European Journal of Computer Science and Information Technology
Vol.8, No.5, pp.18-33, October 2020
Published by ECRTD- UK
Print ISSN: 2054-0957 (Print), Online ISSN: 2054-0965 (Online)

AN ENERGY-EFFICIENT ECC SCHEME FOR WIRELESS SENSOR NETWORKS

Adedoyin Olayinka Ajayi
(Affiliation): Department of Computer Science, Ekiti State University, Ado-Ekiti, Nigeria.
Email: adedoyin.ajayi@eksu.edu.ng

ABSTRACT: *The field of wireless sensor networks (WSNs) combines sensing, computation, and communication into a single tiny device called a sensor. Sensors are equipped with RF radio, processor, memory and hardware. They are also battery powered and therefore have severe energy, bandwidths and memory constraints, and low computational capability. Communication over WSNs is still known to be attack-prone because the constraints of sensors hinder the development of secure modern cryptographic solutions. The Elliptic Curve Cryptography (ECC) technique and the Rivest Shamir Adleman (RSA) algorithm are the two most popular public key cryptographic schemes deployed over wireless networks. The effectiveness of the ECC technique over RSA has been demonstrated in this research. While ECC with very large key sizes is thought to be computationally expensive, it is possible to use smaller primes, or smaller finite fields, with elliptic curves and achieve a level of security comparable to that for much larger integer mod n. Measurements have been made to prove that ECC algorithms can be executed within the memory limits of sensor nodes. An enhanced ECC scheme with collision resistant hash functions is proposed in this research.*

KEY WORDS: wireless sensor networks, sensors, communication,security, ECC, RSA

INTRODUCTION

The field of wireless sensor networks (WSNs) combines sensing, computation, and communication into a single tiny device [1] [2]. The capability of any single device is limited but when networked into hundreds of devices, the technological opportunities are endless [1] [3]. WSNs can comprise of sensor nodes that are either sparsely or densely deployed. A sensor node comprises of a RF radio, processor, memory, battery and sensor hardware. These sensors respond to physical stimuli such as heat, light, sound, pressure, magnetism, humidity, radiation, the presence or nature of biological organisms, geological features, seismic vibrations, specific types of computer data, and so on [1] [4] . The sensors convert the aforementioned stimuli into electrical signal. A simple architecture of a sensor node has been presented in [1] and [5]. WSN applications, such as smart grids, are exposed to numerous security risks such as user privacy invasion and Sybil attack [2] [6].

The risk of intruding on user privacy is also a common factor in the deployment of WSNs. For example, through WSNs, smart grids [6] can capture and analyze data related to power usage, delivery, and generation efficiently. Smart grids can also provide predictive power information (for example, meter reading data, monthly charge, and power usage recommendation) to both utilities and consumers. It can also diagnose power disturbances and outages to avoid the effect of equipment failure and natural accidents. This kind of activity however presents attackers with enough information to use against unsuspecting users [2] [7]. Other Common challenges associated with WSNs are probabilistic channel behaviour, accidental and directed interference or jamming, as well as eavesdropping or unauthorized modification of the communications within the network if not protected by authentication and encryption [7]. WSNs make use of one-to-many and many-to-many communication architectures; however. Lee *et al.* [8] affirmed that this wireless broadcast communication is exposed to security risks. In alternate situations, nodes may be lost due to power exhaustion or more popularly, malicious attacks.

Researchers have viewed attacks against WSNs from two different levels: attack against the security mechanisms and attack against the basic mechanisms such as routing mechanisms [9]. Pathan *et al* [10] outlined the major attacks in WSNs; among these attacks are DoS Attacks on Information in Transit, Sybril attack, Blackhole attack, Hello Flood attack and the Wormhole attack, all labeled insider attacks by Law and Marimuthu [11]. According to Pathan *et al.*, [10], the most serious attacks are these insider attacks. An attacker could compromise a node such that the node drops all packets it is supposed to forward. An attacker could flood the network by broadcasting with a strong signal; or could launch a Sybil attack. Sybil is originally the name of a novel about a character who has multiple personality disorder. A Sybil node claims a different identity to a different neighbour. Doing so, the Sybil node can attract a large amount of traffic to itself. An attacker could launch a wormhole attack. A node would claim itself to be one hop away from the base station so that all surrounding nodes will send packets to it. The node then forwards all the packets via a long-range link to the base station of the attacker [11].

The setup of wireless sensors has severely limited the security options that can be implemented in WSNs [2] [12]. During the creation of an infrastructure, the process of setting up the routes is greatly influenced by energy considerations [12] since sensor nodes are battery-operated. The life time of a sensor node, and hence the sensor network, depends strongly on the battery life time, especially where no power source replenishment is possible in some applications scenarios, including smart grids [12]. Also sensor devices are usually small in power capacity and as such cannot accommodate any more energy than can be provided by batteries. Since the main objectives of sensor nodes are sensing and collecting events, data processing, and data transmission through routing; then the power resource can be divided among three operations which are sensing, computation, and communications. On the other hand, the lifetime of a sensor node plays a key role on energy efficiency and robustness of sensor nodes [13]. As earlier said, route set-up is energy dependent and since the transmission power of a wireless radio is proportional to distance squared or even higher order in the presence of obstacles, multi-hop routing will consume less energy than direct communication. In order to prevent attacks, encryption of the communication data and mutual authentication between sensor nodes are needed.

In spite of the fact that there have been numerous security schemes in the research area, big concerns still remain about how such schemes impact the extreme energy limitations of wireless sensors [14]. Sensor nodes are restricted in power consumption (sensors are battery-powered), bandwidth, memory, and calculation capability. As noted in Simplicio Jr [15], Commercial motes usually have 8-128Kb of code memory, 4-10Kb of data memory (RAM) and are equipped with 8-

bit or 16-bit processors operating at 4-16MHz. These constraints hinder the deployment of most modern cryptographic solutions known to be secure. Complex algorithms in the cryptographic world usually take longer to run and also consume more energy than can be provided by battery-powered sensors. Crypto methods, such as encryption and authentication using public-key cryptosystems, are not logical either, because the computational capability and low memory of sensor nodes prevent them from operating such algorithms within ample time [8]. Numerous researches have focused on developing crypto-algorithms for sensor nodes; however there is always an obvious trade-off between the strength of those algorithms and the need to develop them to operate within the computational and energy capacity of the sensor nodes. Most known crypto-algorithms have sacrificed the key size of associated schemes, while some others have stopped short of defining relationships between sensor nodes and central base stations. This research has attempted to find solutions to some of the issues including computational and energy efficiency. The research was set out to design an energy-efficient intrusion prevention model in Wireless Sensor Networks, simulate the model designed and determine the robustness of the developed model by comparing with some existing intrusion prevention system models using standardized parameters.

Review of commonly used Public Key Cryptosystems

The Rivest-Shamir-Adleman (RSA) cryptosystem is one of the most popular Public Key algorithms used in cryptographic systems. RSA uses multiplication of large primes and modulo arithmetic. The disadvantage is that it is slow on limited environments with low memory and processor power because of the use of modulo arithmetic with long operands. Also, to be secure, RSA requires longer keys [16]. The Elliptic Curve Cryptography (ECC) Technique is another PK system based on the intractability of ECDLP [17]. It can implement with smaller key sizes and bandwith savings in comparison with RSA, and specifically, there is choice of field over which the curve can be defined to make it suit limited environs. Some suitable mentions include NTRU, based on polynomial rings and Braid, based on braid groups.

Comparing RSA and ECC

Numerous researches have carried out a comparison analysis of RSA and ECC. Popular factors considered in analysis include size of keys, time of execution of their encryption processes and time of execution of their decryption processes. It has been determined that ECC of key size 113 bits provides the same level of security as RSA of key size 512 [17] [18].

Analytical results comparing key sizes, as depicted in [18], is shown in the Table 1 below.

Table 1: Comparison by Identical Security Level Providing Key Sizes (Source: [18])

RSA(bit)	ECC (bit)
512	113
768	136
1024	160
2048	282
4096	409

The advantage of ECC over RSA in relation to key sizes is made obvious from the analysis shown in Table 1. Also, another advantage of ECC is that there is a choice of field over which the curve can be defined to make it suit limited environments. The size of implementable keys is enough leverage for us to choose ECC as our base algorithm as we intend to choose much larger keys for our implementation process. Also it has been determined that, while RSA is faster than ECC in its encryption processes, ECC tends to decrypt faster and also creates much smaller file sizes for encrypted data, as shown in Table 2. Also, using ECC, the size of data that can be encrypted in one step is much larger than in case of RSA, where this restriction (number of steps) is quite strong [18].

Table 2: Size of Data Files after Encryption with RSA versus ECC (Source: [18])

Common Key Size (bit)	**Size of data to be encrypted (byte)**	**Size of encrypted data with RSA (byte)**	**Size of encrypted data with ECC (byte)**
ECC 113 = RSA 512	22	64	73
ECC 131 = RSA 768	22	96	77
	54	96	109
ECC 160 = RSA 1024	22	128	83
	54	128	115
	86	128	147
ECC 283 = RSA 2048	22	256	115
	54	256	147
	86	256	179
	214	256	307
ECC 409 = RSA 4096	22	512	147
	54	512	179
	86	512	211
	214	512	339
	470	512	595

Endrodi [17] compared the time for encryption for both systems. The public key for RSA used for the experiment is [*e*, *m*], (*m* = *p***q*; *p* and *q* are primes). Then the encryption function is shown in equation 1.

$$E\ (x) = x^{e} \bmod m; \qquad 1$$

where $x < m$;

Typical parameters used for the experimental process include 512 – 4096 bit modulus, with a recommended 1024 bit; e was chosen to be a small value (e = 65537 = 10001H) to speed up the process. The maximal size of data that can be encrypted in one step is determined by the modulus [19].

The public key used for ECC is given by P (= B⊗B⊗…⊗B = k⊙B), where r is a random integer. The encryption function is given by

$$E(M) = (Y_1, Y_2) \tag{2}$$

where $Y_1 = r \odot B$, $Y_2 = M \otimes r \odot (k \odot B)$ and r is a random integer.

The parameters used in the experimental process include a 110 – 570 bit key size, with a recommended 160-bit key size. As we have mentioned, to reach a given security level, a much smaller key size is needed in case of ECC than in case of RSA, that is, the *security-per-key-bit* rate is higher. To speed up the process, the researcher used pre-computed tables for B and P.

The graph in Figure 1 shows the comparison of encryption times for RSA and ECC. When the exponent (e) for RSA is set to 65537, RSA is actually 4-5 times faster than ECC. But on randomizing e, the encryption time is slower than ECC defined over prime fields. ECC over other field types takes more time to process.

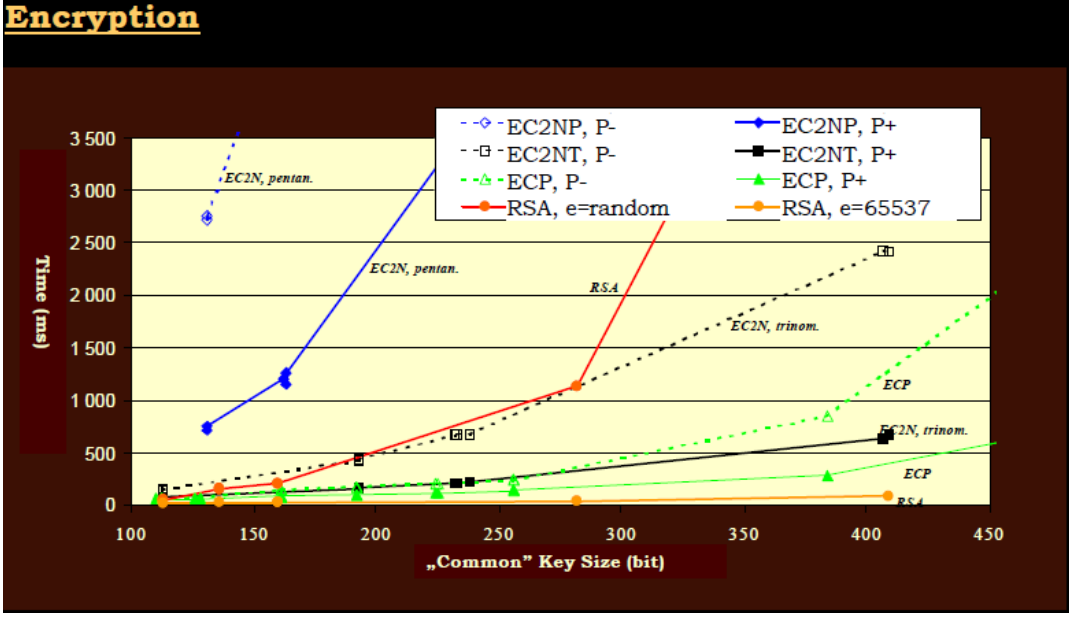


Figure 1: Comparing Time of Encryption: RSA v ECC (Source: [18])

The secret key used for the decryption process of RSA is (d, m). The decryption function is given by

$$D(y) = y^d \bmod m \tag{3}$$

The secret key used for the decryption process of ECC is $D(Y_1, Y_2) = Y_2 \otimes (-)\, k \odot Y_1$. The parameters were retained.

The graph in Figure 2 shows the comparison of decryption times for RSA and ECC. It is obvious from the graph that ECC decrypts faster than RSA.

All these variables show a justification of our decision to choose ECC over RSA.

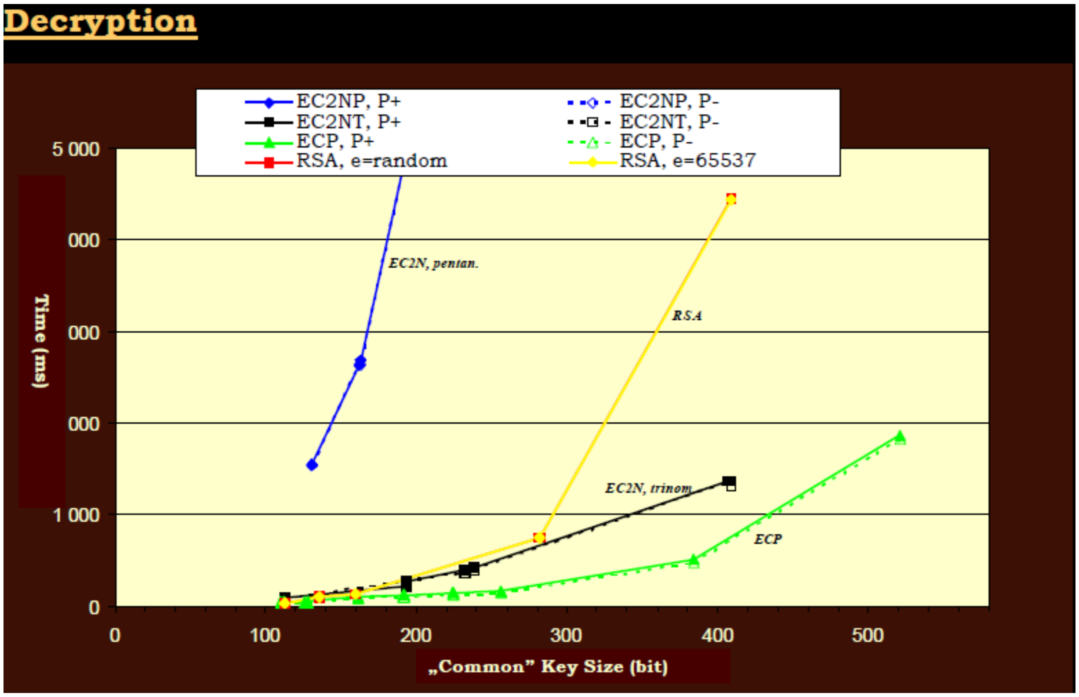


Figure 2: Comparing Time of Decryption: RSA v ECC (Source: [18]).

Description of the Problem

Figure 3 shows the basic architecture of the frame and the presence of an attacker that wants to be a part of the network. The deployment of new nodes is a valid function in the frame since the span of a sensor network may be extended, and such an attacker might take advantage of that function. Nodes in the sensor network may lose data because of power exhaustion or malicious attacks by outsider nodes, almost inevitably [14] [19]. New nodes are therefore also inevitably deployed.

In order to prevent malicious nodes from joining the sensor network, access control is a designed requirement for controlling sensor node deployment. In other cases, an adversary attempts to take over a valid node in the network and thus listen to and take part in the network communications. The scheme proposed does not only establish two keys in its authentication procedure for new nodes during deployment, it also prevents outside interference from attacking nodes, hoping to read the contents of the message sent between nodes. The mode of controlling access for nodes wishing to join the network is each user's public key which should be communicated from the central node during message sending; but message decryption is done with both the public key and the private (secret) key. The schemes should be simple enough to offer computational efficiency, storage, energy, and bandwidth savings. It is assumed that all nodes have the same transmission range with other nodes in the frame network.

In this scheme, there is no time boundary for accessing the frame, but a time *w* exists where *w* is the expiration time of a node, due to power limitations and limited battery life. Another instance to justify the expiration variable is a case where a sensor node *A* cannot verify another node *E*, then enabling node *E* to consume memory or computing time of the recipient node *A* and eventually exhaust its resources, the simple reason is that all messages are being protected from current messages to future messages. This implies there is a node N_{NV} (for $V = 1,2,3,\ldots\beta$) that is part of or aspiring to be part of the frame; assuming T to be an instance time period for the frame.

Without losing the simplicity concept, the scheme we developed aimed to achieve the following specific objectives:

I. Access Control and Node Authentication: a node hoping to be a message recipient proves itself to be an *s*-secure node, thereby establishing its identity with its neighboring nodes and showing that it has the right to access the frame through access control.

II. Keys Establishment: as earlier asserted, the scheme makes use of asynchronous mode of encryption in its authentication, leading to creation of two keys- public and secret. This makes sure that there is a common key between any two nodes; the other key is unique for each node. A shared key separates the frame from other sensor networks in the region, as it denotes the relationship that exists between the node and one of its direct neighbours. Through the concepts of Huang [20] and based on elliptic curve cryptography (ECC) [17] [18], the keys are created between a node and its neighboring nodes to provide secure message sending with minimum power consumption. Other methods in the developed Scheme will be defined.

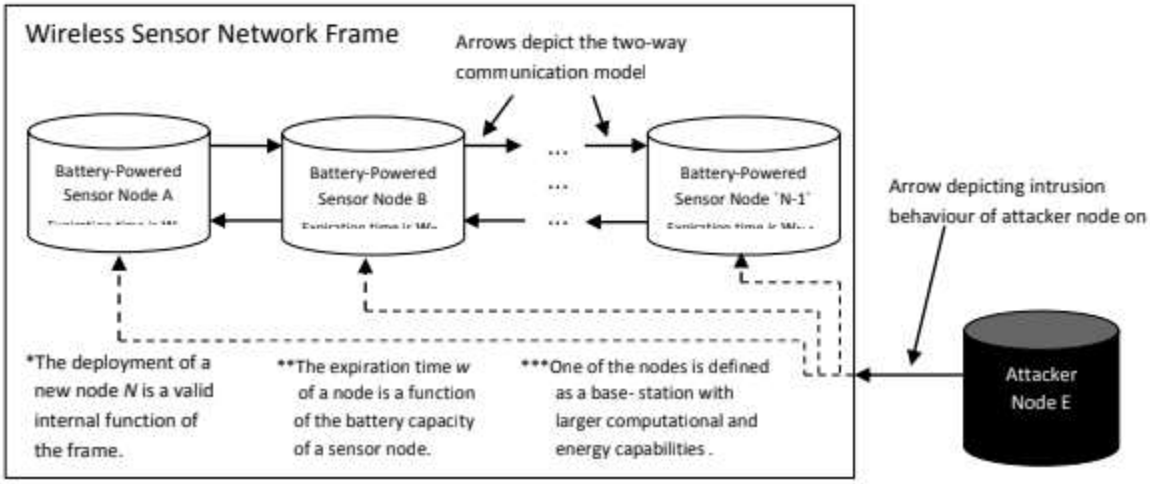

Wireless Sensor Network Frame
Arrows depict the two-way communication model
Battery-Powered Sensor Node A
Battery-Powered Sensor Node B
Battery-Powered Sensor Node 'N-1'
...
Arrow depicting intrusion behaviour of attacker node on
Attacker Node E
*The deployment of a new node *N* is a valid internal function of the frame.
**The expiration time *w* of a node is a function of the battery capacity of a sensor node.
***One of the nodes is defined as a base- station with larger computational and energy capabilities .

Figure 3: Architecture of the WSN frame showing nodes and presence of an attacker (broken lines show attack routes)

Developed Frame Schema

The proposed system is made up of two modules based on the Elliptic Curve Cryptography technique which explores the difficulty of the Discrete Logarithm Problem (DLP) [17]. The modules are the Cryptographic and Intra-node Messaging units. The cryptographic module comprises authentication and key establishment protocols.

Let P and Q be two points on an elliptic curve such that:

$$Q = P^k \qquad (4)$$

From equation (4), k is a scalar and it is computationally infeasible to obtain k, if it is sufficiently large. Thus, k is the discrete logarithm of Q to base P. That is:

$$k = (Q) \qquad (5)$$

Furthermore, the elliptic curve is defined over a finite prime field Fp to make it accurate and efficient.

An elliptic curve over a finite field GF(q) (a Galois Field of order q) is composed of a finite group of points (x_i, y_i), where integer coordinates x_i, y_i satisfy the long Weierstrass form, as shown in equation (6):

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$$

(6)

and the coefficients a_i are elements in GF(q). The field GF(q) (q is a prime) is simplified to the expression shown in equation (7):

$$y^2 = x^3 + ax^2 + b \qquad (7)$$

where $a,b \in$ GF(q).

More concretely,

$$y^2 mod\, p = x^3 + ax + b\, mod\, p \qquad (8)$$

where $4a^3 + 27b^2$ mod $p \neq 0$.

The elements of the finite field are integers between 0 and p – 1. The prime number p is chosen such that there is a finitely large number of points on the elliptic curve to ensure a secure cryptosystem.

Each value of a and b gives a different elliptic curve. Every point (x, y) which satisfies the above equation, in addition to a point at infinity O lies on the elliptic curve.

The public key of the proposed system is a point in the curve and the private key is a random number. The public key is obtained by multiplying the private key with the generator point $G(x_G,y_G)$ in the curve. The point generator G, the curve parameters a and b, n (the order of the elliptic curve), h (an hash function equivalent to h(E(F_p)/n); h(E(F_p)) is the number of points on the elliptic curve) and p constitutes the domain parameters of ECC.

During system frame deployment, the frame's base station (with expiration time T) chooses a large prime number q, for $2^{160} < q < 2^{256}$, and an elliptic curve E_q. The initial assumption is that the base station node is totally secure and cannot be breached. Then, the frame selects a secret key $x \in E_q$ and computes the system's public key $Q = xP$ of the point over the elliptic curve E_q [20]. The frame's base station then generates a collision resistant hash function h. For each node AB_{NV} (for V= 1,2,3,…, β) that wants to be a part of the network, the frame's base station generates a random number r_{NV} and the expiration time $w_{NV} > T$ then computes the point

$R_{NV} = r_{NV}P = (Rx_{NV,} \parallel Ry_{NV})$ (9)

Assuming Nodes (A)lice and (B)ob have surrounding nodes, and assuming $\beta = (1,2,3,…,n)$, the total number of nodes in the frame is then the set {A, B, AB_{N1}, AB_{N2}, AB_{N3}, …, AB_{Nn}}in the frame region. Nodes A (or B) generates a random number $r_{A(B)}$ and the expiration time $w_{A(B)} > T$ then computes the point:

$R_{A(B)} = r_{A(B)}P = (Rx_{A(B),}Ry_{A(B)})$, (10)

and the value

$s_{A(B)} = r_{A(B)} + c_{A(B)}x$ mod$q$, (11)

where $c_{A(B)} = h(N_{A(B)} \| Rx_{A(B)} \| Ry_{A(B)} \| w_{A(B)})$, depending on whether A(or B) is the frame base station. The function h refers to the previously generated hash function.

For Node A and Node B, $h(N_A\|Rx_A\|Ry_A\|w_A) \approx h(N_B\|Rx_B\|Ry_B\|w_B)$ should be almost non-existent, that is, it should be “hard" to find two distinct messages sent within the frame that hash to the same result.

Each neighbour node AB$_{NV}$ (V= 1,2,3,…, β), the system base station first generates a random number r_{NV} and the expiration time $w_{NV} > T$ then computes the point

$$R_{NV} = r_{NV}P = (Rx_{NV},Ry_{NV}) \qquad (12)$$

and the value

$$s_{NV} = r_{NV} + c_{NV}x \bmod q, \qquad (13)$$

where $c_{NV} = h(N_{NV}\|Rx_{NV}\|Ry_{NV}\|w_{NV})$ and $\|$ refers to the concatenation of operations function.

Therefore, each node AB$_{NV}$, $h(N_{N1}\|Rx_{N1}\|Ry_{N1}\|w_{N1}) \approx h(N_{N2}\|Rx_{N2}\|Ry_{N2}\|w_{N2}) \approx h(N_{N3}\|Rx_{N3}\|Ry_{N3}\|w_{N3}) \approx \ldots \approx h(N_{N\beta}\|Rx_{N\beta}\|Ry_{N\beta}\|w_{N\beta})$ should be almost non-existent.

It is necessary, however, to choose a fixed length for group generator P, instead of defining the length in an interval. This condition is necessary to make our hash function h attack resistant. In order to improve on the scheme developed by Huang [20], a prime $q \approx 2^{256}$ giving a 1:12 key ratio for equivalent RSA key size is used. The definition of q relates to a random attacker activity where the attacker, node E, selects a random message and hopes that the modification remains undetected. An ideally secure hash function has probability of successful attack as $l=2^n$. The feasibility of this attack depends on the action taken in case of detection of an erroneous result on the expected value of a successful attack and on the number of attacks that can be carried out. In most applications, this implies that $n = 32$ bits is not sufficient, since it is possible that the opponent can attack several messages off-line and in parallel. In that case, the hash code should have a length of 64 bits or more. For Elliptic curves with 256 bits key-size, the size of a signature on a 4-byte hash is about 256*2 = 512 bits = 64 bytes.

In the next phase, the frame preloads the elliptic curve Eq, the frame public key Q, the generator P of the group $G = (P)$ over the elliptic curve Eq, c-r hash function h, its expiration time $w_{A,B,NV}$, and the secret pair $(R_{A,B,NV}, s_{A,B,NV})$ to node $N_{A,B,NV}$, for $V = 1, 2, \ldots, \beta$; where β is an integer. Rx_{NV} and Ry_{NV} are the x-component and y-component of point R_{NV}.

Keys Establishment

The developed authentication and key establishment protocol is shown in Table 3. According to the Diffie-Hellman algorithm over elliptic curve, two nodes, N_A and N_B, can obtain their common shared (pair-wise) key K_{AB} by using their secret parameter t_A and t_B , respectively, since $K_{AB} = t_BC_A = t_AC_B = t_At_BP$ over elliptic curve E_q, in so doing, authenticating each other for secured transmissions. Recall we assume T to be an instance time period in the frame. The key establishment process for two nodes N_A and N_B is further described below.

Step 1:

The node N_A generates a random number t_A and computes the point C_A over the elliptic curve Eq, where

$$C_A = t_AP \qquad (14)$$

It then sends C_A and its identity N_A to the node N_B.

Similarly, node N_B generates a random number t_B and computes the point C_B over elliptic curve Eq, where

$C_B = t_BP$ (15)

Table 3: The proposed authentication and key establishment protocol.

Node (*A*)lice	**Relationship with respect to broadcasting and broadcast time period *T***	**Node (*B*)ob**
Initiate node A over a random point *t* on the elliptic curve and compute $C_A = t_AP$. Where *P* is the generator of a cyclic group $G = (P)$ of points over the elliptic curve *Eq*.	$\xrightarrow{C_A, N_A}$ $\xleftarrow[C_B, N_B]{}$	Initiate node B over a random point *t* on the elliptic curve and compute $C_B = t_BP$. Where *P* is the generator of a cyclic group $G = (P)$ of points over the elliptic curve *Eq*.
Compute $C_{AB} = t_AC_B = (Kx_{AB}, Ky_{AB})$ since *P* is a pair point on the curve, and $z_A = t_A + m_As_A\text{mod}q$ m_A is a 4-bit random number * $h(N_A \Vert Kx_{AB} \Vert Ky_{AB})$. *h* is a cryptographic hash function	$\xrightarrow{z_A, R_A, w_A}$ $\xleftarrow[z_B, R_B, w_B]{}$ where *w* is the expiration time for a sensor node. The function *R* is a pair based on the elliptic curve and defined over the cyclic group *G* and group generator *P*.	Compute $C_{AB} = t_BC_A = (Kx_{AB}, Ky_{AB})$ since *P* is a pair point on the curve, and $z_B = t_B + m_Bs_B\text{mod}q$ m_B is a 4-bit random number * $h(N_B \Vert Kx_{AB} \Vert Ky_{AB})$ *h* is a cryptographic hash function
Check $w_B > T$ Verify $z_BP = \mathrm{C}_B + m_B (R_B + c_BQ)$ (→Auth_N)	$\xrightarrow{\text{verify}}$ $\xleftarrow[\text{verify}]{}$	Check $w_A > T$ Verify $z_AP = \mathrm{C}_A + m_A (R_A + c_AQ)$ (→Auth_N)

N_B also sends C_B and its identity N_B to node N_A. The random numbers t_A and t_B are not re-useable.

Step 2:

After receiving C_B, node N_A computes a shared session key $K_{AB} = t_AC_B = t_At_BP = (Kx_{AB} ,Ky_{AB})$, where Kx_{AB} and Ky_{AB} are the *x*-component and *y*-component of K_{AB}, respectively, and signature

$z_A = t_A + m_As_A\text{mod } q$, (16)

where $m_A = \omega * h(N_A \Vert Kx_{AB} \Vert Ky_{AB})$, and ω is a 4-bit random binary integer.

Then, it delivers the signature z_A, the expiration time w_A of N_A, and its point $R_A = (Rx_A,Ry_A)$ to node N_B, where Rx_A and Ry_A are the x-component and y-component of R_A, respectively .
Similarly, after receiving C_A, node N_B computes a shared session key $K_{AB} = t_BC_A = t_Bt_AP = (Kx_{AB} ,Ky_{AB})$ and signature;

$$z_B = t_B + m_Bs_B \bmod q, \tag{17}$$

where $m_B = \varpi * h(N_B \| Kx_{\mathrm{AB}} \| Ky_{AB})$, and ϖ is the complement of a previously used ω on the next authentication procedure.

Table 4: Frame Notations

Symbol		**Description**
N_A	:	Assumed Base Station Identity
$N_{NV}\,(V = 1,2,3,\ldots,\beta)$	:	Nodes (Neighbors) Surrounding $N_{A.}$
x, also $\approx r \longrightarrow (R_x, R_y)$	:	Point on E_q representing individual node secret key on the affine coordinate system
Q	:	System public key
$K_{A,NV}$	:	Shared session key between N_A and N_{NV}
$z_{A,NV}$	:	Signature of N_A and neighbor nodes N_{NV}
$w_{A,NV}$	:	Expiration time of N_A and neighbor nodes N_{NV}
T	:	Time instance period for the frame
Auth_N	:	Authentication Code of Nodes that enables capability to send messages
$h(\,)$	:	Hash function (collision resistant)
$\|$	:	Concatenation

Then, it also delivers the signature z_B, the expiration time w_B, and its point $R_B = (Rx_B ,Ry_B)$ to node N_A. Also, Rx_B and Ry_B are the x-component and y-component of R_B, respectively.
Table 4 provides a list of notations that are enabled in the frame.

Node Authentication

Confirming the identity and message sending and receiving capability of a neighbouring node N_B, Node N_A confirms their shared session key K_{AB} in the following way:

Step 1

It is assumed that Node N_A is a legit node, and the base station. Node N_A compares the expiration time w_B of N_B with the time instance T relayed from the frame. If the expiration time $w_B < T$ then node N_B is rejected and continued otherwise.

Step 2

Node N_A computes c_B and m_B, where

$$c_B = h(N_B \| Rx_B \| Ry_B \| w_B) \tag{18}$$

and

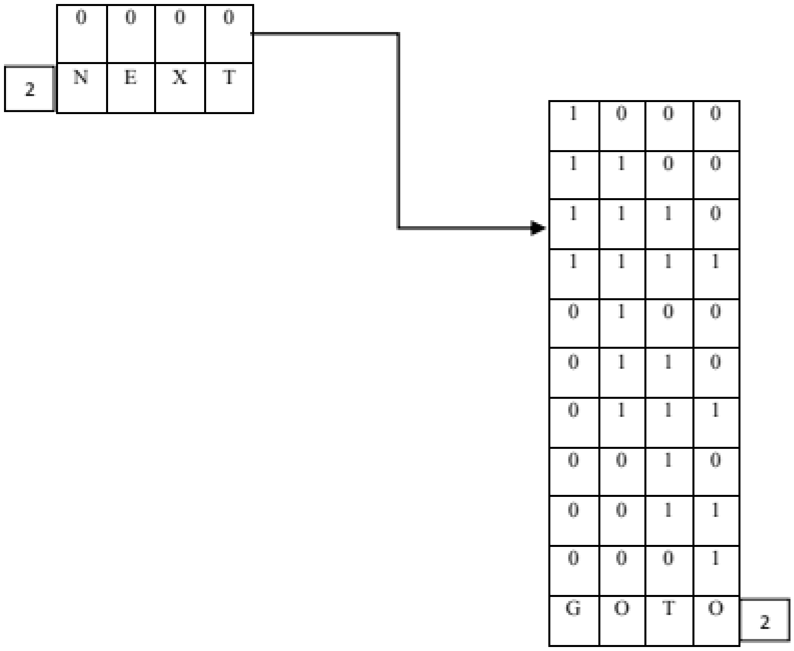
$$m_B = \omega * h(N_B \| Kx_{A,B} \| Ky_{A,B}\,) \tag{19}$$

Step 3

Node N_A checks if the condition $z_BP = C_B + m_B(R_B + c_BQ)$ (→Auth_N) holds, where Q is the public key of the frame. And then if the condition holds, then the signature is accepted; otherwise, the signature is rejected.

Step 4
Base station N_A always complements a bit of a previously used 4-bit binary number ω to derive the next binary number ϖ to be used on the next node authentication; ω is reset after every binary ϖ generated. In this manner, the 4-bit binary sequence is automatically exhaustive without following the numbering pattern. This particular property is illustrated in Figure 4 and allows the hash function to always satisfy the c-r property. In summary, the base station keeps a tab of previously hashed signals/messages by generating a one-bit nonce (on the next authentication trip) on every 4-bit binary number attached to a message m.

Step 5
ϖ is derived from the previous authentication procedure, and in the same steps as 2, 3 and 4, the next neighbour node, N_{NV} (for $V = 1,2,3,\ldots,\beta$), and from the signature z_{NV}, the expiration time w_{NV} , and its point $R_{NV} = (Rx_{NV} ,Ry_{NV})$ of node N_{NV}; if $w_{NV} < T$, the node is denied, otherwise, the condition $z_{NV}P = A_{NV} + m_{NV}(R_{NV} + c_{NV}Q)$ (→Auth_N) is verified, and then node N_A, the base node, can verify the identity of node N_{NV}, where $c_{NV} = h(N_{NV}\_Rx_{NV}\_Ry_{NV}\_w_{NV})$ and $m_{NV} = \varpi * h(N_{NV} \| Kx_{A,NV} \| Ky_{A,NV})$.



	0	0	0	0
2	N	E	X	T

1	0	0	0	
1	1	0	0	
1	1	1	0	
1	1	1	1	
0	1	0	0	
0	1	1	0	
0	1	1	1	
0	0	1	0	
0	0	1	1	
0	0	0	1	
G	O	T	O	2

Figure 4: The base station attached a one-bit nonce to previously hashed signals
In this manner, two nodes, N_A and N_B , could achieve mutual authentication and generate a common session key $K_{A,B}$, as shown in Table 3.

A detailed analysis of the Auth-N protocol is given in Table 5. In the table, for any Node N_{NV} that wants to join the frame and send messages to other nodes, it first agrees a session and a corresponding session key with the recipient node N_B. N_B's first action is to contact the frame's base station to confirm the authenticity of the sending node. As previously assumed, Node N_A is always the frame's base station. The total time, T_n, it takes nodes to achieve authentication and compute a shared key is an important variable to measure the expected energy consumption rate of the developed scheme.

Table 5. Auth_N Protocol for Node Authentication

N_{NV} computes:	$C_{NV} = t_{NV}P$
N_{NV} receives C_B and computes:	$K_{NV,B} = t_{NV}C_B = t_{NV}t_BP = (Kx_{NV,B}, Ky_{NV,B})$
N_{NV} computes:	$m_{NV} = \omega * h(N_{NV} \Vert Kx_{NV,B} \Vert Ky_{NV,B})$,
:	$z_{NV} = t_{NV} + m_{NV}s_{NV} \bmod q$
:	w_{NV}
$N_{NV} \rightarrow N_{B:}$	C_{NV}, m_{NV}, z_{NV}, w_{NV}, affine pair (Rx_{NV}, Ry_{NV}) for R_{NV}
N_B computes:	w_B and pre-loaded values c_B, z_B, m_B, $R_{B,}$ C_B
$N_B \rightarrow N_A$	C_{NV}, c_{NV}, m_{NV}, z_{NV}, w_{NV}, affine pair (Rx_{NV}, Ry_{NV}) for R_{NV}, c_B, w_B, z_B, m_B, $R_{B,}$ C_B
N_A receives:	C_{NV}, c_{NV}, m_{NV}, z_{NV}, w_{NV}, affine pair (Rx_{NV}, Ry_{NV}) for R_{NV}, c_B, w_B, z_B, m_B, $R_{B,}$ C_B
N_A computes:	Check if z_B, w_B and w_{NV} are valid.
	w_{NV}, $w_B < T$? *Terminate*
	w_{NV}, $w_B > T$? *Continue Authentication*
	Confirm $z_BP = C_B + m_B(R_B + c_BQ)$
	Verify N_B
	$c_{NV} = h(N_{NV} \Vert Rx_{NV} \Vert Ry_{NV} \Vert w_{NV})$
N_A verifies:	$z_{NV}P = C_{NV} + m_{NV}(R_{NV} + c_{NV}Q)$
	If true: *verify* N_{NV} signature sign in
	Otherwise: *Reject* Signature
$N_A \rightarrow N_B$:	Auth_N, C_{NV}, m_{NV}, z_{NV}, $w_{NV,}$ $C_{A,}$ $w_{A,}$ z_A
N_B computes:	Check if w_A and z_A is valid.
	$w_A < T$? *Terminate*
	$w_A > T$? *Continue Authentication*
	Verify N_A
	Auth_N C_{NV} received
$N_B \rightarrow N_{NV}$	Auth_N
N_{NV}	Verify N_B (Auth_N)
RESET ω *to* ϖ	

Keys Renewal

To eliminate security risks, the frame's shared (public) key Q must be refreshed frequently. To renew the public key, the frame's central node N_A selects a new secret key $x' \in Z_q$ and computes the new frame public key $Q' = x'P$ of the point over the elliptic curve E_q and then broadcasts Q' to all nodes in the frame. Each node N_{NV} surrounding legitimate node N_A then computes R (recall R is a pair on the EC)

$$R_A = r'_A P = (Rx'_{A,} Ry'_A), \tag{19}$$

and the value

$$s_A = r_A + c_A x' \bmod q, \tag{20}$$

where $c_A = h(N_A \| Rx'_A \| Ry'_A \| w_A)$.

The system does the same for each neighbour node N_{NV} (for $V = 1,2,3,\ldots,\beta$) in its system and in the next phase, the frame preloads Eq, the frame's public key Q, the generator P of the group $G = (P)$ over the elliptic curve Eq, the new c-r hash function $h(\,)$, nodes' expiration time $w_{A,NV}$, and the latest secret pair ($R_{A,NV}$, $s_{A,NV}$) to node $N_{A,NV}$. The shared session key $K_{A,NV}$ and individual node signature z are obtained likewise.

CONCLUSION AND FUTURE WORK

This research has indicated the extreme energy limitations of these sensor nodes. Since a sensor node is battery-operated, the life time depends strongly on the battery life time. Also, the lifetime of a sensor node plays a key role on its energy efficiency and robustness since route set-up of WSNs is energy dependent. Research has, most times, focused on making sensor networks feasible and useful; and not much emphasis has been placed on security.

This research has also indicated that the increasingly widespread deployment of sensor networks has almost simultaneously heightened security issues. Since, transmitted data in sensors is via wireless communication, mechanisms to prevent unauthorized users from prying on transmitted information or introducing malicious data into the network have to be put in place, to prevent, for example, leakage of user private information. Further, WSNs make use of one-to-many and many-to-many communication architectures; this wireless broadcast communication is exposed to security risks, to put it more concretely, an adversary can eavesdrop and alter communication messages, and insert malicious messages. In alternative situations, nodes in a sensor network may be lost due to power exhaustion or malicious attacks. In order to extend the lifetime of the sensor network, new node deployment is necessary.

During new node deployment, in military scenarios for example, adversaries may directly deploy malicious nodes or manipulate existing nodes to introduce malicious ''new'' nodes through many kinds of attacks. All of these facts make it very necessary to absolutely guarantee the safety and security of information communicated in the WSN. The severe energy constraints of WSNs have affected the deployment of robust security schemes that can guarantee safety and security of information communicated in the WSN; these constraints hinder the deployment of most modern cryptographic solutions known to be secure. Complex algorithms in the cryptographic world usually take longer to run and also consume more energy than can be provided by battery-powered sensors. While there have been numerous security schemes in that research area, concerns still remain about how such schemes impact the extreme energy limitations of wireless sensors. The researches on security have sacrificed strength and robustness of their schemes because of the low computational capacity, bandwidth, memory and energy constraints in sensor nodes. This research work has developed a security scheme, based on elliptic curve cryptography system, which can execute within the energy limits of wireless sensor nodes.

In our future work, the developed enhanced ECC scheme with collision resistant hash functions will be implemented on the J-Sim platform and measurements will be conducted to determine possible improvement in network lifetime. Security simulations will be carried out. The effect of selected variables on sensor node lifetime will be described and analyzed, and a comparison with other models for security and energy consumption will be made.

References

[1] Ajayi A.O., Alese B.K. and Adetunmbi A.O. (2015) "*An Appraisal of Wireless Sensor Networks: Profiles and Characters*", International Journal of Communications, Network and System Sciences. USA

[2] Farooq U. (2019), "Wireless Sensor Network Challenges and Solutions", Available at: https://www.researchgate.net/publication/331299729_Wireless_Sensor_Network_Challenges_and_Solutions, Last accessed: April 2020.

[3] Hill J.L. (2003). "*System Architecture for Wireless Sensor Networks*" Doctor of Philosophy dissertation, University Of California, Berkeley.

[4] Ilyas M. and Mahgoub I. (2005) "*Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems*" .CRC PRESS LLC, Boca Raton, USA.ISBN 0-8493-1968-4. Doi: http://dx.doi.org/10.1201/9780203489635

[5] Kao W. (2012). "*Sensor Devices and Sensor Network: Applications for the Smart Grid/Smart Cities*". Sensors Con, 2012.

[6] Ajayi A.O., B.K. Alese, S.E. Fadugba and K.O. Owoeye (2014), "*Sensing the Nation: Smart Grid's Risks and Vulnerabilities*", International Journal of Communications, Networks and System Sciences. USA.

[7] Akyol B., Kirkham H, Clement S., and Hadley M. (2010). "*A Survey of Wireless Communications for the Electrical Power System*", Technical Report, Pacific Northwest National Laboratory.

[8] Lee H.R., Choi Y.J., and Kim H.W. (2005). "*Implementation of TinyHash based on Hash Algorithm for Sensor Network*", Proceedings of World Academy of Science, Engineering and Technology, Volume 10, Issn 1307-6884. December.

[9] Wahid A.and Kumar P. (2015). "*A Survey On Attacks, Challenges and Security Mechanisms In Wireless Sensor Network*", International Journal for Innovative Research in Science & Technology, Vol. 1, Issue 8, January.

[10] Pathan, A.S.K., Hyung-Woo Lee and Choong Seon Hong, (2006) "*Security in wireless sensor networks: issues and challenges*", Advanced Communication Technology (ICACT), Page(s):6.

[11] Ryu J.H., Irfan M. and Aamir Reyaz A. (2015). "*A Review of Sensor Network Issues and Robotics*", Journal of Sensors: Recent Advances in Security and Privacy for Wireless Sensor

Networks pp 10-23. Hindawi Publishing Corporation. Available at: http://downloads.hindawi.com/journals/specialissues/247350.pdf

[12] Kumar P., Singh M.P. and Triar U.S. (2012). "*A Review of Routing Protocols in Wireless Sensor Network*", International Journal of Engineering Research & Technology (IJERT), Vol. 1 Issue 4, June

[13] Al-Obaisat Y and Braun R. (2006)."*On Wireless Sensor Networks: Architectures, Protocols, Applications, and Managment*". Institute of Information and Communication Technologies ,University of Technology, Sydney, Au.

[14] Elshrkawey M., Elsherif S.M. and Wahed M.E. (2018), "An Enhancement Approach for Reducing the Energy Consumption in Wireless Sensor Networks", Journal of King Saud University - Computer and Information Sciences, Volume 30, Issue 2, April, Pages 259-267Available at: https://www.sciencedirect.com/science/article/pii/S131915781730023X, Last accessed: February 2020.

[15] Simplicio Jr M.A. (2010). "*Message Authentication Algorithms for Wireless Sensor Networks*", PhD thesis submitted to EscolaPolitecnica da Universidade de Sao Paulo.Sao Paulo, Brazil.

[16] Karu, P. (2000). "*Practical Comparison of Fast P-K Cryptosystems*", Tik-110.501 Seminar on Network Security, HUT TML

[17] Alese, B.K. (2004). "*Design of Public Cryptosystem using Eliptic Curve*". Ph.D Thesis, Federal University of Technology, Akure, Nigeria.

[18] Endrodi C. (2002). "*Efficiency Analysis and Comparison of Public Key Algorithms*", CS ^{2}, Conference of PhD Students in Computer Science 4th July, Search Laboratory.

[19] Le X.H., Lee S., Butun I., Khalid M., Sankar R., Kim M., Han M., Lee Y-K., and Lee H. (2009). "*An Energy-Efficient Access Control Scheme for Wireless Sensor Networks based on Elliptic Curve Cryptography"*, Journal Of Communications and Networks, Vol. 11, No. 6, December.

[20] Huang, H-F., (2011). "*A New Design of Access Control in Wireless Sensor Networks*", International Journal of Distributed Sensor Networks, Volume 2011, Article ID 412146, 7 pages doi:10.1155/2011/412146. Hindawi Publishing Corporation.