About Symbolic Encryption: Separable Encryption Systems

Adrian ATANASIU
Faculty of Mathematics, University of Bucharest
Str. Academiei 14, 70109, Bucharest, ROMANIA

Victor MITRANA[1]
Faculty of Mathematics, University of Bucharest
Str. Academiei 14, 70109, Bucharest, ROMANIA

Abstract. This paper continues the investigation regarding the operation of substituting subwords of a given word with other strings, an operation useful in cryptography. Some results concerning the closure properties of the families in the Chomsky hierarchy are presented. A set of different necessary conditions for the separable encryption systems are established. Possible applications in the authentication signature are finally mentioned.

1 Preliminaries

To substitute some subwords of a word with other strings in the aim of hiding the original message is one of the well-known techniques in cryptography. In [1] and [2] the substitution operation as a generalization of the insertion and deletion operations [7] has been introduced. A substitution can be viewed as a production of the form $x \longrightarrow y$ where the words x, y are given or are elements of some formerly defined languages. To apply sequentially such a substitution to a given text w means that one occurrence of x is replaced by y whilst in the parallel substitution all non-overlapped occurrences of x are simultaneously replaced by y. Thus, different texts are obtained, according to different decompositions of w with respect to x.

Some necessary conditions for the reversability of the sequential and parallel substitutions have been established in [1] and [2], respectively, for some particular words y. More recently [8], the reversability problem of the parallel substitution has

[1]Research supported by the Alexander von Humboldt Foundation

been solved for all possible words y. In [4], the definition of separable encryption systems and some basic properties have been presented.

In the sequel, the basic notions and notations necessary in the following sections will be presented. For formal languages details we refer to [6]. An alphabet is a finite nonempty set; if $V = \{a_1, a_2, \ldots, a_n\}$ is an alphabet, then any sequence $w = a_{i_1} a_{i_2} \ldots a_{i_k}, 1 \leq i_j \leq n, 1 \leq j \leq k$, is called word (string) over V. The length of the word w is denoted by $|w|$ and equals k. The empty word is denoted by $e, |e| = 0$. The set of all words over V is denoted by V^* and $V^+ = V^* - \{e\}$.

For a finite set A denote by $card(A)$ the cardinality of A. For two words x, y we denote by $N_x(y)$ the number of occurrences of x in y, that is

$$N_x(y) = card(\{\alpha | y = \alpha x \beta\})$$

and extend this notation to

$$N_A(y) = \sum_{x \in A} N_x(y)$$

Note that we count all different occurrences of x, including the overlappings. For $w, x, y \in V^*$, the sequential substitution of x by y in w is defined as

$$w(x \longrightarrow y) = \{uyv | w = uxv\}$$

while the parallel substitution is defined as:

$$w(x \Longrightarrow y) = \{z | z = z_0 y z_1 y \ldots y z_n | n > 0\}$$

such that

$$\begin{aligned} w &= z_0 x z_1 x \ldots x z_n, \\ N_x(z_i) &= 0,\ 0 \leq i \leq n. \end{aligned}$$

Moreover,

$$L(x \longrightarrow y) = \bigcup_{w \in L} w(x \longrightarrow y), \qquad L(x \Longrightarrow y) = \bigcup_{w \in L} w(x \Longrightarrow y)$$

The sequential substitution corespondes to the usual rewriting steps in rewriting systems whereas the parallel one coresponds to the Indian type of parallel rewriting [5].

In this paper we consider a generalization of the previous operations, namely by considering more substitution rules instead of just one, used in parallel. Thus, an encryption system may be viewed as a multi-agent system in which the encryption rules cooperate in order to encrypt the plain text. Such a case is more closely related to the practical way of encrypting messages by various cryptographical systems.

2 Encryption rules and systems

Let V be an alphabet and $P \in V^* \times V^*$ be a finite nonempty set of rewriting rules

$$P = \{x_i \longrightarrow y_i | 1 \leq i \leq k\}$$

For $w \in V^*$, the encryption of w by means of P is the set

$$w(P) = \{z_0 y_{i_1} z_1 y_{i_2} \dots z_{n-1} y_{i_n} z_n | \text{ for some } n \geq 1\}$$

where

$$\begin{array}{ll} (i) & w = z_0 x_{i_1} z_1 x_{i_2} \dots z_{n-1} x_{i_n} z_n,\ 1 \leq i_j \leq k,\ 1 \leq j \leq n, \\ (ii) & N_{\{x_p | 1 \leq p \leq k\}}(z_j) = 0,\ \text{for any } 0 \leq j \leq n \end{array}$$

Note that for $k = 1$ one obtains the parallel substitution on words:

$$w(P) = w(x_1 \Longrightarrow y_1).$$

Conventions:

- P: encryption formal key (efk);
- $x \longrightarrow y$: encryption rule;
- (w, P): encryption formal system (efs)
- w: the clear-text; the elements of $w(P)$ are called crypto-texts. (the terms are very closed to the similar ones defined in [9].

An efk P is called *efk with insertion* if P contains at least a rule of the form $e \longrightarrow y$. The efk P is called *efk with deletion* if P contains at least a rule of the form $x \longrightarrow e$. From technical reasons we restrict our work to efk without insertion.

Examples:

(i). Any monoalphabetic encryption system (Caesar, afin) is an efk;

(ii). The usual substitutions in the formal languages theory are efk with $|x_i| = 1$, for all i.

Let $w \in V^*$ and P be a efk. An encryption rule $x \longrightarrow y \in P$ is called useless on w if $N_x(w) = 0$. Obviously, for an efs (w, P) it is preferable to choose a simple efk, without useless rules (in [3] an algorithm to remove the useless rules can be found). In the following we consider that all the encryption formal systems have only useful rules. The encryption of w is deterministic if $card(w(P)) = 1$. All classical encryption systems are deterministic (and this feature seems to be a weakness of these systems).

The natural extension of the encryption of a word to a language, by means of a given set of rules P, is defined as:

$$L(P) = \bigcup_{w \in L} w(P)$$

We say that a family of languages $\mathcal{L}$ is closed under encryption if, for any finite set of rewriting rules P and any language $L \in \mathcal{L}$, we have $L(P) \in \mathcal{L}$.

3 Encryption and the Chomsky hierarchy

A full trio is a class of languages closed under arbitrary and inverse homomorphisms and intersection by regular sets.

Theorem 1 . *Any full trio is closed under encryption.*

Proof. Let $\mathcal{L}$ be a full trio and $L \subseteq V^*$ be a language in $\mathcal{L}$. For a given efk $P = \{x_i \longrightarrow y_i | 1 \leq i \leq n\}$, define the homomorphisms

$$\begin{aligned} h &: (V \cup \{c_1, c_2, \ldots, c_n\})^* \longrightarrow V^*, c_i \notin V, 1 \leq i \leq n, \\ & \quad h(a) = a, \text{ for any } a \in V, \\ & \quad h(c_i) = x_i, 1 \leq i \leq n, \end{aligned}$$

$$\begin{aligned} g &: (V \cup \{c_1, c_2, \ldots, c_n\})^* \longrightarrow V^*, \\ & \quad g(a) = a, \text{ for any } a \in V, \\ & \quad g(c_i) = y_i, 1 \leq i \leq n. \end{aligned}$$

Note that $c_1, c_2, \ldots, c_n$ are n new symbols in spite of the fact that the strings $x_1, x_2, \ldots, x_n$ may not be distinct.

We state that

$$L(P) = g(h^{-1}(L) \cap (((V^* - \{x_i | 1 \leq i \leq n\})\{c_i | 1 \leq i \leq n\})^*(V^* - \{x_i | 1 \leq i \leq n\})))$$

Indeed, the strings in $h^{-1}(L)$ are those strings of L in which some occurrences of the subwords $x_1, x_2, \ldots, x_n$ are replaced by the corresponding symbols $c_1, c_2, \ldots, c_n$. The intersection with the above regular language ensures the substitution of all occurrences of the strings $x_1, x_2, \ldots, x_n$.

From the closure properties of the family $\mathcal{L}$ it follows that $L(P) \in \mathcal{L}$. □

Corollary 1 . *The families of regular, context-free and recursively enumerable languages are closed under encryptions.*

Clearly, any family closed under encryption is closed under homomorphism. Consequently,

Corollary 2 . *The family of context-sensitive languages is closed under encryptions without deletion but it is not closed under arbitrary encryptions.*

4 Some properties of the separable systems

Let (w, P) be an encryption formal system with $w(P) \neq \emptyset$. The system is *separable* ([4]) if for any two different non-empty subsets P_1, P_2 of P, the sets $w(P_1)$ and $w(P_2)$ are disjoint.

For example, for $P = \{b \longrightarrow a, ab \longrightarrow aa\}, w = aab$, we may take $P_1 = \{b \longrightarrow a\}, P_2 = \{ab \longrightarrow aa\}$ which implies $w(P_1) = w(P_2) = \{aaa\}$, hence (w, P) is not separable.

In the sequel, we are going to provide a few simple and necessary conditions for an encryption formal system to be separable.

Theorem 2 . *Let (w, P) be a separable efs.*
1. If $x \longrightarrow y, x \longrightarrow z \in P$, then $y = z$.
2. If $x \longrightarrow x \in P$, then $P = \{x \longrightarrow x\}$.

Proof. Let $w = x_1 x x_2 x \dots x x_k$ be a decomposition of w such that $N_x(x_i) = 0$, $i = 1, \dots, k$. If $P_1 = \{x \longrightarrow y\}, P_2 = \{x \longrightarrow y, x \longrightarrow z\}$, then $x_1 y x_2 y \dots y x_k \in w(P_1) \cap w(P_2)$, hence (w, P) is not separable, contradiction. In order to prove the second assertion, assume that $x \longrightarrow x \in P$ and $P \neq \{x \longrightarrow x\}$. Take $P_1 \subseteq P - \{x \longrightarrow x\}$ and $P_2 = P_1 \cup \{x \longrightarrow x\}$. Obviously, $w(P_1) \cap w(P_2) \neq \emptyset$, hence our supposition is false. □

Theorem 3 . *Let (w, P) be a separable efs, $x \longrightarrow y \in P$ and \$ be a new symbol. Then, for any $z \in w(x \Longrightarrow \$)$, the encryption system $(z, P - \{x \longrightarrow y\})$ is separable.*

Proof. Assume that exists $z \in w(x \Longrightarrow \$)$ such that $(z, P - \{x \longrightarrow y\})$ is not separable. Let $z = u_1 \$ u_2 \$ \dots \$ u_k$; therefore exists $P_1, P_2 \subseteq P, P_1 \neq P_2$, with $z(P_1) \cap z(P_2)$ non-empty (obviously, $x \longrightarrow y$ belongs neither to P_1 nor to P_2). Let $v_1 \$ v_2 \$ \dots \$ v_k \in z(P_1) \cap z(P_2)$ and take

$$P_1' = P_1 \cup \{x \longrightarrow y\},$$

$$P_2' = P_2 \cup \{x \longrightarrow y\}.$$

Clearly, P_1', P_2' are different subsets of P.

Because $z \in w(x \Longrightarrow \$)$ it follows that $w = u_1 x u_2 x \dots x u_k$. Therefore, $v_1 y v_2 y \dots y v_k \in w(P_1') \cap w(P_2')$, contradiction. □

Remark. The reciprocal statement of the second assertion does not hold. For example, if $w = abba, P = \{a \longrightarrow b, b \longrightarrow a, ab \longrightarrow ba\}$, then (w, P) is not separable, while $(\$ba, \{a \longrightarrow b, b \longrightarrow a\})$, $(\$bb\$, \{b \longrightarrow a, ab \longrightarrow ba\})$ and $(a\$\$a, \{a \longrightarrow b, ab \longrightarrow ba\})$ are separable.

A natural question concerns an eventual link between the encryption and the parallel/sequential substitution. For separable efs such a link exists being provided by the following construction.

Let $(w, P), P = \{x_i \longrightarrow y_i | 1 \leq i \leq n\}$ be a separable efs. Take n new symbols $c_1, c_2, \ldots, c_n$ and consider the sequence

$$W_0 = \{w\}$$

$$W_{i+1} = \bigcup_{k=1}^{n} W_i(x_k \longrightarrow c_k), i \geq 0$$

Now, it is clear that

$$w(P) = h(\bigcup_{i=0}^{max\{N_{x_i}(w)|1\leq i\leq n\}} W_i(x_1 \Longrightarrow c_1)(x_2 \Longrightarrow y_2)\ldots(x_n \Longrightarrow y_n))$$

where h is a homomorphism which replaces the symbols c_i by y_i and leaves unchanged the other symbols. Of course, $max\{N_{x_i}(w)|1 \leq i \leq n\}$ is the upper bound for the number of the terms in the union above. Sometimes, $w(P)$ can be expressed as a finite union of parallel substitutions only. For instance, if for any pair (x_i, x_j) there are at most two overlapped occurrences of them, then

$$w(P) = \bigcup_{\sigma \in S_n} w(x_{\sigma(1)} \Longrightarrow y_{\sigma(1)})(x_{\sigma(2)} \Longrightarrow y_{\sigma(2)})\ldots(x_{\sigma(n)} \Longrightarrow y_{\sigma(n)})$$

where S_n is the set of all n-permutations.

Denote by $Sub(w)$ the set of all non-empty subwords of a given word w and $Sub_y(w) = \{x | w = uxv, N_y(x) = 1\}$.

Let us define $\lambda_w : P \longrightarrow 2^{Sub(w)}$, $\lambda_w(x \longrightarrow y) = Sub_x(w)$.

Example: Take $P = \{ab \longrightarrow xy, bab \longrightarrow yx\}, w = abbbab$. Then:

$$\lambda_w(ab \longrightarrow xy) = \{ab, abb, abbb, abbba, bbbab, bbab, bab\}$$

$$\lambda_w(bab \longrightarrow yx) = \{abbbab, bbbab, bbab, bab\}$$

For $w = ababab$, we have

$$\lambda_w(ab \longrightarrow xy) = \{ab, aba, bab\}$$

$$\lambda_w(bab \longrightarrow yx) = \{abab, bab, baba, ababa, babab\}$$

Lemma 1 . *Let (w, P) be a separable efs, $P = \{x_i \longrightarrow y_i | 1 \leq i \leq k\}$. Then, for any decomposition $w = z_0 x_{i_1} z_1 x_{i_2} z_2 \ldots x_{i_n} z_n$ such that $N_{\{x_1, x_2, \ldots, x_k\}}(z_i) = 0, 0 \leq i \leq n$, the relation*

$$\textit{for any } 1 \leq m \leq k \textit{ exists } 1 \leq j \leq n \textit{ with } x_m = x_{i_j},$$

holds.

Proof. Assume that there is a string x_i and a decomposition of w as above, such that x_i is not a term of that decomposition. We infer that $w(P) \cap w(P - \{x_i \longrightarrow y_i\}) \neq \emptyset$ which is a contradiction. □

Theorem 4 . *Let (w, P) be a separable efs. For any $x \longrightarrow y, x' \longrightarrow y' \in P, \lambda_w(x \longrightarrow y) \cap \lambda_w(x' \longrightarrow y')$ is non-empty.*

Proof. Due to the previous lemma, in any decomposition of w,

$$w = z_0 x_{i_1} z_1 x_{i_2} z_2 \ldots x_{i_n} z_n,$$

we can find at least one term equal to x and at least one term equal to x'. More precisely, there are $1 \leq j, k \leq n$ such that $x_{i_j} = x$ and $x_{i_k} = x'$.

Take the closest pair of the occurrences of x and x', respectively, say x_{i_j} and x_{i_k}. We can write w as either $w = w_1 x_{i_j} u x_{i_k} w_2$ or $w = w_1 x_{i_k} u x_{i_j} w_2$, therefore $\lambda_w(x \longrightarrow y) \cap \lambda_w(x' \longrightarrow y')$ contains either $x_{i_j} u x_{i_k}$ or $x_{i_k} u x_{i_j}$. □

Theorem 5 . *If (w, P) is a separable efs, then λ is an one to one mapping.*

Proof. Suppose that $\lambda_w(x \longrightarrow y) = \lambda_w(x' \longrightarrow y')$. Because $x \in \lambda(x \longrightarrow y) = \lambda(x' \longrightarrow y')$, it follows that x' is a subword of x. Analogously, x is a subword of x'. In conclusion $x = x'$. From the Theorem 2 we deduce that $y = y'$. □

5 Applications

The encryption formal system (L, P) is *partially separable* if for any $w \in L$, the efs (w, P) is separable.

The system (L, P) is *strongly separable* if the following conditions hold:

(i) (L, P) is partially separable;

(ii) for any $w_1, w_2 \in L$, and any $P_1 \neq P_2$ subsets of P, we have $w_1(P_1) \cap w_2(P_2) \neq \emptyset$.

We are going to list below two possible applications of the separable efs. Of course, other applications (in genetics, for instance) might be of interest, too.

A) *Authentication.*

Let us suppose that the data basis B uses the strongly separable system (L, P) with P large enough ($card(P) \geq 100$). A subset P' of P is earmarked to every user A of the data basis. In this way the set P' exactly identifies the user A.

Whenever A asks for access to the data basis, the authentication protocol follows the next steps:

Step 1. A asks for access by announcing the public-key $i(A)$;

Step 2. B selects at random a string $w \in L$ and sends it to A; at the same time B determines the set of valid words $w(P')$ associated to A.

Step 3. A answers with $z \in w(P')$, choosed at random, too;

Step 4. B verifies whether $z \in w(P')$; if z is a valid word, then B allows the access of A to the data basis.

The protocol can be modified in order to use a neutral agent (a judge, say C), in the following way:

Step 1. A sends its public-key $i(A)$ to B ;

Step 2. B selects at random a string $w \in L$ and sends it to A and C;

Step 3. A computes $w(P')$, chooses at random a string $z \in w(P')$ and sends it to C;

Step 4. C computes the set P' such that $z \in w(P')$ and sends it to B;

Step 5. B verifies whether P' is the earmarked set of $i(A)$, and allows the access of A to the data basis in the affirmative case.

B) *Cryptography.*

The encryption algorithm is based on the knapsack problem. Let (w, P) be a separable efs, with the rules of P arbitrarily ordered; $card(P) = n$ (suppose that n is large enough).

The plain text x is divided into blocks of equal length: $x = x_1x_2..x_p$, $|x_i| = r$, (excepting eventually the last block x_p), $5r \leq n < 5(r+1)$.

One uses a binary codification (for example $A - 00001, ..., Z - 11010$). To each block x_i a binary string of length $5r$ is associated. One constructs the subset $P' \subseteq P$ consisting of those rules which corespond to the digits 1 in the codification of x.

An arbitrary string $z \in w(P')$ is emited.

The decryption means to identify the set P'. A parser can be used in this aim.

6 References

1. A. Atanasiu, V. Mitrana - Substitution on words and languages. In *Developments in Language Theory. At the Crossroads of Mathematics, Computer Science and Biology* (A. Salomaa, G. Rozenberg, eds.), World Scientific Publishing, 1994, 51–60.

2. A. Atanasiu, V. Mitrana - Parallel substitution on words and languages. *Proc. of the* 9^{th} *ROSYCS'93, Iasi* (V. Felea, G. Ciobanu, eds.), 1993, 24–30.

3. A. Atanasiu - Substitution on Words and Languages; Applications to Cryptography. In *Mathematical Aspects of Natural and Formal Languages* (Gh. Păun ed.), World Scientific in Computer Science vol 43 (1994), 1–12.

4. A.Atanasiu - Substitution on words and languages; separable cryptation systems. *The* 10^{th} *ROSYCS'96, Iasi, 30-31 mai 1996.*

5. J. Dassow, Gh. Paun - *Regulated Rewriting in Formal Language Theory*, Springer-Verlag, Berlin, Heidelberg, 1989.

6. M. A. Harrison - *Introduction to Formal Language Theory*, Addison-Wesley, Reading Mass., 1978.

7. L. Kari - *On insertion and deletion in formal languages*, PhD. Thesis, Univ. of Turku, Finland, 1991;

8. S. La Tore, M. Napoli, D. Parente - Parallel word substitution, Fundamenta Informaticae, 27, 1(1996), 27–36.

9. A. Salomaa - *Public-key Cryptography*, Springer-Verlag Heidelberg 1990;

10. J.-C. Spehner - La reconnaisance des facteurs d'un mot dans un texte, *Theoretical Computer Science* 48 (1986), 35–52.