URESC, Unbalanced Rotor Enhanced Symmetric Cipher

H. ElKamchouchi and A. ElShafee

Members IEEE
E-mail : helhamchouchi@ieee.org, ashafee@ieee.org

ABSTRACT

URESC is a new cryptosystem, stands for Unbalanced Rotor Enhancing Symmetric Cipher. It uses main cryptographic terms of substitution and permutation, and key dependency back-to-back with unbalanced rotor to achieve the requested goals of the design like perfect statistics and large period. Each wheel of unbalanced rotor contains different number of elements, which are pre-selected of prime number greater than 256 and less than 512. Rotor wheels are implemented using modulo 2^{72} operations. URESC has 64 bits basic block, substitutions are performed on each single basic block at a time. The resulted block of rotor called expanded basic block, of 72 bits. Two types of permutations are used horizontal and vertical. Finally a key dependency operation is applied. The key of each round is generated using a small rotor called key-rotor. URESC achieves memory-less, normalized ciphertext statistics, and small processing speed trend.

Keywords: *Cryptography, Coercive force, Encoding.*

1. INTRODUCTION

In order to develop a new cryptosystem, there are three guide lines, which must be taken into consideration while building block ciphers. These are substitution, permutation which were defined by the cryptography godfather Shannon [1] in the middle of the twentieth century, then key dependency to achieve privacy of cryptosystem per user. All successfully known cryptosystems, follows the same rules with little bits of variations depending on designer point of view. For example Rivest [2] used rotation, with their rotating order depending on the encrypted data itself. RIJNDAEL [3] used simple mathematical operations to achieve ordinary permutation and substitution. In other lately published cryptosystems REBC [4], KAMFEE [5], CYCLONE [6], ROTRIX [7], and REBC2 [8], rotors are always used. In each of these cryptosystem rotors are being used with unique concept, and different methodology. In our new proposed cryptosystem unbalanced rotors are being used. The new rotor is described as unbalanced because each wheel has a different number of items, a different length, and of course a different stepping length. Each wheel has its own field as each wheel has its own prime numbers or elements. Beside usage of rotor the basic operations; substitution, permutation, and key dependency are presented.

2. URESC STRUCTURE

2.1 Key generation

Round key is generated using a key-rotor of 8 cylinders; each rotor contains 256 elements presenting all the ASCII characters. The first key is generated from the user key; the second key is generated from the first key, and so on till the end of plaintext. Considering K_r to present the encryption key of round r, and K_{r-1} is to present the encryption key of round r-1, Fig. 1, shows the key generation process.

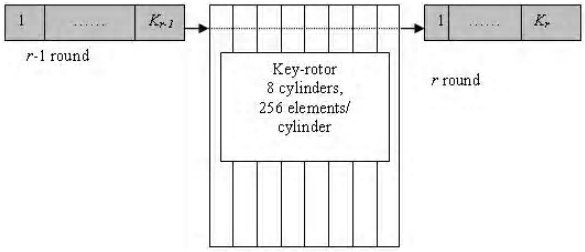

Fig. 1: Round *r* key generation

2.2 URESC Block Length and Key Length

URESC has three different forms, depending on its block length. Each level has corresponding different key length, rounds, and rotor wheels. The following TABLE I summarizes these levels.

Table 1: Uresc Types

Block length (bytes)	Output block length (bytes)	Number of rounds	Generated key length (bytes)	Number of rotor wheels
8	9	4	74	8
16	18	8	288	16
32	36	16	1120	32

2.3 URESC Basic Block

URESC has 8 bytes basic block. All basic operations like substitution, permutation and key dependency depend on this block basis. Thus URESC-8 has a single basic block, URESC-16 has two basic blocks, and URESC-32 has four basic blocks.

2.4 URESC Encryption

URESC uses three basic operations: substitution, permutation, and key dependency. These operations are basic terms as in any cryptosystems. . Fig. (2 to 4), shows URESC forms.

2.5 URESC Substitution

Substitution uses modulo 2^{64} multiplication and addition, as given by (1).

$$y = (a \times x) + b \bmod 2^{64} \quad (1)$$

Where; b is a generated round key

a :is a preselected value, having a valid $a^{-1} \bmod 2^{64}$.

x : presents plaintext basic block.

y: presents ciphetext basic block.

2.6 URESC Permutation

There are two types of permutation, horizontal and vertical. For URESC-8 input block will be arranged as an array of 1×8, URESC-16 will be arranged of as an array of 2×8, and URESC-32 will be arranged as an array of 4×8, as shown in Fig. (5 to 7). Permutation order is saved in an array of eight elements, randomly selected (0 to 7). The 1st permutation process uses 1st order of the array, 2nd permutation process uses 2nd order of the array, and so on, and the 9th permutation process uses 1st order of the array gain and so on. The vertical permutation is used in URESC-16 and URESC-32, as shown in Fig. (8, 9).

2.7 URESC Key Dependency

The key dependency process uses 264 modulo additions, using the equation (2).

$$y = x \otimes k \bmod 2^{64} \quad (2)$$

Where, y: represents cipher text basic block.

x: represents plaintext basic block.

k: represents key basic block.

2.8 URESC Rotor

Rotor is used in the middle of the encryption process. each wheel of the rotor has a unique primary number of elements greater than 2^8 and less than 2^9. The rotor is implemented using a linear affine transformation as shown in equation (3).

$$y_i = (p_i \times x_i) + (b_i + n_i) \bmod P_i \quad (3)$$

Where, y_i: represents ciphertext character of order i.

P_i: represents a random value less than 29 presenting the rotor's wheel i core.

x_i: represents the plaintext character of order i.

b_i: represents key character of order i.

p_i: represents number of elements inside the rotor.

n_i: represents the rotor rotation order.

After each encryption process, the rotor rotation order is incremented by one (n_i +1), and after the rotor

completes its rotations ($n_i = P_i$), the next rotor rotation order is incremented by one (n_{i+1} +1).

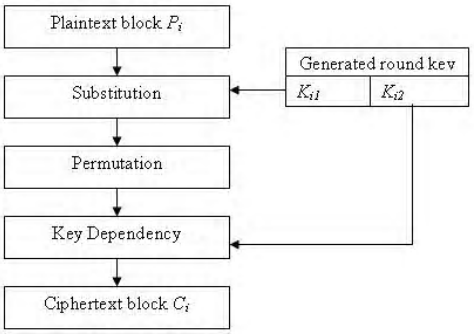

Fig. 2: URESC-8 single roud structure

2.9 The Structure of Rounds Follows the Rotor

As noticed, each byte encrypted by the rotor becomes nine bytes, so the basic block encrypted by the rotor becomes nine bytes instead of eight bytes, which is called the expanded basic block. The substitution process of expanded block is given by (4).

$$y = (a \times x) + b \bmod 2^{72} \quad (4)$$

To excute permutation process on expanded blocks, blocks will arranged as an array of 9 columns and x rows based upon used URESC form, and the permutaion order saved in expanded permutation order of 9 elements contains the numbers of 0 to 8 randomly arranged. Fig. (10 to 12) shows URESC different forms general structure.

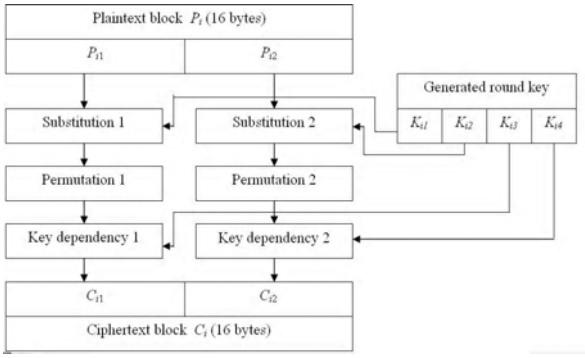

Fig. 3: URESC-16 single round structure

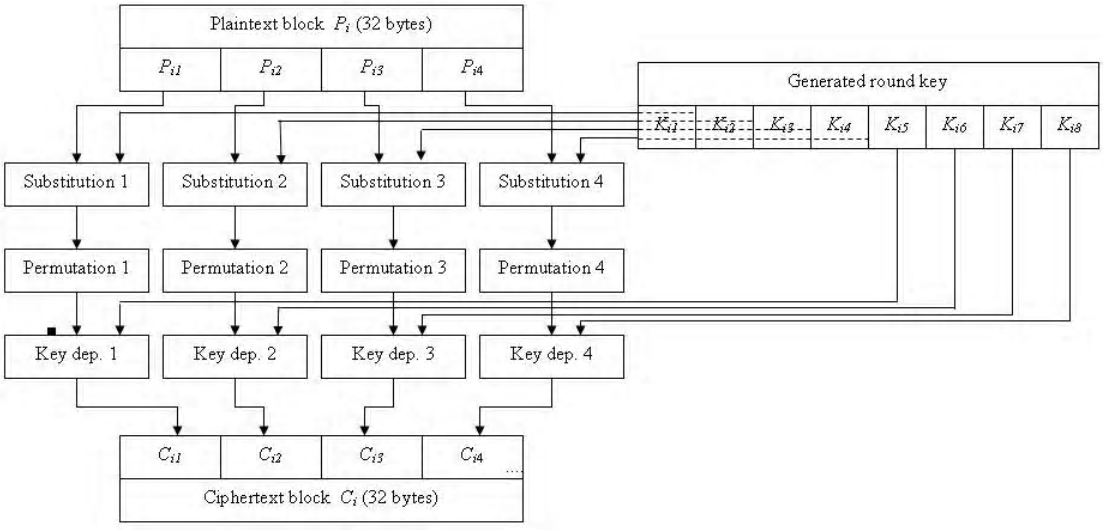

Fig. 4: URESC-32 single round structure

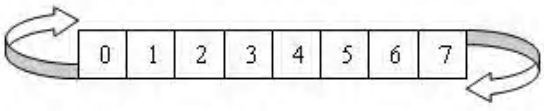

Fig. 5: URESC-8 horizontal permutation

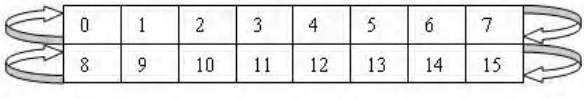

Fig. 6: URESC-16 horizontal permutation

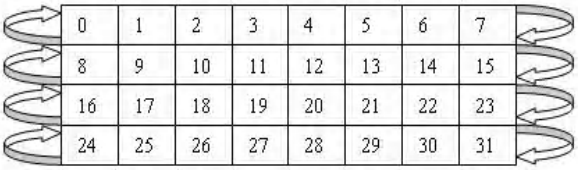

Fig. 7: URESC-32 horizontal permutation

Fig. 8: URESC-16 vertical permutation

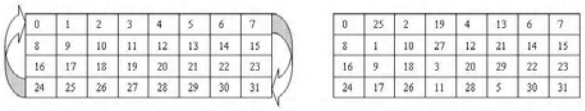

Fig. 9: URESC-32 vertical structure

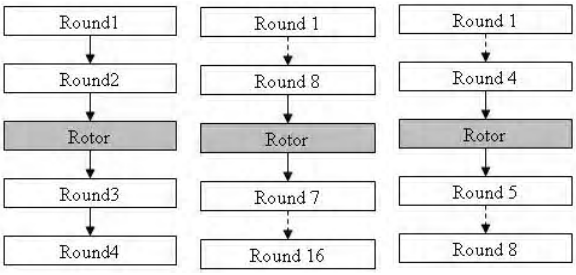

Fig. 10, 11, 12: URESC-8, 16, 32 overall structures

3. URESC FEATURES

3.1 Static Memory Requirements (secret data)

Table 2: static memory requirement for uresc-32

Element Name	#	Memory requirements
Key rotor	1	8 × 2 = 16 bytes
Substitution 8 bytes basic block	1	8 × 64 = 512 bytes
Horizontal permutation	1	32 bytes
Vertical permutation	1	8 bytes
Rotor	1	288 bits
Substitution (expanded basic blocks)		4608 bits
Horizontal permutation (expanded basic blocks)		36 bytes
Vertical permutation (expanded basic blocks)		9 bytes
Total		1225 bytes ≈ 1.2 kb

3.2 URESC Dynamic memory requirements

Table 3: dynamic memory requirement for uresc-32

Element Name	#	Memory requirements
Substitution	1	36 bytes
Horizontal Permutation	1	36 bytes
Vertical permutation	1	36 bytes
Rotor		36 bytes
Block	1	36 bytes
Generated key	1	36 bytes
Total		216 bytes

3.3 URESC period

The period of URESC is the product of four elements. Which are rotor period, substitution of 8 bytes basic block, substitution of 9 bytes expanded basic block, and the key. The following TABLE IV shows URESC period.

3.4 URESC Operation Modes

Operation modes are developed for block ciphers such as DES [9], GOST [9], RIJNDAEL [3], etc.. URESC period is very large as shown in TABLE III, URESC needs no modes of operations.

3.5 URES brute force attack

P: states for permutation operation

Considering $\phi(2^n)$ as the number of integers that are less than 2^n and has an inverse modulo 2^n, which equals to (2^n-2^{n-1}), or $(2^n/2)$.

Table 4: uresc-32 period

Element	Period
Rotor	$(2^{72})^{32} \approx 3.7 \times 10^{693}$
Substitution of 8 bytes basic block	64
Substitution of 9 bytes basic block	64
Key	32
Total	4.9×10^{698}

Table 5: uresc-32 brute force attack

Secret Data	Brute Force attack
User Key	$(32)^{256} \approx 2$ E 385
Key rotor	$2 \times ({}^{256}P_8) \approx 3.3E19$
Substitution 8 bytes basic block	${}^{9.2\ E\ 18}P_{64}$
Substitution 9 bytes basic block	${}^{2.4\ E\ 21}P_{64}$
Total trials	$6.6E404 \times ({}^{9.2\ E\ 18}P_{64}) \times ({}^{2.4\ E\ 21}P_{64})$

3.6 URESC Ciphertext Statistics (Frequency Cryptanalysis)

URESC gives perfect ciphertext statistics since all characters gave approximately normal distribution. To clarify this a plaintext file is encrypted using URESC and Rijndael [3]. To test the period of URESC a special plaintext file containing a repeated character – delta plaintext file – is encrypted using both URESC and Rijndael. Fig. 13 to Fig. 16 show plaintext statistics, URESC ciphertext statistics, delta plaintext file statistics, and URESC ciphertext statistics of delta plaintext file. Fig. 17, and Fig. 18 show Rijndael ciphertext statistics, and Rijndael ciphertext statistics produced from delta plaintext file.

3.7 URESC cryptanalysis

Due to unbalanced rotor used in URESC and, rotating substitution, the differential [10], linear [11], and related key cryptanalysis become extremely difficult. Cryptanalyst faces many difficulties to produce a general linear expression between input plaintext and output ciphertext. The main problem is the unbalanced rotor, which expand the 64 bits input plaintext into 72 bits ciphertext.

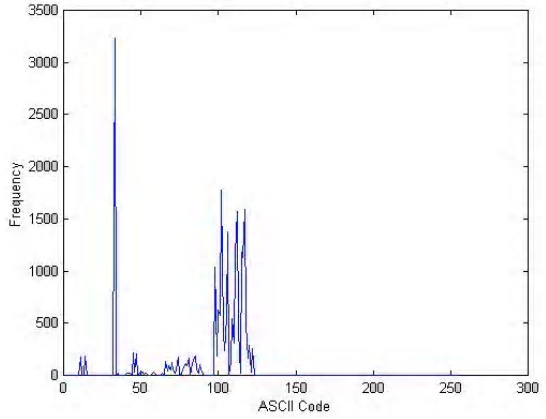

Fig. 13: Plaintext statistics

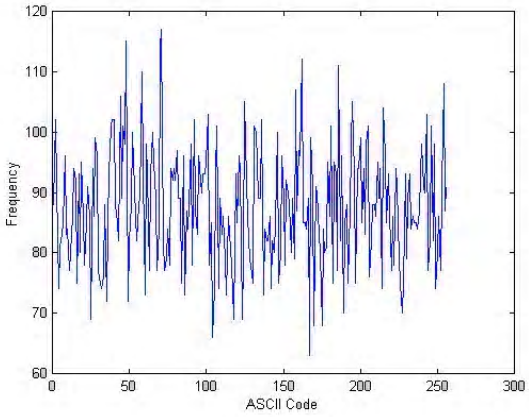

Fig. 14: URESC ciphertext statistics

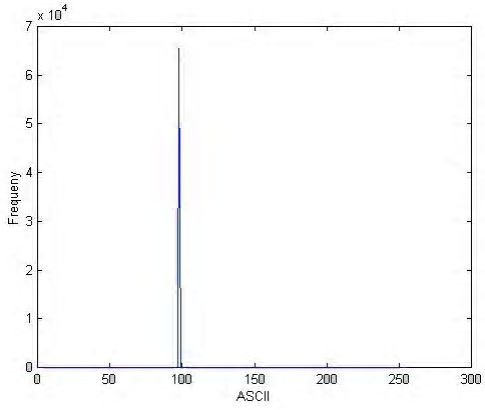

Fig. 15: Delta plaintext statistics

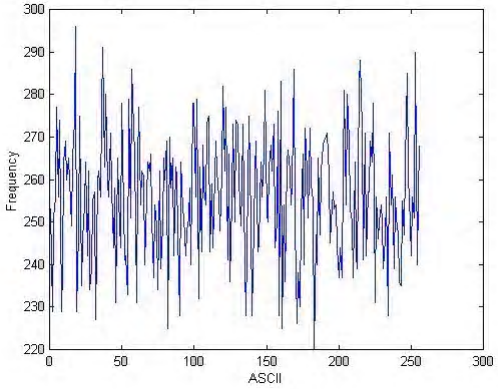

Fig. 16: URESC ciphertext statistics (for delta file)

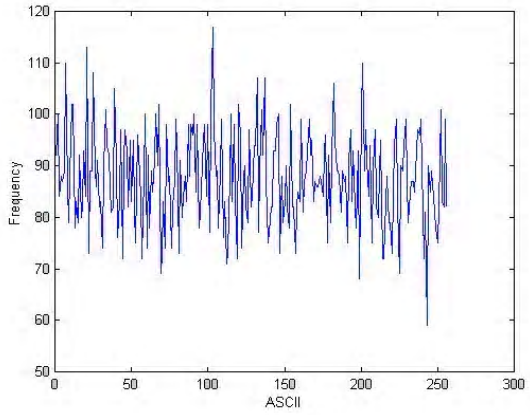

Fig. 17: Rijndael ciphertext statictics

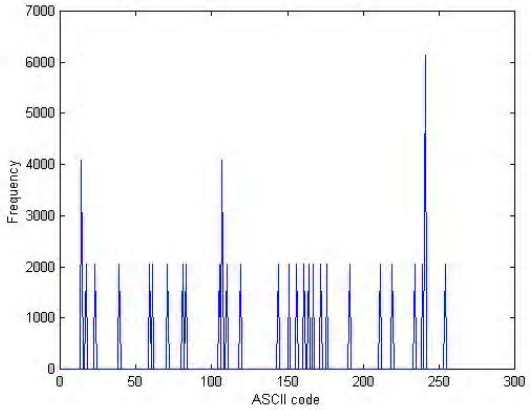

Fig. 18: Rijndael ciphertext statistics (for delta file)

3.8 Processing Speed

Software programs were developed for both Rijndael-256 and URESC then were tested on Pentium DUO, 1.83, 1.83 GHz, 1Gbytes Ram, and 2Mbytes cash under windows XP SP2. Software programs were written using C++ language and complied using Borland C++ 5.5 compiler. The following TABLE VI summarizes the results.

Table 6: processing Speed of Rijndael and uresc

Cryptosystem	Average Processing speed
Rijndael-32	215 Kbytes/Sec
URESC	68 Kbytes/Sec

4. CONCLUSIONS

The proposed URESC uses THE conventional main cryptographic terms of substitution and permutation, and key dependency back-to-back with unbalanced rotor to achieve the requested goals of like-perfect-statistics, large period, and resistance to linear and differential cryptanalysis. URESC uses a small static and dynamic memory to run in less than required memory corresponding to Rijndael. URESC processing speed is greater than Rijndael, as it makes use of modern 64 bites processors to enhance the processing speed. URESC period make it suitable for encrypting huge messages without the need of operation modes.

REFERENCES

1. C.Shannon, "*Conication Theory of Secrecy Systems*", Bell System Tech.. J., Vol. 28, 1949.
2. Rivest: Ronald L. Rivest, "*The RC5 Encryption Algorithm*", document made available by FTP and World Wide Web, 1994.
3. J. Daemen, J. T. Rijmen, "*AES Proposal: RIJNDAEL*". AES Algorithm Submission, 1999.
4. Elkamchouchi, H.M.; Elshafee, A.M., "*REBC, Rotor Enhanced Block Cipher*"; Radio Science Conference, 2002. (NRSC 2002). Proceedings of the Nineteenth National Radio Science Conference, 19-21 March 2002 Page(s):262 - 269. Digital Object Identifier 10.1109/NRSC.2002.1022631
5. Elkamchouchi, H.M.; Elshafee, A.M., "*Dynamically Key-controlled Symmetric Block Cipher KAMFEE*"; Radio Science Conference, 2003. NRSC 2003. Proceedings of the Twentieth National, 18-20 March 2003 Page(s):C19 - 1-12, Digital Object Identifier 10.1109/ NRSC.2003.1217353
6. ElKamchouchi, H.; ElShafee, A., "*Cyclone, the two Dimensional Rotor, Rotor's New Generation*"; Radio Science Conference, 2005. NRSC 2005. Proceedings of the Twenty-Second National, March 15-17, 2005 Page(s):269 – 276.
7. ElKamchouchi, H.; ElShafee, A., "*RotRix, The Arrayed Rotors*"; Radio Science Conference, 2006. NRSC 2006. Proceedings of the Twenty-Third National Radio Science Conference.
8. ElKamchouchi, H.; ElShafee, A., "*REBC2*"; Radio Science Conference, 2007. NRSC 2007. Proceedings of the Twenty-Fourth National Radio Science Conference.
9. *Applied Cryptography*, Second Edition: Protocols, Algorthms, and Source Code in C by Bruce Schneier. Wiley Computer Publishing, John Wiley & Sons, Inc.

10. Eli Biham, Adi Shamir, (1990). *"Differential Cryptanalysis of DES-like Cryptosystems"*. Advances in Cryptology — CRYPTO '90. Springer-Verlag. 2–21.

11. Mitsuru Matsui: *"Linear Cryptanalysis Method for DES Cipher"*. EUROCRYPT 1993: pp386–397.