International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 3 (2015) pp. 8191-8202

http://www.ripublication.com

A Secure Auto-Key Round Combiner Algorithm

A.H.Kashmar[1, 2*] and E.S.Ismail[1]

[1]School of Mathematical, Sciences Faculty of Science and Technology, UniversitiKebangsaan Malaysia, 43600 UKM Bangi, Selangor DE, Malaysia
[2]University of Baghdad, College of Science, Baghdad, Iraq
[*]Corresponding Author: Kashmar992000@yahoo.dk

Abstract

In cryptography,the idea of using several implementations for encryption to obtain better security has been a research topic for several years. A combiner is a mathematical function that mixes two data sources into a single result. The term‘combiner style cipher’refers to a stream cipher.Combiner algorithms need to be secure to resist different types of attacks. Stream ciphers are an important class of symmetric encryption algorithms,and all stream ciphers generally usean ‘additive’combiner, such as exclusive-OR (XOR). Unfortunately, additive combiners have absolutely no strength at all. If an adversary somehow comes up with some given or guessed plaintext, and the matching ciphertext, an additive combiner immediately reveals the confusion sequence. This allows an adversary to recover some of the keystream masking material.To add complexity to the weak exclusive-OR combiner,this paper presents a new combiner algorithm suitable for stream ciphers, calledthe Auto-Key Round Combiner Algorithm (A-KRCA), which applies a nonlinear invertible round function to perform the encryption and decryption processes.We show that theA-KRCA was implemented and that its cryptanalysis demonstrates that the combiner works efficiently, providing absolute security and the necessary complexity against possible attacks.

Key words: Cryptography, stream cipher, S-box, Rijndael algorithm.

1. Introduction

As progressively more information is stored and transmitted in electronic form, the study of cryptography has become increasingly important. This has led to the creation of many encryption algorithms and standards. The Advanced Encryption Standard (AES) [8] is one of the most popular standards, and it is also known as Rijndael [6]. This standard is the cipher of choice for many official and commercial organisations around the world, and it is used in encrypting many forms of electronic data. The AES

uses byte substitution using a table called an S-box. Multiple S-box construction methods have been subsequently developed [1, 4]. The AES may also be implemented efficiently on smart cards [12],and the security of AES was demonstrated by its resistance to attacks when it was applied with ten or more rounds using a key of at least128 bits [3, 10,17]. Other encryption algorithms based on AES have also been developed by Ducet. al[7] and Ferguson et. al [9] gave a simple algebraic representation of Rijndael.

A combiner is the heart of a stream cipher, which generally uses an additive combiner, such as exclusive-OR (XOR). Different methods have been investigated for the design of combiners,for which we let $(x_1, x_2, \dots, x_m)$be a sequence of a binary data stream.For exampleCobas and Brugos[5] has contributed to the design of Boolean Combiners *C*,defined by a Boolean function $C{:}B^m \rightarrow B$, where $B=\{0,1\}$. Pichler[14] designed DynamicCombiners *C*(*t*), which varytheir functions by the clock time *t* of the generator, witha finite machine memory. Jochinger[11] focused on implementation of anFSM Combiner and applied it on CryptoBench. Sarkar [15] presented the Filter-Combiner model for memorylesssynchronous stream ciphers whereas Meier and Przydatek[13] designed Robust Combiners for cryptographic primitives. Armknecht[2] designed principles for combiners with a memory for stream ciphers. Teoet. al[16] presented an analysis of the mixer keystream generator.

Stream ciphers use XOR mixing as a combiner to performthe encryption and decryption processes. However, given some known or guessed plaintext, an adversary can use a plaintext attack to recover some of the pseudorandom masking material.To strengthen the weak XOR combiner, a new combiner algorithm is presented, called the Auto-Key Round Combiner Algorithm (A-KRCA), which uses a nonlinear invertible round function to perform the encryption and decryption processes. The implementation results show that the new combiner algorithm is reasonably efficient and moresecure than a conventional combiner that uses XOR for mixing.

The paper is organised as follows. In section 2, wedescribe the new combiner. We then investigate the relevant transformations in section 3. In Section 4, we describe the most significant combiner properties**.**We give a security analysis for possible attacks in section 5 and we end in section 6 with a few conclusions.

2. Description of the Combiner

The security of the proposed cipher combiner is dependent on having large alphabets in the combining process. The proposed cipher combiner employs a network of simple logic operations. In this new combiner, the substitution-permutation function is designed to replace dynamic substitution. The new combiner shares more than one character in the encryption/decryption process. Therefore, the combiner is built using a ‘round function’that provides the cryptographic strength. Specifically, it can be considered to use the Rijndael algorithm, which forms the basis of the new AES. A modified version of the Rijndael round is at the heart of the combiner function, which operates on 128-bit chunks. The input plaintext block is viewed as 4×4 matrixes of bytes called a state (see Figure 1).

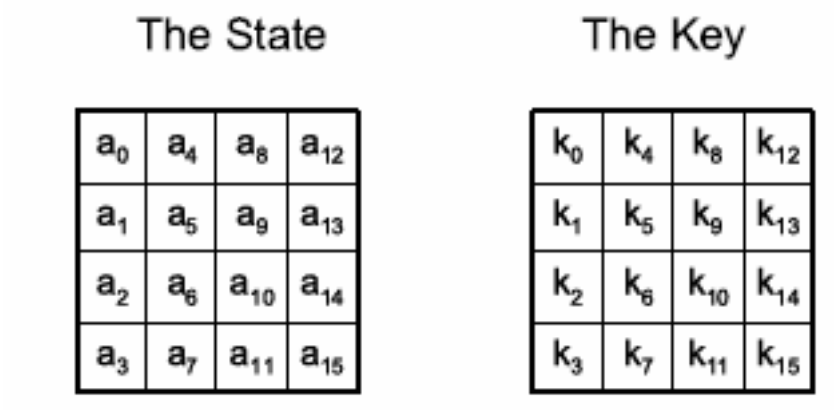


Figure 1: Structure of the state and the key

First, it is XORed with the 16-bytes of the keystream generated by a strong PRNG. Second, each byte is sent though an S-box, S [.]. Third, the three last rows in the matrix are shifted cyclically by a specific offset to the left. Finally, the resulting state is folded dynamically according to a random specification. A graphical description of the combiner (round function) is illustrated in Figure 2.

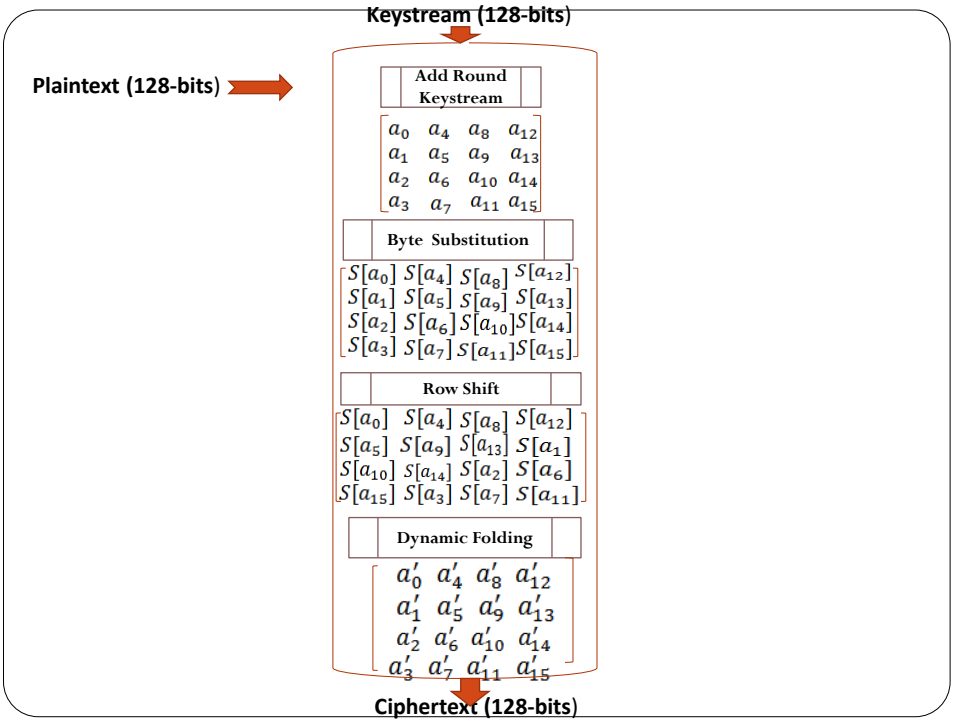


Figure 2: The graphical illustration of (AKRC)

The combiner is implemented efficiently. It consists of four different stages, which include two of permutations and two of substitutions. Add round key is a simple bitwise XOR of the current block with a keystream. Substitute bytes use an S-box to perform a byte substitution of the block. Shift row is a simple permutation. Dynamic fold includes a complex rote using simple operations. All operations make use of elements over GF(2^8), i.e., carried on a byte level. At the start/end of an encryption/decryption, the bytes of the cipher input/output are copied to/from the state array in column-by-column order. The forward function of the new combiner is used forthe encryption process,while the inverse function is used forthe decryption process.

3. Transformations Stages

For the new combiner algorithm, the length of the cipher key is 128 bits. Each round consists of four functions, which are listed below.

3.1 AddRoundKeystreamtransformations

The keystream generated from the PRNG is used byte-by-byte, from lowest to highest index, so there is no need for the keystream array to be in a 2-dimensional form; rather, it is possible to just use them up and move on. This implementation assumes that a counter is initialised to 0 each round. The function AddRoundKeystream uses 16 bytes of expanded key every time it is called (see Figure 3).The operation of the inverse AddRoundKeystream transformation is simply applied by performing the same forward transformation,because AddRoundKeystream is its own inverse (using the XOR operation).

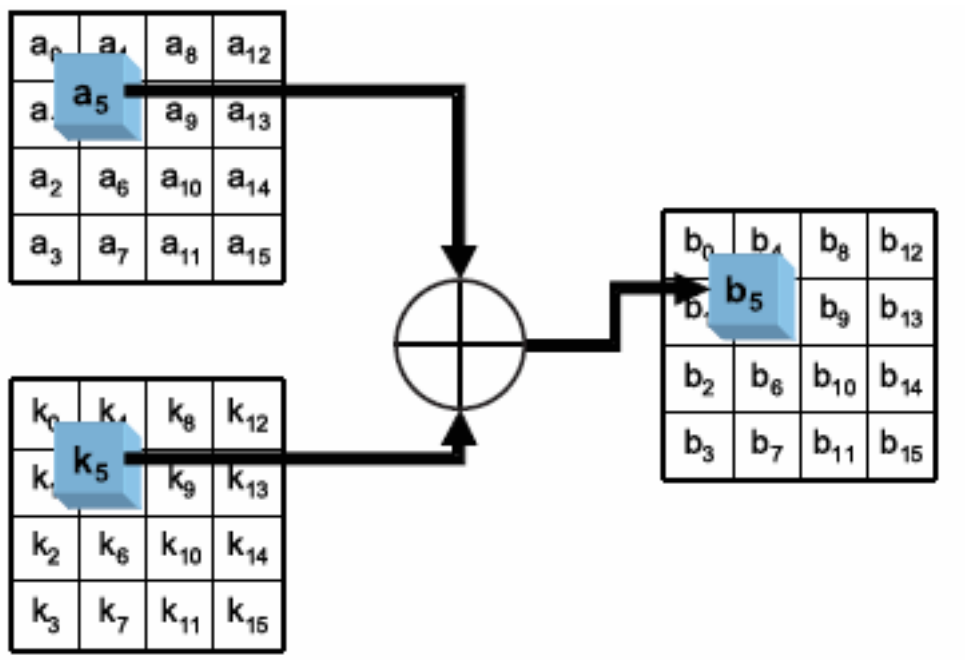


Figure 3: Addroundkey operation

3.2 Byte Substitution Transformations

The SubBytes transformation is a non-linear byte substitution that acts on every byte of the state in isolation to produce a new byte value using the S-box. The proposed algorithm is designed to have restrictions on the amount of ROM available. Thus,it is preferablefor the S-box to use a small amount of memory. Hence,an S-box containing only 256 entries is used.This S-box is a simple table lookup that containsa permeation of all possible 256 eight-values. Rijndael is designed to be resistant to known cryptanalytic attacks. The proposed algorithm design introduces only one keyed (secret) S-Box application as a good shuffler for bytes. The S-box [.] is derived from the Rijndael S-box in a key-dependent fashion. Because the S-box is private, an attacker no longer knows what is input to the S-box. A 32-bit key is used in the shuffle process. The secret S-box is initialised as follows:

$$S[i] = SR[\cdots SR[SR[i \oplus k_0] \oplus k_1] \oplus \ldots \oplus k_3], i = 0, \cdots, 255$$

whereSR [.] is a fixed Rijndael S-box, which is a one-dimensional representation of Table (1).

Table 1: The Substitution Table-S-box[x y] in hexadecimal

		Y															
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
X	**0**	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
	1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
	2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
	3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
	4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
	5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
	6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
	7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
	8	CD	0C	13	EC	5F	97	44	17	CA	A7	7E	3D	64	5D	19	73
	9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	08	D8
	A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	EA	79
	B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
	C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	48	BD	88	8A
	D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
	E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
	F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

The choice of the S-box provides a much stronger diffusion. Each output bit depends on each input bit of the S-Box key (see Fig. 4).

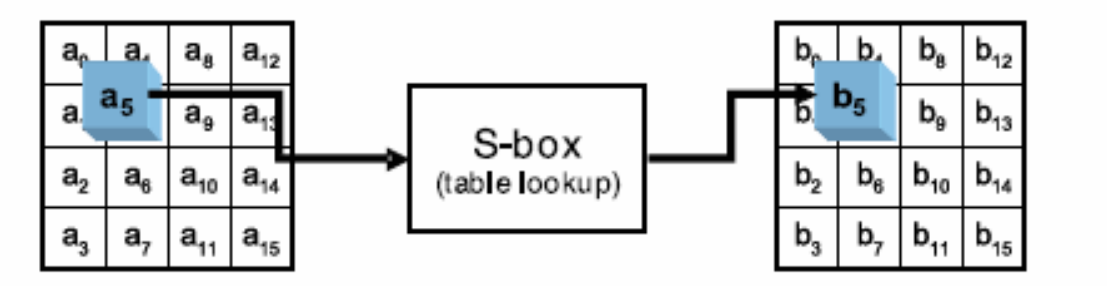


Figure 4: SubBytes operation

In the decryption process, the inverse SubBytes transformation is applied. The same transformation is implemented but with the use of the inverse substitution box. The inverse S-box is a one-dimensional representation of Table (2), which is found for the forward S-box as follows:

Table 2: The Inverse Substitution Table-InverseS-box[x y] in hexadecimal

		Y															
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
X	**0**	52	09	6A	D5	30	36	A5	38	BF	40	A3	9E	81	F3	D7	FB
	1	7C	E3	39	82	9B	2F	FF	87	34	BE	43	44	C4	DE	E9	CB
	2	54	7B	94	32	A6	C2	23	3D	EE	4C	95	08	42	FA	C3	4E
	3	08	2E	A1	66	2B	D9	24	B2	76	5B	A2	49	6D	8B	D1	25
	4	72	F8	F6	64	86	68	98	16	D4	A4	5C	CC	5D	65	B6	92
	5	6C	70	48	50	FD	ED	B9	DA	5E	15	46	57	A7	8D	9D	84
	6	90	D8	AB	00	8C	BC	D3	0A	F7	EA	58	05	B8	B3	45	06
	7	D0	2C	1E	8F	CA	3F	0F	02	C1	AF	BD	03	01	13	8A	6B
	8	3A	91	11	41	4F	67	DC	EA	97	F2	CF	CE	F0	B4	E6	73
	9	96	AC	74	22	E7	AD	35	85	E2	F9	37	E8	1C	75	DF	6E
	A	47	F1	1A	71	1D	29	C5	89	6F	B7	62	0E	AA	18	BE	1B
	B	FC	56	3E	4B	C6	D2	79	20	9A	DB	CD	FE	78	CD	5A	F4
	C	1F	DD	A8	33	88	07	C7	31	B1	12	10	59	27	80	EC	5F
	D	60	51	7F	A9	19	B5	4A	0D	2D	E5	7A	9F	93	C9	9C	EF
	E	A0	E0	3B	4D	AE	2A	F5	B0	C8	EB	BB	3C	83	53	99	61
	F	17	2B	04	7E	BA	77	D6	26	E1	69	14	63	55	21	0C	7D

3.3 Shift Rows Transformations

The action of shifting rows is particularly simple, and can be accomplished just by performing left circular shifts of rows 1, 2 and 3, by amounts of 1, 2, and 3 bytes, respectively. Row 0 is not changed (see Fig. 5).In the decryption process, the action of inverse shifting rows is particularly simple, and can be accomplished just by

performing right circular shifts of rows 1, 2, and 3, by amounts of 1,2, and 3 bytes. Row 0 is not changed.

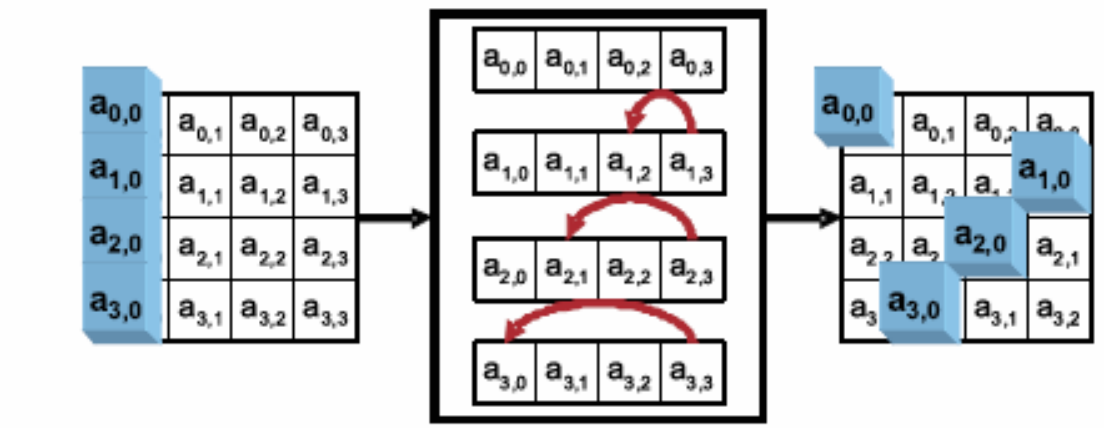


Figure 5: Shiftrows operation

3.4 Dynamic Folding Transformations

In this transformation, a complex rotation is applied to the state array by performing a dynamic permutation. In this stage, the elements of the state array are rearranged dynamically to new positions with higher probabilities than the normal arrangement. To perform the encryption/decryption process at the combiner, specific values are chosen from the keystream generated by the PRNG. The direction for the new permutation and the position to start are extracted from those values to implement the dynamic folding. This will be performed as follows:

a) The first byte of the keystream is ANDed with 0x01
If the result = 0, the Horizontal Direction is right
Else, the Horizontal Direction is left

b) The second byte of the keystream is ANDed with 0x01
If the result = 0, the Vertical Direction is up
Else, the Vertical Direction is down

c) The tried keystream is ANDed with 0x03
If the result = 0, the row is 0
If the result = 1, the row is 1
If the result = 2, the row is 2
If the result = 3, the row is 3

d) The fourth byte of the keystream is ANDed with 0x03
If the result = 0, the column is 0
If the result = 1, the column is 1

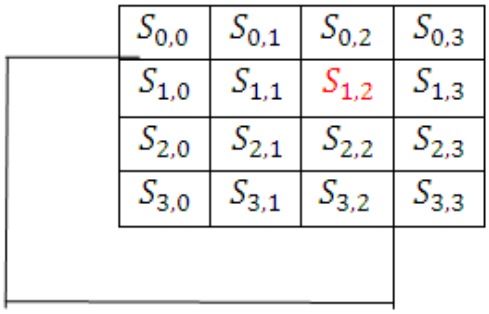
If the result = 2, the column is 2
If the result = 3, the column is 3

Suppose that the original state prior to dynamic folding is as follows:

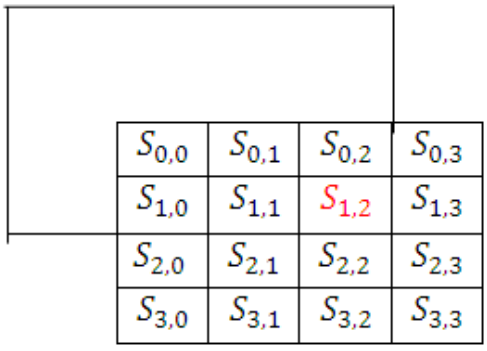
$S_{0,0}$	$S_{0,1}$	$S_{0,2}$	$s_{0,3}$
$S_{1,0}$	$S_{1,1}$	$S_{1,2}$	$S_{1,3}$
$S_{2,0}$	$S_{2,1}$	$S_{2,2}$	$S_{2,3}$
$S_{3,0}$	$S_{3,1}$	$S_{3,2}$	$S_{3,3}$

When selecting any position in this (4×4) array, there are 4 directions to rearrange the state array. Therefore, if the row is 1 and the column is 2, then the position to begin folding is (1, 2) with the possibility of using one of the four directions as follows:

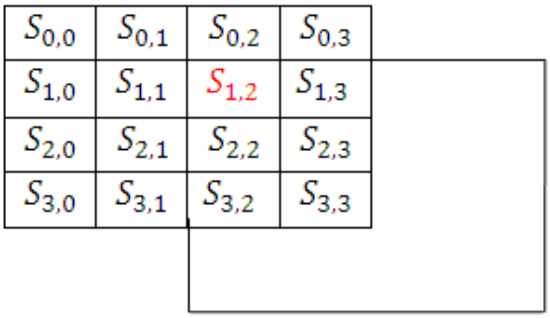
Left-down



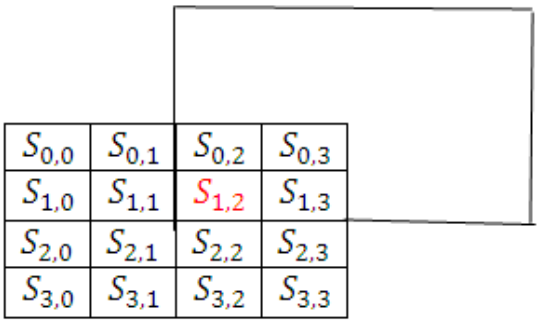
Left-up



Right-down



Right-up



According to the input keystream bytes, if the Horizontal Direction is 0 and the Vertical Direction is 0, then state array will be mutated dynamically to the Right-Up direction starting from position (1,2), and the state will be transformed to the following form:

$S_{2,2}$	$S_{2,3}$	$S_{2,0}$	$S_{2,1}$
$S_{3,2}$	$S_{3,3}$	$S_{3,0}$	$S_{3,1}$
$S_{0,2}$	$S_{0,3}$	$S_{0,0}$	$S_{0,1}$
$S_{1,2}$	$S_{1,3}$	$S_{1,0}$	$S_{1,1}$

This is used to further complicate the cryptanalysis. There are 16 locations in the (4 × 4) state array to start the new permutation according to a specific direction. Instead of using a fixed arrangement for the state, there are 64 (4 × 4) state arrays. Thus, the opponent cannot specify the correct order easily, and this will increase the difficulty. In the decryption process, the action of inverse dynamic folding is performed using the reverse of the forward dynamic folding by returning the folded state array to the original arrangement

4. The Significant Combiner Properties

ARijndael-like round function is employed in the proposed combiner because it has suitable properties [6]. This research considers the following algorithm:

a) The first operation is XOR; this ensures an absolute security to the plaintext, on the other hand the remaining transformations themselves will be a real complexity against the opponent.

b) Using a key-dependent S-box in the combiner ensures large difficulties. Because the S-box is unknown to some degree, it increases the number of probabilities faced by the attacker.

c) XOR represents uniquely the first point of meeting between the plaintext and the confusion sequence. Thus, the adversary cannot analyse the other parts that transmit the effect of bits among each other. In the proposed cipher combiner, the existence of dynamic folding encompasses both rows and columns and involves mutating elements depending on a specific issue. To improve its effectiveness, it increases probabilities against the cryptanalyst, because he cannot distinguish which 4 × 4 table to select among 64 (4 ×4) tables. A key-dependent operation is added to the main body of the combiner before outputting each block.

5 Security Analyses with Possible Attacks

a. Ciphertext Only: If the plaintext data thatwas input to the proposed combiner algorithm is randomized using the keystream sequence by exclusive-OR, then thesedata are treated by many transformations of the round function that include SubBytes,ShiftRows, and dynamic folding. Then, the letter frequency statistics will be concealed, and the result is random-like output.

b. Known Plaintext: When the attacker has some plaintext and the related ciphertext,then this attack when using exclusive-OR combining will expose the keystream. In the round function, each byte is XORed with the keystream and substituted using a keyed S-box,making the attacker unable to determine the plaintext byte. Also the dynamic folding according to the keystream selected ischanged dynamically upon each iteration. This forms a more sophisticated state when deciphering the ciphertext byte.

c. Statistical Attack: The nonlinear transformationof the combiner round function serves to erase the statistical weakness in the keystream. The keystream bit sequences must pass statistical tests.

d. Time-Memory Trade-Off attacks: This type of attack can be applied if the state space of the cipher is too small. These types of attacks do not seem to be applicable to the proposed algorithm.

e. Guess-and-Determine Attacks: The basic idea is to guess the value of some unknown variables in the system, and from the guessed values deduce the value of the other unknown variables. In this algorithm, the dynamic folding in the combiner is the result of the nonlinear mapping in the 4 × 4 state arrays. Thus, this attack is invalid for the proposed algorithm.

f. DifferentialAnalysis: The basic concept is that the attacker exploits the characteristics of a known S-Box or transformations. Differential analysis is invalid to apply to the proposed algorithmbecause the employed tables are "keyed",i.e.,initialized by a particular key. This means that the attacker does not have any prior knowledge of a particular table arrangement.

g. Distinguishing and correlation attacks: The nonlinear combiner of the proposed algorithm has been defined to use an essential amount of inputs (e.g., the shuffled(key dependent) S-box to perform a strong transformations), which makes this type of attack inapplicable.

6. Conclusion

A new Auto-Key Round Combiner Algorithm was presented. The algorithm, A-KRCA, uses a nonlinear invertible round function to perform the encryption and decryption process, which extends the weak classical concept of a simple XOR combiner into a stronger form, which is suitable for computer cryptography.All operations make use of elements over $GF(2^8)$, i.e., carried on a byte level. At the start/end of an encryption/decryption the bytes of the cipher input/output are copied to/from the state array in column-by-column order. The cryptanalysis ofA-KRCAshows that the combiner is implemented efficiently,which provides absolute security and real complexity against possible attacks. In addition, as a further safeguard, there are large difficulties that increasethe number of probabilities against the cryptanalyst.For instance, the huge number of possible keys makes a brute-force attack on A-KRCA impossible.

REFERENCES

[1] Abuelyman, E.S. and M.A. El-Affendi, 2007. A real time S-box construction using arithmetic modulo prime numbers. J. Digital Inform. Manage, 5: 354-260.

[2] Armknecht, F., Krause, M. and Stegemann, D. 2005. Design principles for combiners with memory. Progress in Cryptology - INDOCRYPT2005, volume 3797 of Lecture Notes in Computer Science, pages 104{117.Springer-Verlag, 2005.

[3] Biham, E., O. Dunkelman and N. Keller, 2006. Relatedkey impossible differential attacks on 8-round AES-192. Lecture Notes Computer Science. 3860: 21-33.

[4] Chen, G., 2008. A novel Heuristic Method for Obtaining S-boxes.Journal of Computer Science, JCSSP. 4(12): 999-1002.

[5] Cobas, J. D. G., and Brugos, A. L. 2005. Complexity-Theoretical Approaches to Design and Analysis of Cryptographical Boolean Functions. In Computer Aided Systems Theory-EUROCAST 2005, Springer Verlag, Lecture Notes in Computer Science, LNCS 3643,2005, pp.337-345.

[6] Daemen, J., and Rijmen, V. 2002. The Design of Rijndael, AES - The Advanced Encryption Standard, Springer-Verlag, 2002.

[7] Duc, D.A., T.M. Triet and L.H. Co, 2002. The extended rijndael-like block ciphers. Proceeding of the International Conference on Information Technology, Coding and Computing, Apr. 8-10, IEEE Xpolre, USA. pp.: 183-188.

[8] Federal Information Processing Standards (FIPS 197), 2001. TheAdvanced Encryption Standard.http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[9] Ferguson, N., Schroepple, R. and Whiting, D.2001,Asimple algebraic representation of Rijndael, SAC 2001,LNCS #2259,pp.103-111,Springer Verlag,2001.

[10] Jakimoski, G. and Y. Desmedt, 2004. Related-key differential cryptanalysis of 192-bit key AES variants. Lecture Notes Computer Science, 3006: 208-221.

[11] Jochinger, D. 2006. Implementation of a FSM Combiner and Testing with CryptoBench 2006. Internal Report, Kukla Electronics, August 2006.

[12] Lu, C.F., Y.S. Kao, H.L. Chiang and C.H. Yang, 2004. Fast implementation of AES cryptographic algorithms in smart cards.Preceding of the IEEE 37th Annual International Carnahan Conference on Security Technology, Oct. 14-16, IEEE Xpolre. USA. pp.: 573-57

[13] Meier, R. and Przydatek, B. 2006. On Robust Combiners for Private Information Retrieval and other Primitives. In Advances in Cryptology - CRYPTO ’06, pages 555–569, 2006.

[14] Pichler, F. 2007, A highly nonlinear cellular FSM-combiner for stream ciphers Lecture presented at EUROCAST 2007, Las Palmas, February, 2007, Spain.

[15] Sarkar, P. 2002.The Filter-Combiner Model for Memoryless Synchronous Stream CiphersCRYPTO 2002, LNCS 2442, pp. 533–548, 2002.Springer-Verlag Berlin Heidelberg 2002.

[16] Teo1, Sui-Guan.,Wong1,Kenneth Koon-Ho., Simpson1,Leonie. and Ed Dawson1.2012. State convergence and keyspace reduction of the mixer stream cipher. Journal of Discrete Mathematical Sciences and Cryptography, 15, pp. 89-104

[17] Zhang, W.T., W.L. Wu and L. Zhang, 2007.Related-key impossible differential attacks on reduced-round AES-256. J. Software, 18: 2893-2901.