ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 5, Special Issue 3, March 2016

6th International Conference in Magna on Emerging Engineering Trends 2016 [ICMEET 2016]

On 10th & 11th March, 2016

Organized by

Dept. of ECE, EEE & CSE, Magna College of Engineering, Chennai-600055, India.

Energy Efficiency and Secure Synchronization in Wireless Sensor Networks

M.Sudar Manikandan[1], B.Siva Murugan[2], A.Micheal Enosin[3]

UG Student, Department of IT, Rajas Engineering College, Raja Nagar, Vadakkangulam, Tamil Nadu, India[1]

UG Student, Department of IT, Rajas Engineering College, Raja Nagar, Vadakkangulam, Tamil Nadu, India[2]

UG Student, Department of IT, Rajas Engineering College, Raja Nagar, Vadakkangulam, Tamil Nadu, India[3]

ABSTRACT: This paper address the secure synchronization problem is rectified for WSNs.Secure source based Loose synchronization protocol to securely synchronize the events in the network without transmission of explicit synchronization control messages. Data can be automatically transfer to desired node using auto message option. Nodes use their local time values as a onetime dynamic key to encrypt of each message by a RC4 encryption mechanism. The authorized nodes use their local time to intelligently decode the timing key of the source node. So maintain more secured and authenticity. This protocol provides effective dynamic en-route filtering mechanism, where the malicious data is filter out from the networks. Time based dynamic keying and en-route filtering is twice more energy efficient than existing related work. Loose synchronization protocol event reports generated by the nodes are ordered properly prior to exiting the sensor network. Thus, energy savings from the reduced transmission is used for the local security computation.To achieve our main goal of synchronizing events at the sink as quickly, as accurately. It reduces the number of control messages needed for a WSN to operate providing the key benefits of reduced energy consumption as well as reducing the opportunity for malicious nodes to eavesdrop, intercept, or be made aware of the presence of the network. Both analytical and simulation results are presented to verify the feasibility as well as the energy consumption of the scheme under normal operation and attack from malicious nodes.

KEYWORDS: Secure loose synchronization, secure time synchronization, SOBAS,wireless sensor networks, sensor-based cyber-physical systems

I. INTRODUCTION

In Wireless sensor networks(WSNs) current secure time synchronization protocol send separate synchronization messages and utilize reference points, global positioning systems(GPSs) and dynamic key-based cryptographic mechanisms to ensure that the clocks of each sensor device are securely globally synchronized. In addition to being costly in terms of energy consumption, these control messages (albeit necessary for their schemes) used in traditional protocols make them difficult to deploy in situations where it is required that the radio frequency (RF) footprint of the communicating devices be minimal (e.g., military sensor networks). In fact, many wireless sensor network applications are event-based and centrally controlled (e.g., critical infrastructure monitoring, video surveillance, and patient-data collection), and do not necessarily need to maintain perfect synchronization of all the nodes in the network. Instead, it must be guaranteed that the event reports generated by the nodes are ordered properly prior to exiting the sensor network (i.e., loose synchronization).

Assume, for instance, the military surveillance application in Fig.1.The most valuable piece of information to the headquarters is that the enemy unit intruded the protected zone and is advancing South-West. In this case, the proper ordering of events can be achieved by synchronizing the sink with each source, and does not require that each sensor's clock be globally synchronized. In fact, an accurate knowledge of event ordering may generally suffice for many WSN applications, where the centralized decision authority (not sensors) acts swiftly on the information collected from the network.

Second, because the communication cost is the most dominant factor in a sensor's energy consumption [1], [2], if the synchronization control messages in the network are eliminated as opposed to current "chatty" schemes, some of the energy savings from transmission cost can be utilized for the computation of local security operations. Therefore, motivated by the downside of current schemes and considering the event-based characteristics of WSN applications [3]

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 5, Special Issue 3, March 2016

6th International Conference in Magna on Emerging Engineering Trends 2016 [ICMEET 2016]

On 10th & 11th March, 2016

Organized by

Dept. of ECE, EEE & CSE, Magna College of Engineering, Chennai-600055, India.

and the resource-limited nature of sensors [4], we propose the Secure Source Based Loose Synchronization protocol. Essentially, SOBAS is an energy efficient protocol for WSN applications that do not need perfect synchronization. SOBAS presents an effective technique to securely synchronize the data path in the network, without the transmission of explicit synchronization control messages.

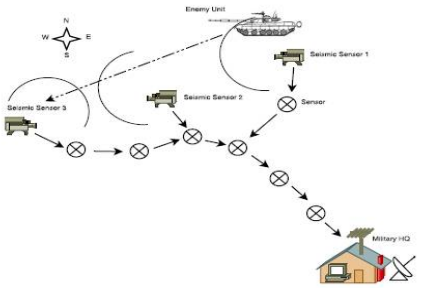

Fig 1 Military surveillance application

Instead of synchronizing each sensor globally as opposed to approaches providing perfect synchronization, we focus on ensuring that each source node is synchronized with the sink and nodes along the data delivery path such that event reports generated by the sink are ordered properly. With SOBAS, we are able to achieve our main goal of synchronizing events loosely at the sink and at the data delivery path as quickly, as accurately, and as surreptitiously as possible. SOBAS is perfectly suitable for WSN applications that do not need perfect synchronization and it is able to provide 7.24 µs clock precision on the data delivery path given today's sensor technology. Simulation results show that SOBAS is an energy efficient scheme under normal operation and attack from malicious nodes. In addition to being suitable acts on the information collected from the network, our novel approach to synchronization with dynamic en-route filtering is also well suited for both WSNs and sensor-based CPS applications where utmost silence is necessary (e.g., military scenarios), as SOBAS is not "chatty." For instance, radio silence is very important for military operations as any radio transmission may reveal troop positions.

II. SOBAS PROTOCOL

There are three main components of the SOBAS protocol:
Time-Based Key Management (TKM), Crypto (CRYPT), and
Filtering-Forwarding-Synch (FFS) Modules.

2.1 Threat model

Source nodes are synchronized and loaded with a network wise initialization vector (IV) predeployment. The IV and local time information will be used to generate the initial and subsequent dynamic keys. Note that the sensor nodes do not have perfect clocks and over time the sensors' clocks gradually diverge from the real clock value due to changes in the environmental conditions such as temperature, humidity, pressure, and vibration. In the worst case, they can accumulate up to several seconds of error per day [6].Thus; the dynamic keys generated in SOBAS will change as a function of time and random drift. As such, the same event reported by different sources located nearby or separate events reported by different sources located elsewhere in the deployment region use different keys. In fact, this is an instrumental property, which we refer to as a Spatio-Temporal Security Property for WSN applications. Also, SOBAS does not incur a cost to discover which keys are shared between any two neighboring node (shared-key discovery phase [5]) via explicit messages because the nodes use the local time information to create the dynamic keys. SOBAS, our main goal is to loosely synchronize and order events at the sink as energy-efficient, precise, and surreptitious as possible to reduce the likelihood of interception by an adversary.

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 5, Special Issue 3, March 2016

6th International Conference in Magna on Emerging Engineering Trends 2016 [ICMEET 2016]

On 10th & 11th March, 2016

Organized by

Dept. of ECE, EEE & CSE, Magna College of Engineering, Chennai-600055, India.

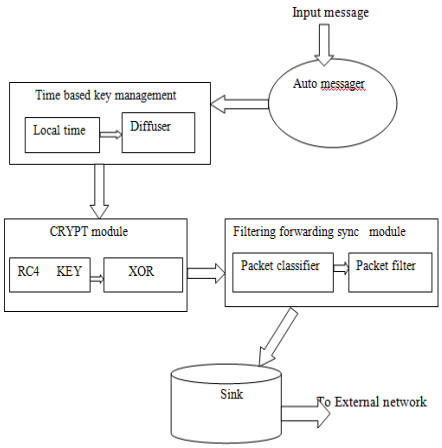

Fig 2 Key generation function and encryption

2.2 Time-Based Key Management Module

In loose synchronization protocol, keys are generated dynamically using local time. This is addressed in the Time Based Key Management module. When a source node has data to send to the sink due to either an external stimulation by the sink or a self-initiated periodic report, it uses its local clock value as the key. However, the TKM module ensures that keys generated in the module are as random. Specifically, the keys are a function of the current local time value (t_l) and either an initialization vector (IV) or previous key, K_{J-1} as

$$K^t_1 = F(t_l, IV) \quad (1)$$

$$K^t_j = F(t_l, K_{j-1}) \quad (2)$$

$$F = G(Dif, R_{e/h}) \quad (3)$$

where G is a function of $R_{e/h}$ and Dif. $R_{e/h}$ is the operation of either encryption or hashing while Dif is the diffuser. The purpose of Dif is to diffuse the bits of XORed t_l and Kj_1 with s-bit left circular shift operation before $R_{e/h}$.
The amount of shift, s, is determined by

$$s=l*n+n, \quad n=0,1\ldots..log2size(K)-1 \quad (4)$$

Where K and l denote the desired key size and the size of the timer/counter in bits for the microcontroller employed by the sensor node. In essence, both Dif and $R_{e/h}$ in F are mainly utilized to increase the randomness of bits in the key K as subsequent readings from the timer. It is possible to design using $R_{e/h}$ stream ciphers (e.g., RC4), cryptographic hash functions (e.g., MD5), or message digests using block ciphers. In Source based loose synchronization protocol, the TKM module internally adopts 128-bit RC4.

2.3 Crypto Module

The CRYPT module addresses the security part of SOBAS. The CRYPT module obtains the dynamic key from the TKM module and performs the necessary security service. This is also the module where the key from the TKM is verified. If the key value received from the TKM module is not correct then a new key is obtained from the TKM module. This process continues until the correct key is found or the packet is marked as malicious to be discarded in the filtering-forwarding-synch module when all attempts to find the correct key are exhausted within the tick window, (T_w).

The CRYPT module incorporates the RC4 algorithm into its body as the encryption mechanism. The rationale for choosing RC4 is due to its proven lightweight computational energy consumption on sensors. The risk of differential cryptanalysis is eliminated since a new key is used as input for each RC4 block. Nonetheless, it may be still possible for an attacker to discover the key based on the first few bytes of the cipher text. However, an adversary can always brute-force the keys is generated in another module, any desired encryption (e.g., AES, 3DES), authentication, or integrity mechanism (e.g., HMAC, CMAC) can be implemented together or separately depending on the security service desired from the WSN application. Three operational modes exist in the CRYPT module to determine how to forward the incoming packet.
Keys are generated randomly in RC4 algorithm. Keys first set the local time value and that values obtain the matching key find in RC4 algorithm. Before RC4 generation swapping that key then randomly selected the key. To encryption of

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 5, Special Issue 3, March 2016

6th International Conference in Magna on Emerging Engineering Trends 2016 [ICMEET 2016]
On 10th & 11th March, 2016
Organized by
Dept. of ECE, EEE & CSE, Magna College of Engineering, Chennai-600055, India.

each message nodes use the local time information to create the dynamic keys. Keys are randomly selected in RC4 algorithm. The authorized nodes use their local time to intelligently decode the timing key of the source node. So maintain more secured and authenticity. This process continues until the correct key is found or the packet is marked as malicious to be discarded in the filtering-forwarding-synch module.

In the first mode, No-reEnc mode, the original incoming packet is forwarded to the upstream node without any re-encryption whereas in the second mode, Full-reEnc mode, the incoming packet is forwarded to the upstream node after re-encryption with the key associated with the local time at the receiver node. For Full-reEnc mode, the forwarder node uses its current local clock value and Kj_1 value to create a new key when re-encrypting the incoming packet. The advantage of the No-reEnc mode is one encryption computation, hence energy is saved by forwarding the original packet.. Specifically, this case occurs if the time difference between the local times of the source and the faraway node is bigger than the total time covered with T_w*Ø.

This is not an issue for Full-reEnc mode because the forwarder nodes refresh the key used to encrypt the forwarded packet. Finally, the third mode is referred to as Selective-reEnc mode, where packets are selectively re-encrypted over some nodes along the data delivery path while these nodes are also loosely synchronized with the source as in Full-reEnc. This mode is specifically introduced in SOBAS to solve the problem of classifying a healthy incoming packet as malicious (i.e., false-positive) that may occur in the No-reEnc mode. Moreover, in Full-reEnc and Selective-reEnc modes, reencrypting the packets using new keys essentially refresh the keys. Eventually, in all the modes when the sink receives the report along the path, it also goes through the same intelligent key-finding procedure as forwarder nodes. This is not an issue for Full-reEnc mode because the forwarder nodes refresh the key used to encrypt the forwarded packet. This is not an issue for Full-reEnc mode because the forwarder nodes refresh the key used to encrypt the forwarded packet.

The advantage of the No-reEnc mode is one encryption computation; hence energy is saved by forwarding the original packet. Therefore, a source centric synchronization path is established up to the sink. The next time a packet from the same source travels over the same path.

2.4 Filtering-Forwarding-Synch Module

The FFS module filters the incoming packet out of the network if it is classified as a bad packet by the CRYPT module or otherwise forwards it to the upstream nodes. In SOBAS, this module is also responsible for the synchronization process of the forwarder node with the source node along a data delivery path toward the sink with the Full-reEnc or Selective-reEnc modes of operation. At this module, the forwarder node gets the source's local clock value from the CRYPT module and updates its local clock value accordingly. Therefore, a source centric synchronization path is established up to the sink. The next time a packet from the same source travels over the same path, the nodes can put less effort in finding the proper time-based key.

The next time a packet from the same source travels over the same path, the nodes can put less effort in finding the proper time-based key. In the Full-reEnc all the nodes along the path do the aforementioned operations whereas in Selective-reEnc only a certain fraction of the nodes (e.g., every third node) along the path do the operations. Hence, Full-reEnc provides "full path synchronization" while Selective-reEnc provides "partial path synchronization" and end-to-end synchronization along a data delivery path. Eventually, when the sink receives the report along the synchronization path, it will be able to see how much a particular source has diverged from the real clock value by extracting the key associated with the time at the source.

The sink also goes through the same intelligent key-finding procedure as forwarder nodes. Hence, the sink calculates $\Delta_i=t_r-(t_i+\sum_h\Theta)$, with t_r being the real clock and h being the hop length while Δ_i and t_i denoting the time offset from source i and local time at source i, respectively. The synchronization described above synchronizes only one path according to a source. There are more synch paths because there may be more than one source in the deployed region and they may be reporting the same event to increase the accuracy of the reports at the sink.

When the sink receives the report along the synchronization path, it will be able to see how much a particular source has diverged from the real clock value by extracting the key associated with the time at the source. The sink also goes through the same intelligent key-finding procedure as forwarder nodes. Nodes information forwards it to the upstream nodes. Therefore, a source centric synchronization path is established up to the sink.

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 5, Special Issue 3, March 2016

6th International Conference in Magna on Emerging Engineering Trends 2016 [ICMEET 2016]

On 10th & 11th March, 2016

Organized by

Dept. of ECE, EEE & CSE, Magna College of Engineering, Chennai-600055, India.

2.5 Auto Message

Auto message scheme is data can be automatically transfer to desired node. Node location are frequently varied, wireless sensor network get the updated information and set the time that data store into buffer or memory. Then that time data can automatically transfer.

So event detection and sending process take the minimum energy expenditure. So many energy is reduced when using this scheme. WSN applications involving event detection and signal estimation/tracking, and not for guaranteed end-to-end data delivery services..

III. SECURITY AND ENERGY CONSUMPTION ANALYSIS

Analyses security and energy consumption of the SOBAS protocol while under attack. Nodes were located randomly in the deployment region and on average, source nodes were 25-35 hops away from the sink. However, the costs for encryption and decryption operations are computed based on the reported values of the implementation of RC4 on real sensor devices. E_{tx}, E_{rx}, and E_{sens} are the energy consumption of sending, receiving a packet and sensing an event, while E_{enc}, E_{dec}, and E_{mac} are the costs of encryption, decryption, and the message authentication code, respectively. Due to the broadcast nature of the wireless medium used in WSNs, attackers may try to eavesdrop, intercept, or inject false messages. In our attack scenario, the total number of healthy source nodes that collect the event information and send it toward the sink is assumed to be fixed, whereas the number of malicious nodes are increased over time. The malicious sensors are randomly located inside the event collection region. We use en-route filtering to remove the malicious data from wireless sensor networks.

3.1 *Analysis of Outside Attacker*

Source based loose synchronization protocol was designed as a dynamic enroute filtering system because it is an effective method for resource-constrained devices like sensors as the bad data is immediately filtered out from the network before propagating too much, helping save in energy. Hence, SOBAS mainly considers the false injection and eavesdropping of messages from an outside malicious node and the insider attacks are outside the scope. In SOBAS, in order for an outside malicious node to be able to successfully inject a false packet, the malicious node must forge the packet. For the packet length, l

$$l=ID||TYPE||\ E_{Kj}(ID||\ DATA||CTR) \quad (5)$$

where ID is the packet ID, TYPE is the packet type, the CTR is the counter, and E_{Kj} is the encryption of the fields with jth key, the complexity of the packet is 2^l. Hence, the probability of a malicious node correctly forging the packet is

$$P_{forge}=1/2^{packet\ size}=1/2^l \quad (6)$$

The probability of an adversary incorrectly forging the packet and therefore the packet being dropped is

$$p_{drop}=1-p_{forge} \quad (7)$$

Since SOBAS authenticates at every hop, forged packets will always be dropped at the first hop with a probability of Pdrop.

3.2 Packet Fragment Reassembly Module

Fragmentation of IP packets for transmission and the need to reassemble the IP packets at a destination.Maximum size of a frame, set by the Maximum Transmission Unit (MTU) value, and the maximum size of a packet are determined independently.It is usually the case that the packet size can far exceed the MTU size.

If the packet size (data plus IP and other headers) exceeds the allowable frame size (usually set by the transport medium limits), the packet must be fragmented at the sender and split across multiple frames for transmission. Frames are always processed immediately, as they arrive (if error-free), but packet fragments cannot be processed until the whole packet has been reassembled.

IV. CHOICE OF TICK WINDOW

The tick window T_w is available for the receiver node to choose from to decrypt the received packets. The window has upper and lower boundary values. In , the T_w value is basically a function of the tick value (Ø). Assuming that the sensor application sends its data periodically at certain time intervals to the sink, T_w can be computed as

$$T_W=\frac{(\lambda+\rho+\tau+\varphi+\mathcal{E})*\delta}{3600*24*Ø} \quad (8)$$

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 5, Special Issue 3, March 2016

6th International Conference in Magna on Emerging Engineering Trends 2016 [ICMEET 2016]

On 10th & 11th March, 2016

Organized by

Dept. of ECE, EEE & CSE, Magna College of Engineering, Chennai-600055, India.

○

where τ is the transmission time of one packet, τ=l/R with l and R being packet length and rate of the WSN link, respectively; ρ is the propagation delay, ρ=χ/cwith χ and cbeing distance between the sensors and the speed of light inthe medium, respectively; λ is the average period of data inbetween sensed reports sent from a sensor; φ is the packetprocessing time, Ɛ is the physical transmission error; δ is thedaily value of the drift per sensor given a deployment area;and Ø is the desired tick value. Also, χ is taken based on the possible maximum distance to consider the worst case scenarios although nodes may be located closer than the maximum distance.Moreover, the probability that kth trial out of T_w keys is the first success is geometrically distributed with parameterp, where p is l T_w. Hence, the probability that T_w keys are triedin a tick window is

$$P\{K=T_w\}=(1-p)^{T_w-1} * p, \quad T_w=1,2,3 \qquad (9)$$

Each T_w value, its corresponding Ø value is also shown in the pot. The probability of success with a smaller value of T_w is greater, and therefore, less computational effort is required to guess the correct key of compared to when T_w is larger. Hence, the efficacy of the Source based loose synchronization protocol depends on the size of this window because the larger the size of T_w, the more time it takes for a receiver to find the key.

V. CONCLUSION

In this project provide secure loose synchronization and networks send separate synchronization messages and utilize dynamic key-based cryptographic mechanisms to ensure that the clocks of each sensor device are securely globally synchronized. The event-based characteristics of WSN tasks and the resource-limited nature of sensors, and finally focusing on the fact that the transmission cost is significant in WSNs, in this work we relaxed the perfect synchronization condition for WSNs and we tackled the secure time synchronization problem from a completely different perspective. As opposed to current “chatty” perfect synchronization schemes, we focused on minimizing the communication in the network so that some of the energy savings from transmission cost could be utilized for the computation of local security operations.

The receiving nodes use their local time to intelligently decode the timing key of the source node. Thus, the protocol avoids extra overhead of control messages.In this way, SOBAS provides an effective dynamic en-route filtering mechanism, where the malicious data is filtered from the network. With SOBAS, we have been able to achieve our main goal of synchronizing events at the sink as quickly, accurately, and surreptitiously as possible. Thus, energy savings from the reduced transmission is used for the local security computation. Both extensive analytical and simulation results verified the feasibility of the SOBAS scheme. SOBAS is able to provide 7:24 s clock precision given today’s sensor technology,which is much better than other comparable schemes , and is at least twice better in energy efficiency than other schemes.

REFERENCES

[1]G.J. Pottie and W.J. Kaiser, “Wireless Integrated Network Sensors,” Comm. ACM, vol. 43, no. 5, pp. 51-58, 2000.
[2] R.Roman, C. Alcaraz, and J. Lopez, “A Survey of Cryptographic Primitives and Implementations for Hardware-Constrained SensorNetwork Nodes,” Mobile