Papílio Cryptography Algorithm

Frederiko Stenio de Araújo, F.S,, Karla Darlene Nempomuceno Ramos, Benjamín René Callejas Bedregal, and Ivan Saraiva Silva

Universidade Federal do Rio Grande do Norte
Departamento de Informática e Matemática Aplicada
59072-970, Natal-RN, Brazil
steniorn@uol.com.br,karla@ppgsc.ufrn.br,{bedregal,ivan}@dimap.ufrn.br

Abstract. Papílio is a Feistel cipher encryption algorithm where the coder process (function F) is based in the Viterbi algorithm. The Viterbi algorithm was proposed as a solution to decode convolutional codes. There are several parameters that define the convolution code and Viterbi algorithm; one of them is the generator polynomial. To use Viterbi algorithm in cryptography, it is necessary to make some modifications. The proposed one does not depend on the parameters of Viterbi nor on the parameters of convolution. In this work we will analyze the cryptographic indices (avalanche, diffusion and confusion) of Papílio considering all possible different polynomials and fix the other parameters.

1 Introduction

A.J. Viterbi in [11] developed the Viterbi Algorithm (VA) in 1967 as a solution for decoding convolutional codes. Convolutional encoder (CE) with Viterbi decoder is a FEC (Forward Error Correction) technique that is particularly suitable to channels where the transmitted signal is corrupted mainly by additive white Gaussian noise [2]. Since then, other researchers have applied VA and CE for other areas of applications such as recognition of handwritten word [8], target tracking [1], image edge detection [7]. Since the CE, increases the length of the input bitstream (it is an injective and not surjective one) and the VA only decode the bitstream generated by CE and some others few which can be recovered (is a partial not injective and not surjective function), this process can not be considered as cryptographic method.

This work apply a modification in the VA, considering specific parameters for the VA and CE, in order to get a bijective function. This bijective function was inserted as the function F in a Feistel cipher with 16 rounds, blocks of 64 bits and keys of 128 bits [9, 10]. To generate the 16 sub-keys, it was used the modified Viterbi (MV). This feistel cipher will be called of Papílio[1]. We will show an study that chooses eight polynomials which provide to Papílio better indices of cryptography and also will provide some evidences that Papílio can be improved in the aspect of the complexity of cryptanalysis and the execution time.

[1] The name Papílio was given because the trellis of VA form a butterfly, and *Papílio Thoas Brasiliensis* is the name of a very common sort of butterflies in Brazil.

2 Modified Viterbi

VA attempts to find the closest "valid" sequence to the received bitstream, that is, a sequence which when applied to the CE results in the received bitstream. Notice that two different sequences used as inputs for the convolution encoder result, necessarily, in two different sequences, that compute an injective function. So, the VA can be seen as a decoder. But, VA only decode bitstream generated by CE. Therefore, for VA to be used in the cryptography it is necessary that it processes any input sequence in a bijective way. The VA will be modified to deal with all possible bitstreams, in such a way that can be seen as a bijective function and therefore appropriated for cryptography.

The MV algorithm proposed increases the code space matching VA with CE. For MV deal with any input sequence, independently of the current state, it was created besides output sequence S_0, an output sequence S_1. S_0 presents the result of VA. S_1 exhibits if each output symbol of S_0 was obtained in agreement with the VA, or if it was obtained in an special shape (MV). When an output symbol of S_0 is obtained in agreement with the VA, S_1 generates the bit 0, otherwise S_1 generates the bit 1. The MV algorithm is initialized to zero state and works as VA until an input symbol of bitstream is not appropriate in the current state, i.e. is "invalid". When this occurs, the symbols of not appropriated label are treated separately for the CE with initial state being the current state of MV. It is observed that CE will generate $\lceil \frac{s}{n} \rceil$ additional labels, being $\frac{n}{s}$ the rate of CE. With this procedure the generated labels can be treated by the VA. The application of the VA would generate a size label n for each one of the generated additional labels. However what interests in the code is the generation of an only size label n. The adopted solution consists of considering, to compose the flow S_0, just the first of the $\lceil \frac{s}{n} \rceil$ size labels n (the bitstream S_1 receives the value 1). The continuation of the code process using the VA is adopted as current state the last state of the process of convolution, until a new "invalid" label is found or the code is finished. The bitstreams S_0 and S_1 are independent. At the end of the code the bitstream are concatenated, in way to generate a bitstream of same length of the original. Through MV it's possible create tables that help the code process. For example, the table 1 exhibits the MV taking into account the CE and VA where $n = 1$, $s = 2$, $Q = 3$, $m = 2$ and generator polynomial $G = 111101$.

3 The Encryption Algorithm Papílio

Papílio is a Feistel cipher encryption algorithm where the function F is the function computed by the MV algorithm whose parameters (codification rate $\frac{n}{s}$, Q, m and the polynomial generator) are opens in a first moment. The main characteristics:

Block length: Actually are considered block of 64 bits. Nevertheless, because the MV does not depends on length of block, its size can be changed in a

Current State	Input	S_0	S_1	Next State	Current State	Input	S_0	S_1	Next State
0	00	0	0	0	2	00	0	1	0
	01	0	1	2		01	1	0	3
	10	1	1	1		10	0	0	1
	11	1	0	2		11	1	1	3
1	00	1	0	2	3	00	0	1	0
	01	0	1	2		01	0	0	1
	10	1	1	1		10	1	0	3
	11	0	0	0		11	1	1	3

Table 1. MV of CE with $n = 1$, $s = 2$, $Q = 3$, $m = 2$ and generator polynomial $G = 111101$.

further implementation turning fix as 128 bit or variable in function of the key;

Size of the key: 128 bits. But, its size could be variable or greater;

Number of rounds: 16. But, this quantity can be reduced to 6 or turned into variable (between 6 to 16) without losing the good cryptographic indices;

Sub-key generations: Papílio uses 16 sub-keys that are generated from the 128-bit encryption key. The sub-keys are stored temporarily in an array. The scheme for generation is as follows. The first four sub-keys, labelled SC_1, SC_2, SC_3 and SC_4, are generated by applying the MV in the 128-bit initial key, which generates two 64-bit bitstream. Applying MV to the two 64-bit bitstreams it generates four bitstreams of 32-bits that corresponds to the first four sub-keys. To generate the four following sub-keys, the four bitstream are concatenated generating an alone of 128-bits and the procedure to generate first four sub-keys is repeated until all 16 sub-keys are generated.

Decryption: as with most block ciphers, the process of Papílio decryption is essentially the same as the encryption process, except the sub-keys that are employed in reverse order. So, use SC_{16} in the first round, SC_{15} in the second round, and so on until SC_1 is used in the last round. This feature avoids implementing two different algorithms, one for encryption and one for decryption;

Operation Modes: Papílio was implemented in the four usual modes (ECB, CBC, CFB and OFB).

Programming language: Papílio was implemented in C.

4 The Choice of Better Polynomials

By simplicity and implementation's performance, was considered for MV a CE and VA with the following parameters: codification rate $\frac{n}{s} = \frac{1}{2}$, $Q = 3$ and $m = 2$. With this values we have 64 (2^{sQ}) possible polynomials.

The idea is to analyze considering the behavior of each polynomials regarding to the avalanche effect (in the key and in the block), the diffusion and the confusion properties and to select the eight polynomials with better results.

First tests and measure used: First was made to each polynomials and operation mode a test for confusion and diffusion based on a book of project Gutemberg [4]. We extract from the book the first 3536 characters (including the spaces), despising the 400 first characters to erase the heading. The keys used in this test was (pseudo)randomly generated (all polynomials used the same keys). The test of avalanche effect (in the block and key) was realized on 50 blocks of plaintext and 50 keys randomly generated.

To measure the avalanche effect in the block was used the arithmetic average of Hamming distances between the encryption of a plaintext block and the encryption (with the same key) of the same plaintext block changing a bit in all possible ways. Analogously, to measure the avalanche effect in the key was used the arithmetic average of Hamming distances between the encryption of a text block and the encryption of the same text block changing a bit on the key in all possible ways. The measure of diffusion was calculated using the standard deviation of frequencies of characters in the cyphertext. The confusion was measured using the average of Euclidean distances between the encryptions of plaintext with the original key and the plaintext with the original key changing only an unique bit. This result is divided by the greatest Euclidean distance possible, which allows us to normalize this value obtained a value between 0 and 1.

The avalanche effect in the block for the modes ECB and CBC is, for the most of polynomials, between 0.45 (45% of bits, or more, are changed) and 0.51 which is a very good index, considering that the ideal value is 0.5. For the modes OFB and CFB, the avalanche effect is constant (0.0156), nevertheless it is not a problem of Papílio, but of the modes, because we are measuring only the avalanche in an unique block, and therefore a change of a bit only affect an unique bit. The avalanche effect in the key still is better, because in the modes ECB and CBC 92% of polynomials matched between 0.48 and 0.51 and in the modes CFB and OFB 79% of polynomials matched between 0.48 and 0.51.

The confusion in the modes ECB and CBC, the half of polynomials (50%) are between 0.38 (38%) and 0.41 which is not ideal (the ideal is similar to avalanche, i.e. 0.5 or 50%) but it is reasonable, more over if we consider that the Rijndael algorithm, using the implementation of Rijndael founded in [6] and in the same conditions of test, obtained confusion index of 40.5%. In the modes CFB and OFB, 47% of polynomials are between 0.38 and 0.41. In all operator modes we have more of 8 polynomials with confusion index greater than 40.

The greatest diffusion index for the mode ECB was 0.0285 and 56% of polynomials have lesser than 0.02. In the mode CBC, the greatest diffusion index was 0.0255 and 81% of polynomials have an index lesser than 0.02. In the mode CFB, the greatest diffusion index was 0.0252 and 80% of polynomials have an index lesser than 0.02. Finally, in the mode OFB, the greatest diffusion index was 0.0257 and 80% of polynomials have an index lesser than 0.02. Thus, in any operator mode the symbols in the ciphertext have, practically, the same distribution which allows us to conclude that the statistical frequencies of symbols in the plaintext were destroyed. Therefore, there is not a statistical relation between the frequencies of symbols in the plaintext and the ciphertext.

Similarity of indices: In order to check if the indices obtained don't depend strongly on texts and key, but only depend on polynomials used, we will make new tests for avalanche effect on the block and diffusion and then we will measure the degree of similarity between the results using the standard deviation of results. For the tests of avalanche was generated 100 series of 50 plaintext blocks and for the diffusion were made 100 series using only 3536 characters despising the 400 firsts of a book of Gutemberg project [4]. For each 10 series was used a different book.

The avalanche affect on the block 98% of the polynomials have an standard deviation lesser than 0.06%, and the diffusion of all polynomials is lesser than 0.35%. Both results are very good, because indicate that Papílio independently of the polynomials is very stable. Since the confusion and avalanche effect on key are, in some sense, subordinated to the avalanche effect on the block, we can conclude that both effects neither depend on the plaintext nor the key used. This also is true for the other modes.

The winner polynomials: With the conviction that the Papílio behavior depends quasi exclusively of polynomials, we will make a championship to determine the polynomial which provides to Papílio the best cryptographic indices. Because the confusion strongly relates to avalanche in the block and the avalanche in the key as well as the diffusion obtained in all tests and in all polynomials well indices, beyond diffusion need more computational effort, we opted to only consider the avalanche effect on the block.

The championship consisted in performing 50 news tests for avalanche effect in the block using keys and block generated randomly, at each test the polynomials that achieve the index more proximate of 50% gain a point. To avoid arrive in local optimum when a polynomial had 20 points this would classify for the next stage and the championship continued without it. In the next stage of championship was performed 100 test considering again random keys and blocks. The selected polynomials was those which obtained 50 points. For simplicity we only make the championship for the ECB mode.

5 Final Remarks

The empirical analysis showed that the proposal cipher has very good performance w.r.t. of avalanche, diffusion and confusion properties. However in spite that these properties are interesting and important, just having these properties does not mean that a cipher is secure. In order to conclude that Papílio is reliable, yet is necessary a treatment of the security of Papílio cipher considering modern cryptanalysis methods, such as linear cryptanalysis and differential cryptanalysis. This study will be made in further works.

Considering that by the similarity degree only a few tests will be necessary to analyze the cryptographic indices of Papílio for each polynomial. But even so, we perform a great number of reliable tests, resulting in the choice of eight polynomials. If we analyze the individual avalanche index round to round of

each one of these polynomials we will see than we can already obtain very good indices from the round 6. This allows the thoght to reduce the rounds number, decreasing the execution time of Papílio or yet turning it variable, which would difficult the cryptanalysis and would improve the execution time. Since MV can be applied to any length of block, we also can increase the size of the block which also would improve the execution time or variable in term of the key. Since we have eight good polynomials, we also could apply different polynomials to each round (the choice would be in function of the sub-key and current block) which would not increase considerably the computational effort but would increase considerably the cryptanalysis difficulty, once that for each block in the plaintext (fixing the key) we have 2^{48} possible ways to encoder it (considering 16 rounds). But, considering that each polynomials if we changed the start state we will have four different results, then quantity of possible combinations of functions can arrive to 2^{80} for each block!! which will turn eventually impossible the cryptanalysis without knowing the key, more over considering that the combination of polynomials and start states will change to each block, thus the knowledge of a combination for a block not will help to know the ciphertext. So, Papílio is a very flexible cryptographic algorithm and with very good cryptographic indices.

References

1. Demirbas, K.: Target Tracking in the Presence of Interference, Ph.D. Thesis, University of California, Los Angeles, 1981.
2. Fleming, C.: *A Tutorial on Convolutional Coding with Viterbi Decoding*. Spectrum Applications, july, 1999.
3. Forney, G.D. Jr.: Convolutional codes II: Maximum-Likehood Decoding. *Information and Control*, 25(3)177-179, 1974.
4. Project Gutemberg. http://www.veritel.com/gutenberg/index.html. Access in march 2003.
5. Hopcroft, J.E. and Ullman, I.: *Introduction to automata theory, languages and computation*. Addison-Wesley, 1979.
6. OpenSSL project. http://www.openssl.org/ last modification in april, 17 of 2002. Access in april, 10 of 2003.
7. Pitas, I.: *A Viterbi algorithm for region segmentation and edge detection*. Proc. CAIP89, Leipzig, pp. 129-133, 1989.
8. Ryan, M.S. and Nudd, G.R.: *The Viterbi Algorithm*. Department of Computer Science, University of Warwick, Coventry, CV4 7AL, England, February, 1993.
9. Schneier, B.: Applied Cryptography. 2nd Edition, New York, John Wiley & Sons, inc., 1996.
10. Stalling, W.: *Cryptography and Network Security: Principles and Practice*. 2nd edition. Prentice Hall, 1998.
11. Viterbi, A.J.: Error Bounds for Convolutional Codes and na Asymptotically Optimum Decoding Algorithm. *IEEE Transactions on Information Theory*, April 1967