On the other hand, dynamic key management schemes perform keying functions (rekeying) either periodically or on demand as needed by the network. The sensors dynamically exchange keys to communicate. Although dynamic schemes are more attack resilient than static ones, one significant disadvantage is they increase the communication overhead due to keys being refreshed or re-distributed from time to time in the network. There are many reasons for key refreshment, including: updating keys after key revocation has occurred, refreshing the keys such that it does not become stale, or changing keys due to dynamic changes in the topology. Here, we seek to minimize the overhead associated with refreshing keys to avoid them becoming stale. Because communication cost is the most dominant factor in sensors energy consumption and message transmission cost for rekeying is an important issue in WSN. It is important to minimize the number of messages to decrease the probability of detection if deployed in enemy territory. This is done to decrease the number of opportunities for malicious entities to eavesdrop or intercept the packets.

Dr. Sivakumar K MSc., ME., Professor & Dean - Computer Science and Engineering , Hindusthan Institute of Technology , Coimbatore.
(Email: rksivakumar@gmail.com)
Mr. D. Kaleeswaran M.E., Assistant Professor - Computer Science and Engineering , St. Michael College of Engineering & Tehnology , Sivaganagai.
(Email:kaleeswaranme@gmail.com

The purpose of the paper is to develop an efficient and secure communication framework for WSN applications. We introduce an **V**irtual **E**nergy–**B**ased **E**ncryption and **K**eying (VEBEK), which is a secure communication framework that verifies data in line and drop false packets from malicious nodes, thus maintaining the health of the sensor network. 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 the function of the residual virtual energy of the sensor, thus requiring no need for rekeying. Therefore, a one-time dynamic key is generated 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 operation under dire communication cases as it may be operating in a high-error-prone deployment are 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. It provides:

- A dynamic en route filtering mechanism that does not exchange explicit control messages for rekeying.
- Provision of one-time keys for each packet transmitted to avoid stale keys.
- A modular and flexible security architecture with a simple technique for ensuring authenticity, integrity, and non-repudiation of data without enlarging packets with MACs; and
- A robust and secure communication framework that is operational in dire communication situations and over unreliable medium access control layers.

II. BACKGROUND AND MOTIVATION

One significant aspect of confidentiality research in WSNs ensures designing efficient key management schemes. This is because regardless of encryption mechanism chosen for WSNs, the keys must be made available to communicating nodes (source and sink). The keys could be distributed to the sensors before the network deployment or they could be redistributed to nodes on demand. Rekeying with control messages is the approach of existing dynamic keying schemes whereas rekeying without extra control messages is the primary feature of VEBEK framework.

Dynamic keying schemes go through the phase of rekeying either periodically or on demand as needed by the network. With rekeying, the sensors dynamically exchange keys for communication. Hence, the energy cost function while sending a message to sink with dynamic key-based schemes can be written as follows (assuming E_{comp} is approximately fixed):

$E_{Dyn} = (E_{Kdisc} + E_{comp}) * E[\eta h] * \chi / \tau$,

Where χ is the number of packets in a message, τ is the key refresh rate in packets per key, E_{Kdisc} is the cost of shared key discovery with the next hop after initial deployment, $E[\eta h]$ is expected number of hops. In the dynamic key-based schemes, τ may change periodically. A good analytical lower bound for $E[\eta h]$ is given as,

$E[\eta h] = \frac{D-tr}{E[dh]} + 1$,

Where D is the end to end distance(m) between sink and the source sensor node, tr is the approximated transmission range(m), $E[d_h]$ is the expected hop distance(m). Finally, E_{Kdisc} can be written as:

$E_{Kdisc} = \{E [N_e] * E_{node}) * M – 2*E_{node}\}$

$E_{node} = E_{tx}+E_{rx}+E_{comp}$,

Where E_{node} is the approximate cost per node for key generation and transmission, $E[N_e]$ is the expected number of neighbors for a given sensor, M is the number of key establishment messages between two nodes, and E_{tx} and E_{rx} are the energy cost for transmission and reception. Given the transmission range of sensors, t_r, total deployment area, A, total number of sensors deployed, N, $E[N_e]$ can be computed as

$E [N_e] = \frac{N * \pi * tr2}{A}$

VEBEK does rekeying without messages. With this analysis, we see that a dynamic key-based scheme spends a large amount of energy transmitting rekeying messages. VEBEK is motivated to provide same benefits but with low energy consumption. It does not exchange extra control messages for key renewals. Hence, energy is only consumed for generating keys. The keys are dynamic; thus, one key per packet is employed.

III. SEMANTICS OF VEBEK

The VEBEK framework is composed of three modules: Virtual Energy-Based Keying, Crypto and Forwarding Module.

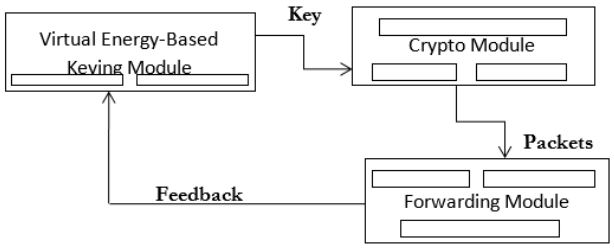

Figure 1: Structure of VEBEK Framework

The virtual Energy-Based keying process involves the creation of dynamic keys. A sensor node computes keys based on its residual virtual energy of the sensor. The crypto module in VEBEK employs a simple encoding process, which is the process of permutation of the bits in the packets according to the dynamically created permutation code generated via RC4. The encoding is a simple encryption adopted. Last, the forwarding module handles the process of sending or receiving of encoded packets along the path to the sink.

A. Virtual Energy-Based Keying module

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. In this module, we compute **D**ynamic **K**ey.

After deployment, sensor nodes traverse several functional states. The states mainly include node-stay-alive, packet reception, packet transmission, encoding, and decoding. As each of the action 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 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). The IVs are pre-distributed to the sensors. VEBEK's virtual energy-based keying module ensures that each packet is associated with a new unique key generated based on the transient value of the virtual energy. After the dynamic key is generated, it is passed to the crypto module, where the desired security services are implemented. The process of key generation is initiated when the data is sensed; thus no explicit mechanism is needed to refresh or update keys. Moreover, the dynamic nature of keys makes it difficult for attackers to intercept enough packets to the encoding algorithm. Each node computes and updates the transient value of its virtual energy after performing some actions. Each action (or state traversal) is associated with a predetermined cost. Since a sensor node will be either forwarding some other sensor's data or injecting its own data into the network, the set of actions and their associated energies for VEBEK includes packet reception (Erx), packet transmission (Etx), packet encoding (Eenc), packet decoding (Edec) energies, the energy required to keep a node alive in the idle state (Ea). 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.

Algorithm 1: Compute Dynamic Key

1. Compute Dynamic Key(E_{vc},ID_{clr})
2. **begin**
3. j ← tx_{cnt}
4. **if** j = 1 **then**
5. K*j* ← F(E_{ini}, IV)
6. **else**
7. K*j* ← F($K_{(j-1)}$, E_{vc})
8. **end if**
9. **return** K_j
10. **end**

In order to successfully decode and authenticate a packet, a receiving node must keep track of the energy of the sending node to derive the key needed for decoding. In VEBEK, the operation of tracking the energy of the sending node at the receiver is called watching and the energy value associated with the watched sensor is called Virtual Perceived Energy ($E_{p)}$.

B. Crypto Module

Due to resource constraints of sensor networks, traditional digital signatures requiring expensive cryptography is not a viable option. The scheme must be simple, yet effective. Simple encoding process is employed to ensure the authenticity and integrity of the sensed data without incurring transmission overhead.

The packets in VEBEK consist of ID, type and data bits. Each node sends these to its next hop. The sensors ID, type and the data are transmitted in a pseudorandom fashion according to the result of RC4. RC4 algorithm takes key and packet fields as input and produces the result as permutation code. The concatenation of each 8-bit output becomes the resultant permutation code. The resultant permutation code is used to encode the <ID|type|data>. Along with the encoded message, an additional copy of the ID is transmitted. The format of final packet becomes

Packet = [ID, $\{ID|type|data\}_k$]

Where $\{x\}_k$ constitutes encoding x with key k.

Instead of the tradition approach of sending the hash value along with the information to be sent, we use the result of the permutation code value locally. When the next node along the path to the sink receives the packet, it generates the local permutation code to decode the packet.

The benefits of simple encoding scheme are: 1) No hash code to transmit so the packet size doesn't grow, avoiding bandwidth overhead 2) technique is very simple, thus ideal for devices with limited resources 3)Key changes dynamically without sending control messages to rekey.

C. Forwarding Module

This module is responsible for the sending of packets initiated at the current node or received packets from other sensor nodes along the path to the sink.

1) Source Node Algorithm

When an event is detected by the source sensor node, the next step is the report to be secured. The source node uses the virtual energy value and an IV to construct the next key. 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 <ID|type|data>. The encoded message and the clear text ID of the originating node is transmitted to the next hop (forwarding node or sink) using the following format [ID, $\{ID|type|data\}_{pc}$], where $\{x\}_{pc}$ constitutes encoding x with the permutation code pc. The local virtual energy value is updated and stored for use with the transmission of the next report.

2) Forwarding 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. Although this node performed actions on the packet (received and forwarded), its local virtual perceived energy value is not updated. This is done to maintain synchronization with the 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 sending node and extracts the energy value to derive the key. It then authenticates the message by decoding the message by comparing the plaintext node ID with the encoded node ID. If the packet is authentic, an updated virtual energy value is stored in the record associated with the sending node. If the packet is not authentic, it is discarded.

3) Addressing Communication Errors via Virtual Bridge Energy

In *V*EBEK, to authenticate a packet, a node must keep track of the virtual energy of the sending node to decode the packet. Ideally, once the authenticating node has the initial virtual energy value of the sending node, the value can be updated by decrementing the cost associated with the actions performed by the sending node. However, communication errors may cause some of the packets to be lost or dropped. Some errors may be due to the deployment region while operating on unreliable underlying protocols (e.g., medium access control protocol). For instance, ACK or data packets can be lost and the sender may not know which one actually was lost. Moreover, malicious packets inserted by the attackers will be dropped intentionally by other legitimate sensors to filter out the bad data from the network. In such communication errors or intentional packet drop cases, the virtual energy used to encode the packet at the sending node may differ from virtual energy value that is stored for the sending node at its corresponding watching node. Specifically, the node that should have received the dropped packet and the nodes above that node lose synchronization with the nodes below. If another packet has to be forwarded by the current watching node using its current virtual energy, the upstream nodes that watch this particular node would discard the packet.

To resolve the potential loss of packets due to possible communication errors in the network, all the nods are configured to store an additional virtual bridge energy value, which we refer to as Virtual Bridge Energy, E_{bi} , value to allow resynchronization of the network at the next watching sensor node that determines that the packet were lost.

That is, subsequent packets generated from the node of interest pass through the next watching node, the next watching node will decode the packet with the virtual perceived energy key of the originating node and reencode the packet with the virtual bridge energy key, thus, the network will be kept synchronized. The watching node always updates and uses this parameter to keep the network bridged.

Another Pertinent point is the determination of packet loss by the first upstream watching node that bridges the network. The VEBEK framework designed to avoid extra messages and not increase the packet size to determine packet loss in the network. Thus, the next

watching node tries to find the correct value of the virtual perceived energy for the key within a window of virtual energies. For this, a sensor is configured with a certain VirtualKeySearchThreshold value.

IV. OPERATIONAL MODES OF VEBEK

The VEBEK protocol provides three security services: Authentication, integrity, and nonrepudiation. The fundamental notion behind providing these services is the watching mechanism. 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.

A. VEBEK I

In the 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. 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 one-hop neighbors in its watch-list. To obtain a neighbor'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 preloaded 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 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 virtualKeySearchThreshold. If the packet is authentic, and this hop is not the final hop, the packet is reencoded 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. Reencoding at very 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 encode/decode that occurs at each hop.

B. 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. 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 virtualKeySearchThreshold 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. If the packet is illegitimate, which is

classified as such after exhausting all the virtual perceived energy values within the virtualKeySearchThreshold 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 reencoding 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.

V. 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.

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), and 3) it unbundles key generation from security services, providing a flexible modular architecture. 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] 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.

[3] 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.

[4] 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.

[5] F. Ye, H. Luo, S. Lu, and L. Zhang, "Statistical En-Route Filteringof Injected False Data in Sensor Networks," IEEE J. Selected Areas in Comm., vol. 23, no. 4, pp. 839-850, Apr. 2005.

[6] Z. Yu and Y. Guan, "A Dynamic En-Route Scheme for Filtering False Data Injection in Wireless Sensor Networks," Proc. IEEE INFOCOM, pp. 1-12, Apr. 2006.

[7] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," Proc. ACM MobiCom, pp. 56-67, Aug. 2002.

[8] C. Kraub, M. Schneider, K. Bayarou, and C. Eckert, "STEF: A Secure Ticket-Based En-Route Filtering Scheme for Wireless Sensor Networks," Proc. Second Int'l Conf. Availability, Reliability and Security (ARES '07), pp. 310-317, Apr. 2007.

[9] S. Zhu, S. Setia, S. Jajodia, and P. Ning, "An Interleaved Hop-by-Hop Authentication Scheme for Filtering of Injected False Data in Sensor Networks," Proc. IEEE Symp. Security and Privacy, 2004.

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