Journal of Engineering and Applied Sciences 13 (24): 10356-10363, 2018
ISSN: 1816-949X
© Medwell Journals, 2018

ECC and AES Based Hybrid Security Protocol for Wireless Sensor Networks

Susan Mohammed, Saif M. Kh. Al-Alak and Hussein A. Lafta
Department of Computer Science, College of Science for Women, University of Babylon, Hillah, Iraq

Abstract: Many applications employ Wireless Sensor Networks (WSN) like tracking, controlling, monitoring and transferring. One of the important features of WSN is security. However, in WSN the solutions of security are different from traditional networks due to limited resources and computational constraints. As long as computer techniques are rapidly evolving, security protocols implemented over time can be broken. Keeping secrecy in WSN's data transmission is a big challenge since this type of wireless networks allows remote data transfer. Cryptography is the security solution of most WSN applications. Recently, asymmetric and symmetric algorithms have been widely used in WSN systems with low computational energy power. This study suggests building a hybrid security protocol of two low power encryption algorithms which are asymmetric Elliptic Curve Cryptography (ECC) algorithm with Advanced Encryption Standard (AES) symmetric algorithm. Different key seeds are generated using a proposed combination of multiple ECC algorithms with a hash function. The resulted keys are used to encrypt the secret data using an AES algorithm in WSN communication. Randomized results of the proposed protocol are tested using the Diehard test. The proposed system has high randomness (more security) than other hybrid systems of previous researches.

Key words: AES, ECC, WSN, randomness, hybrid security, WSN security, key generating

INTRODUCTION

Wireless Sensor Networks (WSN) used in several important areas, for example, the environment, military, manufacturing and healthcare. It is not an easy mission of providing security in sensor networks. Compared to standard desktops, there are many stringent restrictions on sensor nodes such as energy, storage and limited wireless network bandwidth. In spite of the challenges mentioned above, for several sensor networks applications, security is important and even critical, such as homeland security applications and military (Li *et al*., 2014).

Energy in WSNs represents an important factor in designing, controlling and operating the sensor networks. Minimizing the consumed energy in WSNs application is a crucial issue for the network effectiveness in terms of lifetime. Many algorithms and protocols has been proposed and implemented must take into account decreasing the energy consumption (Baadache and Adouane, 2015).

There are mainly two types of ciphering techniques to provide data security and confidentiality. These cryptographic techniques are symmetric key technique and asymmetric key technique. Both the techniques have their own benefits and disadvantages (Prakash and Rajput, 2018; Patt-Shamir, 2007).

The main benefit of the symmetric key algorithms in WSN is that it is fast and has less complexity (low power consumption). However, it is less secure than asymmetric key algorithm because it uses a single secure key for data cryptography (Prakash and Rajput, 2018; Patt-Shamir, 2007). On the other hand, asymmetric algorithms are slow and complex but they provide a high level of security (Ahmed *et al*., 2011).

Prakash and Rajput (2018) attempted to reduce the energy utilization by creating distinctive security protocols and chipset. An AES co-processor intended to reduce the power utilization and to increase the framework execution whose MAC layer executes AES algorithm. ECC used for secure and effective ID protocol for multi-hop WSN which combines symmetric and asymmetric cryptography to manage keys (Prakash and Rajput, 2018).

Landstra *et al*. (2007) suggest a protocol for the management of energy efficient hybrid keys. In which a wireless sensor network heterogeneous security requirements has considered for providing different security levels with least correspondence overhead (Landstra *et al*., 2007).

Bafandehkar *et al*. (2013) presented two public key algorithms which are ECC and RSA cryptography. They discovered that ECC has a great benefit against RSA

Corresponding Author: Susan Mohammed, Department of Computer Science, College of Science for Women, University of Babylon, Hillah, Iraq

because it reduces the amount of data transferred. Thus, it reduces the processing time complexity (Bafandehkar *et al*., 2013).

The main goal of this study is providing a robust protocol for secure data communication in WSNs. As the sensor nodes have limited energy computational and limited memory resources, the proposed protocol should be able to fully utilize them and it should be able to provide proper security and data confidentiality.

Wireless sensor network and security protocols: Wireless networks are attacked much more than wired networks because it worked in an unreliable environment. The sensor nodes are low power devices with limited resources in terms of connectivity and computational capability. Digital asymmetric encryptions are impractical because they require high power. The use of security mechanisms makes the additional burden on the network such as increasing energy consumption and latency which shortens the life of the network (Al-Alak *et al*., 2011).

To protect data a hybrid security protocol was proposed which combine two algorithms, one is a symmetric algorithm which is AES and the other is an asymmetric one, it is the ECC algorithm.

The ECC algorithm was chosen as a public-key algorithm because it is the most secure public-key algorithm. In addition, ECC uses a smaller key size compared with other algorithms which led to reduce the amount of memory, processing time, bandwidth and power consumption which it is required (Al-Alak *et al*., 2011).

Elliptic curve group might provide the same security level afforded by an RSA-based system with a large modulus and correspondingly larger key. The equation of ECC algorithm is:

$$y^2 = x^3+ax+b$$

The AES algorithm was chosen as a symmetric key algorithm because it is the highest secure symmetric algorithm and it is easy to implement which making it successful and acceptable (Al-Alak *et al*., 2011).

The block and key sizes must be determined before using the protocol to encrypt the message. The allowed sizes of blocks in the AES algorithm are 128, 168, 192, 224 and 256 bits. The sizes of keys allowed in the AES algorithm are 128, 192 and 256 bits, respectively (Prakash and Rajput, 2018). The standard AES algorithm (which used in this study) uses the 128 size for the block and the key.

MATERIALS AND METHODS

The proposed research: The goal of the proposed research is building a more secure WSN security system with regard energy consumption. The proposed key generation model satisfied this goal. The proposed protocol dividing the secret message into n blocks of 128 bits size. Each block then encrypted with a different generated key. Figure 1 shows a general view of the proposed encryption protocol. Three proposed methods are used to generate keys.

Two key seeds generation: In the first proposed method, two 128 bit initial random key seeds r0, k0 are used to generate two sets of keys: r0, r1, ..., rn, k0; k1, ..., kn where the sets are generated according to the following Eq. 1 and 2:

$$ri = ECCenc\ (TCk1, ri\text{-}1)\ \ 0<i \le n \tag{1}$$

$$ki = ECCenc\ (TCk2, ki\text{-}1)0<i \le n \tag{2}$$

Where:
ECCenc = Elliptic curve function (ECC algorithm)
TCk1, TCk2= Trust center keys

The trust center must randomly generate same r0 and k0 in the sender and receiver sides. Then, the hash function f, Eq. 3 is used to obtain the final keys Keyi which are used in the encryption process:

$$Keyi = f\ (ki, rn\text{-}i+1)\ 1 \le i \le n \tag{3}$$

Three key seeds generation: For more security, three 128-bit initial random key seeds r0, k0, m0 are used to generate three sets of keys: r0, r1, ..., rn; k0, k1, ..., kn; m0, m1, ..., mn. The first and the second sets are generated according to Eq. 1 and 2, respectively. The third set will be generated according to the following Eq. 4:

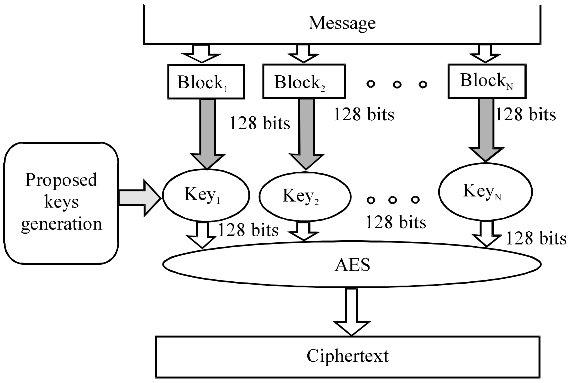

Fig. 1: A general diagram of the proposed protocol

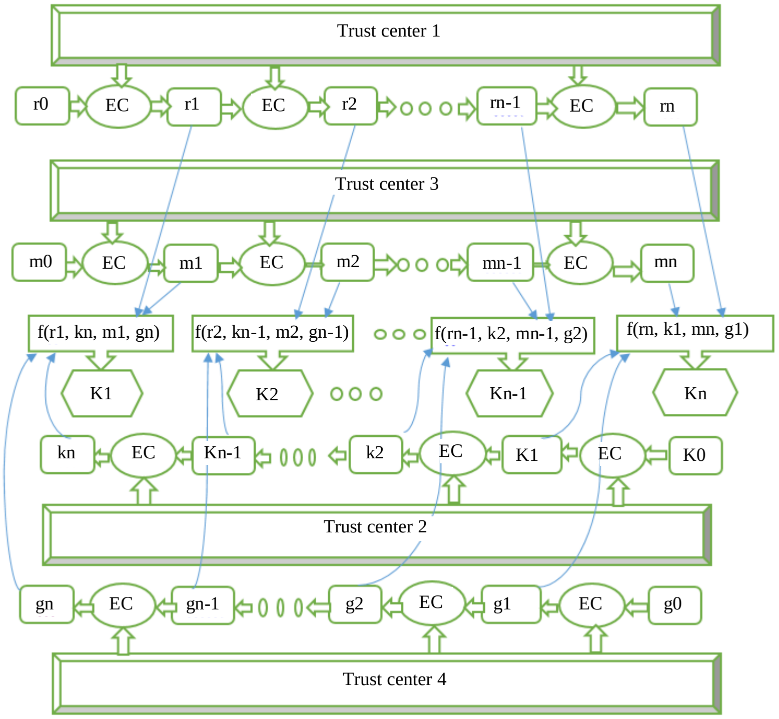

Fig. 2: The diagram of the proposed key generation model

$$mi = ECCenc\ (TCk3,\ mi\text{-}1)\ \ 0<i \leq n \tag{4}$$

where TCk3 is trust center key. Then, the hash function f is used to obtain the final keys Keyi as in the following Eq. 5:

$$Keyi = f\ (ki,\ rn\text{-}i\text{+}1,\ mi)\ 1 \leq i \leq n \tag{5}$$

Four key seeds generation: For more security, four 128 bit initial random key seeds r0, k0, m0, g0 are used to generate four sets of keys: r0, r1, ..., rn; k0, k1, ..., kn; m0, m1, ..., mn; g0, g1, ..., gn. The first, second and third sets are generated according to Eq. 1-3, respectively and the fourth set will be generated according to the following Eq. 6:

$$gi = ECCenc\ (TCk4,\ gi\text{-}1)\ 0<i \leq n \tag{6}$$

where, TCk4 is trust center key. Then, the hash function f is used to obtain the final keys Keyi as in the following Eq. 7:

$$Keyi = (ki,\ rn\text{-}i\text{+}1,\ mi,\ gn\text{-}i\text{+}1)\ 1 \leq i \leq n \tag{7}$$

The keys generated in each of the three previous cases from the ECC algorithm used by AES algorithm for encrypting and decrypting data. Every block of plaintext (Bi) was encrypted by AES algorithm to produce a block of Ciphertext (Ci) as shown in the following Eq. 8:

$$Ci = AESki\ (Bi) \tag{8}$$

In order to decrypt any block of the Ciphertext Ci which decrypted by AES algorithm must use the same key Ki that used for encrypting the same block of message. The equation of decrypting is:

$$Bi = AESki\ (Ci) \tag{9}$$

The diagram of proposed protocol is demonstrated in Fig. 2.

Encryption process: In proposed protocol, AES algorithm along with proposed generator of keys is used to encrypt secret message.

Algorithm 1; A proposed encryption protocol:

```
Input: k0, r0, m0, g0; key seeds
Message; secret message
PK1, PK2, PK3, PK4; Generated public keys
Output: C1, C2, C3; The ciphertext
Key1, Key2, Key3; Proposed generated keys
Begin
Blocksize = 128
For i = 1 to n
        si = ECC (si-1, PK1)
         ri= ECC (ri-1, PK2)
         mi= ECC (mi-1, PK3)
gi = ECC (gi-1, PK4)
K1i = f (si, rn-i+1)
K2i = f (si, rn-i+1, mi)
```

```
K3i = f (si, m-i+1, mi, gn-i+1)
No-of-blocks = Message/blocksize
No-of-Blocks-in-Group = No-of-blocks/10
Count = 1
Key1 = k1icount
Key2 = k2icount
Key3 = k3icount
For j = 1 to no-of-blocks
C1j = AES (Bj, Key1)
C2j = AES (Bj, Key2)
C3j = AES (Bj, Key3)
If (j mod (No-of-Blocks-in-Group) = 0) then
            Key1 = Kcount+1
            Key2 = Kcount+1
            Key3 = Kcount+1
End
```

Randomness tests: One measurements of security system is randomness test of data. The encrypted data which has more security strength, it has more randomness. In order to measure the strength and effectiveness of a proposed encryption system, it must compute the randomness. It has been compared the randomness of plaintext and three cipher text of proposed methods to illustrate the security of ECC-AES proposal. The test of randomness has executed by a Diehard test (Ramesh *et al*., 2012).

In this study, a 17 MB data has been tested to evaluate the randomness of the AES-ECC proposed algorithm. The 128 bit block and key size have been used in the executed experiments.

The tests of Diehard: Diehard tests are a battery of statistical tests for measuring the quality of a random number generator (Rameshet *al*., 2012). The Diehard program executes 18 different tests. These tests are:

- Birthday spacing
- Overlapping permutations
- Ranks of matrices
- Monkey tests
- Count the ones
- Parking lot test
- Minimum distance test
- Random spheres test
- The squeeze test
- Overlapping sums test
- Runs test
- The craps test

The output of the Diehard is groups of p-values which belong to the interval (0, 1). The interval in the method is dividing into three areas: (safe, doubt, failure). The limit of safe area is 0.25 >p-value > = 0.75. The limit of doubt area is 0.1>p-value> = 0.25 and 0.75>p-value> = 0.9. The limit of failure area is 0>p-value > = 0.1 and 0.9>p-value> = 1.

The values of p can be lying in one of the three regions (failure, doubt and safe region). Whenever the failure region contains a large number of points, the key seed test sample will move away from the high randomness. On the other hand, the points which closer to the safe region, the randomness will be higher (Al-Alak *et al*., 2011; Ramesh *et al*., 2012).

RESULTS AND DISCUSSION

A file with size 17 MB was used as a benchmark to verify the randomness of proposed research. The Diehard tests p-values of the plaintext are shown in the Table 1. The randomness of plaintext file is shown in Fig. 3.

The second test is performed on the ciphertext encrypted by pure AES algorithm. The Diehard tests p-values are shown in the Table 2. The results of the test are displayed in Fig. 4.

Table 1: p-values of a Diehard test of plaintext

Test names	Fail	Doubt	Safe
Birthday test	2	1	6
Overlapping 5-pemutation	2	0	0
Test of binary rank (31*31)	0	0	1
Test of binary rank (32*32)	0	0	1
Test of binary rank (6*8)	21	3	1
Test of bit stream	20	0	0
OPSO	23	0	0
OQSO	28	0	0
DNA	31	0	0
Count the 1's	2	0	0
Count the 1st test for specific bytes	25	0	0
Parking lot	8	2	0
Minimum distance	1	0	0
3D spheres	4	3	13
Squeeze test	1	0	0
Overlapping sum	2	5	3
Runs test	2	1	1
Craps test	1	0	1

Table 2: p-values of a Diehard test of AES Ciphering

Test names	Fail	Doubt	Safe
Birthday test	3	5	1
Overlapping 5-pemutation	0	1	1
Test of binary rank (31*31)	0	0	1
Test of binary rank (32*32)	0	0	1
Test of binary rank (6*8)	4	7	14
Test of bit stream	3	6	11
OPSO	7	9	7
OQSO	6	9	13
DNA	4	5	22
Count the 1's	2	0	0
Count the 1st test for specific bytes	9	6	10
Parking lot	1	3	6
Minimum distance	0	1	0
3D spheres	4	4	12
Squeeze test	0	0	1
Overlapping sum	1	4	5
Runs test	0	1	3
Craps test	0	1	1

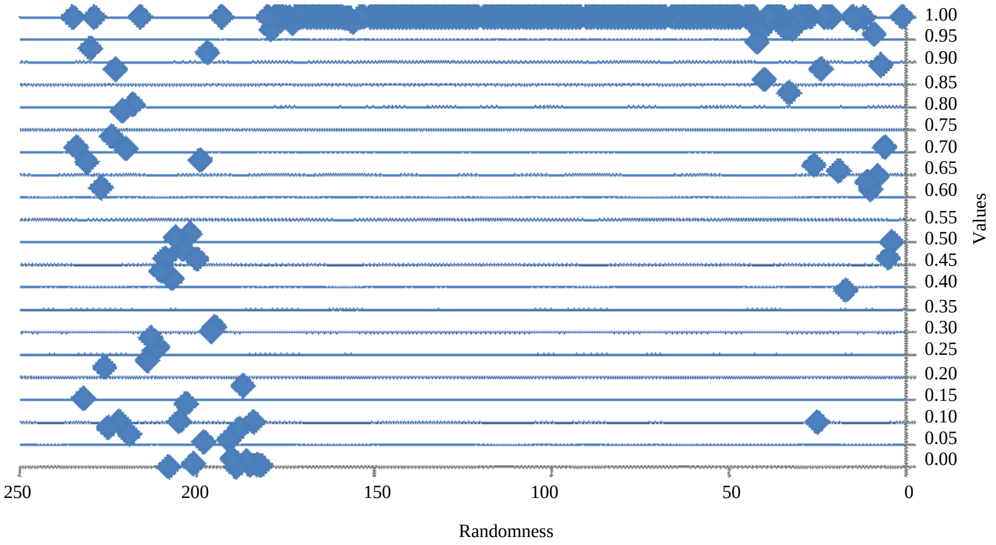

Fig. 3: Randomness values of plaintext

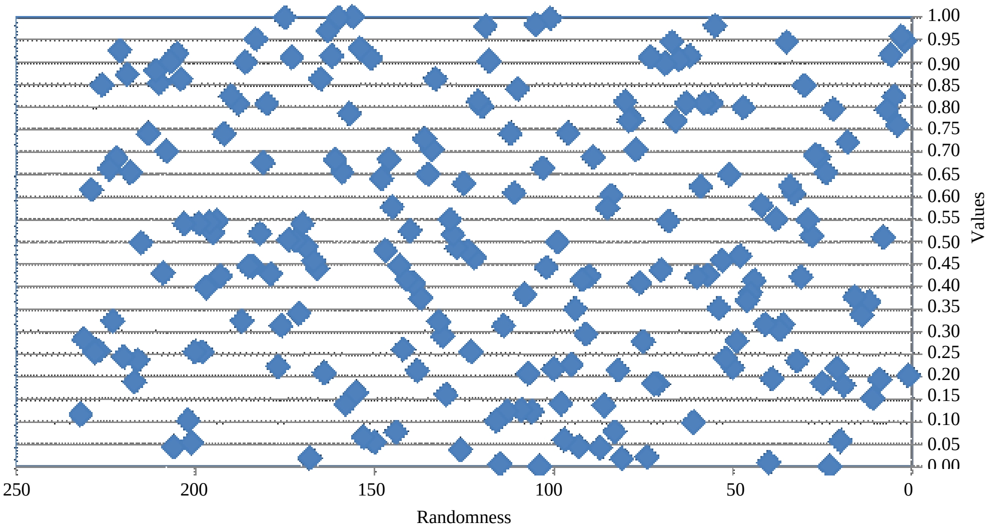

Fig. 4: Randomness values of ciphertext encrypted using standard AES algorithm

The third test is performed on the ciphertext encrypted by the first method of proposed protocol (two key seeds). The Diehard tests p-values are shown in the Table 3. The randomness results of this test are displayed in Fig. 5.

The fourth randomness test is performed on the ciphertext encrypted by the second method of proposed protocol which uses three key seeds. The results of the Diehard tests p-values are shown in the Table 4. The randomness results are displayed in Fig. 6.

The fifth test is performed on a ciphertext encrypted which produce by the third method of the proposed protocol which uses four key seeds. The results of the Diehard tests p-values are displayed in the Table 5. The randomness results are shown in Fig. 7.

From previous tests, it is possible to conclude that when using a standard AES algorithm to encrypt the plaintext, the p-values in the fail area are decreased (from 173 in the plaintext file into 44 in the ciphertext file) while the p-values in a safe area are increased (from 27 into 109). Increasing p-values in safe area means increasing randomness thus increasing security. Where increasing p-values in fail area means decreasing randomness thus decreasing security.

When using proposal two key seeds with AES algorithm for encrypting the same plaintext file, the p-values in the fail area are decreased (from 173 in the plaintext file into 34 in the ciphertext file) while the p-values in the safe area are increased (from 27-112).

Table 3: p-values of a Diehard test of proposed protocol with two key seeds ciphering

Test names	Fail	Doubt	Safe
Birthday test	0	5	4
Overlapping 5-pemutation	0	1	1
Test of binary rank (31*31)	0	1	0
Test of binary rank (32*32)	0	0	1
Test of binary rank (6*8)	3	5	17
Test of bit stream	4	8	8
OPSO	3	8	12
OQSO	2	9	17
DNA	6	9	16
Count the 1's	0	1	1
Count the 1st test for specific bytes	5	11	9
Parking lot	1	3	6
Minimum distance	0	0	1
3D spheres	6	6	8
Squeeze test	0	0	1
Overlapping sum	1	2	7
Runs test	2	0	2
Craps test	1	0	1

Table 4: p-values of a Diehard test of proposed protocol with three key seeds ciphering

Test names	Fail	Doubt	Safe
Birthday test	2	1	6
Overlapping 5-pemutation	1	0	1
Test of binary rank (31*31)	0	0	1
Test of binary rank (32*32)	0	0	1
Test of binary rank (6*8)	5	10	10
Test of bit stream	2	8	10
OPSO	3	6	14
OQSO	5	6	17
DNA	6	6	19
Count the 1's	0	0	2
Count the 1st test for specific bytes	6	8	11
Parking lot	1	2	7
Minimum distance	1	0	0
3D spheres	3	10	7
Squeeze test	0	0	1
Overlapping sum	1	3	6
Runs test	2	1	1
Craps test	0	1	1

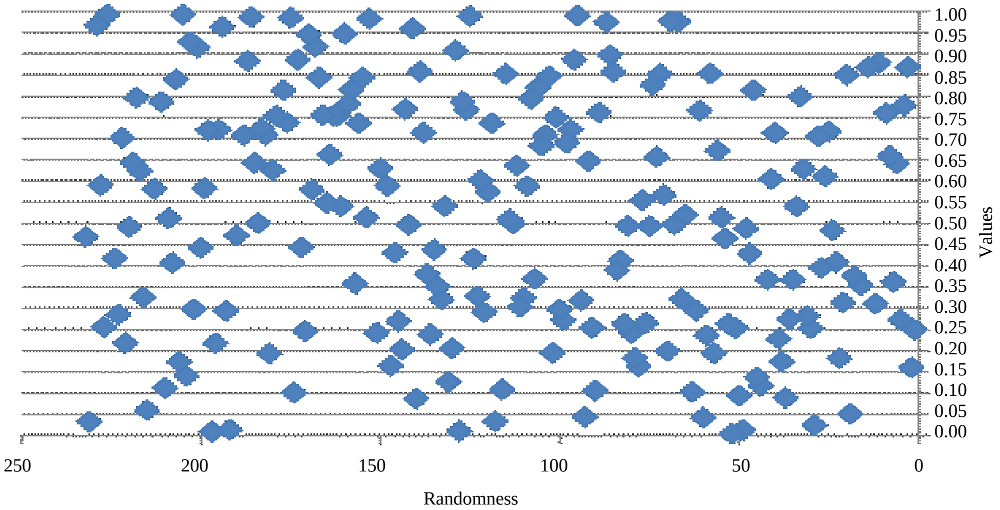

Fig. 5: Randomness values of ciphertext encrypted using the proposed protocol with two key seeds ciphering

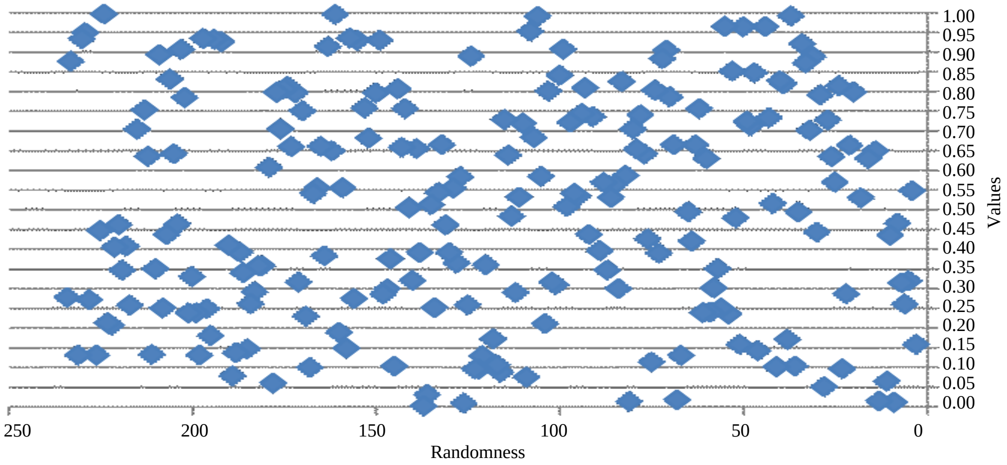

Fig. 6: Randomness values of ciphertext encrypted using proposed protocol with three key seeds ciphering

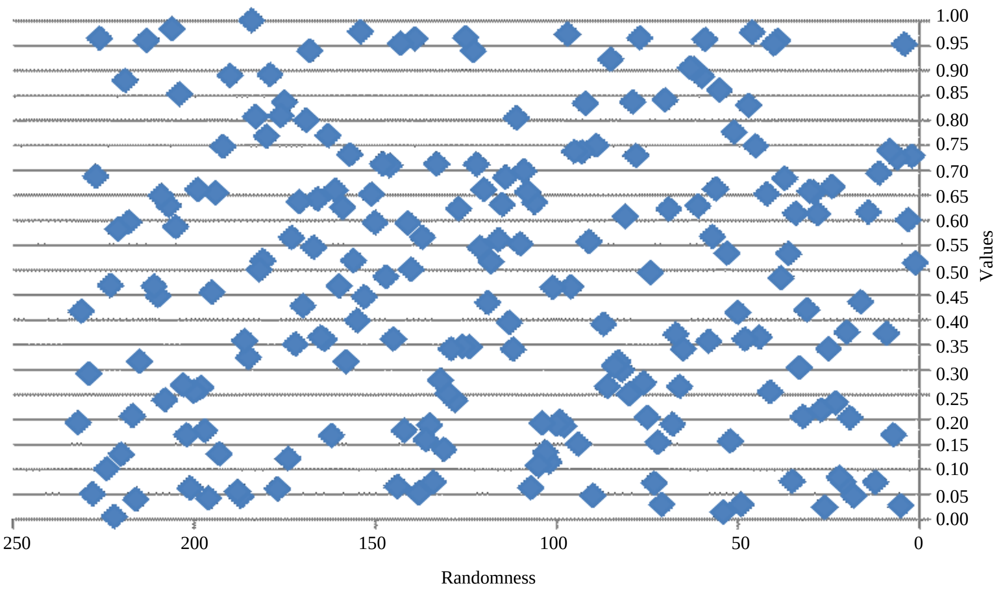

Fig. 7: Randomness values of ciphertext encrypted using proposed protocol with four key seeds ciphering

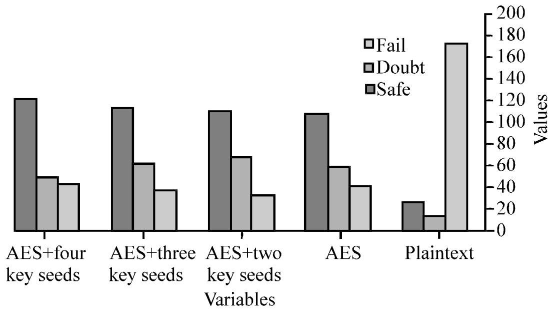

Fig. 8: p-values comparison of randomness tests

Table 5: p-values of a Diehard test of proposed protocol with four key seeds ciphering

Test names	Fail	Doubt	Safe
Birthday test	2	1	6
Overlapping 5-pemutation	1	0	1
Test of binary rank (31*31)	0	0	1
Test of binary rank (32*32)	0	0	1
Test of binary rank (6*8)	7	4	14
Test of bit stream	6	5	9
OPSO	4	5	14
OQSO	3	10	15
DNA	7	5	19
Count the 1's	0	0	2
Count the 1st test for specific bytes	3	6	16
Parking lot	3	3	4
Minimum distance	0	1	0
3D spheres	3	5	12
Squeeze test	1	0	0
Overlapping sum	2	4	4
Runs test	2	0	2
Craps test	0	1	1

When using proposal three key seeds with AES algorithm for encrypting the same plaintext file, the p-values in the fail area are decreased (from 173 in the plaintext file into 38 in the ciphertext file) while the p-values in the safe area are increased (from 27-114).

When using proposal four key seeds with AES algorithm for encrypting the same plaintext file, the p-values in the safe area are increased (from 27 in the plaintext file into 121 in the ciphertext file). The results of comparison among files before and after applying the proposal work are displayed in Fig. 8.

CONCLUSION

In order to increase the security in WSN, this study suggests building a hybrid security protocol of two low power consumption encryption algorithms which are ECC and AES algorithms.

Three methods of the proposed protocol are implemented on secret message. The proposed protocol three cases: case one, using two key seeds, case two, using three key seeds and case three, using four key seeds. One can conclude that when increasing the number of key seeds that using in generating keys; the randomness will increased which means increasing security.

The key generation was proposed by combining multi ECC algorithms to support AES encryption algorithm. By calculating the randomness of plaintext file in several situations (without encryption, after encryption using the standard AES algorithm and after applying the three methods of the proposed protocol), the randomization increased at each proposal. Thus, security increased. There is a trade off between increasing security and energy consuming. A balance must be made between the complexity of the encryption protocol and the amount of energy consumed.

REFERENCES

Ahmed, S., K. Samsudin, A.R. Ramli and F.Z. Rokhani, 2011. Effective implementation of AES-XTS on FPGA. Proceedings of the International Conferencs on TENCON 2011-2011 IEEE Region 10, November 21-24, 2011, IEEE, Bali, Indonesia, ISBN:978-1-4577-0256-3, pp: 184-186.

Al-Alak, S., Z. Ahmed, A. Abdullah and S. Subramiam, 2011. AES and ECC Mixed for ZigBee wireless sensor security. World Acad. Sci. Eng. Technol., 5: 535-539.

Baadache, A. and R. Adouane, 2015. Minimizing the energy consumption in wireless sensor networks. Adhoc Sens. Wireless Networks, 27: 223-237.

Bafandehkar, M., S.M. Yasin, R. Mahmod and Z.M. Hanapi, 2013. Comparison of ECC and RSA algorithm in resource constrained devices. Proceedings of the 2013 International Conference on IT Convergence and Security (ICITCS), December 16-18, 2013, IEEE, Macao, China, pp: 1-3.

Landstra, T., M. Zawodniok and S. Jagannathan, 2007. Energy-efficient hybrid key management protocol for wireless sensor networks. Proceedings of the 32nd IEEE Conference on Local Computer Networks (LCN 2007), October 15-18, 2007, IEEE, Dublin, Ireland, ISBN:0-7695-3000-1, pp: 1009-1016.

Li, W., P. Yi, Y. Wu, L. Pan and J. Li, 2014. A new intrusion detection system based on KNN classification algorithm in wireless sensor network. J. Elect. Comput. Eng., 2014: 1-8.

Patt-Shamir, B., 2007. A note on efficient aggregate queries in sensor networks. Theor. Comput. Sci., 370: 254-264.

Prakash, S. and A. Rajput, 2018. Hybrid Cryptography for Secure Data Communication in Wireless Sensor Networks. In: Ambient Communications and Computer Systems, Proceedings of the International Conferences on Ambient Communications and Computer Systems, Perez, G.M., S. Tiwari, M.C. Trivedi and K.K. Mishra (Eds.). Springer, Singapore, Asia, ISBN:978-981-10-7385-4, pp: 589-599.

Ramesh, P.S., F.E.M. Priya and B. Santhi, 2012. Review on security protocols in wireless sensor networks. J. Theoret. Applied Inform. Technol., 38: 79-82.