Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 7.056

IJCSMC, Vol. 9, Issue. 10, October 2020, pg.45 – 63

Enhancing the Security of Spin Framework by Combining Min AES with Geoencryption

Usman Abdullahi Adam[1]; Mitul Patel[2]
[1] uthmanmantissa@gmail.com; [2] mitul.patel@ppsu.ac.in
Department of Computer Science, P P Savani University, Surat, India

DOI: 10.47760/IJCSMC.2020.v09i10.007

ABSTRACT: Wireless sensor networks are contemporary technologies that are used for various purposes in different fields including military and security monitoring, home security, industrial processes/activities monitoring among others. The sensors used in the Wireless Sensor Networks has the capability of sensing or monitoring an environment, measure or capture data and send it wirelessly to a control unit for further processing. Such captured data needs to be secured using cryptography to prevent unauthorized interception and modification. Here we have attempted to enhance the security of SPIN (Sensor Protocol for Information via Negotiation) framework by combining the secure cryptographic algorithm MinAES with geo-encryption. The geo-encryption mechanisms are used to ensure delivering the data to the right and authorized entities at a known location only. This paper presents a strategy to implement a Hybrid Security Algorithm (MinAES + geoEncryption) to provide a secure and reliable means of accessing and delivering data between wireless network components.

KEYWORDS: Wireless Sensor Networks, Geo-encryption, SPIN, Min AES.

1. Introduction Wireless Sensor Networks (WSN)

A Wireless Sensor Network (WSN) is a type of network that consist of large distributed network of devices that are dispersed, self-directed and low powered devices known as sensor nodes (Also called motes). These sensor nodes are spatially dispersed, lightweight, battery-operated and embedded devices that are networked to work together in other to collect, process, share and deliver data among sensor nodes and the users of such networks. Also the sensors have restricted computing and processing capabilities [21] [Fig 1].

Motes are the small computers, which work collectively to form the networks. Motes are energy-efficient, the multi-functional wireless device [21]. The needs for motes in industrial applications are numerous. A collection of motes captures information from the environment to render particular application objectives. They make connections and communicate with each other in different configurations to get maximum performance and efficiency. Motes communicate with each other through transceivers. In WSN the number of sensor nodes can be in the order of hundreds or even thousands. Contemporarily, a wireless network is the most popular service utilized in industrial and commercial applications, because of its technological advancement in a processor, communication, and usage of low power embedded computing devices [21].

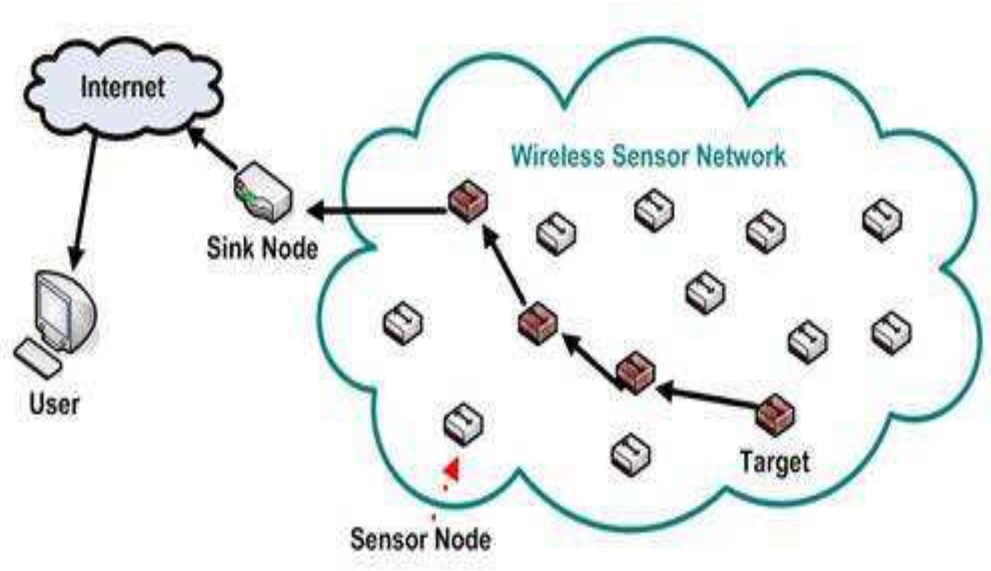


Fig 1: Wireless sensor networks (WSNs) [24]

In contemporary real-time applications, the sensor nodes are providing different tasks including neighbor node discovery, smart sensing, data storage, data processing, data harmonization, target tracking, control and monitoring, node localization, synchronization, as well as efficient communication between sensor nodes and base station [21].

Wireless Sensor Networks (WSNs) are of great benefits due to their low-cost, small-scale factor, smart sensor nodes. Moreover, not only that sensor nodes can be employed in cumbersome and dangerous areas of interest, for monitoring or controlling the region, but they can also be deployed to automate mundane tasks [22].

Early sensory devices were expensive and limited by the computational and communicational capabilities of current smart sensor nodes, which now have the capabilities to sense, process, store, and forward data, all powered by a battery. Myriad applications exist that leverage WSNs as low-cost solutions for observing the habitat and environment, from military and civilian surveillance and target detection and tracking applications to precision like farming and agriculture, patient monitoring in health care, residential applications like energy management, for safety and efficiency in vehicular networks to outer space explorations. The diversity of the applications of WSNs imposes varying design, implementation, and performance requirements on the WSNs [22].

2. Security in Wireless Sensor Network

Owing to the nature of sensitive and valuable information delivered by the sensor networks, they are prone and vulnerable to many attacks and data breaches [6]. Moreover, WSNs are easily compromised by attackers due to wireless communications use of broadcast transmission medium and their lack of tamper resistance. Therefore, an attacker can eavesdrop on all traffic, inject malicious packets, replay older messages, or compromise a sensor node. Generally, sensor nodes have most worried about two major security obstacles, which are privacy-preserving and node authentication. By privacy it means that data confidentiality is achieved under the security mechanism, and hence it allows network communications between sensor nodes and the control station to take place securely. Also, a well-structured authentication mechanism can ensure that no unauthorized node can fraudulently participate and get sensitive information from WSNs. As a result, several schemes have been proposed to secure communications in WSNs [6].

In addition to the characteristics mentioned above, security threats and requirements are also critical for a variety of sensor network applications. In recent years, several security issues in WSNs have been proposed [6]. The following are some security threats and requirements in WSNs [6].

- **Passive attacks:** In passive attacks (such as eavesdropping attacks), the attacker can un-intrusively monitor on the communication channel between two communicating nodes to collect and discover valuable information without hindering the communication.
- **Active attacks**: active attacks (such as node replication attacks, Sybil attacks, wormhole attacks, and compromised node attacks) can either be: external attacks and internal attacks. In the former attacks (such as Sybil attacks and wormhole attacks), a node does not belong to a sensor network and it can first eavesdrop on packets sent or received by normal participating nodes for the malicious purpose of compromising, interfering, guessing, or spamming, and then add invalid packets to disrupt or alter the network functionalities. While in the latter attack, the compromise is caused by using a sensor node that is part of the network to perform the malicious act.
- **Sybil attacks:** a sensor node can illegitimately claim multiple IDs by either directly forging false IDs, or else impersonating legal IDs. This harmful attack may lead to serious threats to distributed storage, routing algorithm, and data aggregation.
- **Wormhole attacks**: In this attack the malicious node may be located within a transmission range of legitimate nodes while legitimate nodes are sometimes not within the transmission range of each other. Thus, the malicious node can tunnel control traffic between legitimate nodes and nonexistent links that are controlled by the malicious node. Finally, the malicious node can drop a tunneled packet or carry out attacks on routing protocols.
- **Internal attacks:** (such as node replication attacks and node compromised attacks) are usually caused by compromised members who belong to the sensor network in question, and hence internal attacks are more difficult to safeguard against than external attacks.
- **Node replication attacks:** when a sensor node is compromised by attackers, they can directly place many replicas of this compromised node at different areas within the networks.

Thus, attackers may use these compromised nodes to subvert the network functionalities, for example by injecting false sense data.

- **Compromised attacks**: due to the lack of tamper resistance in sensor nodes, attackers may compromise a sensor node and use it to establish communication channels with non-compromised sensors to launch other more serious attacks within the sensor network.

In line with the aforementioned description of the security threats in WSN, we can deduce that a secure sensor network needs the following requirements.

- **Node authentication**: For this requirement, a deployed sensor node proves its validity to its neighboring sensors and the manager node. Thus, an invalid outsider would be unable to send compromised data into the networks and the manager node can confirm that received sensed data has come from a valid sensor node, not from malicious outsiders. This also implies that a sensor node joined in WSNs has been authenticated and it has the right to access the sensor network.
- **Location awareness:** The damage cannot be spread from the victimized area to the entire network by security attacks even if the sensor node is compromised. A secure communication scheme must limit the damage’s scope caused by the intruders; the mechanism of location awareness is used for this purpose.
- **Key establishment:** For the sensor-to-sensor key establishment, a shared key is established by two communication nodes to protect communications. Thus, all sensed data transmitted between participants could be verified and protected even if an attacker eavesdrops on the communications between nodes or injects illegal sensed data into networks, this requirement still provides an adequate level of security.
- **Confidentiality:** Path-key establishment in every session must be secure against malicious intruders even if those attackers collect transmission packets.
- **Perfect forward secrecy**: In a two-party path-key establishment, a scheme is said to have perfect forward secrecy if revealing of the secret key to an intruder cannot help him/her derive the session keys of past sessions.
- **Key revocation**: When the back-end system or the manager node decides to terminate a sensor utilizing a task, or when a sensor is lost, the sensor must not be allowed to make use of the credential which stores to connect to networks.
- **Re-keying:** By introducing a re-keying mechanism, a manager node can conveniently update a sensor's credentials without the intervention of the back-end system to reduce the communication interactions and management burden on that back-end system.

3. Encryption Techniques in WSNs

Compared to classic computer networks, wireless sensor networks show many unique features. Therefore, security methods that can be used in conventional networks cannot be used in wireless sensor networks [1, 7]. In sensitive wireless sensor networks applications such as monitoring enemy lines or monitoring border zones, security protocols that provide confidential data transmission from the sensor nodes to the base station must be used.

However, low processor and radio capacities of sensor nodes do not allow the implementation of traditional security protocols in sensor networks [1, 8]. To make sensor networks practical, many improvements are currently being made in software and hardware. However, the basis of every work done is the reduction of the energy consumption per unit and hence the extension of the life of the sensor nodes. The main reason for this is that replacing or refilling sensor nodes and used energy sources in the working environment is often impossible and costly [1, 9].

Security methods designed for traditional networks and widely used in many applications today cannot be implemented directly in wireless sensor networks because sensor nodes have limited energy resources, insufficient memory capacities, and limited processing capabilities. Because of this, the security methods developed for wireless sensor networks take into account safety, energy, memory usage, and latency. A security method in which all resources are actively used will be ideal for sensor networks [1, 10]. In sensor networks, encryption algorithms are used to provide data confidentiality. However, it should not be forgotten that sensor networks have limited hardware resources and that the first objective is energy efficiency while providing requirements. Otherwise, a protocol that consumes a lot of energy, even if it provides all security measures, would be useless for sensor networks [1, 11].

The following are the cryptographic framework used in WSNs:

3.1.Symmetric Cryptographic Frameworks: these are cryptographic frameworks which are based on single shared key both for encryption and decryption. It includes frameworks like:

3.1.1. SPINS: this is optimized for resource-constrained environments and wireless communication. It based on two secure building blocks [2, 13]:

- **SNEP (Secure Network Encryption Protocol)**: SNEP provides data confidentiality, two-party data authentication, and data freshness with less communication cost.
- **µTESLA (The 'micro' version of Timed, Efficient Streaming Loss–tolerant Authentication Protocol):** it provides authenticated broadcast for severely resource-constrained environments. The scheme is resilient to node capture and possible to revoke key. But it is not scalable and the base station becomes the target of attacks. It does not provide a solution for denial of service (DoS) attacks when the malicious node keeps sending the request to negotiate a session key because one adversary can easily trigger a REPLAY attack and exhaust the energy in the sensor nodes[2, 13].

3.1.2. Localized Encryption and Authentication Protocol (LEAP): the protocol was based on the principle that different types of messages exchanged between sensor nodes have different security requirements; a single keying mechanism is not suitable for meeting these different security requirements. LEAP had analyzed under various attack models and it comes out to be very effective in defending against many sophisticated attacks such as HELLO Flood attack, Sybil attack, and Wormhole attack [2, 14].

3.1.3. TinySec: It is the first fully-implemented protocol for link-layer cryptography in sensor networks. The implementation of TinySec is incorporated into the official TinyOS release. It includes some of the trade-offs between performance, transparency, and cryptographic security and a design is based on the needs of applications in the sensor network space. Bandwidth, latency, and energy costs of TinySec are low for sensor network applications. TinySec is easily extensible and has been incorporated into higher level protocols [2, 15].

3.2.Asymmetric Cryptographic Frameworks: are based on two shared keys, private key for decryption and public key for encryption. Asymmetric cryptographic frameworks are further classified as:

3.2.1. RSA based cryptographic frameworks: RSA is computationally intensive and usually execute thousands or even millions of multiplication instructions to perform a single-security operation. The number of clock cycles required to perform a multiplication instruction primarily determines a microprocessor's public key algorithm efficiency. It usually takes a microprocessor thousands of nano-joules to do a simple multiplication function with a 128-bit result [2, 16]. An example of the RSA based Cryptographic framework is the:

3.2.1.1.TinyPK: allow authentication and key agreement between a sensor network and a third party as well as between two sensor networks. TinyPK incorporating the use of TinySec provides the functionality needed for a mote and a third-party to mutually authenticate to each other and to communicate securely. TinyPK is based on the well-known RSA cryptosystem, using e=3 as the public exponent [2, 17].

3.2.2. Elliptical Curve Cryptography (ECC) based cryptographic frameworks: The security of ECC is based on the elliptic curve discrete logarithm problem, which the cryptographic community regards as much more difficult than the integer factorization and discrete logarithm problems that underlie the conventional Rivest-Shamir-Adelman (RSA) and Diffie-Hellman public-key algorithms. ECC has two main advantages:

- ECC public keys are smaller for the same level of security as RSA or Diffie- Hellman-based solutions, thus reducing the number of bits that need to be exchanged.
- ECC public-key operations require fewer computations than conventional public-key methods. The benefit of smaller key is that they need less storage, less bandwidth and, therefore, less energy, thereby reducing processing and communication overhead, which is ideal for energy-constrained sensor nodes. A good example of the ECC based cryptographic framework is the:

3.2.2.1.TinyECC: The primary objective of TinyECC is to provide a ready-to-use, publicly available software package for ECC-based PKC operations that can be flexibly configured and integrated into sensor network applications. TinyECC provides a number of optimization switches, which can turn specific optimizations on or off based on developer's need. Different combinations of the optimizations have different execution time and resource consumptions, giving developers great flexibility in integrating TinyECC into sensor network applications [2, 18].

3.2.3. Pairing based cryptographic frameworks: Cryptography using Pairings (PBC) is an emerging field related to ECC, which has been attracting the interest of international cryptography community, since it enables the design of original cryptographic schemes and makes well-known cryptographic protocols more efficient.

3.2.3.1.TinyPBC: it allows sensor nodes to exchange keys in an authenticated and non-interactive way. They also present the fastest pairing computation on an 8-bit platform, which shows the best figures for binary field multiplication on an 8-bit platform [2, 19].

3.2.3.2.TinyPairing: is an efficient and lightweight pairing-based cryptographic library for sensors. It provides a better way to compute quickly as it consumes low memory for both the cases RAM and ROM by deliberately choosing some super-singular elliptic curve as the pairing group and some specific finite field, which defines the elliptic-curve pairing group [2, 20].

4. Introduction to Geo-Encryption

The term “geo_encryption” or “location-based encryption” refer to a security algorithm that limits the access or decryption of information content to specified locations and/or times [3]. The algorithm does not replace any of the conventional cryptographic algorithms, but instead adds an additional layer of security. [5]

Traditional encryption is used to provide assurance that only authorized users can use the secure content. However, there are circumstances where the security provided by traditional encryption is not adequate. In many instances, it would still be useful to have an additional layer of security that provides assurance that the secure content can only be used at authorized location and/or time. The concept of location based encryption or geo-encryption is being developed for such a purpose [5].

The capability has tremendous potential benefits to applications such as managing classified/secure, wireless sensor networks data and digital movie distribution where controlling access is the predominate concern.

A DYNAMIC TOLERANCE DISTANCE is also considered in final key, which provides a range of location coordinates to the client to decrypt the data, it increases the accuracy & inconsistency of coordinates. Time, as constraints could be used on the decryption location. Data can be encrypted using time constraints, for example, data can access only for a particular time period like an organization provides the access of data in working hours only [3].

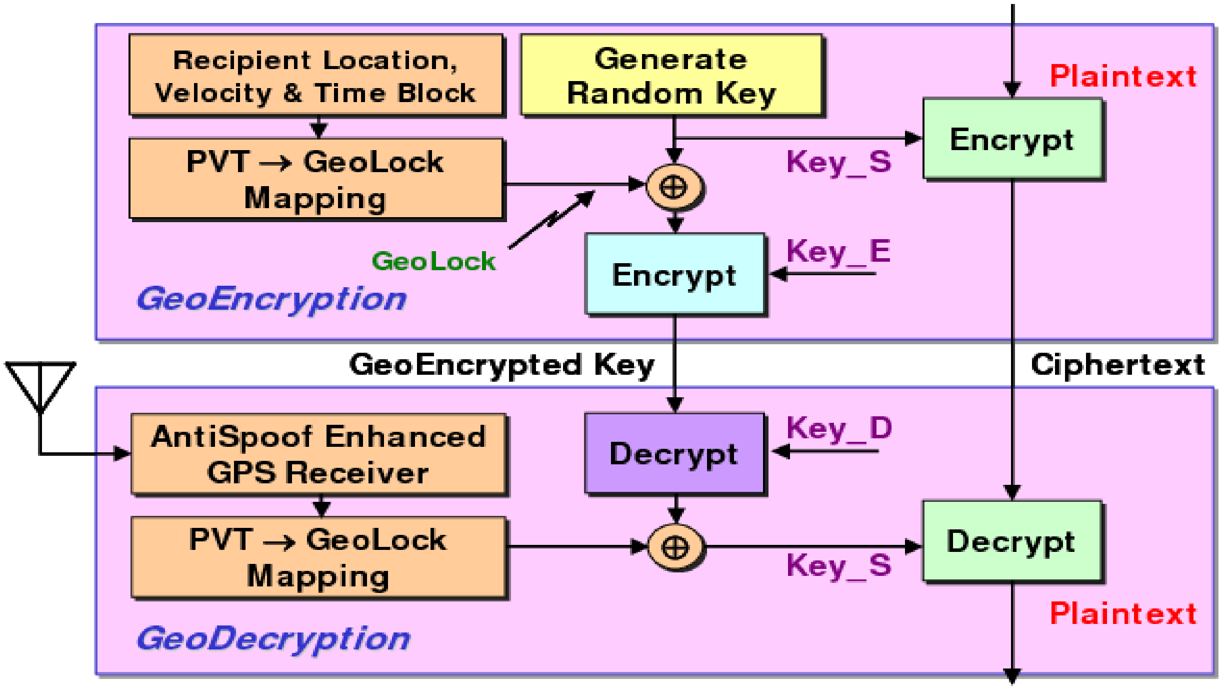


Fig 2: Geo-encryption and Decryption with Key generation using receiver's geo location. [5]

Geo-encryption works as follows [Fig 2]:

Encryption

a. The sender generates any random key of his choice that he will be using for encrypting the plaintext.
b. He then gets the receiver's geolocation (longitude and latitude), Velocity and time required for the decryption, known as PVT all together.
c. The PVT is then applied a geo-lock mapping function to convert it to real co-ordinate location for granting access.
d. The PVT after being Geo-locked is exclusive ored with sender's encryption key and then send to the receiver as decryption key.

Note: the sender encrypts the plaintext with his own random generated key not with the geo-locked key.

Decryption

a. The receiver gets it location using GPS
b. Apply the geo-lock mapping function to the PVT obtained by the GPS to obtained location co-ordinates.
c. The output from above is then exclusive ORed with the send key by the sender
d. The result from step 3 is then use to decrypt the send cipher text.

Note: if the receiver is not at the right position/location the cipher text will not be decrypted. Because the exclusive ORing with key send by the sender confirms the location and reveal the real decryption key.

5. Introduction to Min AES

The Advanced Encryption Standard (AES) is one of the secured known encryption algorithm currently. It has a block size of 128 bits, and supports key sizes of 128, 192 and 256 bits. The number of rounds is 10, 12 or 14 for the three different key sizes respectively. Just like the DES, the AES is expected to draw much attention from cryptographers and cryptanalysts alike within the space of time from now until the next few decades [4].

Min AES is just a small version of the real AES with all the parameters significantly reduced while preserving the original structure of AES [4].

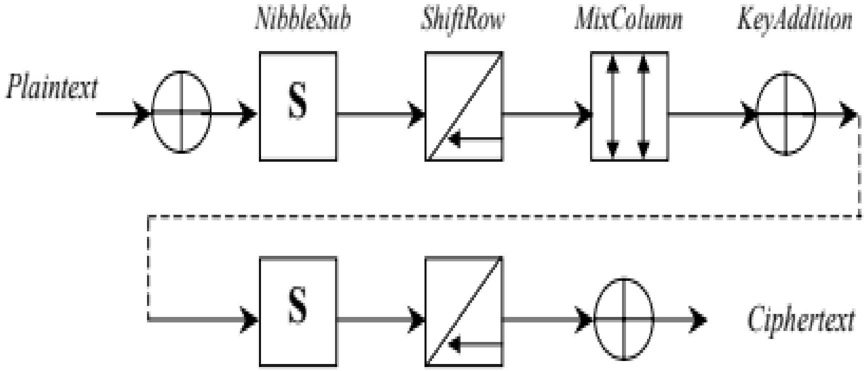


Fig 3: The Min-AES Encryption process [4]

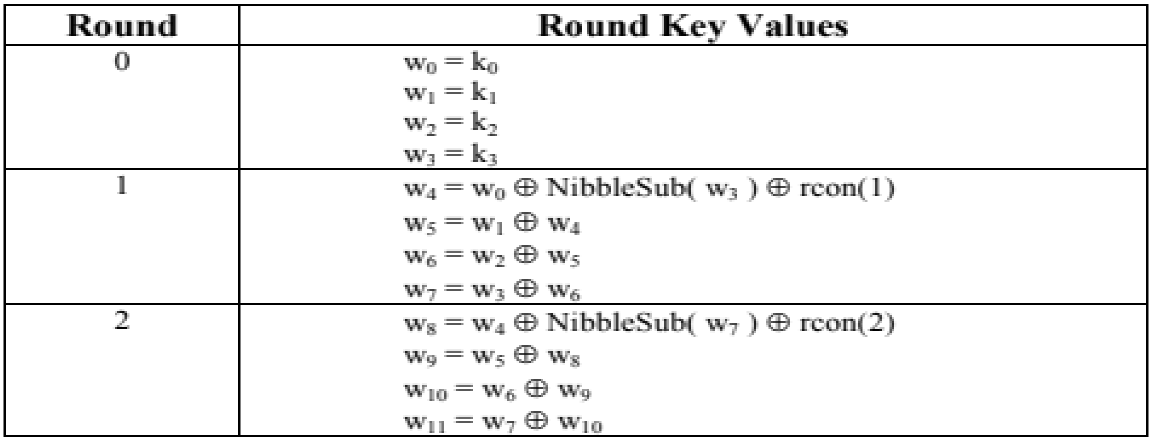
5.1. Components of Min AES

Min-AES have all components in Nibble (4bits) as oppose to the real AES that works on 16 words (128bits) [Fig 3]. The components of MinAES are as follows:

- **NibbleSub:** NibbleSub is a simple operation that substitutes each input nibble with an output nibble according to a 4 × 4 substitution table (S-box) [4].
- **ShiftRow:** Shift Row rotates each row of the input block to the left by different nibble amounts. The first row is unchanged while the second row is rotated left by one nibble. [4]
- **MixColumn**: MixColumn takes each column of the input block and multiplies it with a constant matrix to obtain a new output column [4].
- **AddRound_Key:** KeyAddition causes each bit of the input block, D = (d0, d1, d2, d3) to be exclusived-Ored with the corresponding bit of the ith round key, Ki = (k0, k1, k2, k3) to obtain the 16-bit output block E = (e0, e1, e2, e3) [4].

In Mini-AES, the 16-bit secret key is passed through a key-schedule to produce one 16-bit round key, K0 to be used prior to the first round, and a 16-bit round key, Ki for use in each round of

Mini-AES. Mini-AES encryption is defined to have 2 rounds, hence three round keys, K0, K1 and K2 are generated [4] [Table 1].



Round	Round Key Values
0	$w_0 = k_0$ $w_1 = k_1$ $w_2 = k_2$ $w_3 = k_3$
1	$w_4 = w_0 \oplus \text{NibbleSub}(\ w_3\ ) \oplus \text{rcon}(1)$ $w_5 = w_1 \oplus w_4$ $w_6 = w_2 \oplus w_5$ $w_7 = w_3 \oplus w_6$
2	$w_8 = w_4 \oplus \text{NibbleSub}(\ w_7\ ) \oplus \text{rcon}(2)$ $w_9 = w_5 \oplus w_8$ $w_{10} = w_6 \oplus w_9$ $w_{11} = w_7 \oplus w_{10}$

Table 1: Round key generation for Min AES [4].

6. Existing Works

6.1.The SPIN (Security Protocol for Sensor Networks) framework is one of the symmetric cryptographic frameworks used on WSNs as a security mechanism for communication between sensors consists of SNEP (Secure Network Encryption Protocol) and µTESLA (the micro version of the Timed, Efficient, Streaming, Loss-tolerant Authentication Protocol). The SPIN broadcast all the information to every node in the network. Every node has related data with the neighboring node. The protocol distributes information to all nodes while the user doesn't require exchanging data between nodes. SPIN is a 3-stage protocol which uses three messages such as ADV, REQ & DATA. ADV is advertising new data, REQ is requested for data. DATA is the message itself. When a node wants to share data it broadcast an ADV message containing data. If the neighbor node is interested in receiving the data then it sends an REQ message back to the node for data transmission & DATA is send to the node. Then the neighboring nodes repeat this process with its neighbors and the whole sensor area network will receive a copy of the data [23].

6.2.Extended SPINS Framework

6.2.1. Two-phase Hybrid Cryptography Algorithm [Fig 4]

Two-phase Hybrid Cryptography Algorithm (THCA) is a method that combines the aspects of symmetric and asymmetric techniques in performing a two-parallel phase (Fig 3). They aim to avoid the obscurities in available technique through the realization of high-security levels without increasing the execution time. In the encryption, the plaintext is divided into n blocks, Bi. Each block comprises of 64 bits. If n is not an integer number and has a fraction, DRSM algorithm uses padding with null for the last block to be 64 bits. The encryption process is divided into two phases. In Phase I, plain text is divided into n/2 blocks. First, part is encrypted using (DES) algorithm and Second part are used encrypted (RSA) algorithms.

Phase II is performed after of Phase I by combining the encrypted blocks together. To increase the security, the output of Phase II the blocks are encrypted by using (AES) algorithm.

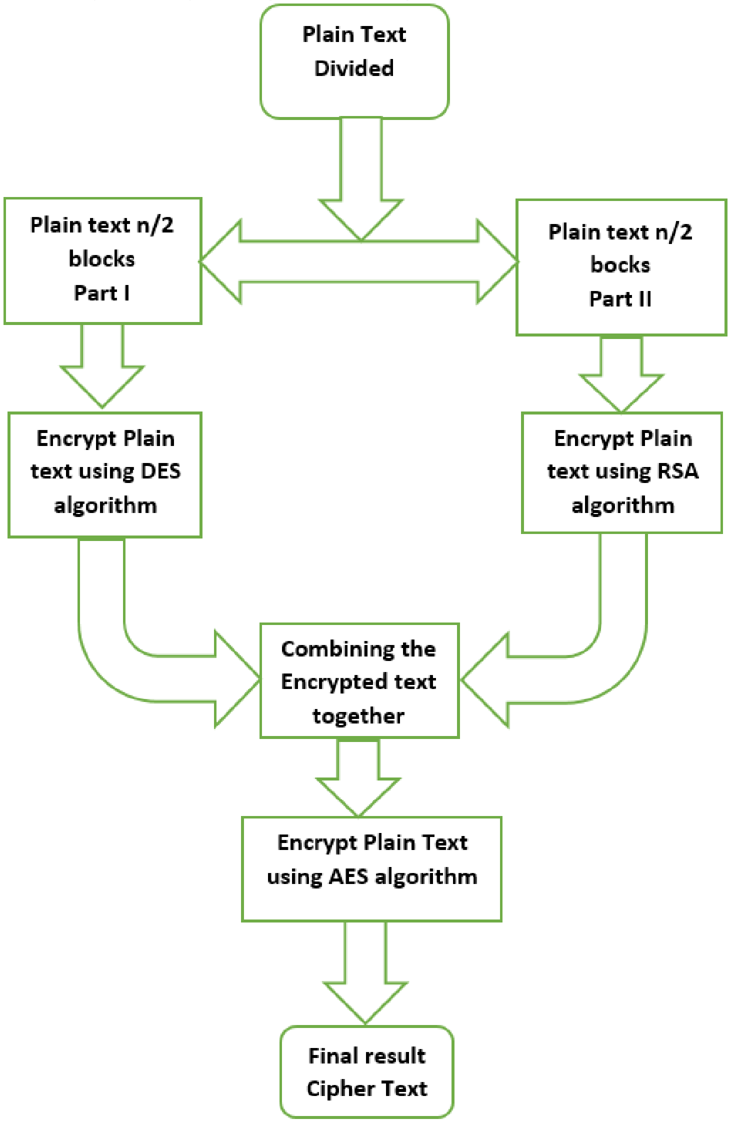


Fig. 4: Two-phase Hybrid Cryptography Algorithm ESPINS [12]

6.3.Data Encryption Standard (DES): The DES (Data Encryption Standard) algorithm is the most widely used encryption algorithm in the world. It is a block cipher that uses the same binary key both to encrypt and decrypt data blocks. It is called a symmetric key cipher. DES operates on 64-bit “plaintext” data blocks, processing them under the control of a 56-bit key to produce 64 bits of encrypted cipher-text. Similarly, the DES decryption process operates on a 64-bit cipher-text block using the same 56-bit key to produce the original 64-bit plaintext block. It uses a series of operations to encrypt a data block. These primitives are used to inverse the encryption operation. The predominant weakness of DES is its 56-bit key which, more than sufficient for the time period in which it was developed, has become insufficient to protect against the unbreakable force attack by modern computers [23].

6.4.The Advanced Encryption Standard (AES): is used to encrypt (encipher) and decrypt, (decipher), information.as shown in below figure. The Key Development generates a Key Agenda that is used in Cipher and Converse Cipher procedures. Cipher and Inverse Cipher are composed of a specific number of rounds. The number of rounds to be performed through the application of the algorithm depends on the key length [23].

7. Limitations of Existing Work

Various research has already taken place on the aspect of geo-encryption with AES, but it have certain limitations that makes it infeasible for use on wireless sensor networks. We have listed out some limitations here:

a. Encryption algorithms used on SPIN framework such as AES and DES cannot provide effective and efficient security output, because sensor nodes have limited energy resources, insufficient memory capacities and limited processing capabilities [7, 9].
b. The dissemination of data in the network through SPIN protocol takes long time [23].
c. AES algorithm despite being a secured algorithm, works on a block size of 128bit and a key of 128, 192, or 256bits which is beyond the storage and memory capacity that a sensor node and base sinks can utilize for the encryption process [7, 8, 9].
d. A sensor node with much more computation consumes more energy [23].
e. Also the cost, implementation and maintenance of real AES on WSNs is high and cannot be adopted effectively [9].
f. Few sensor nodes may be used several times and those nodes may lose energy early than other nodes in the network [23]

8. Proposed System

8.1.Description

It is in line with the aforementioned limitation that this research paper proposed a new strategy of tackle the limitations of existing works. The proposed system aims at providing a secured means of data transmission between nodes, base stations (sinks) and users of a wireless sensor networks by using one of the secured cryptographic algorithms (Min AES) with geo-encryption to ensure communication only between authorized entities of such network that are uniquely identified and known to each other by their respective geolocations and special tokens. This will help in tackling all the security threats face by WSNs (Sybil attack warm-hole attacks etc.) and to provide effective way of enhancing the SPIN protocol.

8.2.Proposed System Architecture

The system will have the following modules:

- ✓ **Min AES MODULE:** with 3 rounds, 16bits block size (As discussed previously) with a common 16bits key agreed by sending sensor and intended receiver which is changed accordingly by the key expansion module for each round.
- ✓ **CORDINATES CONVERSION MODULE:** Each sensor node needs to keep track of all it communicating node exact geo locations (longitude and latitude degrees) as well as a special token peculiar to each sensor (4 hexadecimal digits e.g BEF9) which will be use to generate a 16bits word to be exclusive ored with the output of the Min AES operation. Both values generated by these modules at both ends need to be the same for correct encryption and decryption processes.

8.3.Algorithm Steps for the Proposed System Based on Spin Framework

Encryption [Fig 5]

Step 1: Sensor node with captured data sends an ADV broadcast message to all neighboring sensors on WSNs

Step 2: Receiving sensor respond with REQ request to get the data.

Step 3: Sending sensor receives the REQ and obtain the requesting/receiving sensor’s geo-coordinates.

Step 4: Sending sensor confirms the requesting sensor as part of illegible sensors to share data with, else it rejects the request.

Step 5: The sending sensor perform encryption using Min AES with a 16bits key shared among sensors.

Step 6: Convert the geo-coordinates into 16bits word using the coordinate conversion process.

Step 7: XOR the Min AES output with the resulting 16bits of the coordinate conversion process.

Step 8: The resulting output from step 7 is the cipher text to be transmitted to the receiving sensor.

Decryption [Fig 5]

Step 1: Receiving sensor receives the cipher text and apply Min AES decryption process using the predefined shared key between sensors.

Step 2: Apply the coordinate conversion process to obtain a 16bits word using the receiver’s geo-coordinate.

Step 3: XOR the resulting output from step 2 with the output of the Min AES decryption operation.

Step 4: The resulting output in step 3 above is the plaintext send by the sending sensor.

8.4 Flow Chart Diagram for Proposed System

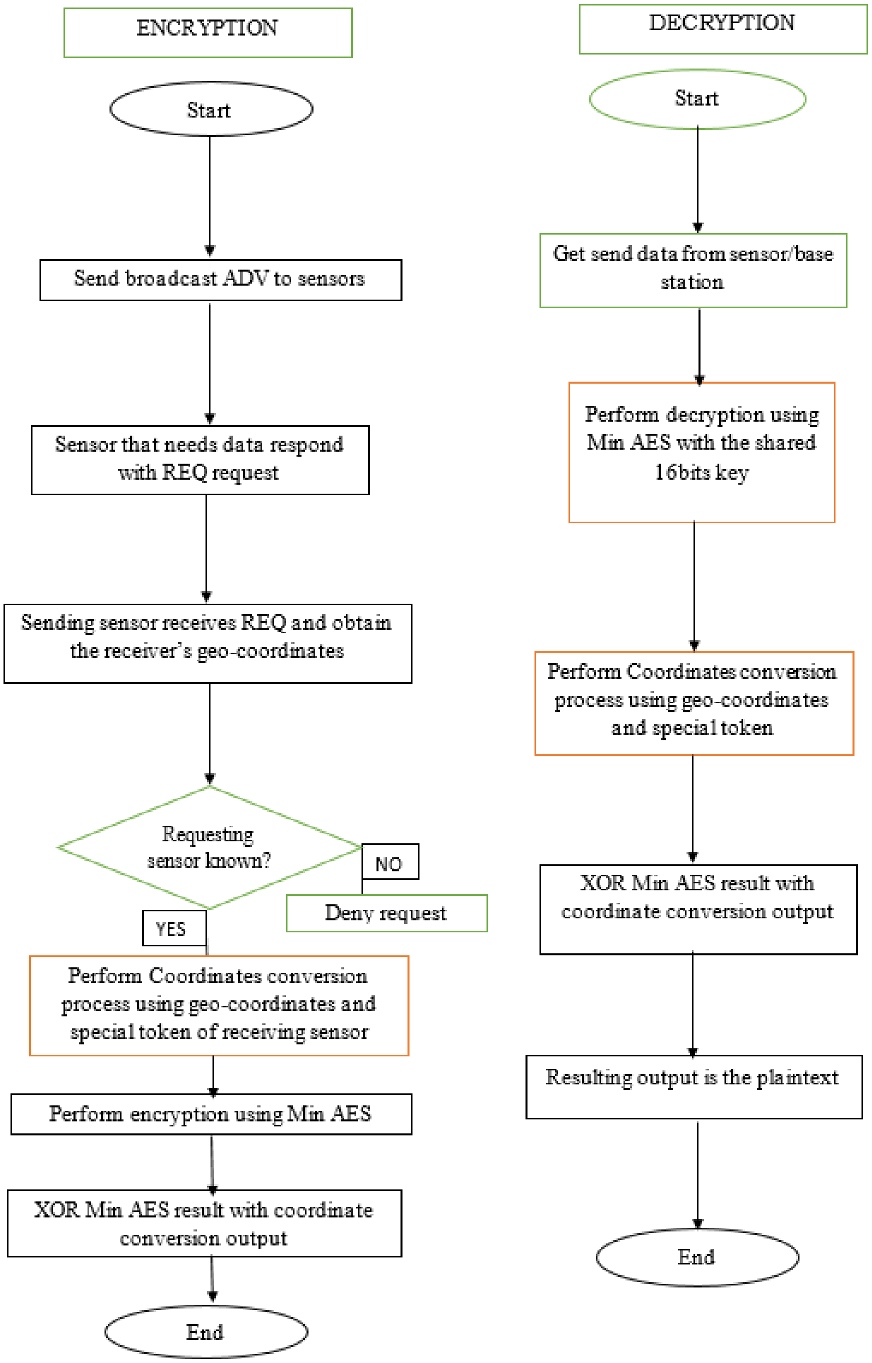
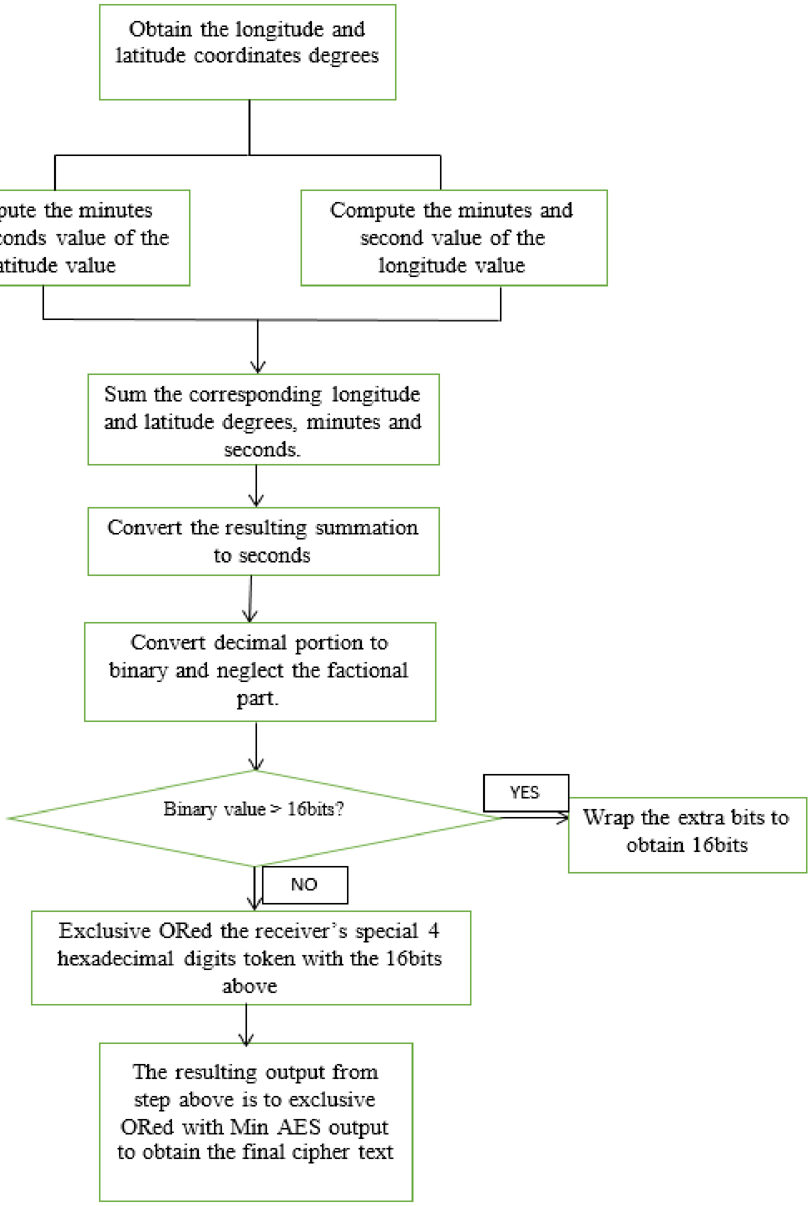

ENCRYPTION
Start
Send broadcast ADV to sensors
Sensor that needs data respond with REQ request
Sending sensor receives REQ and obtain the receiver's geo-coordinates
Requesting sensor known?
NO
Deny request
YES
Perform Coordinates conversion process using geo-coordinates and special token of receiving sensor
Perform encryption using Min AES
XOR Min AES result with coordinate conversion output
End
DECRYPTION
Start
Get send data from sensor/base station
Perform decryption using Min AES with the shared 16bits key
Perform Coordinates conversion process using geo-coordinates and special token
XOR Min AES result with coordinate conversion output
Resulting output is the plaintext
End

Fig. 5: Flow chart for the encryption and decryption of the proposed system.

8.5 Coordinates Conversion Process



Obtain the longitude and latitude coordinates degrees
Compute the minutes and seconds value of the latitude value
Compute the minutes and second value of the longitude value
Sum the corresponding longitude and latitude degrees, minutes and seconds.
Convert the resulting summation to seconds
Convert decimal portion to binary and neglect the factional part.
Binary value > 16bits?
YES
Wrap the extra bits to obtain 16bits
NO
Exclusive ORed the receiver's special 4 hexadecimal digits token with the 16bits above
The resulting output from step above is to exclusive ORed with Min AES output to obtain the final cipher text

Fig 6: The coordinate conversion process

8.6 Example of Coordinate Conversion Process [Fig 6]

Suppose the receiving sensor coordinates that receives a broadcast ADV by a particular sensor that have the data to share in a WSN using SPIN framework is obtained as

Latitude= 43.677719° and Longitude= -79.624817°

We need to convert the coordinates from degree to time

1° = 1 Hour

STEP 1 A:

Taking the Latitude

The decimal part remains as it is, while the first 2 digits after the decimal represent the minutes and are converted as follows:

0.67 * 60 minutes = 40.2 minutes. Represented as 40'

We also need to convert fractional part of the minutes to seconds as follows:

0.2 * 60 seconds = 12 seconds. Represented as 12"

The remaining fractions represents the seconds and are converted as follows

0.007719 * 3600 = 27.7884 seconds represented as 27.7884''

Therefore, number seconds = 12 + 27.7884 = 39.7884 seconds.

FINAL LATITUDE **VALUE = 43 hours 40 minutes 39.7884seconds**

STEP 1 B:

Taking the Longitude

Same as above

0.62 * 60 = 37.2'

0.2 * 60 = 12''

Then the seconds' portion

0.004817 * 3600 = 17.3412''

FINAL LONGITUDE VALUE = **-79ours 37minutes 29.3412seconds**

STEP 2:

We then sum up the resulting coordinates time values

43hours 40minutes 39.7884seconds

+

-79hours 37minutes 29.3412seconds

-22 hours 77minutes 69.1296 seconds

NOTE: we are to neglect the sign of the summation result if it comes up to be negative

STEP 3:

We then convert the above summation result to seconds

22hours * 3600seconds = 79,200seconds

77minutes * 60seconds = 4620seconds

Total time in seconds = 79200+4620+69.1296 = 83,889.1196 seconds

STEP 4:

We then convert the decimal portion of the above second (**83,889**) value to binary and neglect the fractional part (**.1196**). If it exceeds 16bits we wrap the extra bits to the right.

65,536 32,768 16384 8192 4096 2048 1024 512 256 128 64 32 16 8
4 2 1

1 0 1 0 0 0 1 1 1 1 0 1 1 0
0 0 1

Since the binary value is greater than 16bits we need to wrap the extra bits

0 1 0 0 0 1 1 1 1 0 1 1 0 0
0 1

1

0 1 0 0 0 1 1 1 1 0 1 1 0 0
1 0

STEP 5:

Each sensor node on the WSN is given a unique 4 hexadecimal digits token known only between the sensors. The token is then converted to binary and then exclusive ored with the result obtained at step 4 above

Suppose the receiving sensor unique token is B E F 9

B E F 9

1011 1110 1111 1001

⊕ **0 1 0 0 0 1 1 1 1 0 1 1 0**
0 1 0

1 0 1 1 1 1 1 0 1 1 1 1 1
0 0 1

1 1 1 1 1 0 0 1 0 1 0 0 1
0 1 1

The 16bits binary value obtained above in step 5 is to be exclusive ored with the Min AES encryption output to obtain the final cipher text

The receiver needs to compute same and then exclusive ored it with the Min AES decryption output to get the real and final plain text.

9. CONCLUSION

SPIN Protocol used on wireless sensor networks provides certain level of security that guarantees for data confidentiality. However, the processing time taken in WSNs using SPIN framework takes longer time to accomplish due to limited memory, computing power and hardware requirement of sensors to enable them adapt to the AES algorithm. It is in line with this that the proposed system aims at using Min AES which is a version of the advanced encryption standard algorithm with all components reduced but maintain the normal structure, operations and security of real AES to tradeoff the limitation of the sensors that makes it impossible to handle AES effectively.

In addition, the Min AES is to be coupled with geo_encryption (use of sensor latitude and longitude) approach to provide a reasonable level security of data on a wireless sensor by using a special of form of algorithm to make the geo-coordinates provides and abstract level of security that can guarantees confidentiality of data across WSNs.

REFERENCES

[1]. Murat Dener, *"Comparison of Encryption Algorithms in Wireless Sensor Networks", ITM Web of Conferences 22, 01005 (2018)*

[2]. Gaurav Sharmaa, Suman Balaa, Anil K. Vermaa, "*Security Frameworks for Wireless Sensor Networks-Review*", 2nd International Conference on Communication, Computing & Security [ICCCS-2012]

[3]. Himanshu Pant,Vinay Kaushik et al, "*GEO-ENCRYPTION TO ACCESS THE DATA USING AES ALGORITHM*", International Journal of Engineering Applied Sciences and Technology, 2016.

[4]. Raphael Chung-Wei Phan, "*Mini Advanced Encryption Standard (Mini-AES): A Testbed for Cryptanalysis Students"*, Published in Cryptologia, XXVI (4), 2002.

[5]. Logan Scott, Dorothy E. Denning. "*A location-based encryption technique and some of its applications*", Researchers gate January 2003.

[6]. Chun-ta Li, "*Security of Wireless Sensor Networks: Current Status and Key Issues*", Intech Open Published: December 14th 2010.

[7]. C-Y. Chong, S.P. Kumar, "*Sensor Networks: Evolution, opportunities, and challenges*", Proc IEEE, 91(8), 1247-1256, (2003).

[8]. M. Dener, "*Security Analysis in Wireless Sensor Networks*", International Journal of Distributed Sensor Networks, 2014, Article ID 303501, 1-9, (2014).

[9]. R. Lin, Z. Wang, Y. Sun, "*Energy Efficient Medium Access Control Protocols for Wireless Sensor Networks and Its State-of-Art*", IEEE, pp 669-674, (2004).

[10]. I.F. Akyıldız, W. Su, Y. Sankarasubramaniam, E. Çayırcı, "*A survey on sensor networks*", IEEE Communications Magazine, 40(8), 102-114, (2002).

[11]. T. Kavitha, D. Sridharan, "*Security Vulnerabilities in Wireless Sensor Networks: A Survey*", Journal of Information Assurance and Security, 5, 31-44, (2010).

[12]. Khalid Abdullah et al, "*Extended SPINS framework for security wireless sensor networks*", international journal of computer science issues, 2017

[13]. Perrig, A., Szewczyk, R., Wen, V., Culler, D., Tygar Spins: "*Security Protocols for Sensor Networks"* 7th annual ACM/IEEE international conference on mobile computing and networking, 2017.

[14]. Zhu, S., Setia, S., Jajodia, S., 2003. *"LEAP: Effcient Security Mechanisms for Large-Scale Distributed Sensor Networks"* ACM Conference on Computing and Communication.

[15]. Karlof, C., Sastry, N., Wagner "*TinySec: A Link Layer Security Architecture for Wireless Sensor Networks*" ACM Conference on Embedded Networked Sensor Systems (SenSys 2004), Baltimore, MD.

[16]. Carman, D., Kruus, P., Matt, B., 2000. Constraints and approaches for distributed sensor network security. NAI Labs, TR

[17]. Watro, R., Kong, D., Cuti, S., Gardiner, C., Lynn, C., Kruus "*TinyPK: securing sensor networks with public key technology*", 2nd ACM Workshop on Security of ad hoc and sensor Networks (SASN'04)

[18]. Liu, A., Ning, "*TinyECC: A configurable library for elliptic curve cryptography in wireless sensor networks*", 2008, Washington, DC, USA, IEEE Computer Society.

[19]. Oliveira, "*TinyPBC: Pairings for authenticated identity-based non-interactive key distribution in sensor networks*", 2008, 5th International Conference on Networked Sensing Systems.

[20]. Xiong, X., Wong, D., Deng, "*TinyPairing: A Fast and Lightweight Pairing-based Cryptographic Library for Wireless Sensor Networks",* WCNC 2010, IEEE Communications Society.

[21]. Shodhganga, "*Introduction to wireless sensor networks*", (https://shodhganga.inflibnet.ac.in/bitstream/10603/22912/7/07_chapter_01.pdf), 2018.

[22]. Zille Huma Kamal and Mohammad Ali Salahuddin, "*Wireless Sensor and Mobile Ad-Hoc Networks: Vehicular and Space Applications*", D. BenHaddou and A. Al-Fuqaha (Eds.), pp. 3-32, Springer, New York, 2015.

[23]. Bavarva, Arjav & Patel, Nidhi & Kathiriya, Hiren. (2013). WIRELESS SENSOR NETWORK USING ZIGBEE. International Journal of Research in Engineering and Technology. 2. 1038-1042. 10.15623/ijret.2013.0206021.