International Journal of Engineering Research and Development
e-ISSN: 2278-067X, p-ISSN: 2278-800X, www.ijerd.com
Volume 11, Issue 03 (March 2015), PP.01-16

SEAD: Source Encrypted Authentic Data for Wireless Sensor Networks

Lata B T[1], Vidya Rao[1], Sivasankari H[1], Tejaswi V[2], Shaila K[1], Venugopal K R[1], L M Patnaik[3]

[1]*Department of Computer Science and Engineering University Visvesvaraya College of Engineering, Bangalore University, Bangalore- 560 001.*
[2]*Department of CSE, National Institute of Technology, Surathkal, India.*
[3]*Honorary Professor, Indian Institute of Science, Bangalore, India.*

Abstract:- One of the critical issues in WSNs is providing security for the secret data in military applications. It is necessary to ensure data integrity and authentication for the source data and secure end-to-end path for data transmission. Mobile sinks are suitable for data collection and localization. Mobile sinks and sensor nodes communicate with each other using their public identity, which is prone to security attacks like sink replication and node replication attack. In this work, we have proposed Source Encrypted Authentic Data algorithm (SEAD) that hides the location of mobile sink from malicious nodes. The sensed data is encrypted utilizing symmetric encryption ---Advanced Encryption Standards (AES) and tracks the location of the mobile sink. When data encounters a malicious node in a path, then data transmission path is diverted through a secure path. SEAD uses public encryption ---Elliptic Curve Cryptography (ECC) to verify the authenticity of the data. Simulation results show that the proposed algorithm ensures data integrity and node authenticity against malicious nodes. Double encryption in the proposed algorithm produces better results in comparison with the existing algorithms.

Keywords:- Cryptography, Mobile Sink, Secure Routing, Wireless Sensor Networks, Advanced Encryption Standard, Elliptic Curve Cryptography.

I. INTRODUCTION

Wireless Sensor Network (WSNs) is an infrastructure having less, low cost, dynamic topology with tiny sensor nodes distributed across the region. The sensor nodes are capable of sensing, data processing and communicating. WSN is an emerging field of technological research with a wide range of applications, such as detecting and tracking the passage of troop and tanks in a battle field, environmental monitoring, and measuring traffic flow on road etc.

WSNs needs certain security covers to information and resources from attacks and misbehaviors, i.e., Authenticity, integrity, availability, non-repudiation and confidentiality. Among all, authentication and integrity are important principles in military applications.

It is vital to provide security for sensed data in WSNs. Selection of an appropriate cryptographic algorithm preserves the security of data. Cryptographic algorithm meets the constraints of sensor nodes such as memory, data size, processing time and battery. Symmetric key encryption and asymmetric key encryption techniques are commonly used in WSNs.

During symmetric key encryption, a single secret key is shared between the entities in a communication. Each entity uses shared key to encrypt and decrypt the data. In asymmetric key encryption, a pair of keys are used; they are public key and private key. Each node generates both the keys and distributes their public key among other nodes in a network by keeping their private key undisclosed. The sender encrypts the data using the receiver public key and at the receiver, the data is decrypted using receiver's private key.

One of the major applications of WSNs is military application, where the sensor nodes are deployed in unattended and vulnerable regions. Here, the sensor nodes are static and carry mission critical information that needs to be protected. These applications use Mobile Sink (MS) to collect the information from all the nodes. MS maximizes the lifetime of sensor networks as it moves within the range and collects the data from the nodes.

Sensor nodes communicate with MS using their public identities which results in node replication and sink replication attacks. When the network is compromised, malicious nodes replicate indefinitely and influences the network. The replicated malicious nodes hack the aggregated data and inject false data into the

network. Hence, it is essential to provide secure communication along the link with data integrity. Appropriate cryptographic schemes are needed to provide data integrity and authentication against malicious nodes.

A. WSN Architecture

A typical WSN environment is composed of a large number of small, low data rate and inexpensive nodes. Nodes are scattered in a controlled environment and interacts with the physical world. The objective of the sensor nodes are to collect specific type of data by monitoring and controlling the predefined areas. The data collected at each sensor node is transmitted to the base station (the sink/ the control station) *via* collaborative routing. In collaborative routing, the sensor node in the network behave like a *data source*, which senses specific data by interacting with the physical environment or like a *data router*, which transmits the processed data to its immediate neighboring nodes to reach the sink.

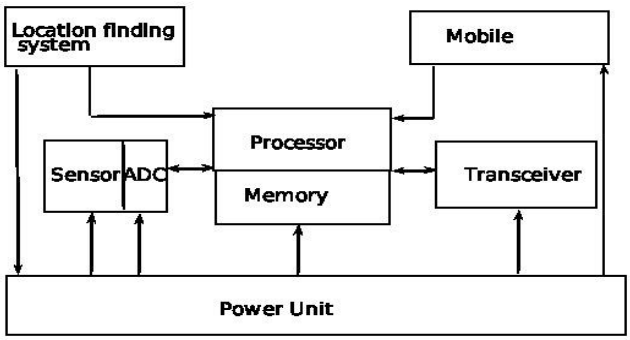

Fig. 1: Wireless Sensor Node Architecture

Figure 1, depicts the architecture of sensor node, which integrates software and hardware components for sensing, data processing and communication. These components are classified into transceiver unit, power unit, sensing unit and processing unit. They are embedded with application dependent components such as location finding system and mobilizing unit. The sensor interacts with physical environment and converts the data into digital signal by the analog-to-digital converter. The digital signals are fed into processing unit which transmits to the network through the transceiver unit.

Location finding system provides position of nodes which is essential for routing process. Nodes are identified by the node *ids*. Nodes can be moved to collect information from a desired area using mobile unit.

Motivation: Preserving confidentiality and ensuring integrity of data against malicious nodes is achieved through cryptographic algorithms. Symmetric Key Cryptography (SKC), Public Key Cryptography (PKC), hash functions etc, are different cryptographic algorithms [13]. The Advance Encryption Standard (AES) is widely used in WSNs as it employs single key which are shared among the two entities for encryption and decryption. In WSNs, these shared keys are exchanged through broadcast mode of communication which is vulnerable to attacks. Hence it is essential to secure the key exchange mechanism in WSNs against malicious entities.

Contribution: This paper proposes a light weighted cryptosystem called Source Encrypted Authentic Data (SEAD), a combination of AES and PKC. The authentic source node tracks the location of the mobile sink using the basic overhearing property of WSNs. Nodes in the network gets the sink location from Mobile Sink Tracking System (MSTS). In MSTS, it is assumed that all the nodes are aware of their position and the location of one hop neighbor. These neighboring nodes can overhear the transmitted packets even when they are not destined to them. Using this overhearing property, MSTS updates the location of mobile sink to the source node. Once the mobile sink is tracked, the source node encrypts the data using SEAD algorithm.

Organization: The rest of this paper is organized as follows. Related works are discussed in Section II. Background including concepts of malicious nodes and some of the cryptographic methods are described in Section III. System analysis and problem definition are presented in Section IV. The proposed algorithm SEAD and its mathematical model are developed in Section V and VI respectively. Finally, the performance analysis of the proposed algorithm is explained in section VII. Conclusions are presented in section VIII.

II. RELATED WORK

Data security is a challenge in WSN, as sensor network suffers from many constraints like, limitation of energy, low computational capability, small memory, and the use of unsecured communication channel.

Yong et al., [1] have discussed the above constraints along with security requirements and attacks with corresponding counter measures in WSNs. The attacks are classified under five categories like, cryptography, secure routing, secure data aggregation, intrusion detection and key management.

Jamal et al., [2] analyzed flat-based routing, hierarchical-based routing and location-based routing in WSNs. It addresses the issues on multipath query, negotiation of network architecture and quality based protocols.

Tung et al.., [3] presented a MoteSec-Aware protocol for security. Virtual Counter Manager (VCM) is implemented to detect replay and jamming attacks based on symmetric key cryptography using AES. It achieves high security by consuming less energy. Energy consumption is approximately proportional to the number of users. This work can be further enhanced to reduce storage overhead by using Bloom Filter for large-scale networks.

Pawan et al., [4] exploited public key nature protocols to define a hybrid key establishment algorithm for symmetric key cryptography. A certificate scheme has been proposed based on Elliptic Curve Cryptography for deriving pairwise link keys in WSNs. Though this scheme is more expensive than symmetric key algorithms, it is inherently secure due to the PKC (Public Key Cryptography) characteristics. This protocol is resilient to node compromise attacks. This work can provide higher security for mobile sensor nodes in massive scale networks by changing the content of the certificate.

Li et al., [5] introduced a scalable authentication scheme based on Elliptic Curve Cryptography which enables intermediate node authentication. It allows any node to transmit unlimited number of messages without suffering threshold problem. It provides message source privacy by incurring little communication overhead. It is more efficient in terms of computational and communication overhead compared with polynomial-based approach.

Nachiketh et al., [6] focus on battery life and energy requirements on cryptographic algorithms such as AES, ECC, RSA, SSL, Blowfish, SHA, SHA1, RC5, 3DES. They observe that asymmetric and hash algorithms have the highest and least energy cost, respectively and the energy consumption of symmetric algorithms depend on encryption/decryption cost and key size.

Noemie et al., [7] adopted the Differential Fault Analysis (DFA) techniques originally used on AES-128 in order to obtain the keys of AES-192 and AES-256 and based on a known DFA on key expansion. Ohyoung et al., [8] designed an optimal AES algorithm for low-power WSN node. It uses only one S-Box by which power consumption and logic usage is reduced compared with the block-wide and folded designs at the same throughput(1 Mbps). It satisfies the data rate requirements of the IEEE 802.15.4 standard.

Shi et al., [9], developed a Design-For-Secure-Test (DFST) technique for pipelined AES. It is a countermeasure to prevent secret information being leaked out of crypto hardware during test. It significantly reduces the application time, test data volume and test generation effort. Lu [10] proposed a key management scheme for WSN based on ECC and clustering by which computation speed has been improved.

Yun et al.., [11] presented a comprehensive survey of WSN security issues. Ho et al., [12] presented the design and implementation of a crypto processor, a special-purpose microprocessor optimized for the execution of cryptography algorithms such as AES, KASUMI, SEED, triple-DES, ECC, RSA crypto algorithms. It consists of a 32-bit RISC processor block and co-processor blocks. This work does not have resilence to side channel attacks in the private and public key cryptographics.

Saif et al., [13] designed a security protocol for ZigBee wireless sensor network in MAC layer. By generating 128-bits secrete key using AES algorithm over 128-bits of data, they have achieved security towards data. With this secrete key, they have mixed the Elliptic Curve Cryptography (ECC) algorithm which provides authentication over communication link. Through this parallel execution of AES and ECC, the scheme has provided both encryption process and authentication respectively against the cipher text and replay attacks. The results have failed to show same performance for larger data size.

Kishore et al., [14], used Elliptic Curve Cryptography (ECC) for data security in WSNs. They have generated a secret key using corresponding point on elliptic curve and this secret key is used to generate private key ring and is embedded into sensor nodes at the time of deployment. When two nodes share common private key, a link is established between them. The proposed system is checked based on various sensor network attacks like Sybil, brute force and random attacks. They have evaluated the performance based on connectivity and resilience against node capture.

Manju et al., [15] described the malicious node behaviour under Ad-Hoc network. They have surveyed methods to defend against these malicious nodes which includes security through cryptographic (Message Digest 5 (MD5), Digital Signatures, Secure Hash Algorithms etc), Trusted Third Party (TTP), secure protocols (Secure Aware Ad-Hoc Routing protocol (SAR), ARAN (Authenticated Routing for Ad Hoc Networks), Efficient security Ad-Hoc On-Demand distance vector(ES-AODV) etc) and Intrusion Detection System (IDS).

Sung et al., [16] depicted a neighbor-based malicious node detection scheme for WSN. This scheme is based on trustworthiness of sensor nodes during normal operation. Trustworthiness of sensor nodes depends on

confidence level which is analyzed based on occurrence of normal nodes and malicious nodes. In this scheme, detection and mis-detection rates are maintained with high and low values, respectively. It is shown that event detection accuracy is high while maintaining low false alarm rate. Calculating the confidence level would lead to increase of processing time.

Sung et al., [17] designed a dual threshold scheme to detect malicious node in a WSN. This scheme employs two thresholds to bridge the gap event detection accuracy and false alarm rate. They have analyzed their work with respect to Malicious Node Detection Rate (MDR), Mis-detection Rate (MR), False Alarm Rate (FAR), Event Detection Accuracy (EDA), Event Region Detection Rate (ERDR) and Boundary False Alarm Rate (BFAR). Results obtained for MDR, EDA, FAR, ERDR and BFAR shows that the proposed dual threshold system performs better than single threshold for detecting the malicious nodes. Though this technique bridges the gap between event detection and false rate, the selection of dual threshold values is crucial and it works for only static sink and not suitable for mobile sink.

Lin et al., [18] proposed concealed data aggregation scheme for multiple applications (CDAMA) in WSNs. The proposed scheme is designed as a multi-application environment in which the base station extracts application specific data from aggregated cipher text. CDAMA provides Concealed Data aggregation (CDA) between multiple groups and includes scalar multiplication on elliptic curve for encryption and decryption. Selecting a point on elliptic curve would always be a challenge as they have selected a constant value obtained by Broker's approach.

Du et al., [19] presented an efficient key management scheme for hybrid sensor network which utilizes the special communication pattern in sensor networks and ECC. They have proposed a centralized key management scheme where a server generates a pair of ECC public and private keys for each pair of low-end sensors. Each low-end sensor is preloaded with the private keys and each high end sensor are preloaded with public keys. Using these keys, the data could be encrypted and decrypted. The proposed scheme is compared with Eschenauer and Gligor (E-G) scheme: the comparison is done based on total storage, energy consumption and probability of an independent secure link being compromised under which ECC based key management shows better performance than E-G scheme. But the same scheme would consume more energy when homogeneous sensor network is considered.

Liu et al., [20] have proposed PKC based broadcast authentication scheme using signature amortization for WSN. The broadcast messages are authenticated using one signature.

III. BACKGROUND WORK

A. Malicious Node Behaviour

In WSN, malicious nodes are identified through the compromised behaviour of node which threatens the security principles (Confidentiality(C), Integrity (I), Availability (Av), Authenticity (Au) and Non-Repudiation (NR)). Malicious node generates incorrect readings which are not relevant [15], [16], [17]. The characteristic of malicious nodes are identified by the following characteristics:

- *Fake routing*: Whether there exists a path between nodes or not, a malicious node can send fake routes to the legitimate nodes in order to get the packets or to disturb the operations.
- *Stale Packets*: Malicious nodes create stale packets to create confusion in the network.
- *Message Tampering*: A malicious node alters the content of the packets.
- *Denying from Sending Message*: Malicious node denies from sending messages to the legitimate nodes.

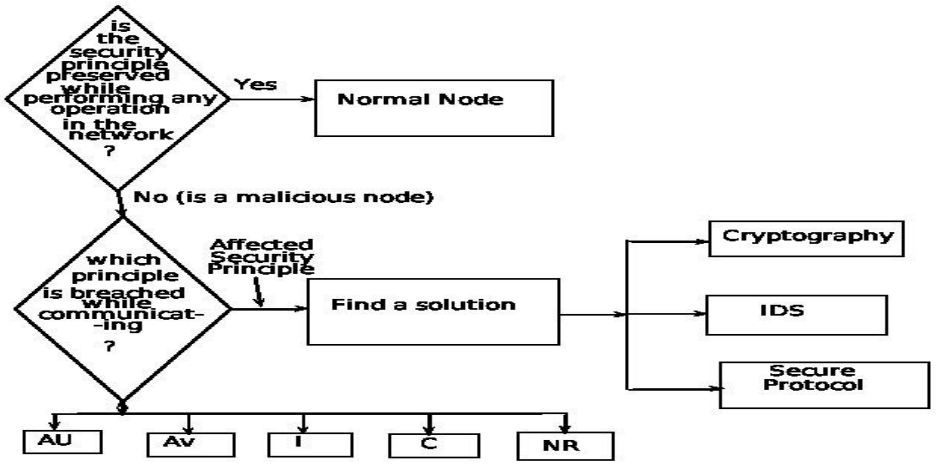

Fig. 2: Flowchart for Malicious Node Behaviour

The behaviour of a malicious node is explained in Figure 2.

B. Advance Encryption Standards (AES)

Advance Encryption Standard (AES) is a successor of symmetric encryption called Data Encryption Standard (DES). AES is based on the logic known as Substitution and Permutation Network (SPN). AES operates on 4x4 arrays of bytes and has a fixed block size of 128-bits and a flexible key size of 128, 192, or 256-bits. Encryption takes place in series of rounds, and each round have four stages as in Figure 3 [23]. These stages include:

- *Sub-byte generation*: Every byte in the plain text is replaced by a lookup table content called S-box table.
- *Shift-rows*: Every row in this stage is shifted cyclically for k-bytes and *k* depends on the key and the row number.
- *Mix-column*: Four bytes in each column is combined by linear mixing generated in the column.
- *Add-round key*: Each byte of the stage is combined with a round key; round key is different from secrete key, which is different for each round derived from Rijndale (*Rijndale* is a family of ciphers with different key and block sizes) key scheduler.

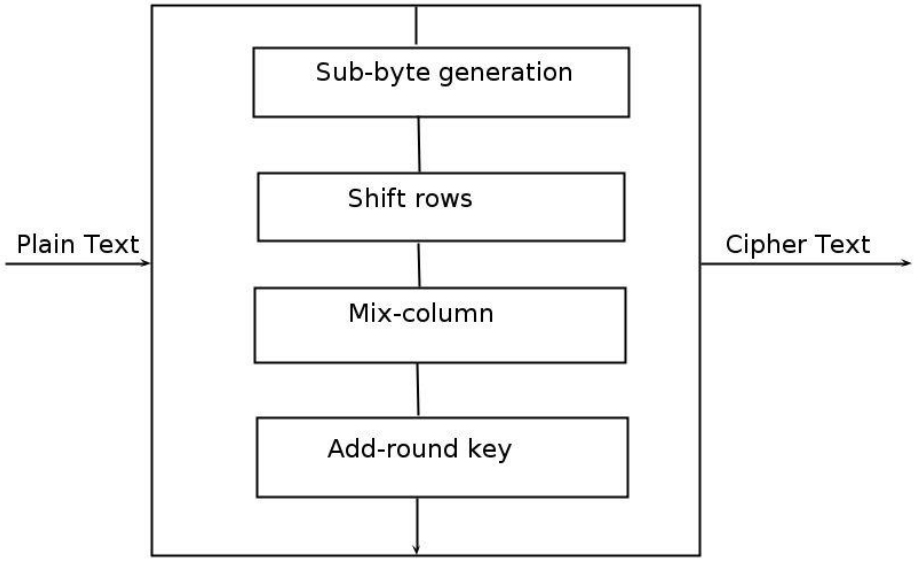

Fig. 3: AES round transformations

C. Elliptic Curve Cryptography (ECC)

Elliptic Curve Cryptography is a public key cryptography or asymmetric key cryptography that encrypts the data by particular individuals [13], [14], [18], [19], [20], [24]. ECC uses smaller key size to reduce the time required for encryption process. ECC is more compatible to sensor network as it consumes lower energy to generate keys.

Elliptic Curve Cryptography is an asymmetric or public key cryptography. ECC is mathematically defined over the elliptic curve $y^2=x^3 + ax + b$; where $4a^3 + 27b^2 !=0$; for each value of '*a*' and '*b*' there exist different elliptic curve. All points *(x, y)* which satisfies the above equation plus a point at infinity lie on the elliptic curve. The public key is obtained by multiplying the private key with the generator point G in the curve. The parameters of ECC include the generator point *G*, the curve parameter *'a'* and *'b'*, together with few more constants. The security is based on the difficultly of a different problem, which is called the Elliptic Curve Discrete Logarithm Problem (ECDLP).

ECDLP can be defined by considering two points *S* and *T* on an elliptic curve such that $k * S = T$, where *k* is a scalar. It is difficult to obtain *k*, for a given value of *S* and *T*. *k* is the discrete logarithm of *T* to the base *S*. multiplication of a scalar *k* with any point *S* on the curve to obtain another point *T* is called point multiplication on the curve which is the major operation in ECC.

D. Encryption using ECC

Sender *'A'* communicates to the receiver *'B'* by encrypting the data with public key of '*B'* which is known to all. Only '*B'* can decrypt the message with its private key. To encrypt and send a message Y_m to *B*, *A* chooses a random positive integer *k* and produces the cipher text C_m by using *B's* public key Y_B as shown below.

$$C_m = [k*G, Y_m + k*Y_B]$$

Where G is a point on elliptic curve defined over the Galios Field Eq(a,b) whose order is a large value *n*.

E. Decryption using ECC

To decrypt the cipher text, *B* multiples the first point in the pair by *B's* private key n_B and subtracts the result from the second point as shown by equation.

$$Y_m + k * Y_B - n_B (k*G)$$
$$= Y_m + k (n_B * G) - n_B (k*G)$$
$$= Y_m$$

A key exchange between users *A* and *B* can be explained in following steps:

1. A selects a positive integer $n_A < n$ as A's private key. where *n* is a set of random numbers.
2. A generates a public key $Y_A = n_A * G$ which is a point in *Eq(a, b)*.
3. B select an integer $n_B < n$ as B's private key.
4. B generates a public key $Y_B = n_B * G$ which is a point in *Eq(a, b)*.
5. Public keys are exchanged between A and B. A generates the secret key $k = n_A * Y_B$ and B generates the secret key $k = n_B * Y_A$

F. Shortest path routing:

Finding the shortest path is essential in WSN. Dijkstra invented an algorithm to find the shortest path throughout the network. Dijkstra's algorithm computes the least cost path from source node to all other nodes in the network graph *G = (V, E)*, with non-negative arc weight *V(i, j)*.

IV. SYSTEM MODEL AND PROBLEM DEFINITION

A. Network Model

In routing against malicious node, we assume that N sensor nodes are deployed randomly in the monitored region. The network is modeled as an undirected graph *G=(V,E)*, where $V = v_1, v_2, .. v_n$ is a set of sensor nodes. All nodes are connected when they are in the transmission range. *E* is the set of edges representing the connection between the nodes with respect to their communication range R_c. Event can occur anywhere in the monitored region having a radius of R_e, the sensor nodes present in the sensing region R_e forms a cluster C_i, where *i* indicates the node *id*. A source node N_s is elected from the cluster C_i, and forwards the encrypted data towards the mobile sink (MS) by tracing the position of MS.

B. Problem Definition

Wireless Sensor Network is considered as a graph *G=(V,E)* where *V* is the vertex/node set and *E* is an edge set. Given the cluster nodes set $C_i \in V$, the objective is to design a secure routing algorithm to ensure data integrity and authentication over communication channel. The proposed system Source Encrypted Authentic Data (SEAD) provides integrity and authentication to the sensed data and avoids anonymity among the keys of different source. *The objectives of this work are:*

1. To provide integrity to sensed data.
2. To route the data securely among authentic nodes.

Assumptions:

1. The sensor nodes are static and contain uniform energy of 50J.
2. Single sink node, which moves randomly inside the monitored region
3. Variable cryptographic key size (128, 192 and 256 bits).
4. Data packet size 128 bits.
5. Communication range of about 20m.
6. Number of rounds is four.

V. DESCRIPTION OF PROPOSED ALGORITHM

Source Encrypted Authentic Data (SEAD) forwards data through hop-by-hop forwarding and provides security against intruders in suspicious WSNs. It works in three phases they are (i) source election, (ii) encryption of source data with AES and (iii) authentication of data with ECC. SEAD uses three keys: (i) Public key for encryption/signature verification, (ii) Private Key for decryption/signature generation and (iii) Secrete key for data encryption.

The network is deployed with *(N-1)* number of nodes having equal amount of energy P_i and communication range R_c across 100X100 meters. The N^{th} node is assumed to be the sink, which is mobile in nature. When an event *S* occurs in the network, the nodes within the event range R_e forms a cluster C_i, where *i* indicate number of nodes within the cluster *C*. A source node Ns is selected among C_i which has more energy *i.e.,* $0 < P_i < P_t$. The N_s tracks the MS using tracking mechanism. Once MS is tracked, the N_s encrypts the communication data using secrete key encryption algorithm to obtain Source Encrypted Data (SED) which is a mixture of cipher text and last round key. N_s authenticates the communication link by executing Source Encrypted Authentic Data (SEAD). Each node in the shortest path route authenticates itself by executing SEAD. Finally SEAD reaches MS where the decryption is applied by reverse process.

A. Mobile Sink Tracking System (MSTS)

Overhearing property of wireless communication plays an important role in tracking the Mobile Sink (MS) [21]. Here it is assumed that, all nodes are static and a sink can move freely; each node is aware of their position and the location of one-hop neighbor. Two nodes can communicate with each other only if they have a bidirectional link between them. The neighboring nodes can over hear the transmitted packets even when they are not destined to them. This property is called overhearing property of nodes.

For example consider the Figure 4, it is assumed that the source node N_s has obtained location of MS by some destination location service and forwards data packets continuously to MS along the path $N_s \rightarrow N_1$ $N_1 \rightarrow N_2 \rightarrow N_3 \rightarrow MS$ by Geographic routing. Now consider N_s is the near dead node and wants to communicate with sink MS. The dashed circle represents the radio range of node. The MS broadcasts its Location Announcement Message (LAM) periodically to its neighboring nodes as it moves in the monitored region.

When MS moves to a new site as shown in Figure Fig. 4 (a), node N_3 can obtain the new location of MS by intercepting the Location Announcement Message (LAM). Thus node N_3 resets the new LAM message information of MS in the header of subsequently received data packet to the newly obtained location, and then forwards them to the MS. At the same time, the node N_2 which is in radio range of node N_3 can overhear the transmission from node N_3 to MS, and updates the MS location on its header as in Figure Fig. 4 (b). Similarly, node N1 obtains MS location by the node N_2 as node N_1 lies in node N_2's radio range. The LAM finally reaches the source node, where it gets the exact current location of MS and tries to forward the data to MS through the path $N_s \rightarrow N_1$ $N_1 \rightarrow N_2 \rightarrow N_3 \rightarrow MS$ as in Figure Fig. 4 (c) and (d).

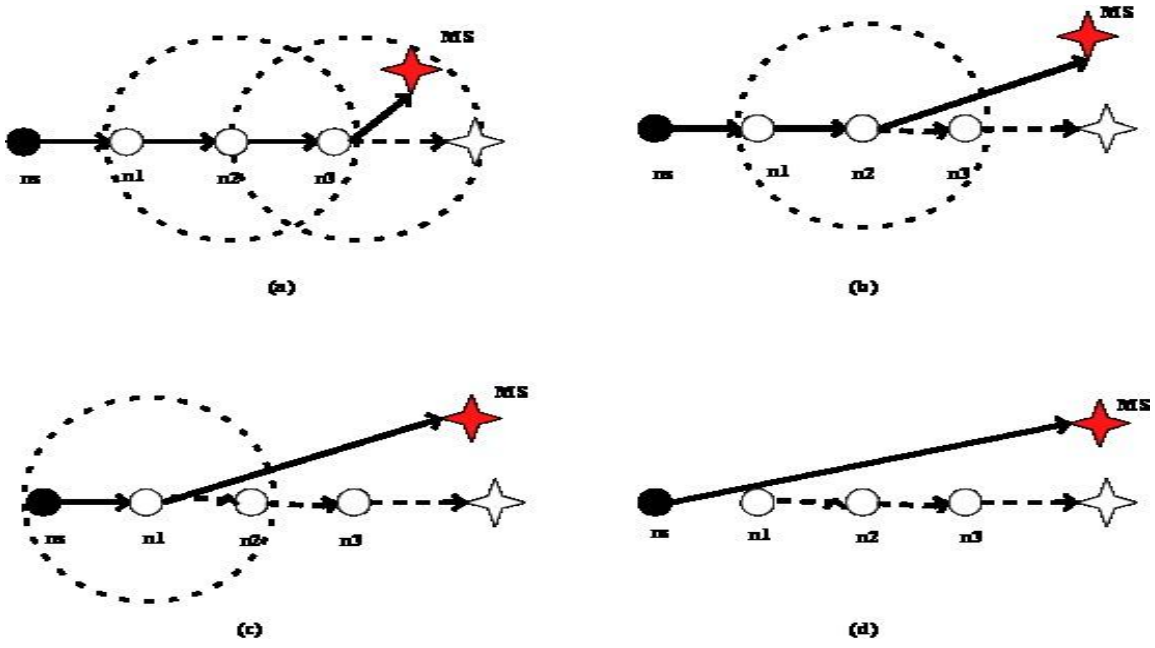

Fig. 4: Mobile Sink Tracking Scenario

B. Source Encrypted Authentic Data (SEAD)

In this SEAD algorithm there are two parts: first, selecting the source node from the cluster having more energy and secondly to provide security to data. The overview of SEAD is explained in the Figure 5 and Algorithm 1.

Initially, the sensor nodes are deployed randomly within the defined area. Each node stores its location and one hop neighbor location in their database. A Mobile Sink (MS) is deployed randomly which broadcast its location as it moves from one location to other.

In Figure 5, when event '*S*' occurs, the nodes in the vicinity of the event form a cluster. One of the 'i_{th}' node and which is responsible for forming the cluster is elected as Source Node N_s, which tracks the location of Mobile Sink(MS). The location of Source Node (N_s) and message will be encrypted by Advanced Encryption standard (AES) algorithm during the first round of encryption is called as Source Encrypted Data (SED). SED is encrypted by ECC during the second round of encryption which is called as Source Encrypted Authentic Data (SEAD). Doubly encrypted message is transmitted to the Mobile Sink. During the transmission phase, when it encounters malicious node, it changes the transmission path and tries to reach the MS. SEAD will be decrypted by the MS and decrypted message is saved at the MS.

In Algorithm 1, when an event (S) is generated, the nodes in the vicinity of the event form a cluster. A node is elected among this cluster as a source node which forwards the sensed data to the mobile sink. The source node has to secure the data before it has to be forwarded and hence encrypts the data using the Advance Encryption Standard (AES) as in phase II of Source Encrypted Authentic Data (SEAD) Algorithm. The sensed data follows four steps. Firstly, the sensed data is matched onto the predefined table and a new matched data is obtained. This new data is shifted cyclically row wise in second step. In step three, shifted data is again mixed linearly among the columns with 4-bytes at a time. Finally, the round key is *XOR*'d to obtain the cipher text.

To make the cipher data more secure, the last round key is mixed in-between the cipher text to obtain the Source Encrypted Data (SED) in phase II.

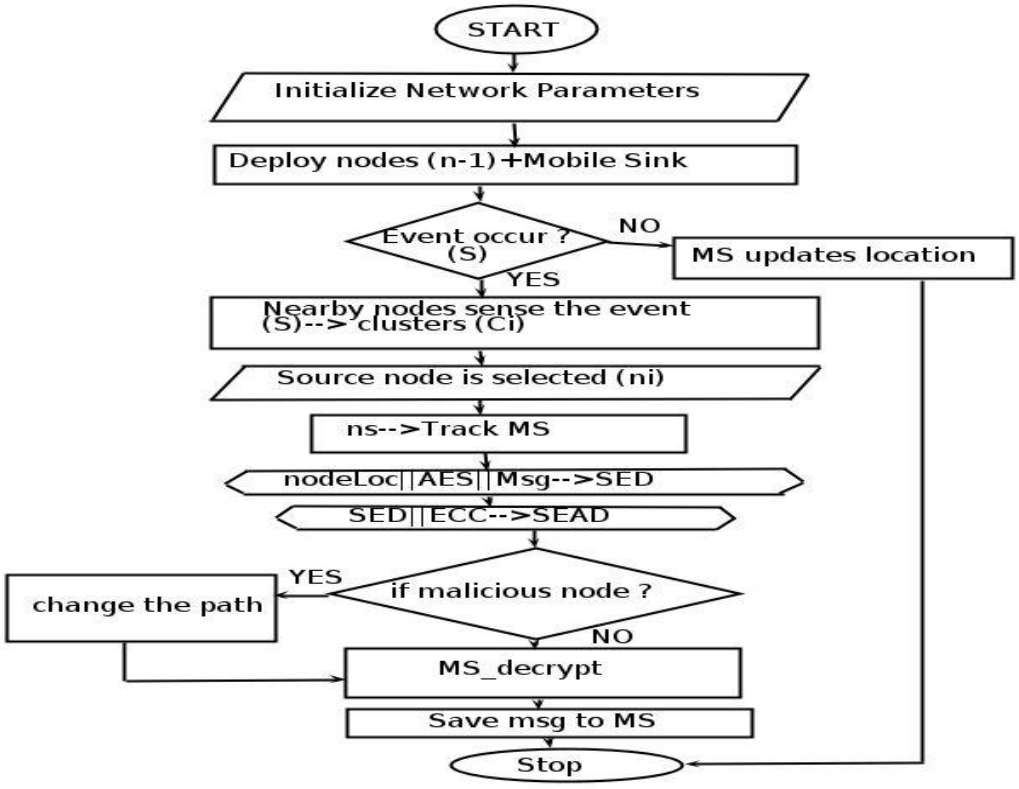

Fig. 5: Overview of Proposed SEAD System

In phase III, public and private keys are generated and the shared key is exchanged for authentication purpose. The public key is generated using private key of the node and a point on the elliptic curve whose order is *n*. A secret key is generated using this public key, and is shared among other nodes. At the time of second encryption, this shared secret key along with public key of receiver is used by sender to encrypt the SED to form a Source Encrypted Authentic Data (SEAD).

At the time of decryption, the private key of receiver is used to decrypt the SEAD to SED and decryption of AES is executed to obtain the secured data.

VI. MATHEMATICAL MODEL

Once the network is deployed, each node can communicate with its one hop neighbor if and only if the distance among them is less than their communication range *i.e.,* if nodes have overlapping communication range (R_c, then they can communicate with neighbors. This can be obtained as:

$$R_c > \sqrt{(x_1-x_2)^2 + (y_1-y_2)^2} \quad (1)$$

where (x_1, y_1) and (x_2, y_2) are the coordinates of the source node and its neighbor. When an event occurs, the nodes within the range of an event form a cluster \/ group. A source node is elected from this cluster, based on available energy of the node and it plays a role of forwarding the sensed data towards mobile sink. Mathematical equations for integrity, authentication and calculation of energy is explained in three phases. (i) encryption of source data, (ii) authentication of source data and (iii) energy model.

A. *Phase 1: Encryption of Source Data*

The source node tracks the location of sink based on over-hearing property as described in Mobile Sink Tracking System (MSTS). Once the sink traces the source node and it calculates the shortest route towards sink and forwards the sensed data by encrypting it.

1) Case 1: Generation of Cipher text

The sensed data undergoes AES encryption with four rounds. In the first round, the data is matched with a predefined block of data called S-box and it is replaced with S-Box data as in Figure 6.

After matching with S-box (Figure 6), the newly obtained sensed data called *state* is shifted row wise (Figure 7) and mixed column wise (Figure 8) as follows:

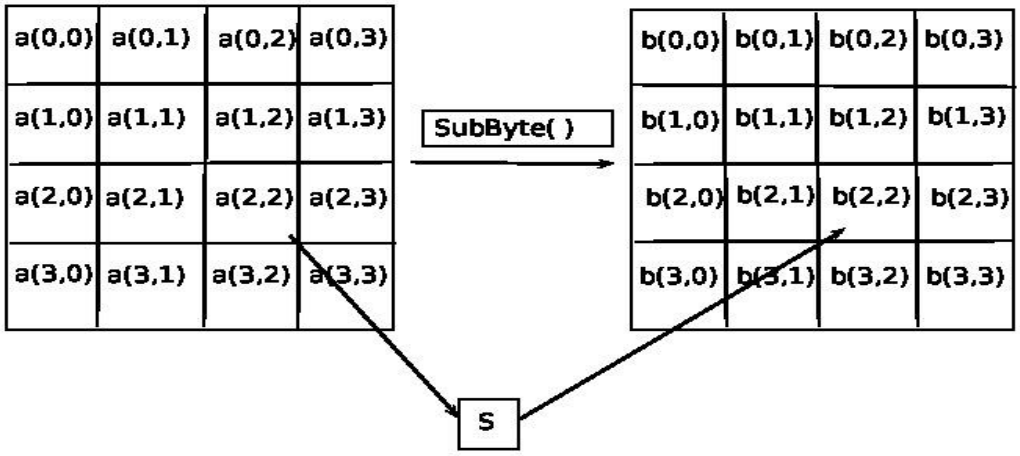

Fig. 6: Matching of Sensed Data with S-box

The Mix columns transformation operates on the State column-by-column, treating each column as a four-term polyn

$$b(x) = d(x) \oplus a(x) \quad (2)$$

$$\begin{pmatrix} b(0,1) \\ b(0,c) \\ b(0,2) \\ b(0,3) \end{pmatrix} = \begin{pmatrix} 01\ 02\ 03\ 01 \\ 02\ 03\ 01\ 01 \\ 01\ 01\ 02\ 03 \\ 03\ 01\ 01\ 02 \end{pmatrix} \begin{pmatrix} a(0,1) \\ a(0,c) \\ a(0,2) \\ a(0,3) \end{pmatrix}$$

Where *a(x)* is the state of previous step and *b(x)* is new state. As a result of this multiplication the four bytes are replaced as follows:

$$b_{(0,C)} = (02 \oplus a_{(0,C)}) \oplus (03 \oplus a_{(1,C)}) \oplus a_{(2,C)} \oplus a_{(3,C)}$$

$$b_{(1,C)} = a_{(0,C)} \oplus (02 \oplus a_{(1,C)}) \oplus (03 \oplus a_{(2,C)}) a_{(3,C)}$$

$$b_{(2,C)} = a_{(0,C)} \oplus (a_{(1,C)}) \oplus (02 \oplus a_{(2,C)}) \oplus (03 \oplus a_{(3,C)})$$

$$b_{(3,C)} = (03 \oplus a_{(0,C)}) \oplus (a_{(1,C)}) \oplus (a_{(2,C)}) \oplus (02 \oplus a_{(3,C)}) \quad (3)$$

The State *b(x)* is XOR'd with secret round key matrix to obtain the cipher text. In each round, a different secrete key is generated by random function and the cipher data is obtained. After the final round, the last round secrete key is mixed in-between the cipher data for better encryption. We insert the secrete key at every even position of the cipher data to obtain Source Encrypted Authentic Data.

$$(\text{Secrete\_key} \,\%\, 2, \text{cipher}) \geq \text{SED} \quad (4)$$

The SED is encrypted using Elliptic Curve Cryptography to provide authentication over the cipher data. This is done by assuming, S and T as two points on elliptic curve, defined by the Elliptic Curve Discrete Logarithm Problem (ECDLP) as

$$K * S = T = S + S + S... \text{ k times } = O \quad (5)$$

O is defined as point of infinity.

Definition of Elliptic Curve: Let $p > 3$ be a prime. Let $a, b \in Z_p$ be constants such that $4a^3 + 27b^2 != 0 \bmod p$. A non-singular elliptic curve is the set E of solutions $(u, v) \in Z_p X Z_p$ to the equation

$$v^2 \bmod p = u^3 + a * u + b \bmod p \quad (6)$$

together with a special point *O* called the point at infinity.

The same formulas can be used to define addition. (E, +) forms a addition group. We denote this group as E(GF(q)). To determine elements of E(GF(q)), we have to try all possible $u \in Z_p$, compute $u^3 + a * u + b \bmod p$ and then find if the resulting value is a quadratic residue of $\bmod p$.

2) Case 2: Public Key Generation

The source node generates a random key $n_{SA} < n$ as its private key (i) and calculates its public key as (ii) $Y_{SA} = n_{SA} * \beta$, where (iii) β is called set of generator point on the elliptic curve with n as its order. The next node/hop in the path selects as random $n_{SB} < n$ as its private key and generates its public key as $Y_{SB} = n_{SB} * \beta$.

3) Case 3: Key Exchange Mechanism

The source node computes its shared key as

$$K = Y_{SA} * n_{SB} \quad (7)$$

The next node in the path computes its shared key as,

$$K = Y_{SB} * n_{SA} \quad (8)$$

4) Case 4: Encryption Process

Encryption of data i.e., SED from source node N_s to next hope is shown as

$$SEAD = [\, K \oplus \beta, SED + K \oplus Y_{SB} \,] \quad (9)$$

Here the source node N_s uses the public key of next hop to obtain the Source Encrypted Authentic Data (SEAD).

5) Case 5: Decryption Process

SEAD can be decrypted by substituting the product of first point on pair G from the second point with the private key n_{SB}.

$$\begin{aligned} \text{SEAD} &= \text{SED} + \text{K} \oplus (n_{SB} \oplus \beta)\, n_{SB}\, (\text{K} \oplus \beta) \\ &= \text{SED} \end{aligned} \tag{10}$$

Finally, the SED undergoes the reverse process of AES to decrypt and to obtain secured sensed data information at mobile sink.

After matching with S-box, the newly obtained sensed data called State is shifted row wise (Figure 7) and mixed column wise (Figure 8).

B. Energy Model

Total energy total dissipated P_{Tdiss} at a sensor node is attributed to four basic energy consumption sources: (i) energy used for sensing P_{sense}, (ii) transmission P_{trans}, (iii) reception P_{recp} and (iv) encryption $P_{encrypt}$. Here all the nodes are homogeneous in nature and power consumed for sensing P_{sense} and reception P_{recp} is same for all the nodes. The transmitting and receiving power of MS are higher than other nodes because of the additional data processing and aggregation task associated with it. The total energy dissipated by a sensor node is given as

$$P_{Tdiss} = P_{sense} + P_{trans} + P_{recp} + P_{encrypt} \tag{10}$$

The energy dissipated for transmission depends on the distance between the transmitter and receiver nodes which are located at $N_t(X_t,Y_t)$ *and* $N_r(X_r,Y_r)$ respectively. *i.e.,*

$$\text{Dist} = \sqrt{((X_t,X_r)^2 + (Y_t,Y_r)^2)} \tag{12}$$

Through distance formula, we can calculate energy dissipated at transmission as,

$$P_{trans} = P_i * \text{Dist} \tag{13}$$

where, P_i is the available energy at the node *"i"*. The energy consumed during encryption, depends on size of the encryption key K_{size}, size of data S_{data}, amount of available energy at the node, time taken to encrypt T_{encryp} and total number of rounds in AES encryption (*NOR = 4*).

$$P_{encrypt} = (S_{data} + K_{size}) * P_i * T_{encryp} * \text{NOR} \tag{14}$$

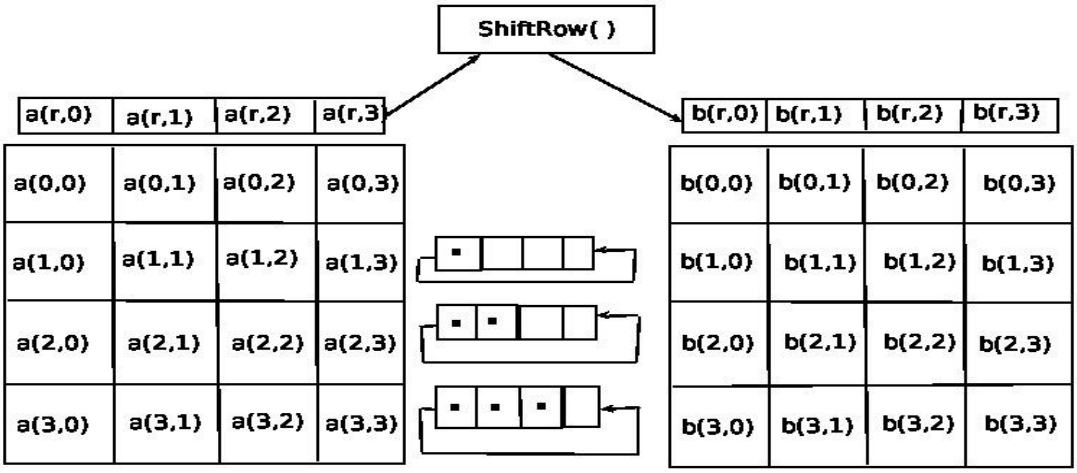

Fig. 7: Row-shift of Sensed Data

Table I: Notations

Symbols	Definitions	Symbols	Definitions
N	Total Number of Sensor Nodes Deployed	MS	Mobile Sink
P_i	Available Energy at node i	P_{min}	Threshold Energy of Each Node
P_t	Average Energy of WSN	S	Event
R_c	Communication Range of Sensor Nodes	R_e	Sensing Range of event
C_i	Cluster with I number of Nodes	N_s	Source node from the cluster
LAM	Location announcement message	R	Number of round in symmetric encryption
n_{SA} & n_{SB}	Private keys of adjacent nodes	Y_{SA} & Y_{SB}	Public keys of adjacent nodes
β	A point on elliptic curve with an order of n	P_{sense}	Sensing Power
P_{recep}	Reception Power	P_{Tdis}	Total Energy Dissipated
$P_{encrypt}$	Encryption Power	P_{trans}	Transmission Power
NOR	Number of Rounds		

Table II: Energy Dissipation in SEAD Based on Different Key Size

Energy Dissipated(nJ)	Key\ Size (bits)	Encryption Time(ms)
6.452	128	1.356
10.167	192	2.054
13.107	256	2.350

VII. IMPLEMENTATION AND PERFORMANCE ANALYSIS

In MATLAB simulation setup, 100 nodes are deployed randomly within the area of 100m X 100m; among them we have assumed to have 20 malicious nodes. The sink is made to move randomly within the defined region. The simulation set up is varied from 10, 20, 30, ..., 100 nodes with a single mobile sink.

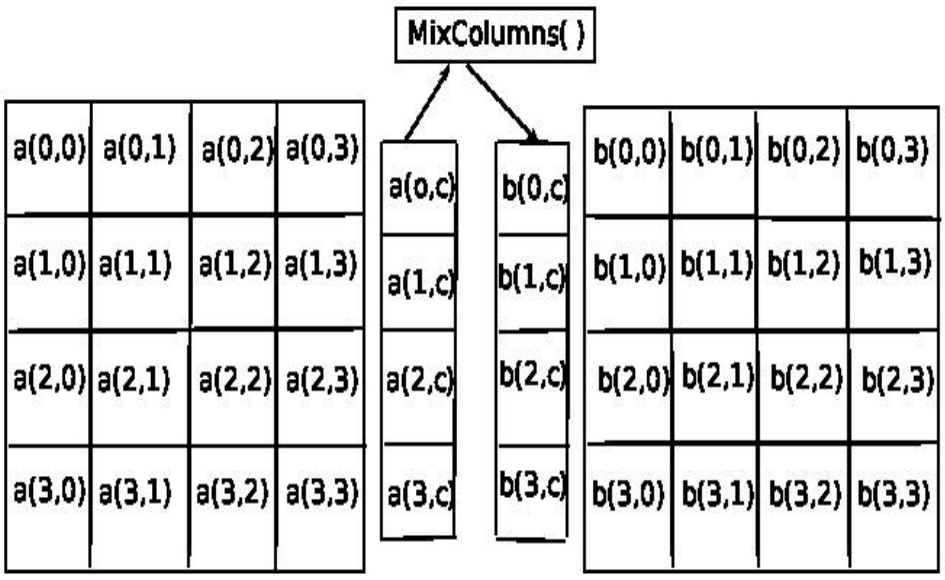

Fig. 8: Column Mixing of State

Table III: Simulation Parameters

Parameter Type	Test values
Number of Authentic Nodes	80
Number of Malicious Nodes	20
Data Packet Size	128 Bits
Encryption Key Size	128 Bits
Sensor Node Transmission Range	20mtr
Sensor Node	Static
Sink Location	Mobile

The communication path between source and destination is chosen randomly. Path is changed when it encounters a malicious node. Initially, the data size and key size is taken as 128-bits, which is later varied to check the performance.

The malicious node generates incorrect reading and misleads the report generated with respect to the event detection accuracy. It is observed that the malicious nodes consume more energy and minimizes the network lifetime. Figure 9 infers the relationship between number of nodes in the network and number of malicious nodes in the communication path. It shows that the path is hacked maximum number of times in ECC algorithm than AES algorithm. SEAD shows minimized number of occurrences of malicious nodes in the communication path.

Table IV: Decryption time of SEAD with variable key size

Key Size	Decryption Time(ms)
128	1.35
192	2.10
256	2.40

Table IV presents the decryption time of SEAD for variable key size. The time taken to decrypt the data with key size of 128-bits, 192-bits and 256-bits is 1.35ms, 2.10ms and 2.40ms respectively.

Table V: Crypt Analysis with Respect to Different Algorithms

Algothims	Encryption Time(ms)	Decryption Time(ms)
AES	1.32	1.25
ECC	0.95	0.48
SEAD	1.6	1.46

The time taken to encrypt and decrypt the data at the node level based on SEAD, AES and ECC algorithm is shown in Table V. It is observed that to encrypt the 128-bits data, the AES encryption takes about 1.32ms and ECC encryption takes 0.95ms and SEAD requires 1.6ms to encrypt the same data. The decryption of 128-bits data using AES and ECC takes about 1.25ms and 0.48ms respectively. Whereas SEAD takes about 1.46ms to decrypt the same data. The encryption and decryption of data using SEAD takes maximum time when compared to other algorithms but is more secure than AES and ECC.

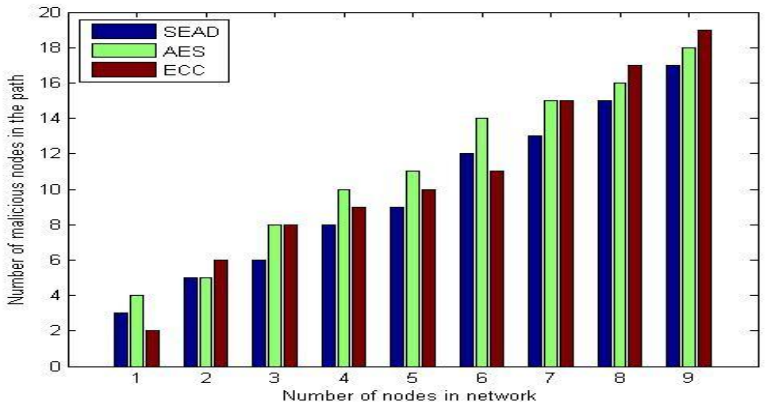

Fig. 9: Occurrences of malicious nodes in the communication path

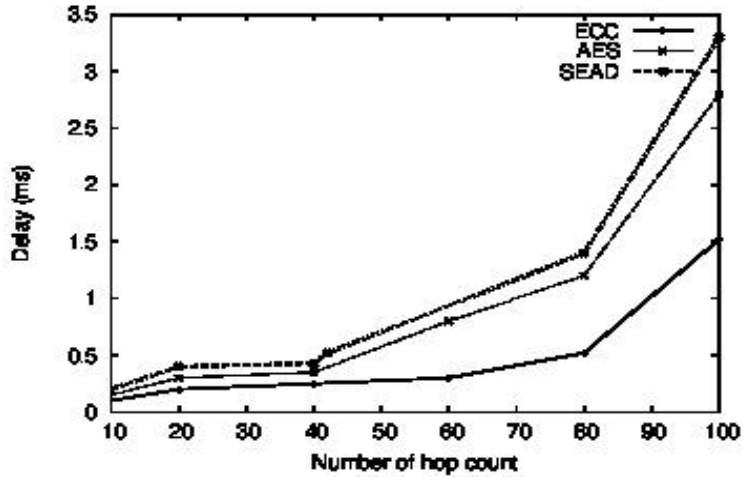

Fig. 10:Time delay in network for AES, ECC and SEAD.

Algorithm 1: SEAD: Source Encrypted Authentic Data.

Input: Total number of nodes, Location of the nodes;
Output: Secure Routing
Source Encrypted Authentic Data (SEAD)

Phase I:

```
begin
    Initialize N, Pmin to record the coordinates of
    sensor nodes and save the location of nodes
    N = 0;
    Pmin = 0
    Ni,x = Ni,y = location
    if (event == true) then
        event = S
       Ci = nearest(S)
          if (Pi > Pmin) then
             Ns = Ni
```

Phase II: Encryption of Source Data with AES

```
begin
        Initialize rounds (R = 4) and event;
        predefinedtable[ ] = sensed data[ ];
        void SEAD(plaintext, exp cipher, Key);
        state = plaintext; Nb = exp cipher;
        for R = 0; R _ round; R + +; do
                AddRoundKey(state, key, 0, Nb);
                Subytes(State);
                ShiftRows(State);
                MixColumn(State);
                AddRoundKey(State, key, rand*Nb, (R+1)*Nb-1);

        SubByte(State);
        ShiftRow(state);
        AddRoundKey(State, key, Nr*Nb, (Nr+1)*Nb-1);
        out = state;
        return out;
```

Phase III: Authentication of Source Data with ECC

begin

Select Source Encrypted Data (SED);
$n_{SA} = n_{SB}$ = Privatekeysofcorrespondingnodes.
β = nth order of a point on elliptic curve;
n = smallest positive integer on curve;
begin
public key generation.
$Y_{SA} = n_{SA} * \beta$;
$Y_{SB} = n_{SB*} \beta$;
Key exchange.
A secret key (K) is generated using public and private keys;
Secret key is exchanged among the nodes;
Encryption at source;
SED is again encrypted using secret key and public key of destination node = SEAD;
decryption at destination; SEAD is decrypted back to SED using private key of destination

Figure 10 describes the comparison between SEAD, AES and ECC for network delay. The total delay in the network when AES and ECC are used is approximately 2.8ms and 1.3ms for 100 nodes. As the SEAD is a combination of AES and ECC, the delay is 3.3ms for 100 nodes. It is observed that the delay in network is more when SEAD is implemented. Though the SEAD algorithm is slower it is more secure than AES and ECC.

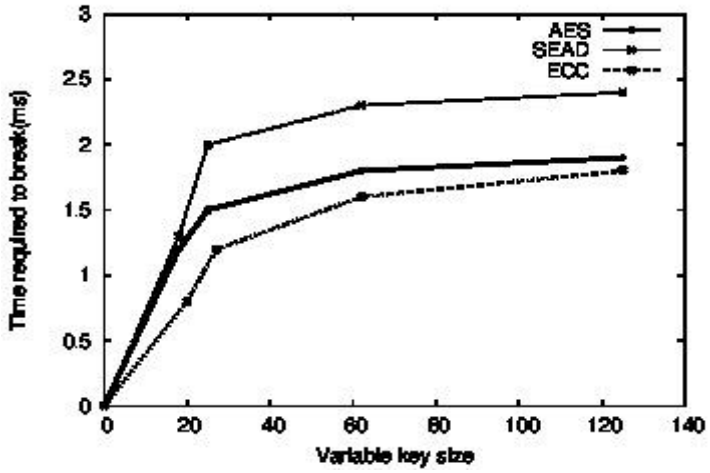

Fig. 11: Cryptanalysis of network with respect to SEAD, AES and ECC.

Figure 11 depicts the time required to break the variable size data for SEAD, AES and ECC encryption algorithm. It shows that to break a 128-bits of data, AES takes about 1.6ms, ECC takes 1.4ms and SEAD requires 2.3ms to break a data of size 128-bits. The performance of crypt analysis using SEAD is much better than AES and ECC.

VIII. CONCLUSIONS

Wireless Sensor Networks are constrained with limited battery, lifetime and security issues. In this paper, we propose Source Encrypted Authentic Data (SEAD) algorithm to provide confidentiality and authentication for secure routing. A mathematical model is proposed to generate a private key. The proposed algorithm follows double encryption using Advanced Encryption Standard (AES) and Elliptical Curve Cryptography (ECC) to ensure data integrity and authentication respectively. Each source node checks their neighbor through the node identifier. When the source node encounters the malicious node in the selected path, it changes the existing path for the successful transmission of data. Though the time and energy consumption of the proposed algorithm is higher, it is inevitable where the security of the data is crucial. Security algorithm with lower energy consumption and delay for the same level of security has to be addressed in future work.

REFERENCES

[1]. Y. Wang, G. Attebury and B. Ramamurthy, "A Survey of Security Issues in Wireless Sensor Networks," IEEE Communications and Survey Tutorials, 2nd Quarter, vol. 6, no. 2, pp. 1-15, 2006.

[2]. J. N. Al-Karaki, and A. E. Kamal, "Routing Techniques in Wireless Sensor Networks: A Survey," ICUBE initiative of Iowa State University, Ames, IA 50011.

[3]. Y. Tsou, C. Lu, and S. Kuo, "MoteSec-Aware: A Practical Secure Mechanism for Wireless Sensor Networks," IEEE Transactions on Wireless Communications, vol. 12, no. 6, pp. 2817-2829, June 2013.

[4]. P. Porambage, Pardeep Kumar, C. Schmitt, A. Gurtov and M. Ylianttila, "Certificate-Based Pairwise Key Establishment Protocol for Wireless Sensor Networks," IEEE, 16th International Conference on Computational Science and Engineering (CSE), pp. 667-674, 2013.

[5]. Y. Li, J. Li, J. Ren and J. Wu, "Providing Hop-by-Hop Authentication and Source Privacy in Wireless Sensor Networks", 31st Annual IEEE International Conference on Computer Communications, pp. 3353-3357, 2012.

[6]. N. R. Potlapally, S. Ravi, A. Raghunathan and N. K. Jha, "A Study of the Energy Consumption Characteristics of Cryptographic Algorithms and Security Protocols," IEEE Transactions on Mobile Computing, vol. 5, no. 2, pp. 128-143, Feb. 2006.

[7]. N. Floissac and Y. L'Hyver, "From AES-128 to AES-192 and AES-256, How to Adopt Differential Fault Analysis Attacks," IEEE Workshop on Fault Dianosis and Tolerance in Cryptography (FDTC), Nara, Japan, pp. 43-53, Sept. 2011.

[8]. O. Song and J. Kim, "Compact Design of the Advanced Encryption Standard Algorithms for IEEE 802.15.4 Devices," Journal of Electrical Engineering and Technology , vol. 6, no. 3, pp. 418-422, 2011.

[9]. Y. Shi, N. Togawa, M. Yanagisawa and T. Ohtsuki, "Design for Secure Test - A Case Study on Pipelined Advanced Encryption Standard," IEEE International Symposium on Circuits and Systems (ISCAS), Southeastern Louisiona, US, pp. 142-152, May 2007.

[10]. M. Lu, "Study on Secret Key Management Project of WSN Based on ECC," Journal of Networks, vol. 7, no. 4, pp. 652-659, April 2012.

[11]. Y. Zhou, Y. Fang, and Y. Zhang, "Securing Wireless Sensor Networks: A Survey," IEEE Commmunications, vol. 10, no. 3, pp. 6-28, 2008.

[12]. H. W. Kim, and Sunggu, "Design and Implementation of a Private and Public Key Crypto Processor and Its Application to a Security System," IEEE Transactions on Consumer Electronics, vol. 50, no. 1, pp. 214-224, Feb. 2014.

[13]. Saif A., Z. Ahmed, A. Abdullah and S. Subramiam, "AES and ECC mixed for ZigBee Wireless Sensor Networks," World Academy of Science, Engineering and Technology, 2011.

[14]. K. Rajendiran, R. Sankararajan, and R. Palaniappan, "A Secure Key Predistribution Scheme for WSN using Elliptic Curve Cryptography," ETRI Journal, vol. 33, no. 5, Oct. 2011.

[15]. R. Saini and M. Khari, "Defining Malicious Behavior of a Node and its Defensive Methods in Ad Hoc Network", International Journal of Computer Applications, vol. 20, no. 4, pp. 1-20, Apr. 2011.

[16]. S. Yim and Y. Choi, "Neighbor-Based Malicious Node Detection in Wireless Sensor Networks", Scientific Research on Wireless Sensor Network, vol. 4, pp. 219-225, Sept. 2012.

[17]. S. Lim and Y. Choi, "Malicious Node Detection using a Dual Threshold in Wireless Sensor Networks", Journal of Sensor and Actuator Network, vol. 2, pp. 70-84, Feb. 2013.

[18]. Y. Lin, S. Chang and H. Sun, "CDAMA: Concealed Data Aggregation Scheme for Multiple Applications in Wireless Sensor Networks", IEEE Transactions on Knowledge and Data Engineering, vol. 25, no. 7, pp. 791-801, July 2013.

[19]. X. Du, M. Guizani, Y. Xiao and H. Chen, "A Routing-Driven Elliptic Curve Cryptography Based Key Management Scheme for Heterogeneous Sensor Networks", IEEE Transactions on Wireless Communication, vol. 8, no. 3, March 2009.

[20]. Y. Liu, J. Li and M. Guizani, "PKC Based Broadcast Authentication using Signature Amortization for WSNs", IEEE Transactions on Wireless Communication, vol. 11, no. 6, June 2012.

[21]. Fucai Y., E. Lee, S. Park, J. Lee, M. Jin, and S. Kim, "Location Update Scheme for Mobile Sink in Wireless Sensor Networks," IEEE Communications Society , ICC 2010.

[22]. N. Koblitz, and A. J. Menezes, "A Survey of Public-Key Cryptosystems", http://www.uncc.edu, Aug. 7, 2004.

[23]. N. Kangude, P. Wani, S. Raut, "Advanced Encryption Standard (AES)," ijcse.net, Computer Security Standard, Cryptography (IJCSE), vol. 1, no. 3, pp. 118-126, April 2011.

[24]. William Stallings, Cryptography and Network Security, Fourth Edition, Prentice Hal, 2011.