A New ShiftColumn Transformation: An Enhancement of Rijndael Key Scheduling

Salasiah Sulaiman Zaiton Muda Julia Juremi Ramlan Mahmod Sharifah Md. Yasin
Department of Computer Science, Faculty of Computer Science and Information Technology,
Universiti Putra Malaysia
43400 UPM Serdang, Selangor, Malaysia
salasiah_sulaiman@yahoo.com

ABSTRACT

In this paper, we proposed a new approach for key scheduling algorithm which is an enhancement of the Rijndael key scheduling. This proposed algorithm was developed to improve the weaknesses that has in the Rijndael key schedule. The key schedule function in Rijndael block cipher did not receive the same amount of attention during design phase as the cipher components. Based on our research, there are several properties in key schedule that seemed to violate the design criteria, which was published by NIST, and this has led to many types of attack performed on Rijndael block cipher. Thus we proposed an approach called ShiftColumn, operates by shifting bit and the result will be shifted with different offsets. This transformation is added as the last function after the RCon function. Our new approach improves the security of the original Rijndael key scheduling, by enhancing the bit confusion and diffusion of the subkey, which is output that is produced from the key schedule transformation. The subkeys produced by the proposed approach have been proven to be a better result on both properties compared to the subkeys that were produced from Rijndael key scheduling transformation.

Keywords-component; Rijndael; Key Schedule; Proposed Approach of Key Schedule; Cryptography; Security

1 INTRODUCTION

Cryptography is a science and art of transforming messages to make them secure and immune to attacks [1]. There are three mechanism in cryptography; symmetric key, asymmetric key, and hashing. Symmetric key only use single key encryption and decryption while asymmetric key used two different key; public key to encrypt and private key to decrypt. Hashing is a message digest of fixed length. This cryptography mechanism was used in many applications such as bank cards, computer passwords, and electronic commerce that help to secure the use of technology which depends on the type of cryptography mechanism used.

Symmetric key mechanism was also known as other term; single-key encryption. The main drawback of this mechanism is the two parties must share the single key. There are two different schemes in symmetric key mechanism, which are either to use block cipher or stream cipher. In 1977, the first publicly available cryptographic algorithm which was adopted by National Bureau of Standards (now National Institute of Standards and Technology (NIST)) as Federal Information Processing Standard 46 (FIPS PUB 46) was Data Encryption Standard (DES) [2]. DES is a symmetric block cipher system that was widely used for more than two decades as encryption scheme by US federal agencies and private sector.

In 1997, NIST initiated a process to select a symmetric-key encryption algorithm that also implement block cipher scheme to replace DES as Advanced Encryption Standard (AES). NIST announced that fifteen out of twenty one of received algorithms have been selected as first candidates in First AES Candidate Conference in August 1998. After a year, in Second AES Candidate Conference, five out of fifteen were selected as finalist candidates; MARS, RC6, Rijndael, Serpent, and Twofish. In October 2000, in the Third AES Candidate Conference, Rijndael was announced by

NIST as the Advanced Encryption Standard. AES was published as FIPS 197 in December 2001 [3].

Technology advances in information technology and computer security have made everything including a cipher vulnerable and can be exploited to attack. There are many efforts that have been done to redesign and reconstruct AES block cipher with one objective, which is to improve the block cipher [4]. Thus, the rapid growth of computer technology and its resources may make this time shorter than NIST estimated time to break the algorithm [5].

Cryptanalysis is a new way of study to break a cipher compared to the exhaustive key search which was used as the basic technique to identify the correct key. The growth in the computer speed is always improving day by day and it is possible that in the near future, the safety of AES can be broken [6].

From analysis that has been made, the best public cryptanalysis for AES or Rijndael block cipher is the related-key attack [1]. Related- key attack was first introduced by Eli Biham [7]. This related-key attack examines the different between keys.

Study has been made and the result shows, among the AES candidates, Rijndael key schedule fall into a category in which knowledge of a round subkey yields bits of other round subkeys or the master key after some simple arithmetic operations or function inversions [8]. The Rijndael key schedule appears to be a more ad hoc design compared to cipher itself and it has much slower diffusion structure than the cipher and contains relatively few non-linear elements [7]. This is because of the fact that Rijndael block cipher has been attacked and exploited from the weaknesses found in the key schedule structure. Latest attack on Rijndael key schedule were improved by [9] on the impossible differential attack which reached up to 7-rounds for AES 128-bit key and also 8-round on 256-bit key compared to previous result by [10]. [9] also has successfully improved the time complexity of the differential attack on 7-round AES 192-bit by [11]. In 2009, there were two related-key attacks on the full round AES 192-bit and 256-bit the AES key schedule by [12].

Nevertheless, to enhance the Rijndael key schedule security, there are two significant properties to be focus on; confusion and diffusion. These are two properties of a secure cipher which were identified by Claude Shannon [13]. AES cipher algorithm managed to attain both of these properties, however in the key schedule; it is somewhat less rigorous in obtaining these properties [14]. An important theoretical foundation for bit confusion and bit diffusion is the idea of Frequency and Strict Avalanche Criterion (SAC) test, respectively [15]. The SAC obviates the need for a widely used approximation, allowing more accurate evaluation of the bit diffusion to key schedule [15] and the frequency test is to evaluate the confusion bit properties. Both of these properties shall be obtained in this research together with designing a new approach for Rijndael key schedule in order to enhance the security of the cipher.

2 PROCESS OF KEY SCHEDULING

Rijndael (new AES) block cipher has two part of transformations; cipher (round) and key schedule. Key schedule is an iterative component in a block cipher. A goal of a strong key schedule is to make the cipher to be resistant from various kinds of attacks. The key schedule has been studied for many years but there are many mathematical properties and weaknesses of this design which were insufficiently discovered in order to make the block cipher fully secured [15][16].

Key schedule is a transformation which uses master key (secret key) as an input value in algorithm to produce round keys (subkeys). The master key input can be 128-bit, 192-bit, or 256-bit key; however in this research 128-bit key is use as input. It is stated that 128-bit is the minimum requirement input in block cipher [17]. Rijndael key schedule involves three different byte-oriented transformation in each round; RotWord, SubBytes and RCon.

2.1 Rijndael Key Schedule Process

The subkeys are derived from the cipher key (master key) using the key schedule algorithm. RotWord performs a one-byte circular left shift on input (e.g., [a, b, c, d]) which was taken from the rightmost input of the master key (Fig. 1). The process of RotWord will produce an output word (e.g., [b, c, d, a]). This process of RotWord is as illustrated in Fig. 2. SubByte performs a function that returns a 4-byte word in which each byte is the result of applying the Rijndael S-box to the byte at the corresponding position in the input word, which is the result from RotWord function. Continuing from the example given previously, Fig. 3 shows that the SubByte process will take the output produced from RotWord process (e.g., [b, c, d, a]) and produced a new output (e.g., [k, m, p, t]). RCon is a 4-byte value in which the rightmost three bytes are always zero. The input word (input from the leftmost column in master key) will be exclusive OR (XOR) with the result from SubByte and also XOR with RCon input. Fig. 4 shows the process of RCon where the leftmost column in master key (e.g., [e, f, g, h]) is XOR with the output from SubByte (e.g., [k, m, p, t]) and input from RCon (e.g., [s, 0, 0, 0]), which will produce the output (e.g., [v, x, y, z]).

This output (v, x, y, z) will be the first in the round. This process will be repeated until the output produces the same value as the input master key – 128-bit key. The illustration shown in Fig. 5 is the summary of Rijndael key schedule transformation that includes the all processes and examples of outputs.

e	j	h	a
f	k	m	b
g	l	r	c
h	m	f	d

Figure 1. Example of master key input

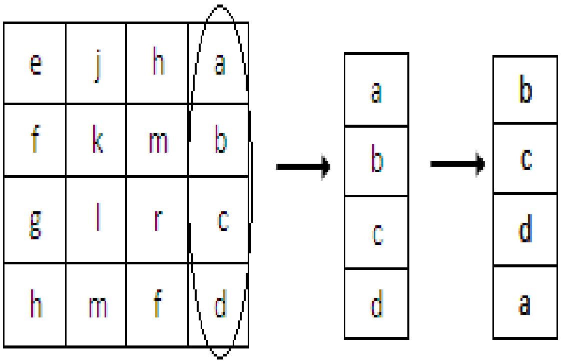

Figure 2. Illustration of RotWord process.

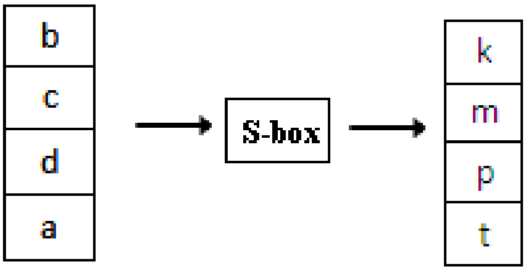

Figure 3. Illustration of SubByte process.

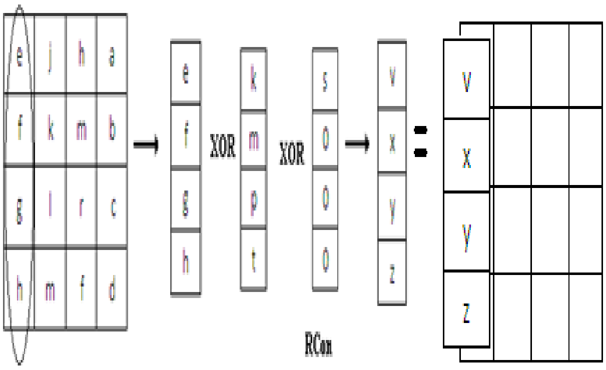

Figure 4. Illustration of RCon process.

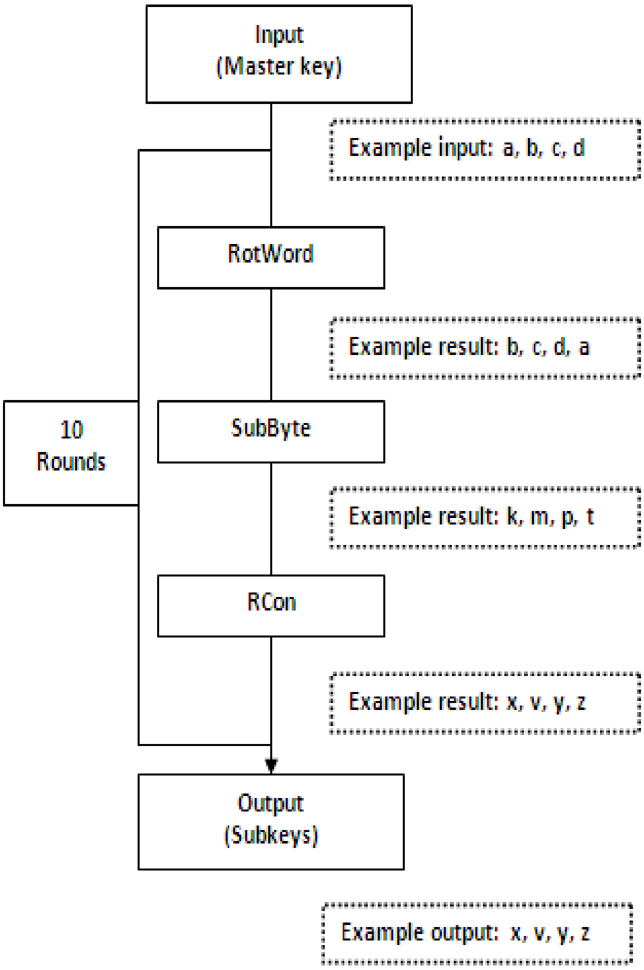
Input
(Master key)
Example input: a, b, c, d
RotWord
Example result: b, c, d, a
10
Rounds
SubByte
Example result: k, m, p, t
RCon
Example result: x, v, y, z
Output
(Subkeys)
Example output: x, v, y, z

Figure 5. Summary of Rijndael key schedule process.

2.2 The Proposed Approach Key Schedule Process

In our proposed approach, one new transformation, ShiftColumn, is added in the key expansion algorithm and the new algorithm is the enhanced version of Rijndael key schedule algorithm.

ShiftColumn operates by shifting columns with different offsets. ShiftColumn is one extra transformation added to the algorithm and the process is adapted from ShiftRow in cipher transformation of Rijndael block cipher but it is more complex than the ShiftRow process, where the proposed approach contains shift column, bitwise and cyclic shift.

The proposed approach involves left shifting the bit value in the column, and then the value is XOR within the same column but with different row. Next, the bit value in the column will be shifted to the right. Lastly, the whole column (one selected column) will be shifted with different offset. For example the result from RCon (e.g., [v, x, y, z]) will be inputted into the ShiftColumn process and the process will produce an output (e.g., [f, j, r, u]), which will also become the first word in the round for the proposed key schedule and all the functions (RotWord, SubByte, RCon, ShiftColumn) will be repeated again until finished all the key schedule transformation. The result for full key schedule transformation (128 bit) will produce 10 subkeys and each subkey will contains 4 words. The output subkeys will be use for evaluation to obtain bit confusion and bit diffusion properties. The illustration for the ShiftColumn is shown in Fig. 6. The process of the proposed approach which includes all the processes (RotWord, SubByte, RCon and ShiftColumn) is summarized as illustrated in Fig. 7.

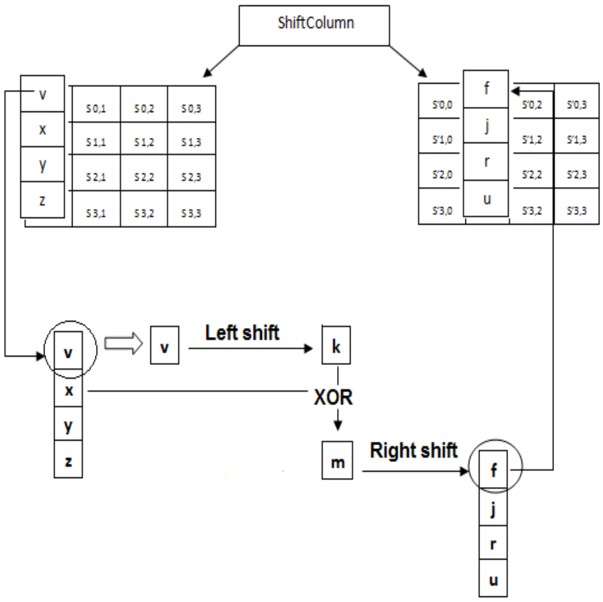

Figure 6. Illustration of the proposed approach transformation (ShiftColumn).

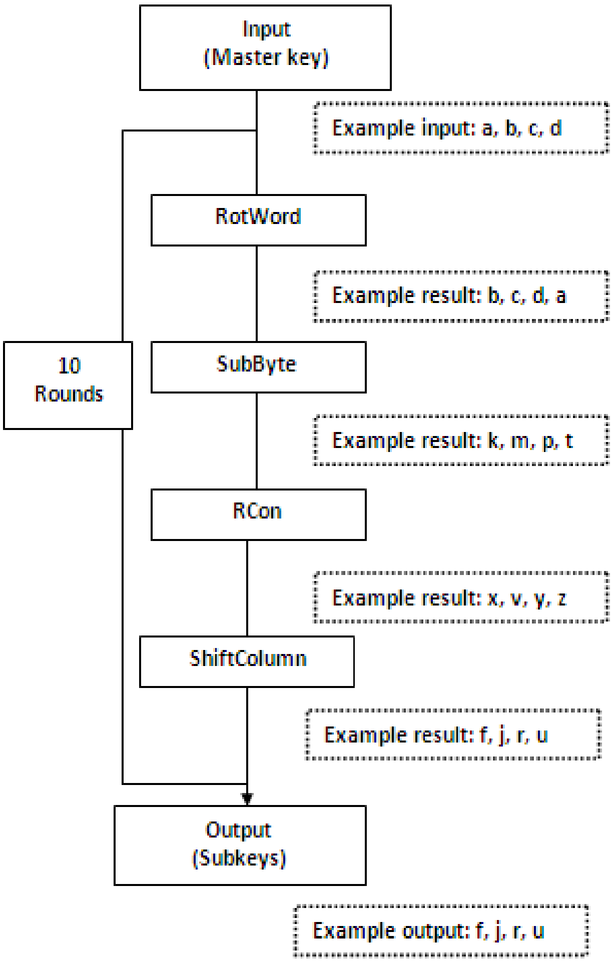
Input
(Master key)
Example input: a, b, c, d
RotWord
Example result: b, c, d, a
10
Rounds
SubByte
Example result: k, m, p, t
RCon
Example result: x, v, y, z
ShiftColumn
Example result: f, j, r, u
Output
(Subkeys)
Example output: f, j, r, u

Figure 7. Key schedule process of the proposed approach.

3 TEST

Confusion and diffusion are two properties of the operation of a secure cipher [12]. The frequency test was performed to measure the bit mixing property where it is a basic measure which is fundamental in achieving confusion property and SAC test was performed as a measure of the bit diffusion property that checks one bit change in the input, on average, changes to half the bits in the output [13]. Both tests will be measured by the probability value (p-value).

The frequency test is performed using NIST Statistical Test (from NIST test package) that focus on proportion of zeroes and ones with the purpose to determine whether the number of zeroes and ones are approximately the same in the sequence as would be expected for a truly random [15]. The frequency test result is determined by p-value; if the computed p-value is below than 0.01, then it can be concluded that the sequence is non-random or otherwise it can be concluded as sequence is random and satisfies at the 0.01% critical level [18].

SAC test is generated by using the SPSS software through one-sample kolmogorov-smirnov test (1-sample K-S test). Decision rule for this research is that if the p-value is more than 0.05, then we will accept the null hypothesis, if otherwise, we will then reject the null hypothesis and accept the alternate hypothesis. Null hypothesis indicate that the bit diffusion is satisfied at the 0.05% critical level.

4 DISCUSSION

The experiment was conducted using 20 subkeys as input for two compulsory tests in achieving confusion and diffusion properties; SAC test (one-sample kolmogorov-smirnov – poisson distribution) and frequency test. Critical values are assigned for both of the tests. This subkeys were obtained from the output produced using the Rijndael key schedule transformation and also from the output of the proposed approach.

SAC test: Results show that the proposed approach of key schedule algorithm obtained a better result than Rijndael key schedule though the 3 subkeys for both approaches (Rijndael and proposed approach) have failed the test. The graph plotted in Fig. 8 shows that more than half of the subkeys, from the proposed approach yields higher p-value which also means higher in bit diffusion property that contribute to a secure cipher.

Frequency test: Fig. 9 shows the result of p-value from the frequency test, where two of the subkeys failed the test for both of the algorithms. This shows that the proposed approach get a higher bit confusion properties which contribute to a more secure algorithm of key schedule compared to Rijndael.

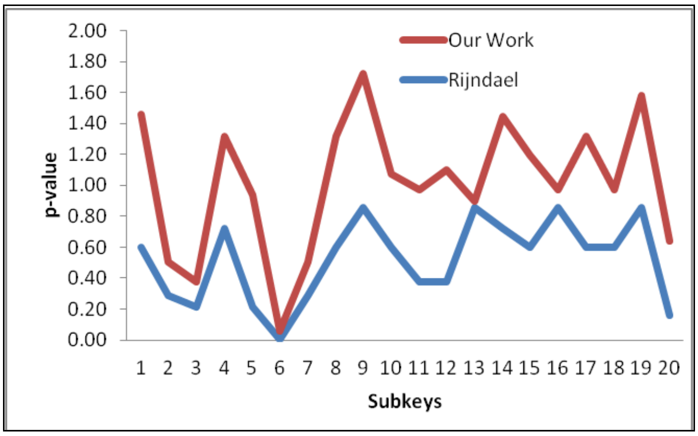

Figure 8. The result of SAC test.

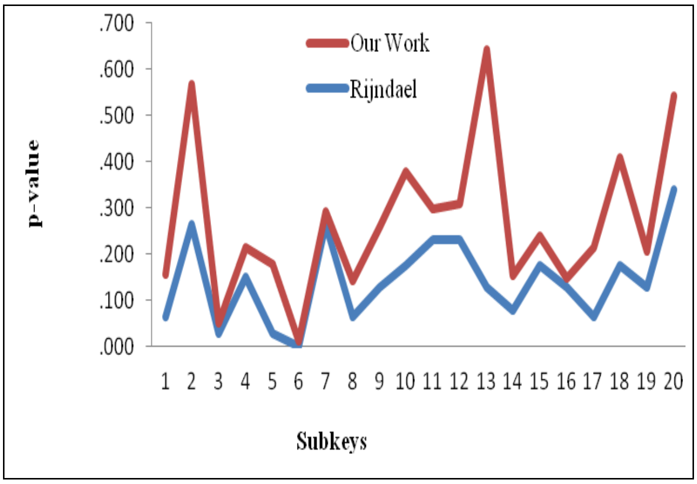

Figure 9. The result of frequency test.

5 CONCLUSION

This research focused on achieving bit confusion and diffusion on key schedule algorithm for the proposed approach using 128-bit key size. The analysis produced in this research is used to combat weaknesses in Rijndael key schedule algorithm. Fig. 8 and Fig. 9 shows comparison between the frequency test and SAC test results for both Rijndael key schedule and the proposed approach.

As a conclusion of the results, this research has achieved its objective. After analyzing both key schedule algorithms (Rijndael and proposed approach), somehow, the proposed approach shows better result in both test by achieving better results on both of the properties (confusion and diffusion).

For future enhancement, cryptanalysis attack can be performed on the proposed approach as part of the evaluation test. The result from the cryptanalysis attack will help in permitting in subversion or evasion.

6 REFERENCES

[1] Settia, N.: Cryptanalysis of Modern Cryptographic Algorithms. In International Journal of Computer Science and Technology, 1(2), pp. 166-169 (2010).

[2] Wright M. A.: The evolution of the Advanced Encryption Standard. Network Security, vol. 1999, pp. 11-14 (1999).

[3] Jamil, T.: The Rijndael Algorithm. In Potentials, IEEE, 23(2), pp. 36-38(2004).

[4] Juremi, J., Mahmod, R., and Sulaiman, S.: A Proposal for Improving AES S-Box with Rotation and Key-Dependent. In Cyber Security, Cyber Warfare and Digital Forensic (CyberSec), 2012 International Conference , pp. 38-42 (2012).

[5] Ali, S.A.:Improving the Randomness of Output Sequence for the Advanced Encryption Standard Cryptographic Algorithm. Universiti Putra Malaysia (2005).

[6] Jing, M-H., Chen, J-H., and Chen, Z-H.: Diversified Mixcolumn Transformation of AES. In Information, Communications & Signal Processing, 2007 on 6th International Conference, pp. 1-3 (2007).

[7] Ferguson N., et al.: Improved cryptanalysis of Rijndael. In Fast Software Encryption. LNCS, vol. 1978, pp. 213-230. Springer Berlin/Heidelberg (2001).

[8] Carter, G., Dawson, E., and Nielsen, L.: Key Schedule Classification of the AES Candidates. In Proceedings of the end AES Conference, Rome, Italy, pp. 1-14 (1999).

[9] Wentao, Z., Wu, W., and Dengguo, F.: New Results on Impossible Differential Cryptanalysis of Reduced AES. In Information Security and Cryptology - ICISC 2007. LNCS, vol. 4817, pp. 239-250. Springer Berlin/ Heidelberg (2007).

[10] Hee, C.J., Kim, M., Lee, J.Y., and Kang, S.W.: Improved Impossible Differential Cryptanalysis of Rijndael and Crypton. In Information Security and Cryptology — ICISC 2001. LNCS, vol. 2288, pp. 39-49. Springer Berlin/Heidelberg (2002).

[11] Phan, R. C.-W.: Impossible Differential Cryptanalysis of 7-Round Advanced Encryption Standard (AES). In Information Processing Letters, vol. 91, pp. 33-38 (2004).

[12] Biryukov, A., and Khovratovich, D.: Related-Key Cryptanalysis of the Full AES-192 and AES-256. In Advances in Cryptology – ASIACRYPT 2009. LNCS, vol. 5912, pp. 1-18. Springer Berlin/Heidelberg (2009).

[13] Shannon, C.: Communication Theory of Secrecy Systems. In Bell System Technical Journal, vol. 28, pp. 656-715, (1949).

[14] May, L., Henricksen, M., Millan, W., Carter, G., and Dawson, E.: Strengthening the Key Schedule of the AES.

In Information Security and Privacy. LNCS, vol. 2384, pp. 226-240. Springer Berlin/Heidelberg (2002).

[15] Muda, Z., Mahmod, R., and Sulong, M.R.: Key Transformation Approach for Rijndael Security. In Information Technology Journal, 9, pp. 290-297 (2010).

[16] Daemen, J., and Rijmen, V.: The First 10 Years of Advanced Encryption. In IEEE Security and Privacy, vol.8, pp. 72-74 (2010) .

[17] Nechvatal, J., Barker, E., Bassham, L., Burr, W., Dworkin, M., Foti, J., and Roback, E.: Report on the Development of the Advanced Encryption Standard (AES). Technical Report, NIST (2000).

[18] Rukhin, A., et al.: A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications. NIST Special Publication 800-22 (2001).