Decrypting the Hill Cipher via a Restricted Search over the Text-Space

Florent Dewez, Valentin Montmirail

Decrypting the Hill Cipher via a Restricted Search over the Text-Space

Florent Dewez
INRIA, Modal team-project
Lille–Nord Europe research center, France
florent.dewez@inria.fr

Valentin Montmirail
Université Côte d'Azur
I3S, CNRS, Nice, France
vmontmirail@i3s.unice.fr

Abstract

Developed by L. S. Hill in 1929, the Hill cipher is a polygraphic substitution cipher based on matrix multiplication. This cipher has been proved vulnerable to many attacks, especially the known-plaintext attack, while only few ciphertext-only attacks have been developed. The aim of our work is to study a new kind of ciphertext-only attack for the Hill cipher which is based on a restricted search over an explicit set of texts, called orbits, and not on a search over the key-space; it is called Orbit-Based Attack (OBA). To explain in a convenient setting this approach, we make use of basic notions from group action theory; we present then in details an algorithm for this attack and finally results from experiments. We demonstrate experimentally that this new method can be efficient in terms of time-execution and can even be faster on average than the classical Brute-Force Attack in the considered settings.

1 Introduction

The Hill cipher is a relatively old polygraphic substitution cipher based on linear algebra and invented by Lester S. Hill in 1929 (Hill, 1929; Hill, 1931). For a plaintext of M characters composed of m blocks of n characters in an alphabet with p elements, the Hill cipher considers each block as an element of the vector space $(\mathbb{Z}_p)^n$ and multiplies each of them by the same $n \times n$ invertible matrix, called the secret key, to compute in output the whole ciphertext.

Because of its linear nature, it suffers mainly from the known-plaintext attack, *i.e.* attacker can obtain one or more plaintexts and their corresponding ciphertexts, as stated in (Stinson, 2002). This weakness has lead to many modifications of the original version of this cipher; see for instance (Ismail et al., 2006; Mahmoud and Chefranov, 2009; Toorani and Falahati, 2009; Toorani and Falahati, 2011). Regarding the ciphertext-only attack, *i.e.* the attacker is assumed to have access only to a set of ciphertexts, it is said in (Wagstaff, 2002; Stinson, 2002) that performing a ciphertext-only attack on the Hill cipher is "much harder" than performing a known-plaintext one. Indeed it seems that only few such attacks have been developed, all of them supposing an a priori knowledge on the language and making a search over the key-space; we refer the reader to the papers (Bauer and Millward, 2007; Yum and Lee, 2009; Leap et al., 2016; McDevitt et al., 2018).

Further it is known that, in the case of no restrictions on the considered language or alphabet, "the best publicly known ciphertext-only attack on Hill cipher requires full search over all possible secret keys", as stated in (Khazaei and Ahmadi, 2017). Note that this paper indeed proposes a new attack but only in the case of meaningful English texts with an alphabet of size 26. In the case where p is a prime number, the Brute-Force Attack tests almost p^{n^2} matrices (Overbey et al., 2005, Lemma 4.3).

In view of this, we propose in the present paper to study another kind of ciphertext-only attack which is not based on a search over the key-space but rather over restricted regions of the text-space, regions called orbits. This attack, denoted Orbit-Based Attack (OBA), lies on a partition into orbits of the text-space induced by the Hill cipher. In this paper, we prove the existence of this partition exploiting group action theory and we make explicit the orbits by exploiting the property that Hill preserves the linear combinations of blocks in a given text. The ciphertext and the associated plaintext being necessarily in the same orbit, our theoretical results assure that the size of their orbit, and hence the maximal number of texts to test, is smaller than the number of keys. To make clear the ideas of our approach and avoiding too technical computations, we assume that the size of the alphabet is a prime number; similar results are expected to hold true in more general settings.

An algorithm for the OBA is then proposed. Our aim here is to show that this attack can be faster in terms of time-execution on average over random texts than the above mentioned Brute-Force Attack (BFA) in the case where the only assumption is that the size of the alphabet is a prime number. Even though the computational complexities are proved to be roughly the same, we illustrate by means of numerous experiments that the OBA permits to speed-up on average the runtime of the decryption process as compared to the BFA.

Proceedings of the 2nd International Conference on Historical Cryptology, pages 13-22
Mons, Belgium, 23-26 June, 2019

2 Preliminaries

In this section, we define rigorously the Hill cipher for the sake of completeness and we introduce some notations which will be used throughout the rest of this paper.

2.1 The Hill cipher

We start by the definition of the Hill cipher we use in this paper; we refer to (Hill, 1929) for the original one.

Definition 1 (Hill cipher). *A plaintext string X of size $M = mn$ over an alphabet having p characters is defined as a vector of size M over $\mathbb{Z}_p$ using an arbitrary bijection between the elements of the alphabet and the elements of $\mathbb{Z}_p$. The plaintext X is splitted into m blocs of size n such that $X = X_1X_2\dots X_m$. An invertible $n\times n$ matrix K over $\mathbb{Z}_p$, called the key-matrix, is then chosen. Afterwards we construct a block diagonal matrix A of size $M\times M$ over $\mathbb{Z}_p$ whose main diagonal sub-matrices are equal to K. The encryption is finally performed by considering each X_i as a vector of $(\mathbb{Z}_p)^n$ and by computing the ciphertext $Y = Y_1Y_2\dots Y_m$ as follows:*

$$Y = AX \ (mod\ p)\,,$$

which is equivalent to $Y_i = KX_i$ $(mod\ p)$, for all $i \in \{1,\dots,m\}$. Thanks to the invertible nature of K, A is invertible as well and the decryption is performed by computing:

$$X = A^{-1}Y \ (mod\ p)\,.$$

Throughout the rest of this paper, we choose p as a prime number for the sake of simplicity. This implies in particular that the set $\mathbb{Z}_p$ is the field of p elements and so the division is well-defined, making the arguments and computations easier. However, one may plan to generalise the present results to the case where no assumption on p is made, covering hence more realistic cases.

We define now the set of invertible block diagonal matrices.

Definition 2. *Let $GL_n(\mathbb{Z}_p)$ be the set of invertible matrices of size $n\times n$ over $\mathbb{Z}_p$. A matrix A of size $M\times M$ over $\mathbb{Z}_p$ belongs to $\mathcal{G}_{M,n}$ if and only if there exists $K \in GL_n(\mathbb{Z}_p)$ such that*

$$A = \begin{pmatrix} K & & & \\ & K & & \\ & & \ddots & \\ & & & K \end{pmatrix}.$$

We note that $GL_n(\mathbb{Z}_p)$ and $\mathcal{G}_{M,n}$ are clearly in bijection.

We are now in position to define the Hill cipher map, which will be proved to be a group action in the following section.

Definition 3 (Hill cipher map). *Let $\mathcal{H} : \mathcal{G}_{M,n} \times (\mathbb{Z}_p)^M \longrightarrow (\mathbb{Z}_p)^M$ be the map defined by*

$$\forall (A,X) \in \mathcal{G}_{M,n}\times(\mathbb{Z}_p)^M \qquad \mathcal{H}(A,X) := AX\,.$$

2.2 Group action theory

Group action theory offers a convenient setting to describe the attack proposed in this paper. While the results can be actually proved without invoking this theory, the latter may be helpful to make clear the effects of Hill on the texts. Consequently, we recall some abstract results from group action theory for the sake of completeness; their proofs can be found for instance in (Smith, 2008, Chapter 10).

In the rest of the present section, the notation G will refer to a group whose group law and identity element are respectively represented by $\cdot$ and e. We start by recalling the notion of a (left) group action on a set.

Definition 4 (Group action). *Let G and S be respectively a group and a set. A map $\varphi : G\times S \longrightarrow S$ is said to be a group action of G on S if and only if it satisfies the two following properties:*

- Identity: *for all $s \in S$, we have*

$$\varphi(e,s) = s\,;$$

- Compatibility: *for all $g,h \in G$ and $s \in S$, we have*

$$\varphi(g\cdot h, s) = \varphi(g,\varphi(h,s))\,.$$

We define now the orbit and the stabiliser of an element $s \in S$: the orbit of s is the set of elements of S to which s can be sent by the elements of G, while the stabiliser of s is the set of elements of the group G which do not move s. Let us emphasise that an element $s \in S$ can not be sent outside its orbit by definition.

Definition 5 (Orbit and stabiliser). *Let $\varphi : G\times S \longrightarrow S$ be a group action of a group G on a set S and let $s \in S$.*

1. *The orbit $Orb_\varphi(s)$ of s is defined as follows:*

$$Orb_\varphi(s) = \{y \in S \mid \exists g \in G \quad y = \varphi(g,s)\}\,.$$

2. *The stabiliser $Stab_\varphi(s)$ of s is defined as follows:*

$$Stab_\varphi(s) = \{g \in G \mid \varphi(g,s) = s\}\,.$$

These two notions are closely related: it is shown that the orbit of an element $s \in S$ is isomorphic to the quotient of the group G by the stabiliser of s. Roughly speaking, this means that it is sufficient to move s by all the elements of the group G which do not fix s to recover the whole orbit of s. In the finite group case, this permits to compute the cardinal of the orbit of a given element $s \in S$:

Corollary 1. *Let $\varphi : G\times S \longrightarrow S$ be a group action of a finite group G on a set S and let $s \in S$. Then we have*

$$\left|Orb_\varphi(s)\right| = \frac{|G|}{\left|Stab_\varphi(s)\right|}\,,$$

where $|Z|$ denotes the cardinal of a given set Z.

To conclude this subsection, we mention that an action of a group G on a set S defines an equivalence relation on S whose equivalence classes are given by the orbits. Since two equivalence classes are either equal or disjoint, the set of the orbits under the action of G forms a partition of S; this is recalled in the following result:

Theorem 1. *Let $\varphi : G \times S \longrightarrow S$ be a group action of a group G on a set S.*

1. *Let $s,t \in S$. Then we have either*

$$Orb_\varphi(s) = Orb_\varphi(t)$$

or

$$Orb_\varphi(s) \cap Orb_\varphi(t) = \emptyset\,.$$

2. *Let $\mathcal{R} \subseteq S$ be a set of orbit representatives, in other words a subset of S which contains exactly one element from each orbit. Then the family $\{Orb_\varphi(s)\}_{s\in\mathcal{R}}$ forms a partition of S.*

An illustration of Theorem 1 is given in Figure 1.

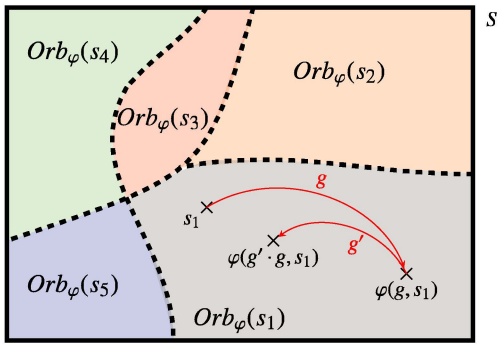

Figure 1: Illustration of a partition of S created by the group action φ

3 Group Action Theory for Hill Cipher

We start this section by proving the following property of Hill inherited from its linear nature: it preserves the linear combinations of any given text. This is stated in the following proposition:

Proposition 1. *Let $X = X_1 \dots X_m \in (\mathbb{Z}_p)^M$ be a plaintext. Suppose that the block X_i is a linear combination of q other blocks $X_{i_1}, \dots, X_{i_q}$,* i.e.,

$$\exists \lambda_1^{(i)}, \dots, \lambda_q^{(i)} \in \mathbb{Z}_p \qquad X_i = \sum_{k=1}^{q} \lambda_k^{(i)} X_{i_k}\,. \tag{1}$$

Then the i-th block of the ciphertext $Y = \mathcal{H}(A,X)$, with A an element of $\mathcal{G}_{M,n}$ associated with a key-matrix $K \in GL_n(\mathbb{Z}_p)$, satisfies

$$Y_i = \sum_{k=1}^{q} \lambda_k^{(i)} Y_{i_k}\,.$$

Proof. Since we have $Y_i = KX_i \; \forall i \in \{1,\dots,m\}$, it is sufficient to multiply equality (1) by K to obtain the result. □

Our goal in the present section is to study some consequences of this property. Especially, we shall prove the existence of a partition of the text-space due to the Hill cipher map. To do so, we shall exploit group action theory, whose principles will be illustrated in the setting of Hill, to structure our arguments. The results obtained here are at the root of the Orbit-Based Attack presented in the next section. Further, we mention that our work seems to formalise the principle of the approach developed in (McDevitt et al., 2018).

It is important to note that, the Hill cipher being a symmetric-key cipher, the results presented here can be interpreted in two ways: the relation $Y = \mathcal{H}(A,X)$ can describe either the cipher of the plaintext X leading to the ciphertext Y, or the decipher of a ciphertext X leading to the plaintext Y. Therefore an input text X can be interpreted as a plaintext (resp. ciphertext) and the output text Y as a ciphertext (resp. plaintext) if we consider a cipher (resp. decipher).

We start our study by showing that the Hill cipher map given in Definition 3 is actually a group action.

Theorem 2. *The Hill cipher map given in Definition 3 is a group action.*

Proof. It is easy to show that the set $\mathcal{G}_{M,n}$ is actually a subgroup of $GL_M(\mathbb{Z}_p)$, so it is itself a group. Further the *Identity* and *Compatibility* points of Definition 5 are satisfied thanks to the basic properties of matrix multiplication (multiplication by the identity matrix and associativity). □

From this theorem, it follows that the text-space is split into orbits which are stable under the Hill cipher map; in other words, if we choose an input text and we apply the Hill cipher map to it, then the resulting output text is necessarily inside the orbit of the input text, as illustrated in Figure 1 in an abstract setting.

Corollary 2. *1. Let $X, X' \in (\mathbb{Z}_p)^M$. Then we have either $Orb_\mathcal{H}(X) = Orb_\mathcal{H}(X')$ or $Orb_\mathcal{H}(X) \cap Orb_\mathcal{H}(X') = \emptyset$.*

2. *Let $\mathcal{X} \subseteq (\mathbb{Z}_p)^M$ be a set of orbit representatives, in other words a subset of texts which contains exactly one text from each orbit. Then the family $\{Orb_\mathcal{H}(X)\}_{X\in\mathcal{X}}$ forms a partition of $(\mathbb{Z}_p)^M$.*

3. *For all $X \in (\mathbb{Z}_p)^M$, we have*

$$\left|Orb_\mathcal{H}(X)\right| = \frac{\left|GL_n(\mathbb{Z}_p)\right|}{\left|Stab_\mathcal{H}(X)\right|}\,.$$

Proof. Simple application of Theorem 1 and Corollary 1. □

According to Corollary 2, the number of elements of an orbit given by an input text X depends on the cardinal of the stabiliser of X. In the following proposition, we describe explicitly the stabiliser of any input text X by exploiting the property that the Hill cipher preserves linear combinations (see Proposition 1); in particular, this will permit to derive the cardinal of the orbit of X in Corollary 3. Further let us mention that we do not treat the case of the input text given by $0_{(\mathbb{Z}_p)^M}$ since any matrix belonging to $\mathcal{G}_{M,n}$ is in its stabiliser.

Proposition 2. *Let $X = X_1 \dots X_m \in (\mathbb{Z}_p)^M \setminus \{0_{(\mathbb{Z}_p)^M}\}$. Suppose that there exist $1 \leqslant q \leqslant n$ and $i_1, \dots, i_q \in \{1, \dots, m\}$ such that $X_{i_1}, \dots, X_{i_q}$ are linearly independent and, for each $i \notin \{i_1, \dots, i_q\}$, X_i is a linear combination of $X_{i_1}, \dots, X_{i_q}$. Then $A \in Stab_{\mathcal{H}}(X)$ if and only if*

$$A = \begin{pmatrix} P\widetilde{K}P^{-1} & & & \\ & P\widetilde{K}P^{-1} & & \\ & & \ddots & \\ & & & P\widetilde{K}P^{-1} \end{pmatrix},$$

with

- $P = (X_{i_1} | \dots | X_{i_q} | V_{q+1} | \dots | V_n) \in GL_n(\mathbb{Z}_p)$ *where $V_{q+1}, \dots, V_n$ are vectors of $(\mathbb{Z}_p)^n$ such that $\{X_{i_1}, \dots, X_{i_q}, V_{q+1}, \dots, V_n\}$ is a basis of $(\mathbb{Z}_p)^n$;*
- $\widetilde{K} \in GL_n(\mathbb{Z}_p)$ *is of the form*

$$\begin{pmatrix} 1 & 0 & \dots & 0 & \tilde{k}_{1,q+1} & \dots & \tilde{k}_{1,n} \\ 0 & 1 & \ddots & 0 & \tilde{k}_{2,q+1} & \dots & \tilde{k}_{2,n} \\ \vdots & \ddots & \ddots & \vdots & \vdots & & \vdots \\ 0 & \dots & 0 & 1 & \tilde{k}_{q,q+1} & \dots & \tilde{k}_{q,n} \\ 0 & \dots & \dots & 0 & \tilde{k}_{q+1,q+1} & \dots & \tilde{k}_{q+1,n} \\ \vdots & & & \vdots & \vdots & & \vdots \\ 0 & \dots & \dots & 0 & \tilde{k}_{n,q+1} & \dots & \tilde{k}_{n,n} \end{pmatrix}. \quad (2)$$

If $q = n$ then $Stab_{\mathcal{H}}(X) = \{I_M\}$, where I_M is the identity matrix of size M.

Proof. Choose $X = X_1 \dots X_m \in (\mathbb{Z}_p)^M \setminus \{0_{(\mathbb{Z}_p)^M}\}$ and assume that $X_{i_1}, \dots, X_{i_q}$ are linearly independent, where $1 \leqslant q \leqslant n$ and $i_1, \dots, i_q \in \{1, \dots, m\}$, and that

$$\exists \lambda_1^{(i)}, \dots, \lambda_q^{(i)} \in \mathbb{Z}_p \qquad X_i = \sum_{k=1}^{q} \lambda_k^{(i)} X_{i_k},$$

for all $i \notin \{i_1, \dots, i_q\}$. If $q \neq n$, choose $V_{q+1}, \dots, V_n \in (\mathbb{Z}_p)^n$ such that the family $\{X_{i_1}, \dots, X_{i_q}, V_{q+1}, \dots, V_n\}$ is a basis of $(\mathbb{Z}_p)^n$; let us mention that such vectors exist according to the incomplete basis theorem (Artin, 2011, Proposition 3.15). Hence the matrix P defined in the statement of Proposition 2 is invertible and satisfies for all $k \in \{1, \dots, q\}$,

$$PE_k = X_{i_k} \iff P^{-1}X_{i_k} = E_k, \quad (3)$$

where E_k is the k-th vector of the canonical basis of $(\mathbb{Z}_p)^n$. Furthermore, for a given matrix $\widetilde{K}$ of the form (2), the following relation is true for each $k \in \{1, \dots, q\}$,

$$P\widetilde{K}P^{-1}X_{i_k} = P\widetilde{K}E_k = PE_k = X_{i_k}. \quad (4)$$

Then we deduce that, for all $i \notin \{i_1, \dots, i_q\}$,

$$\begin{aligned} P\widetilde{K}P^{-1}X_i &= P\widetilde{K}P^{-1}\left(\sum_{k=1}^{q} \lambda_k^{(i)} X_{i_k}\right) \\ &= \sum_{k=1}^{q} \lambda_k^{(i)} P\widetilde{K}P^{-1}X_{i_k} = \sum_{k=1}^{q} \lambda_k^{(i)} X_{i_k} \\ &= X_i. \end{aligned} \quad (5)$$

We are now in position to prove the equivalence stated in Proposition 2.2.

⇐ If a matrix $A \in \mathcal{G}_{M,n}$ is given by

$$A = \begin{pmatrix} P\widetilde{K}P^{-1} & & & \\ & P\widetilde{K}P^{-1} & & \\ & & \ddots & \\ & & & P\widetilde{K}P^{-1} \end{pmatrix},$$

where $\widetilde{K}$ is an invertible matrix of the form (2), then A satisfies

$$AX = \begin{pmatrix} P\widetilde{K}P^{-1}X_1 \\ P\widetilde{K}P^{-1}X_2 \\ \vdots \\ P\widetilde{K}P^{-1}X_m \end{pmatrix} = \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_m \end{pmatrix} = X,$$

according to the relations (4) and (5). This proves that $A \in Stab_{\mathcal{H}}(X)$.

⇒ Let $A \in \mathcal{G}_{M,n}$ be an element of the stabiliser of X. It follows

$$\forall k \in \{1, \dots, q\} \qquad KX_{i_k} = X_{i_k},$$

where $K \in GL_n(\mathbb{Z}_p)$ is the key-matrix. By using relation (3), we obtain

$$\forall k \in \{1, \dots, q\} \qquad P^{-1}KPE_k = E_k.$$

We observe then that the matrix $\widetilde{K} := P^{-1}KP$ is of the form (2) and is invertible since it is similar to $K \in GL_n(\mathbb{Z}_p)$. This finally proves that

$$A = \begin{pmatrix} P\widetilde{K}P^{-1} & & & \\ & P\widetilde{K}P^{-1} & & \\ & & \ddots & \\ & & & P\widetilde{K}P^{-1} \end{pmatrix}.$$

□

As a consequence of the preceding result, we are able to give the cardinal of the orbit of any input text, *i.e.* the number of texts which can be attained from this input. We note that this cardinal depends only on the number q of linearly independent blocks within the input text.

Corollary 3. *Let $X = X_1 \dots X_m \in (\mathbb{Z}_p)^M \setminus \{0_{(\mathbb{Z}_p)^M}\}$. Suppose that there exist $1 \leqslant q \leqslant n$ and $i_1, \dots, i_q \in \{1, \dots, m\}$ such that $X_{i_1}, \dots, X_{i_q}$ are linearly independent and, for each $i \notin \{i_1, \dots, i_q\}$, X_i is a linear combination of $X_{i_1}, \dots, X_{i_q}$. Then we have*

$$\left|Orb_{\mathcal{H}}(X)\right| = \prod_{k=0}^{q-1} (p^n - p^k) .$$

Proof. First of all, we recall from (Rotman, 1965, Theorem 8.13) that the cardinal of $GL_n(\mathbb{Z}_p)$ is given by

$$\left|GL_n(\mathbb{Z}_p)\right| = \prod_{k=0}^{n-1} (p^n - p^k) .$$

Let us now compute the cardinal of $Stab_{\mathcal{H}}(X)$. According to Proposition 2, this is actually equal to the number of invertible matrices of the form (2), namely

$$\left(\begin{array}{c|c} I_q & \widetilde{K}_{1,2} \\ \hline \mathbf{0} & \widetilde{K}_{2,2} \end{array}\right) .$$

Such a matrix being invertible, the sub-matrix $\widetilde{K}_{2,2}$ is invertible as well; hence we have

$$\prod_{k=0}^{n-q-1} (p^{n-q} - p^k)$$

choices for the sub-matrix $\widetilde{K}_{2,2}$. Once this sub-matrix is fixed, it remains to choose $\widetilde{K}_{1,2}$, which does not have any restriction: thus there are $p^{q(n-q)}$ choices for the sub-matrix $\widetilde{K}_{1,2}$. Consequently, we obtain

$$\begin{aligned} \left|Stab_{\mathcal{H}}(X)\right| &= p^{q(n-q)} \prod_{k=0}^{n-q-1} (p^{n-q} - p^k) \\ &= \prod_{k=q}^{n-1} (p^n - p^k) . \end{aligned}$$

Finally, by using Corollary 2.3, it follows

$$\left|Orb_{\mathcal{H}}(X)\right| = \frac{\prod_{k=0}^{n-1} (p^n - p^k)}{\prod_{k=q}^{n-1} (p^n - p^k)} = \prod_{k=0}^{q-1} (p^n - p^k) .$$

□

The previous corollary shows that the number of elements of an orbit given by an input text is always smaller or equal to the number of elements in the key-space $GL_n(\mathbb{Z}_p)$. Theoretically this means that if we consider an oracle able to answer in $O(1)$ whether a matrix is the key or whether a text is the corresponding plaintext of the ciphertext of interest, then performing an exhaustive search on the key-space would be in $O(\prod_{k=0}^{n-1}(p^n - p^k))$ and on the orbit of the ciphertext in $O(\prod_{k=0}^{q-1}(p^n - p^k))$ with $q \leqslant n$ (q being the number of linearly independent blocks in the ciphertext). Since in most of the cases $q = n$, this assures that these two kinds of search share the same worst-case complexity. The key idea of the OBA lying on a search on the orbit of the ciphertext, the preceding remark assures that it will be theoretically at worst as efficient as the BFA.

It remains to make practicable such a ciphertext-only attack. To do so, we have to describe explicitly the orbit of any given text. This is provided in the following theorem whose proof exploits once again the property that the Hill cipher preserves linear combinations; see Proposition 1.

As previously, we do not treat the case of the input text given by $0_{(\mathbb{Z}_p)^M}$ since its orbit is equal to the singleton $\{0_{(\mathbb{Z}_p)^M}\}$.

Theorem 3. *Let $X = X_1 \dots X_m \in (\mathbb{Z}_p)^M \setminus \{0_{(\mathbb{Z}_p)^M}\}$. Suppose that there exist $1 \leqslant q \leqslant n$ and $i_1, \dots, i_q \in \{1, \dots, m\}$ such that $X_{i_1}, \dots, X_{i_q}$ are linearly independent and*

$$\exists \lambda_1^{(i)}, \dots, \lambda_q^{(i)} \in \mathbb{Z}_p \qquad X_i = \sum_{k=1}^{q} \lambda_k^{(i)} X_{i_k} ,$$

for all $i \notin \{i_1, \dots, i_q\}$. Then $Y = Y_1 \dots Y_m \in (\mathbb{Z}_p)^M$ belongs to $Orb_{\mathcal{H}}(X)$ if and only if $Y_{i_1}, \dots, Y_{i_q}$ are linearly independent and

$$\forall\, i \notin \{i_1, \dots, i_q\} \qquad Y_i = \sum_{k=1}^{q} \lambda_k^{(i)} Y_{i_k} .$$

Proof. Choose $X = X_1 \dots X_m \in (\mathbb{Z}_p)^M \setminus \{0_{(\mathbb{Z}_p)^M}\}$ and suppose that there exist $1 \leqslant q \leqslant n$ and $i_1, \dots, i_q \in \{1, \dots, m\}$ such that $X_{i_1}, \dots, X_{i_q}$ are linearly independent and

$$\exists \lambda_1^{(i)}, \dots, \lambda_q^{(i)} \in \mathbb{Z}_p \qquad X_i = \sum_{k=1}^{q} \lambda_k^{(i)} X_{i_k} ,$$

for all $i \notin \{i_1, \dots, i_q\}$. Define now the set $E_q(X)$ as follows: $Y = Y_1 \dots Y_m \in (\mathbb{Z}_p)^M$ belongs to $E_q(X)$ if and only if it satisfies

$$\left\{ \begin{array}{l} Y_{i_1}, \dots, Y_{i_q} \text{ are linearly independent} \\ \forall\, i \notin \{i_1, \dots, i_q\} \qquad Y_i = \displaystyle\sum_{k=1}^{q} \lambda_k^{(i)} Y_{i_k} \end{array} \right. .$$

Hence we have to show $Orb_{\mathcal{H}}(X) = E_q(X)$. To do so, we prove an inclusion and the equality between the two cardinals.

$\boxed{\subseteq}$ Let $Y = Y_1 \dots Y_m \in (\mathbb{Z}_p)^M$ be an element of $Orb_{\mathcal{H}}(X)$. Then, by definition, there exists $A \in \mathcal{G}_{M,n}$ such that $Y = AX$, *i.e.*,

$$\forall\, i \in \{1, \dots, m\} \quad Y_i = KX_i ,$$

where K is the key-matrix. As an immediate consequence of the linear independence of

$X_{i_1},\dots,X_{i_q}$, the vectors $Y_{i_1},\dots,Y_{i_q}$ are linearly independent, and by Proposition 1, we have

$$\forall\, i \notin \{i_1,\dots,i_q\} \qquad Y_i = \sum_{k=1}^{q} \lambda_k^{(i)}\, Y_{i_k}\ .$$

This shows that $Orb_{\mathcal{H}}(X)$ is included in $E_q(X)$.

$\boxed{=}$ The cardinal of the set $E_q(X)$ is given by the number of linearly independent families of q vectors belonging to $(\mathbb{Z}_p)^n$, that is to say

$$\prod_{k=0}^{q-1} (p^n - p^k)\ .$$

We employ then Corollary 3 which shows that $Orb_{\mathcal{H}}(X)$ and $E_q(X)$ have the same cardinal.

This finally proves $Orb_{\mathcal{H}}(X) = E_q(X)$. □

The benefit of this result lies on the fact that it enumerates all the possible output texts from any given input text via the Hill cipher map: this permits to develop an algorithm for the OBA decrypting Hill cipher without studying the key-space. Moreover, even if it is unlikely in practice that the number q of linear independent blocks is strictly smaller than n, we expect an efficiency gain for the OBA in this case.

4 Algorithm and computational complexity of the Orbit-Based Attack

The preceding theoretical results provide all the ingredients to create an algorithm for the Orbit-Based Attack, which consists in making a search in the orbit of the ciphertext of interest, and to determine its computational complexity. We emphasise that the algorithm proposed here is not intended to be optimised and some steps could be refined. The reader can find information on computational complexity in (Papadimitriou, 1994).

Throughout the rest of this section, we consider a ciphertext C of m blocks, each block having a size of n characters (with $n \leq m$) in an alphabet of size p. Moreover we assume that the ciphertext has q linearly independent blocks (with $q \leq n$), thus $m-q$ dependent ones. The only condition we put here is that p is a prime number.

The main idea of the algorithm is to build sequentially the elements of the orbit of C by exploiting Theorem 3 until the plaintext associated with C is found. From a practical point of view, this can be achieved by placing firstly C in a $n \times m$ matrix over $\mathbb{Z}_p$, called $\mathtt{C}_q$ and such that its k-th column corresponds to the k-th block of C, and by applying then the three following steps:

1. Performing a Gaussian elimination (Golub and Van Loan, 2012) column by column on $\mathtt{C}_q$ to make explicit the linear combinations of the blocks within C; the same computations are made on an identity matrix of size $m \times m$ to store the indices and coefficients of the linear combinations.

2. Extracting the indices of the q independent blocks and the coefficients giving the $m-q$ linear combinations.

3. Building the elements of the orbit of C in a random manner until the right plaintext is found. To do so, each element is built by first choosing randomly q independent blocks and the $m-q$ remaining blocks are then deduced by using the linear combinations determined in the two preceding steps.

Algorithm 1: LU-type decomposition of $\mathtt{C}_q$

Data: $\mathtt{C}_q$ the matrix storing the ciphertext, `Id` the identity matrix of size $m \times m$
Result: $\mathtt{C}_q$ is lower triangular, `LC` contains the indices of the linearly independent blocks and the coefficient of the linear combinations

```
for (k=1, k ≤ n, k++) do
    j ← firstNotNull(C_q[k,i],1 ≤ i ≤ n);
    if (C_q[k,j] ≠ 0) then
        divide column j by C_q[k,j] in C_q and in Id;
        swap columns j and k in C_q and Id;
        for (i=k+1, i ≤ m, i++) do
            substract to column i the column k
              multiplied by C_q[k,i] in C_q and in Id;
        end
    end
end
LC = Id;
```

Lower-Upper decomposition of $\mathtt{C}_q$

Lower–upper (LU) decomposition or factorisation factors a matrix as the product of a lower triangular matrix and an upper triangular matrix. It can be obtained by the Gaussian elimination and the factors contain the information on the linear dependencies of the columns. Let us specify step 1. with Algorithm 1 which performs an LU decomposition on the matrix $\mathtt{C}_q$.

Let us determine the complexity of this initialisation step. Roughly speaking, at the k-th step, we search for a non-zero element on the k-th row (this step is omitted in the complexity computation), then we multiply the chosen column of size $n-k+1$ by the inverse of its non-zero element on the k-th row (such a computation is supposed to be cost-less) and we swap two columns if necessary, and we multiply the $m-k$ remaining columns by scalars and make $m-k$ additions. Furthermore, the same operations are made on the matrix `Id`: at the k-step, we work with columns of size k. We repeat these operations q times since the rank of $\mathtt{C}_q$ is equal to q. Adding the computational complexity of

each operation up gives:

$$\begin{aligned}
&C_1(m,n,q)\\
&= \sum_{k=1}^{q}\Big((m-k+1)(n-k+1)+(m-k)(n-k+1)\Big)\\
&\quad + \sum_{k=1}^{q}\Big((m-k+1)k+(m-k)k\Big)\\
&= 2(n+1)\sum_{k=1}^{q}(m-k)+(n+1)\sum_{k=1}^{q}1\\
&= 2(n+1)\left(mq-\frac{q(q+1)}{2}\right)+(n+1)q\\
&= 2nmq+2mq-nq^2-q^2\,.
\end{aligned}$$

Obtaining the linear combinations within C_q

In the second step, we search in the output matrix LC the indices of the linearly independent blocks of the ciphertext C and the coefficients of the linear combinations. In view of the construction of the matrix LC in Algorithm 1, it is an upper triangular matrix up to a permutation if columns have been swapped. We note that q rows have been filled, corresponding to the q steps to make lower triangular C_q; moreover, if the k-th column of the lower triangular matrix C_q is zero, then the k-th column of LC gives the coefficients of one of the $m-q$ linear combinations within C. Hence determining the indices of the independent blocks of C consists in making searches in Id, such operations are supposed to be negligible in terms of computational complexity. Thus this second step is cost-less.

The algorithm that can perform this search is depicted in Algorithm 2.

Recovering gradually the orbit of C

The two preceding steps combined to Theorem 3 permit to build all the elements of the orbit of the ciphertext C. Here we do not aim at recovering the whole orbit but rather to build element by element and to test whether the right plaintext is obtained. To do so, for each text to build, we choose randomly q linearly independent blocks of size n and we put them in the text in such a way that their indices are given by the matrix LC from the second step; this generation is supposed to be negligible. Then we compute the $m-q$ associated linearly dependant blocks of size n from the q independent blocks by using once again the information contained in LC. This computation is depicted in Algorithm 3.

We observe that, for each linearly dependent block, we multiply the q independent blocks of size n by coefficients contained in LC and we add these q resulting blocks up to find the linearly dependent block. Supposing from now now that τ_1 tries are necessary to find the q right independent blocks of the plaintext P, we obtain then the computational complexity for the third step:

$$\begin{aligned}
C_3(m,n,q,\tau_1) &= \tau_1(m-q)(qn+(q-1)n)\\
&= 2\tau_1 mnq-\tau_1 mn-2\tau_1 q^2 n+\tau_1 nq\,.
\end{aligned}$$

Algorithm 2: Get the linear combinations

Data: LC the matrix storing the different linear combinations, C_q the lower triangulated
Result: the indices of the free columns and the coefficient of the linear combinations.

```
1  size = 0;
2  ind = ∅;
3  for (i=1, i ≤ C_q.nbLines(), i++) do
4  |  if (C_q[i][i] ≠ 0) then size = size +1 ;
5  end
6  for (i=1, i ≤ LC.nbLines(), i++) do
7  |  if (ind.size() == size) then return ind;
8  |  for (j=1, j ≤ size, j++) do
9  |  |  if (LC[i][j] ≠ 0) then ind.add(i) ;
10 |  end
11 end
12 for (col=ind.size(), col ≤ LC.nbCols(), col++) do
13 |  for (line=1, line ≤ LC.nbLines(), line++) do
14 |  |  lineOK = true;
15 |  |  for (k=1, k ≤ ind.size(), k++) do
16 |  |  |  if (ind[k] == line) then lineOK =
   |  |  |    false ;
17 |  |  end
18 |  |  if (lineOk ∧ LC[line][col] ≠ 0) then
19 |  |  |  result[0][col-ind.size()] = line;
20 |  |  end
21 |  end
22 end
23 for (i=1, i ≤ ind.size(), i++) do
24 |  for (col=ind.size(), col ≤ LC.nbCols(), col++)
   |   do
25 |  |  result[i+1][col-ind.size()] =
   |  |   -LC[ind[i]][col];
26 |  end
27 end
28 return result;
```

Note that τ_1 is bounded by the number of families having q linearly independent blocks of size n, namely,

$$\prod_{k=0}^{q-1}(p^n-p^k)\le p^{nq}\,, \tag{6}$$

which can be very large. Nevertheless, we emphasise that if some prior knowledge on the text are available, such as the language, then the mean number τ_1 of tries can drastically diminish, reducing the computational complexity of this step.

Finally, the final cost of the Orbit-Based Attack (OBA) is:

$$\begin{aligned}
C^{OBA}(m,n,q,\tau_1) &= C_1(m,n,q)+C_3(m,n,q,\tau_1)\\
&= 2(1+\tau_1)mnq+2mq\\
&\quad +\tau_1 nq-(1+2\tau_1)nq^2\\
&\quad -\tau_1 mn-q^2\,.
\end{aligned}$$

Algorithm 3: Re-build the linearly dependent blocks

Data: `res`: a randomly initialised matrix, `LC` the matrix storing the different linear combinations
Result: a potential plaintext where each linearly dependant block has been computed

```
1  for (c=1, c ≤ LC.nbCols(), c++) do
2      col = LC[0][c];
3      for (line=1, line ≤ LC.nbLines(), line++) do
4          res[line][col] = 0;
5          for (k=1, k ≤ ind.size(), k++) do
6              res[line][col] = res[line][col] +
                 (res[line][ind[k]] × LC[k+1][c])
7          end
8      end
9  end
10 return decrypt(res);
```

Soundness, completeness and termination of the OBA

The Orbit-Based Attack to decrypt the Hill cipher is depicted in Algorithm 4. Note that the line 10 works as an odometer, meaning that if the loop does not stop, all the matrices will be generated.

Algorithm 4: Orbit-Based Attack

Data: *cipher*: the ciphertext that we want to decipher
Result: the corresponding plaintext

```
1  ⟨in,LC⟩ = Algorithm_1(cipher,Id);
2  ind = Algorithm_2(LC,in);
3  for (c=1, c ≤ LC.nbCols(), c++) do
4      for (l=1, l ≤ LC.nbLines(), l++) do
5          guess[l][c] = random() % (p);
6      end
7  end
8  guessMessage = Algorithm_3(guess,LC);
9  while (¬findWord(guessMessage)) do
10     guess = guess + 1;
11     guessMessage = Algorithm_3(guess,LC);
12 end
13 return guessMessage;
```

We prove now that Algorithm 4 satisfies the soundness, the completeness and the termination, which are the three main characteristics of an algorithm.
First let us prove the soundness, which means that if the algorithm gives an answer, then it is the expected one.

Proposition 3. *Algorithm 4 is sound.*

Proof. The soundness is a direct consequence of the use of the oracle "findWord". Indeed, since the only way to return a solution is to exit the while-loop which is possible only if "findWord" returns true. The solution is thus necessarily the corresponding plaintext. □

Now let us prove the completeness, meaning that the algorithm gives an answer for any input.

Proposition 4. *Algorithm 4 is complete.*

Proof. By having a look at the line 10 of Algorithm 4, we can see that each time we did not find the corresponding plaintext, we increment one value in the guess matrix, in the same way as in an odometer. By searching in such way all the matrices present in the orbit text for any ciphertext, we will perform an exhaustive search over its orbit, guaranteeing in such way the completeness of the algorithm. □

Finally let us prove the termination, stating that the time needed by the algorithm to terminate is finite.

Proposition 5. *Algorithm 4 terminates.*

Proof. The proof of termination is straightforward. There is no choose, nor backtrack in the algorithm. Moreover each loop iterates over finite domains: the columns or rows in matrices for Algorithms 1, 2 and 3, and the texts belonging to the orbit of the ciphertext for Algorithm 4. Therefore Algorithm 4, based on Algorithms 1, 2 and 3, terminates. □

5 Experiments

In this section, we present and comment on results from numerous experiments. Our aim here is to compare the new Orbit-Based Attack with the classical Brute-Force Attack, which is the only efficient ciphertext-only attack for the Hill cipher in case where no information on the text is available (Khazaei and Ahmadi, 2017).

Computational complexity of the Brute-Force Attack

Before commenting on the experiments, let us talk about the Brute-Force Attack (BFA). We recall that it consists in testing all the invertible matrices of size $n \times n$ over $\mathbb{Z}_p$ until the right key is found.

Here we propose an algorithm for the Brute-Force Attack (BFA). For the sake of completeness, we inform the reader that our implementation of the BFA is quite naive. However, no matter the implementation, the empirical results should remain the same. Indeed, both approaches being mainly based on the same basic matrix operations, an optimisation for the BFA would also be an optimisation for the OBA as a side-effect. The Brute-Force Attack (BFA) is working as follows: we pick randomly a matrix of size $n \times n$, this generation being supposed negligible as previously. Such a random matrix has a high probability to be invertible, as stated in (Overbey et al., 2005). Moreover we suppose that we need τ_2 tries on average to find the correct matrix. It remains to multiply each m block of size n by

the chosen matrix of size $n \times n$. For each m block, it corresponds to make n multiplications and $n-1$ additions of columns of size n. We thus have:

$$C^{BFA}(m,n,q,\tau_2) = \tau_2 mn(n+n-1)$$
$$= 2\tau_2 mn^2 - \tau_2 mn\,.$$

Let us now remark that the quantity τ_2 is bounded by the number given in (6) with $q = n$ and we have $\tau_1 = \tau_2$ without any prior knowledge on the corresponding plaintext or on the key-matrix. In this setting, one can conclude that both the Orbit-Based Attack and the Brute-Force Attack are in $O(p^{n^2})$. Nevertheless these results are mainly theoretical and, in practice, one or the other attack may be faster in terms of time-execution, motivating the following subsections.

Empirical results for the Orbit-Based and Brute-Force Attacks

In this section, we present empirical results generated on a cluster of Xeon, 4 cores, 3.3 GHz with CentOS 7.0 with a memory limit of 32GB and a runtime limit of 9,000 seconds per text per attack. Each text is uniformly randomly generated to avoid any statistical bias. The size of the text is given by $n \times m$, where m is the number of blocks and n the number of characters in each block, and we select randomly each character with a probability of $\frac{1}{p}$, where p is the size of the alphabet.

Here is our experimental protocol: $p \in \{5,7,11,13,17,19,23,29\}$, $m = 100 : n = \{1,2,3,4,5,6\}$, $m = 200 : n = \{2,4,6,8,10,12\}$, ..., $m = 900 : n = \{9,18,27,36,45,54\}$

For each triplet $\{p,m,n\}$, we generate 300 different random texts. Thus we consider 129,600 different random texts, each of them is enciphered via the Hill cipher with random key-matrices. To compare properly the OBA and the BFA, we decipher each ciphertext with the two attacks; therefore 259,200 experiments are conducted with a time-out of 9,000 seconds, representing 73 years of computations in the worst-case.

First of all let us give the ratio of how many times the OBA has been faster than the BFA over the 129,600 different texts: it is equal to 99.91%. This shows that the OBA is faster in most of the cases considered here than the BFA. This first result shows that the OBA seems to furnish an attack for the Hill cipher more efficient than the classical OBA on the set of considered parameters.

To access the results, we redirect the reader to the following external link: `https://bit.ly/2QeYhH2`. They permit to compare the time-executions for the two attacks: in each sub-figure, each point represents a text whose x-coordinate and y-coordinates are respectively the time needed by the BFA and OBA. A point localised below the diagonal means that the OBA has been faster than the BFA. The first part of the Figures focuses on the parameters p and m (the different values for n are not distinguished) while the second part focuses on p and n (the different values for m are not distinguished). Basically, these Figures demonstrate that

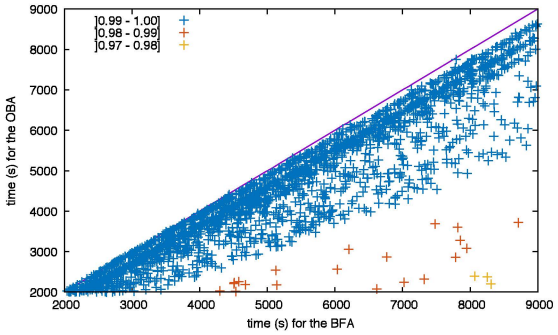

Figure 2: Scatter-plots of BFA vs OBA. Each dot corresponds to a random text where there is, at least, one block which is a linear combination of the others.

the time-execution tends to increase when m or n become large: this is logical since, in these cases, the text contains more characters and requires hence more computations to be decrypted. We also observe a small impact of the parameter p on the gain: the larger p is, the larger the number of points below the diagonal seems to be. This means that the OBA seems to be more efficient than the BFA when one studies ciphertexts in rich alphabets.

Impact of the number of independent blocks

As explained in Section 2.2, the size of the orbit of a ciphertext C, whose influence on the computational complexity of the OBA can be seen through τ_1, depends on the number q of linearly independent blocks within C; by standard linear algebra arguments, we have $q \leq n$. Obviously the number of possible key-matrix does not depend on this parameter q and so, whatever the value of q is, the complexity of the BFA remains the same. Hence one expects the OBA deciphers faster on average ciphertexts having $q < n$ independent blocks than the BFA does. Our aim here is to illustrate this fact in practice by using our experiments.

Min	1^{st} Qu.	Median	Mean	3^{rd} Qu.	Max
0.973	0.987	0.993	0.995	0.999	1.00

Table 1: Summary of $\frac{q}{n}$ over all our examples.

For the sake of completeness, a simple statistical study of the ratio $\frac{q}{n}$ over our 129,600 texts is given in Table 1. As expected, this ratio is generally close to 1 for our random texts, as expected, only few texts do not have the maximum number of independent blocks. One interesting issue would be to make the same simple study for meaningful texts in a given language.

In Figure 2, we observe the distribution of the time-executions but with respect to the ratio $\frac{q}{n}$; note that the plot starts at 2,000 seconds. It is clear that the percentage of independent blocks impacts strongly the time-execution of the OBA: the fewer independent blocks the ciphertext has, the faster the decryption via OBA

is. While such texts are rare according to the above table, the OBA takes a huge advantage against the BFA in such cases, even in the case where the ratio is close (but different) to 1.

6 Conclusion

In this paper, we formalised the Orbit-Based Attack, whose principle has been firstly introduced in (McDevitt et al., 2018), by applying basic notions from group action theory; this provides a new type of ciphertext-only attack for Hill. This attack is based on the fact that Hill can not cipher an input text to any output one: the latter belongs necessarily to an explicit set associated to the former, namely its orbit, whose size is proved to be smaller than the one of the key-space. This attack consists then in making a search over only the orbit of the ciphertext.

We focused then on the computational complexity and time-execution of an algorithm for the OBA. Even if this algorithm has the same complexity as the one of the classical Brute-Force Attack (consisting in testing all the invertible matrices) in the worst case, our experiments show that this algorithm is faster on average than the Brute-Force Attack in practice. Discussions on the influences of some parameters of the text on the gain in terms of runtime of the OBA over the BFA are given. In particular, our theoretical and experimental results exhibit an interesting gain in the particular situation where the text has not the maximal number of linearly independent blocks.

We finish by discussing on the outlook. Considering ciphertexts for which some prior knowledge on the language are available is an interesting issue, reducing potentially the computational complexity of the OBA by using relevant statistical tools. A hope would be to refine the results obtained in (McDevitt et al., 2018) thanks our formalisation. A comparison with the Row-By-Row Attack from (Bauer and Millward, 2007; Yum and Lee, 2009; Leap et al., 2016) would be interesting as well.

We mention that some steps of the algorithm could be refined in view of optimisation of the attack. Moreover, except the LU-type decomposition, the rest of the algorithm can be parallelized, reducing potentially the runtime of the OBA. An empirical evaluation could be then performed as future works.

A last interesting issue would be to study whether the principle of the Orbit-Based Attack can be applied to other polygraphic substitution ciphers.

Acknowledgments Valentin Montmirail has been supported by IDEX UCA$^{\text{JEDI}}$. The authors thank also T. Defourneau for discussions on group action theory.

References

Michael Artin. 2011. *Algebra*, volume 2. Pearson Prentice Hall, Advanced Mathematics Series edition.

Craig P. Bauer and Katherine Millward. 2007. Cracking Matrix Encryption Row by Row. *Cryptologia*, 31(1):76–83.

Gene H. Golub and Charles F. Van Loan. 2012. *Matrix computations*, volume 3. JHU Press.

Lester Sanders Hill. 1929. Cryptography in an Algebraic Alphabet. *The American Mathematical Monthly*, 36(6):306–312.

Lester Sanders Hill. 1931. Concerning Certain Linear Transformation Apparatus of Cryptography. *The American Mathematical Monthly*, 38:135–154.

I. A. Ismail, Mohammed Amin, and Hossam Diab. 2006. How to Repair the Hill Cipher. *Journal of Zhejiang University-Science A*, 7(12):2022–2030, Dec.

Shahram Khazaei and Siavash Ahmadi. 2017. Ciphertext-only Attack on d × d Hill in $O(d \times 13^d)$. *Inf. Process. Lett.*, 118:25–29.

Tom Leap, Tim McDevitt, Kayla Novak, and Nicolette Siermine. 2016. Further Improvements to the Bauer-Millward Attack On the Hill Cipher. *Cryptologia*, 40(5):452–468.

Ahmed Y. Mahmoud and Alexander G. Chefranov. 2009. Hill Cipher Modification Based on Eigenvalues HCM-EE. In *Proc. of SIN'09*, pages 164–167.

Tim McDevitt, Jessica Lehr, and Ting Gu. 2018. A parallel time-memory tradeoff attack on the hill cipher. *Cryptologia*, 42(5):408–426.

Jeffrey Overbey, William Traves, and Jerzy Wojdylo. 2005. On the Keyspace of the Hill Cipher. *Cryptologia*, 29(1):59–72.

Christos H. Papadimitriou. 1994. *Computational complexity.* Addison-Wesley.

J. Rotman. 1965. *The Theory of Groups*. Boston: Allyn and Bacon.

Jonathan D.H. Smith. 2008. *Introduction to Abstract Algebra*. Textbooks in Mathematics. Chapman and Hall/CRC.

Douglas Stinson. 2002. *Cryptography: Theory and Practice*. CRC/C&H, second edition.

Mohsen Toorani and Abolfazl Falahati. 2009. A Secure Variant of the Hill Cipher. In *Proc. of 14th IEEE Symposium on Computers and Communications (ISCC'09)*, pages 313–316.

Mohsen Toorani and Abolfazl Falahati. 2011. A Secure Cryptosystem Based on Affine Transformation. *Security and Communication Networks*, 4(2):207–215.

Samuel S. Wagstaff. 2002. *Cryptanalysis of Number Theoretic Ciphers*. CRC Press, Inc., Boca Raton, FL, USA.

Dae Hyun Yum and Pil Joong Lee. 2009. Cracking Hill Ciphers with Goodness-of-Fit Statistics. *Cryptologia*, 33(4):335–342.