Research Journal of Information Technology

ISSN 1815-7432

Academic Journals Inc.

www.academicjournals.com

Research Journal of Information Technology 5 (1): 24-34, 2013
ISSN 1815-7432 / DOI: 10.3923/rjit.2013.24.34

Randomness Improvement of AES using MKP

[1]Saif Al-Alak, [2]Zuriati Zukarnain, [2]Azizol Abdullah and [2]Shamala Subramiam
[1]Department of Computer Science, College of Science for Women, Babylon University, Babylon, Iraq
[2]Department of Communication Technology and Networks, Faculty of Computer Science and Information Technology, University Putra Malaysia, 43400 Serdang, Selangor, Malaysia

Corresponding Author: Saif Al-Alak, B2-12-17 Kantan Court Sec4 Bukit Serdang Seri-Kembangan 43300 Selangor, Malaysia Tel: 0060129361538

ABSTRACT

Randomness is a high impact property of the ciphertext that evaluates the strength of a cryptography system. Advanced Encryption Standard (AES) is a symmetric block cipher algorithm that passed the randomness test. AES algorithm is widely used in a computer communication system for securing data transfer. In previous work we proposed a Multiple-key Protocol (MKP) for AES algorithm. In this paper we tested the randomness of MKP-AES algorithm by using the diehard statistical tests software. The randomness of AES is improved when MKP is used.

Key words: Advanced encryption standard, diehard, multiple-key protocol, randomness, statistical test

INTRODUCTION

Many computer applications are utilizing a symmetric key algorithm for securing the transferred data where it is preferred to be adapted for multimedia cryptography (Salem *et al*., 2011; Abomhara *et al*., 2010). Some platforms are utilizing the Advanced Encryption Standard (AES) (NIST, 2001) for message encryption like IEEE 802.15.4 (LAN/MAN, 2006) which is a Low Rate Wireless Personal Area Network (LR-WPAN). Many specifications and standards are built over IEEE 802.15.4 platform for example: ZigBee (http://www.zigbee.org), IPv6 (Montenegro *et al*., 2007) and Wireless HART (STG, 2007). Also BACnet runs over IEEE 802.15.4 by 6LoWPAN (Lv *et al*., 2010). In addition to above applications the AES is utilized for mobile Ad hoc network environment encryption (Sharma *et al*., 2006).

The secret key affects the strength of AES algorithm. Muda *et al*. (2010) increased the security of key schedule. Multiple-Key Protocol (MKP) (Al-Alak *et al*., 2011) is utilizing multiple keys for AES algorithm to improve the algorithm. All trusted block cipher crypto systems should pass the randomness test before declaration as a good system. The randomness test can be used to measure the cryptographic strength of block cipher system. The AES algorithm is already passed the randomness test. The research question is how the MKP-AES can impact the randomness of the AES algorithm?

The paper performs an experimental evaluation for the MKP-AES algorithm by measuring the algorithm randomness which computes the strength of the cryptosystem. It compares the information randomness as a plaintext and ciphertext to show the randomness of MKP-AES. Also it compares the randomness of ciphertext received from AES (with random and fixed single secret key) and MKP-AES to show the randomness improvement of MKP-AES. The randomness test is performed by a diehard (Random, 2006) which is powerful statistical test that performs a list of tests

to measure the randomness. This study has been conducted to check the randomness of MKP-AES algorithms. The key length in our experiments was 128 bit.

ADVANCED ENCRYPTION STANDARD

In 1996, the National Institute of Standards and Technology (NIST) issued in a Federal Information Processing Standards Publications (FIPS PUBS) 197 an AES to be used instead of Data Encryption Standard (DES) (Rabah, 2005a) for symmetric encryption. AES is simply and flexibly handled (Zaidan *et al*., 2010) algorithm and used a symmetric secret key for encrypting data. In the next subsections we explained briefly how AES work.

Encrypting: AES algorithm performs a sequence of transformations on blocks of input data (plaintext) to produce a ciphertext. Encryption Transformations are following: AddRoundKey, SubBytes, ShiftRows and MixColumns. The operation that converts plaintext to ciphertext is called an Encryption. Figure 1 shows the sequence of transformations that are applied on a block of plaintext to convert to a ciphertext block. Number of round (Nr) is depending on the key size.

SubBytes transformation is a non-linear byte substitution that operates separately on each byte of the input block using a substitution table (S-box). ShiftRows transformation produces cyclically shifting to the bytes in the last three rows of the input block over different numbers of bytes (offsets). The first row is not shifted. ShiftRows transformation interchanges bytes in the top with lowest bytes. MixColumns transformation operates on the input column-by-column; it treats each column as a four-term polynomial. AddRoundKey transformation adds a round key to the input by a bitwise XOR operation.

The size of plaintext blocks and ciphertext blocks are 128-bit (16 octets). The key length can be 128, 192 and 256 bit and number of round key is 10, 12 and 14, respectively. Each plaintext block is organized as a two dimension array of byte, which has 4 rows and 4 columns. Also, the key is treated in the same way except it may have 4, 6 or 8 columns instead based on its length.

Decrypting: Decryption is an operation that converts the received blocks from encryption operation (ciphertext) into their original shape (plaintext). The transformations that needed for decryption are inversed of encryption transformations. They are AddRoundKey, InvSubBytes, InvShiftRows and InvMixColumns. The transformations are applying in the same sequence as encryption. The decryption operation has more time overhead than encryption operation (Xiao *et al*., 2006).

```
Cipher (input, output, key)
Begin
    AddRoundKey (input, key)
    For round = 1 to Nr-1
        SubBytes (input)
        ShiftRows (input)
        MixColumns (input)
        AddRoundKey (input, key)
    End for
    SubBytes (input)
    ShiftRows (input)
    AddRoundKey (input, key)
    output = input
End
```

Fig. 1: AES single block encryption

Key expansion: In each round the sub-key is produced from the original key via a sequence of transformations. The key transformations are SubWord, RotWord and Rcon. SubWord applies the S-box on four bytes input. RotWord performs a cyclic permutation. Rcon is a round constant word array. The sub-keys are different from each other.

MULTIPLE-KEY PROTOCOL ADVANCED ENCRYPTION STANDARD

MKP employs multiple keys for ciphering the plaintext. The MKP protocol can be utilized to improve the strength of the AES algorithm in the encryption and decryption operations. MKP has multiple secret keys.

Encryption: The message is dividing to n blocks as shown in Eq. 1 and the blocks are classifying to groups. The number of groups is determined by the level of security. The blocks belong to group i are utilizing the secret key K_i for data encryption as shown in Eq. 2. The encryption algorithm (E) is AES and the block size is 16-octet:

$$M = \sum_{i=1}^{i=n} B_i \tag{1}$$

$$C_i = E_{ki}(B_i) \tag{2}$$

Decryption: In the decryption operation the protocol utilizes the secret keys to deciphering the blocks of ciphertext as shown in Eq. 3. Because the AES algorithm is symmetric the MKP used the same secret keys for blocks ciphering and deciphering:

$$B_i = E_{ki}(C_i) \tag{3}$$

Keys generation: MKP adapted Elliptic Curve Cryptography (ECC) (Rabah, 2005b) to generate the secret keys because it require less computational power, memory and communication bandwidth in compare with other traditional public key crypto-algorithms (Rabah, 2005c).

The MKP assumes that there are two secure initial keys r_0 and k_0 that are established by a trust center for the communication. The list of secret keys ($K_1 \ldots K_n$) is generated in sender and receiver node. The two lists of parameters $r_1, r_2, \ldots r_n$ and $k_1, k_2, \ldots k_n$ are computed by ECC as shown in Eq. 4 and 5:

$$r_i = ECC_{enc}(TC_{PK}, r_i\text{-}1) \tag{4}$$

$$k_i = ECC_{enc}(TC_{PK}, k_{i-1}) \tag{5}$$

The ECC_{enc} codes the input using the trust center public key (TC_{PK}). The trust center should established r_0 and k_0 to sender and receiver by node's public key securely. The secret key K_i is computed by a hash function (f) as shown in Eq. 6:

$$K_i = f(k_i, r_{n-i+1}),\ 1 \le i \le n \tag{6}$$

RANDOMNESS

General definition: It is a measure of independence for any sequence of integer numbers. All confident generators must have a good randomness degree. Many statistical tools are used to check the randomness of generated numbers. For example diehard is used to evaluate the randomness of RPG100 generator (FDK, 2003).

Diehard: Diehard battery of randomness tests is a set of powerful statistical tests used to test the randomness of numbers. The diehard program includes 18 different independent tests. The tests are following:

- **Birthday spacing test:** In this test, from n days of a year, m birthday are chosen. The more frequent value between birthdays is j, which is considered as asymptotically Poisson distributed
- **Overlapping 5-permutation (OPERM5) test:** This test counts the number of the occurrence of five sequence numbers cumulatively. Each five sequence number produces 120 possible orders, which should have equal probability
- **Binary rank test (for 31×31 matrices):** This test computes the rank value for 31×31 binary matrix over {0, 1}. Each cell in the matrix is the 31 left most significant bits of random integer of test sequence
- **Binary rank test (for 32×32 matrices):** This rank test is same as previous test except the matrix size is for 32×32
- **Binary rank test (for 6×8 matrices):** The size of matrix in this test is 6×8, which cells are 6 bytes chosen from 6 random integers of test sequence
- **Monkey (bit-stream) tests:** This test considers the test sequence is consisting of 20-bit words. Then it counts j (the number of missing 20-bit words), where there are 220 possible distinct 20-letter words. The value of j should be normally distributed. The test is repeated 20 times
- **Overlapping-pairs-sparse-occupancy (OPSO) test:** This test considers a word that is consisting of two of 10-bit (each letter is 10 bits length) letters, where the total number of available letters is 1024. The test counts the value of missing words which should be normally distributed
- **Overlapping-quadruples-sparse-occupancy (OQSO) test:** This test is same as previous test except that the word is consisted of four of 5-bit letters, where the total number of available letters is 32
- **DNA test:** It is same as two previous tests except that the word is consisted of ten of 2-bit letters, where the total number of available letters is 4 (C, G, A and T)
- **Count-the-1's test on a stream of bytes:** This test suppose that the test sequence is consisted of five letters words, each letter is either A, B, C, D or E. Each byte of the sequence is mapped to one of the five letters. The frequency of 1's in the byte determines the mapped letter, where 0, 1 or 2 yield A, 3 yields B, 4 yields C, 5 yields D and 6, 7 or 8 yields E. The test counts the overlapping words, where the disappeared words should be in known distribution
- **Count-the-1's test for specific bytes:** This test is applying to previous test but on specified bytes
- **Parking lot test:** In this test circular units are randomly placed in a square, where the square is 100×100 and circle radius is 1. When the circle is overlapped the other then tries again. After 12000 tries, k circular units are succeeding to be place in the square. The value of k should be close to normal distributed

- **Minimum distance test:** This test computes the minimum distance between $(n^2-n)/2$ pairs of points, where n = 8000 random points in a square of side = 10000. This operation is repeated 100 times. The square of minimum distances should be exponentially distributed with a known mean
- **3D spheres test:** In this test 4000 spheres are distributed in a cube (with edge = 1000), where the center points of the spheres are chosen randomly and the spheres are large enough to be close to the next one. The volume of a smallest sphere is exponentially distributed with a known mean. This test is repeated 20 times
- **Squeeze test:** The random integers of sequence test are floating to get uniforms on [0, 1). The floating integers are multiplying with k (k = 231) j times, to reduce k to 1. The chi-square test is computed using frequency of j when its magnitude is belonging to known range. The j value is computed 100,000 times
- **Overlapping sum test:** In this test, the sequence test random integers are floated to uniform [0, 1). The overlapping sums (S's) from sequence of 100 consecutive random floats are computed. The sums should be normally distributed with characteristic mean and sigma
- **Runs test:** In this test, 10000 random integers from sequence test are floating to be in [0, 1) uniform, then the runs up and runs down to them are counted. The counts should follow a known distribution
- **Craps test:** This test counts the number of wins and the number of throws for craps game. The game is repeated 200000, where the counts should follow known distribution

Diehard test is calculating the p-values. The magnitude of p-value is in an interval [0, 1) for truly random numbers. The p-value is computed by $p = F(X)$, where F is distribution of random X.

AES and randomness: Rijndael, Serpent and TwoFish are the accepted block cipher algorithms by NIST as a final list to select AES. They are passed in randomness test, because they are able to generate random ciphertext (Soto and Bassham, 2000). AES had been check for randomness with 128-bit key size by NIST statistical test. The evaluation of a ciphertext by using statistical test showed that there is no means by which to computationally distinguish ciphertext from true source (Juan, 2011).

METHODOLOGY

The pretest-posttest design is adopted in this project. The project tested a binary file to evaluate the randomness of MKP-AES algorithm, where the size of the file is 255 MB.

The project tested another binary file with 11 MB size, to exam the randomness improvement of MKP-AES algorithm over standard AES.

The MKP-AES algorithm is executed with 10 secret keys. The MKP-AES algorithm generates secret keys by ECC. The randomness's of the original plaintext, ciphertexts of original AES algorithm with fixed key and random key and ciphertext of MKP-AES algorithm, are compared to show algorithm effectiveness.

Diehard software (version: DOS, Jan 7, 1997) is used to compute the randomness of algorithms in the experiments of this project. The ciphertexts yield from encrypting a file with block cipher cryptosystem, had tested by diehard.

DATA REPRESENTATION

The output of the diehard is groups of p-values which belongs to interval [0, 1). The project used a method mentioned by Alani (2010) for data representation. The method is dividing the interval to three areas: failure, doubt and safe. The limit of failure area is $0<\text{p-value}\leq 0.1$ and $0.9<\text{p-value}\leq 1$. The limit of doubt area is $0.1<\text{p-value}\leq 0.25$ and $0.75<\text{p-value}\leq 0.9$. The limit of safe area is $0.25<\text{p-value}\leq 0.75$.

The p-values points may be in one of three areas (failure area, doubt area and safe area). The sequence test sample is deviating from randomness, when more points lie in the failure area. In the other side, the sample is closing to randomness, when more points lie in the safe area.

EXPERIMENTAL RESULTS

First experiment: In the first experiment one binary file with 255 MB size is tested twice. In the first test the plaintext is evaluated in terms of randomness and in the second test the ciphertext is evaluated. The result of first experiment is shown in Fig. 2. The p-values belong to diehard tests were as following: In birthday test the p-value was close to zero. In overlapping 5-permutation test one of the p-value was close to 0.7 (in the safe area) and other one was very close to one. In binary rank test (for 31×31 matrices) the p-value was very close to one. In binary rank test (for 32×32 matrices) the p-value was very close to one. In binary rank test (for 6×8 matrices) the p-value was one. In monkey tests there were thirteen p-values ones and the other seven p-values were very close to one. In OPSO test there were sixteen p-values ones and the other seven p-values were very close to one. In OQSO test three p-values were ones and other twenty five values were very close to one. In DNA test seventeen p-values were very close to one, seven p-values were close to 0.8 (in doubt area) and seven p-values were close to 0.6 (in safe area). In count-the-1's test on a stream of bytes the two p-values were ones. In count-the-1's test for specific bytes sixteen p-values were ones and other nine p-values were very close to one. In parking lot test the p-value was very close to one. In minimum distance test the p-value was one. In 3D spheres test the p-value was very close to one. In squeeze test the p-value was very close to one. In overlapping sum test the p-value of was very close to 0.3 (in safe area). In runs test one of the p-values was close to zero and other three p-values were between 0.3 and 0.7 (in safe area). In craps test one of the p-value was very close to one and the other p-value was close to 0.8 (in double area).

The second test is performed on the ciphertext received from MKP-AES algorithm and the results are shown in Fig. 3. The diehard tests results in terms of p-values improvement were as

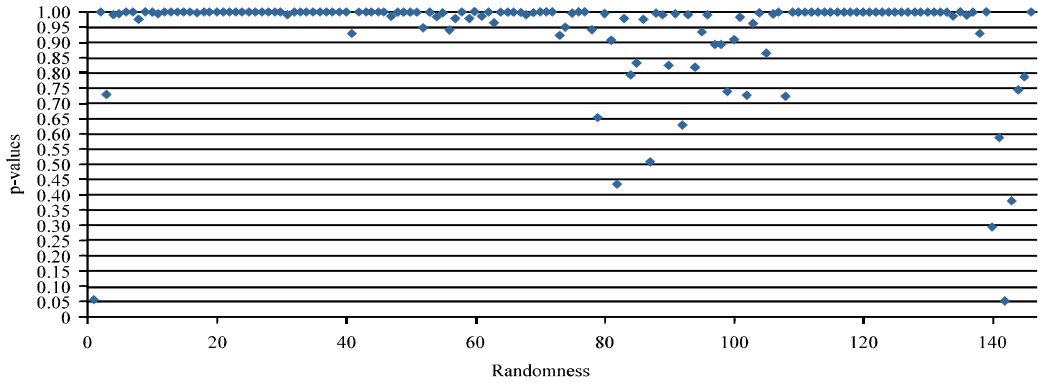

Fig. 2: p-value from plaintext (first experiment)

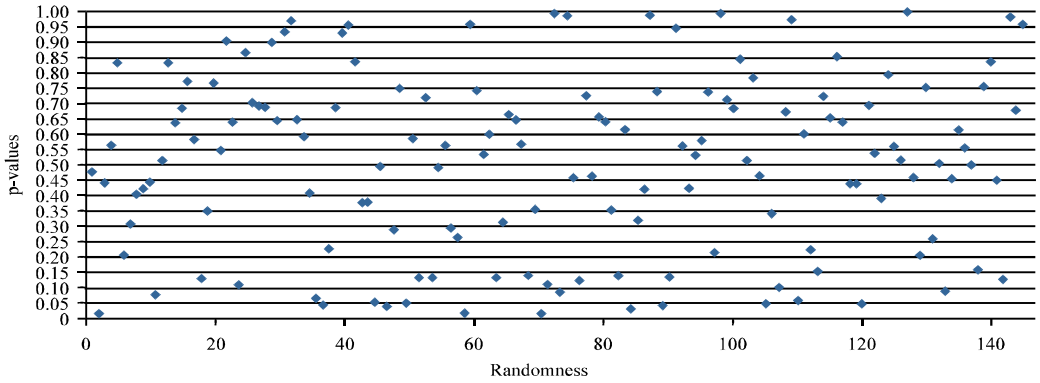

Fig. 3: p-value of ciphertext from MKP (first experiment)

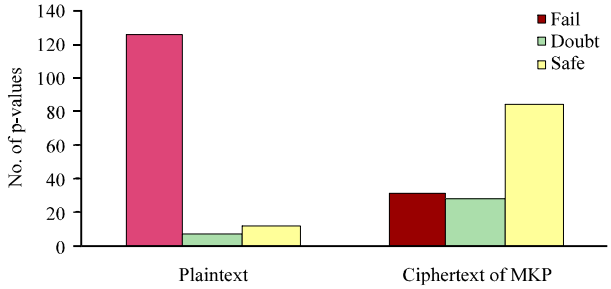

Fig. 4: p-values comparison of first experiment tests

following: In birthday test the p-value became in the safe area. In binary rank test (for 31×31 matrices) the p-value became in the safe area. In binary rank test (for 32×32 matrices) the p-value became in the doubt area. In binary rank test (for 6×8 matrices) the p-value became in the doubt area. In monkey test twelve p-values became in the safe area and six p-values became in the doubt area. In OPSO test twelve p-values became in the safe area and three p-values became in the doubt area. In OQSO test fifteen p-values became in the safe area and six p-values became in the doubt area. In DNA test twenty p-values became in the safe area and four p-values became in the doubt area. In count-the-1's test on a stream of bytes one of the p-values became in the safe area. In count-the-1's test for specific bytes sixteen p-values became in the safe area and five p-values became in the doubt area. In parking lot test the p-value became in the safe area. In minimum distance test the p-value became in the safe area. In 3D spheres test the p-value became in the safe area. In squeeze test the p-value became in the doubt area. In craps test one of the p-value became in the safe area.

In this experiment the tests are proved the randomness of the MKP-AES algorithm. Figure 4 compares the randomness state of the plaintext and its ciphertext received from MKP-AES. It is clear that the using of MKP-AES reduces the number of p-values in fail area and increases the number of p-values in the safe area.

Second experiment: In the second experiment four binary files with 11 MB size are tested. In the first test the plaintext is tested to evaluate its randomness before ciphering and the results are shown in Fig. 5. The p-values belong to diehard tests were as following: In birthday test the p-value

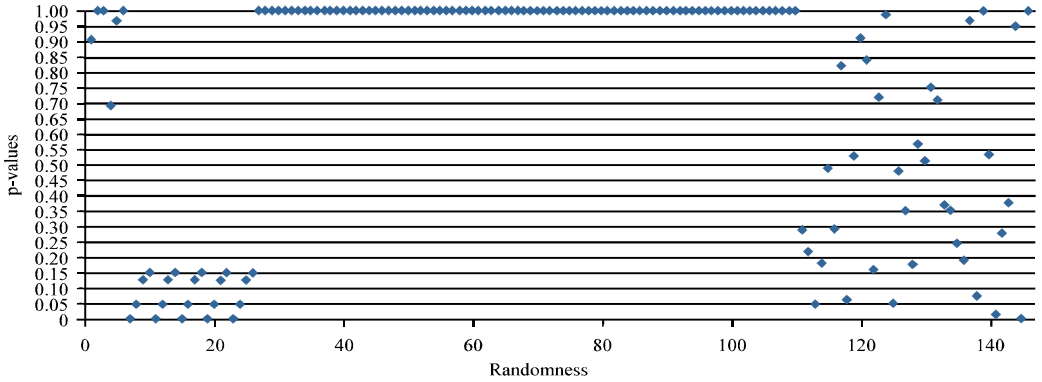

Fig. 5: Plaintext randomness

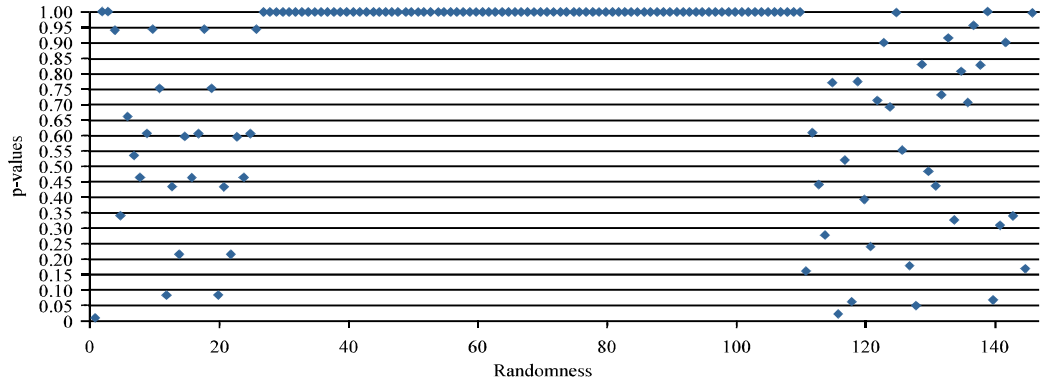

Fig. 6: Ciphertext randomness of fixed key AES

was close to one. In overlapping 5-permutation test the p-values were ones. In binary rank test (for 31×31 matrices) the p-value was very close to 0.7 (in safe area). In binary rank test (for 32×32 matrices) the p-value was very close to one. In binary rank test (for 6×8 matrices) the p-value was one. In monkey tests ten p-values were very close to zero and ten p-values were very close to 0.15 (in doubt area). In OPSO, OQSO, DNA and count-the-1's on a stream of bytes tests all the p-values were ones. In count-the-1's test for specific bytes three p-values were close to zero, two p-values were close to ones, six p-values in doubt area and other p-values were in safe area. In parking lot test the p-value was close to 0.8 (doubt area). In minimum distance test the p-value was close to one. In 3D spheres test the p-value was close to zero. In squeeze test the p-value was one. In overlapping sum test the p-value was very close to 0.5 (in safe area). In runs test one p-values was close to zero, one p-value was close to one and other two p-values were close 0.3 (in safe area). In craps test one of the p-value was very close to zero and the other p-value was one.

In the second test the randomness of ciphertext is examined and the results are shown in Fig. 6. The ciphertext is received from AES algorithm with fixed secret key. The diehard tests results in terms of p-values improvement were as following: In binary rank test (for 32×32 matrices) and binary rank test (for 6×8 matrices) the p-values became in the safe area. In monkey test eleven p-values became in the safe area and four p-values became in the doubt area. In count-the-1's test for specific bytes twelve p-values became in the safe area and seven p-values became in the doubt area. In parking lot test the p-value became in the safe area. In 3D spheres test the p-value became

in the doubt area. In runs test three p-value became in the safe area. In craps test one of the p-value became in the doubt area.

In the third test the randomness of ciphertext is computed using diehard tests and the results are shown in Fig. 7. The ciphertext file is received from AES algorithm with random secret key. The diehard tests results in terms of p-values improvement were as following: the p-value in binary rank test for 31×31 matrices, for 32×32 matrices and for 6×8 matrices became in the safe area. In monkey test seventeen p-values became in the safe area and three p-values became in the doubt area. In count-the-1's test for specific bytes fourteen p-values became in the safe area and five p-values became in the doubt area. In parking lot test the p-value became in the safe area. In minimum distance test the p-value became in the safe area. In overlapping test the p-value became in the safe area. In runs test two p-value became in the safe area.

The fourth test is evaluated the randomness of the ciphertext that received from MKP-AES and the results are shown in Fig. 8. The diehard tests results in terms of p-values improvement were as following: In birthday test the p-value became in the doubt area. In overlapping 5-permutation test one p-value became in the doubt area. In binary rank test (for 31×31 matrices) the p-value became in the safe area. In binary rank test (for 32×32 matrices) the p-value became in the safe area. In binary rank test (for 6×8 matrices) the p-value became in the safe area. In monkey test twelve p-values became in the safe area and five p-values became in the doubt area. In OPSO test thirteen p-values became in the safe area and five p-values became in the doubt area. In OQSO

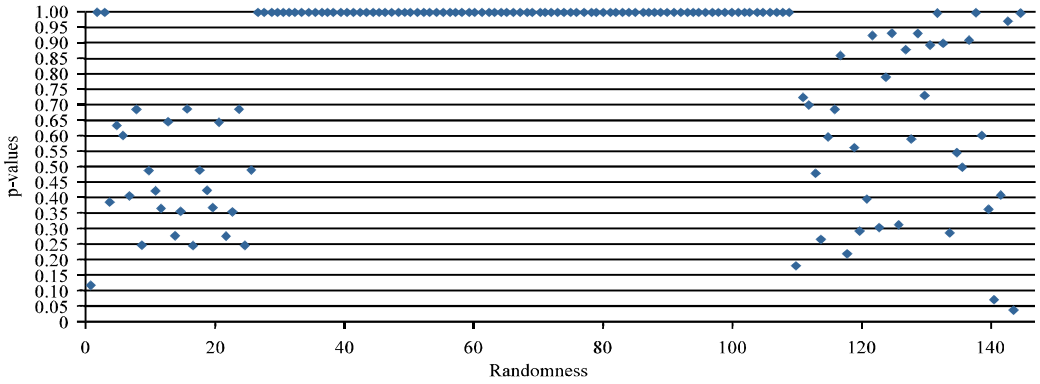

Fig. 7: Ciphertext randomness of random key AES

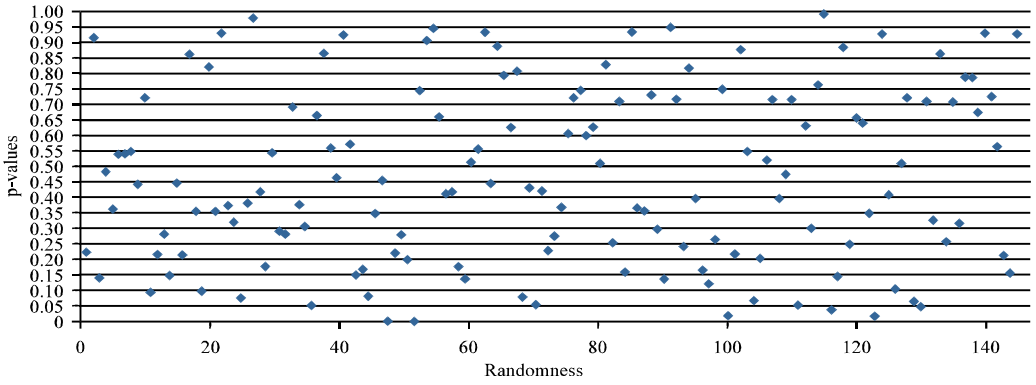

Fig. 8: Ciphertext randomness of MKP-AES

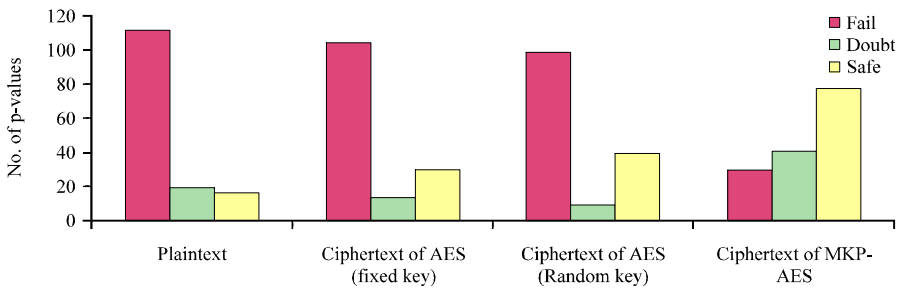

Fig. 9: p-values comparison of second experiment tests

test fifteen p-values became in the safe area and seven p-values became in the doubt area. In DNA test seventeen p-values became in the safe area and ten p-values became in the doubt area. In count-the-1's test on a stream of bytes the p-values became in the doubt area. In count-the-1's test for specific bytes thirteen p-values became in the safe area and four p-values became in the doubt area. In parking lot test the p-value became in the safe area. In minimum distance test the p-value became in the safe area. In 3D spheres test the p-value became in the doubt area. In squeeze test the p-value became in the doubt area. In runs test two p-value were in the safe area and one p-value was in the doubt area. In craps test one of the p-value became in the doubt area.

In the second experiment the tests are performed over plaintext and three ciphertexts. The p-values resulted from all tests are compared in Fig. 9. The number of p-values in failure area is reduced from 111 to 104, 98 and 29 in AES (fixed key), AES (random key) and MKP-AES algorithms, respectively. Also the number of p-values in safe area is improved from 16 in plaintext to 29, 39 and 77 in ciphertexts of AES (fixed key), AES (random key) and MKP-AES algorithms, respectively. From second experiment we found that AES algorithm has no impact on some diehard tests like over lapping 5-permutation, OPSO, OQSO, DNA, Count the 1's on a stream of bytes and squeeze tests. While MKP-AES improved the randomness when it changed most of the p-value of previous mentioned tests to be in safe and doubt areas. Now it is clear that MKP-AES algorithm has improved the randomness of ciphertext over original AES algorithm.

REFERENCES

Abomhara, M., O.O. Khalifa, O. Zakaria, A.A. Zaidan, B.B. Zaidan and H.O. Alanazi, 2010. Suitability of using symmetric key to secure multimedia data: An overview. J. Applied Sci., 10: 1656-1661.

Al-Alak, S., Z. Ahmed, A. Abdullah and S. Subramiam, 2011. AES and ECC mixed to raise ZigBee wireless sensor security. Proceedings of the International Conference on Sensor Networks, Information and Ubiquitous Computing, September 30, 2011, Singapore, pp: 535-539.

Alani, M., 2010. Testing randomness in ciphertext of block-ciphers using diehard tests. Int. J. Comp. Sci., 10: 53-57.

FDK, 2003. The evaluation of randomness of RPG100 by using NIST and diehard tests. FDK corporation, http://www.fdk.co.jp/cyber-e/pdf/HM-RAE104.pdf

Juan, S., 2011. Randomness testing of the AES candidate algorithms. http://csrc.nist.gov/archive/aes/round1/r1-rand.pdf

LAN/MAN, 2006. Part 15.4: Wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (WPANs), IEEE-SA standards board. Standards Committee of the IEEE Computer Society, http://profsite.um.ac.ir/~hyaghmae/ACN/WSNMAC1.pdf

Lv, Z.Y., P. Zhou, L. Tang and X.X. Jing, 2010. The connective mechanism of BACnet and 6LoWPAN. Inform. Technol. J., 9: 1222-1225.

Montenegro, G., N. Kushalnagar, J. Hui and D. Culler, 2007. Transmission of IPv6 Packets over IEEE 802.15.4 Networks. RFC 4944, IETF, September 2007. http://tools.ietf.org/html/rfc4944

Muda, Z., R. Mahmod and M.R. Sulong, 2010. Key transformation approach for Rijndael security. Inform. Technol. J., 9: 290-297.

NIST, 2001. Announcing the advanced encryption standard (AES). NIST special publication federal information processing standards publication 197. http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

Rabah, K., 2005a. Implementation of elliptic curve diffie-hellman and ec encryption schemes. Inf. Technol. J., 4: 132-139.

Rabah, K., 2005b. Theory and implementation of data encryption standard: A review. Inf. Technol. J., 4: 307-325.

Rabah, K., 2005c. Theory and implementation of elliptic curve cryptography. J. Applied Sci., 5: 604-633.

Random, 2006. Diehard battery of tests random. http://physik.uibk.ac.at/statistik/Diehard_random_tests.pdf

STG, 2007. In-mesh: WirelessHARTTM field device software. Software Technologies Group, http://www.stg.com/wireless/STG_Data_Sheet_WiHART_Software.pdf

Salem, Y., M. Abomhara, O.O. Khalifa, A.A. Zaidan and B.B. Zaidan, 2011. A review on multimedia communications cryptography. Res. J. Inform. Technol., 3: 146-152.

Sharma, S., R.C. Jain and S. Bhadauria, 2006. A power efficient encryption algorithm for multimedia data in mobile Ad hoc network. Trends Applied Sci. Res., 1: 416-425.

Soto, J. and L. Bassham, 2000. Randomness testing of the advanced encryption standard finalist candidates. http://csrc.nist.gov/publications/nistir/ir6483.pdf.

Xiao, Y., H.H. Chen, B. Sun, R. Wang and S. Sethi, 2006. Mac security and security overhead analysis in the IEEE 802.15.4 wireless sensor networks. EURASIP J. Wireless Commun. Network., 2006: 1-12.

Zaidan, A.A., B.B. Zaidan, A.K. Al-Frajat and H.A. Jalab, 2010. An overview: Theoretical and mathematical perspectives for advance encryption standard/Rijndael. J. Applied Sci., 10: 2161-2167.