Efficient Key Management for Big Data Gathering in Dynamic Sensor Networks

Farah I. Kandah, Oliver Nichols, Li Yang
Department of Computer Science and Engineering,
University of Tennessee, Chattanooga, TN 37403

***Abstract*—Recent years have shown a huge growth in technology, which led to an explosive growth in the volume of data which is referred to as "Big Data". Wireless sensors are considered one of the highly anticipated contributor to big data nowadays, therefore, avoiding misleading or forged data gathering in case of sensitive and critical data through secure communication is vital. This can be achieved though cryptography, but due to the limited resources in sensor networks, an efficient security schemes need to be implemented to avoid the resources consumption. Symmetric cryptography is very applicable to sensor networks due to its efficiency, which requires the use of key management for key distribution so that both communicating devices have the same key. To address that in dynamic environments, key distribution needs to be capable of accepting new devices and completing the exchange more efficiently, therefore, in this paper we propose a Centralized Stateful Connection (CSC) that provides efficient key management for dynamic sensor networks. Our scheme will maintain a balance between efficiency and security that is achieved by using public key encryption at the beginning of the node's life in the WSN and later shifts to symmetric encryption for the remainder of the communication.**

Keywords: Key management, Cryptography, Wireless Sensor Networks (WSNs)

I. Introduction

Recent years have shown high tendency towards small, inexpensive, low-power, distributed devices known as sensor nodes. Although these devices have limited processing and storage capabilities, when coordinated with each other, through their wireless communication capabilities, they can form a sensor network with the ability to measure a given physical environment in great details. One of the emerging application that attracted much attention recently is the Internet of Things (IoT) which is capable of generating massive amount of data. At the core of this technology is the sensor network, which has a high potential to improve people's lives and businesses. Wireless sensors are considered one of the highly anticipated contributor to big data nowadays. Although the amount of data generated by a single sensor node is not big enough to be considered as big data, the overall data generated by multiple sensors in a network can produce a significant portion of the big data. The limited resources of sensor devices (slow processor, small memory/storage, and limited power source) attracted much research attention towards this field with a focus on overcoming these limitations to improve the efficiency of WSN devices and the WSN as a whole [1] [2] [9] [10]. In addition to efficiency, security is another issue for constrained devices. To support the performance of such devices, a balance between security and efficiency has to be provided.

Security requirements in WSN is similar to conventional computer networks where confidentiality, availability, integrity, and authenticity must be considered when forming and building these networks. Cryptography is a popular mechanism that can provide various security services for applications and devices, but due to limitations in WSNs capabilities, not all security solutions designed for conventional computer networks can be implemented directly in WSN [5]. Symmetric encryption is considered to be faster and more efficient compared to asymmetric encryption [8]. To use symmetric encryption, two nodes are required to have the same key. This is accomplished through key management, which provides a way to distribute and manage keys between devices. Key distribution can be accomplished through a number of ways. Some of these methods are computationally expensive, which make them not very suitable for sensor networks with constrained memory, energy and processing resources.

To address the aforementioned limitations, we presented a Centralized Stateful Connection (CSC) scheme that provides efficient key management for dynamic sensor networks. Our scheme aims to maintain a balance between efficiency and security that is achieved by using public key encryption at the beginning of the node's life in the WSN and later shif to symmetric encryption for the remainder of the communication.

The remainder of the paper is organized as the following: Section II covers the literature review, followed by our motivations in Section III. In Section IV we define the problem statement of the paper, and presented our proposed scheme in Section V. We illustrate the CSC scheme in Section VI followed by our discussion in Section VII, and finally we conclude the paper in Section VIII.

II. Related Work

Hummen *et al.* in [3] proposed a way to accomplish key exchange, authentication, and authorization in Internet of Things. The authors discussed the use of Datagram Transport Layer Security (DTLS) and how it might be improved in terms of efficiency for constrained devices. DTLS addresses two issues that TLS cannot cope with in lossy environments; (***i***) independent decryption of records and (***ii***) unreliable delivery of messages. One of the issues with DTLS is the use of public key cryptography, which is costly and very expensive.

Lalitha *et al.* in [4] proposed a key management scheme called energy efficient cluster-based key management (EECBKM) in a hierarchal WSN. In their scheme, a cluster head (CH) is chosen based on the capabilities of the node. The cluster head gathers information about all the member nodes and sends the information to the base station (BS). The BS in turn will distribute cluster keys and an Exclusion Basis System (EBS) key set to each CH. Later, the CH distributes keys from the EBS keys set to each node. When a node wants to communicate within the cluster, the node uses its key to talk to the CH. The CH grants a key to the nodes who wish to communicate. In this scheme, the nodes communicate using a key from the CH. For a node to communicate between clusters, the two CHs communicate with each other and each CH will grant a key to its node, later the two nodes will send encrypted messages through their respective CH using the provided key and the CH will deliver the message to the other CH where it will be passed to the destination node.

Seo *et al.* in [7] proposed a certificateless-effective key management (CL-EKM) scheme. In this scheme, public key cryptography was used. In particular, this scheme uses Elliptic Curve Cryptography (ECC) with 160 bit key length. CL-EKM uses four types of keys: a certificateless public/private key, an individual node key, a pairwise key, and a cluster key. The certificateless public/private key is generated by the base station and installed in the node. This key is used to authenticate and generate a pairwise key between two nodes. An individual node key is used to encrypt communication between the node and base station. A pairwise key between two nodes is first generated using each node's certificateless public/private key pairs. After the pairwise key is established, it is used to encrypt the rest of the conversation between the two nodes. The cluster key is used to encrypt broadcast messages within a cluster. The cluster key is distributed to nodes using the pairwise key [7]. Although, the non use of certificates in CL-EKM eliminates computational overhead related to certificate management, but the overhead of the use of public key encryption remains an issue. It is worth to mention that, with the use of certificatless scheme some incompatible issues could occur where nodes might be unable to communicate with systems outside the network that do use certificates.

III. Our Motivations

Providing high security at low cost is not a trivial task for dynamic sensor networks.

Due to the sensor nodes limitations, security cannot be expected to be provided in the same way as that in desktop computers or servers, therefore, efficient key management needs to be utilized in WSN in such a way that will take advantage of the benefits of symmetric encryption.

Based on previous work we observed that the key exchange is accomplished either by the use of public key encryption or pre-shared keys. On one hand, installation requires additional work and is not practical for consumers with little technical knowledge. On the other hand, pre-shared key schemes do not easily support dynamic environments where nodes leave and enter the network. Moreover, pre-shared keys might not be feasible if one of the devices is not under the operator's control. With public key encryption, the fallbacks of pre-shared keys is addressed because pre-configuration can be minimized and configuration is not required within a WSN. Despite all the benefits that come with public key encryption, there still high expense when applying to WSN. Therefore, our focus is on two goals; to support a dynamic environment and to lessen the expense of security in WSN through minimizing the use of public key cryptography which will only be used once to establish the connection with the gateway.

IV. Problem Statement

First in this section, we will describe our system model and notations. Then, formally we will define the research problem we are going to study. In this work the terms sensor and node are interchangeable.

Definition 1. ***Gateway* (G)*: a machine on the WSN that has direct access to the internet and satisfies the conditions for a base station.*** □

Definition 2. ***Base Station* (BS)**: *a machine on the WSN with a reliable power source, a sufficient security, and has more resources than nodes.* □

Definition 3. ***Node (n)***: *a small device on the network that provides a service, typically a sensor. These devices are typically have limited resources without reliable power source.* □

We state the Centralized Stateful Connection (CSC) problem as the following:

Definition 4. ***CSC problem:*** *Given a hierarchal wireless sensor network, where nodes are forming the lower level (leaves) of the network while the gateway/base stations form the upper level of the wireless sensor networks. The Centralized Stateful Connection (CSC) scheme seeks a centralized/dynamic key management scheme while minimizing the network's resource consumption to create a secure dynamic sensor network environment.* □

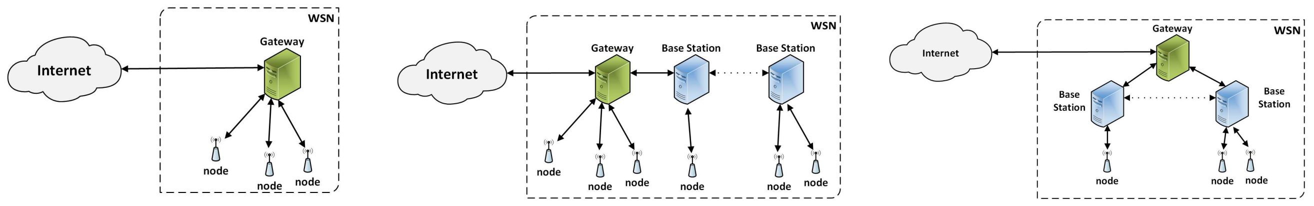
V. Centralized Stateful Connection

In this section we will present our proposed scheme named Centralized Stateful Connection (CSC). We designed our scheme to establish a secure connection between the sensor nodes and the gateway to provide a key exchange mechanism within the WSN. Throughout the explanation of the CSC, gateway and base station will be used interchangeably.

A. Requirements

CSC is designed for hierarchal wireless sensor networks with at least two levels where the nodes are on the lower level (leaves) and the gateways/base stations are on the upper levels for the WSN as demonstrated in Fig. 1. For large WSNs, multiple base stations are allowed. For this scenario, there would be at least one gateway that has direct access to the internet while multiple base stations can form a hierarchal structure similar to Qin *et al.*'s work in [6].

The gateway must have a reliable power source and be hardened against various attacks. Nodes will need to be able to communicate with the gateway. The communication is not



(a) Two-tires (Single Gateway/Base station) (b) Two-tires (Single Gateway/Multiple base stations) (c) Two-tires (Single Gateway/Multiple base stations)

Fig. 1: Wireless sensor network - hierarchal architecture

required to be direct where nodes are allowed to communicate on a mesh network with traffic relaying capabilities.

B. Connection establishment

When a new node (n) enters the WSN, it will first establish a connection with the gateway. The sensor node will send its MAC address (MAC_n) and its public key (PK_n) to the gateway (G). The gateway will use this information to create a cookie (c). Later, the gateway will send its public key (PK_G) to the node and will encrypt a nonce (no) with the node's public key and send it to the node.

$$G \rightarrow [no]_{PK_n} \tag{5.1}$$

To finalize and acknowledge the connection, the node will decrypt and encrypt the nonce (from the gateway), its MAC address, and its public key with the gateway's public key and send it back to the gateway.

$$n \rightarrow [no \oplus MAC_n \oplus PK_n]_{PK_G} \tag{5.2}$$

When the node acknowledges, the gateway will create a connection with the node. Later, the gateway will generate a symmetric key (SK_{Gn}) to secure the connection between the node and the gateway. The gateway will encrypt the symmetric key with the node's public key and send the message back to the node.

$$G \rightarrow [SK_{Gn}]_{PK_n} \tag{5.3}$$

The gateway and the node can now communicate securely within the WSN using symmetric encryption. Note that, each node connection will use a different symmetric key.

The gateway will use a database to store information on each connection. The database fields are shown in Table. I and Table. II. These tables are used for storing information about connections. Note that, uncompleted connections will remain in the cookie table. Each connection contains a connection identifier (ID), node's public key (PK_n), MAC address (MAC_n), symmetric key (SK_{Gn}), state of connection (C_S), and a timestamp (T_n) for the node's last communication. The state of the connection (C_S) displays whether the device is online or offline (not responding on the network). If a node drops from the network and rejoins, the node will again connect with the gateway by sending its MAC address and public key. The gateway will search its database for any previous connections and find the symmetric key used with that node. If the node forgot its symmetric key, the node can request to reestablish the connection. The node and gateway will use public cryptography again only for key exchange.

TABLE I: Connection table

Connection ID (*ID*)	Uniquely identifies each connection
Public Key (PK_n)	Node's Public Key
MAC address (MAC_n)	Node's MAC address
Symmetric Key (SK_{Gn})	Symmetric key used with this node
State (C_S)	Node's state: [online, offline]
Timestamp (T_n)	Last contact with this node

TABLE II: Cookie entry table

Nonce (no)	Random value
Cookie (*c*)	SHA-1 (MAC Address ‖ public key ‖ nonce)

C. Dropping Connections

There might exist some situations where connections need to be dropped. The state and timestamp fields are both used to help address this issue.

The state (C_S) is used for the device's status on the network. When nodes do not respond after a certain period of time, the gateway will send a heartbeat to check if the device is still on the network. When the node does not respond after several heartbeats, the state of the connection changes to offline.

The timestamp (T_n) is the date and time of the node's last communication with the gateway. When a node is offline for a long period of time, the connection is deleted. Deleting old connections will save storage space and database search time on the gateway.

D. Communication and Data transmission

After the node establishes a connection with the gateway, the node can begin communicating with other devices on the same WSN or outside the network. In either case, the gateway will assist with the key exchange to lessen the load on constrained sensor devices. Any key that the gateway creates for nodes should be deleted from storage and memory after distribution to the nodes. Removal of these keys from the gateway mitigates the consequences of a gateway being compromised.

*1) **Communicating inside the network:*** If a node wishes to communicate with another node in the same WSN, the node send a message (M) to the gateway, requesting a connection with the other node. The gateway will verify the connection acceptance by the other node and passes it to the original node. If the response was an accept, the gateway will generate a symmetric key and distribute the key to each node using their personal symmetric keys.

In a larger network where multiple base stations exist such as in Fig. 1(b) and Fig. 1(c), the node would request a connection in the same way to its base station. The base station will communicate with other base stations to locate which station's network has the destination node. With the connection acceptance, the original node's base station will generate and relay the symmetric key for the two nodes.

*2) **Communicating onside the network**:* A node might wish to communicate outside the WSN. To communicate outside the WSN, the node notifies the gateway of the request. The gateway will attempt to create a connection with the requested destination. Depending on the connection, the gateway might, for example, use the Transport Layer Security (TLS) for a secure connection where the gateway would use public key cryptography for authentication and key exchange. When the gateway has the key, the gateway will distribute the key to the node. Later, the node will encrypt the message using the new key and pass it to the gateway. The gateway in turn forwards the message to the destination. Note that, the node is not required to encrypt the message again with the gateway's key because the message is already encrypted.

VI. Illustration

The following sub-sections will further discuss CSC and provide illustrations for better understanding.

A. New nodes

As a node enters the WSN, the node will begin communicating with the gateway to establish a connection. The node will begin by sending a hello message to the gateway. This message will include the node's public key and MAC address. The gateway will acknowledge the hello by providing the gateway's public key. These messages are sent in plain text on the network. The gateway will send an encrypted nonce to the node using the node's public key. The node will acknowledge by returning the nonce using the gateway's public key. The gateway will save the connection and deliver the symmetric key to the node using the node's public key. Finally, the node will acknowledge using the symmetric key. From this point forward, the node will use the unique symmetric key to communicate with the gateway.

B. Communicating within the same network

This example demonstrates key exchange between two nodes within the same WSN that has multiple base stations as depicted in Fig. 2. The source node will begin by requesting to communicate to the destination node. The source node's base station will check its local nodes. If the base station is unable to find the node, it will check with other base stations. The source base station will look for the destination node locally, if not found it will check with the neighboring base station. Upon finding the destination base station, communication request verification will start and the destination node's response will be sent back to the source node. On accept, the source node's base station will create the symmetric key for both nodes to use. It will deliver the key to the source node and the destination base station which it turn will deliver it to the destination node.

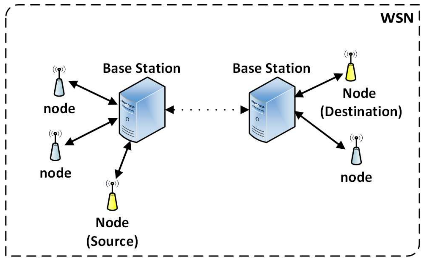


Fig. 2: Communication inside a WSN

C. Communicating outside the same network

To communicate outside the WSN as depicted in Fig. 3, the node will use similar procedures as communicating within the WSN. The node will request to communicate with a machine outside the network. The gateway will use public key cryptography to exchange keys with the outside machine. Certificates can be used to authenticate the outside machine. If client authentication is required, the gateway can send the node's public key. The gateway can affirm that the public key belongs to that node because the node must have the private key to obtain the shared symmetric key. When the gateway and outside machine agree on a symmetric key, the gateway securely passes the key to the node. The node can now communicate to the outside machine via the gateway, which will forward the traffic between the two machines.

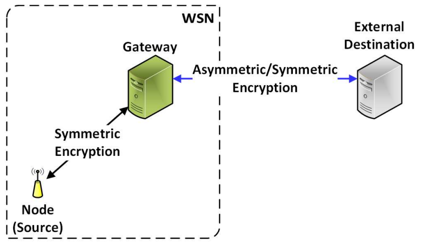


Fig. 3: Communication outside a WSN

VII. Discussion

In this section, we will discuss different aspects of our proposed scheme and its performance. Our discussion will be divided into two parts: the first part will present the performance of our proposed scheme in terms of energy consumption, which is one of the most important aspects when it comes to WSNs with limited resources. Later in the next part we will focus on the security aspect by presenting various possible attacks and discuss how our proposed CSC scheme deals with such attacks.

A. Energy consumption

To illustrate the performance of our scheme, we implemented our solution (denoted by CSC), and compared it with the energy efficient cluster-based key management in [4] (denoted by EECBKM). We considered a wireless networks with n nodes with the same configuration as in [4]. The results shown are the average of five test runs for various scenarios, where ten connections between different pair of nodes in a single cluster are considered. We used the ***energy consumption ratio*** for performance evaluation, which is defined as the overall energy consumed through setting and running up the connections to the total nodes' energy. Note that, the smaller the energy consumption ratio is, the better the scheme.

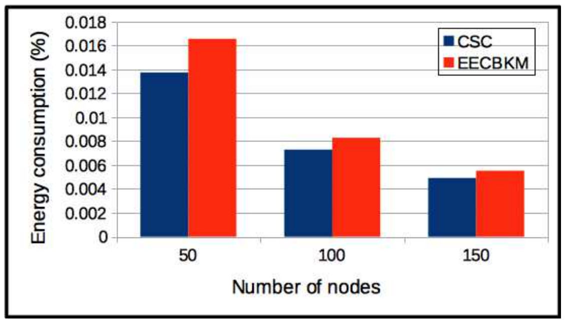


Fig. 4: Energy consumption ratio (10 connections)

It can be seen in Fig. 4 that our CSC scheme outperforms the EECBKM scheme in terms of energy consumption, due to the fact that less number of nodes are being involved in establishing and running up the connections and the reduction of the load on the nodes.

B. Security

In this subsection, we will go over some possible attacks and discuss how our proposed CSC scheme deals with them. This does not constitute as a complete or full listing of attacks, but as a short overview.

1) Discovering Devices and Connection Integrity: It could be possible for an attacker to discover devices on the WSN. This can be done by sending spoofed MAC addresses and public keys to the gateway, requesting for a connection. The gateway will respond differently for nodes that have a connection and those that do not. Even though this attack exists, it would be difficult for an attacker to spoof the correct MAC address and the public key together.

It is also possible for an attacker to spoof the same MAC address as another node and use a different public key. This could possibly drop the legitimate node's connection. To counter this threat, the gateway will create separate entries of different corresponding MAC addresses and public keys. The connection identifier is used to determine which symmetric key to use.

2) Denial of Service: The gateway could be subject to a Denial of Service (DoS) attack. This can be accomplished by sending many connection requests to the gateway, which will eventually fill up the database in the gateway. The current measure taken to prevent this attack is cookies. The gateway has a separate database table for cookies, which contains only two attributes: nonce and cookie. After the gateway creates, encrypts, and sends the nonce to the node, the node is required to encrypt and send the nonce back along with the node's MAC address and public key to the gateway using the gateway's public key. The gateway will look in the cookie table using the nonce as the index. The gateway will then compute a hash to ensure the cookie's match, which finalizes the connection. Connections that are not finalized are more likely to be dropped compared to completed connections.

3) Node Compromising: A node could be easily compromised if physical security measures are not taken. But since nodes do not share common keys with the gateway, compromising a node does not lead to the revealing of other node's connections with the gateway. The severity of a node being compromised depends much on the application of the nodes. If nodes are used to monitor patients in a hospital, the severity of a node compromise is very high. On the other hand, nodes detecting light for city street lights might not have as much impact.

VIII. Conclusion and Future Work

In this work, we presented the CSC scheme which provides a way to securely exchange and manage keys in wireless sensor networks. This solution easily accommodates for a dynamic environment where nodes will enter and leave the network. We showed the performance of our proposed scheme in terms of energy consumption and provided a discussion how our scheme can help the network in overcoming different attacks.

In the future, we are planing to provide more comparisons considering the security and the resource consumption aspects to show the performance of the networks with our proposed scheme.

References

[1] Sangeeta Bhattacharya. *Achieving Application Quality of Service in Resource-constrained Wireless Sensor Networks*. PhD thesis, St. Louis, MO, USA, 2008. AAI3332063.

[2] Wan-Hee Cho, Jiho Kim, and Ohyoung Song. An efficient resource management protocol for handling small resource in wireless sensor networks. *International Journal of Distributed Sensor Networks*, page 9, 2013.

[3] R. Hummen, H. Shafagh, S. Raza, T. Voig, and K. Wehrle. Delegation-based authentication and authorization for the ip-based internet of things. In *Sensing, Communication, and Networking (SECON), 2014 Eleventh Annual IEEE International Conference on*, pages 284–292, June 2014.

[4] T. Lalitha, R. Saravana Kumar, and R. Hamsaveni. Efficient key management and authentication scheme for wireless sensor networks. *Am. J. Applied Sci.*, 11(6):969–977, 2014.

[5] OWASP. Guide to cryptography. https://www.owasp.org/index.php/Guide_to_Cryptography, February 2012.

[6] Zhongyuan Qin, Xinshuai Zhang, Kerong Feng, Qunfang Zhang, and Jie Huang. An efficient identity-based key management scheme for wireless sensor networks using the bloom filter. *Sensors*, 14(10):17937, 2014.

[7] S. H. Seo, J. Won, S. Sultana, and E. Bertino. Effective key management in dynamic wireless sensor networks. *IEEE Transactions on Information Forensics and Security*, 10(2):371–383, Feb 2015.

[8] J. Stretch. Symmetric encryption, asymmetric encryption, and hashing. http://packetlife.net/blog/2010/nov/23/symmetric-asymmetric-encryption-hashing/, 2010.

[9] D. Takaishi, H. Nishiyama, N. Kato, and R. Miura. Toward energy efficient big data gathering in densely distributed sensor networks. *IEEE Transactions on Emerging Topics in Computing*, 2(3):388–397, Sept 2014.

[10] A. Selcuk Uluagac, Christopher P. Lee, Raheem A. Beyah, and John A. Copeland. *Wireless Algorithms, Systems, and Applications: Third International Conference, WASA 2008, Dallas, TX, USA, October 26-28, 2008. Proceedings*, chapter Designing Secure Protocols for Wireless Sensor Networks, pages 503–514. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.