

A Block Cipher Based on Shuffle and θ -Deletion on Trajectories

D. Jayaseelan Samuel¹, and P.J. Abisha²

^{1,2}Department of Mathematics, Madras Christian College,
Tambaram, Chennai – 600 059, India

¹jayaseelansamuel@mcc.edu.in,

²abishajeyanthi@mcc.edu.in

February 10, 2017

Abstract

Different public key cryptosystems based on formal language theory have been proposed and studied. But the application of formal language theory to symmetric key cryptography is the focus of our interest. In this paper, we present a new block cipher which uses shuffle on trajectories for encryption and a new type of deletion, namely, θ -deletion on trajectories for decryption.

AMS Subject Classification: 94A60, 11T71, 68P25

Key Words and Phrases: Symmetric key cryptosystem, Block cipher, Trajectories, Shuffle, deletion, θ -deletion on trajectories

1 Introduction

Symmetric key encryption schemes provide secure communication for a pair of communication partners. An enemy who intercepts a message should not get any significant information about its content. Among all the algorithms, symmetric key encryption algorithms have the fastest implementation in hardware and software. Therefore, they are very well-suited to the encryption of

large amount of data. On the other hand public key cryptography methods [2, 3] are useful in exchanging the secret key of the symmetric key encryption schemes. Different public key cryptosystems based on formal language theory have been proposed [1, 8]. Symmetric key cryptosystems based on operations from formal language theory, namely, shuffle and deletion on trajectories were proposed and studied [9].

In formal language theory, word and language operations with the aid of trajectories provide a general method for the study of properties of sequential insertion and deletion operations [4, 5, 6, 7]. A trajectory gives a constraint on the scattered insertion of a word into another one, or scattered deletion of a word from another word. Moreover, deletion on trajectories is an inverse of shuffle on trajectories. Intuitively, one can understand that the concept of insertion and deletion may be applied for encryption and decryption respectively.

One of the standard techniques in symmetric key encryption is of block cipher which works on messages of fixed length. In this paper, a new type of deletion on trajectories called θ -deletion is introduced. A new block cipher which uses shuffle on trajectories for encryption and θ -deletion on trajectories for decryption is proposed and its security is investigated.

2 Shuffle and Deletion on Trajectories

An alphabet Σ is a finite and nonempty set of symbols. In the sequel we shall use a fixed alphabet Σ . Σ is assumed to be non-singleton, if not stated otherwise. The set of all words (over Σ) is denoted by Σ^* . This set includes the empty word λ . The length of a word w is denoted by $|w|$. $|w|_a$ denotes the number of occurrences of a in w , for $a \in \Sigma$. For a nonnegative integer n and a word w , we use w^n to denote the n concatenated copies of w and w^R to denote the reversal of the word w .

Definition 2.1. [6] A binary word operation is a mapping $\diamond : \Sigma^* \times \Sigma^* \rightarrow 2^{\Sigma^*}$, where 2^{Σ^*} is the set of all subsets of Σ^* . For any languages X and Y over Σ , we define

$$X \diamond Y : \bigcup_{u \in X, v \in Y} u \diamond v$$

The word operation $\diamond$ defined by $u\diamond v = v\diamond u$ is called reverse $\diamond$. For many examples of common binary operations we refer the reader to [7]. Here we recall two of them.

Shuffle (or scattered insertion)

$$u \sqcup v = \{u_1v_1\dots u_kv_ku_{k+1}, u = u_1\dots u_ku_{k+1}, v = v_1\dots v_k, |k \geq 1\}.$$

Scattered deletion

$$\alpha \rightsquigarrow \beta = \{u_1\dots u_ku_{k+1} \mid k \geq 1, \alpha = u_1v_1\dots u_kv_ku_{k+1}, \beta = v_1\dots v_k\}$$

The shuffle on trajectories was introduced in [6] as a concept generalizing the above insertion word operation. Trajectory is essentially a syntactical condition specifying how two words α and β are merged together into a resulting word. Formally, a trajectory is a string over the trajectory alphabet $V = \{0, 1\}$. Let t be a trajectory, $t \in V^*$. Let α, β be two words over Σ . Some necessary definitions are recalled [4, 6].

Definition 2.2. [6] The shuffle of α with β on the trajectory t , denoted by $\alpha \sqcup_t \beta$, is defined as follows

$$\alpha \sqcup_t \beta = \{\alpha_1\beta_1\dots\alpha_k\beta_k \mid \alpha = \alpha_1\dots\alpha_k, \beta = \beta_1\dots\beta_k, t = 0^{i_1}1^{j_1}\dots 0^{i_k}1^{j_k} \text{ where } |\alpha_m| = i_m \text{ and } |\beta_m| = j_m \text{ for all } m, 1 \leq m \leq k\}$$

Observe that in $\alpha \sqcup_t \beta$ the positions of 0's in t correspond to letters of α , while 1's correspond to letters of β .

Example : Let α and β be the words $\alpha = a_1a_2\dots a_8, \beta = b_1b_2\dots b_5$ and assume that $t = 0^31^20^310101$. The shuffle of α and β on the trajectory t is $\alpha \sqcup_t \beta = \{a_1a_2a_3b_1b_2a_4a_5a_6b_3a_7b_4a_8b_5\}$

Definition 2.3. [5] The deletion of β from α on trajectory t is the following binary word operation:

$$\alpha \rightsquigarrow_t \beta = \{\alpha_1\dots\alpha_k \mid \alpha = \alpha_1\beta_1\dots\alpha_k\beta_k, \beta = \beta_1\dots\beta_k, \text{ where } t = 0^{i_1}1^{j_1}\dots 0^{i_k}1^{j_k}, |\alpha_m| = i_m \text{ and } |\beta_m| = j_m \text{ for all } m, 1 \leq m \leq k\}$$

It follows from the above definition that if $|\alpha| \neq |t|_0$ and $|\beta| \neq |t|_1$, then $\alpha \rightsquigarrow_t \beta = \phi$. Observe that $\alpha \sqcup_t \beta = \gamma$ if and only if $\gamma \rightsquigarrow_t \beta = \alpha$.

Example : Let $\alpha = 110001001111100, \beta = 00011100$ and $t = 001110000111011$ then $\alpha \rightsquigarrow_t \beta = 1110011$.

In this paper, we define a special type of deletion on trajectories.

Definition 2.4. Let Σ be an alphabet, $V = \{0, 1\}$ be the trajectory alphabet and $\theta \in V$. If $\alpha = a_1 a_2 \dots a_n, a_i \in \Sigma$ ($1 \leq i \leq n$) and the trajectory $t = t_1 t_2 \dots t_n, t_i \in V$ ($1 \leq i \leq n$) the θ -deletion from α by the trajectory t is given by $\beta = b_1 b_2 \dots b_n$, where

$$b_i = \begin{cases} a_i & \text{if } t_i = \theta \\ \lambda & \text{otherwise} \end{cases}$$

We denote β by $\alpha \smile_{\theta} t$.

We note that, if θ occurs in the i -th position of the trajectory t , then the corresponding letter a_i in α is retained. A letter in α is deleted if the corresponding position letter in the trajectory is not θ . Hence, $\alpha \smile_{\theta} t$ is a subword of α whose letters are the letters of α with positions correspond to the positions of θ in t . Also $|\beta| = |t|_{\theta}$.

Example : If $\alpha = 110001001111100$ and $t = 001110000111011$ then $\alpha \smile_0 t = 1110011$ and $\alpha \smile_1 t = 00011100$.

3 Block ciphers using shuffle on Trajectory

In this section, we propose a new block cipher based on shuffle on trajectories in which both of the encryptor and decryptor use the same pseudorandom generator and its outputs are used as the trajectories. The encryptor and decryptor receive the same trajectories when the same seed is given as input. This seed is a function of two strings called the core vector and the initialization vector. Size of these two vectors and the size of the plaintext block as well as the ciphertext block are the same. Let it be n . Let Σ be the alphabet. The core vector $S \in \Sigma^n$ is a fixed word and it is a part of the secret key. To encrypt the first block, we choose the initialization vector $c_0 \in \Sigma^n$ as a true random word and keep c_0 as the first block of the ciphertext. Thus, both the encryptor and the decryptor will calculate the same seed, and obtain the same pseudorandom sequence w . There is a fixed positive integer k which is the number of rounds. The pseudorandom sequence w is divided into k words $w_1, w_2, \dots, w_k$ each of length n . These strings are taken as trajectories in each round of the encryption and decryption. To encrypt the next plaintext block the previous ciphertext block will be used as the initialization vector.

Encryption

The encryption procedure is the following. Initially, we set the initialization vector $c_0 = r_1 r_2 \dots r_n$ as a true random word. Let S be the core vector. The encryptor reads the plaintext block-by-block and generates the second, third, etc.. blocks of the ciphertext in the following way.

We describe the procedure to encrypt the i -th block. Let the plaintext in the i -th block be $m = m_1 m_2 \dots m_n \in \Sigma^n$. First the pseudorandom generator takes $g(S, c_{i-1})$ as input, where $g : \Sigma^n \times \Sigma^n \rightarrow \Sigma^n$, and generates a word $w_1 w_2 \dots w_k$, where $w_1, w_2, \dots, w_k \in \Sigma^n$ are pseudorandom sequences. Then, $u = f(m, c_{i-1})$ is computed, where $f : \Sigma^n \times \Sigma^n \rightarrow \Sigma^n$ is a function which is invertible. For simplicity, one can choose $\Sigma = \{0, 1\}$ and f and g as XOR. In each round of the encryption this input u is split in to two strings u_0 and u_1 and shuffled according to the trajectory. Here u_0 is the prefix of u whose length is equal to the number of 0's in the trajectory used in that round and u_1 is the suffix of the word u such that $u = u_0 u_1$. Clearly, length of u_1 is equal to the number of 1's in the trajectory. At the end of the k -th round, we obtain the ciphertext block c_i . The exact steps are given in the following algorithm.

Step 1: compute $u = f(m, c_{i-1})$

Step 2: for $j = 1$ to k do

set $t = w_j$

set $u = u_0 u_1$, where $|u_0| = |t|_0$ and $|u_1| = |t|_1$

if j is odd then

set $u = u_0^R \sqcup_t u_1$

else

set $u = u_0 \sqcup_t u_1^R$

end for

Step 3: set $c_i = u$

Decryption

To decrypt the i -th block, first the pseudorandom generator takes $g(S, c_{i-1})$ and generates the same pseudorandom sequences $w_1 w_2 \dots w_k$, where $w_1, \dots, w_k \in \Sigma^n$. In each round of the decryption, 0-deletion and 1-deletion are performed. The procedure is given in the following algorithm.

Step 1: set $u = c_i$

Step 2: for $j = k$ to 1 in steps of -1 do

set $t = w_j$

set $u_0 = u \smile_0 t$ and $u_1 = u \smile_1 t$
 if j is odd then
 set $u = u_0^R u_1$
 else
 set $u = u_0 u_1^R$
 end for

Step 3: set $m = f^{-1}(u, c_{i-1})$

Correctness

Clearly $\alpha_0 \sqcup \sqcup_t \alpha_1 = \alpha$ if and only if $\alpha \smile_0 t = \alpha_0$ and $\alpha \smile_1 t = \alpha_1$. This shows the correctness of the algorithms.

4 Illustration

The working procedure of the above system is illustrated by a small example.

Encryption

Let $c_{i-1} = 0^3 1^2 0 1^2 0^3 1 0 1^2$ and the message to be encrypted in the i -th block be $0^2 1^2 0^3 1 0^2 1 0 1 0 1$.

We fix the number of rounds $k = 2$. Suppose that the output of the pseudorandom generator is $w_1 = 1^3 0^2 1^2 0^3 1^4 0$ and $w_2 = 0^2 1 0 1^4 0^5 1^2$. Then the encryption is as follows.

Step 1: Set $u = m \oplus c_{i-1} = 0^2 1 0 1 0 1 0^3 1^4 0$

Step 2:

Round 1: Here $t = w_1 = 1^3 0^2 1^2 0^3 1^4 0$ and so $|t|_0 = 6$ and $|t|_1 = 9$

set $u_0 = 0^2 1 0 1 0$ and $u_1 = 1 0^3 1^4 0$

set $u = u_0^R \sqcup \sqcup_t u_1 = 1 0^3 1 0 1 0 1 0^3 0^2$

Round 2: Here $t = w_2 = 0^2 1 0 1^4 0^5 1^2$ and so $|t|_0 = 8$ and $|t|_1 = 7$

set $u_0 = 1 0^3 1 0 1 0$ and $u_1 = 1 0 1^3 0^2$

set $u = u_0 \sqcup \sqcup_t u_1^R = 1 0^4 1^3 0 1 0 1 0^2 1$

Step 3: Set $c_i = u = 1 0^4 1^3 0 1 0 1 0^2 1$

Thus, the ciphertext corresponds to the i -th block is $1 0^4 1^3 0 1 0 1 0^2 1$.

Decryption

Let the ciphertext in the $(i-1)$ -th block be $c_{i-1} = 0^3 1^2 0 1^2 0^3 1 0 1^2$ and ciphertext in the i -th block be $1 0^4 1^3 0 1 0 1 0^2 1$. Since number of rounds $k = 2$, by taking $g(S, c_{i-1})$ as input the pseudorandom generator generates the same random sequences $w_1 = 1^3 0^2 1^2 0^3 1^4 0$ and $w_2 = 0^2 1 0 1^4 0^5 1^2$.

Step 1: Set $u = c_i = 1 0^4 1^3 0 1 0 1 0^2 1$

Step 2:

Round 1: Here $t = w_2 = 0^2 1 0 1^4 0^5 1^2$

set $u_0 = u \smile_0 t = 10^3 1 0 1 0$ and $u_1 = u \smile_1 t = 0^2 1^3 0 1$

set $u = u_0 u_1^R = 10^3 1 0 1 0 1 0 1^3 0^2$

Round 2: Here $t = w_1 = 1^3 0^2 1^2 0^3 1^4 0$

set $u_1 = u \smile_0 t = 0 1 0 1 0^2$ and $u_2 = u \smile_1 t = 10^3 1^4 0$

set $u = u_0^R u_1 = 0^2 1 0 1 0 1 0^3 1^4 0$

Step 3: Set $m = u \oplus c_{i-1} = 0^2 1^2 0^3 10^2 1 0 1 0 1$

Thus, the corresponding message block is $0^2 1^2 0^3 10^2 1 0 1 0 1$.

5 Security

To get the desired security it is advisable to take 128-bits blocks. One possible attack for this system is by considering encryption of a message with two identical blocks in adjacent places. Since the encryption is performed by cipherblock chaining mode (CBC mode), and since each time when we encrypt, a completely new random initialization vector is used, this kind of attack does not reveal anything about the plaintext and the key. In a cryptosystem using shuffle on trajectories described in [9] the ciphertext contains two components. It is noticed that, if the characters are the same in both the components in a particular position then it is possible to find the corresponding character of the trajectory. To overcome this problem, in this paper, we break the message in to two blocks and shuffle them according to the trajectory and this process is repeated k times in CBC mode. Hence such kind of attacks are not successful.

6 Conclusion and Future Work

In this paper, we presented a block cipher based on shuffle and θ -deletion on trajectories. We note that, in this cryptosystem we are using the cipherblock chaining mode (CBC mode) and so this system encrypts the same ciphertext differently with different initialization vectors. The encryption of a plaintext block not only depends on the key but also on the preceding plaintext blocks. Therefore, if the order of the ciphertext blocks is changed or if ciphertext blocks are replaced, then the decryption becomes impossible. However, serious

security analysis and rigorous machine-independent investigation should be necessary in the future work.

References

- [1] P.J. Abisha, D.G. Thomas, K.G. Subramanian, Public key cryptosystems based on free partially commutative monoids, Proceedings of INDOCRYPT 2003, LNCS 2904, pp. 218 - 227, 2003.
- [2] H. Delf, H. Knebl, Introduction to Cryptography Principals and Applications, Springer, 2002
- [3] W. Diffie, M. Hellman, New Directions in Cryptography, IEEE Transactions and Information Theory, IT-22(6), pp. 644 - 654, November 1976.
- [4] M. Domaratzki, Deletion along trajectories, Tech. Report 2003-464, School of Computing, Queen's University, 2003; Theoret. Comput. Sci. 320 (2004) 293-313.
- [5] L. Kari, P. Sosik, Aspects of Shuffle and Deletion on Trajectories, Theoretical Computer Science 332, pp. 47 - 61, 2005.
- [6] A. Mateescu, G. Rozenberg, A. Salomaa, Shuffle on Trajectories: syntactic constraints, TUCS Tech. Report No. 41, Turku Center for Computer Science, 1996; Theoretical Computer Science 197, pp. 1 - 56, 1998.
- [7] G. Rozenberg, A. Salomaa (Eds.), Handbook of Formal Languages, Springer, Berlin, 1997.
- [8] A. Salomaa, Computation and Automata, Cambridge university press, 1986.
- [9] D.J. Samuel, P.J. Abisha, Symmetric Key Cryptosystem based on Shuffle and Deletion on trajectories, Cryptography, Automata and Learning Theory (Eds. D. Gnanaraj Thomas, P. Abisha Jeyanthi), Narosa Publishing House, pp. 78-83, September 2010

