International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-8 Issue-10, August 2019

A New Hybrid Cryptography Technique in Wireless Sensor Network

Alakananda Tripathy, Sateesh Kumar Pradhan, Alok Ranjan Tripathy, Ajit Kumar Nayak

Abstract: ***The wireless sensor network is a large number of tiny nodes installed in insecure environment for monitoring, gathering and transferring data and are prone to security threats for its limited resources. In order to transmit the data and to protect from different attacks in the network, security is maintained. To achieve confidentiality, authenticity and authorization of data which secure the data from different attacks cryptographic algorithm were used. The number of keys used in the cryptographic algorithm determines the security of the data. Cryptographic algorithms are broadly classified into two types symmetric cryptography and asymmetric cryptography. In the symmetric key cryptographic algorithm, a secret key is shared in the network and in asymmetric key cryptographic algorithm two keys are used for data security. In wireless sensor network, symmetric key cryptography required more storage to store the key among all the nodes of the network and in asymmetric key cryptography more computation time is require for the data encryption and decryption. To avoid memory and computation overhead we proposed a hybrid cryptosystem to handle the security in the wireless sensor network. Initially shared key is exchanged among nodes using ECC which is a public key algorithm. Data is encrypted and decrypted using RC4 symmetric key algorithm. Various performance measures such as time taken for encryption and decryption process and memory needed for storing cipher text data. The proposed model shows faster encryption of data and takes less memory for key storage as compared to the traditional approach.***

Index Terms**: Security, Cryptography, Symmetric key, Cipher text, Encryption, Decryption.**

I. INTRODUCTION

Wireless Sensor Networks consists of a number of small devices, called sensors node. The basic applications of wireless sensor networks are earthquake monitoring, ocean monitoring, health care monitoring and many military applications [17]. Each node of the sensor network is able to process, communicate sense and monitor the physical environment like temperature, pressure, humidity, noise level, etc. [7]. The basic advantage of these networks is that it performs processing of large streams of data into useful information in the network. So the protection of these data is a major issue. The sensor is a tiny device with limited in their energy, memory and storage space for code so the security becomes a challenging issue. Unlike traditional network sensor nodes are deployed in battlefield, in building monitoring, burglar alarms and in a critical system like hospitals and airports [21]. So the confidentiality of data is to be maintained while communicating in the sensor network otherwise it may result in a physical attack. The sensor network also interacts with the physical environments which lead to the security overhead [9]. The sensor node and the base station need to verify the authenticity of the data they received from a trusted sender or not otherwise an adversary can trick the data. It is also needed to maintain the integrity of the data.

This paper aims to provide an efficient algorithm for data communication in the sensor network. A Cryptographic algorithm is used to achieve the different security requirements as follows:

1. Data availability means the data must be available when needed. It also specifies the network is also available when required.
2. Data confidentiality is the security of the sensitive data from unauthorized access which can be achieved using encryption algorithms.
3. Data authenticity confirms the identity of data that is sent from valid users.
4. Data integrity confirms the data not be changed by an unauthorized user. Digital signature is used to provide data integrity.

By maintaining the security requirements in the sensor network data can be protected from different types of attacks like active attack and passive attack.

The passive attack is to monitor and listen to the communication link by unauthorized nodes [5]. In an active attack the data stream in the communication link can be modify by an adversary. The most common types of active attacks are:

1) Denial of Services (DoS) Attack

It is an attempt to make sender or receiver or the resources unavailable to a legitimate user. It basically infuses the target by making excessive communication request. The attacker can degrade the network performance by providing extra data packets in the network. This may result in loss of the packet, the performance of the network slows down by reducing the quality of service of a network.

Manuscript published on 30 August 2019.
*Correspondence Author(s)
Alakananda Tripathy, Department of Computer Science and Engineering, S'O'A Deemed to be University, Bhubaneswar, India.
Sateesh Kumar Pradhan, Department of Computer Science and Applications, Utkal University, Bhubaneswar, India.
Alok Ranjan Tripathy, Department of Computer Science, Ravenshaw University, Cuttack, India. E-Mail: tripathyalok@gmail.com
Ajit Kumar Nayak, Department of CSIT, S'O'A Deemed to be University, Bhubaneswar, India

2) Sink Hole Attack

A node attracts all the traffic in the sensor. The malicious node is placed in such a way that it rejects the entire packet that is routed through it [16]. If the malicious node stays in the network for a longer period of time then the number of node increases to transmit data. The malicious node is placed near the base station to create a sinkhole.
The intruder has more power to do computation and communication than other nodes in the network and establish a connection with the base station.

3) Sybil Attack

A single node makes duplicate copies of itself and present in a different locations. This attack targets storage, multipath routing, resource allocation [8] and the maintenance of topology. To prevent sybil attack encryption techniques are used.

4) Wormhole Attack

Two nodes make a tunnel. One end of the tunnel is near to the base station while the other end is placed far from the base station. The tunnel provides high quality route to the other nodes to transmit the packet by misleading them and also they encapsulate the packet which is transmitting in out of band channel**.**

5) HELLO Flood Attack

In HELLO Flood attack adversary advertise HELLO packets to the nodes present in isolated places in a sensor network. The other nodes in the network try to send messages to the base stations [24] using this route. The attacker send HELLO packet with more energy to one node to another and the node which respond to the HELLO flood message waste their energy.

6) Node Replication Attack/Cloning Attack

A node is added by an adversary by creating a copy of a node present in the sensor network. This may disrupt the network's performance and also corrupt the packet which may lead to a disconnected network. By gaining physical access to the network the attacker can get a copy of the cryptographic keys for the duplicate node and can easily manipulate the segment of the network [8]. In order to protect the network from different types of attack discussed in this section and to maintain the different security requirement it is essential to have some security mechanism. In our next section 2, some of the security mechanism is being discussed like different key distribution techniques and cryptographic algorithms. In Section 3 and 4 we explain the Elliptic curve cryptography and RC4 cryptography which is used in our proposed hybrid model ECC-RC4 discussed in section 5 which is proposed by taking the advantages of both the cryptographic algorithm. Section 6 consists of results and discussion depending on different parameters like encryption and decryption time, storage space, keys needed in the network. Section 7 concludes the paper.

II. SECURITY MECHANISM

A. Key Pre Distribution Scheme

The key plays an important role in the cryptographic algorithm to perform encryption and decryption in order to maintain data confidentiality and authenticity. In key pre distribution scheme sensor node first store some keys before deployment. In order to communicate message after deployment, the sensor node uses the keys. The nodes involved in communication can use the secret key to encrypt the data for communication. Only the secret is stored in the node is the advantage of this scheme. But if once the node is compromised then the security of the entire network is in risk. To overcome this limitation each sensor node stores N-1 number of pair wise secret keys where N is the total number of nodes. If the number of nodes in the network is N then a number of keys to be stored in the network is N*(N-1)/2. Thus compromising one node cannot affect the entire network. But it is impractical to use because if the number of node increases in the network then a large number of keys are to be used as there is limited storage space. In this scheme it is also difficult to add new node to the existing network. So in some key predistribution scheme only probability of existence is there for the shared key between the nodes and in some schemes the key exists between the nodes. The different key distribution techniques are explained below:

1) Random Key Predistribution Technique

Eschenauer and Gligor [12] proposed the random key predistribution scheme. In this method, key are distributed before the nodes are deployed this is known as initialization phase. Random pools of keys are being picked from the total possible key space. From the key pool S each node selects the key randomly and stores it in its memory. This random set of keys is known as the node's key ring. The keys are selected in such a way that two random subsets of keys will be in the key pool *S* and one key will be shared with some probability *p*.

2) Q-Composite Key Predistribution Scheme

Perrig, Chan and Song [6] discussed a q-composite key predistribution scheme based on a random key distribution scheme. This scheme improves the random predistribution scheme by increasing the number of keys. Here q common keys are required instead of one for key setup by increasing the resilience of the network against node capture. In this scheme, two nodes find q keys before generating the shared key and establishing secure communication.

3) Blom's Key Predistribution Scheme

This scheme was proposed by Blom [3] where any pair of nodes find a secret pair wise keys between them. This scheme uses $\lambda+1$ memory space with λ smaller than N. It is not resilient against node capture.

4) Pair wise Key Predistribution Scheme

Based on Blom's scheme Du, Deng, Han, Varshney, Katz and Khalili [11] propose a new key predistribution technique. This scheme improves the resilience of the network. In this scheme the network is imagined as a graph and in which each node is a vertex of the graph and if there is an edge between nodes then they can establish a key. The graph in this technique is a connected graph rather than the complete graph as described in Blom's scheme.

Retrieval Number: I8736078919/19©BEIESP
DOI: 10.35940/ijitee.I8736.0881019
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

5) Polynomial Pool Based Key Predistribution Scheme

Blundo, Santis, Herzberg, Kutten, Vaccaro, & Yung [4] discussed polynomial pool based key distribution technique. This scheme was based on the pair wise key predistribution scheme. In this scheme, the key setup server was maintained offline which provides a unique id to each sensor node to deploy in the target field for key predistribution.

The server then generates a symmetric bivariate t-degree polynomial $f(x,y)=\sum_{i,j=0}^{t} a_{ij}x^i y^j$ over a finite field F_q, where the coefficients $a_{ij}\,(0\le i,j\le t)$ are randomly chosen and q is a prime number that is large enough for a cryptographic key with the property $f(x,y)=f(y,x)$.

6) Deterministic Key Distribution

Lee and Stinson [15] proposed a deterministic key predistribution technique. Here, the sensor network is modeled as a complete bipartite graph. Each sensor node has a unique id and the edges are decomposed into a star like sub graph where each vertex is a center of one star and r/2 numbers of distinct stars are leaf. A sensor node will receive two keys one will obtain from the key pool and the other is the hash key. It can be used in a large sensor network and improves the resilience against node capture.

B. Cryptography

Cryptography is the art of hiding text. Wireless sensor network involves in monitoring earthquake, wildlife and, numerous military application. So in order to maintain the confidentiality of data cryptographic algorithm is used. In wireless sensor network applications are needed to be protected from eavesdropping, alteration. Cryptography is basically categorized into two types public key cryptography and private key cryptography based on the number of keys needed for data encryption. In symmetric key cryptography, keys are to be distributed before deployment which needs more memory to store the secret key. In asymmetric key cryptography, more resource is required for computation. To overcome the limitation of both the technique we proposed ECC-RC4 hybrid model which have the features of symmetric and asymmetric key cryptography by making it efficient for sensor network. This section describes some of the symmetric and asymmetric key cryptographic algorithms**.**

1) Symmetric Key Cryptography

In symmetric key cryptography, a shared key is used for data encryption and decryption. To perform symmetric encryption keys are to be distribute initially before the deployment. The different key predistribution scheme is discussed above. Popular symmetric key algorithm is AES, RC4, RC5, and Blowfish.

AES

AES is known as Advance encryption standard [18] it became effective in 2006 and was also known as Rijndael and was designed by Vincent Rijmen, Joan Daemen. AES is a block cipher technique which encrypts the data using keys of length 128, 192 and 256 bits. It is based on substitution and permutation network and is efficient in hardware and software implementation. It has a fixed block size of 128 bits. The keys are used to generate cipher text using different rounds like 128 bits key needed 10 rounds, 192 bits keys needed 12 round and 256 bits keys needed 14 rounds. Each round of the algorithm consists of several processing steps.

RC4

It was designed by Ron Rivest. It was a stream cipher technique to generate the cipher text use keys from 40 to 2048 bits. This technique is faster to generate the cipher text.

RC5

RC5 is a symmetric block cipher [19] designed by Ronald Rivest in 1994. RC5 is a block cipher algorithm with variable size of block from 32 bits, 64 bits, and 128 bits and the key size are from 0 to 2040 bits and the number of rounds in this algorithm is from 0 to 255. For a block size of 64 bits, 12 rounds and 128 bits key are used. RC5 is feistel structure network consisting of a number of modular addition and exclusive OR operations.

Blowfish

It is a symmetric cipher designed by Bruce Schneier [20] in 1993 and is the successor of Twofish algorithm. It has 64 bit block size and variable length key from 32 bits to 448 bits. It is a feistel cipher and consists of 16 rounds of operations.

In the given table 1 we discuss the different symmetric key algorithm, number of keys used and the number of round function.

The different symmetric key algorithm and its different parameters like type whether it is a block cipher or stream cipher, size of the plaintext, length of the key and number of rounds are being summarized in table 1. RC4 is a stream cipher algorithm while AES, RC5, DES, and Blowfish are the block cipher algorithm. DES, AES, RC5, and Blowfish

Table 1: Different Symmetric Key Algorithm

Symmetric Algorithm	Block/Stream	PlainText Size	Keys Length in bits	Number of Round
DES	Block cipher	64 bits	56 (8 parity bits)	16
AES	Block cipher	128 bits	128,192,256	12,14,16
RC5	Block cipher	32 bits,64 bits,128 bits	0-2040	0 to 255
Blowfish	Block cipher	64 bits	32 to 448	16
RC4	Stream cipher		40 to 2048	1

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

perform 16, 12, 20, and 16 rounds of operation whereas RC4 performs only 1 round of operation and the size of the input for different algorithm is basically 64 bits and 128 bits, RC5 takes input of 32 bits of input which is suitable for sensor network but it takes more time for encryption as it need 20 rounds of operation.
In the case of RC4, it takes comparatively less time as the number of rounds is fewer.

2) Asymmetric Key Cryptography

Asymmetric key cryptography basically uses two keys one for encryption and the other key for decryption. In wireless sensor network two commonly used public key cryptographic algorithms are RSA and ECC. This section describes both the technique and advantages and disadvantages.

RSA

Ron Rivest, Adi Shamir, and Leonard Adleman designed the RSA algorithm [27] in 1997. It is the asymmetric key algorithm and widely used for secure data communication. In this cryptosystem the data is encrypted using the public key and decrypted using the private key. The public key is based on two large prime numbers where the prime number is kept secret. This algorithm consists of four steps key generation, key distribution, encryption, and decryption. RSA algorithm takes more time to encrypt and decrypt the data as the key size is large as the size ranges from 1024 to 4096 bits.

Elliptic Curve Cryptography (ECC)

The asymmetric key cryptography based on elliptic curve was suggested by Neal Koblitz and Victor S. Miller in 1990 is known as Elliptic curve cryptography(ECC) [14]. Elliptic curve can be used for key generation and encryption of data. The key length in ECC is 160, 224, 256, and 512 bits. The Elliptic curve cryptography is faster, smaller, and more efficient than RSA cryptosystem [1]. ECC also has good battery backup [2]. Wander, Gura, Eberle, Gupta, & Shantz [23] compared energy needed for the key generation of RSA and ECC which shows that ECC is more efficient than RSA and takes less time for the computation than RSA. The key size is ECC is smaller as compared to RSA.
The given table 2 discusses ECC and RSA algorithm based on different parameters like memory needed to store the key, time complexity, generation of key and the length of the key. The memory required for ECC is comparatively less than RSA, the time needed to compute ECC algorithm is O (n^2) whereas RSA needed O(n^3) and generating the keys for ECC is faster than RSA. On the basis of comparison of different algorithm based on the different parameter we have taken ECC for generating and distributing the key and RC4 to provide security to the message to be communicated in the insecure channel. The traditional Elliptic curve cryptosystem and RC4 algorithm were discussed in the next section.

Table 2: Comparison Between RSA and ECC

Method	RSA	ECC
Memory	Large	Comparatively less
Time	O(n^3)	O(n^2)
Complexity		
Key Generation	Slow	Fast
Keys(in bits)	160-512	1024-4096

III. ELLIPTIC CURVE CRYPTOSYSTEM (ECC)

Elliptic curve cryptography otherwise known as ECC [26] [14] is based on the technique of algebraic structure of elliptic curve over a finite field. Smaller keys are generated using ECC than the RSA key generation algorithm. Elliptic curves are not ellipse but the equation is generated by calculating the circumference of the ellipse.

Definition: An elliptic curve over a prime field $E_p(a,b)$ is defined as:

$$y^2 \equiv x^3 + ax + b \pmod p, \tag{1}$$

where p is the prime number and a and b belongs to the field of integer and must satisfy the condition

$$4a^3 + 27b^2 \neq 0 \pmod p \tag{2}$$

The set of all points (x, y) that satisfy the equation (1) besides point 0 where $p+(-p)$ becomes infinity [13] is the elliptic curve group over the finite field p.

A. Elliptic Curve Operations

To perform operations on an elliptic curve, points on the curve are needed to be calculated. The basic operations that perform on points are addition of point, point doubling, scalar multiplication and inverse operation.

Point Addition

Let there are two points on the curve $P(X_p, Y_p)$ and $Q(X_p, Y_p)$ and adding these two points result in $R(X_r, Y_r)$ which also lies on the curve. The result of an addition is generated based on some condition and it is as follow:

Case 1: If $P \neq Q$, then $R(X_r, Y_r) = P + Q$ (3)

$$X_r = \left(\lambda^2 - X_p - X_q\right) \bmod p \tag{4}$$

$$Y_r = \left(\lambda\left(X_p - X_r\right) - Y_p\right) \bmod p \tag{5}$$

Where λ is slope and $\lambda = \dfrac{Y_q - Y_p}{X_q - X_p}$

Case 2: If $X_p = X_q$ but $Y_p \neq Y_q$ then $P + Q = 0$

Point Doubling

Two points on the curve $P(X_p, Y_p)$ and $Q(X_q, Y_q)$ are the same then the resultant $R(X_r, Y_r)$ is

If $P = Q$, then $R(X_r, Y_r) = P + P$ (7)

(8)

Retrieval Number: I8736078919/19©BEIESP
DOI: 10.35940/ijitee.I8736.0881019
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

$$Y_r = \left(\lambda\left(X_p - X_r\right) - Y_p\right) \bmod p \quad (9)$$

Where λ is slope and $\lambda = \dfrac{3X_p^2 + a}{2Y_p}$

Scalar Point Multiplication

The multiplication between two points is not possible so it was done by repeated addition. When a point multiplication and point doubling are done then an integer value is multiplied with point where k is the scalar value and P is the Point. So the scalar multiplication is defined as

$R = P + P + ... + P$ for k times (10)

Let 2P=P+P.

Inverse Operation

Let $P = (X_p, Y_p)$, then the negative of the point P is defined as

$Q = -P = \left(X_q - Y_q\right)$, where $P + Q = 0$ [10].

B. Key Generation using ECC

To generate key elliptic curve are used over a finite field F_p [13]. Let P be a point in the curve and P has prime number order up to n. The elliptic curve, point P and the prime number are the parameters. The private key is an integer d randomly selected from the interval $[1, n-1]$ and the public key is $Q = dP$. If the parameters are known then also it is difficult to determine the value of d (i.e private key) which is known as elliptic curve discrete logarithm problem (ECDLP).

Algorithm for Key Generation Using ECC [13]

Algorithm1: Elliptic Curve Key Generation

INPUT: Elliptic Curve Domain Parameter
OUTPUT: Public key Q and Private key d

1. Let $d \in [1, n-1]$
2. Find $Q = dP$
3. Return Q and d

IV. STREAM CIPHER GENERATION USING RC4

RC4 [25] is a symmetric key algorithm used for data encryption and decryption. RC4 is a stream cipher algorithm. In this algorithm, message is encrypted by performing XOR operation with the key. The key used in RC4 is 128 bits or 256 bits. The key stream is independent of the plaintext. Figure 1 explains the block diagram of the RC4 algorithm. The steps involved in the encryption are explained below:

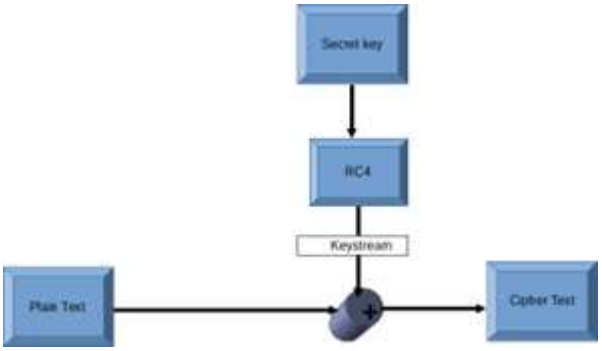

Figure 1: RC4 Block Diagram

Pseudo random number generation algorithm (PRGA) is used to generate random number to encipher the message and key scheduling algorithm is used to initialize the key.

A. Key Scheduling Algorithm (KSA)

It generates the state array.

Algorithm 2: Key Scheduling Algorithm

1. Set the state vector S.
2. Repeat steps 3 and 4 for $i = 0$ to 255
3. update the state vector as $S[i] = i$
4. update the temporary vector as $T[i] = k[i \bmod (k)]$
5. initialize $j = 0$
6. Repeat steps 7 and 8 for $i = 0$ to 255
7. update $j = \left(j + S[i] + T[i]\right) \bmod 256$
9. swap the values of state vector using $\left(S[i], S[j]\right)$
10. stop

where S is the state array, T is the temporary vector and K is the key array and k is the key length.

B. Pseudo Random Number Generator

It generates the key stream one by one and XOR [k] with the next byte of the message to encrypt or decrypt.

Algorithm 3: Pseudo Random Number Generator

1. initialize $i = 0$ and $j = 0$
2. Repeat steps 3 to 7 for message encryption
3. update $i = \left(i + 1\right)\left(\bmod 256\right)$
4. update $j = \left(j + S[i]\right)\left(\bmod 256\right)$
5. swap the values in the state vector using $\left(S[i], S[j]\right)$
6. update the key using $k = \left(S[i] + S[j]\right)\left(\bmod 256\right)$
7. compute the ciphertext using $C_i = M_i$ XOR $S[k]$
8. stop

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

(11)

V. PROPOSED ECC-RC4 (PER) ALGORITHM

A hybrid cryptography model ECC-RC4 (PER) is proposed for key generation and distribution using the elliptic curve cryptosystem and RC4 is used to achieve data confidentiality. It can overcome the limitation of the key distribution as required in private key cryptography and also provide confidentiality of data in less time than asymmetric key cryptography which is been discussed in the previous section. The different parameters based on which the system was proposed are encryption and decryption time, number of keys needed, storage space required to store the cipher text and also the key size. In this scheme key generation and distribution are done using ECC and the encryption and decryption are done using the RC4 algorithm which is a stream cipher. This algorithm avoids the limitations of sharing more number of the secret key as needed in symmetric key cryptography in wireless sensor network which is more time consuming. It also increases the security of the secret key. The PER algorithm can make the maximum use of limited resources along with providing the security of data. RC4 is a stream cipher [22] technique that requires a single key for encryption and decryption used to make the message confidential while communicating in the insecure channel. The key used for encryption is generated using ECC which is public key cryptography algorithm. In public key cryptography, the complexity of the algorithm is directly proportional to the size of the data. Public key cryptography is more secure but takes more time for encryption and decryption. As the complexity is proportional to the length of the message. To avoid more computation time for data encryption and decryption RC4 algorithm is used where the key ranges from 40 to 2048 bits. The proposed hybrid algorithm consists of three steps initialization, key generation, encryption and decryption.
The system architecture of the hybrid model is shown in figure 2:

Figure 2: System Workflow

The steps of the Proposed ECC-RC4 (PER) algorithm are broadly categorized into three steps as discussed below:

A. Initialization

The generator point G, the curve parameters a and b along with few more constants of ECC will be sent to all the nodes in the network for data communication.

B. Key Generation

In the wireless sensor network, a node has to share the secret key in the network before communication. The elliptic curve (ECC) is used for generating and sharing the key. ECC increases the security of the key by making difficult to identify the private key. The following steps discuss the ECC key generation algorithm.

- Let both the node involves in communication choose a random number n_x from interval $[1, n-1]$, as the private key. Let the private key is d for node A and e for node B.
- Now both the node will compute the public key by multiplying the private key with the base point G. i.e. $Pub_a = d.G$ and $Pub_b = e.G$.
- The public key will be broadcasted to another node.
- Node A computes the secret key K by multiplying his private key d with node B public key (Pub_b).

$$K = e.Pub_a = d.Pub_b$$

Algorithm 4 describes the key generation procedure for Proposed ECC-RC4 (PER)

Algorithm 4: PER Key Generation

INPUT: Curve Parameter
OUTPUT: 16 Bytes or 32 Bytes Key
1. let all nodes $N_i \in N$
2. Node choose a random number n_x as private key from the interval $[1, n-1]$.
3. compute public key for a node A is $Pub_a = n_x.G$ //point multiplication
4. broadcast the public key Pub_A to other nodes $N_j \in N$

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

5. if node A sends data to node B then the secret key is computed using $k = n_A \times Pub_A = n_B \times Pub_B = n_A \times n_B \times G$
//k is the secret key
6. stop

Once the key is generated data encryption is performed using PER hybrid encryption algorithm. For data encryption key can be either 16 or 32 bytes long.

C. Data Encryption and Decryption using RC4

For data encryption and decryption RC4 stream cipher is used. First a state vector of length 255 is initialized then a temporary vector is initialized by using the equation $TV[i] = k[i \bmod (k)]$ where k is the secret key.

Before the encryption of data values, the state vector is updated using the equation $j = \left(j + V[i] + TV[i]\right) \bmod 256$ then Swap $(V[i], V[j])$ where $V[i]$ the state vector is and $TV[i]$ is the temporary vector, i and j are the index number of vector. Each byte of the message is encrypted using $C_i = M_i$ XOR $V[k]$ where C_i is the cipher text M_i is the plaintext and $V[k]$ is the key obtained using $k = \left(V[i] + V[j]\right) \bmod 256$.

The Algorithm 5 is the Proposed ECC-RC4 (PER) Hybrid Algorithm to encrypt the data.

Data Decryption

The decryption process is same as the encryption. The Algorithm 6 discusses the decryption process.

Algorithm 5: PER Hybrid Encryption Algorithm

INPUT: Message as Plaintext
OUTPUT: Ciphertext
1.The plaintext should be string of 16 bytes long
2. initialize the state vector V
3.repeat steps 3 and 4 for $i = 0$ to 255
4. set the state vector $V[i] = i$
5. set the temporary vector using $TV[i] = k[i \bmod (k)]$
6. set $i = 0$, $j = 0$
7. repeat steps 8 and 9 for $i = 0$ to 255
8.update $j = \left(j + V[i] + TV[i]\right) \bmod 256$
9. swap the values in the state vector using $\left(V[i], V[j]\right)$
10.encrypt the message using steps 11 to 15
11.update $i = \left(i + 1\right) \bmod 256$
12.update $j = \left(j + V[i]\right) \bmod 256$
13. swap the values in the state vector using $\left(V[i], V[j]\right)$
14.compute the key $k = \left(V[i] + V[j]\right) \bmod 256$
15. compute the ciphertext $C_i = M_i$ XOR $V[k]$
16. stop

Algorithm 6: PER Hybrid Decryption Algorithm

INPUT: Ciphertext
OUTPUT: Plaintext
1. The ciphertext should be string of 16 bytes long
2. set $i = 0$, $j = 0$
3. repeat steps 4 and 5 for $i = 0$ to 255
4. update $j = \left(j + V[i] + TV[i]\right) \bmod 256$
5. swap the state vector using $\left(V[i], V[j]\right)$
6. decrypt the ciphertext using step 7 to 11
7.update $i = \left(i + 1\right) \bmod 256$
8.update $j = \left(j + V[i]\right) \bmod 256$
9. swap the values in the state vector $\left(V[i], V[j]\right)$
10.compute the key $k = \left(V[i] + V[j]\right) \bmod 256$
11. plaintext $P_i = C_i$ XOR $V[k]$
12. stop

VI. RESULTS AND DISCUSSION

The result is being discussed based on different parameters like key size, the number of keys needed for the key predistribution in wireless sensor network, encryption time of the algorithm, decryption time of the algorithm, memory or the storage space needed to store the ciphertext and throughput. Table 3 shows the keys needed for pre distribution in symmetric key, asymmetric key and our proposed hybrid scheme. In order to perform encryption and decryption keys are needed to be distribute in the network initially.

In symmetric key

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

cryptography, each node must contain the key before encryption and decryption as one shared secret key is used for encryption and decryption. So it is very difficult to predistribute large number of keys in the network as the sensor node have limited whereas in case of asymmetric key cryptography number of keys used are comparatively fewer. The proposed model also uses less number of keys than symmetric key.

Table 3: Keys for Predistribution

Algorithm	Number of Keys for Predistribution
Symmetric	$n(n-1)/2$
Asymmetric	$2n$
Hybrid	$2n$

Table 4 shows the number of keys required by the symmetric key, asymmetric key and hybrid cryptosystem for different number of nodes. The symmetric key cryptography required more number of keys than the asymmetric key and hybrid cryptosystem. The nodes store almost two times more number of key in predistribution phase of symmetric key than in asymmetric key cryptography.

Table 4: Number of Keys Needed for Different Number of Node

Number of Nodes	Keys stored in Symmetric	Keys stored in Asymmetric	Keys stored in Hybrid
10	45	20	20
20	190	40	40
50	1225	100	100
100	4950	200	200
500	124750	1000	1000

To generate the key for our proposed algorithm elliptic curve cryptography is used as the key size is comparatively less than RSA as the sensor node has less space for storage and RSA key size is larger than ECC. Table 5 shows the number of keys needed for ECC and RSA.

Table 5: Keys Used in ECC and RSA

RSA(in bits)	ECC (in bits)
1024	160
2048	224
3072	256
7680	384
15360	512

A. ***Encryption Time***

It is the time required for an algorithm to encrypt the plaintext to ciphertext before sending it to the receiver. The encryption time of AES, RC4 and proposed PER algorithm is generated for different input sizes with key 128 bits to generate ciphertext is shown in table 6. RC4 algorithm takes comparatively less time than AES to encrypt a message of the same length. As RC4 takes less time in our proposed algorithm we RC4 is used to provide encryption of data.

Table 6: Encryption Time of AES, RC4 and ECC-RC4 for Different Input Size and Key Size

Data Size (in bytes)	AES	RC4(128)	PER(128)	PER(256)
82	0.009585142135562	0.0002281665802	0.00922226905	0.0091340542
183	0.0202071666718	0.000338792800903	0.00933289527	0.0093262196
245	0.0270209312439	0.000808000564575	0.00980210304	0.0094060898
576	0.0296030044556	0.000783205032349	0.00977730751	0.0098781586
970	0.100502967834	0.00151395797729	0.01050806045	0.0104911327
1500	0.154325962067	0.00255107879639	0.01154518126	0.0113370419
2688	0.266507863998	0.00505495071411	0.01404905319	0.0131771564
5371	0.531640052795	0.0118668079376	0.01018078327	0.0176382065
8059	0.798007965088	0.0155639648438	0.02455806732	0.0227372646
10745	1.06198501587	0.0169258117676	0.0291991424	0.0260779858

The encryption time is also generated in our proposed system which takes less time to execute and generate the ciphertext than the traditional AES algorithm. The comparison between different symmetric key algorithms like AES, RC4, and our proposed hybrid algorithm is shown in the figure. 3 with key size 128 and 256 bits and with different size of data. The proposed PER algorithm takes less time to encrypt the message with key 256 bits as compared to AES, and PER with key 128 bits. PER hybrid algorithm with key size 256 bits needed less time to encrypt the message than proposed algorithm with 128 bits key. From table 5 we can say that the proposed algorithm with key 256 bits required less time for encryption of data than PER and AES if the length of the message is larger.

Retrieval Number: I8736078919/19©BEIESP
DOI: 10.35940/ijitee.I8736.0881019
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

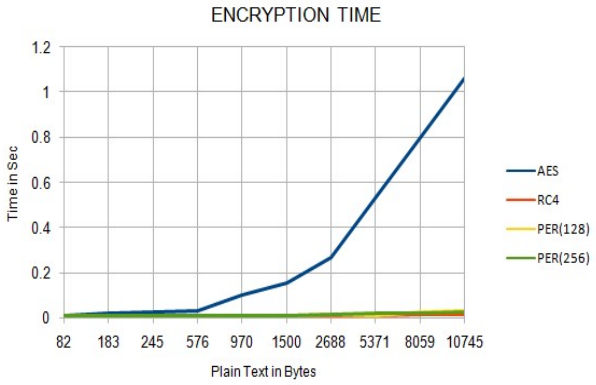

Figure 3: Encryption Time for AES, RC4, and PER

Decryption time is the time required to convert ciphertext to plaintext. The decryption time for AES, RC4 and our proposed PER for different input size is discussed in table 7. The RC4 algorithm takes comparatively less time to decrypt the ciphertext than the AES algorithm. As it is a stream cipher it is faster than the AES algorithm. We also discussed the decryption time for the proposed PER algorithm for different input sizes. Figure 4 shows the decryption time of AES, RC4 and PER.

Table 7: Decryption Time of AES, RC4 and PER for Different Input Size and Key Size

Data Size (in bytes)	AES	RC4	PER(128)	PER(256)
82	0.0108301639557	0.0002392676906	0.00923337016	0.0093440652
183	0.0208840370178	0.000349883901	0.009349883901	0.0094572297
245	0.0265829563141	0.0091007765675	0.00990418013	0.0095070999
576	0.0318059921265	0.0011043060443 5	0.01009840852	0.0108893788
970	0.1037199497 22	0.001741681883	0.01073578436	0.0116722456
1500	0.161454916	0.0036620988741	0.01265620135	0.0124480321
2688	0.2733709812 16	0.0060516082422	0.01504571072	0.0142882664
5371	0.5453920364 38	0.0121779180487	0.02117202052	0.0188493075
8059	0.8182189464 57	0.017774159549	0.02676826202	0.0238383757
10745	1.09042191505	0.0271369228787	0.03613102535	0.0280889978

The proposed PER algorithm takes less time decrypt message with key 256 bits as compared to AES, and proposed algorithm with key 128 bits for larger message.

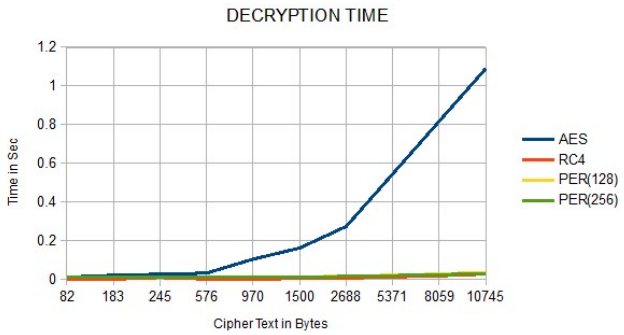

Figure 4: Decryption Time for AES, RC4, and PER

B. Memory Requirement

Memory is required to store the ciphertext after the message is being encrypted. The encrypted message is then sent to the receiver. In table 8 we discuss the memory required to store the ciphertext of different cryptographic algorithms for the key size 128 bits. AES needed 27 KB of space to store ciphertext of 10 KB of data whereas RC4 and PER needed 21 KB of space to store ciphertext for 10 KB of plaintext using key 128 bits and 256 bits. RSA algorithm required more space to store ciphertext for 10 KB of the message using key 1024 bits. Symmetric key cryptography needed less space for storing the ciphertext than asymmetric key cryptography. AES and RSA algorithm required 2 times to 3 times more space for storing the ciphertext of the same length as compared to the PER hybrid algorithm with key 128 bits and 256 bits. In figure 5 shows the memory space desired for the different encryption algorithms for different size of the plaintext.

Table 8: Storage Needed for Ciphertext for AES, RC4, RSA and PER with 128 and 256 bits Key

Data Size (in bytes)	AES	RC4	RSA	PER(128)	PER(256)
82	289	161	244	162	165
183	555	493	551	495	370
245	712	368	735	369	488
576	778	579	594	579	1152
970	3300	2088	3094	2080	1940
1500	4000	3052	4373	3048	3004
2688	6088	5410	6789	5406	5347
5371	13943	10745	14807	10742	10743
8059	20031	16155	21596	16150	16118
10745	27886	21490	29614	21484	21490

In the proposed hybrid PER algorithm encryption of the data is achieved using the key of different length 128 bits and 256 bits. The algorithm can be used with different keys based on the availability of the memory to store the ciphertext. The proposed algorithm required more memory space for encrypting data of larger length with key 128 bits as compared with an algorithm with key 256 bits.

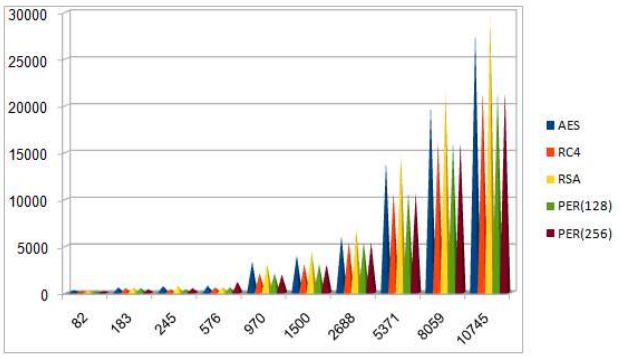

Figure 5: Memory Requirement for Storing Ciphertext for AES, RC4, RSA and PER

C. Throughput

Throughput of an algorithm is the speed of encryption or decryption time. If the throughput of an encryption algorithm increases then the algorithm will consume less power. In figure 6 and table 9 the throughput of AES, RC4 is compare with PER of key length 128 and 256 keys. Throughput of the proposed algorithm PER is better than AES. The hybrid PER algorithm consumes less power than AES. RC4 has better throughput than other algorithm and it is faster and consumes less power but it is not more secure as it is stream cipher so in order to make the keys of RC4 algorithm more complex elliptic curve cryptography is used to generate the keys.

Table 9: Throughput of Different Algorithm

AES	RC4	PER(128)	PER(256)
0.08	4.52	1.76	1.75

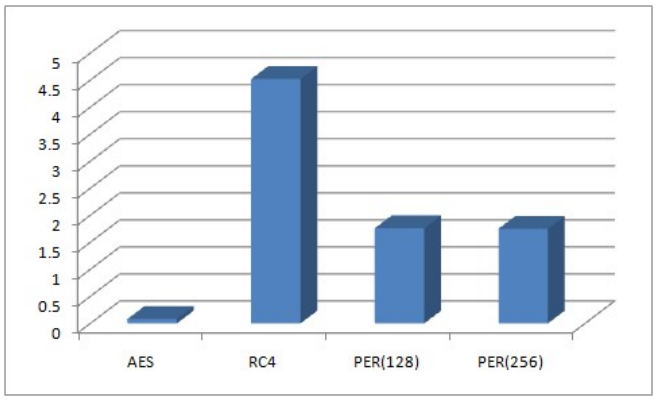

Figure 6: Throughput of AES, RC4 and Proposed PER Algorithm.

VII. CONCLUSION

A hybrid security algorithm is proposed for wireless sensor network. It is basically designed to solve problem like memory constraints, computation time and strengthen the key used for encryption and decryption. In the proposed model there is a difficulty in finding the key even if the curve parameters are known. It provides better security by enhancing the security of the key. It is robust against attacks in different layer. The proposed ECC-RC4 (PER) algorithm use elliptic curve cryptography and RC4. It uses the advantages of both symmetric and asymmetric key cryptography. Elliptic curve cryptography is used for key distribution as in asymmetric key cryptography less storage is needed for key predistribution. RC4 is symmetric stream cipher used for message encryption and decryption as it needs less space to store the ciphertext as compared to symmetric encryption algorithm. RC4 algorithm is also faster among the symmetric and asymmetric algorithm. The hybrid model offers better security as compared to other as it use elliptic curve to generate shared key which is difficult to crack by attacker. The model provides faster encryption and decryption time and need less space to store the encrypted message as compared to other algorithm like AES and RSA. It provides better encryption and decryption time for shorter message when the key length is 128 bits and for longer message when the key length is 256 bits as compared to AES with same length of message. The throughput of the proposed hybrid model is better than the AES algorithm. It also consumes less power.

REFERENCES

1. Amin, F., Jahangir, A. H., & Rasifard, H. (2008). Analysis of public-key cryptography for wireless sensor networks security. world academy of science, engineering and technology, 41, 529-534.
2. Bhanot, R., & Hans, R. (2015). A review and comparative analysis of various encryption algorithms. International Journal of Security and Its Applications, 9(4), 289-306.
3. Blom, R. (1984, April). An optimal class of symmetric key generation systems. In Workshop on the Theory and Application of of Cryptographic Techniques (pp. 335-338). Springer, Berlin, Heidelberg.
4. Blundo, C., De Santis, A., Herzberg, A., Kutten, S., Vaccaro, U., & Yung, M. (1992, August). Perfectly-secure key distribution for dynamic conferences. In Annual international cryptology conference (pp. 471-486). Springer, Berlin, Heidelberg.
5. Buch, D., & Jinwala, D. C. (2010, December). Denial of Service Attacks in Wireless Sensor Networks. In International conference on current trends in technology, Nuicone.
6. Chan, H., Perrig, A., & Song, D. (2003, May). Random key predistribution schemes for sensor networks. In IEEE symposium on security and privacy (Vol. 197).
7. C K Marigowda and ManjunathShingadi, *"Security Vulnerability Issues In WirelessSensor Networks: A Short Survey"*International Journal of Advanced Research in Computer and Communication Engineering Vol. 2, Issue 7, July 2013
8. C. KARLOF AND D. WAGNER, *"SECURE ROUTING IN SENSOR NETWORKS: ATTACKS AND COUNTERMEASURES"*, ELSEVIER'S ADHOC NETWORKS JOURNAL, SPECIAL ISSUE ON SENSOR NETWORK (SNPA), 2003,SEPTEMBER, PP. 293-315.
9. Chen, X., Makki, K., Yen, K., & Pissinou, N. (2009). Sensor network security: A survey.*IEEE Communications Surveys and Tutorials*,*11*(2), 52-73.
10. Dawahdeh, Z. E., Yaakob, S. N., & Othman, R. R. B. (2016). A New Modification for Menezes-Vanstone Elliptic Curve Cryptosystem. *Journal of Theoretical and Applied Information Technology*,*85*(3), 290.
11. Du, W., Deng, J., Han, Y. S., Varshney, P. K., Katz, J., & Khalili, A. (2005). A pairwise key predistribution scheme for wireless sensor networks. ACM Transactions on Information and System Security (TISSEC), 8(2), 228-258.

12. Eschenauer, L., & Gligor, V. D. (2002, November). A key-management scheme for distributed sensor networks. In Proceedings of the 9th ACM conference on Computer and communications security (pp. 41-47). ACM.
13. Hankerson, D., Menezes, A. J., & Vanstone, S. (2005). Guide to elliptic curve cryptography.Computing Reviews,46(1), 13.
14. Koblitz, N., Menezes, A., & Vanstone, S. (2000). The state of elliptic curve cryptography. Designs, codes and cryptography, 19(2-3), 173-193.
15. Lee, J., & Stinson, D. R. (2004, August). Deterministic key predistribution schemes for distributed sensor networks. In International Workshop on Selected Areas in Cryptography (pp. 294-307). Springer, Berlin, Heidelberg.
16. Pathan, A. S. K., Lee, H. W., & Hong, C. S. (2006, February). Security in wireless sensor networks: issues and challenges. In *2006 8th International Conference Advanced Communication Technology* (Vol. 2, pp. 6-pp). IEEE.
17. Perrig, A., Stankovic, J., & Wagner, D. (2004). Security in wireless sensor networks.
18. Rijmen, V., & Daemen, J. (2001). Advanced encryption standard. Proceedings of Federal Information Processing Standards Publications, National Institute of Standards and Technology, 19-22.
19. Rivest, R. L. (1994, December). The RC5 encryption algorithm. In International Workshop on Fast Software Encryption (pp. 86-96). Springer, Berlin, Heidelberg.
20. Schneier, B. (1993, December). Description of a new variable-length key, 64-bit block cipher (Blowfish). In International Workshop on Fast Software Encryption (pp. 191-204). Springer, Berlin, Heidelberg.
21. Sharma, S., Bansal, R. K., & Bansal, S. (2013, December). Issues and challenges in wireless sensor networks. In 2013 International Conference on Machine Intelligence and Research Advancement (pp. 58-62). IEEE.
22. Singhal, N., & Raina, J. P. S. (2011). Comparative analysis of AES and RC4 algorithms for better utilization. International Journal of Computer Trends and Technology, 2(6), 177-181.
23. Wander, A. S., Gura, N., Eberle, H., Gupta, V., & Shantz, S. C. (2005, March). Energy analysis of public-key cryptography for wireless sensor networks. In Third IEEE international conference on pervasive computing and communications (pp. 324-328). IEEE.
24. Y.-C. Hu, A. Perrig, and D. B. Johnson, "Packet Leashes: A Defense Against Wormhole Attacks in Wireless Networks," Proc. IEEE INFOCOM 2003, Apr. 2003
25. Mousa, A., & Hamad, A. (2006). Evaluation of the RC4 algorithm for data encryption. IJCSA, 3(2), 44-56.
26. Hankerson, D., & Menezes, A. (2011). Elliptic curve cryptography (pp. 397-397). Springer US.
27. Rivest, R. L., Shamir, A., & Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, 21(2), 120-126.

AUTHORS PROFILE

Alakananda Tripathy is an Assistant professor in the Department of Computer Science and Engineering, S'O'A Deemed to be University, Bhubaneswar. She completed her M.Tech in Computer Science and Engineering from Biju Pattnaik University of Technology.

Sateesh Kumar Pradhan is Professor at Utkal University, Bhubaneswar, Odisha. He earned his Ph.D in Computer Science from Utkal University. He has published more than 35 research articles in conference proceedings and journal. His current research interest includes Cloud Computing, Network Security, Computer Security and IT Forensics and Intrusion Detection.

Alok Ranjan Tripathy is an Assistant Professor in the Department of Computer Science, Ravenshaw University, Cuttack. He completed his M.Tech and awarded Ph.D in Computer Science from Utkal University. He is a life member of ISTE and the Odisha Information Technology Society (OITS). His areas of interest include Algorithms, Cloud Computing, Network Security and WSN.

Ajit Kumar Nayak is Professror and head of the Department of Computer Science and Information Technology, Institute of Technical Education and Research, S'O'A Deemed to be University Bhubaneswar. He earned his M.Tech and Ph.D in Computer Science from Utkal University in 2001 and 2010 respectively. He has published more than 40 research articles in conference proceedings and journals. He is a member of IEEE and IET and life member of the Orissa Information Technology Society (OITS). His current research interest includes mobile ad-hoc networks, wireless sensor networks and language computing.
.