A Chosen Ciphertext Attack on a Public Key Cryptosystem Based on Lyndon Words

Ludovic Perret
ENSTA, UMA
32 Boulevard Victor, 75739 Paris Cedex 15, France
lperret@ensta.fr

Abstract

In this paper, we present a chosen ciphertext attack against a public key cryptosysten based on Lyndon words [7]. We show that, provided that an adversary has access to a decryption oracle, a key equivalent to the secret key can be constructed efficiently, i.e. in linear time.

1 Introduction

In this paper, we study the security of a public key cryptosystem, introduced by Siromoney and Mathew, using Lyndon words [7]. To our knowledge, the first security analysis of this scheme is due to Gonzalez-Vasco and Steinwandt [4]. They emphasize that, in some cases, one can derive from a ciphertext some informations about the corresponding plaintext. We present in this paper a stronger result. Indeed we show that, provided that an adversary has access to a decryption oracle, a key equivalent to the secret key can be constructed efficiently, i.e. in linear time. Recall that in a chosen ciphertext attack scenario, an adversary has access to a decryption oracle returning, for a ciphertext c, the plaintext m corresponding to c.

The paper is organized as follows. In section 2, we introduce the notations and the necessary material to present the system studied. In section 3, we describe the Lyndon words based scheme. In section 4, we present some properties of Lyndon words, and also describe some particular properties of the Siromoney and Mathew scheme. These properties will permit us to prove that the chosen ciphertext attack described in section 5 allows indeed recovers a key equivalent to the secret key.

2 Preliminaries

We introduce in this part the notations used throughout the paper.

2.1 Words, Free Monoids, and Morphisms

An *alphabet* is a finite nonemptyset, the elements of an alphabet will be called *letters*, and sequences of letters are *words*. In particular the *empty word*, denoted by λ, is the sequence of length zero. The set of all words over an alphabet B is denoted by B^*. If x and y are words over B, their *concatenation* xy is obtained by writing x and y one after the other. This

operation is obviously associative and the empty word λ is a neutral element for B^*. Thus, B^* is a *free monoid* generated by B with respect to the concatenation of words and with neutral element λ. The *length* of the word w, denoted by $|w|$, is the number of letters in w where each letter is counted as many times as it appears. By definition $|\lambda| = 0$.
A *monoid homomorphism* g, between two free monoids B^* and A^*, is a mapping satisfying:

$$\begin{cases} g(\lambda) = \lambda, \text{and} \\ g(uv) = g(u)g(v), \text{ for all } u, v \in B^*. \end{cases}$$

In particular, a monoid homomorphism is completely determined by the images of the letters in its domain, i.e. if $u = b_1 b_2 \cdots b_k$ with $b_i \in B$, then $g(u) = g(b_1)g(b_2)\cdots g(b_k)$.
We shall denote by $\mathrm{Hom}(B^*, A^*)$, the set of monoid homomorphisms from B^* to A^*. A bijective monoid homomorphism is called a *monoid isomorphism*, and we denote by $\mathrm{Iso}(B^*, A^*)$ the set of monoid isomomorphisms from B^* to A^*.

2.2 Lyndon words

An *ordered alphabet* $(A, \prec)$, is an alphabet together with a total order over the letters of A.
A *lexicographically ordered alphabet* $\big(A = \{a_1, a_2, \ldots, a_k\}, \prec\big)$, is an indexed alphabet $A = \{a_1, a_2, \ldots, a_k\}$, ordered in the following way $a_1 \prec a_2 \prec \cdots \prec a_k$.
A *lexicographical order* over the free monoid A^* is extended from $(A, \prec)$ in the following way. Let $(u, v) \in A^* \times A^*$, we have:

$$u \prec v \Longleftrightarrow \begin{cases} \text{there exists a nonempty word } w \in A^* \text{ such that } uw = v, \text{ or} \\ \text{there exist words } r, s, t \in A^* \text{ and } (a, b) \in A \times A, \text{ with } a \prec b, \\ \text{such that } u = ras \text{ and } v = rbt. \end{cases}$$

We shall say that two words $(u, v) \in A^* \times A^*$ are *conjugate*, if there exists $(x, y) \in A^* \times A^*$ such that $u = xy$ and $v = yx$. A *Lyndon word* is then a word, in A^*, which is strictly less than any of its conjugate. We denote by $Lyn(A)$, the set of Lyndon words over A^*, i.e.:

$$Lyn(A) = \{u \in A^* : u \prec v, \text{ for any conjugate } v \in A^* \text{ of } u\}.$$

An interesting property of Lyndon words is:

Theorem 1 (Chen, Fox, Lyndon). *Any nonempty word $w \in A^*$, can be written uniquely as a decreasing product of Lyndon words, i.e.:*

$$w = w_1 w_2 \cdots w_k, \tag{1}$$

such that for each $i, 1 \le i \le k$, $w_i \in Lyn(A)$ and $w_1 \succeq w_2 \succeq \cdots \succeq w_k$.

A proof of this theorem can be found, for example, in [5].
The decomposition (1) of w in a concatenation of Lyndon words is called *standard factorization* of w. It is to be noted that such a decomposition can be computed in linear time [1].

2.3 Thue systems

Let B be a finite alphabet. When dealing with strings, the appropriate notion of rewriting systems is that of a *Thue system*, namely a subset T of $B^* \times B^*$. We here only consider *finite*

Thue systems, i.e. T is a finite set of pairs of strings. Each pair $(u, v) \in T$ is called a *rule*. The *single step reduction relation* on B^* induced by T is defined as follows: for any $s, t \in B^*$, $s \to_T t$ iff there exist $x, y \in B^*$ and $(\ell, r) \in T$ such that $s = x\ell y$ and $t = xry$. We shall call *Thue congruence*, denoted by $\overset{*}{\leftrightarrow}_T$, the reflexive, symmetric, and transitive closure of $\to_T$. Two words $u, v \in B^*$ are *congruent with respect to* T iff $u \overset{*}{\leftrightarrow}_T v$, and the *congruence class* of a word u is the set $\{v \in A^* : u \overset{*}{\leftrightarrow}_T v\}$.

3 Description of the Siromoney and Mathew cryptosystem

In this part, we describe the principle of the public key cryptosystem based on Lyndon words proposed in [7]. For further details concerning it, we refer the reader to [7].

Key generation.
Let $\Sigma = \{\sigma_1, \ldots, \sigma_n\}$ be the plaintext alphabet, and $\big(A = \{a_1, a_2, \ldots, a_k\}, \prec\big)$, be a lexicographically ordered alphabet. Moreover, let $L_1, \ldots, L_n$ be disjoint subsets of $Lyn(A)$, B be another alphabet of cardinality much greater than that of A, and $g \in \mathrm{Hom}(B^*, A^*)$ mapping every letter of B either to a letter of A or to the empty word.
Let $E = \{w_1, \ldots, w_m\} \subset g^{-1}(\cup_{1 \le i \le n} L_i)$, such that:

$$|w_i| < |w_j| \Longrightarrow g(w_i) \prec g(w_j), \text{ for all } i, j, 1 \le i, j \le m.$$

Moreover, let $T \subset B^* \times B^*$ be a Thue system such that:

$$\forall (u, v) \in T, g(u) = g(v).$$

Finally, for each $i, 1 \le i \le n$, we set $E_i = g^{-1}(L_i) \cap E$. In this setting, the public key is $\big(\Sigma, B, T, \{E_i\}_{1 \le i \le n}\big)$ and the secret key is $\big(A, g, \{L_i\}_{1 \le i \le n}\big)$.

Encryption. To encrypt a message $m = z_1 \cdots z_p \in \Sigma^*$, we proceed as follows:

> For each letter z_j of m, we select $e_j \in E_i$, with i being such that $\sigma_i = z_j$, the e_j's satisfying $|e_j| \ge |e_{j+1}|$, for all j, $1 \le j < p$. Let $e = e_1 e_2 \cdots e_p$.
> The ciphertext $c \in B^*$ is then a word in the congruence class of e, i.e. $c \overset{*}{\leftrightarrow}_T e$.

Decryption. To decrypt a ciphertext $c \in B^*$, we:

> Compute the standard factorization of $g(c)$. Let $e_1 \succeq e_2 \succeq \cdots \succeq e_p$ be words of $Lyn(A)$ such that $g(c) = e_1 e_2 \cdots e_p$.
> For each word e_j of $g(c)$, we select a letter $m_j \in \Sigma$ such that $m_j = \sigma_i$, if $e_j \in L_i$.
> The plaintext is then the word $m = m_1 m_2 \cdots m_p$.

4 Properties

In this section, we present some properties of Lyndon words as well as properties of the Siromoney and Mathew scheme described in section 3. These properties will permit to present, in section 5, a cryptanalysis of this cryptosystem.

4.1 Some properties of Lyndon words

Let g be a monoid homomorphism from B^* to A^*. We start this part by giving a simple link between the Lyndon words of A^* and the Lyndon words lying in the image of B^* by g,

Lemma 4.1. *Let $g \in \mathrm{Hom}(B^*, A^*)$, and $\big(A = \{a_1, a_2, \ldots, a_k\}, \prec\big)$ be a lexicographically ordered alphabet. Moreover, let $g(B)$ be the image of B by g, i.e. $g(B) = \{g(b) : b \in B\} \subseteq A$. Let $\prec_{g(B)}$ be the total order on $g(B)$ induced by $\prec$, w.l.o.g. we set $g(B) = \{a_{i_1}, a_{i_2}, \ldots, a_{i_\ell}\}$, with $1 \leq i_1 \leq i_2 \leq \cdots \leq i_\ell \leq k$ and $a_{i_1} \prec_{g(B)} \cdots \prec_{g(B)} a_{i_\ell}$, then:*

$$Lyn(A) \cap g(B^*) = Lyn\big(g(B)\big).$$

with $g(B^*) = \{g(b) : b \in B^*\} = g(B)^*$.

Proof. First remark that any conjugate v of a word $u \in g(B^*)$ must also lie in $g(B^*)$. Thus, $Lyn(A) \cap g(B^*) = \big\{u \in g(B^*) : u \prec v, \text{ for any conjugate } v \in g(B^*) \text{ of } u\big\}$. Moreover, one can see at once that for all $(u, v) \in g(B^*) \times g(B^*)$, $u \prec v \Longleftrightarrow u \prec_{g(B)} v$. Finally:

$$\begin{aligned} Lyn(A) \cap g(B^*) &= \big\{u \in g(B^*) : u \prec_{g(B)} v, \text{ for any conjugate } v \in g(B^*) \text{ of } u\big\} \\ &= Lyn\big(g(B)\big). \end{aligned}$$

□

Now, let $\big(A, \prec\big)$ and $\big(A', \prec_{A'}\big)$ be two lexicographically ordered alphabets, and i be a monoid isomomorphism from A^* to A'^*. We prove now that, if i is suitably chosen, then i preserves the orders $\prec$ and $\prec_{A'}$. That is, stated more formally:

Lemma 4.2. *Let $\big(A = \{a_1, a_2, \ldots, a_k\}, \prec\big)$ and $\big(A' = \{a'_1, a'_2, \ldots, a'_k\}, \prec_{A'}\big)$ be lexicographically ordered alphabets. Moreover, let $i \in \mathrm{Iso}(A^*, A'^*)$, such that:*

$$i(a_1) \prec_{A'} i(a_2) \prec_{A'} \cdots \prec_{A'} i(a_k).$$

Then, for all $(u, v) \in A^* \times A^*$, we have:

$$u \prec v \Longleftrightarrow i(u) \prec_{A'} i(v).$$

Proof. By definition, $u \prec v$ implies that:

$$\begin{cases} \exists \text{ a nonempty word } w \in A^* \text{ such that } uw = v, \text{ or} \\ \exists r, s, t \in A^* \text{ and } (a, b) \in A \times A, \text{ with } a \prec b, \text{ such that } u = ras \text{ and } v = rbt. \end{cases}$$

If $uw = v$, for a nonempty word $w \in A^*$, then $i(u)i(w) = i(v)$, i.e. $i(u) \prec_{A'} i(v)$. Otherwise, there exist words $r, s, t \in A^*$ and $(a, b) \in A \times A$, with $a \prec b$, such that $u = ras$ and $v = rbt$. We then deduce that $i(u) = i(r)i(a)i(s)$ and $i(v) = i(r)i(b)i(t)$. Since $a \prec b$, the definition of i implies that $i(a) \prec_{A'} i(b)$, i.e. $i(u) \prec_{A'} i(v)$. The proof of the "only if" part can be done similarly.

□

From this last lemma, we deduce that the sets of Lyndon words of A^* and A'^* are isomorphic.

Proposition 1. *Let $\big(A = \{a_1, a_2, \ldots, a_k\}, \prec\big)$ and $\big(A' = \{a'_1, a'_2, \ldots, a'_k\}, \prec_{A'}\big)$ be two lexicographically ordered alphabets. Moreover, let $i \in \mathrm{Iso}(A^*, A'^*)$ be such that $i(a_1) \prec_{A'} \cdots \prec_{A'} i(a_k)$, then:*

$$i\big(Lyn(A)\big) = Lyn(A'),$$

with $i\big(Lyn(A)\big) = \big\{i(u) : u \in Lyn(A)\big\}$.

Proof. We first remark that, for all $u \in A^*$:

$$v \text{ is a conjugate of } u \Longleftrightarrow i(v) \text{ is a conjugate of } i(u).$$

Indeed, if v is a conjugate of u, we know that there exists $(x, y) \in A^* \times A^*$, such that $u = xy$ and $v = yx$. Thus, $i(u) = i(x)i(y)$ and $i(v) = i(y)i(x)$, i.e. $i(v)$ is a conjugate of $i(u)$.
Now, if $i(v)$ is a conjugate of $i(u)$, we know that there exists $(x', y') \in A'^* \times A'^*$ such that $i(u) = x'y'$ and $i(v) = y'x'$. Let $(x, y) \in A^* \times A^*$ be the unique pair such that $i(x) = x'$ and $i(y) = y'$. Since $i \in \mathrm{Iso}(A^*, A'^*)$, we have $u = i^{-1}(i(u)) = i^{-1}(x'y') = xy$ and $v = i^{-1}(i(v)) = i^{-1}(y'x') = yx$, i.e. v is a conjugate of u.
We can now conclude the proof of this proposition. Indeed, let $u' \in i\big(Lyn(A)\big)$, i.e. there exists $u \in Lyn(A)$ such that $u' = i(u)$. By definition all the conjugates v of u are such that $u \prec v$. Thus, by lemma 4.2, all the conjugates $i(v)$ of $u' = i(u)$ are such that $u' \prec_{A'} i(v)$. Therefore $u' \in Lyn(A')$, since all the conjugates $v' \in A'^*$ of u' are of the form $v' = i(v)$, for some conjugate $v \in A^*$ of u.
Finally, let $u' \in Lyn(A')$. Since $i \in \mathrm{Iso}(A^*, A'^*)$, there exists a unique $u \in A^*$ such that $u' = i(u)$. By definition all the conjugates v' of $u' = i(u)$ are such that $u' \prec v'$. Since for all $v' \in A'^*$, there exists a unique $v \in A^*$ such that $i(v) = v'$, and according to lemma 4.2, we get that $u \prec_A v$. Therefore $u \in Lyn(A)$, since all the conjugates $v \in A^*$ of u are of the form $v = i^{-1}(v')$, for some conjugate $v' \in A'^*$ of u'. Thus $u' \in i\big(Lyn(A)\big)$. □

4.2 Some properties of the scheme based on Lyndon words

In the remaining of this part, $\big(A = \{a_1, a_2, \ldots, a_k\}, g, \{L_i\}_{1 \le i \le n}\big)$ will always be a secret key corresponding to a public key $\big(\Sigma, B, T, \{E_i\}_{1 \le i \le n}\big)$ of the scheme described in section 3.

Proposition 2. *Let $g(B)$ be the image of B by g, i.e. $g(B) = \{g(b) : b \in B\} \subseteq A$, and $\prec_{g(B)}$ be the total order on $g(B)$ induced by $\prec$, w.l.o.g. we set $g(B) = \{a_{i_1}, a_{i_2}, \ldots, a_{i_\ell}\}$, with $1 \le i_1 \le i_2 \le \cdots \le i_\ell \le k$ and $a_{i_1} \prec_{g(B)} \cdots \prec_{g(B)} a_{i_\ell}$. Finally, let $g' \in \mathrm{Hom}\big(B^*, g(B^*)\big)$ be derived from g by restricting its image to $g(B^*)$.*
If for each $i, 1 \le i \le n$, we denote by L'_i the restriction of L_i to words of $g(B^*)$, i.e. $L'_i = L_i \cap g(B^*)$, then $\big(g(B), g', \{L'_i\}_{1 \le i \le n}\big)$ is a key equivalent to the secret key $\big(A, g, \{L_i\}_{1 \le i \le n}\big)$, in the sense that any message encrypted with $\big(\Sigma, B, T, \{E_i\}_{1 \le i \le n}\big)$ can be successfully decrypted using $\big(g(B), g', \{L'_i\}_{1 \le i \le n}\big)$.

Proof. Let $E = \{w_1, \ldots, w_m\} \subset g^{-1}(\cup_{1 \le i \le n} L_i)$. In order to show that $\big(g(B), g', \{L'_i\}_{1 \le i \le n}\big)$ is a key equivalent to the secret key $\big(A, g, \{L_i\}_{1 \le i \le n}\big)$, it is sufficient to prove that:

$$\begin{cases} i)\ g'(E_i) \subseteq L'_i, \text{ for all } i, 1 \le i \le n. \\ ii)\ \{L'_i\}_{1 \le i \le n} \text{ are disjoint subsets of } Lyn\big(g(B')\big). \\ iii)\ |w_i| < |w_j| \Longrightarrow g'(w_i) \prec_{g(B)} g'(w_j), \text{ for all } i, j, 1 \le i, j \le m. \\ iv)\ \forall (u, v) \in T, g'(u) = g'(v). \end{cases}$$

Indeed, let $c \in B^*$ be a ciphertext obtained with the encryption process described in section 3. Conditions $i)$ and $iv)$ guarantee that $g(c)$ lies in $L'_1 \times \cdots \times L'_n$. Moreover conditions $ii)$ and $iii)$ guarantee that the standard factorization of $g(c)$ permits to recover, as explained in section 3, the plaintext corresponding to c. We shall now prove $i) - iv)$.
$i)$ By construction, $g(E_i) \subseteq L_i$, for all $i, 1 \le i \le n$. Since $g(E_i) \cap g(B^*) = g'(E_i)$, we get that

$g'(E_i) \subseteq L_i \cap g(B^*) = L'_i$, for all $i, 1 \leq i \leq n$.
ii) Since, for each $i, 1 \leq i \leq n, L'_i \subseteq L_i$, the fact that the L_i's are disjoint subsets of $Lyn(A)$ implies that the L'_i's are also disjoint subsets of $Lyn(A)$. Finally, since for each $i, 1 \leq i \leq n, L'_i \subseteq Lyn(A) \cap g(B^*)$, we obtain, according to lemma 4.1, that the L'_i's are disjoint subsets of $Lyn\big(g(B)\big)$.
iii) For all $(u, v) \in B^* \times B^*$, one can see at once that $\big(g'(u), g'(v)\big) = \big(g(u), g(v)\big) \in g(B^*) \times g(B^*)$. Moreover, we have $g(u) \prec g(v) \iff g'(u) \prec_{g(B)} g'(v)$. Thus, since $|w_i| < |w_j|$ implies that $g(w_i) \prec g(w_j)$, for all $i, j, 1 \leq i, j \leq m$, then:

$$|w_i| < |w_j| \Longrightarrow g'(w_i) \prec_{g(B)} g'(w_j), \text{ for all } i, j, 1 \leq i, j \leq m.$$

iv) Obvious, as soon as we remark that $\forall (u, v) \in B^* \times B^*, \big(g'(u), g'(v)\big) = \big(g(u), g(v)\big)$. □

Remark 4.1. *In the sequel, we will always suppose that $g(B) = A$. We would like to emphasize that such a hypothesis is very natural, since it means that all the letters of the secret alphabet A are used during the decryption process. Moreover, it is also actually not restrictive at all, since according to proposition 2, a key equivalent to the secret key $\big(A, g, \{L_i\}_{1\leq i\leq n}\big)$ can be constructed by restricting the image of g to $g(B^*)$.*

The next result is particularly interesting. Indeed, we prove that a key equivalent to the secret key can be easily constructed from a suitable monoid isomorphism.

Theorem 2. *Let $\big(A' = \{a'_1, a'_2, \ldots, a'_k\}, \prec_{A'}\big)$ be a lexicographically ordered alphabet, and $i \in \mathrm{Iso}(A^*, A'^*)$ be such that $i(a_1) \prec_{A'} i(a_2) \prec_{A'} \cdots \prec_{A'} i(a_k)$.
If we set $g' = i \circ g$, then $\big(A', g', \{i(L_i)\}_{1\leq i\leq n}\big)$ is a key equivalent to the secret key $\big(A, g, \{L_i\}_{1\leq i\leq n}\big)$, in the sense that any message encrypted with $\big(\Sigma, B, T, \{E_i\}_{1\leq i\leq n}\big)$ can be successfully decrypted using $\big(A', g', \{i(L_i)\}_{1\leq i\leq n}\big)$.*

Proof. Let $E = \{w_1, \ldots, w_m\} \subset g^{-1}(\cup_{1\leq i\leq n} L_i)$. As previously, it is sufficient to prove that:

$$\begin{cases} i)\ g'(E_i) \subseteq i(L_i), \text{ for all } i, 1 \leq i \leq n. \\ ii)\ \{i(L_i)\}_{1\leq i\leq n} \text{ are disjoint subsets of } Lyn(A'). \\ iii)\ |w_i| < |w_j| \Longrightarrow g'(w_i) \prec_{A'} g'(w_j), \text{ for all } i, j, 1 \leq i, j \leq m. \\ iv)\ \forall (u, v) \in T, g'(u) = g'(v). \end{cases}$$

i) By construction, $g(E_i) \subseteq L_i$, for all $i, 1 \leq i \leq n$. Thus $(i \circ g)(E_i) = g'(E_i) \subseteq i(L_i)$, for all $i, 1 \leq i \leq n$.
ii) By construction, $L_i \subseteq Lyn(A)$, for all $i, 1 \leq i \leq n$. Thus, according to proposition 1, $i(L_i) \subseteq Lyn(A')$, for all $i, 1 \leq i \leq n$. Finally, since $i \in \mathrm{Iso}(A^*, A'^*)$, the fact that the L_i's are disjoint subsets obviously implies that the $i(L_i)$'s are also disjoint subsets.
iii) As $|w_i| < |w_j|$ implies that $g(w_i) \prec g(w_j)$, for all $i, j, 1 \leq i, j \leq m$, then thanks to lemma 4.2, we have for all $i, j, 1 \leq i, j \leq m$:

$$|w_i| < |w_j| \Longrightarrow g(w_i) \prec g(w_j) \iff (i \circ g)(w_i) = g'(w_i) \prec_{A'} g'(w_j) = (i \circ g)(w_j).$$

iv) For all $(u, v) \in T, g(u) = g(v)$, thus $\forall (u, v) \in T, (i \circ g)(u) = g'(u) = g'(v) = (i \circ g)(v)$. □
Informally, this last result means that the secret key is independent of the choice of the alphabet A.

In a chosen ciphertext attack scenario, an adversary has access to a decryption oracle, denoted here by $\mathcal{O}$, returning for a ciphertext $c \in B^*$, the plaintext $m \in \Sigma^*$ corresponding to c, i.e. $\mathcal{O}(c) = m$. We study now what kind of information about the secret key can be obtained from such an oracle.

Lemma 4.3. *Let $c \in B^*$ be a ciphertext encrypting a plaintext $m \in \Sigma^*$. For all $b \in B$, we have $g(b) = \lambda \Longrightarrow \mathcal{O}(cb) = m$.*

Proof. $g(b) = \lambda$ implies that $g(cb) = g(c)$, thus $g(cb)$ and $g(c)$ have the same standard factorization. Therefore, cb and c encrypt the same plaintext m. $\square$

By choosing more carefully the challenges submitted to the oracle, we obtain:

Proposition 3. *Let $\left(A, g, \{L_i\}_{1\leq i\leq n}\right)$ be a secret key corresponding to a public key $\left(\Sigma = \{\sigma_1, \ldots, \sigma_n\}, B, T, \{E_i\}_{1\leq i\leq n}\right)$ of the system described in section 3.*
Let $b \in B$, $i \in \{1, \ldots, n\}$, and $\tilde{e}_i \in E_i$ with $|\tilde{e}_i| \geq |e_i|$, for all $e_i \in E_i$. Finally, let $\tilde{c}_i \in B^*$ be in the congruence class of $\tilde{e}_i b$, i.e. $\tilde{c}_i \overset{*}{\leftrightarrow}_T \tilde{e}_i b$. We then have:

$$g(b) = \lambda \Longleftrightarrow \mathcal{O}(\tilde{c}_i) = \sigma_i.$$

Proof. By construction, we know that $g(\tilde{c}_i) = g(\tilde{e}_i b)$. Therefore, $g(b) = \lambda$ implies that $g(\tilde{c}_i) = g(\tilde{e}_i)$. Thus $g(\tilde{c}_i)$ and $g(\tilde{e}_i)$ have the same standard factorization, and so $\tilde{c}_i$ and $\tilde{e}_i$ encrypt the same plaintext, which is $\sigma_i \in \Sigma$.
To prove the $\Longleftarrow$ part of this equivalence, we remark that due to the particular constraints of the system, we have for all $e_i \in E_i$:

$$|\tilde{e}_i| \geq |e_i| \Longrightarrow g(\tilde{e}_i) \succeq g(e_i).$$

Suppose now, by contradiction, that $g(b) \neq \lambda$. Since $\mathcal{O}(\tilde{c}_i) = \sigma_i$, we know that there exists $e'_i \in E_i$ such that $g(\tilde{c}_i) = g(e'_i)$. Indeed, the only way to encrypt $\sigma_i \in \Sigma$ is to choose an element in the congruence class of an $e'_i \in E_i$. Moreover, since $g(\tilde{c}_i) = g(\tilde{e}_i b)$, we get that $g(\tilde{e}_i)g(b) = g(e'_i)$, i.e. $g(\tilde{e}_i) \prec g(e'_i)$, contradicting the definition of $\tilde{e}_i$. $\square$

Thus, to test if $g(b) = \lambda$, it suffices to choose an i, take $\tilde{e}_i$ and submit $\tilde{c}_i \overset{*}{\leftrightarrow}_T \tilde{e}_i b$ to $\mathcal{O}$.
We explain in the next section how to exploit these informations.

5 Description of the Chosen ciphertext attack

We can now present a chosen ciphertext attack against the system of Siromoney and Mathew (described in section 3). Let $\left(A = \{a_1, a_2, \ldots, a_k\}, g, \{L_i\}_{1\leq i\leq n}\right)$ be a secret key corresponding to a public key $\left(\Sigma = \{\sigma_1, \ldots, \sigma_n\}, B, T, \{E_i\}_{1\leq i\leq n}\right)$ of this system. The principle of the attack is first to submit suitable challenges to a decryption oracle $\mathcal{O}$. These challenges will permit us to obtain the letters of B mapped, by (the secret morphism) g, to the empty word.
Note that the idea of using an oracle, to obtain information about the secret key, has been already successfully used, in [3] and [2], to attack other schemes based on the word problem. Due to the constraint imposed on the public Thue system T, we can then easily obtain the letters of B having the same images in A by g. This partial knowledge of g allows to construct a morphism, say $\tilde{g}$, which is isomorphic to g. Finally, using properties of section 4, we prove that such a $\tilde{g}$ is sufficient to construct a key equivalent to the secret key.
We describe now precisely the different steps of the attack.

Step 1. How to recover the letters mapped to the empty word ?
In order to test whether a letter $b \in B$ maps to the empty word by g, an attacker starts by randomly selecting $i \in \{1, \dots, n\}$, and $\tilde{c}_i \overset{*}{\leftrightarrow}_T \tilde{e}_i b$, with $\tilde{e}_i$ being defined as in proposition 3. The attacker then submits $\tilde{c}_i$ to a decryption oracle. According to proposition 3, we know that if $\mathcal{O}(\tilde{c}_i) = \sigma_i$ then $g(b) = \lambda$, and otherwise $g(b) \neq \lambda$. Thus, with $|B|$ queries to a decryption oracle, we exactly obtain the set of letters of B mapped to the empty word by g.
We shall denote this set by:

$$C_\lambda = \{b \in B : g(b) = \lambda\}.$$

Step 2. How to recover the letters having the same images ?
In order to obtain the letters of B having the same images in A by g, we first observe that:

Lemma 5.1. *For all $(u, v) \in T$, let (u', v') be the pair obtained from (u, v) by removing the letters of u (resp. v) lying in C_λ. The Thue system $T' = \{(u', v') : (u, v) \in T\}$, obtained from T with this process, is a length-preserving Thue system, i.e. $|u'| = |v'|$, for all $(u', v') \in T'$.*

Proof. For all $(u', v') \in T'$, we have $g(u') = g(u) = g(v) = g(v')$. Thus, since $|u'| = |g(u')|$ and $|v'| = |g(v')|$, we get that $|u'| = |g(u')| = |g(v')| = |v'|$. □

From this simplified Thue system T', we can easily obtain the letters of $B_\lambda = B \setminus C_\lambda$ having the same images in A by g. Indeed, as $g(u') = g(u) = g(v) = g(v')$, we deduce that for all $(u' = u'_1 u'_2 \cdots u'_k, v' = v'_1 v'_2 \cdots v'_k) \in T'$:

$$g(u'_i) = g(v'_i), \text{for all } i, 1 \leq i \leq k.$$

Thus, we obtain for[1] all $b \in B_\lambda = B \setminus C_\lambda$, the set of letters $b' \in B \setminus \{b\}$, such that $g(b') = g(b)$. We shall denote this set by:

$$C_b = \{b' \in B : g(b') = g(b)\}, \text{ for all } b \in B_\lambda.$$

Step 3. How to recover a key equivalent to the secret key?
We will now show how to construct, from the sets C_λ and $\{C_b\}_{b \in B_\lambda}$, a key equivalent to the secret key. To do so, let $B_{dis} \subseteq B$ be defined as follows:

$$\begin{cases} \cup_{b \in B_{dis}} C_b = B_\lambda, \text{and} \\ C_b \not\subset \cup_{b' \in B_{dis} \setminus \{b\}} C_{b'}, \text{ for all } b \in B_{dis}. \end{cases}$$

This alphabet is simply constructed by assigning a unique representative to each set C_b, $b \in B_\lambda$. We stress that, since we have supposed $g(B) = A$ (see remark 4.1), then $|A| = |B_{dis}|$. Indeed, for each $a \in A$, there exists a unique $b \in B_{dis}$ such that $g(b) = a$. Thus, we can w.l.o.g index the letters of B_{dis} in the following way: $B_{dis} = \{b_1, \dots, b_k\}$, with $g(b_i) = a_i$, for all $i, 1 \leq i \leq k$. Moreover, we define an ordered alphabet $\left(A_B = \{\alpha_b : b \in B_{dis}\}, \prec_{A_B}\right)$, with $\alpha_{b_1} \prec_{A_B} \alpha_{b_2} \prec_{A_B} \cdots \prec_{A_B} \alpha_{b_k}$. (Note that $|A| = |B_{dis}| = |A_B|$). Finally, let $\tilde{g} \in \mathrm{Hom}(B^*, A_B^*)$ be defined in the following way:

$$\begin{cases} \forall b' \in C_\lambda \cup \{\lambda\}, \ \tilde{g}(b') = \lambda, \text{ and} \\ \forall b' \in B_\lambda, \ \tilde{g}(b') = \alpha_b, \text{ with } b \in B_{dis} \text{ s.t. } g(b) = g(b'). \end{cases}$$

[1]More precisely, for all the letters used in the rules of T'. If we denote by $\mathrm{alph}(T')$ the set of letters used in the rules of T', we can w.l.o.g. suppose that $B = C_\lambda \cup \mathrm{alph}(T')$. Otherwise, there exists letters in B which are useless, i.e. not used in the encryption (resp. decryption) process. Note that our attack can be very easily adapted in this case. Finally, we mention that in such a setting, information about the plaintext can be derived from the ciphertext (see. [4]).

In fact such a $\tilde{g}$ is equal, up to an isomorphism, to g. Moreover, this isomorphism preserves the orders $\prec$ and $\prec_{A_B}$. Indeed:

Theorem 3. *Let $g \in \mathrm{Hom}(B^*, A^*)$ be a secret morphism as defined in section 3. Let $\big(A_B, \prec_{A_B}\big)$ and $\tilde{g} \in \mathrm{Hom}(B^*, A_B^*)$ being defined as previously. There exists $i \in \mathrm{Iso}(A_B^*, A^*)$, such that $g = i \circ \tilde{g}$, and $i^{-1}(a_1) \prec_{A_B} i^{-1}(a_2) \prec_{A_B} \cdots \prec_{A_B} i^{-1}(a_k)$.*

Proof. We give a constructive proof of this theorem. Let $i \in \mathrm{Hom}(A_B^*, A^*)$ be such that:

$$\begin{cases} i(\lambda) = \lambda, \text{ and} \\ i(\alpha_b) = g(b), \text{ for all } \alpha_b \in A_B. \end{cases}$$

We first prove that such an i lies in $\mathrm{Iso}(A_B^*, A^*)$. To do so, let $j \in \mathrm{Hom}(A^*, A_B^*)$ be such that:

$$\begin{cases} j(\lambda) = \lambda, \text{ and} \\ \forall a \in A, j(a) = \alpha_b, \text{ with } b \in B_{dis} \text{ such that } g(b) = a. \end{cases}$$

Let us prove now that $j \circ i$ is equal to the the identity mapping. We stress that it is sufficient to show that the equality hold for letters, i.e.:

$$\begin{cases} \lambda = (i \circ j)(\lambda), \text{ and} \\ a = (i \circ j)(a), \text{for all } a \in A. \end{cases}$$

Since we have supposed $g(B) = A$, then for all $a \in A, \exists! b \in B_{dis}$, such that $g(b) = a$. Thus:

$$(i \circ j)(a) = i\big(j(a)\big) = i(\alpha_b) = g(b) = a, \text{ for all } a \in A.$$

In addition $(i \circ j)(\lambda) = i\big(j(\lambda)\big) = i(\lambda) = \lambda$. Similarly, in order to show that $j \circ i$ is equal to the identity mapping, it is sufficient to have $\lambda = (j \circ i)(\lambda)$, and $\alpha_b = (j \circ i)(\alpha_b)$, for all $\alpha_b \in A_B$. For all $\alpha_b \in A_B$, we have:

$$(j \circ i)(\alpha_b) = j\big(i(\alpha_b)\big) = j\big(g(b)\big) = \alpha_b.$$

Finally, $(j \circ i)(\lambda) = j\big(i(\lambda)\big) = j(\lambda) = \lambda$. Thus, we conclude that $i \in \mathrm{Iso}(A_B^*, A^*)$, since there exists $j \in \mathrm{Hom}(A^*, A_B^*)$ such that both $i \circ j$ and $j \circ i$ are equal to the identity mapping.

Let us prove that $g = i \circ \tilde{g}$. As previously, we show that $g(\lambda) = (i \circ \tilde{g})(\lambda)$, and $g(b') = (i \circ \tilde{g})(b')$, for all $b' \in B$. We first remark that, for all $b' \in C_\lambda$:

$$(i \circ \tilde{g})(b') = i\big(\tilde{g}(b')\big) = i(\lambda) = \lambda = g(b').$$

Now, let $b' \in B_\lambda$, we have $(i \circ \tilde{g})(b') = i\big(\tilde{g}(b')\big) = i(\alpha_b)$, with $b \in B_{dis}$ s.t. $g(b) = g(b')$. Then:

$$(i \circ \tilde{g})(b') = i(\alpha_b) = g(b) = g(b'), \text{ for all } b' \in B_\lambda.$$

Moreover, $(i \circ \tilde{g})(\lambda) = i\big(\tilde{g}(\lambda)\big) = i(\lambda) = \lambda = g(\lambda)$.

Finally, in order to prove that $i^{-1}(a_1) \prec_{A_B} i^{-1}(a_2) \prec_{A_B} \cdots \prec_{A_B} i^{-1}(a_k)$, we simply remark that $i^{-1}(a_i) = j(a_i) = \alpha_{b_i}$, for all $i, 1 \leq i \leq k$ and $\alpha_{b_1} \prec_{A_B} \alpha_{b_2} \prec_{A_B} \cdots \prec_{A_B} \alpha_{b_k}$. □

Finally, we show that knowledge of $\tilde{g}$ and i is sufficient to construct new secret key:

Corollary 5.1. *Let* $(A, g, \{L_i\}_{1\leq i\leq n})$ *be the secret key corresponding to a public key* $(\Sigma, B, T, \{E_i\}_{1\leq i\leq n})$*. Then* $(A_B, \tilde{g}, \{i^{-1}(L_i)\}_{1\leq i\leq n})$ *is a key equivalent to the secret key* $(A, g, \{L_i\}_{1\leq i\leq n})$*, in the sense that any message encrypted with* $(\Sigma, B, T, \{E_i\}_{1\leq i\leq n})$*, can be decrypted using* $(A_B, \tilde{g}, \{i^{-1}(L_i)\}_{1\leq i\leq n})$*.*

Proof. According to theorem 3, we have $j \circ g = \tilde{g}$, and $j(a_1) \prec_{A_B} j(a_2) \prec_{A_B} \cdots \prec_{A_B} j(a_k)$, with $j = i^{-1} \in \mathrm{Iso}(A^*, A_B^*)$, concluding the proof according to theorem 2. □

We emphasize that this equivalent key is obtained by simply sending $|B|$ challenges to a decryption oracle. The overall complexity of our attack is then linear in the length of the public alphabet B.

6 Conclusion

We have presented in this paper an efficient chosen ciphertext attack against the scheme of Siromoney and Mathew based on Lyndon words [7].
Finally, let $(\Sigma, B, T, \{E_i\}_{1\leq i\leq n})$ be as in section 3. We call SM problem, the general problem of finding - if any - a tuple $(A, g, \{L_i\}_{1\leq i\leq n})$, such that:

$$\begin{cases} i)\ g(E_i) \subseteq L_i, \text{ for all } i, 1 \leq i \leq n. \\ ii)\ \{L_i\}_{1\leq i\leq n} \text{ are disjoint subsets of } Lyn(A). \\ iii)\ |w_i| < |w_j| \Longrightarrow g(w_i) \prec g(w_j), \text{ for all } i, j, 1 \leq i, j \leq m. \\ iv)\ \forall (u, v) \in T, g(u) = g(v). \end{cases}$$

with $E = \{w_1, \ldots, w_m\} \subset g^{-1}(\cup_{1\leq i\leq n} L_i)$.
To our knowledge this problem, which correspond in the scheme studied here to the recovering of a secret key, has not been investigated. We guess that the techniques described in this paper can be used to solve this problem (without using a decryption oracle).

References

[1] J.-P. Duval, *"Factoring words over an ordered alphabet"*, J. Algorithms, 4 (1983), pp. 363–381.

[2] M.I. González-Vasco, R. Steinwandt,*"Pitfalls in public-key systems based on free partially commutative monoids an groups."* Cryptology ePrint archive 2004/012.

[3] M.I. González-Vasco, R. Steinwandt, *"A Reaction Attack on a Public Key Cryptosystem Based on the Word Problem"*, Applicable Algebra Engineering, Communication and Computing, 14(5), 2004, pp.335-340.

[4] M. I. González-Vasco and R. Steinwandt,*"Clouds over a Public Key Cryptosystem Based on Lyndon Words"*, Inform. Process. Lett., vol. 80, pp. 239-242, 2001.

[5] M. Lothaire,*"Combinatorics on Words"*, Addison-Wesley, 1983.

[6] P. S. Novikov,*"On the algorithmic unsolvability of the word problem in group theory"*, Trudy Mat. Inst. Steklov 44 (1955), pp.1-143.

[7] R. Siromoney, L. Mathew, *"A Public Key Cryptosystem Based on Lyndon Words"*, Inform. Process. Lett. 35(1): 33-36 (1990).