CRYPTOGRAPHIC PROTOCOL WITH A PROPOSED CIPHER AND APERIODIC KEY REPLACEMENT

Sivo Daskalov[1] and Milena Karova[2]

Abstract – **The proposed cryptographic protocol implements a cipher with block and key length of 2^n bits. The encryption algorithm consists of several stages each of which swaps or inverts different segments of the plaintext according to the current key. Each key is used a seemingly random number of times between 0 and 15, afterwards the next generated key is encrypted and transmitted using the previous one. The initial key is established using Public key encryption.**

Keywords – **Aperiodic, block cipher, cryptography, segmentation, symmetric key**

I. INTRODUCTION

For a cryptographic protocol to be secure, it is necessary that the cryptographic algorithms used perform well under the conditions of its particular application. Developers of cryptographic algorithms and protocols take measures against an adversary's possible actions and try to ensure that the protocol's goal is achieved with regard to all possible attacks.

Cryptographic protocols are used mainly for: establishment of symmetric session secrets (for one-to-one authentication), signing message digests (for one-to-many, or broadcast authentication), schemes of user or remote workstation authentication and computerized coin tossing [3, 4].

II. CRYPTOGRAPHIC PROTOCOL

A. Description of the block cipher and processing algorithm

The data stream is divided into blocks of length equal to a power of two (2^n). For demonstration of the encryption algorithm and the majority of the experiments a length of 64 bits is chosen. The cipher uses a symmetrical key algorithm with key length equal to the block length. The encryption consists of n ($n=\log_2 blocklength$) major phases and a final inversion phase. In the example shown on Fig. 1 with block length L equal to 64 the phases are 6+1. During the six major phases the ciphertext is divided into segments of length 2^{k-1} where k is the number of the current phase. In each phase the algorithm performs $L/2^k$ operations with neighboring segments. The result of each such operation forms a segment of greater rank, having the combined length of the two original segments and preserving their location in the ciphertext.

[1]Sivo Daskalov is a student in the CST department of the Technical University of Varna, E-mail: sivodaskalov@gmail.com

[2]Milena Karova is an Associate Professor in the Computer Sciences and Technologies department of the Technical University of Varna, 1 Studentska Street, Varna 9000, Bulgaria, E-mail: mkarova@ieee.bg

The processing of each pair of segments is controlled by a predefined bit in the key and there are three possible outcomes of the operation:

- The value of the control bit of the key is 0 – The two segments preserve their location and value.
- The value of the control bit of the key is 1 and the two segments are not identical – In this case the segments preserve their value but swap their location.
- The value of the control bit of the key is 1 and the two segments are identical – In this case each bit of the two segments is inverted.

The last bit of the key is used for an inversion of the cipher text block if the bit's value is 1

TABLE I
BITS OF THE CIPHER KEY USED IN EACH PHASE OF THE ENCRYPTION

Phase number	Segment length	Number of bits of the key used	Location of the used bits
1	1	32	1-32
2	2	16	33-48
3	4	8	49-56
4	8	4	57-60
5	16	2	61-62
6	32	1	63

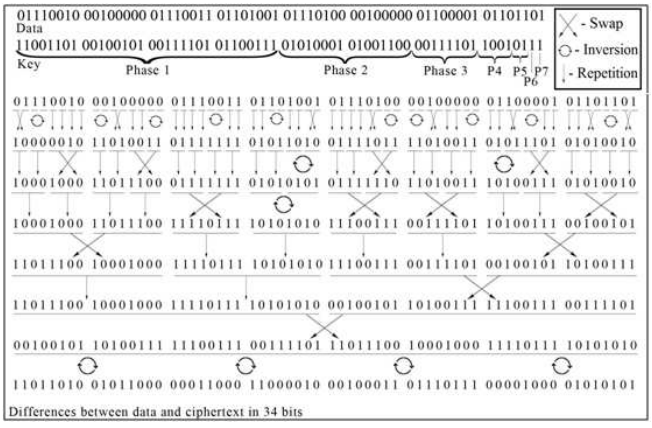

Fig. 1. Demonstration of the phases in the encryption of a data block

B. Initial key generation

The problem with initial key generation can be solved by the implementation of asymmetrical cryptographic approaches such as the usage of a public and private key [1, 5]. After the establishment of the initial key safe transmission of following replacement keys can be performed through the data channel as explained further in this paper.

C. Extended multi-phased cipher with doubled block length

The previously described block cipher for each phase of the encryption is applied to a half of the doubled ciphertext, using half of the key. With the unused part of the key, the XOR operator is performed on the unprocessed half of the ciphertext. This phase of ciphertext generation in the example on Fig. 2 is repeated four times to fully exhaust the combinations of semi-key and semi-block. This process is highly customizable in terms of both number of repetitions of the phase and combinations of semi-keys and semi-blocks for the encryption.

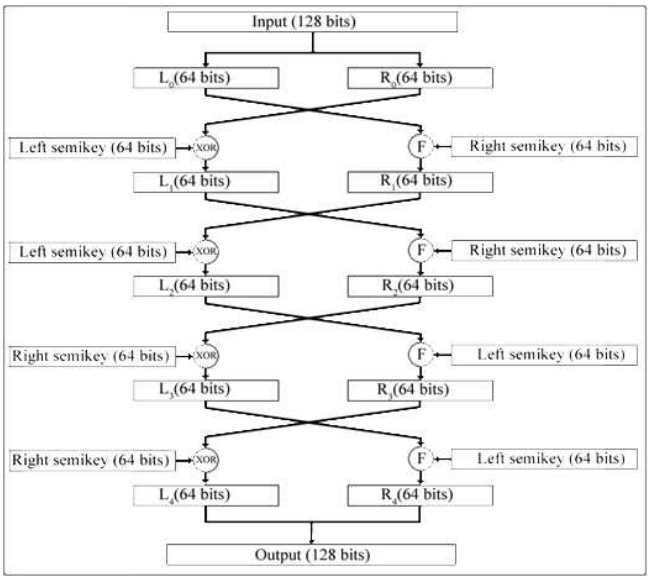

Fig. 2. Demonstration of the multi-phased cipher

D. Aperiodic key replacement

Four bits from each key are selected and form a number between 0 and 15. This number defines how many blocks of data will be encrypted with the given key. This parameter of the key will be referred to as its durability and is between 0 and 15. The position of the four bits defining the key durability is known by both sides in the communication and after the transfer of the said number of encrypted blocks through the channel a key replacement is pending. It is generated pseudo randomly by the sender and each bit of the new key has a 50% probability to be of each of the two logic values. The resulting keys are of unknown number of 'true' bits most often varying between 28 and 36 when the key length is 64 bits and the expected value of this number is half the key length.

Initially, the four bits defining the generated key's durability are positioned at the start of each quarter of the key, at positions 1, 17, 33 and 49. The position of the durability bits is shifted by the durability of the current key and hence increased unpredictability is achieved. The current key is used for the last time when encoding the generated key before its transfer through the channel. This means that before the substitution of a key, between one and sixteen blocks of data will be encrypted, the last one being the generated replacement key. This process continues until the connection is closed or all data blocks are sent successfully. The randomness of the key durability improves the security of the transmission by increasing the unpredictability of the content of the sent blocks since the replacement keys are mixed with the sent data.

III. Experiments

A. *Realization of the algorithm and analysis setup*

The simulation program has been written on the programming language C++ [2]. The encryption and decryption have been simulated on a single machine with the following characteristics:

- Processor – Intel Core i5-3570 3.4GHz Quad-Core
- Memory – GeIL EVO Corsa DDR3 2133MHz 8GB

B. *Performed experiments*

- Processing speed comparison with common encryption algorithms
- Analysis of the difference between plaintext and ciphertext
- Analysis of the randomness of key durability
- Performance analysis of the algorithm when using various block sizes

IV. Results

A. *Processing speed comparison with common encryption algorithms*

The encryption and decryption speed of the proposed cipher with block size of 128 bits and its extended modification, namely Scramble and MultiScramble, has been compared to well established block ciphers such as DES, AES, RC5 and BlowFish. The results of the experiments are shown on Fig. 3 and Table II.

TABLE II
COMPARISON OF ENCRYPTION SPEED FOR VARIOUS ALGORITHMS

	Seconds to encrypt 1 million symbols	Encryption speed [mbps]
DES	1.41	5.67
AES	1.55	5.16
MultiScramble	1.52	5.23
Scramble	0.51	15.38
RC5	0.12	66.67
BlowFish	0.06	133.33

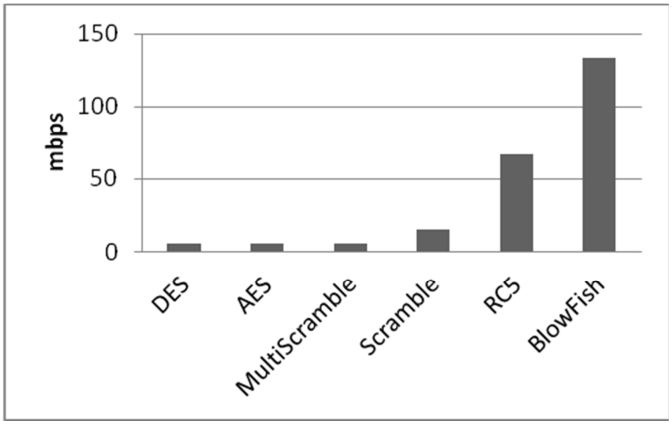

Fig. 3. Processing speed (encryption and decryption) in [mbps]

C. *Analysis of the difference between plaintext and ciphertext*

The differences between plain and encrypted data have been tracked for 32 768 blocks of length 64 bits. The expected value of these differences is 32, which is half of the block size. The results are presented on Fig. 4 and Table III.

TABLE III
ANALYSIS OF DIFFERENCES BETWEEN PLAINTEXT AND CIPHERTEXT

Differences	Block count	Probability
20	78	0.2%
22	315	1%
24	943	2.9%
26	2166	6.6%
28	3958	12.1%
30	5538	16.9%
32	6491	19.9%
34	5634	17.2%
36	3948	12.1%
38	2178	6.7%
40	996	32%
42	368	1.1%
44	104	0.3%
Total:	32 678	100%

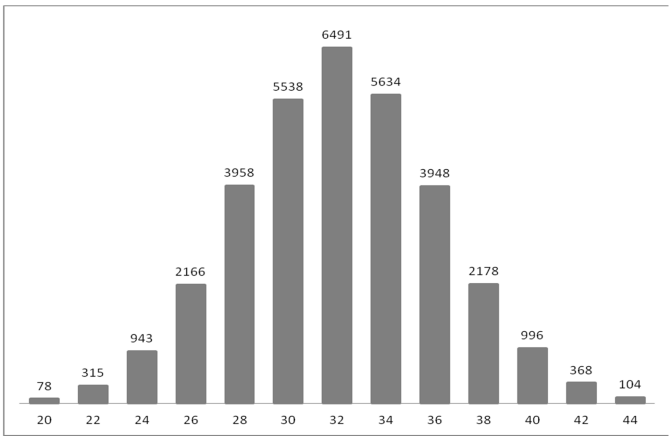

Fig. 4. Analysis of differences between plaintext and ciphertext

D. *Analysis of the randomness of key durability*

The randomness of key durability has been verified by analyzing a chain of 5000 generated keys. The sizes of the resulting key groups are nearly identical, as shown on Table IV, which proves the random nature of the parameter.

TABLE IV
ANALYSIS OF THE RANDOMNESS OF KEY DURABILITY

Durability	Key count	Probability
0	311	6.22%
1	314	6.28%
2	354	7.08%
3	335	6.7%
4	311	6.22%
5	336	6.72%
6	286	5.72%
7	329	6.58%
8	282	5.64%
9	322	6.44%
10	277	5.54%
11	300	6%
12	301	6.02%
13	300	6%
14	300	6%
15	342	6.84%
Total keys:	5000	100%

E. *Performance analysis of the algorithm when using various block sizes*

An evaluation of the time needed to encrypt and decrypt 1 million symbols has been carried out. Its results are shown on Table V and Fig. 5. The data shows that the processing speed is not strictly related to the choice of block size. However, it is clear that the lowest processing time has been reached when using blocks of size 64 bits.

TABLE V
PERFORMANCE SPEED COMPARISON WHEN PROCESSING 1 MILLION SYMBOLS FOR VARIOUS BLOCK SIZES

Block size	Block count	Time needed
16	567708	0.57 seconds
32	284402	0.53 seconds
64	142805	0.51 seconds
128	71463	0.52 seconds
256	36284	0.53 seconds

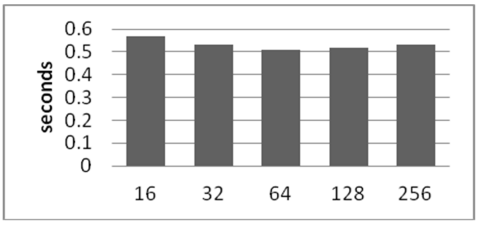

Fig. 5. Performance speed comparison for various block sizes

V. CONCLUSION

Modern block cipher protocols support several modes of operation to provide the confidentiality for the requirements of different applications. In this paper, a novel structure of multi-phased block cipher with doubled block length is proposed. The aperiodic key replacement is applied to improve the security of data transmission. The use of this method is discussed. Various analysis of the algorithm's performance have been performed, including but not limited to comparison with widely used standards. The algorithm's security and possible vulnerabilities in a running communication system are yet to be evaluated.

REFERENCES

[1] Menezes A., Oorschoot P., Vanstone S., Handbook of Applied Cryptography, CRC Press, 1996, ISBN: 13-978-0849385230
[2] Schneier B., Applied Cryptography: Protocols, Algorithms and Source Code in C, Copyrighted Material, ISBN: 13-978-0471117094
[3] Mahalingam Ramkumar, Symetric Cryptographic Protocols, Springer International Publishing, Swizerland, 2014, ISBN: 978-3-319-07583-9
[4] [4]. Goots N, Izotov B., Moldovyan A., Moldovyan N., Modern Cryptography: Protect Your Data with FAST Block Ciphers, A-LIST, LLC 295, East Swedesford Rd, 2003, ISBN: 1-931769-12-5
[5] Basin D., Paterson K., Information Security and Cryptography, Springer, 2014, ISSN: 1619-7100.