Journal of Applied Sciences

ISSN 1812-5654

science alert

ANSInet
an open access publisher
http://ansinet.com

Journal of Applied Sciences 13 (6): 768-789, 2013
ISSN 1812-5654 / DOI: 10.3923/jas.2013.768.789

Randomness Analysis on Blowfish Block Cipher Using ECB and CBC Modes

[1,2]Ashwak Mahmood Alabaichi, [3]Ramlan Mahmood,
[1]Faudziah Ahmad and [4,5]Mohammed S. Mechee
[1]Department of Information Technology, University Utara Malaysia,
06010, Sintok, Malaysia
[2]Department of Computer Science, Faculty of Sciences, Kerbala University, Kerbala, Iraq
[3]Faculty of Computer Science and Information Technology, University Putra Malaysia,
Erdang, Selangor, Malaysia
[4]Department of Mathematics, Faculty of Mathematics and Computer Sciences,
Kufa University, Najaf, Iraq
[5]Institute of Mathematical Sciences, University of Malaya,
50603 Kuala Lumpur, Malaysia

Abstract: Randomness of the output is one of the significant factors in measuring the security of any cryptographic algorithm. Non-random block cipher is vulnerable to any type of attack. This paper presents the National Institute of Standard and Technology (NIST) statistical tests of the Blowfish algorithm to investigate its randomness. Blowfish algorithm with Electronic Codebook (ECB) and Cipher Block Chaining (CBC) modes were conducted for these tests. In addition, comparisons between them were introduced. The analysis showed that Blowfish algorithm with ECB mode was inappropriate with data such as text and image files which have large strings of identical bytes. This inconsistency is due to the majority of the 188 statistical tests of NIST statistical tests failing in all rounds.

Key words: Block cipher, blowfish algorithm, ECB mode, CBC mode, randomness test

INTRODUCTION

Blowfish algorithm was designed by Schneier at the Cambridge Security Workshop in December 1993 to replace the Data Encryption Standard (DES). This fast, free alternative to existing encryption algorithms has since been widely analyzed and gradually accepted as a good and powerful encryption algorithm offering several advantages, among them its suitability and efficiency for implementing hardware. It is also unpatented and therefore does not require any license. The elementary operators of Blowfish algorithm comprise table lookup, addition and XOR with the table being made up of four S-boxes and a P-array. Based on Feistel rounds, Blowfish algorithm is a cipher with the F-function design being a simplified version of the principles employed in DES to provide similar security, faster speed and higher efficiency in software. Due to its good encryption rate in software, no effective cryptanalysis has been found to date (Schneier, 1994; Meyers and Desoky, 2008; Mousa, 2005; Thakur and Kumar, 2011). Even though it is not as well-known as AES, the uniqueness of Blowfish algorithm and the efficiency of its algorithm have led to its growing popularity in the open source community (Cornwell, 2012).

The block cipher requires the generated cipher text to be distributed uniformly when dissimilar plaintext blocks are used during encryption. By statistically analyzing the block cipher it can be determined if the tested algorithm meets this requirement. A non-random block cipher can be susceptible to attacks of many types (Isa and Z'Aba, 2012).

The test suite Rukhin *et al.* (2010) from NIST was selected for testing Blowfish algorithm generated sequences. These statistical tests are appropriate for evaluating generators of random and pseudo-random numbers that cryptographic applications use. To the knowledge of the researchers, there have not been any statistical tests performed on Blowfish algorithm with ECB and CBC modes.

Corresponding Author: Ashwak Mahmood Alabaichi, Department of Computer Science, Faculty of Sciences, Kerbala University, Kerbala, Iraq

The five sections in this study include the following: Section 2 describes the Blowfish algorithm, ECB and CBC modes; section 3 categorizes and explains each Blowfish algorithm Data type for statistical test; section 4 provides the results of the experiment and empirical analysis of the randomness testing on Blowfish algorithm with ECB and CBC modes, while section 5 provides the conclusion and recommendations for future work.

BLOWFISH BLOCK CIPHER

Blowfish algorithm is a symmetric block cipher using a Feistel network, iterating simple encryption and decryption functions of 16 times. Each A Feistel structure offers various advantages which are appeared, particularly in hardware system. The decryption process of the cipher text required only reversing the key schedule. The Blowfish algorithm can be divided into: key expansion and data encryption (Cornwell, 2012; Schneier, 1994; Kumar *et al*., 2010).

Key Expansion of the Blowfish algorithm starts with the P-array and S-boxes and utilizes many subkeys, which have to be precomputed before data encryption or decryption. The Parray comprises eighteen 32-bit subkeys: P1, P2... P18.

In this section a key of maximum 448 bits is converted into several subkey arrays up to a total of 4168 bytes.

There are 256 entries for each of the four 32-bit S-boxes:

- S1, 0, S1, 1,..., S1, 255
- S2, 0, S2, 1,..., S2, 255
- S3, 0, S3, 1,..., S3, 255
- S4, 0, S4, 1,..., S4, 255

How these subkeys are calculated is explained below:

- First, the P-array is initialized followed by the four S-boxes, with a fixed string, which has the hexadecimal digits of p_i
- XOR P1 with the key's first 32 bits, XOR P2 with its second 32 bits and so on, until the key's bits are up to P14. The cycle is iterated through the key bits until the entire P-array has been XOR-ed with key bits
- The Blowfish algorithm is then used to encrypt the all-zero string, employing the described subkeys in steps 1 and 2
- P1 and P2 are replaced with the step 3 out put
- Encrypt the step 3 output with the Blow fish algorithm using the modified subkeys
- Replace P3 and P4 with the output of step 5
- The process is continued and all elements of the P-array are replaced, followed by all four S-Boxes, with the output continuously changing

Data encryption commences with a 64-bit block element of plaintext morphing into a 64 bit ciphertext. The 64-bit segment is instantly broken up into two equal segments that are then used in the Blowfish algorithm as the base. The exclusive-or-operation (XOR) is then executed between the first 32-bit block segment (L) and the first P-array. The obtained 32-bit data go to the F function which permutes the data to give a 32-bit block segment, which is XOR'ed with the second 32-bit segment (R) from the 64-bit plaintext split. On completion of the XOR operation, the 32-bit segments L and R are swapped for future iterations of the Blowfish algorithm. Figure 1 illustrates the architecture of the Blowfish algorithm with 16 rounds. The input is a 64-bit data element, X, which is divided into two 32-bit halves: XL and XR.

F-Function of blowfish algorithm is probably the most complex part of the algorithm and the only part that utilizes the S-Boxes. It accepts a 32-bit stream of data and splits that into four equal sections. Each 8-bit subdivision is changed into a 32-bit data stream using their corresponding S-Box. The obtained 32-bit data is XOR'ed or combined to give a final 32-bit value for permutations of the Blowfish algorithm (note that all addition is modulo 2^{32}). Figure 2 Describes the architecture of the F

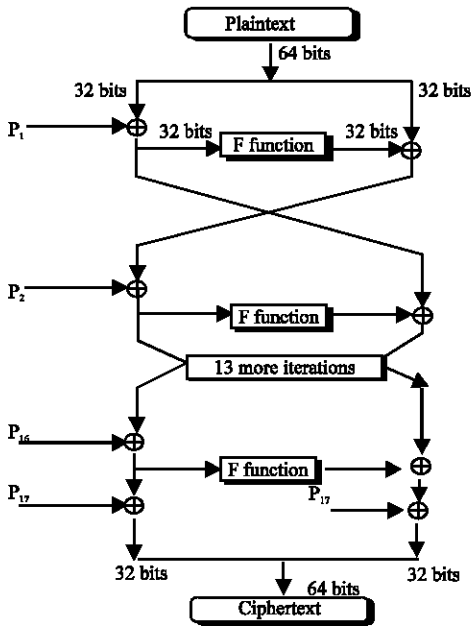


Fig. 1: Blowfish architecture

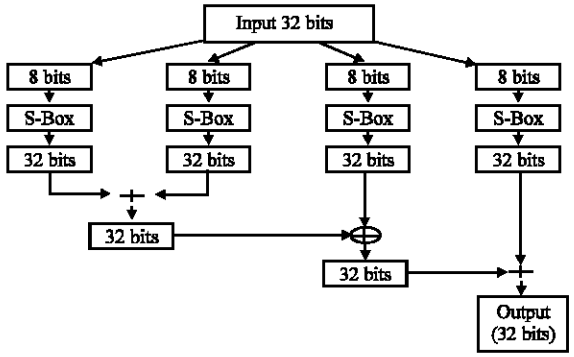


Fig. 2: F-function architecture

function (Cornwell, 2012; Bagad and Dhotre, 2008; Van Tilborg and Jajodia, 2011; Schneier, 1996).

Decryption is similar to encryption but P1, P2...P18 are used in the reverse order: In order to reduce the risks of clever attack on the cipher, consecutive blocks of message can be chained together so that identical blocks of plaintext are not seen as identical blocks in ciphertext. The attacker would then not be able to identify the file type. A random non-zero initialization vector IV of the same length as a common block is used to begin the chain. CBC mode is demonstrated by the following steps (Meyers and Desoky, 2008; Dworkin, 2005).

- CBC encryption:

$$C_1 = CIPH_K (P_1 \oplus IV)$$

$$C_j = CIPH_K (P_j \oplus C_{j-1})$$

- CBC decryption:

$$P_1 = CIPH^{-1}{}_K (C_1) \oplus IV$$

$$P_j = CIPH^{-1}{}_K (C_j) \oplus C_{j-1}$$

for j = 2 ... n

where, P_j is the jth plaintext block. C_j the jth ciphertext block. $CIPH_K$ the forward cipher function of the block cipher algorithm under the key K is applied to the data block X. $CIPH^{-1}{}_K$ is the inverse cipher function of the block cipher algorithm under the key K is applied to the data block X.

The Electronic Codebook (ECB) mode is another mode which is confidentiality-based, with a key assigned to a fixed ciphertext block for each plaintext block, similar to assigning codes in a codebook (Dworkin, 2005).

Table 1: Breakdown of the 188 statistical tests performed through analysis

Statistical test	No. of p-values	Test ID
Frequency	1	1
Block frequency	1	2
Cumulative sum	2	3-4
Runs	1	5
Longest run	1	6
Rank	1	7
FFT	1	8
Non overlapping template	148	9-156
Overlapping template	1	157
Universal	1	158
Approximate entropy	1	159
Random excursions	8	160-167
Random excursions variant	18	168-185
Serial	2	186-187
Linear complexity	1	188

The definition of the Electronic Codebook (ECB) mode is:

- ECB encryption:

$$C_j = CIPH_K (P_j)$$

- ECB Decryption:

$$P_j = CIPH^{-1}{}_K (C_j), \text{ for } j = 1 \ldots n$$

Blowfish data types: Testing the randomness on the Blowfish algorithm was done by applying the NIST Statistical Suite (Rukhin *et al*., 2010). All such testing consisted of 15 core statistical tests that can be viewed as 188 statistical tests under different parameter inputs, Table 1 shows the individual core statistical test, followed by the number of tests done for each core test. In this section, we provide four Categories of Data such as, Random Plaintext/Random 128-Bit keys (Soto, 1999), image, text and video files.

Random plaintext/random 128-bit keys: The basis of this experiment was the data generated with the Blum-Blum-Shub (BBS) pseudorandom bit generator because it is a cryptographically secure pseudo-random bit generator and similar to the data type used in testing Advanced Encryption Standard Finalist Candidates (Menezes *et al*., 1997).

A total of 128 sequences were established for the purpose of examining the randomness of ciphertext (based on random plaintext and random 128-bit keys. Each sequence resulted from the concatenation of 16256 ciphertext blocks of length 64 bits (1040384 bits) using 16256 random plaintexts blocks of length 64 bits and a random 128-bit key in ECB mode one time and another time in CBC.

Image files: This experiment was based on a data set of 128 sequences of image files in different formats. Each sequence resulted from the concatenation of 24580 (1573120 bits) ciphertext block of 64-bit length using 24580 plaintexts blocks of the 64-bit length and a random 256-bit key in ECB mode one time and another time in CBC mode.

Text files: This experiment was based on a data set of 128 text files, with each file consisting of a sequence resulting from the concatenation of 16256 (1040384 bits) ciphertext block of 64-bit length using 16256 plaintexts blocks of 64-bit length and a random 256-bit key in ECB mode one time and another time in CBC.

Video files: This experiment was based on a data set of 128 video files. Each file consisted of a sequence resulting from the concatenation of 16256 (1040384 bits) ciphertext blocks of 64-bit length using 16256 plaintexts blocks of 64-bit length and a random 256-bit key in ECB mode one time and another time in CBC mode.

EXPERIMENTAL RESULTS

Testing the randomness of the Blowfish algorithm was done on the four types of data mentioned in the previous section in both partial and full round considerations (Soto and Bassham, 2000; Isa and Z'Aba, 2012).

Full round testing (FRT): When testing in full round with Blowfish algorithm, all four data types were generated, meaning that the data derived have to complete 16 for all types of data (Soto and Bassham, 2000; Isa and Z'Aba, 2012; Ariffin *et al*., 2011).

Partial round testing (PRT): Soto and Bassham (2000) tested Fesitel network Twofish rounds in pairs. After each round, some of the data bits were left basically intact without change and thus twofish appeared to be non-random after one round under the test conditions but the result was different after two rounds when the data bits were affected. Twofish rounds were tested in pairs with even numbered rounds from 2-14. Therefore, In Partial Round Testing on Blowfish algorithm, all four data types were generated using the Partial round of Blowfish algorithm in pairs from 2-14. In the following we discuss the output of implementing Random Test for the four types of data on Blowfish algorithm with ECB and CBC modes in PRT and FRT, respectively.

Random plaintext/random 128-bit keys: We illustrate the Random Plaintext/Random 128-bit keys files results of PRT and FRT for Blowfish algorithm with ECB and CBC modes in Fig. 3 and 4, respectively. In each of the dished line at 96.09% indicated the smallest

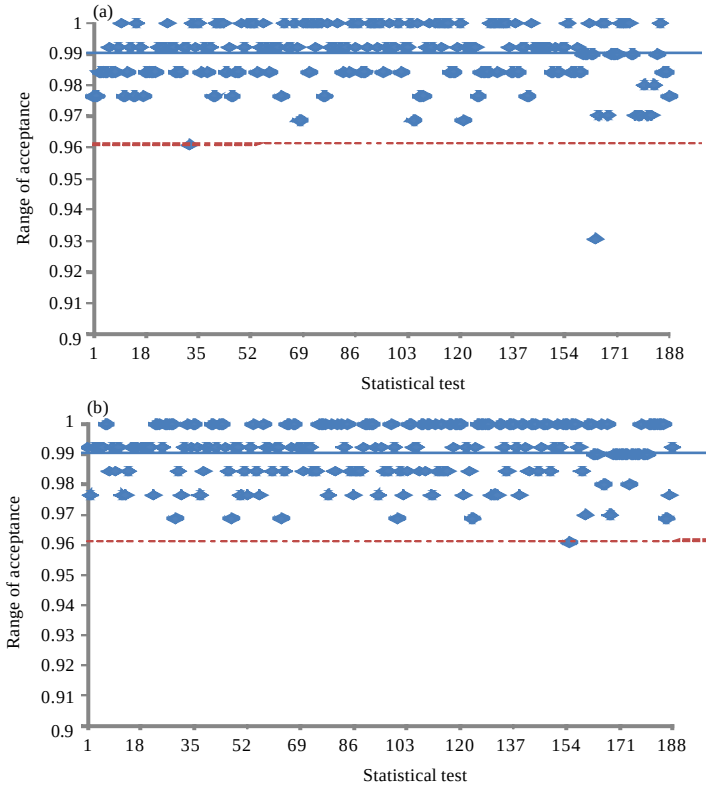


Fig. 3(a-h): Continue

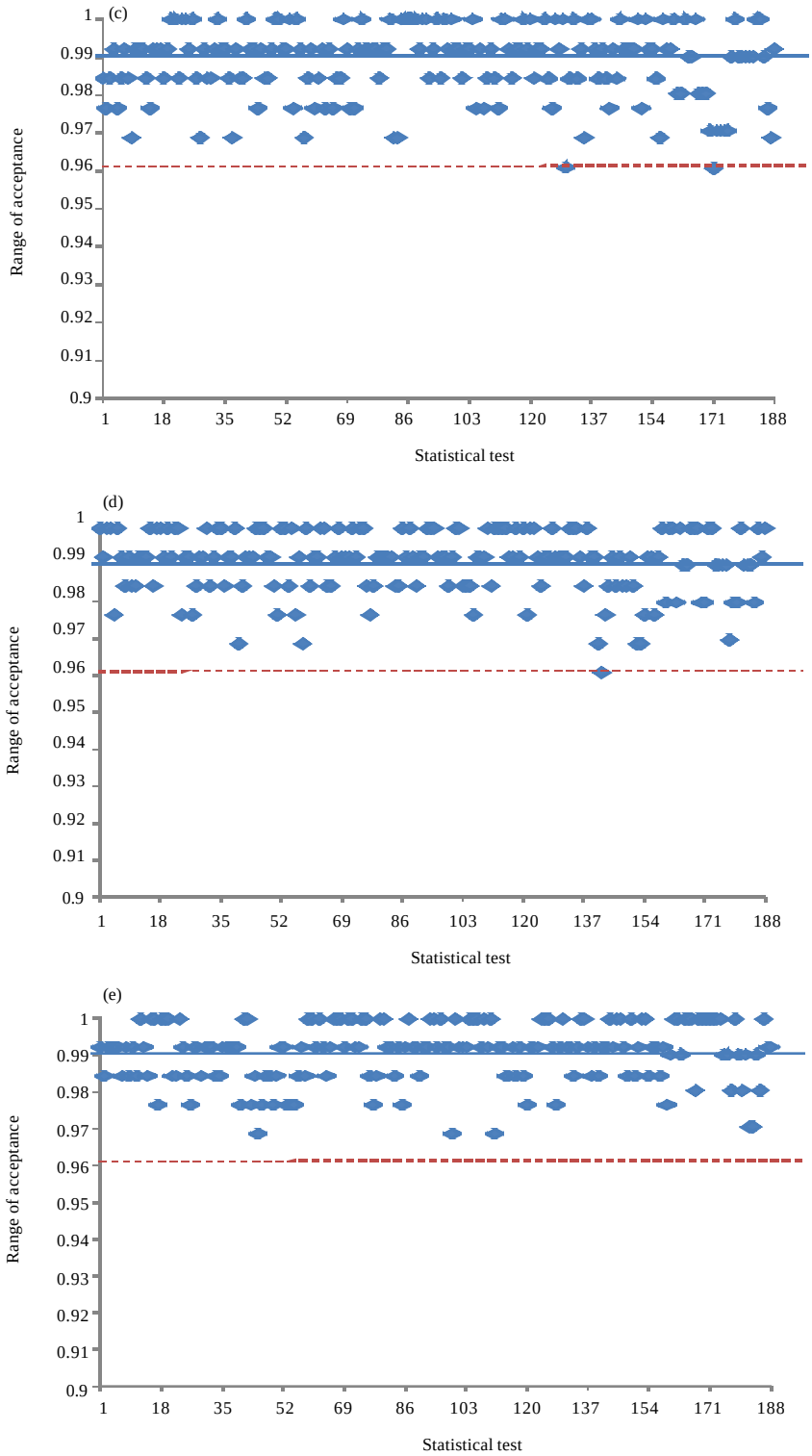


Fig. 3(a-h): Continue

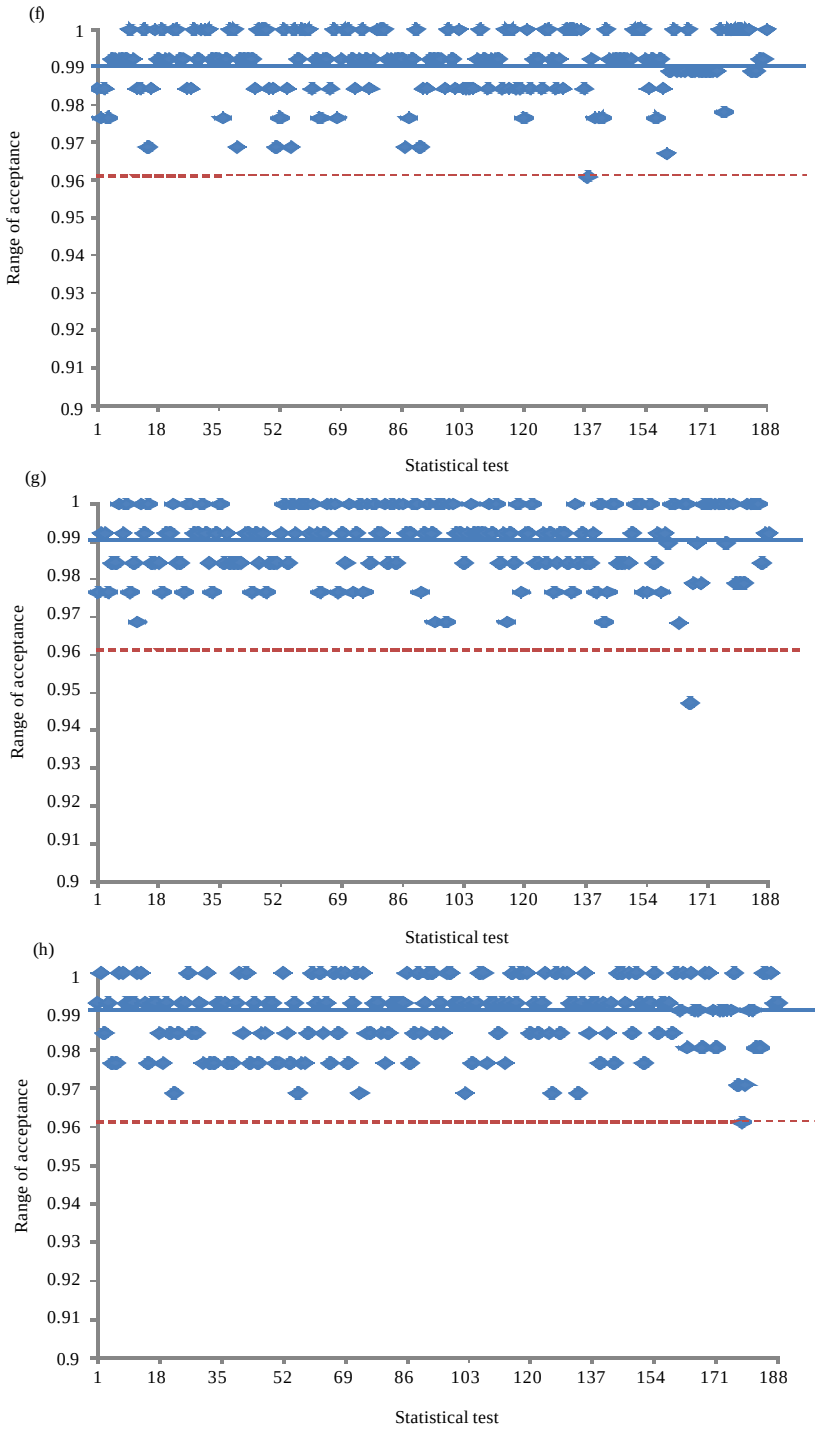


Fig. 3(a-h): Results of random plaintext/random 128-bit keys for (a) Second roud with ECB mode, (b) Fourth round with ECB mode, (c) Sixth round with ECB mode (d) Eighth round with ECB mode, (e) Tenth round with ECB mode, (f) Twelfth round with ECB mode, (g) Fourteenth round with ECB mode and (h) Sixteenth round with ECB mode

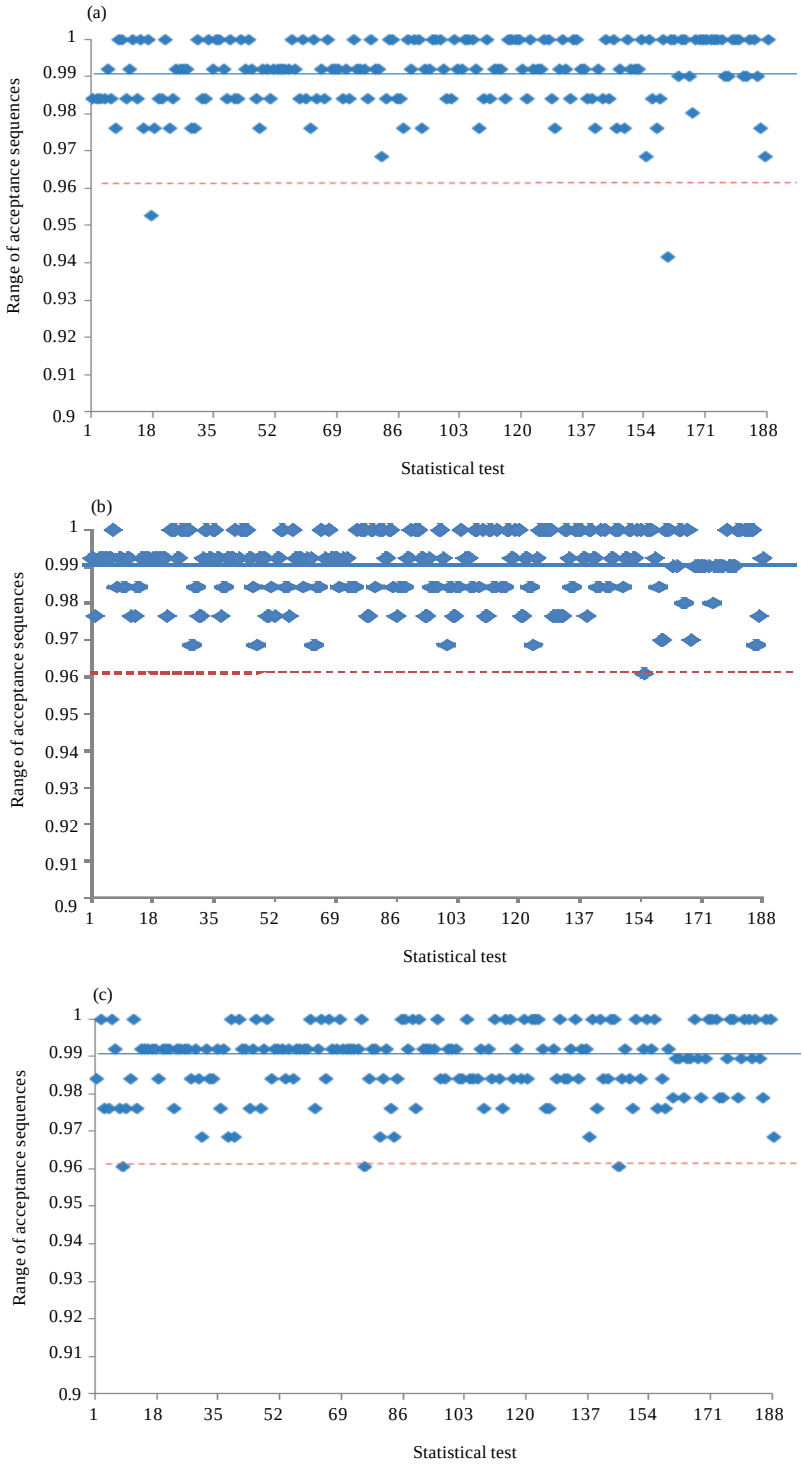
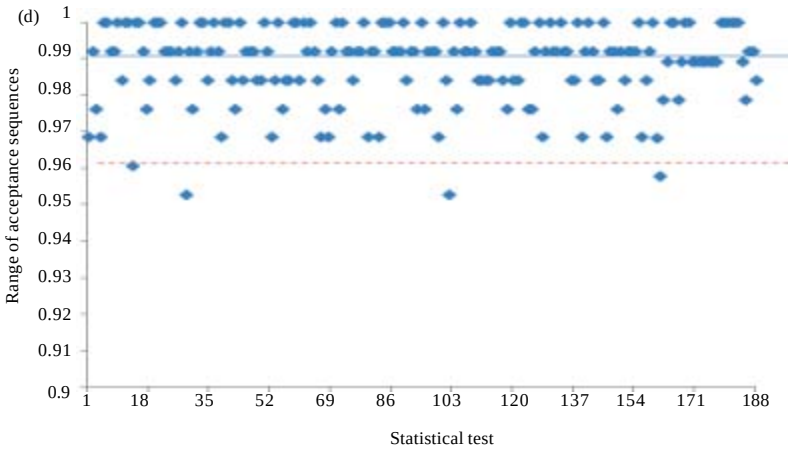
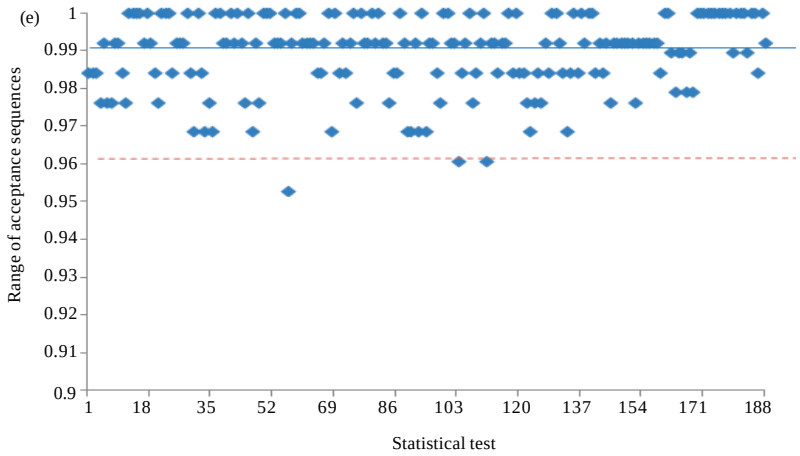


Fig. 4(a-h): Continue





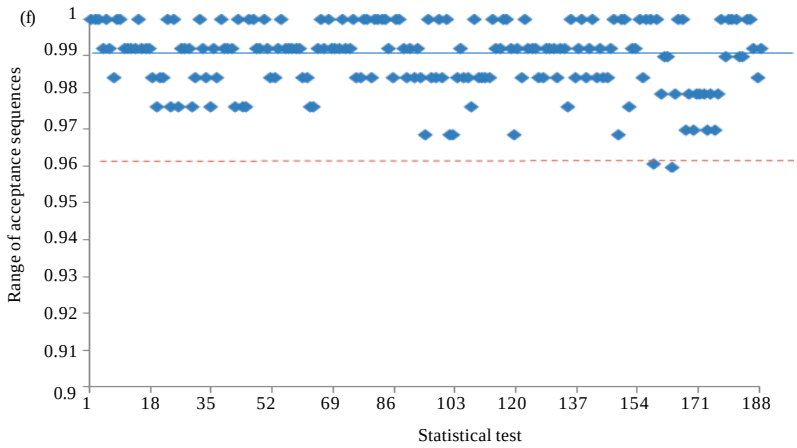


Fig. 4(a-h): Continue

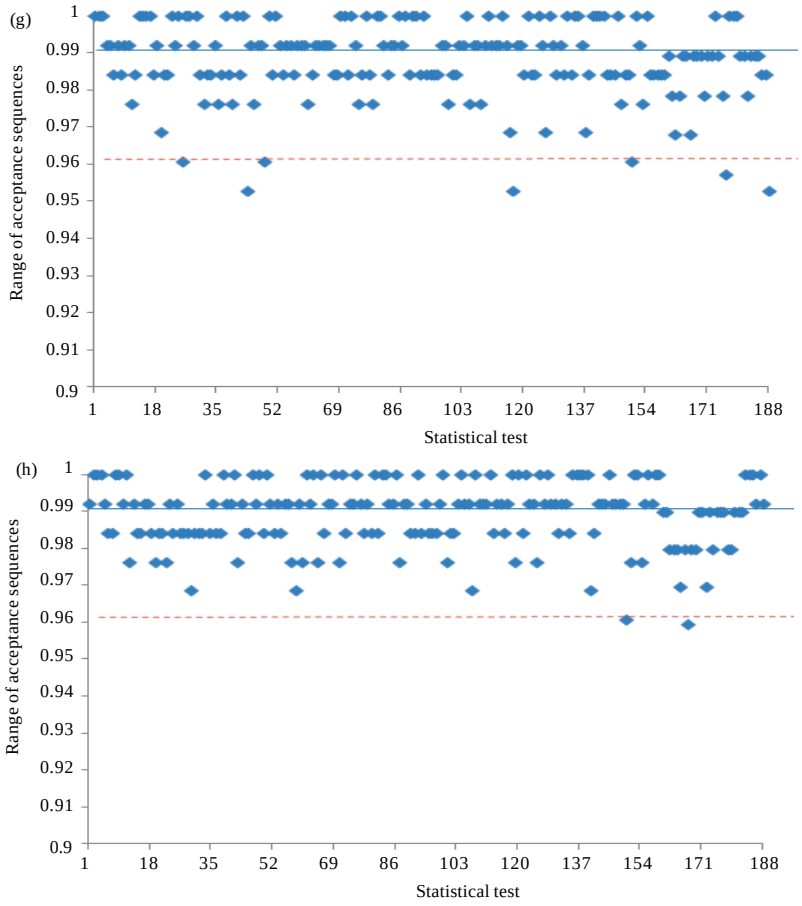


Fig. 4(a-h): Results of random plaintext/random 128-bit keys for (a) Second round with CBC mode, (b) Fourth round with CBC mode, (c) Sixth round with CBC mode, (d) Eighth round with CBC mode, (e) Tenth round with CBC mode, (f) Twelfth round with CBC mode, (g) Fourteenth round with CBC mode and (h) Sixteenth round with CBC mode

proportion satisfying the 0.01 criterion of acceptance, while the solid line at 99% indicated the proportion expected.

It is evident that the output of Blowfish algorithm with both modes on this type of data is random for all rounds because most of the 188 statistical tests were above 96%.

Image files: We illustrate the image files results of PRT and FRT for Blowfish algorithm with ECB and CBC modes in Fig. 5 and 6, respectively.

It is evident that Blowfish algorithm with CBC mode is random for all rounds because most of the 188 statistical tests were above 96%, while the output from the Blowfish algorithm with ECB is non-random for all rounds because most of the 188 statistical tests were below 96%.

Text files: We illustrate the text files results of PRT and FRT for Blowfish algorithm with ECB and CBC modes in Fig. 7 and 8, respectively.

It is evident that the output from Blowfish algorithm with CBC mode is random for all rounds because most of the 188 statistical tests were above 96%, while the output from the Blowfish algorithm with ECB mode is non-random for all rounds as the majority of the 188 statistical tests were below 96%.

Video files: We illustrate the Video files results of PRT and FRT for Blowfish algorithm with ECB and CBC modes in Fig. 9 and 10, respectively.

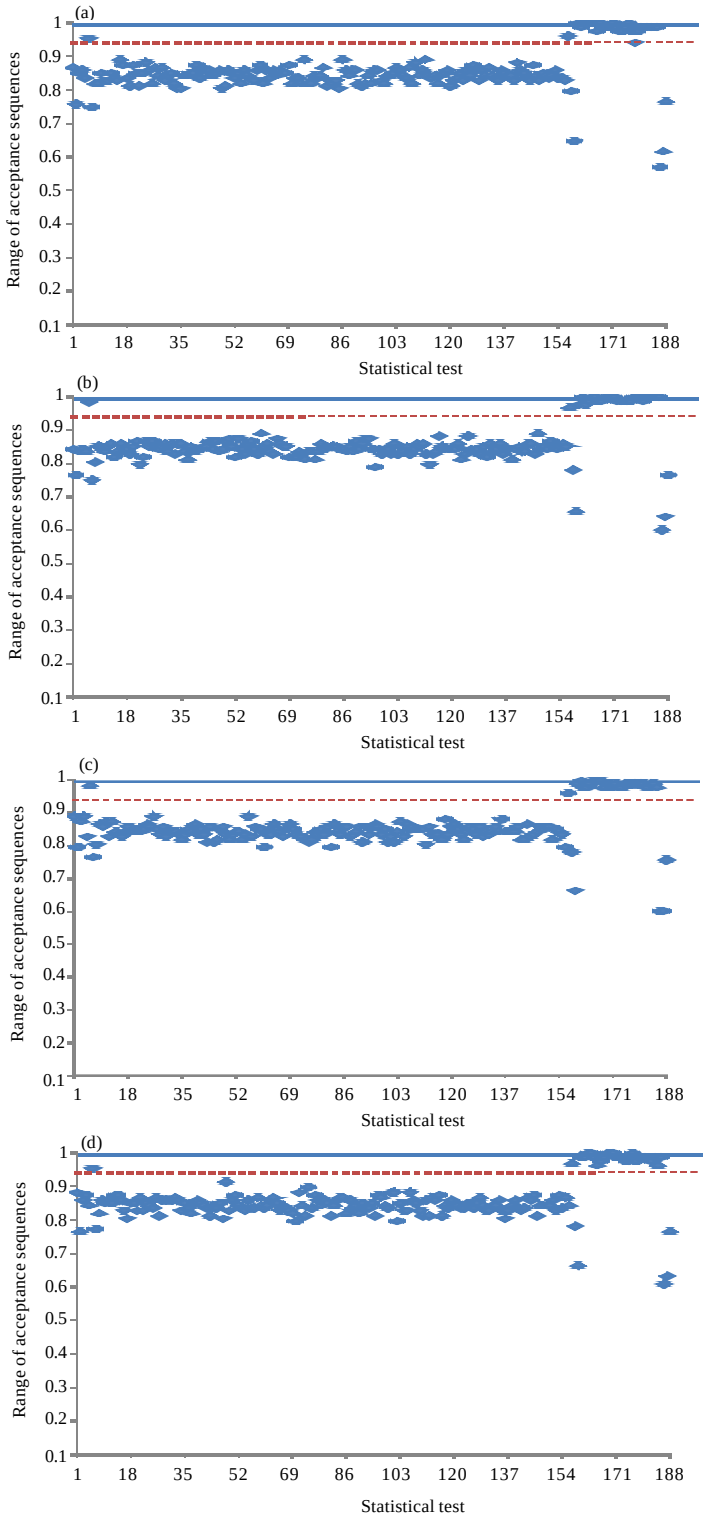


Fig. 5(a-h): Continue

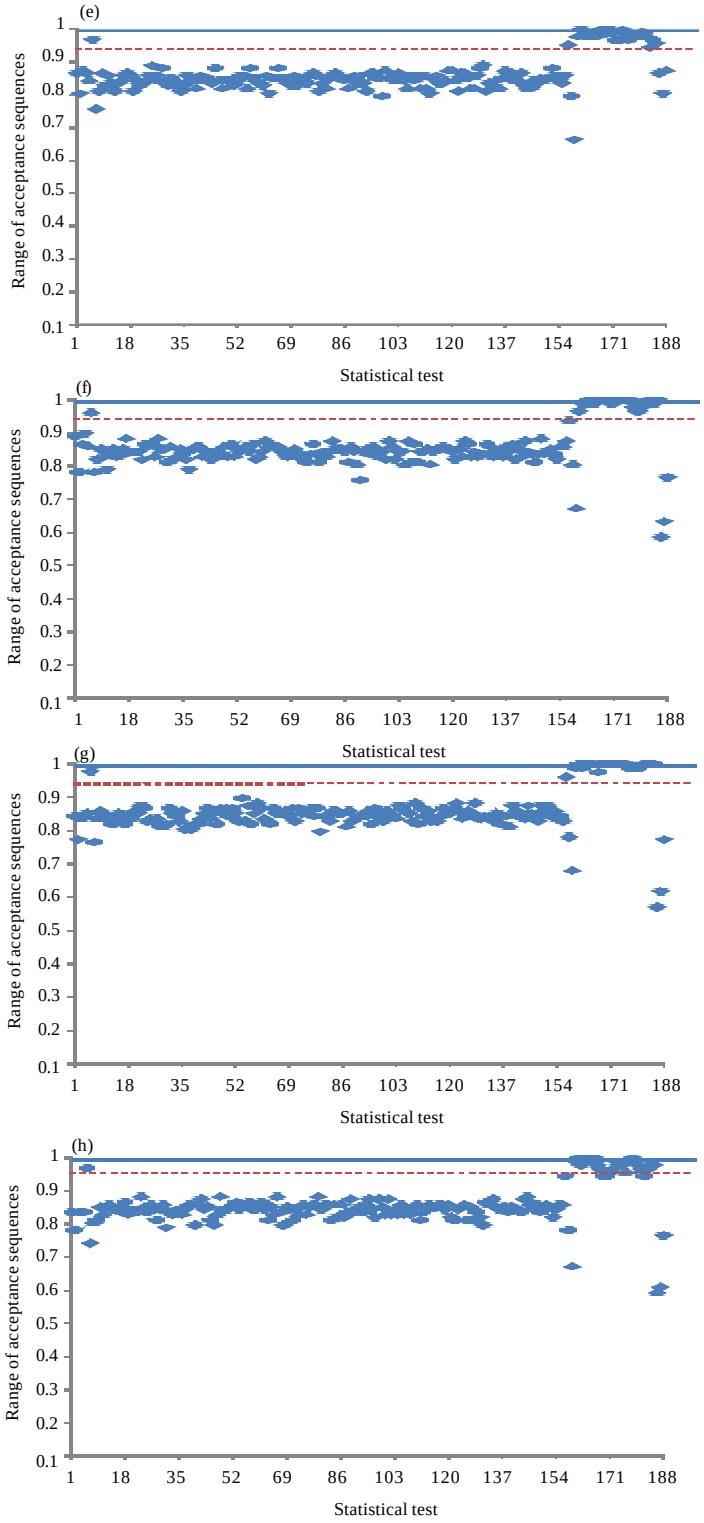


Fig. 5(a-h): Results of image for (a) Second round with ECB mode, (b) Fourth round with ECB mode, (c) Sixth round with ECB mode, (d) Eighth round with ECB mode, (e) Tenth round with ECB mode, (f) Twelfth round with ECB mode, (g) Fourteenth round with ECB mode and (h) Sixteenth round with ECB mode

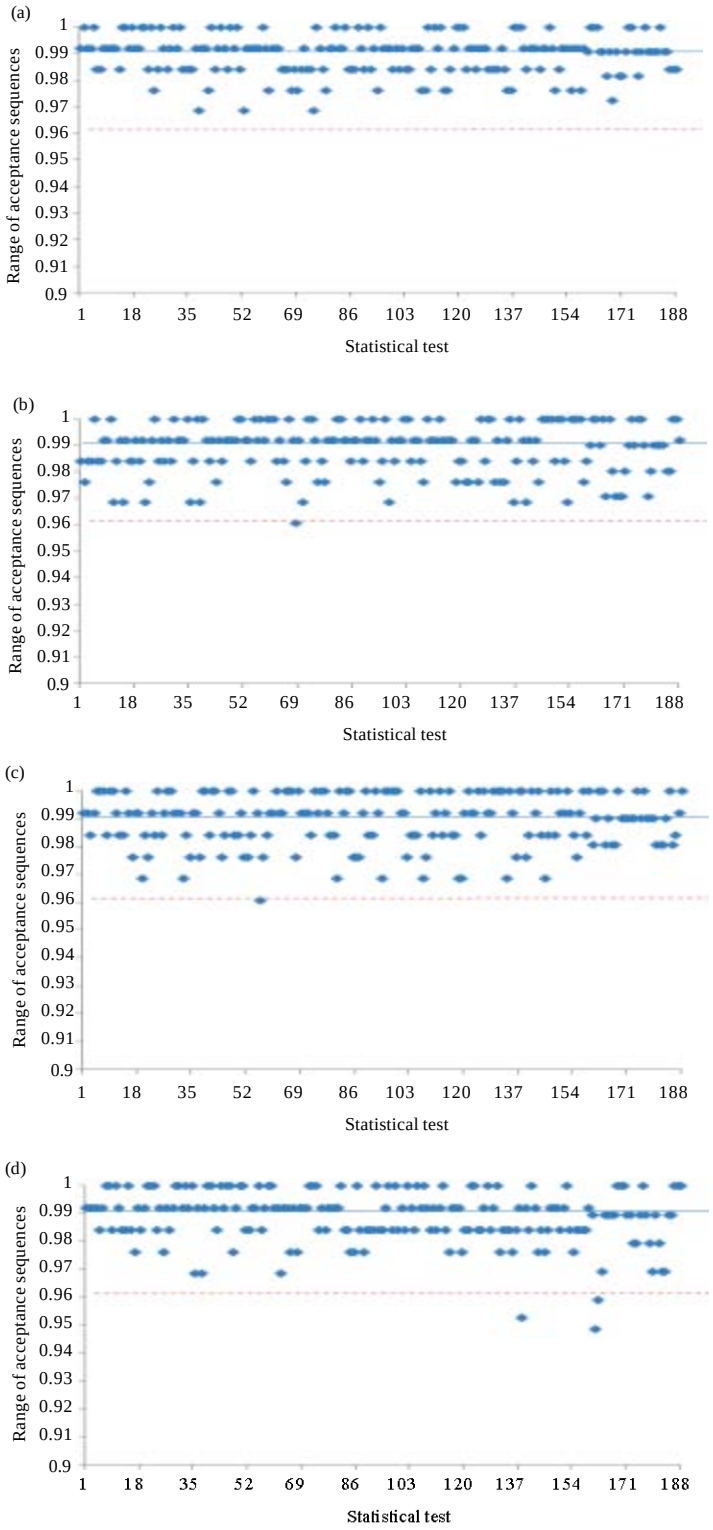


Fig. 6(a-h) Continue

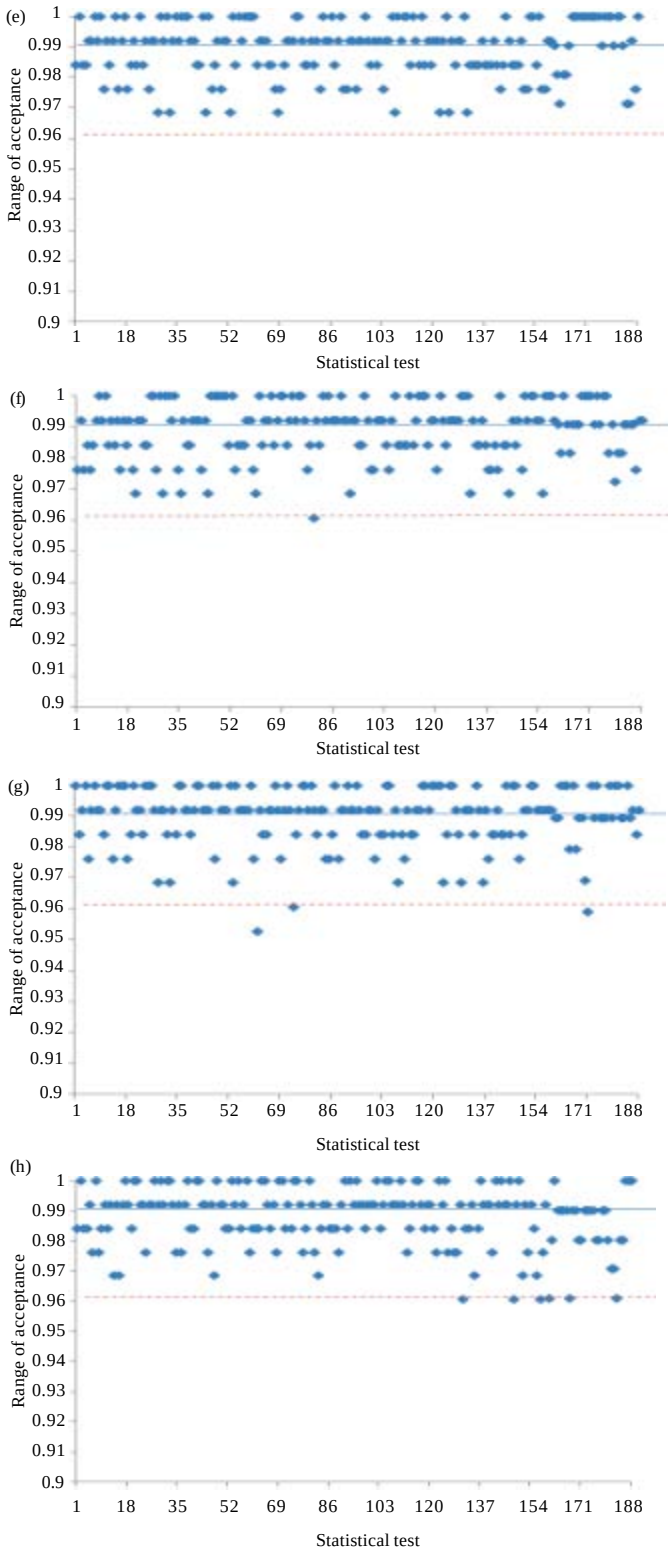


Fig. 6(a-h): Results of image for (a) Second round with CBC mode, (b) Fourth round with CBC mode, (c) Sixth round with CBC mode, (d) Eighth round with CBC mode, (e) Tenth round with CBC mode, (f) Twelfth round with CBC mode, (g) Fourteenth round with CBC mode and (h) Sixteenth round with CBC mode

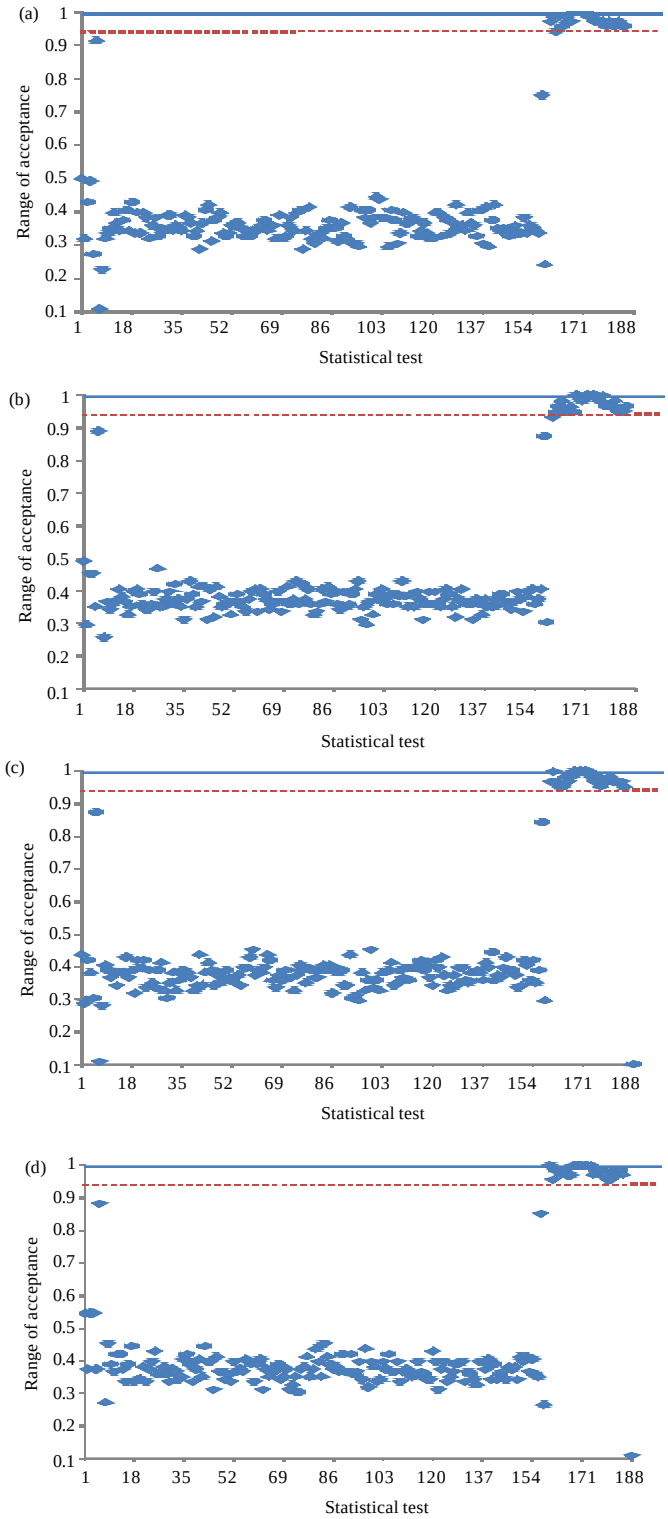


Fig. 7(a-h): Continue

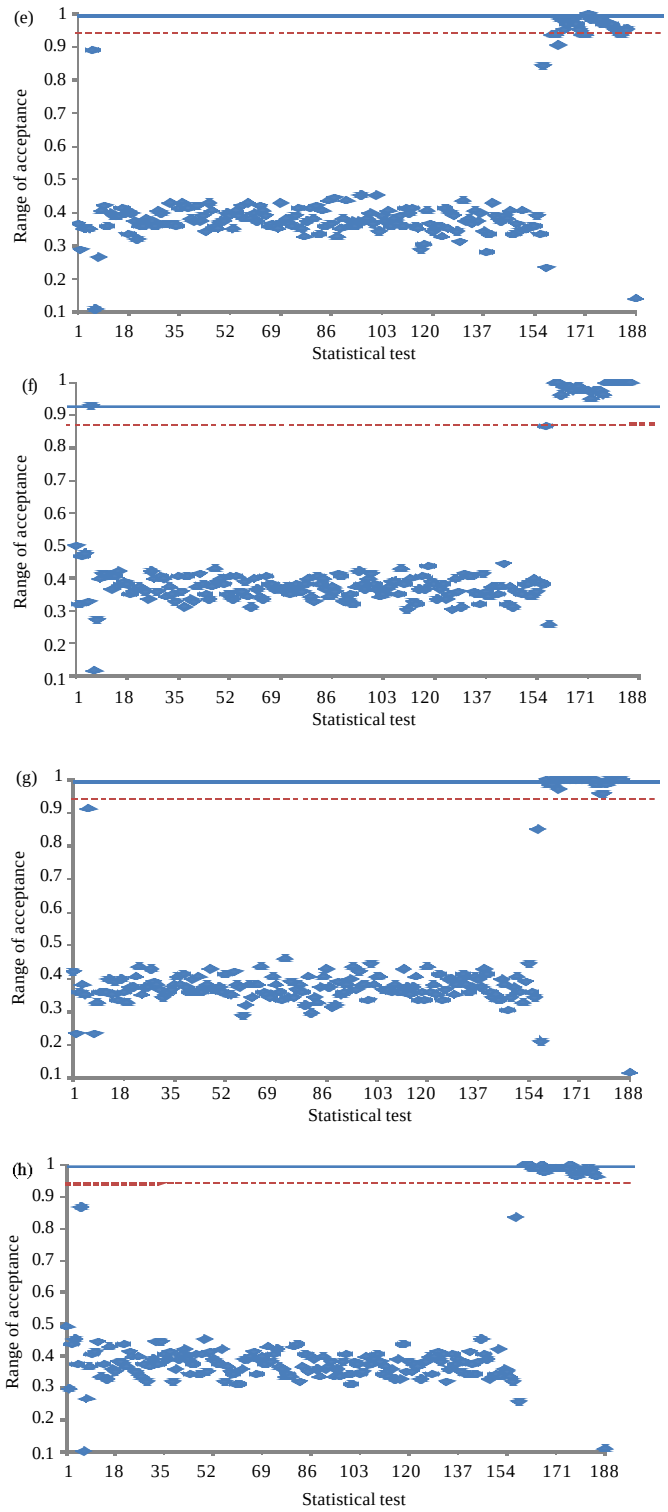


Fig. 7(a-h): Results of text for (a) Second round with ECB mode, (b) Fourth round with ECB mode, (c) Sxth round with ECB mode, (d) Eighth round with ECB mode, (e) Tenth round with ECB mode, (f) Twelfth round with ECB mode, (g) Fourteenth round with ECB mode and (h) Sixteenth round with ECB mode

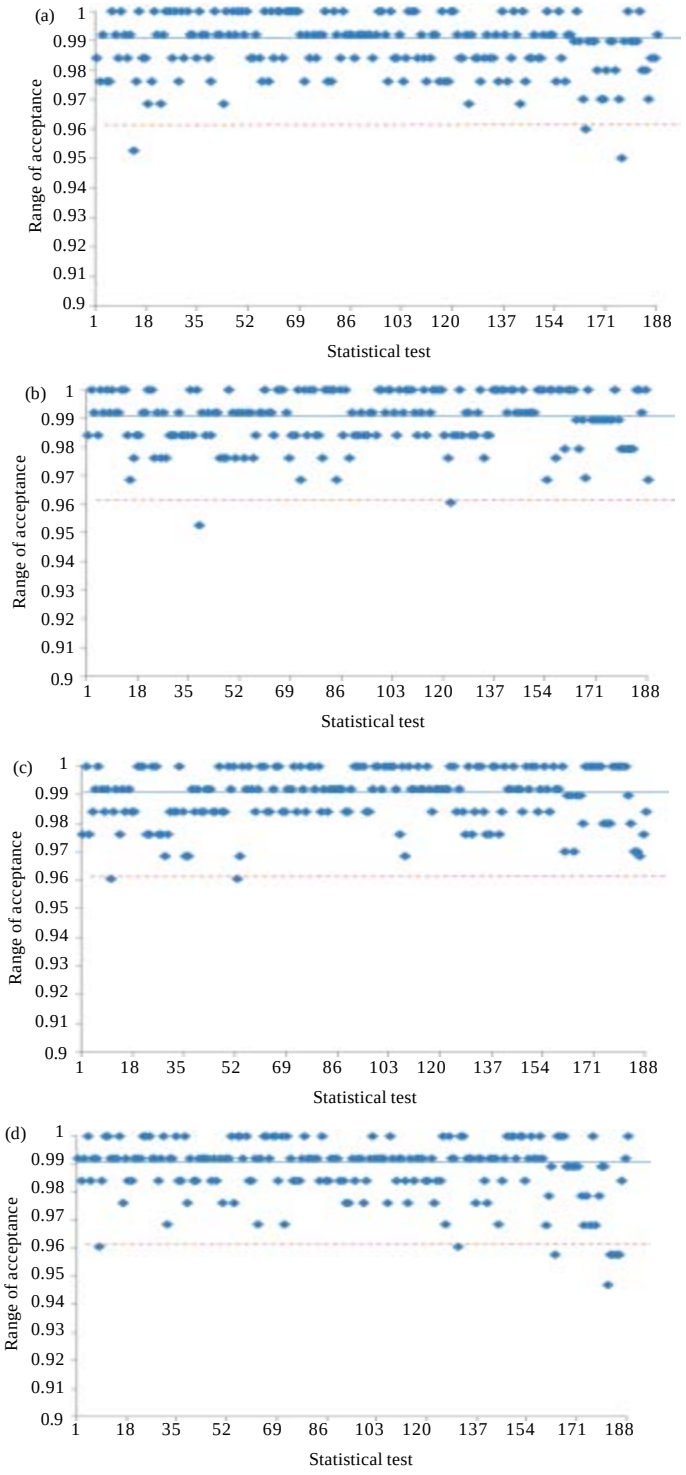


Fig. 8(a-h): Continue

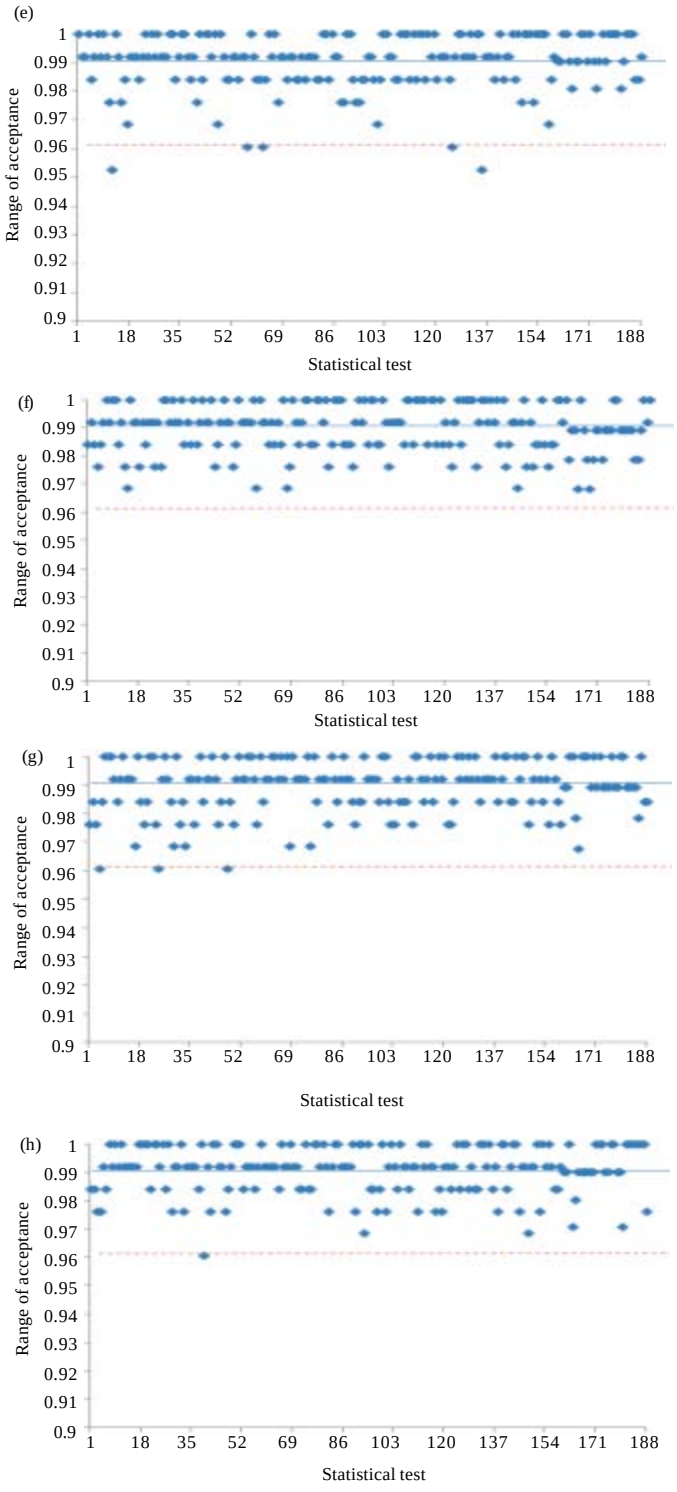


Fig. 8(a-h): Results of text for (a) Second round with CBC mode, (b) Fourth round with CBC mode, (c) Sixth round with CBC mode, (d) Eighth round with CBC mode, (e) Tenth round with CBC mode, (f) Twelfth round with CBC mode, (g) Fourteenth round with CBC mode and (h) Results of text for sixteenth round with CBC mode

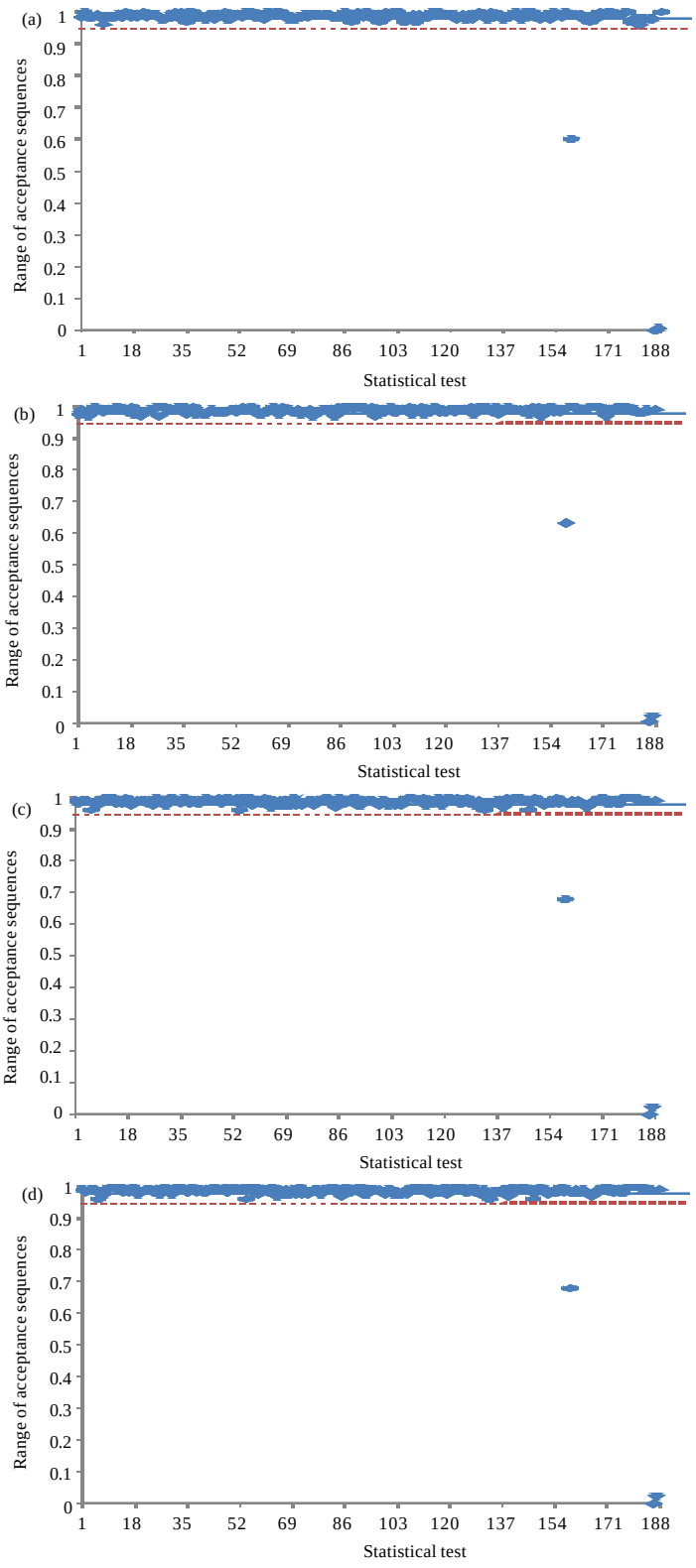


Fig. 9(a-h): Continue

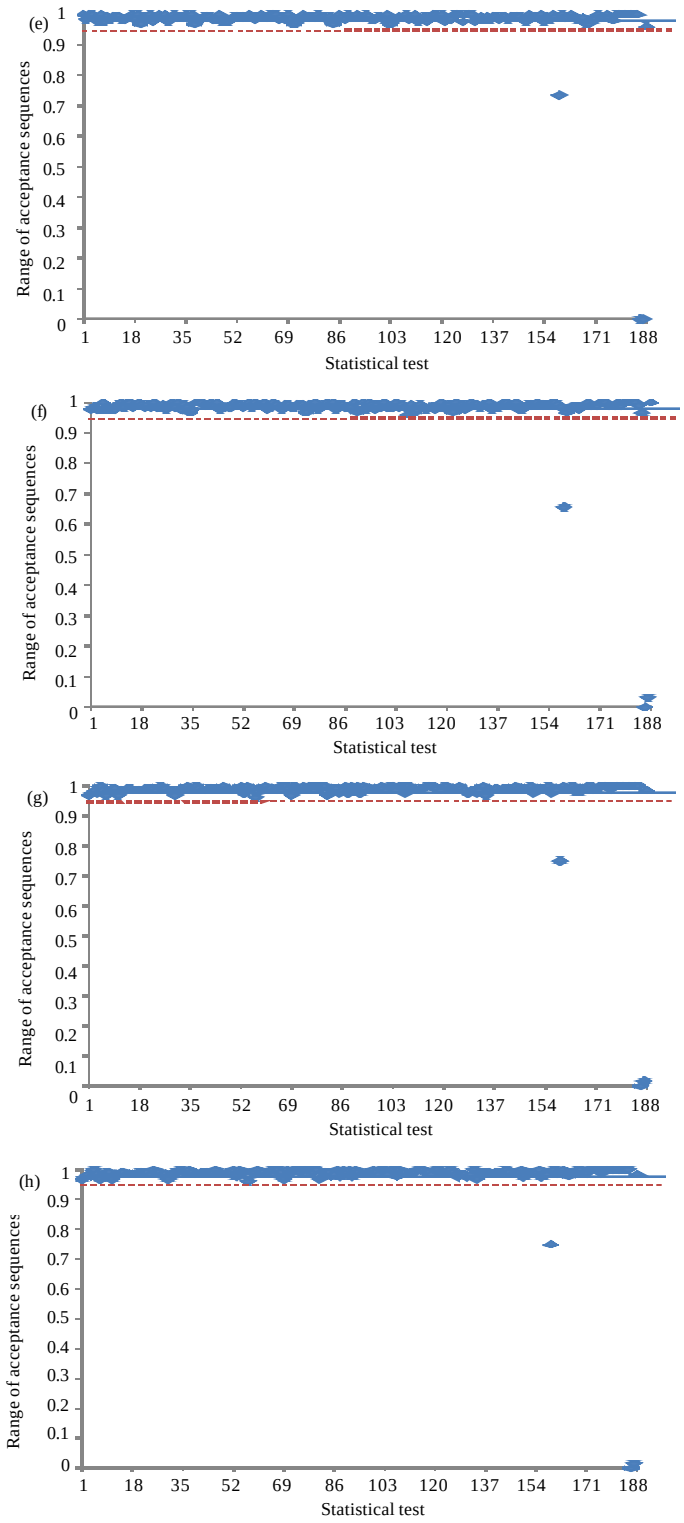


Fig 9(a-h): Results of video for (a) Second round with ECB mode, (b) Fourth round with ECB mode, (c) Sixth round with ECB mode, (d) Eighth round with ECB mode, (e) Tenth round with ECB mode, (f) Twelfth round with ECB mode, (g) Fourteenth round with ECB mode and (h) Sixteenth round with ECB mode

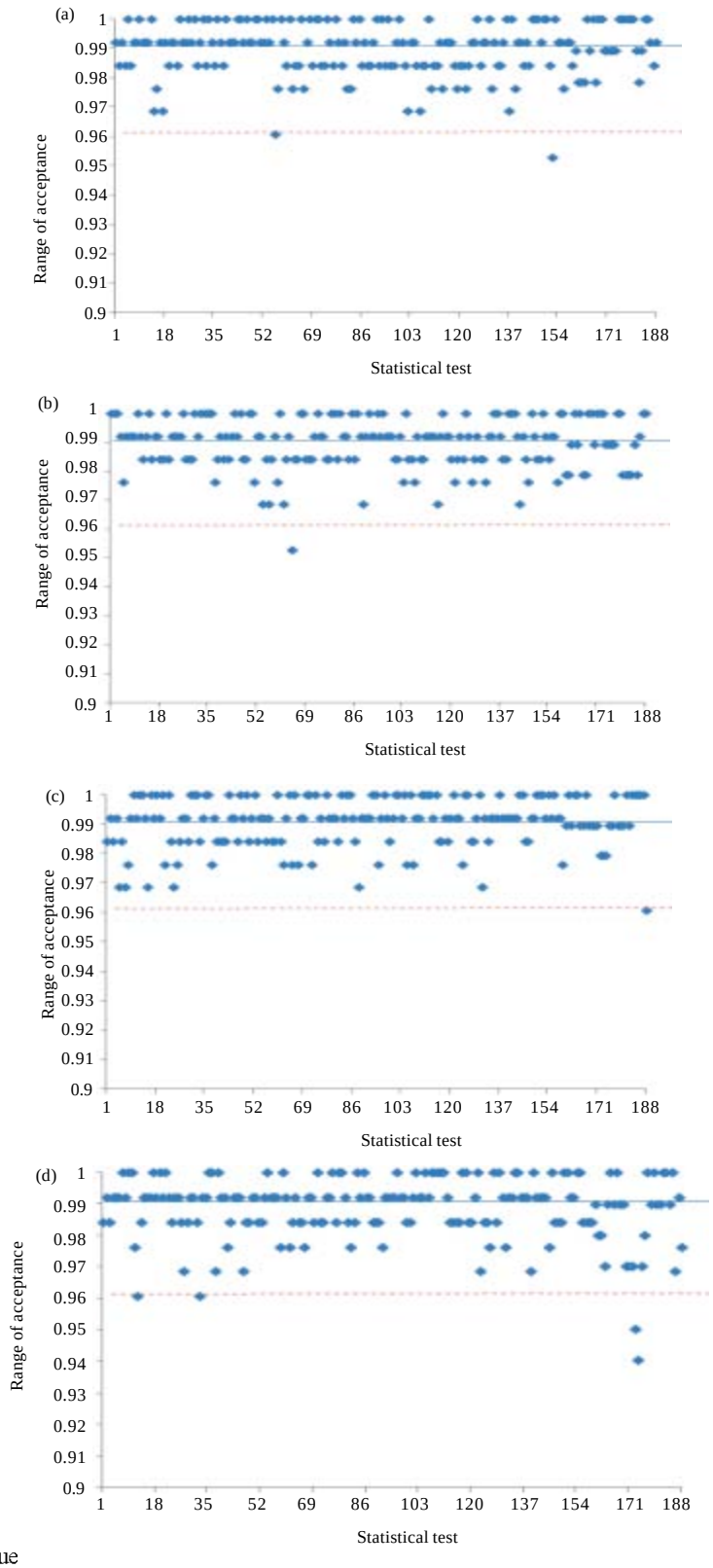


Fig. 10(a-h): Continue

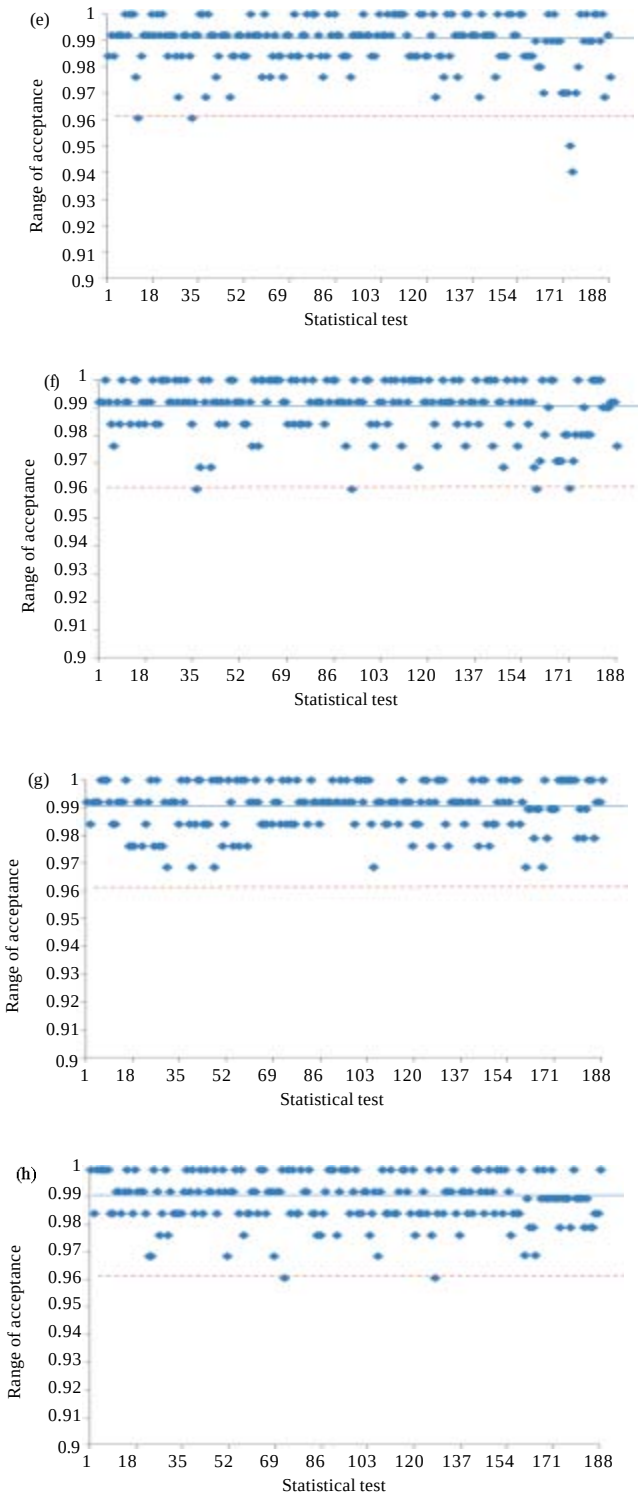


Fig. 10(a-h): Results of video for (a) Second round with CBC mode, (b) Fourth round with CBC mode, (c) Sixth round with CBC mode, (d) Eighth round with CBC mode, (e) Tenth round with CBC mode, (f) Twelfth round with CBC mode, (g) Fourteenth round with CBC mode and (h) Sixteenth round with CBC mode

It is obvious that the output from Blowfish algorithm with CBC mode is random for all rounds because most of the 188 statistical tests were above 96%, while the output from the Blowfish algorithm with ECB mode on completion of first round, seems to be non-random because most of the 188 statistical tests were below 96%. From the end of the second round until the end of the sixteenth round, statistics produced are similar to Blowfish algorithm with CBC mode except Approximate Entropy (159) and serial (186, 187) tests statistical test which were below 96% for all rounds.

CONCLUSION AND FUTURE WORK

From the result it can be seen that Blowfish algorithm with ECB mode is not suitable for image and text files that have large strings of identical bytes. However, using CBC mode with Blowfish algorithm 64-bits could pass NIST tests, meaning that it can be used to encrypt any type of file without restrictions on the file contents. Future work should apply the same experiment on extended Blowfish algorithm 128 bits.

REFERENCES

Ariffin, S., R. Mahmod, A. Jaafar and M.R.K. Ariffin, 2011. Byte permutations in block cipher based on immune systems. Proceedings of the 3rd International Conference on Software Technology and Engineerin, August 12-13, 2011, ASME Press, New York, pp: 1-6.

Bagad, V.S. and A.I. Dhotre, 2008. Cryptography and Network Security. 2nd Revised Edn., Technical Publications, Pune, India, ISBN-13: 9788184313406, Pages: 202.

Cornwell, J.W., 2012. Blowfish survey. Department of Computer Science Columbus State University Columbus,GA.

Dworkin, M., 2005. Recommendation for block cipher modes of operation: The CMAC mode for authentication. National Institute of Standards and Technology, Special Publication 800-38B. csrc.nist.gov/publications/nistpubs/800-38C/SP800-38B.pdf

Isa, H. and M.R. Z'aba, 2012. Randomness analysis on LED block ciphers. Proceedings of the 5th International Conference on Security of Information and Networks, October 25-27, 2012, Jaipur, India, pp: 60-66.

Kumar, R.S., E. Pradeep, K. Naveen and R. Gunasekaran, 2010. A novel approach for enciphering data of smaller bytes. Int. J. Comp. Theory Engine., 2: 654-659.

Menezes, A.J., P.C. van Oorschot and S.A. Vanstone, 1997. Handbook of Applied Cryptography. CRC Press, USA., ISBN: 0-8493-8523-7.

Meyers, R.K. and A.H. Desoky, 2008. An implementation of the blowfish cryptosystem. Proceedings of the IEEE International Symposium, Signal Processing and Information Technology, December 16-19, 2008, Sarajevo, Bosnia and Herzegovina, pp: 346-351.

Mousa, A., 2005. Data encryption performance based on Blowfish. Proceedings of the 47th International Symposium ELMAR, June 8-10, 2005, Zadar, Croatia, pp: 131-134.

Rukhin, A., J. Soto, J. Nechvatal, M. Smid and E. Barker *et al*., 2010. A statistical test suite for random and pseudorandom number generators for cryptographic applications. National Institute of Standards and Technology Special Publication, Report No. 800-22, 2010, pp: 1-131.

Schneier, B., 1994. Description of a New Variable-Length Key, 64- Bit Block Cipher (Blowfish). In: Fast Software Encryption: Cambridge Security Workshop Cambridge, U.K., December 9-11,1993 Proceedings. Anderson, R. (Ed.). Springer-Verlag, London, UK., ISBN: 3-540-58108-1, pp: 191-204.

Schneier, B., 1996. Applied Cryptography: Protocols, Algorithms and Source Code in C. 2nd Edn., John Wiley and Sons, New York, USA., ISBN-13: 978-0471117094, pp: 758.

Soto, J. and L. Bassham, 2000. Randomness testing of the advanced encryption standard finalist candidates. National Institute of Standards and Technology, NIST IR 6483, pp: 1-14. http://csrc.nist.gov/publications/nistir/ir6483.pdf

Soto, J., 1999. Randomness testing of the AES candidate algorithms. National Institute of Standards and Technology, NIST IR 6390, pp: 1-9.

Thakur, J. and N. Kumar, 2011. DES, AES and Blowish: Symmetric key cryptography algorithms simulation based performance analysis. Int. J. Emerg. Technol. Adv. Eng., 1: 6-12.

Van Tilborg, H.C.A. and S. Jajodia, 2011. Encyclopaedia of Cryptography and Security. 2nd Edn., Springer, USA., ISBN 978-1-4419-5906-5.