Nanotechnology Perceptions
ISSN 1660-6795
www.nano-ntp.com

A Bootstrapped Volatile Key Management Scheme for Efficient and Secured Data Transmission in WSNs

Girija Vani Gurram[1], Phanindra Reddy Kannari[2*], Noorullah Shariff C[3]

[1]*Associate Professor, Department of Electronics & Communication Engineering, BITM, Ballari, vanigirijag@gmail.com*
[2]*Associate Professor, Department of Artificial Intelligence & Machine Learning, BITM, Ballari, phanindrareddyk@gmail.com*
[3]*Professor, Department of Artificial Intelligence & Machine Learning, BITM, Ballari, cnshariff@gmail.com*

The recent era has witnessed several advances in Wireless Sensor Networks (WSNs), with few resources presents unprecedented challenges for secured data transmission in an ever-unsecured broadcast network. The situation is more challenging when adversaries are equipped with sophisticated resources having direct access to sensor nodes. Nevertheless, there exists several cryptographic techniques used for node authentication, authorization, data confidentiality, integrity, and other security related services. This paper contributes to novel network security strategies to prevent network compromise. The proposed strategies attempt to boost data integrity, reliability and secure data transmissions. The proposal identifies the strategically important nodes (SIN) in implementing network security measures using volatile symmetric-key management scheme (KMS).

The scheme involves four phases: The first phase involves strategically important nodes (SIN) designation using Grey Wolf Optimization (GWO). Usually, hackers find it difficult to guess and access the keys that are selected on a random basis. To further increase the key protection, the scheme introduces a limited lifespan master key (LLMK). Secondly, to secure the KMS, this phase utilizes SHA for LLMK distribution, and paired node prediction. Next phase performs secret key generation with a gradient approach and the key sharing is performed using Supersingular Isogeny Diffie-Hellman algorithm. The last phase is responsible for the dynamic S-Box generation using the Blowfish algorithm. The generated S-Box are subsequently shuffled to enhance the cryptographic process offering immunity towards attacks.

To evaluate the effectiveness of this approach, the proposed KMS is compared with relevant methods on various performance metrics. The results demonstrate that the proposed scheme achieves notable performance improvements over other methods.

Keywords: Volatile key management, WSNs, GWO, SIDH, Blow fish Algorithm, Dynamic S-Box generation.

1. Introduction

Wireless Sensor Networks have been gaining popularity over a few decades due to their low-cost, low-power, scalable and ease of deployments. However, these networks has been facing serious security challenges based on design due to the limited computational resources and their minimal battery life [1]. Its architecture comprises of numerous nodes that ranges from 100s to 1000s, deployed in a network. The sensor nodes are used for sensing the data that is generated from surroundings and route it back to the base station via hopping techniques. However, these nodes have poor energy and memory resources. Several routing protocols have been devised to improve the network's lifetime and compensate for resource insufficiencies. The wireless structure is prone to various attacks, that would easily alter the sensed data which affects the confidentiality of the network data as well. Hence, data integrity and authentication might be compromised, thus securing against the vulnerabilities using various secure routing protocols and cryptographic techniques are mandatory for the appropriate functioning of the WSNs.

The sensors present in WSNs should share their keys to ensure the secure data transmission among the sensor nodes. In WSN, the key management schemes involves network formation and network initialization. The network formation follows either shared key or key establishment approach. Under the shared key approach, two nodes attempt to discover a public key, whereas in key establishment approach, several SNs establish a common key for secure communication. Various approaches are developed for key establishment and with resource scarcity, designing or structuring key distribution is an interesting task. Some of the KMS discussed in [2] are Public key cryptography, Randomized key distribution, Global master key and transitory master key. As security services like confidentiality, data integrity, etc must be enforced, cryptographic keys are to be utilized. This is where key management performs vital part in WSNs. The volatile key management is gaining traction in recent days due to the limited life span, preventing unauthorized access to the keys. The key management comprises of Symmetric, Asymmetric and Hybrid schemes [3], as shown in the figure 1. To establish volatile keys, a symmetric key management scheme [4] is essential to facilitate the keys between neighboring nodes without using any key ring or pools.

The initialization stage allows a pair of nodes to exchange a plain text message, with the plain text message being encrypted and the encrypted nonce is used by the network node, where a LLMK should only be inserted into some potential nodes (SINs). These SINs ensure the security of the LLMK from the attackers to expose the used key for any given node directly after the deployment process, even if the attacker captures few network nodes. The technique also ensures that the LLMK reaches every node in the network enabling network connectivity.

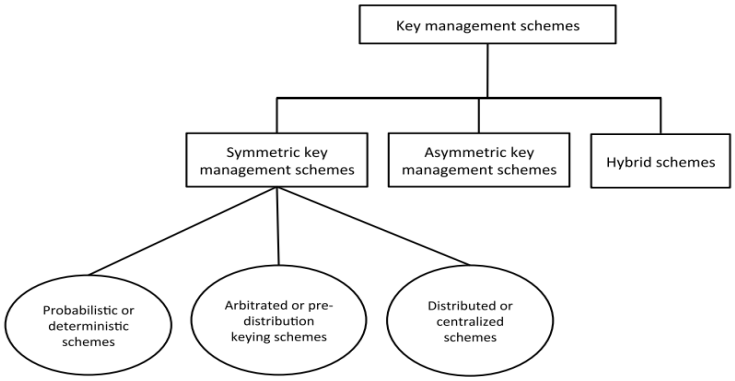

Figure 1: Key Management Scheme – A Hierarchy

Thee proposed work applies a volatile key management scheme to generate the LLMK, SIN node designation using an optimization algorithm, a hashing algorithm to distribute volatile key and finally cryptographic techniques to ensure secure data communication using the shared keys.

The paper discusses the existing security and KMS of WSNs in section 2. The methodology and implementation along with respective algorithms are explained in section 3. The performance analysis and the results discussed in section 4. Section 5 provides conclusion and future enhancements.

2. Related Works

The design of secure and efficient routing protocols for WSNs is an active area of research due to the unique security challenges presented by these resource-constrained networks. The attacks against WSNs can be categorized into two types namely active attacks and passive attacks [5]. There exists a plethora of security mechanisms for secure data transmission, data aggregation, using key exchange and key management schemes. However, the complex broadcast channel impairments and limited resources weakened the full-fledged implement of a given security mechanism in WSNs. This section discusses various existing approaches that attempted to provide secure key exchanges, and transmission in WSNs.

Key Management schemes:

Several research works present the vulnerabilities and complicated nature of WSNs against attacks, for the network utilizing a random pairwise key distribution scheme. The random key distribution schemes experience various issues while assigning the secured keys to sensor nodes, prior to network deployment. [6] investigated that whether resiliency could be achieved in WSNs beneath random pairwise key redistribution scheme. This work presented certain conditions on model parameters such that the network would be un-splitable and unassailable, with maximum probability as the ‘n’ nodes increase. These ideas were developed against an intruder who possessed enough computing resources and good knowledge about network topology, however, with an assumption that only a fraction of sensor nodes are compromised. This research also proved that cryptographic keys should ensure un-splitability and

unassailability under the pairwise key distribution method.

More importantly, compromised sensor nodes in military surveillance or disaster management applications across horizontal and vertical security landscapes is unacceptable. However, a deployment based key distribution scheme had been developed for cellular model in WSN for thwarting attacks [7]. The key distribution model is utilized in all those networks where the random selection of small subset of keys is chosen from the collection of key pools. At least, there should exist one common key from key pool for the secured connection among the nodes. The secured communication link relies on the length and strength of keys used for the connection establishment. The major focus of this research centered around node density, neighbor influence factor, sink node placement, gradient distance and application types on nodes to construct the attack matrix of each mobile cell. An efficient key management strategy along with secured data aggregation over Dynamic WSN had been implemented in [8]. This research work utilized a Key Generation Center (KGC) for dealing with partial private keys. This research work addresses escrow issue with the help of KGC.

The sensor nodes in WSNs would share their keys for secured data transfer. The demand of designing an energy optimized key establishment/distribution in WSNs is increasing due to the limited resources of sensor nodes. Various polynomial based key distribution scheme had been proposed in WSNs to provide a light weighted solution for devices with limited resources. These polynomial – based approaches are highly susceptible toward sensor attacks. Sometimes, the intruders compromise the security system by capturing a fixed number of sensor nodes. [9] developed a polynomial based key management scheme together with probabilistic security that minimizes the security risk of attacks along with minimized computational overhead and memory resources.

To establish a secured data transmission over WSN, several literatures suggested a protocol namely Multi-level Key Management protocol. Under this protocol, every single transmission was undertaken based on available secured key. The research work by [10] developed a protocol for key management referred as MSKM protocol, to perform protected communiqué in the clustered WSNs. This framework implemented in three phases consists of 1) Key Generation 2) Pre-deployment 3) Key Authentication and verification. The key generation is responsible to ensure the secrecy of communication and the necessary keys are generated using homomorphic encryption model. In pre-deployment, node's identity is established. The last phase, a mathematical model had been constructed along with various factors like random numbers, dynamic passwords, etc.

The secured key generation in physical layer (PL-SKG – Physical Layer Secure Key Generation) has some potential benefits to improve the network security posture. This scheme strengthens the security-based protocols by minimizing the quantity of key materials needed for deployment. This scheme is extremely useful in limited resources scenario and also in situations where designing of key management scheme is challenging. [11] discussed about the challenges in PL-SKG scheme, and propose a unique key generation scheme which provided merits on potential and simplicity of correction codes.

The intruder might have unauthorized access which leads to eavesdropping, tampering of data, interception, and modification. Nevertheless, to improve efficacy of WSN, clustering methods are deployed, but sensor nodes' dynamic behavior with limited processing and storage makes

secured transmission, a difficult one. To resolve this issue, effective key management is essential. [12] proposed a secured mechanism called as CAKE - Codeword Authenticated Key Exchange. It is completely based on one-way hashing with OTP and authentication. [13] developed a secured and provable key exchange protocol which is based on ECDH - Elliptical curve Diffie-Hellman for WSNs. Elliptical curve schemes are standard protocol for establishing shared keys. This implementation handles even a lack of authentication related attacks namely Man-in-the-Middle Attack. Enhanced energy utilization is the most primary factor. Thus an efficient method for securing energy in data transfer using Hierarchical and Dynamic ECC[14] had been proposed. LEACH – Low Energy Adaptive Clustering Hierarchy had been utilized for selecting dynamic cluster heads. Here the key management scheme consisted of three phases namely key – computation, key – exchange and, node authentication.

Pair- Node Authentication:

Since WSN is widely applied in critical applications like military surveillance, health care monitoring, etc., the vulnerability of handling the data creates high risk. Since, it is mandatory to maintain confidentiality during data transmission and the data should be authenticated in all aspects. [15] stated that the memory and energy restraints of certain sensor nodes for military applications need light weight cryptographic schemes and propose an asymmetric key cryptographic scheme. The scheme uses the base idea called Elliptic Curve Cryptography and as well use Identity Based Public Key Cryptography.

Here, no permission is provided to third-party thereby in single transmission authentication is accomplished. Thus it is benefitted in a way that the receiving party need not to be active in receiving authenticated message, which decreases the encryption and decryption time without compromising the security level.

When considering large scale scenario of IoT, the trusted authority would be joined by a remote server. This remote server would be accessed through a domestic gateway by using different users to bootstrap the keys that are engaged in secure transmission. But all these users might not fully trust the remote server, especially when they are holding private key of their sensor nodes. So to handle this issue, [16] developed an escrow less and light weighted authenticated key Agreement technique (AKA) for WSNs. Various applications like medical monitoring in healthcare institutions needs highly secured WSN network. To promote that, [17] developed an user authentication scheme and secured data transmission of health care data. This research work discussed various secure transmission schemes like identity-based authentication, mutual authentication, decentralization, multi-routes transmission.

Cryptography:

Cryptography can be defined as the study or practice of several techniques that are involved in secure transmission of data even in the presence of intruders or adversaries. Usually, cryptographic techniques are about developing and analyzing various protocols that tries to stop or prevent third party adversaries or unauthorized public from viewing or intercepting private or sensitive messages. There are several aspects which are major parts of modern cryptography such as data integrity, confidentiality, non-repudiation and authentication. As cryptography shows continuous progress and new advances, WSNs utilized its techniques to improve the security landscape. The cryptography techniques utilized by WSNs are 1) Elliptic

Curve Cryptography, 2) Pairing-based cryptography, 3) Identity based Cryptography.

As WSNs found its wide range of applications, subsequently, outsiders reach to access the sensitive data by directly interacting with sensor nodes increase substantially. Since, WSNs are fragile to attacks, it is essential to provide security to critical or sensitive data such that it should be available only to legitimate users. A scheme based on two factor authentication combined with smart card and password is used to enhance security. But several two factor authentication protocols have been suggested. [18] proposed a 2 factor authentication protocol that is based on ECC for WSN by considering privacy awareness. This novel protocol accomplished several features for real life applications without compromising efficiency.

[19]developed an algorithm which combined a method of merging of both symmetric and asymmetric methodologies such as AES – Advanced Encryption Standard and RSA – Rivest – Shamir – Adleman Algorithm followed by LZW – Limpel – Ziv- Welch compression. The primary objective of this mechanism is to encrypt the plain text via AES which led into faster symmetric encryption. The secondary objective includes encryption of plain text through RSA which leads to the development of secured data process. The last objective is to perform LZW compression. A reversal process of the algorithm at receiver performs decompression and finally the decryption step.

Though there are several techniques proposed, these mechanisms faced below short-comes in certain circumstances. They are as follows,

Most of the existing methods used random number of node selection in order to share the master key before deployment. This might result in selection of malicious nodes, which in turn resulted in poor security.

Some of the techniques used AES, which possess simple encryption process. So, an adversary with enough connected resources can easily break AES encryption.

Most of the existing methods follow non-volatile approach in managing keys. This resulted in increased life span of master key. Master key with high life span would give chance for intrusion.

The proposed security mechanisms attempts to overcome the above-mentioned limitations.

3. PROPOSED WORK

The proposed scheme block diagram is depicted in the figure 2. It involves four modules namely 1) SIN prediction, 2) LLMK distribution and pair node prediction, 3) Key generation 4) Dynamic S-Box generation. To enhance the security of sensitive data against adversaries, a volatile key management scheme has been implemented. A master key with shorter life span (LLMK) gets shared with certain nodes and these nodes are paired up with neighboring nodes for performing data transmission. As mentioned, the master key is inserted into few nodes that are selected using objective function by GWO algorithm. Then, these selected node pairs are ready to transmit data after authentication. The authentication is done between the node which holds the LLMK with broadcast message and the receiver node that got paired.

The concept of volatile key technique is presented briefly in the following part. This concept

is utilized in the proposed work for exploiting some of the basic operations by providing permissions to respond to certain problems and the security measures in WSNs. The symbols utilized in the equations of proposed work are described in Table 1.

Every node present in the network is preloaded by preliminary functions and data before the deployment of nodes. This helps in sharing the common keys among the adjacent neighbor nodes. Further, keys are established separately for each pair of adjacent neighbor nodes in a secured manner through the usage of primary configuration where MAC is utilized before initiating the secured communication in the perspective of authenticated channel. Preliminary data comprises of pseudo random (PR) functions, finite groups that comprise of LLMK (can be inserted only in certain nodes) and the prime numbers as shown in Table.2. Volatile Key is denoted as LLM_k in the proposed work as the master key that would appear in few nodes.

Table 1. Notations and Description

Notation	Description
N	Total number of nodes deployed
n_a	a^{th} node in network
ID_a	Identifier for node a
$S_a \rightarrow$type, * :	Node S_a distributes the message M to each node within the radio communication range
NReq NResp	Pairwise key request and pairwise key response among the nodes
$nonce_a$	Random number produced by S_a
X_a	Random group chosen by S_a
LLM_K	Limited life Master Volatile Key for the secure sharing of $nonce_a$ to n_a, n_b
f(int, int)	function that produces Pseudo-random numbers upon accepting 2 integral types as arguments
K_{ab}	Pairwise secret key for n_a and n_b
Counter	Utilized for restricting a data replay attack; is an alternative to nonce
$K_e(M)$ and K_d (M)	Encryption (e), Decryption (d) of message 'm' using the key K

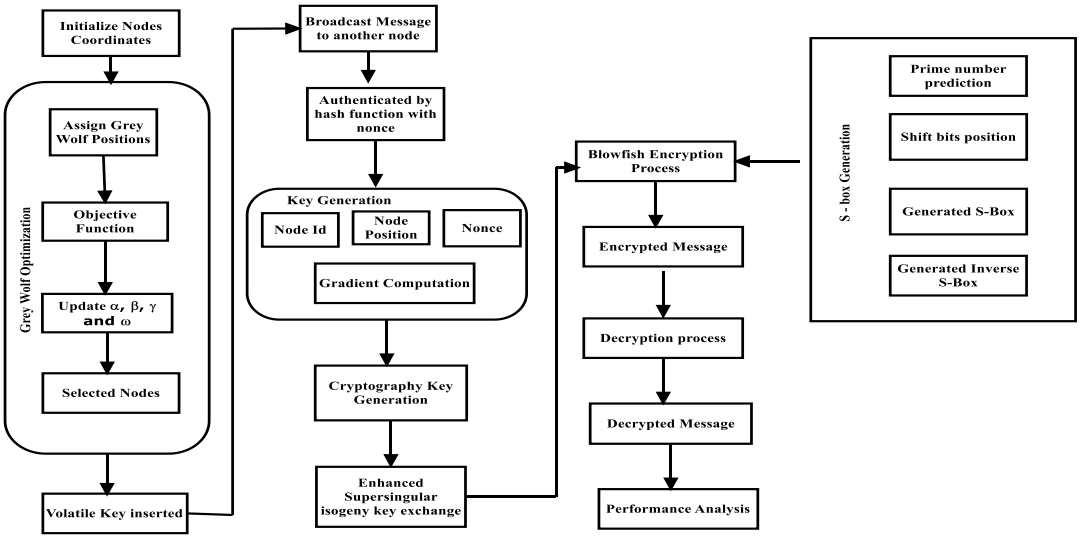

Figure 2: Proposed Scheme Block Diagram

A. LLMK-Node Prediction using GWO:

Previously, the Master Keys (MKs) selection from the key pool has allowed the intruders to predict the MKs resulting into compromised node, network or communications. Thus, to handle this, proposed scheme utilizes optimization algorithm named GWO. The methodology of GWO algorithm is entirely based on hunting nature of wolves. Out of four, only three wolves are included in the hunting process. The decisions like hunting time, sleeping time are determined by wolf α. Thus, α wolf is designated as the leader of the wolf pack. The second and third β and δ wolves respectively support α in making decisions. The last ω wolf is allowed only for eating hence these are not imperative in process of hunting. Initially, all the nodes are initialized with the basic node configuration data. Later, based on objective function, α, β, δ and ω positions are updated. The objective function, based on estimating the information measure that is correlated among the nodes, can be formulated as below,

$$Fit_{val} = (a_1 * b_1) * (a_2 * b_2) * (a_3 * b_3) \quad (1)$$

Where, $b_1 = \frac{\sum_{j=1}^{N} D_{ij}}{C_i^n}$

$b_2 = \frac{E_0}{C_i^n}$

$b_3 = \frac{1}{C_i^n}$

// b_1*,* b_2 *and* b_3 *are energy-based coefficients*

$a_1 = 0.5,\ \ a_2 = 0.2, a_3 = 1 - a_1 - a_2$ *// Coefficient parameters*

$D_{ij} = \sqrt{(X_i - X_j)^2}$

$C_i^n = size(C_n) \qquad if\ D_{ij} < Cover_{area}$

// C_i^n*- node number which are covered by* i^{th}*node*

// $Cover_{area}$*-coverage area of an* i^{th}*node*

The above fitness function is formulated based on distance and energy constraints.

The initialized ***α*, β, δ** *and* **ω** wolves along with scores and its positions are updated respectively using the following functions:

For alpha wolves,

$$\alpha_{wolves}{}^{score} = Fit_{val}, \alpha_{wolves}{}^{pos} = X_i \ \ , \quad if\ Fit_{val} < \alpha_{wolves}{}^{score} \quad (2)$$

For beta wolves,

$$\beta_{wolves}{}^{score} = Fit_{val}, \beta_{wolves}{}^{pos} = X_i \quad if\ Fit_{val} > \alpha_{wolves}{}^{score}\ \ \&\&\ Fit_{val} < \beta_{wolves}{}^{score} \quad (3)$$

For delta wolves,

$$if\ Fit_{val} > {\alpha_{wolves}}^{score}\ \ \&\&\ Fit_{val} > {\beta_{wolves}}^{score}\ \ \&\&\ Fit_{val} < {\delta_{wolves}}^{score}$$
$${\delta_{wolves}}^{score} = Fit_{val}, {\delta_{wolves}}^{pos} = X_i \qquad (4)$$

For omega wolves,

$$if\ Fit_{val} > {\alpha_{wolves}}^{score}\ \ \&\&\ Fit_{val} > {\beta_{wolves}}^{score}\ \ \&\&\ Fit_{val} > {\delta_{wolves}}^{score}\ \ \&\&\ Fit_{val} > {\omega_{wolves}}^{score}$$
$${\omega_{wolves}}^{score} = Fit_{val}, {\omega_{wolves}}^{pos} = X_i \qquad (5)$$

Where X_i represents position updation based on upper bound and lower bound, explained by equation (11). The overall updating of all types of wolves can be formulated as below,

$$X_1 = {\alpha_{wolves}}^{pos} - \left(\frac{\boldsymbol{A}}{|\boldsymbol{C} * ({\alpha_{wolves}}^{pos} - X_i)|}\right) \qquad (6)$$

Where $a = 2 - it * (\frac{2}{iter})$

$A = (2 * a * rand(1,1)) - a$

$C = 2 * rand(1,1)$

Similar process is applied for all wolves and X_2, X_3, X_4.

$$X_i = \frac{(X_1 + X_2 + X_3 + X_4)}{4} \qquad (7)$$

Finally, α position is converged and provides a solution which guarantees, the attacker cannot utilize the LLM_K from SINs instantly after deployment. The final solution, that contain designated SINs and normal node can be obtained by the below equation 8. SINs are estimated based on the updating wolves' positions, the process is repeated for all iterations by updating the wolves position,

$$Best_{Agent} = {\alpha_{wolves}}^{pos}\ \ {\alpha_{wolves}}^{score} > Fit_{val} \quad (8)$$

$$X_i = \frac{(X_1 + X_2 + X_3 + X_4)}{4}$$

The process of identifying SINs using GWO is given in algorithm 1.

Algorithm I: SIN Identification using GWO

Input: *Total number of nodes N*
Output: *Identified Master Node M_N, Normal Node N_N*

Procedure:
Step 1:
Initialize the wolves (Sensor Nodes) position randomly from, $X_i(i = 1,2,,\dots,N)$ $X_j(i = 1,2,,\dots,N)$
Maximum iteration, $iter = 500$
Lower Bound, $lb = -100$
Upper Bound, $ub = 100$

Initialize $\alpha_{wolves}{}^{score}$, $\beta_{wolves}{}^{score}$, $\delta_{wolves}{}^{score}$ *and* $\omega_{wolves}{}^{score}$

Step 2: *iterate the process to maximum iterations,*

$While\ it < iter$

$$Flag_{up}{}^{i} = \begin{cases} 1 & X_i > ub \\ 0 & else \end{cases}$$

$$Flag_{Lb}{}^{i} = \begin{cases} 1 & X_i < lb \\ 0 & else \end{cases}$$

Update the position based on the updating ub *and* lb

$$X_i = X_i * \left(Flag_{up} - Flag_{Lb}\right) + \left(ub * Flag_{up}\right) + (lb * Flag_{Lb}) \quad (11)$$

$$X_j = X_j * \left(Flag_{up} - Flag_{Lb}\right) + \left(ub * Flag_{up}\right) + (lb * Flag_{Lb})$$

Step 3:

Execute the objective function based on the estimation of information measure, correlation among the nodes, using equation (1).

Step 4:

Update **α, β, δ** *and* **ω** *wolves score and position, using equation (2 – 5).*

Step 5: *Update position of all wolves, using equation (6).*

Iterate the process for all wolves and estimate, using equation (7).

Step 6: *Best agents are estimated based on the updating wolves' positions, using equation (8).*

End while

Step 7: *This process is repeated for all iterations and the final solution identifies s*trategically important nodes (SIN).

B. Pair node prediction, volatile key (LLMK)and shared key generation using Secure Hash Algorithm (SHA):

The SINs and neighbor nodes, that are ready for transmission, are authenticated. The authentication is done through SHA, which involves generating the nonce message and shared key for the node which holds the volatile key and the node involved in communication. The volatile key has been generated for a SIN along with limited key's lifetime. Limited Life Volatile Master key (〖LLM〗 _K) generation can be formulated as below.

$$LLM_K = \left\{M_N{}^{id}, rand_{0\ to\ 1}, randint_{1\ to\ 10}\right\} \quad (12)$$

Where $M_N{}^{id}$ – *Master node Id,*

$rand$ – *random value generated between 0 to 1*

$randint$ - *random value generated between 1 to 10*

After volatile key generation, the master node sends the request message NReq to the adjacent neighbor nodes along with prime number which is selected from the prime number group (in table 2).

$Pair_{node} = N_N{}^{id}$ $\{neighbor\ node\ list\}$ *(13)*

where $N_N{}^{id}$ = Neighbor node.

Table. 2 Prime Number Groups

Group	Prime Numbers			
A	P1	P2	P3	P4
B	P1	P5	P6	P7
C	P2	P5	P8	P9
D	P3	P6	P8	P10
E	P4	P7	P9	P10

Further, among the various neighbor nodes around the SIN, only one node is designated as a pair node. A finite number of prime numbers are selected in a random manner and are partitioned into groups in such a manner that every intersection among two separate groups comprise of at least one prime number. Through this the process of key computation is distributed among two adjacent nodes, where each node S_a and S_b do randomly select one group. The intersection of these groups is represented as common prime number which are selected by S_a and S_b.

The neighbor node S_b generates nonce message by employing hash algorithm by considering the prime numbers. There exist five different available hash functions such as 160, 224, 256, 384, and 512 bit combinations. Out of these only one hash function is selected at any point of time.

$nonce_b = hash(X_b)$

The SHA is used for authentication and generates a cryptographic message digest algorithm which is similar to MD4 group of hash functions. A hash function can be defined as a function which takes the variable of size m as input and generates fixed-size string output, commonly known as hash value h (i.e. h=H(m)). Hash functions along with these properties have various computational uses. However, when it is employed in cryptography these functions requires, a) the input data 'x' of any length, b) The fixed length output, c) H(x) can be easy for computing some given value of x, d) H(x) is a unique function.

Algorithm II: Secure Hash Algorithm

Input: *Common Prime number,* ID_a *and* ID_b
Output: *Predicted* ***SINs*** *message security*

Procedure:
Step 1: *Initialize variables:*
m_{len}*=64*

m_{fact}*=1*

ρ_{pad}*=64 //pad zero*

h0 := 0x67452301

h1 := 0xEFCDAB89

h2 := 0x98BADCFE

h3 := 0x10325476

h4 := 0xC3D2E1F0

Step 2: *Pre-processing:*

ml = message length in bits (always a multiple of the number of bits in a character).

m_{size}*=512*(*m_{fact}*);*

m_d*=sign(rand(1,* m_{len}*)-0.5)+1)*0.5;*

for uu=1: m_{len}

$m_d(uu)$*=mod(uu,2)*

End

Let initialize the message input,

msg_{inp}*=message_input;*

for ii=1: $length(msg_{inp})$

$m_{str}(ii)$*=strcat(* $m_{str}, dec2bin(hex2dec(msg_{inp}(ii), 4)))$

End

m_{str}*=*$m_{str}(1: m_{len})$

m_{raw_512}*=char();*

blk_{num}*=fix(length(*m_{str}*)/512**m_{fact}*)*

If (mod(length(m_{str}*))),(512,* m_{fact}*)==0)*

blk_{num}*=*$blk_{num} - 1$

End

$exnd_{num}$*=512**m_{fact}**(*$blk_{num} + 1$*)-mod(length(*m_{str}*))*

for k=1:128(*$blk_{num} + 1$*)** m_{fact}

$m_{raw\_}512$*=strcat(*$m_{raw\_}512, dec2hex(bin2dec(m_{str})4 * (k - 1) + 1: 4 * k))$

end

*while (mod(diff_k,512**m_{fact}*))*

$int_{\_k}$*=*$int_{\_k}$*+1*

$diff_{\_k}$=$abs(int_{\_k}$+1-448*m_{fact}

Step 4: *Process the message m in consecutive 512bit chucks*

for each chunk

break chunk into sixteen 32-bit big-endian words w[i], 0 ≤ i ≤ 15;

Step 5: *Extend the sixteen 32-bit words into eighty 32-bit words:*

for i from 17 to 80

w[i] := (w[i-3] xor w[i-8] xor w[i-14] xor w[i-16]) leftrotate

Step 6: *Initialize hash value for this chunk:*

a := h0

b := h1

c := h2

d := h3

e := h4

Main loop:

for i from 0 to 80

if 0 ≤ i ≤ 20 then

f := (b and c) or ((not b) and d)

k := 0x5A827999

else if 21 ≤ i ≤ 40

f := b xor c xor d

k := 0x6ED9EBA1

else if 41 ≤ i ≤ 60

f := (b and c) or (b and d) or (c and d)

k := 0x8F1BBCDC

else if 61 ≤ i ≤ 80

f := b xor c xor d

k := 0xCA62C1D6

temp := (a leftrotate 5) + f + e + k + w[i]

e := d

d := c

c := b leftrotate 30

b := a

a := temp

Step 7: *Add this chunk's hash to result (Authenticated message) so far:*

h0 =mod(bin2dec(h0)+ bin2dec(a'),4294967296);

h1 = mod(bin2dec(h1)+ bin2dec(b'),4294967296);

h2 = mod(bin2dec(h2)+ bin2dec(c'),4294967296);

h3 = mod(bin2dec(h3)+ bin2dec(d'),4294967296);

h4 = mod(bin2dec(h4)+ bin2dec(e'),4294967296);

Ho= dec2hex(h0,32)

H1= dec2hex (h1,32)

H2= dec2hex (h2,32)

H3= dec2hex (h3,32)

H4= dec2hex (h4,32)

$A_{mes} = strcat(H0, H1, H2, H3, H4)$

Step 8: Verified authentication message by SIN node which is generated and received from the chosen pair node.

Shared key generation

Shared key is generated at S_b and is denoted as K_{ab}.

$$K_{ab} = f(nonce_b, ID_{a,} ID_b, S_a^x, S_a^y, S_b^x, S_b^y)$$

Where, $S_a^x, S_a^y, S_b^x, S_b^y$ are the two-dimensional coordinates of SIN node and pair node location respectively.

Pair node sends a response message NResp to the SIN along with hash function and selected prime number group.

$$NRep_b = \{hash_{func}, X_b\}$$

SIN upon receiving this response message, then generates nonce at S_a. Further, the shared key using generated nonce $nonce_a$ is generated using chosen hash function and common prime number.

$$nonce_a = hash(X_a)$$

Next, the SIN encrypts the volatile key LLM_k using generated shared key K_{ab}.

$$E(LLM_K) = E_f(K_{ab})$$

Key generation using gradient approach and Key sharing using Super-singular Isogeny Diffie Hellmann (SIDH):

Encrypted volatile key LLM_k is transmitted to S_b using super singular Diffie Hellmann algorithm.

SIDH:

This cryptographic algorithm has been utilized for establishing a secret key among dual parties under the uncertain and unreliable transmission channel. It has been designed in such a way that it resists cryptanalytic attack caused by an intruder. It claims as one of the small key sizes along with compression. The public key with 2699 bit is utilized at 128 bit quantum security level. This SIDH establishes a set of super singular elliptic curves along with their isogenies. An isogeny (ϕ) between elliptic curves E, E' can be defined in equation 14,

$$\phi: E \rightarrow E' \quad (14)$$

The SIDH can be set up as the prime of the form as,

$$p = l_A^{e_A}.l_B^{e_B}.f \mp 1 \qquad (15)$$

Where l_A, l_B - > small primes,

e_A, e_B -> Exponents,

f -> small co-factor

Thus, the curve has two large subgroups such as $E[l_A^{e_A}]$ and$E[l_B^{e_B}]$, that are assigned for paired nodes respectively. Each node starts the protocol by choosing a random cyclic subgroup (secretly) and compute the corresponding secret (isogeny). The efficiency of the SIDH can be analyzed during key exchange among the paired nodes, where each of them transmits information of two co-efficients (modp^2) which in turn defined two elliptic curve points and an elliptic curve. Each elliptic curve needs $log_2 p^2$ bits. Hence for two curve points, it amounts to 8$log_2 p^2$+4 bits, which is 6144 bits for 768 bit modulus p. But it can be reduced to 2640(330 bytes) using compression techniques. Along with compression methods, SIDH has same bandwidth constraint as 3072 – bit Diffie-Hellman exchange. Thus, it makes the SIDH applicable to schemes that have smaller memory space requirements.

Algorithm III: Key generation using SIDH and Key-sharing using SIDH

Input: *Authentication Message M, pair nodes Id* PN^{id}*, Pair node position* PN_X *and* PN_Y

Output: *Secret Key K*

Procedure:

Step1: *consider the authentication message as nonce.*

$Nonce = double(M)$ *// convert the authentication message into integer format.*

Step 2: *Combine this information and perform gradient computation,*

$C = \{PN^{id}, PN_X, PN_Y, Nonce\}$

$G = \frac{\partial C}{0.5}\, i + \frac{\partial C}{0.5}\, j \qquad (16)$

Step 3: *Form the secret key based on the unique estimated gradient,*

$K = unique(G)$

Step 4: *Share the generated secret key to the pair node via Super singular Isogeny Diffie Hellman algorithm,*

$Receiver_{key} = SIDH(K)$ *(17)*

Decrypt volatile key LLM_K at S_b,

$$D(LLM_K) = D_f(K_{ab})$$

D. Blow Fish Algorithm based generated Dynamic S-Box using Key

Blow Fish Algorithm:

Blowfish is said to be a 64 – bit block cipher, lightweight, and highly secured algorithm[20]. It can be implemented through software or hardware implementation. The confidentiality of key encryption is utmost significant in case of symmetric ciphers (i.e. Blowfish). Hardware implementation of blowfish has higher advantages than software implementation viz., rapid processing, minimal delay and extremely secured. It is one such employment of hardware which uses hard-wired components. In this blow fish algorithm, XOR operation is represented by the symbol ⊕. Almost 18 sub keys are required to execute this algorithm, along with non-linear function F. This F function utilizes 4 Dynamic S-Boxes or substitution boxes, each containing 256 [32-bit] entries. The S- boxes are generated using the algorithm IV.

Algorithm IV: Blow Fish Algorithm based generated Dynamic S-Boxes using Key K.

Input: *Secret Key K*
Output: P_{arr}, S_{box}

Procedure:

Step 1: *Convert hexadecimal key into decimal format,*

$K_1 = hex2dec(K)$ *// hexadecimal to decimal*

Step 2: *Convert decimal to binary format and concatenate each adjacent rows of the key element to horizontal arrangement,*

$K_2 = horzcat(dec2bin(K_1))$ *// $horzcat$ – horizontal wise concatenation of two vectors, $dec2bin$ – decimal to binary conversion.*

Step 3: *Initialize P-array and Dynamic S-Box using random values and updated key matrix,*

$P_{arr} = horzcat(dec2bin(hex2dec(rand)))$

$S_{box} = horzcat(dec2bin\left(hex2dec(Txt_{file})\right))$

$P_{arr} = P_{arr} \oplus K_2$ *// ⊕ - bitwise XOR operation*

$S_{box} = S_{box} \oplus K_2$

Generated s box with the dimension of $S_{box}[4][256]$

Blow Fish Encryption & Decryption:

Here the block size of the proposed cipher is 64 – bits, which will encrypt the block 'x' and then is also decrypted.

Algorithm V: Blow Fish Algorithm based Encryption and Decryption.

Input: *Plain text* Mes_{PT}
Output: *Encrypted Text* $Crypt_T$, *Decrypted Text* $Decrypt_T$

Step 1*: Perform blow fish encryption process,*
Let $L = Mes_{PT}(1:4)$ *// Consider 1 to 4 bits*
$R = Mes_{PT}(5:8)$ *// Consider 5 to 8 bits*
$for\ (int\ i = 0\ ;\ i < 16\ ;\ i\ +=\ 2)\ \{$
$L = L \oplus P_{arr}[i];$
$R = R \oplus Mes_{PT}(L);$
$R = R \oplus P_{arr}[i+1];$
$L = L \oplus Mes_{PT}(R);\ \}$
$L = L \oplus P_{arr}[16];$
$R = R \oplus P_{arr}[17];$
$Crypt_T = swap\ (L, R);$
Step 2: *Perform blow fish decryption process,*
Let $L = Mes_{PT}(1:4)$ *// Take 1 to 4 bits*
$R = Mes_{PT}(5:8)$ *// Take 5 to 8 bits*
$for\ (int\ i = 16\ ;\ i\ >\ 0\ ;\ i\ -=\ 2)\ \{$
$L = L \oplus P_{arr}[i+1];$
$R = R \oplus Mes_{PT}(L);$
$R = R \oplus P_{arr}[i];$
$L = L \oplus Mes_{PT}(R);\ \}$
$L = L \oplus P_{arr}[1];$
$R = R \oplus P_{arr}[0];$

$$Decrypt_T = swap\ (L, R);$$

It is noted that, these substitution-boxes are generated for every encryption, thus, the schemes dynamic nature increases the difficulty for the attacker in predicting the keys.

4. PERFORMANCE ANALYSIS

This section analyzes the proposed scheme's performance using the following key metrics:

- SIN (LLMK) nodes vs. Node Density: The impact of node density on the master key's (LLMK) lifetime, where the Sensor nodes are randomly deployed within an 300m*300m simulation area is investigated. Considering the three different random deployment scenarios, the LLMk is inserted into the central nodes and border nodes in the WSN. This enables to observe the scheme's behavior at time 't' and determine how node location affects execution time, security attributes and resource utilization.

- Delay Time vs. Number of Neighbor Nodes: This analysis explores the relationship between the number of neighboring nodes and the delay experienced in the key distribution process with varying node density.

- Energy Consumption Overhead: The energy consumption of the proposed scheme

compared to existing solutions is evaluated.

SIN vs. Node Density (Detailed Analysis):

Deployment Scenario 1

The first scenario focuses on randomly deploying the SNs and assessing the influence of parameters on the execution time of the proposed keying technique. Here, a low execution time is associated with the initial position and number of nodes holding the master key. As shown in figure 3, the proposed scheme efficiently performs the keying process across the entire network within 400ms. Additionally, out of 100 nodes, 94 successfully receive the volatile key, resulting in significantly higher key availability and key distribution in the WSN compared to the previous Self-VKS scheme [21].

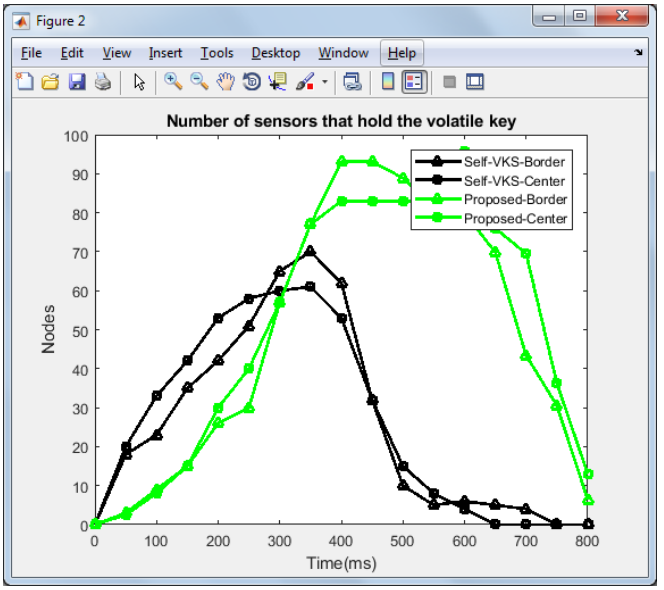

Figure 3 LLMK vs node density – deployment scenario #1

ii. Deployment Scenario 2

As illustrated in figure 4, the proposed scheme efficiently completes the keying process throughout the network within 410 milliseconds. Furthermore, 88 out of 100 nodes successfully receive the volatile key, achieving a superior level of key availability within the network compared to the prior Self-VKS scheme [21].

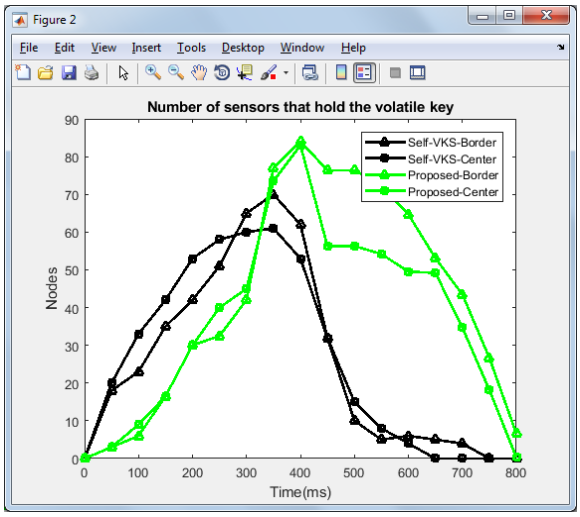

Figure 4. LLMK vs node density – deployment scenario #2

i. Deployment Scenario # 3

As shown in figure 5, the proposed scheme efficiently performs the keying process throughout the network within 405 milliseconds. Additionally, 91 out of 100 nodes successfully receive the volatile key, demonstrating a significantly higher level of key availability compared to the prior Self-VKS scheme [21].

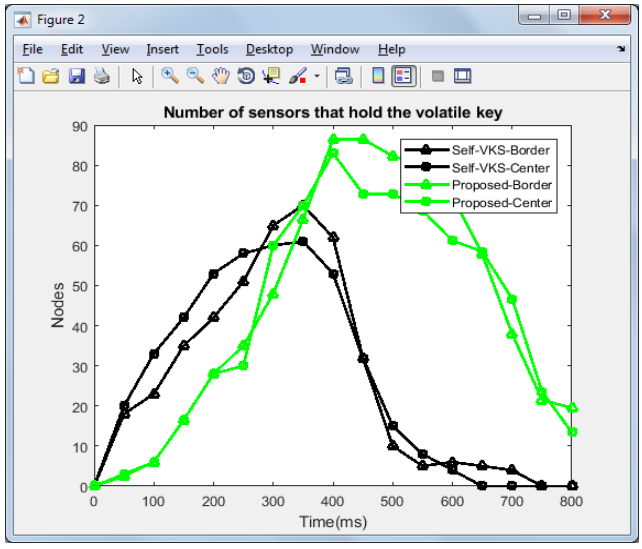

Figure 5. LLMK vs node density – deployment scenario #3

1) LLMK lifetime Vs node density

This section examines the impact of node density on the lifetime of volatile keys within the Wireless Sensor Network (WSN). Traditional approaches to secure communication in WSNs incur significant collision management overhead due to the broadcast nature of the

communication channel and processing time associated with volatile key communication. To address this challenge, prior works have resorted to extending the key lifetime to around 25 seconds [21].

In contrast, the proposed scheme eliminates the need for extended key lifetimes. Each node (ith node) can successfully communicate the volatile key to its 'n' neighboring nodes within the WSN before the key expires, achieving this feat within a 10-second time interval. Figures 6 to 8 visually depict the relationship between volatile key lifetime and time within the WSN with respect to the node density.

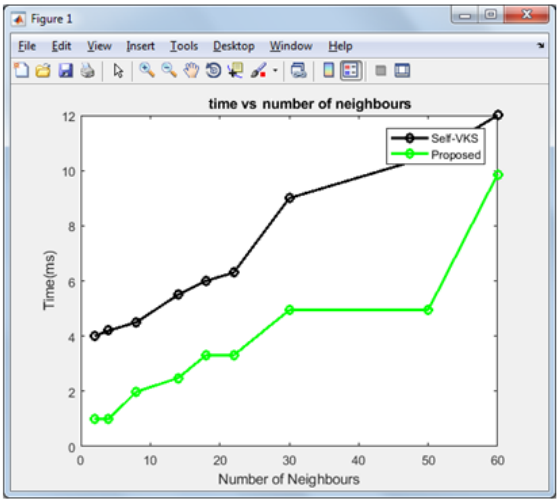

Figure 6. LLMK lifetime vs node density – for deployment scenario #1

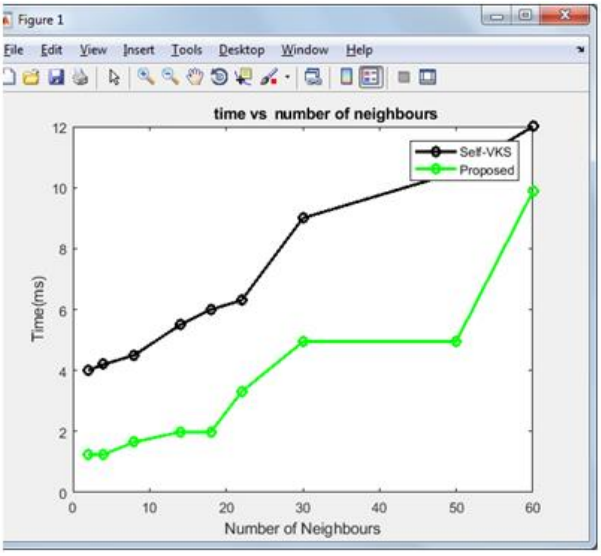

Figure 7. LLMK lifetime vs node density – for deployment scenario #2

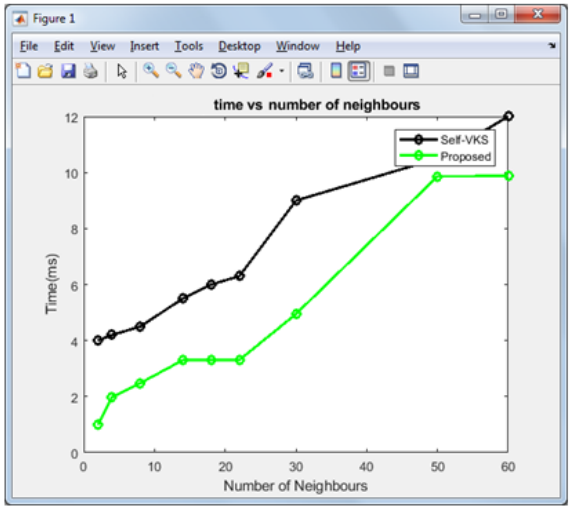

Figure 8. LLMK lifetime vs node density – for deployment scenario #3

2) Energy consumption overhead

Energy Efficiency Analysis:

The proposed scheme minimizes energy consumption during session establishment between SNs due to several factors. Unlike traditional approaches, not all SNs in the Wireless Sensor Network (WSN) need to generate and encrypt nonces before the keying process begins. This significantly reduces energy expenditure.

The total energy consumption for the network can be expressed as Em = (k + 1)N, where Em represents the energy spent as a function of the number of messages sent (N) and k denotes the average number of neighbors between each node pair [21]. This formula highlights the impact of several factors on energy efficiency:

- Node Density (N): As 'N' increases the total number of messages transmitted increase as well, thus resulting into spending more energy per node.
- Packet Size: Larger packet sizes contribute to increased energy consumption.
- Encryption/Decryption Cost: The choice of encryption and decryption algorithms and their key size significantly affect node energy and other resource usage.

During the keying process, each sensor transmits a discovery message and receives a maximum of k messages from its neighbors. Based on these calculations, the average energy consumption per sensor (Em) can be estimated as (k + 1) [21].

5. CONCLUSION

This proposal presents a novel key management scheme (KMS) designed for dynamically scalable, secure data communications in Wireless Sensor Networks (WSNs). The suggested deterministic technique makes use of an encrypted nonce that corresponds to a plain text

message. Because of this, nodes may effectively continue the key formation process even in situations when initially just a portion of nodes have the master key. Through this approach, the number of messages needed for key establishment is minimized, resulting in a considerable reduction of data transmission energy usage. In addition, capacity and processing demands on sensor nodes are substantially reduced. Critical sensor node behaviors including dynamic node movement, secure node addition, and key renewal are all fully covered by the method. The efficiency of the scheme in terms of keying time and robustness against different WSN assaults is shown by a thorough security study.

6. FUTURE SCOPE

Asynchronous Duty cycles:

It is suggested that asynchronous sensor duty cycles be investigated for optimal energy use and routing effectiveness.

Relevance to IoT:

The incorporation of IoT into WSNs increases the ubiquitous nature of sensor networks. Future study might be beneficial in evaluating the suggested key management, routing, and security systems in the context of Internet of Things applications with a consideration of devices with limited resources.

Considering Dynamic Networks:

The existing technique for managing symmetric volatile keys requires a static network design with fixed placements for sensor nodes. Future research should focus on creating and assessing these techniques for dynamic network topologies on mobility of nodes in heterogeneous environments.

Enhanced Security with Surveillance Nodes:

Adding "surveillance nodes" to the typical network deployment plan might bolster WSN security. These nodes would monitor network traffic, including potentially harmful behavior. Through the identification and elimination of bad actors, surveillance nodes establish connections only with securely configured nodes. Moreover, it is promising to use artificial intelligence techniques in these monitoring nodes for compromised node rehabilitation and recovery.

References

[1] A. Rani and S. Kumar, "A survey of security in wireless sensor networks," in 2017 3rd International Conference on Computational Intelligence & Communication Technology (CICT), 2017, pp. 1-5.

[2] F. Gandino, C. Celozzi, and M. Rebaudengo, "A key management scheme for mobile wireless sensor networks," Applied Sciences, vol. 7, p. 490, 2017.

[3] P. Singh and N. S. Gill, "A Survey on Key management Schemes in Wireless Ad Hoc Networks," International Journal of Applied Engineering Research, vol. 13, pp. 268-272, 2018.

[4] A. Laouid, M.-L. Messai, A. Bounceur, R. Euler, A. Dahmani, and A. Tari, "A dynamic and distributed key management scheme for wireless sensor networks," in Proceedings of the International Conference on Internet of things and Cloud Computing, 2016, p. 70.

[5] M. Elhoseny and A. E. Hassanien, "Secure data transmission in WSN: an overview," in Dynamic Wireless Sensor Networks, ed: Springer, 2019, pp. 115-143.
[6] O. Yagan and A. M. Makowski, "Wireless sensor networks under the random pairwise key predistribution scheme: Can resiliency be achieved with small key rings?," IEEE/ACM Transactions on Networking (TON), vol. 24, pp. 3383-3396, 2016.
[7] P. Ahlawat and M. Dave, "Deployment Based Attack Resistant Key Distribution with Non Overlapping Key Pools in WSN," Wireless Personal Communications, vol. 99, pp. 1541-1568, 2018.
[8] P. A. R. B. P. Shraddha Deshmukh, Harshad Nakade, "Implementation of Effective Key Management Strategy with Secure Data Aggregation in Dynamic Wireless Sensor Network," International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), vol. 4, pp. 358-364, March-April 2018.
[9] A. Albakri, L. Harn, and S. Song, "Hierarchical Key Management Scheme with Probabilistic Security in a Wireless Sensor Network (WSN)," Security and Communication Networks, vol. 2019, 2019.
[10] R. V. Saraswathi, L. P. Sree, and K. Anuradha, "Multi-stage Key Management Scheme for Cluster based WSN," International Journal of Communication Networks and Information Security, vol. 10, p. 552, 2018.
[11] K. Moara-Nkwe, Q. Shi, G. M. Lee, and M. H. Eiza, "A novel physical layer secure key generation and refreshment scheme for wireless sensor networks," IEEE Access, vol. 6, pp. 11374-11387, 2018.
[12] P. S. Mehra, M. N. Doja, and B. Alam, "Codeword Authenticated Key Exchange (CAKE) light weight secure routing protocol for WSN," International Journal of Communication Systems, vol. 32, p. e3879, 2019.
[13] U. Iqbal and S. Shafi, "A provable and secure key exchange protocol based on the elliptical curve diffe–hellman for wsn," in Advances in Big Data and Cloud Computing, ed: Springer, 2019, pp. 363-372.
[14] C. G. Krishnan, K. Sivakumar, and E. Manohar, "An Enhanced Method to Secure and Energy Effective Data Transfer in WSN using Hierarchical and Dynamic Elliptic Curve Cryptosystem," in 2018 International Conference on Smart Systems and Inventive Technology (ICSSIT), 2018, pp. 1-7.
[15] D. Suhag, S. S. Gaur, and A. Mohapatra, "A proposed scheme to achieve node authentication in military applications of wireless sensor network," Journal of Statistics and Management Systems, vol. 22, pp. 347-362, 2019.
[16] M. A. Simplicio Jr, M. V. Silva, R. C. Alves, and T. K. Shibata, "Lightweight and escrow-less authenticated key agreement for the internet of things," Computer Communications, vol. 98, pp. 43-51, 2017.
[17] C.-H. Liu and Y.-F. Chung, "Secure user authentication scheme for wireless healthcare sensor networks," Computers & Electrical Engineering, vol. 59, pp. 250-261, 2017.
[18] Q. Jiang, N. Kumar, J. Ma, J. Shen, D. He, and N. Chilamkurti, "A privacy-aware two-factor authentication protocol based on elliptic curve cryptography for wireless sensor networks," International Journal of Network Management, vol. 27, p. e1937, 2017.
[19] Y. Alkady, M. I. Habib, and R. Y. Rizk, "A new security protocol using hybrid cryptography algorithms," in 2013 9th International Computer Engineering Conference (ICENCO), 2013, pp. 109-115.
[20] M. Suresh and M. Neema, "Hardware implementation of blowfish algorithm for the secure data transmission in Internet of Things," Procedia Technology, vol. 25, pp. 248-255, 2016.
[21] A. Laouid et al., “A self-managing volatile key scheme for wireless sensor networks,” J. Ambient Intell. Humaniz. Comput., vol. 10, no. 9, pp. 3349–3364, 2019, doi: 10.1007/s12652-018-0772-9.