Indonesian Journal of Electrical Engineering and Computer Science
Vol. 2, No. 2, May 2016, pp. 431 ~ 451
DOI: 10.11591/ijeecs.v2.i2.pp431-451

A Lightweight Symmetric Cryptography Scheme for Identifying Compromised Node in WSN

Yassine Maleh*[1], Abdellah Ezzati[2]
[1]Departement of Mathematics and Computer Sciences, LAVETE Laboratory, Hassan 1st University, 577 Casablanca Road, FST de Settat, Km 3, Morocco
[2]LAVETE Laboratory, Faculty of Science and Technology
*Corresponding author, e-mail: y.maleh@uhp.ac.ma

Abstract

Wireless Sensor Network (WSN) is consisting of independent and distributed sensors to monitor physical or environmental conditions, such as temperature, sound, pressure, etc. The most crucial and fundamental challenge facing WSN is security. Due to minimum capacity in-term of memory cost, processing and physical accessibility to sensors devices the security attacks are problematic. They are mostly deployed in open area, which expose them to different kinds of attacks. In this paper, we present an illustration of different attacks and vulnerabilities in WSN. Then we proposed a new lightweight cryptography algorithm for identifying compromised node in WSN called Leap Enhanced. Our evaluations on TOSSIM give a precise and detailed idea of the extra cost of consumption of resources needed to ensure the high level of expected security compared to other cryptography schemes in literature.

Keywords*: Wireless Sensor Network, Key management, LEAP, Cryptography, TinyOS*

1. Introduction

Inexpensiveness, energy efficiency, and consistency in performance is the need of the day for electronic communication. This has led to the advancement of wireless technologies and micro-electro-mechanical systems (MEMS). Which in turn made it is possible to make low power tiny devices that run autonomously. Which evolve a new class of distributed networking named wireless sensor networks (WSNs).

Today we find this kind of network in a wide range of potential applications, including security and surveillance, control, actuation and maintenance of complex systems and fine-grain monitoring of indoor and outdoor environments. The majority of these applications are deployed to monitor an area and to have a reaction when they register a critical event. The data does not need to be confidential in areas such as capturing indoor and outdoor environmental events. However, the confidentiality of data can be essential in other applications, such as for the security of a territory in military [1] [2].

The different characteristics of sensor networks such as (energy limited, low power calculation, use of radio waves, etc.) represents a challenge for researchers community. One of the key objectives is the energy consumption of nodes that aims to extend the life of such network. The necessary protocols for the functioning of WSN such as medium access protocols, routing and security must take into account the constraints of network nodes while saving as much as possible their energy consumption. The purpose of the proposed new protocols for WSN is to make a compromise between the quality of service QoS provided by these solutions and the respect of the limitations imposed by network nodes.

Therefore, it is necessary to use effective mechanisms to protect this type of networks. However, it is well known that the encryption systems represent a first line of defense against all types of attacks. Furthermore, cryptographic techniques must be designed to detect the execution of the most dangerous attacks. In addition, these techniques must be small to fit the limited resources of the WSN.

Sensor nodes are limited in terms of computing, memory and energy capacities, these limitations affect negatively the functioning of the special smart techniques that provide the required security. Key management protocols provide safe paths in sensor networks. The idea is before opening the safe route; nodes must share some identification and authentication

Received February 21, 2015; Revised April 23, 2016; Accepted April 30, 2016

information. These security keys also need a management system that takes responsibility for the creation, maintenance and security of key distribution [2].

In this work we propose a lightweight security scheme for identifying compromised node in WSN. Our proposed scheme is based on the famous LEAP protocol, proposed by Zhou. This protocol is designed to secure network communications with the primary goal is to immediately limit the impact of the security of a compromised node on the network. The protocol supports four types of keys that are exchanged to meet the different requirements of the security of WSN:

- Individual Key: Shared with the base station
- Pairwise Key: Shared with another sensor node
- Cluster Key: Shared with several neighboring nodes
- Global Key: Shared by all nodes in the network.

The critical assumption that leap+ has considered is that within Tmin a node cannot be compromised. This hypothesis seems convenient, but only under ideal conditions, it is possible that Tmin be greater than the one assumed. To address this limitation, we propose two models, the first use a periodic verification "Periodic Check" to detect the compromised node. The second model execute a sequence number in each node and compared them after the pairwise key establishment step with the information stored in the base station BS, which then makes the decision on whether to delete the shared key. Our evaluation have been implemented on TOSSIM simulator, and it give a precise and detailed idea of the extra cost of the consumption of resources required to ensure the high expected level of security.

This work will be organized as well. In the second part, we identify the various attacks and vulnerabilities in WSN. We present then a state of the art in terms of key management algorithms for WSN. Later, we will discuss and analysis our proposed scheme. We evaluate Leap Enhanced in terms of connectivity, memory complexity, scalability, and resistance to attacks, and then we compare it to other schemes proposed in the literature. This topic ends with a general conclusion and a set of perspectives.

2. Different Kind of Attacks and Vulnerabilities in Wireless Sensor Networks

The different characteristics of wireless sensor networks (energy limited, low-power computing, use of radio waves, etc...) expose them to many security threats. The different characteristics of wireless sensor networks (energy limited, low-power computing, use of radio waves, etc...) expose them to many security threats. Without proper security measurements sensor network will be exposed to multiple attacks. The two general categories of attacks that are possible on a wireless network generally are Active and Passive Attacks listening and monitoring the communication channel by unauthorized attackers is considered as passive attack. These attacks make it possible to retrieve data from the network but do not influence over it behavior. [4] [5] [6].

2.1. Routing Attacks in Sensor Networks

2.1.1. Spoofed, Altered, or Replayed Routing Information

The most direct attack against a routing protocol is to target the routing information exchanged between nodes. By spoofing, altering, or replaying routing information, adversaries may be able to create routing loops, attract or repel network traffic, extend or shorten source routes, generate false error messages, and partition the network, increase end-to-end latency and so on.

2.1.2. Selective Forwarding

Only certain packets can be drop selectively by a malicious node. This is effective specially if is combined with an attack which gather much traffic through the node. It is assumed in sensor networks that nodes sincerely forward and receive messages. In situation where some of the compromised node refused to forward packet the neighbors may start using another route. [7] [8]

2.1.3. Spoofed, Replayed and Altered Routing Information

Ad hoc routing that is unprotected is vulnerable to these kinds of attacks because every node act as a router and hence can directly affect routing information. Like:

- Extend or shorten service routes
- Generate false error messages
- Create routing loops
- Increase end-to-end latency

2.1.4. Sybil Attacks

This kind of attack targets fault tolerant schemes like multi path routing, distributed storage and topology maintenance. A node duplicate itself and present in the multiple locations. In the network a single node, present multiple identities in a Sybil attack as shown in figure 1. To prevent Sybil attack authentication and encryption techniques can be used. [7]

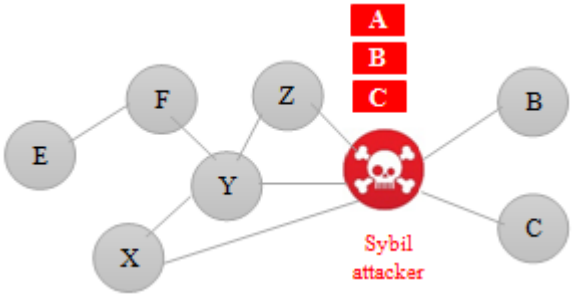

Figure 1. Sybil attack

2.1.5. Sinkhole Attack

Sinkhole attack is to attract traffic to a specific node. The main goal of the attacker here is to attract nearly all the traffic from a particular area through a compromised node. This can be done by making the compromised node the most attractive to the neighbor nodes.

2.1.6. Wormhole Attack

In this kind of attack an attacker records packet at one location, tunnel them to another location in the network and then retransmit them into the network as shown in figure 2.

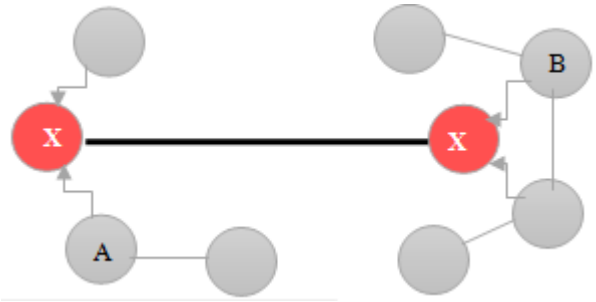

Figure 2. Wormhole attack

2.1.7. Hello Flood Attack

In hello flood attack, the attacker uses hello packets as a weapon to convince the sensors in the network. Attacker sends a routing protocol hello packet from one node to another with more energy. An attacker with a high processing power and transmission range sends Hello packets to a number of sensor node that are isolated in a large area within the network. The sensors thus get convinced that the adversary is their neighbor. As a result, the victim nodes try to go through the attacker, as they know that it is their neighbor while sending the information to the base station (BS) as shown in figure 3.

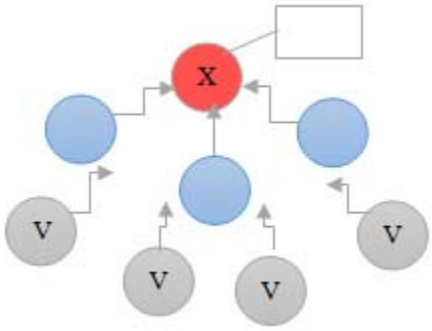

Figure 3. Hello flood attack

2.2. Denial of Service Attacks

In this kind of attack, the objective of the malicious adversary is to avoid the provisioning of service. The most common and basic dos attacks can be:

2.2.1. Power Exhaustion Attack

An attacker impose a complex task to a sensor node in order to use its battery life. As we are aware of the fact sensor are low power devices with limited supply of energy. In addition, sensor nodes are limited with computational capabilities, so this attack can slow down the reaction time.

2.2.2. Jamming Attack

This is the primary physical layer DoS attack against WSN. In this kind of attack, the attacker emits radio frequency signals that do not follow an underlying MAC protocol, which lead to a problem that none of the member of the network in the affected area will be able to send or receive any packet. In other words, the adversary tires to transmit signals to the receiving antenna at the same frequency band or sub band as the transmitter, which causes radio interference. The attacker needs high energy to disrupt continuously the network [9].

2.2.3. Node Subversion

In this kind of attack, the whole sensor network may be compromised because the capture node may reveal cryptographic keys.

2.2.4. Node Outage

In this kind of attack a situation, occur where the node stops its function. Hence, the sensor network protocol should be robust enough to mitigate the effects of node outages by providing an alternate route.

2.2.5. Physical Attacks

As we have mentioned repeatedly that sensor network operate in outdoor environment mostly which make highly susceptible to physical attack. This code be physical node destruction. This attack is different from others in the sens that here a node can be permanently damaged or destroyed.

2.2.6. Node Replication Attacks

In this kind of Attack a node is added to the existing sensor network by copying the node ID of an existing sensor node. This can severely disrupt the sensor network performance. The packet can be misrouted or even corrupted, which can lead to disconnected network or false sensor reading etc. Once the attacker get physical access to the entire network, it can copy the keys to the replicated sensor nodes.

2.2.7. Passive Information Gathering

If the sensor network is not encrypted, an adversary having powerful resources can collect information from it. If the intruder is:

- Appropriate power receiver
- Well designed antenna

Then it can easily pick off the data stream. If the message interception contains the physical location of sensor nodes then it allows attacker to locate the node and destroy them. The adversary can observe the specific content of a message that includes timestamps, message IDs and other fields. Strong encryption techniques need to be used in order to prevent passive information gathering.

3. Related Works

Wireless sensor networks are highly vulnerable against attacks, it is very important to adopt some mechanisms that can protect the network from all kinds of attacks. It must be ensure that the system is protected before, during and after any kind of attack. The most important tools that ensure security and its services are the security primitives [10]. Those primitives are symmetric key encryption (SKE), public key cryptography (PKC) and hash functions [11]. SKE and Hash functions are the primitives that can be called the building blocks which offer a basic protection of the information flow, because these assure the confidentiality and integrity of the channel. Public key cryptography assures protection from the participation of external entities and eliminates the problem of a malicious insider, which try to use more than one identity. PKC assures this by allowing authentication of the peers involved in the information exchange. Based on the primitives it is possible to create a better network services. It is also equally important to have a key management system for constructing a secure key infrastructure.

3.1. Methods and Protocols Classification

Most methods based on symmetric, asymmetric or hybrid systems solve the problem of key establishment through a predistribution phase. The predistribution of encryption keys in a WSN is the fact of storing these keys in the memory nodes before deployment. In literature, we find several classifications of cryptographic key management systems, such as papers in [12] - [14].

Some classifications methods are based on key sharing between two nodes (Pairwise) or more nodes (Group-wise), and others rely on exploiting the probabilities, combinatory analysis, etc. We chose to make a classification, which includes all key management and distribution models into two large families. The first family contains the asymmetrical schemes and the second includes the symmetrical schemes. Figure 4 illustrates this classification. In the following, we will detail the main models in literature.

3.2. Symmetric Schemes

The schemes in this category use symmetric mechanisms in order to establish a common key between two nodes in a WSN. This is accomplished in three steps:

- Key predistribution: keys stored in memory before deployment constitute the key ring of node. If there is a common key between two nodes, they can create a secure connection between them.
- Shared-key discovery: After deploying the communication protocol is responsible for discovering the common key between two neighboring nodes.
- Path-key establishment: if there is no common key between two nodes wishing to communicate, there must then find a secure path between them. This path goes through a set of nodes that already contains secure links. Once the path established, the two nodes can use it to secure communication.

We present in the following symmetrical schemes according to the decomposition of figure 4.

3.2.1. SPINS

SPINS is a suite of security building blocks proposed by Perig and several other authors in [15]. It is optimized for resource constrained environments and wireless communication. SPINS has two secure building blocks: SNEP and µTESLA. SNEP uses a shared counter between the two communicating parties and applies the counter in calculating encryption and a message authentication code (MAC) to provides data confidentiality, semantic security, data integrity, two-party data authentication, replay protection, and weak message freshness. What's

more, the protocol also has low communication overhead, for the counter state is kept at each end point and the protocol only adds 8 bytes per message. For applications requiring strong freshness, the sender creates a random nonce (an unpredictable 64-bit value) and includes it in the request message to the receiver. The receiver generates the response message and includes the nonce in the MAC computation. µTESLA provides authenticated broadcast for severely resource-constrained environments. µTESLA constructs authenticated broadcast from symmetric primitives, but introduces asymmetry with delayed key disclosure and one-way function key chains. SPINS realizes an authenticated routing application and a security two-party key agreement with SNEP and µTESLA separately with low storage, calculation and communication consumption. However, SPINS still have some underlying problems as follows:

- It doesn't consider the possibility of DOS attack;
- Due to use the pairwise key pre-distribution scheme I the security routing protocol, SPINS rely on the base station excessively;
- SPINS does not consider the update of communication key. There must be practical key update mechanism to realize forward security;
- SPINS cannot solve the problem of hidden channel leak and compromise node.

3.2.2. LEAP

LEAP (Localized Encryption and Authentication Protocol) is a key management protocol for sensor networks that is designed to support in-network processing with the prime goal at the same time to restrict the security impact of a node, which is compromised to the immediate network neighborhood. The idea of leap+ was motivated after having this interesting observation that different types of messages that are exchanged between sensor nodes have different requirements of security. This observation gives the conclusion that a single keying mechanism is not suitable for meeting these different security requirements [16] [17] [18]. For each node leap support the establishment of four types of keys:

- Individual key: Shared with the base station;
- Pairwise key: Shared with another sensor node;
- Cluster Key: Shared with multiple neighboring nodes;
- Global key: Shared by all nodes in the network.

The packets that each node exchanged in a sensor network can be classified into several categories, which is based on different criteria for example:

- Control packets vs Data packets
- Broadcast Packets Vs Unicast Packets
- Queries or commands Vs Sensor readings and so on.

The security requirement for each packet is different it depends on the category it falls in. Almost all type of packets requires authentication while confidentiality is only for some types of packets. Here it is mentioned that single keying mechanism is not appropriate for all the secure communication that are needed in sensor networks.

3.2.3. Tinysec

Karlof et al. [19] propose the TinySec Protocol, the first full implementation of a secure architecture at the data link layer for WSN. This implementation supports two security options: a message authentication with data encryption (TinySec-EA) and authentication of messages without data encryption (TinySec-Auth). As SPINS, TinySec uses standard cryptographic algorithms to ensure privacy and message integrity check. The authors of Tinysec find that Skipjack algorithm [20] is more suitable for WSN than RC5 (algorithm used by SPINS). Indeed, evaluations of TinySec have shown that RC5 needs a pre-key calculation using 104 bytes of RAM. TinySec uses the CBC encryption mode (Cipher Block Chaining) instead of the CTR (used by SPINS). Indeed, the CTR will provide for more packet encryption the same random numbers. These numbers are used primarily in the production of the encryption keys sequences; their repetition can weaken the security level of this solution and subsequently allowing adversaries to discover the content of messages. TinySec is an implementation rather than a key distribution proposal, he comes to complete a key distribution method suited to the expanded network. Two nodes need two-shared symmetric key to communicate. The first used to encrypt messages and the second for calculating MAC (code) messages.

3.3. Asymmetric Schemes

The schemes in this category use the mechanisms of asymmetric systems in order to establish a common key between two nodes or a group of nodes of a WSN.

3.3.1. Micro-PKI

Munivel et al [21] propose a method for WSN called micro-PKI (Public Key Infrastructure Micro), a simplified version of conventional PKI. The base station has a public key and another private. The public key is used by the network nodes to authenticate the base station, and the private key is used by the base station to decrypt data sent from the nodes. Before deployment, the public key of the base station is stored in all nodes. The authors include in their scheme two types of authentication (Handshake). The first type of authentication occurs between a network node and the base station. The node generates a symmetric session key and encrypts it with the public key of the base station. To ensure the integrity of messages exchanged, the authors propose to integrate with each message a MAC (code) using the same encryption key of the message. For new nodes who wish to join the network, they simply store in these nodes, the public key of the base station before deployment.

3.3.2. Tiny PK

Watro et al. [22] proposed a method called TinyPK based on the use of public keys and the principle of Diffie-Hellman to establish a secret key between two nodes in a WSN. TinyPK uses a trusted authority to sign the public keys of nodes. The CA key is predistributed to all nodes before deployment so they can check key neighbors after deployment. The choice of the RSA algorithm for encryption involves a great consumption of time and energy of the nodes. Thus, the basic operations can take dozen seconds, which will reduce the network lifetime as well as affect reactivity.

3.3.3. PKKE & CBKE

The PKKE and CBKE protocols proposed by Zigbee using the identity of nodes in their method of key establishment. The goal is to use these identities to create a single shared key between each pair of nodes in a network. However, the creation of the shared key is performed with interactions between the two nodes. It means, methods require sending and receiving multiple messages on both sides before the creation of the key. To save power nodes that want to share a secret and those intermediate nodes, several methods have been proposed to remove these interactions. These methods are known in the field of cryptography as the ID-NIKDS [23] (Identity-Based Non-Interactive Key Distribution Scheme).

3.3.4. C4W

Jing et al. [24] proposed a method called C4W based on the use of the identity of nodes to calculate public keys. The nodes themselves are able to calculate the public keys of other nodes using their identities. What could replace the role of a certificate. Before deployment, the nodes and the base station are loaded with their own keys (private / public key ECC) and public information on the network nodes. The C4W method uses the principle of Diffie-Hellman key exchange to create a single shared key between two nodes without using certificates

3.4. Synthesis

We have studied different types of distribution and key establishment in WSN. The diagrams SPINS and LEAP use master keys in the key establishment. This reduces the storage of keys in the memory of the nodes. However, the resistance to attacks is low. Given that the master key can be compromised at any time, the keys established after the deployment by using this key can be compromised also. By adopting a symmetric system, they are the most suitable and among the most rapid in terms of calculation. Note that the symmetric diagrams are costly in operations (if they exist) of renewal and revocation of keys since they use secret keys in order to exchange other secret keys. The problem is simpler in the asymmetric diagrams since the public keys do not need to be secret.

The scheme Chan et al., Representing probabilistic diagrams shows that consumes low power and do not require much computing capacity. However, large sizes key rings stored in the memory nodes before deployment makes this scheme one of the most expensive symmetrical schemes in terms of memory occupation. It cannot resist to attacks of type physical

nodes captures. While PIKE scheme provides better connectivity between network nodes, but it shows low performance in term of scalability.

We can see that the schema TinyPBC is the most suitable of asymmetric patterns. It is resistant to most known attacks in the RCSF. The fact of using the coupling in order to establish a unique key shared between two nodes has helped reduce the need for large storage capacity in memory. In addition, the creation of this key is performed without interaction between the nodes, which saves the time of calculation, and the energy consumed due to these interactions. The diagrams using the principle of certificates and PKI remain the most expensive in calculation and in energy consumption. The comparison between the diagrams symmetrical and asymmetrical may differ depending on the desired level of security in the network. We note in the table of comparison that the symmetric diagrams can be chosen for their timeliness and the asymmetric diagrams for their resistance against attacks.

4. LEAP Enhanced

In the previous chapters, it has been tried to give a general description of some of the state of the art algorithms available in the literature. It can be concluded that sensor network nodes are mostly deployed in unattended adversarial environment for example battlefield. Therefore, it is extremely important for the applications of many sensor networks to have a security mechanism, which provide authentication and confidentiality. The unique issue that needs to be considered in sensor network before selecting a key sharing approach is its impact on the effectiveness of in-network processing. The proposed algorithm support in-network processing and provide security properties similar to those provided by pairwise key sharing schemes. Similar to leap+ and other protocol the proposed solution is also based on the observation that different types of messages exchanged between sensor nodes have different security requirements, which lead us to the conclusion that a single keying mechanism is not suitable for meeting these different security requirements. Like leap+ the proposed algorithm support the establishment of four different types of keys:

- **Individual key:** Shared with the base station
- **Pairwise key:** Shared with another sensor node
- **Cluster key:** With multiple neighboring nodes it is shared
- **Global key:** Shared by all nodes in the network

4.1. Assumption In-Term of Network and Security

The following important assumption has been made while studying and designing the protocol

- A static sensor network where nodes are not mobile
- The base station work as a controller
- The power supplied to the base station is supplied with long-lasting power
- All nodes are equal in computational and communication capabilities
- Every node has enough space to store hundred of bytes of keying materials
- Nodes installation can be done both either through aerial scattering physical installation.
- In advance The immediate neighboring nodes of any sensor node are not known
- All the information a node holds becomes known to the attacker If it is compromised
- Attacks of the physical layer and media access control layer are not considered

4.2. Ket Management

4.2.1. Individual Key

To have secure communication between the node and the base station this key is used. Every node in the network have its own individual key. Individual key is also important in the sense that it can be used to compute the message authentication code if the message is to be verified by the base station. This can be also used to send alert to the base station if there is any abnormal behavior observed. Base station can use individual key to encrypt any sensitive information such as keying material or special instructions to individual node. It is important to mention that individual key is preloaded into the network before deployment [6]. The individual key is generated as follows:

$$IKu = fKm\ (u) \quad (1)$$

Where *IKu* is the individual key, *f* is a pseudo-random function, *Km* is the master key, and *u* is any node for which we want to find individual key.

4.2.2. Pairwise Keys Establishment

The most important step is to have pairwise key between nodes. It is very important from the security point of view as if the key is compromised its effect is localized. The pairwise key is shared between one-hop neighbors. A sensor node communicates with its immediate neighbor through pairwise key. The important assumption made to establish pair wise keys are:

- Node doesn't know its neighbor pairwise key before deployment. Pairwise key is created after deployment
- The nodes of the network are stationary nodes
- A node that is added to the network will discover most of its neighbors at the time of deployment.

The pairwise is generated by following these steps:

4.2.2.1 Key Predistribution

An initial key generated by the controller is loaded to each node. Each node then derive its master key as:

$$Ku = fKin\ (u) \quad (2)$$

Where *Ku* is the master key generated by the node *u*, *f* is a pseudo-random function, *Kin* is the initial key, and *u* is any node for which we want to derive the master key.

4.2.2.2 Neighbor Discovery

Every node try to find its neighbor by broadcasting a HELLO message. This Hello message contains id of the node. Also a timer is started which fires after time Tmin. This node then wait for any node v which respond tho this HELLO with an ack message having the id of the node v. Ack from neighbor is authenticated using master key kv. The master key is derived as:

$$Kv = fKin\ (v) \quad (3)$$

Where *Kv* is the master key of node *v*, f is a pseudo-random function, *kin* is the initial key, and *v* is any node that wants to find its master key.

4.2.2.3. Pairwise Key Establishment

Any two nodes between the network let say node *u* and v compute the pairwise key as:

$$Kuv = fKv\ (u) \quad (4)$$

Where *kuv* is the pairwise key between node *u* and *v*, *f* is a pseudo-random function, *kv* is the master key of node, and *u* is the node id of any node *u*.

4.2.2.4. Key Erasure

When the timer expires after Tmin, node *u* erases *Kin* and all the master keys of its neighbor, which was computed during the neighbor discovery phase.

4.2.3. Cluster Keys Establishment

Cluster key is established between a node and all its neighbors. Using cluster key a node encrypt broadcast message. To establish the cluster key any node u generate a random key and then encrypt this random key with the pairwise key already generated. Then the generated cluster key is transmitted to each neighbors.

4.2.4. Global Key Establishment

A key shared by the base station and every node is the global key. It is important and is used when the base station (controller) want to generate a confidential message.

4.3. The Proposed Models for Identifying Compromised Node

The critical assumption that leap+ has considered is that within Tmin a node cannot be compromised. This idea seems practical but only in an extreme ideal condition. There is possibility that Tmin would in reality be greater than the one assumed. As an example if nodes are dropped and scattered from airplanes, the scattered nodes may arrive in different parts of the network at different times even if dropped simultaneously and hence will need some time to setup the network and exchange pairwise key. Taking the advantage of this an adversary may observe a node and obtains the key and if the global key is compromised, the whole network becomes at risk. Since this is a very serious threat to security different algorithms has been studied and proposed a model that detect the compromised node and take necessary action to delete the compromised node from the network [25] [26] [27] [28] [29]. Some of the algorithms available in literature to detect compromise nodes are:

- Detecting Compromised Nodes in Wireless Sensor Network by Rick Mckenzie, Min song, Mary Mathews, sachin shetty,
- A Framework for identifying Compromised Nodes in sensor Network by Qing Zhang, Ting Yu, Peng Ning
- Malicious Node Detection in wireless sensor Networks using by Idris M.Atakli, Hongbing Hu, Yu chen
- Sensor Node Compromise Detection, The location Perspective by Hui song and liang xie

4.3.1. 1st Proposed Model for Identifying Compromised Node

A time called Tp has been chosen and run it iteratively after a certain period of time to check if there is any node compromised. It can be run directly after pairwise keys are exchanged and the nodes start communication so that to be sure, that none of the node is compromised and the network has successfully exchanged all the keys securely. The steps to do the tasks would be [30].

Step 1: Periodic check for node detection

An iterative-periodic PERIODIC-CHECK (Tp) routine has been run on every node to check if it is under attack or not. Duration of "Tp" is the tradeoff of the "Complexity" and "Attacker Threat". In the former case Tp could be increased to so as to minimize over all network complexity in terms of packet exchange. In the later case, Attacker Threat, Tp could be minimized to have more frequent checks on Node compensation.

Now suppose a node is compromised exactly after the CHECK period just finished, that is "Tp+t" In this case rather than to wait for the next period (t=2Tp suppose), the node itself send a Help broadcast message. The base station receive the help broadcast message and take the necessary action explained in the next step.

Step 2: Suspension of the compromised node

Upon Reception of the HELP from the node under attack, the Base Station broadcasts an ALERT message. The alert message contains id of the sender of the HELP (i.e the node under attack, which has been named as "in_danger_id"). Nodes receiving ALERT message, inspects for the "in_danger_id" and then compares with the the list of IDs present in its NEIGHBOR LIST. If a match is there, node deletes the pairwise key already established with that node. If no-match is found, node just keeps this id for a specified amount of time, and whenever initiates PAIR-WISE key exchange process, compares this key in order not to establish any pair-wise key with a node that is infected. In Figure 5, we present the proposed model for the detection of compromised node.

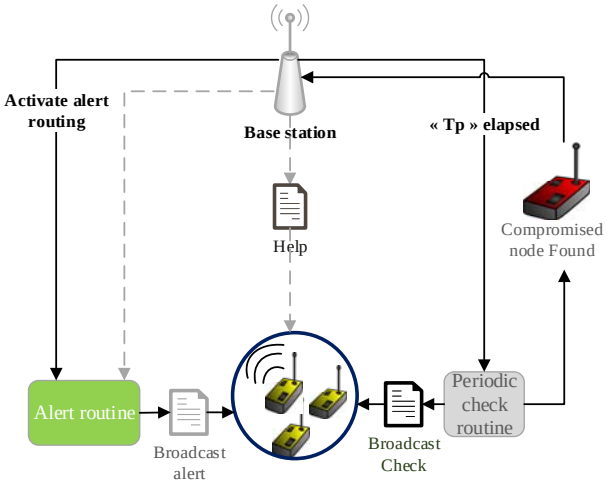

Figure 5. First Model for detection compromised node

4.3.2. 2nd Model for Identifying Compromised Node

The first model is suitable for applications where safety is the main concern and it represents a compromise in terms of energy and complexity. Therefore, we propose a very simple and straight forward model, where the complexity has been reduced to maximum and the global key security is ensured up-to a greater extent. In this model, a sequence number has been added to every node before deployment. This sequence number or nod specific number is used to find if an adversary has taken the advantage of the key establishment time and have added its own node to get information of the network. After the pairwise keys are established the Generator or the base station (who have already the information of every node deployed) send a request to the network and ask for its sequence number, every node send its sequence number to the base station. Using this information to the previously stored information if the result is positive it stay silent. If a compromised node is detected it send a broadcast message informing all the nodes about the compromised node. Figure 6 shows the second model proposed for the detection of compromised node.

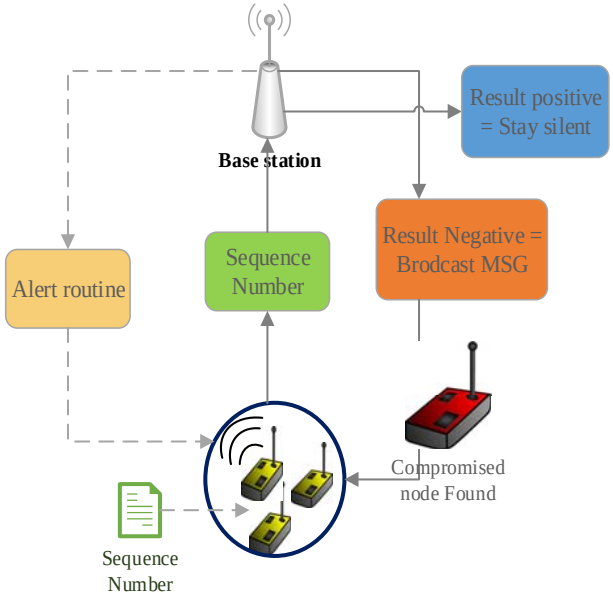

Figure 6. Second Model for detection compromised node

5. Performance and Security Analysis

TOSSIM is used to simulate the Entire TinyOS applications. The goal is to achieve accurate and scalable simulation of TinyOS application. TOSSIM is a TinyOS library, which work by replacing components with simulation implementations [31] [32]. The key requirements for a TinyOS simulator are:

- Bridging

The simulator must work as bridge between the algorithm and implementation. It must provide developer to verify and test the code, which then will run on real hardware.

- Completeness

Completeness in-term of covering maximum possible system interactions and accurately capturing behavior at a wide range of levels.

- Fidelity

The simulator must have the capability to capture the behavior of the network at a fine grain. It is important to check the timing interactions on a mote and between motes both for evaluation and testing.

- Scalability

The simulator must have the capability of handling large networks of thousands of nodes in a wide range of configurations.

5.1. Security Aanalysis

5.1.1. Analysis in Terms of Performance

The pairwise key establishment scheme implemented has the following computational overhead. The two nodes that want to establish the pairwise key has to verify a message authentication code MAC from every neighbors and evaluate a pseudo-random function to create pairwise key. The ACK message has two fields one for a node ID and one for a MAC. Hello message only includes node ID. The storage space required is only for one key Kin. Hence, it can be easily concluded that the computational, communication and storage overhead for establishing pairwise key for our scheme is small.

5.1.2. Analysis in Terms of Security

This scheme is very efficient in-term of security thanks to the pairwise key established between two nodes that even if a key is compromised its effect is localized and the whole network is saved from being compromised. The scheme has been analyzed in the defense against various types of attacks. As its survival was discussed. When there is an attack, and a node is compromised, the opponent takes advantage of this and can launch attacks using compromised nodes. If this compromise node is detected, the rekeying model can effectively revoke attacked the node from the network.

5.1.3. Key Connectivity

Connectivity is defined as the probability that a node can establish a secure link with all of its neighbors. The deterministic key management protocols such as the case of our model and LEAP are based on the initial key generation which provides a full connectivity (100%) with a low cost in terms of memory storage space, unlike probabilistic protocols that require a higher cost in terms of storage memory, and their connectivity depend on the size of keyring preloaded into nodes. The probability that two neighboring nodes X and Y share a secret depends on the size of subsets of shared secrets and according to the network size. The probability of establishing a secure link depends on the probability of sharing a secret. LEAP Enhanced is based on the pre-deployment of the same initial key to all nodes. This mechanism offers the simplicity of implementation and provides complete network connectivity.

5.1.4. Memory Complexity

In our scheme like LEAP [5], a node needs to keep four types of keys. If a node has D neighbors, it needs to store an individual key, D pairwise keys, D cluster keys and a global key. In a sensor network packet transmission rate is generally low. For example, the readings may be generated and transmitted periodically, and the routing information may be exchanged less frequently. So, a node could store a key ring to a reasonable length. L is the number of keys to a stored node to its chain. Thus, the total number of keys that stores a node:

L + D + 2D + 1 + 1
1 ⟵ Individual Key, 1 ⟵ Group Key, D ⟵ Cluster Keys
2d ⟵ Pairwise Key, L ⟵ F Function

Although the memory space is a scarce resource for the current generation of sensor nodes (4 KB of RAM in a Berkeley Mica Mote), storage is not a problem in our system. For example, when d = 20 and L = 20, a node 82 stores keys (A total of 656 bytes when the key size is 8 bytes). Overall, we conclude that the proposed model is scalable and efficient in communication and storage.

5.1.5. The Resistance and Defense Against Various Attacks

LEAP assumes that the initialization phases and key installation run in a maximum time Tmin and an attacker take at least a time T> Tmin to compromise or capture a node and retrieve information contained in its memory. In addition the mechanism proposed by Zhou et al [16] based on the revocation of the node and the suppression of Kinit key to the memory of all nodes and regeneration of symmetric keys between each pair of nodes, our model improves security by the implementation of two approaches to the detection of the compromise discussed earlier node. Control routing information is authenticated by the local broadcast authentication scheme that prevents most external attacks. Possible attacks that an opponent can throw in the hope of creating routing loops, retrieve network traffic or generate error messages are :

- Spoof
- Alter
- Replay routing information
- Selective forwarding Attack
- HELLO Flood Attack
- Node cloning Attack

This scheme cannot prevent the opponent to launch such attacks but it can counteract or minimize their consequences. This localization effect also helps to detect these types of attacks.

- The Alter attack: It can also be detected as the sending node can hear the message being modified during its transmission phase through compromise node. It is also worth mentioning that if a node is compromised and detect the key regeneration system can effectively revoke the network node (see section 4).
- The Hello Flood attack: An adversary may attempt to send a Hello message to each node with a high transmission power to convince all neighboring nodes. Here Hello flood attack will not be successful beyond the neighborhood discovery phase because each node only accepts packages from neighbors who are authenticated. The same for the node cloning attack cannot also go beyond the neighbor discovery phase.
- Sybil attack: In the algorithm, a MAC ID of the node, its level, and the identifier of his father is calculated to authenticate the sender and recipient. Therefore, a node cannot play a role of other nodes.
- The Node capture attack: When a node is captured, it does not affect its neighbors. Indeed, after a node is captured, what an attacker can do? Since it has the shared key with the base station, it may send false information to the base station. The latter can have a mechanism to check issuer's nodes behavior. The attacker also has access to the key of sons of the victim that enables him later to send messages to these useless sons to consume their energy and their cause battery depletion.
- Our model prevents an opponent to launch the wormhole or sinkhole attack because the node knows all his neighbors after the Neighbor Discovery step. Since an attacker cannot derive the key from the initial key or master key initial.
- The resilience against node capture: or resistance against node capture, this metric measures how all the network is compromised when a node is compromised, and the influence of this node on the network security. In our scheme unlike the LEAP protocol, which assumes that Tmin in the node cannot be compromised, we set up two mechanisms to counter the effect of the compromise node.

5.2. Experimentation

5.2.1. Pairwise Key Generation Time Analysis

- Performed with a difference of two nodes ten times repeatedly

As the time available for the generation of pairwise key is short therefore the time for the successful generation of the pairwise key with different nodes model has been analyzed. The main interest was to check if the density of network has any effect in the successful generation of pairwise key with in the available time. Interestingly it has been noticed that the algorithm successfully generate pairwise key on different node model within the available time. It has been also noticed that each pair of neighbor successfully generate the same pairwise key that is uniquely identified to the specific pair of nodes. The experiment has been performed with a difference of two, five and ten nodes repeatedly ten times each. Almost the same result has been noticed for each experimental model with successful generation of the pairwise key.

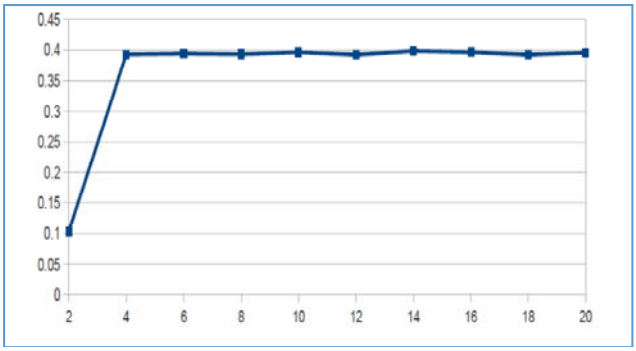

Figure 7. Pairwise key generation time analysis with different nodes

- Performed with a difference of five nodes ten times repeatedly

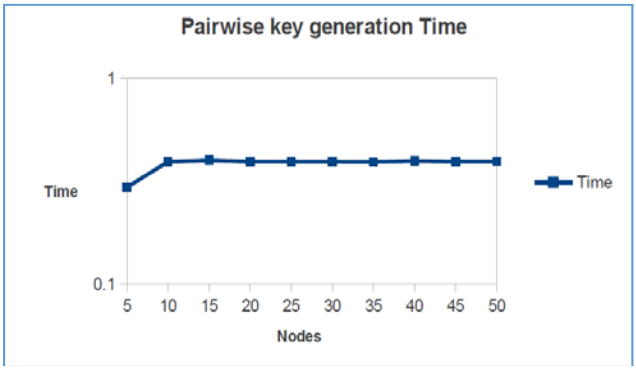

Figure 8. Pairwise key generation time analysis

- *Performed with a difference of 10 nodes ten times repeatedly*

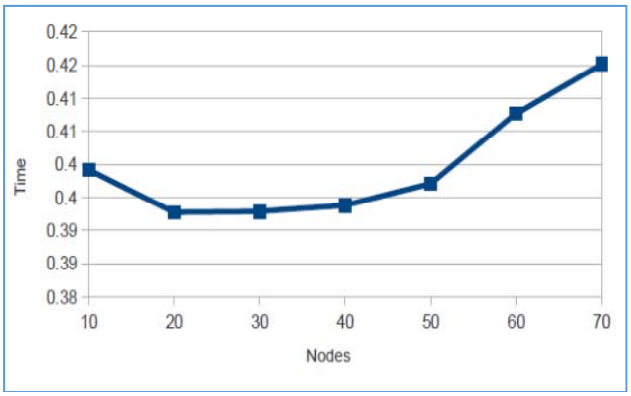

Figure 9. Pairwise key generation time analysis

5.2.2. Individual Key Generation Time Analysis

The most important task was to check the successful completion of pairwise key generation. Apart from that the time for generation of individual keys for different node models has also been analyzed. It has been confirmed that each pair of node successfully generate a unique individual key.

- Performed with a sample of ten Nodes

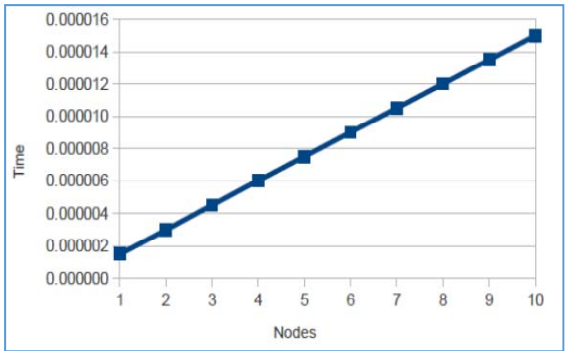

Figure 10. Individual key generation Time analysis

- Performed with a sample to 20 Nodes

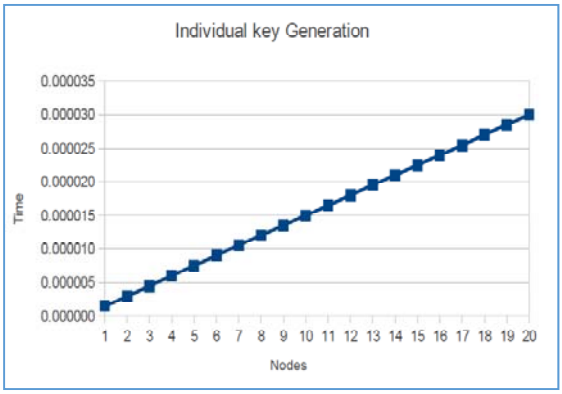

Figure 11. Individual key generation Time analysis

We can observe that our model take a short time to generate key establishment. However, the key distribution takes a significant time when the number of nodes increase. Several factors can influence the distribution key time, for example, network topology, network density…

5.2.3. Scalability

To evaluate the scalability of Leap Enhanced, we conducted various experiments on WSNs by varying the number of nodes from 10 to 100, we calculated the maximum number of messages that can be received by a node depending on the number of sensor nodes in the network. Figure 8 summarizes these results.

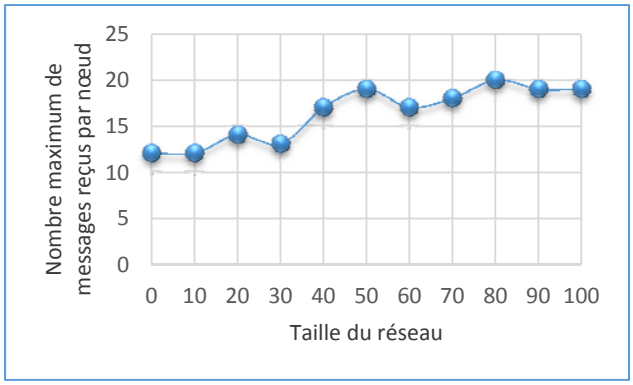

Figure 12. Maximum number of message received by a node Vs network size

We can observe that our model does not generate additional costs when the number of nodes in the network increases. The number of messages received by nodes remains stable with respect to the network density.

5.2.4. Energy Consumption

We use PowerTossim simulator to evaluate the average power consumption of a sensor node in the neighbor discovery and installation phases session keys. This energy is calculated on the basis of instructions executed for cryptographic operations (MAC calculation by the issuer, MAC verification by the receiver, calculating individual key computed by the key-pair) and radio operations (transmission and reception of neighbor discovery messages, exchange and MAC Hello messages). Figure 9 illustrates the variation of the energy consumed by the number of nodes of the network nodes in the presence of attackers.

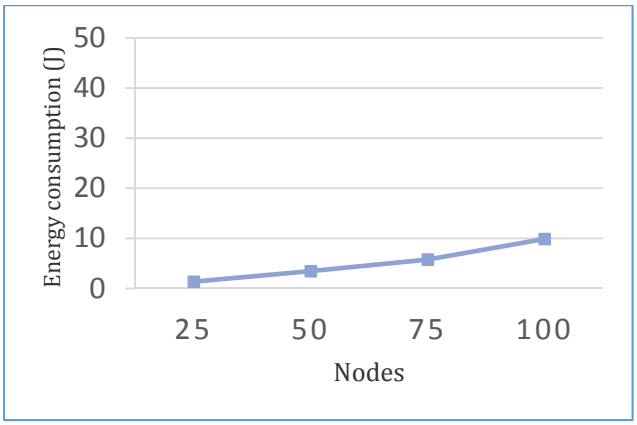

Figure 13. Energy Consumption

In Figure 9, we see that the amount of energy expended increases with the growth of nodes and network size. This increase in energy consumption is mainly due to increasing number of packets exchanged for setting, calculation and verification of encryption keys by the sensors nodes.

5.2.5. Scalability

To evaluate the scalability of Leap Enhanced, we conducted various experiments on WSNs by varying the number of nodes from 10 to 100, we calculated the maximum number of messages that can be received by a node depending on the number of sensor nodes in the network. Figure 8 summarizes these results.

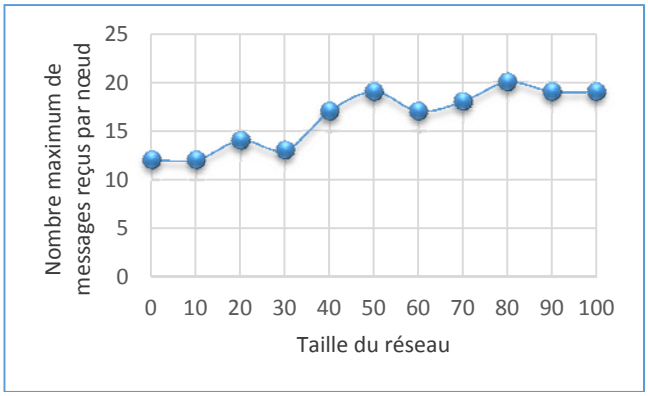

Figure 14. Maximum number of message received by a node Vs network size

We can observe that our model does not generate additional costs when the number of nodes in the network increases. The number of messages received by nodes remains stable with respect to the network density.

5.3. Comparison and Discussion

5.3.1 Methods of Analysis and Comparison

In this section, we evaluate the performance of our solution with some existing schemes in the literature. Several criteria are supported in order to compare the different key management methods. We present in Figure 10 criteria and measures to achieve this assessment.

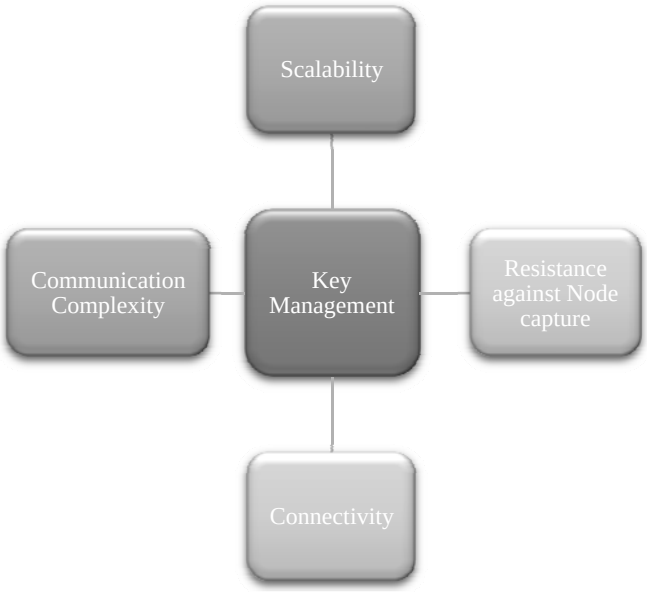

Figure 15. The evaluation criteria for key management methods in WSN

- The complexity of communication: The proposed key management method should take into consideration that the nodes have been deployed to collect information. They need their memory space to store their data and their embedded energy to ensure their application role. We identify this metric with the following three values:

1. Low: the protocol does not require much resource for memory storage and power consumption,
2. Midium : The protocol requires an average consumption in terms of energy and memory,
3. High : The protocol is greedy for power consumption and memory storage.

- Connectivity: the probability that two nodes (or more) share a key.
- Resilience against the node capture: this metric measures the impact of a compromised node on the security of the rest of the network. We quantify this metric with the following three values:

1. Good resistance: the compromise node only affects its neighbors (local influence)
2. Low resistance: the compromise node affects its neighbors and also some non-neighboring nodes,
3. Very low resilience: if the node of a compromise led to compromise the entire network.

- Scalability: this metric Protocol flexible measurement with the network size. In other words. Scalability is a measure very important to consider when designing the algorithms proposed, particularly for sensor networks. To quantify the scalability, we use the following values:

1. Good: The protocol does not lead to additional costs when the number of nodes in the network increases,
2. Fair: the protocol induced reasonable cost when the number of nodes increases,
3. Limited: the cost of the protocol depending on the number of nodes.

Table 1. Comparison with other schemes

Schemes	Criteria				
	Scalability	Connectivity	Resilience against node capture	Communication Complexity	
				Memory Usage	Energy Consumption
Leap Enhanced	Good	100%	Good	Low	Low
TinyPK [19]	Medium	The probability that two nodes share a key P_1	Very low	Midium	High
SPINS [15]	Medium	100%	Good	Low	Midium
TinySec [16]	Medium	The probability that two nodes share a key P_2 w P_1	Low	Midium	High
LEAP [5]	Good	100%	Low	Low	Low
PKKE [20]	Medium	The probability that two nodes share a key P_1	Low	High	High

5.3.2 Comparison with Other Schemes

In Table 1, we compare our model with the existing methods in the literature based on the criteria already discussed in Figure 10 and in our paper [30] and the work of the authors [33] [34]. We note that our model shows the most effective resistance to the effects of the node captures. LEAP Enhanced is equivalent to LEAP in terms of scalability. This equivalence is the fact that both schemes use symmetric keys for moving to the scale more easily. We also observe that, in terms of connectivity, this scheme is equivalent to several symmetric systems that because of the use of NeighbDisc phase after deployment. Any node can establish a link with its neighbor by asking his public key to the base station. Our solution has a low communication complexity compared to other solutions, with connectivity equivalent to 1 (no concept of probability). Having a low complexity means that the proposed solution is lightweight.

Comparison developed in this part concludes that our method has a high level of security against attacks compared to those described in our review of the literature. The results showed that our model have a low storage cost, and the connectivity is 100%. Our data encryption is symmetric; the time consumed in the cost and energy is equivalent to other methods such as symmetric spins and LEAP. The solution that we have proposed has full connectivity, with a small generation of the number and size of exchanged packets.

6. Conclusion

Security is one of the major challenges in wireless sensor networks due to their areas of application. They can be implemented in a very critical system such as hospitals, airports, military applications, alarms system, control of the environment, smart homes and traffic monitoring. The main objective of this work is to address the issue of security in sensor networks. Sensor networks are vulnerable against external and internal attacks due to their unique characteristics; they are limited in terms of computing power, communication and memory. Conventional security mechanisms are not suitable for the sensor network because of constraints discussed prior. Thus, the study of the key management systems in WSN led us to conclude that the use of shared symmetric key systems is most suitable for this type of network. An important observation for any type of key management system is that a single mechanism of encryption is not suitable to meet the different security requirements. In this work, different encryption algorithms for WSN have been studied in depth, in order to propose a lightweight key management system called LEAP Enhanced. The results show that our model shows a good performance in terms of connectivity, memory resources management and resistance against attacks. The rapid growth of sensor networks in different areas especially those criticisms drew the attention of many researchers to work on the safety part. In future work we will try to implement our algorithm to adapt for 6LoWPAN networks in Internet of Things Context IoT.

References

[1] Y Maleh, A Ezzati, Y Qasmaoui and M Mbida. "A Global Hybrid Intrusion Detection System for Wireless Sensor Networks". *Elsevier Procedia Computer Science*. 2015; 52: 1047–1052.

[2] Y Maleh, A Ezzati. "A review of security attacks and intrusion detection schemes in wireless sensor network". *International Journal of Wireless & Mobile Networks (IJWMN)*. 2013; 5(6).

[3] J Lopez, R Roman and C Alcaraz. "Analysis of Security Threats, Requirements, Technologies and Standards in Wireless Sensor Networks". *Springer Lecture Notes in Computer Science*. 2009; 5705: 289-338.

[4] C Karlof, D Wagner. "Secure routing in wireless sensor networks: attacksand countermeasures". *Ad Hoc Networks*. 2003; 1: 293–315.

[5] Z Benenson, Peter M Cholewinski, Felix C Freiling. "Vulnerabilities and Attacks in Wireless Sensor Networks". Laboratory for Dependable Distributed Systems, University of Mannheim, 68131 Mannheim, Germany.

[6] Mayank Saraogi. "*Security in wireless sensor networks*". Proc. ACM Symp. Embedded Networked Sensor Systems, ACM Press. 2004, doi: 10.1.1.105.592.

[7] Sarmad Ullah khan. "Key management in wireless sensor networks, IP-Based sensor networks, content centric networks". 2013

[8] Carlos Aleixandre tudo. "Clustering algorithms for wireless sensor networks and security threats". Goteborg, Sweden. 2010.

[9] AR Alazemi. "Defending WSNs against jamming attacks". *American Journal of Networks and Communications*. 2013: 28-39.

[10] A Huang. "Security primitives for ultra-low power sensor nodes in wireless sensor networks". Faculty of Engineering, the Built Environment and Information Technology, University of Pretoria. 2005.

[11] G Gaubatz, J Kaps, B Sunar. "*Public Key Cryptography in Sensor Networks—Revisited*". Department of Electrical & Computer Engineering Worcester Polytechnic Institute, 100 Institute Road, Worcester, MA 01609, U.S.A.

[12] AM Hegland, E Winjum, SF Mjolsnes, C Rong, O Kure, et P Spilling. "A survey of key management in ad hoc networks". *IEEE Communications Surveys & Tutorials*. 2006; 8(3): 48–66.

[13] SA Camtepe and B Yener. "Key Distribution Mechanisms for Wireless Sensor Networks: a Survey". 2005.

[14] S Ruj, A Nayak, et I. Stojmenovic. "Key Predistribution in Wireless Sensor Networks When Sensors Are Within Communication Range". In *Theoretical Aspects of Distributed Computing in Sensor Networks*, S Nikoletseas and JDP Rolim, Éd. Berlin, Heidelberg. 2011: 787‑832.

[15] A Perrig, R Szewczyk, JD Tygar, V Wen, et DE Culler. "SPINS: security protocols for sensor networks". *Wirel. Netw*. 2002; 8(5): 521–534.

[16] S ZHU, S SETIA, S JAJODIA. "LEAP+: Efficient Security Mechanisms for Large-Scale Distributed Sensor Networks". *ACM Transactions on Sensor Networks*. 2006; 2(4): 500–528.

[17] C Celozzi, F Gandino and M Rebaudengo. "Improving Key Negotiation in Transitory Master Key Schemes for Wireless Sensor Networks". Politecnico di Torino.

[18] Chae Hoon Lim. "LEAP++: A Robust Key Establishment Scheme for Wireless Sensor Networks 2008 IEEE DOI 10.1109/ICDCS.Workshops. 2008; 93

[19] C Karlof, N Sastry, and D Wagner. "*TinySec: a link layer security architecture for wireless sensor networks*". In Proceedings of the 2nd international conference on Embedded networked sensor systems, New York, NY, USA. 2004: 162–175.

[20] EF Brickell. "The SKIPJACK Algorithm". 1993; 28: 1‑7.

[21] E Munivel et GM Ajit. "Efficient Public Key Infrastructure Implementation in Wireless Sensor Networks". in *International Conference on Wireless Communication and Sensor Computing.* 2010: 1‑6.

[22] R Watro, D Kong, S Cuti, C Gardiner, C Lynn, et P Kruus. "*TinyPK: securing sensor networks with public key technology*". in Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks, New York, NY, USA. 2004: 59–64.

[23] R Sakai, K Ohgishi, and M Kasahara. "Cryptosystems based on pairing". in Symposium on Cryptography and Information Security (SCIS'00), Japan. 2000: 26–28.

[24] Q.Jing, J Hu, and Z Chen. "C4W: An Energy Efficient Public Key Cryptosystem for Large-Scale Wireless Sensor Networks", in 2006 *IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS)*. 2006: 827‑832.

[25] H Wen, J Luo, L Zhou. "Lightweight and effective detection scheme for node clone attack in wireless sensor networks". IET Wirel. Sens. Syst. 2011; 1(3): 137–143.

[26] H Choi, S Zhu, Thomas F La Porta. "*SET: Detecting node clones in Sensor Networks"*. Department of Computer Science and Engineering the Pennsylvania State University.

[27] Q Zhang, T Yu, P Ning. "A Framework for Identifying Compromised Nodes in Sensor Networks". North Carolina State University.

[28] Yi Tao Wang, R Bagrodia. “ComSen: A Detection System for Identifying Compromised Nodes in Wireless Sensor Networks”. SECURWARE 2012: The Sixth International Conference on Emerging Security Information, Systems and Technologies, 2012.
[29] Idris M Atakli, H Hu, Yu Chen, Wei-Shinn Ku, Z Su. “Malicious Node Detection in Wireless Sensor Networks using Weighted Trust Evaluation”. *Simulation of Systems Security (SSSS’08)*, Ottawa, Canada. 2008; 14 –17: 27.
[30] Y Maleh and A Ezzati. “An Efficient Key Establishment Protocol for Wireless Sensor Networks’’. *The International Symposium on Ubiquitous Networking, LNEE*. 2015: 273-281.
[31] David Gay, P Levis, D Culler, E Brewer. “nesC 1.1 Language Reference Manual”, May 2003
[32] http://tinyos.stanford.edu/tinyos-wiki/index.php/TOSSIM.
[33] J Jang, T Kwon and J Song. "A Time-Based Key Management Protocol for Wireless Sensor Networks". Third International Conference, ISPEC 2007, Springer LNCS, Hong Kong, China. 2007: 314-328
[34] M Messai, M Alouiat, H Seba. "Tree based protocol for key management in wireless sensor networks". *Springer EURASIP Journal on Wireless Communications and Networking.* 2010.