STREAM/BLOCK CIPHERS, DIFFERENCE EQUATIONS AND ALGEBRAIC ATTACKS

ROBERTO LA SCALA* AND SHARWAN K. TIWARI**

ABSTRACT. In this paper we model a class of stream and block ciphers as systems of (ordinary) explicit difference equations over a finite field. We call this class "difference ciphers" and we show that ciphers of application interest, as for example systems of LFSRs with a combiner, TRIVIUM and KEELOQ, belong to the class. By using Difference Algebra, that is, the formal theory of difference equations, we can properly define and study important properties of these ciphers, such as their invertibility and periodicity. We describe then general cryptanalytic methods for difference ciphers that follow from these properties and are useful to assess the security. We illustrate such algebraic attacks in practice by means of the ciphers BIVIUM and KEELOQ.

1. INTRODUCTION

Algebraic cryptanalysis concerns the possibility to perform an attack to a cipher as an instance of polynomial system solving. This idea dates back at least to Shannon who in his foundational paper of modern Cryptography [36] wrote that security can be essentially assumed "if we could show that solving a certain (crypto)system requires at least as much work as solving a system of simultaneous equations in a large number of unknowns, of a complex type". In the last 20 years, these cryptanalytic ideas have become actual algorithms and implementations by the work of many cryptographers: see, for instance, the long reference list of the comprehensive book of G. Bard entitled "Algebraic Cryptanalysis" [3]. Among this vast literature, we would like to mention the pioneering work of N. Courtois [12, 13] and his methods for polynomial system solving, such as XL and ElimLin [11]. It is important to reference also the role played by the F4-F5 algorithms due to J.-C. Faugère [21, 22] for providing complexity formulas for Gröbner bases computations over (semi)regular systems [5, 4]. Indeed, such complexity is much lower than in the worst case and these systems naturally arise in the context of multivariate cryptography. In other words, such complexity formulas provide essential cryptanalysis for multivariate protocols as Rainbow [7] which is among the three NIST post-quantum signature finalists.

The usual formalism of algebraic cryptanalysis are systems of polynomial equations and generally Commutative Algebra (over finite fields) but, as a matter of

---

2000 *Mathematics Subject Classification.* Primary 11T71. Secondary 12H10, 13P10.

Key words and phrases. Stream and block ciphers; Algebraic difference equations; Gröbner bases.

The first author acknowledges the support of the University of Bari, Grant ref. 73251. The second author thanks the Scientific Analysis Group, DRDO, Delhi, for the support. DOI: https://doi.org/10.1016/j.jsc.2021.09.001 .

fact, a natural modeling of many stream and block ciphers are systems of (algebraic ordinary) explicit difference equations. In fact, such ciphers are defined as recursive rules determining the evolution of a vector, with entries in a finite field, which is called the state or register of the cipher. This evolution runs along a discrete time corresponding to clocks or rounds. Because of their simple structure, explicit difference systems provide such a recursion, that is, the existence and uniqueness of solutions once given their initial state. Despite this simplicity, the solutions of these systems can be extremely involved and hence interesting for the purposes of Cryptography. We mention, for instance, the complex evolution of a discrete predator-prey model [1] or the security properties of the stream cipher Trivium [14] where both underlying systems are just few quadratic explicit difference equations.

The formal theory of algebraic difference equations is called Difference Algebra and it was introduced by J.F. Ritt [35] as a discrete couterpart of his celebrated Differential Algebra. This theory provides important insights about the structure of the solutions of a system of difference equations (see [8, 31, 40, 41]). In particular, by mimicking Commutative Algebra, one has the notion of difference ideal and corresponding difference variety, as well the methods of difference Gröbner bases [16, 30]. The use of difference equations in Cryptography has a long tradition because of Linear Feedback Shift Registers (briefly LFSRs) which are just linear difference equations over the field with two elements. These very simple recursive rules have been used at least since the 1950s to obtain, for instance, pseudorandom number generators. For a classical reference, see [17]. The problem with LFSRs is that they are easy to attack because of linearity and hence a standard strategy to enhance security consists in considering a system of LFSRs together with a combining (or filtering) non-linear function. Many pseudorandom generators and stream ciphers are of this kind such as the Geffe generator and the cipher E0 of the Bluetooth protocol. We reference the book [29] for a detailed guide to them.

Another option consists in having non-linear explicit difference equations governing the evolution of the internal state of the cipher and a linear function for defining the keystream output. The prototype of such stream cipher is Trivium. Note that its explicit system has a state transition function which is invertible. This is a possible flaw because some opponents may recover the initial state containing the key by attacking any internal state. Despite many attacks also of this kind [20, 26, 33], the three quadratic explicit difference equations of Trivium remain inexpugnable. On the other hand, if an invertible system contains a subsystem that can be used to evolve independently the keys, this flaw becomes a resource for defining a block cipher. These ideas appear in the definition of the block cipher KeeLoq which has been cryptanalized in a critical way [9, 10] because of the short period of the state transition function of its key subsystem.

The present paper is organized as follows. In Section 2, for any base field, we introduce the formalism of Difference Algebra for the purpose of providing a precise definition of systems of (ordinary) explicit difference equations from an algebraic viewpoint. We introduce then the notion of state transition endomorphism and we apply the methods of difference Gröbner bases of difference ideals [16, 30] to obtain a key property contained in Theorem 2.13. This property, together with Theorem 5.4, will be used in Section 5 to obtain the equations satisfied by the keys for a general algebraic attack to a difference stream cipher.

In Sections 3 and 4, by means of the notion of state transition endomorphism we define key properties of explicit systems such as their invertibility and periodicity. The use of symbolic computation, namely Gröbner bases, provides effective methods to check invertibility and compute inverse systems which is essential for the cryptanalysis of difference ciphers. In Section 4 we also review, from an algebraic point of view, the theory to maximize the period of a system of LFSRs over a prime finite field.

In Section 5, we finally introduce the class of "difference stream and block ciphers" as ciphers that are defined by explicit difference systems over a finite field. The motivation is that many ciphers of application interests can be easily modeled in this way and we can provide general methods for the algebraic cryptanalysis of this class. This theory can be used therefore for developing new ciphers. As already mentioned, in Section 5 we obtain the equations satisfied by the keys of a difference stream cipher which are compatible with a known keystream. We show that a finite number of elements of the keystream are enough to provide these equations. By means of the invertibility property studied in Section 3, we also suggest how to reduce the degree of the equations. We discuss then guess-and-determine strategies for polynomial system solving over a finite field and provide details for the case that the system has a single solution. Indeed, this is often the case in the cryptanalysis of stream ciphers once a sufficient amount of elements of the keystream is known. For difference block ciphers we discuss a general algebraic attack which follows from the condition that the key subsystem has a short period. This attack has been introduced for the cipher KeeLoq in the papers [9, 10].

Finally, in Section 6 and 7 we illustrate the above algebraic attacks by applying them to concrete ciphers, namely Bivium and KeeLoq. Although this is not our primary goal in the present paper, we are able to obtain speedup with respect to similar attacks to Bivium [20]. We also improve the previous polynomial system solving time [9, 10] in the attack to KeeLoq. An interesting comparison of Gröbner bases vs SAT solvers is also provided in our tests. Some conclusions are draw in Section 8.

2. Explicit difference system

Let $\mathbb{K}$ be any field and fix an integer $n > 0$. Consider a set of variables $X(t) = \{x_1(t), \ldots, x_n(t)\}$, for any $t \in \mathbb{N} = \{0, 1, 2, \ldots\}$. Put $X = \bigcup_{t\geq 0} X(t)$ and denote by $R = \mathbb{K}[X]$ the polynomial algebra in the infinite set of variables X. Moreover, consider the injective algebra endomorphism $\sigma : R \to R$ such that $x_i(t) \mapsto x_i(t+1)$ for all $1 \leq i \leq n$ and $t \geq 0$. We call σ the *shift map of* R. The algebra R, endowed with the map σ, is called the *algebra of ordinary difference polynomials (with constant coefficients)*. We also need the following notations. For any integers $r_1, \ldots, r_n \geq 0$ and $t \geq 0$, we define the subset

$$\bar{X} = \{x_1(0), \ldots, x_1(r_1 - 1), \ldots, x_n(0), \ldots, x_n(r_n - 1)\} \subset X$$

and the subalgebra $\bar{R} = \mathbb{K}[\bar{X}] \subset R$.

Definition 2.1. *Let $r_1, \ldots, r_n \geq 0$ be integers and consider some polynomials $f_1, \ldots, f_n \in \bar{R}$. A* system of (algebraic ordinary) explicit difference equations *is by* *definition an infinite system of polynomial equations of the kind*

$$\begin{cases} x_1(r_1+t) &=& \sigma^t(f_1), \\ & \vdots & \\ x_n(r_n+t) &=& \sigma^t(f_n). \end{cases} \quad (t \geq 0)$$

Such a system is denoted briefly as

$$\begin{cases} x_1(r_1) &=& f_1, \\ & \vdots & \\ x_n(r_n) &=& f_n. \end{cases} \tag{1}$$

A $\mathbb{K}$-solution of the system (1) *is an* n*-tuple of functions* $(a_1,\ldots,a_n)$ *where each* $a_i : \mathbb{N} \to \mathbb{K}$ *satisfies the equations* $x_i(r_i+t) = \sigma^t(f_i)$*, for all* $t \geq 0$*. The element* $a_i(t) \in \mathbb{K}$ *is called the* value of the function a_i at the clock $t \geq 0$*.*

Definition 2.2. *Consider an explicit difference system* (1)*. We define the algebra endomorphism* $\bar{\mathrm{T}} : \bar{R} \to \bar{R}$ *such that, for any* $i = 1,2,\ldots,n$

$$x_i(0) \mapsto x_i(1), \ldots, x_i(r_i-2) \mapsto x_i(r_i-1), x_i(r_i-1) \mapsto f_i.$$

If $r = r_1 + \ldots + r_n$*, we denote by* $\mathrm{T} : \mathbb{K}^r \to \mathbb{K}^r$ *the polynomial map corresponding to* $\bar{\mathrm{T}}$*. For any polynomial* $f \in \bar{R}$ *and for each vector* $v \in \mathbb{K}^r$*, one has that*

$$\bar{\mathrm{T}}(f)(v) = f(\mathrm{T}(v)). \tag{2}$$

If $(a_1,\ldots,a_n)$ *is a* $\mathbb{K}$*-solution of* (1)*, we call the vector*

$$v(t) = (a_1(t),\ldots,a_1(t+r_1-1),\ldots,a_n(t),\ldots,a_n(t+r_n-1)) \in \mathbb{K}^r$$

the state of $(a_1,\ldots,a_n)$ at the clock $t \geq 0$*. In particular,* $v(0)$ *is the* initial state of $(a_1,\ldots,a_n)$*. Then, the function* T *maps the* t*-state* $v(t)$ *into the* $(t+1)$*-state* $v(t+1)$*, for all clocks* $t \geq 0$*. We call* $\bar{\mathrm{T}}$ *the* state transition endomorphism *and* T *the* state transition map of the explicit difference system (1)*.*

Example 2.3. *Let* $\mathbb{K} = \mathbb{Q}$ *and consider the following explicit system*

$$\begin{cases} x(1) &=& x(0)^2 + y(0)^2, \\ y(1) &=& 2x(0)y(0). \end{cases}$$

If $\bar{R} = \mathbb{K}[x(0),y(0)]$*, the corresponding state transition endomorphism* $\bar{\mathrm{T}} : \bar{R} \to \bar{R}$ *is defined as* $x(0) \mapsto x(0)^2 + y(0)^2, y(0) \mapsto 2x(0)y(0)$*. One computes that the* $\mathbb{K}$*-solutions of the above system are the functions* $a, b : \mathbb{N} \to \mathbb{K}$ *such that, for all* $t \geq 0$

$$a(t) = \frac{(a(0)+b(0))^{2^t} + (a(0)-b(0))^{2^t}}{2},$$
$$b(t) = \frac{(a(0)+b(0))^{2^t} - (a(0)-b(0))^{2^t}}{2}.$$

We have the following existence and uniqueness theorem for the solutions of an explicit system.

Theorem 2.4. *Denote by* $V_{\mathbb{K}}$ *the set of all* $\mathbb{K}$*-solutions of the explicit difference system* (1)*. We have a bijective map* $\iota : V_{\mathbb{K}} \to \mathbb{K}^r$ *such that*

$$(a_1,\ldots,a_n) \mapsto (a_1(0),\ldots,a_1(r_1-1),\ldots,a_n(0),\ldots,a_n(r_n-1)).$$

In other words, the system (1) *has a unique* $\mathbb{K}$*-solution once fixed its initial state. Moreover, the maps* ι, ι^{-1} *are both polynomial ones.*

Proof. Consider the state transition map $\mathrm{T} : \mathbb{K}^r \to \mathbb{K}^r$ of (1) which is a polynomial map. Observe that all powers $\mathrm{T}^t : \mathbb{K}^r \to \mathbb{K}^r$ $(t \geq 0)$ are also polynomial maps. If

$$v(t) = (a_1(t), \ldots, a_1(t + r_1 - 1), \ldots, a_n(t), \ldots, a_n(t + r_n - 1))$$

denotes the t-state of a $\mathbb{K}$-solution $(a_1, \ldots, a_n) \in V_{\mathbb{K}}$, the inverse map

$$\iota^{-1} : v(0) \mapsto (a_1, \ldots, a_n)$$

is obtained in the following way. The value $a_1(t)$ is the first coordinate of the vector $v(t) = \mathrm{T}^t(v(0))$, $a_2(t)$ is its $(r_1 + 1)$-th coordinate and so on. Since projections and T^t are polynomial maps, we conclude that ι^{-1} is also such a map. □

Consider the state transition endomorphism $\bar{\mathrm{T}} : \bar{R} \to \bar{R}$ of the system (1). Note that all powers $\bar{\mathrm{T}}^t : \bar{R} \to \bar{R}$ $(t \geq 0)$ are also endomorphisms whose corresponding polynomial maps are the functions $\mathrm{T}^t : \mathbb{K}^r \to \mathbb{K}^r$. For all $1 \leq i \leq n$ and $t \geq 0$, we define the polynomial

$$f_{i,t} = \bar{\mathrm{T}}^t(x_i(0)) \in \bar{R}.$$

By the argument of Theorem 2.4 and the identity (2), it follows that $a_i(t) = f_{i,t}(v(0))$, for all $\mathbb{K}$-solutions $(a_1, \ldots, a_n) \in V_{\mathbb{K}}$.

We briefly introduce now the notion of difference Gröbner basis which provides very often an alternative way to compute the polynomial $f_{i,t}$. For a complete reference we refer to [16, 30].

Definition 2.5. *Let I be an ideal of the algebra R. We call I a* difference ideal *if $\sigma(I) \subset I$. Denote $\Sigma = \{\sigma^t \mid t \geq 0\}$ and let G be a subset of R. Then, we define $\Sigma(G) = \{\sigma^t(g) \mid g \in G, t \geq 0\} \subset R$. We call G a* difference basis *of a difference ideal I if $\Sigma(G)$ is a basis of I as an ideal of R. In other words, all elements $f \in I$ are such that $f = \sum_i f_i \sigma^{t_i}(g_i)$ where $f_i \in R, g_i \in G$ and $t_i \geq 0$. In this case, we denote $\langle G \rangle_\sigma = \langle \Sigma(G) \rangle = I$.*

Consider an explicit difference system (1) and define the subset

$$G = \{x_1(r_1) - f_1, \ldots, x_n(r_n) - f_n\} \subset R.$$

If $I = \langle G \rangle_\sigma$, we have that $(a_1, \ldots, a_n)$ is a $\mathbb{K}$-solution of the system (1) if and only if this is a simultaneous $\mathbb{K}$-solution of all polynomials $f \in I$. In other words, by substituting the variables $x_i(t)$ of each $f \in I$ with the elements $a_i(t) \in \mathbb{K}$, one always obtains zero. Then, we also say that $(a_1, \ldots, a_n)$ is a $\mathbb{K}$-*solution of the difference ideal* I and we put $V_{\mathbb{K}}(I) = V_{\mathbb{K}}$. For defining Gröbner bases, one needs to introduce monomial orderings on R.

Definition 2.6. *Let $\prec$ be a total ordering on the set $M = \mathrm{Mon}(R)$ of all monomials of R. We call $\prec$ a* monomial ordering of R *if the following properties hold:*

(i) *$\prec$ is a multiplicatively compatible ordering, that is, if $m' \prec m''$ then $mm' \prec mm''$, for all $m, m', m'' \in M$;*
(ii) *$\prec$ is a well-ordering, that is, every non-empty subset of M has a minimal element.*

In this case, it follows that

(iii) *$1 \prec m$, for all $m \in M, m \neq 1$.*

Indeed, from $1 \succ m$ and the property (i) it follows that we have an infinite strictly decreasing sequence

$$1 \succ m \succ m^2 \succ \ldots$$

which contradicts the property (ii). Even though the variables set X is infinite, by Higman's Lemma [25] the polynomial algebra $R = K[X]$ can always be endowed with a monomial ordering. For the following version of this key result, see for instance [2], Corollary 2.3 and also the remarks at the beginning of page 5175 of that reference.

Proposition 2.7. *Let $\prec$ be a total ordering on M which verifies the properties $(i),(iii)$ of Definition 2.6. If the restriction of $\prec$ to the variables set $X \subset M$ is a well-ordering then $\prec$ is also a well-ordering on M, that is, it is a monomial ordering of R.*

To introduce difference Gröbner bases, we need monomial orderings that are compatible with the shift map.

Definition 2.8. *Let $\prec$ be a monomial ordering of R. We call $\prec$ a* difference monomial ordering of R *if $m \prec m'$ implies that $\sigma(m) \prec \sigma(m')$, for all $m, m' \in M$.*

Note that if $\prec$ is a difference monomial ordering, we have that $m \prec \sigma(m)$, for all $m \in M, m \neq 1$. Indeed, by assuming $m \succ \sigma(m)$ one obtains an infinite strictly decreasing sequence

$$m \succ \sigma(m) \succ \sigma^2(m) \succ \ldots$$

which contradicts the property of $\prec$ of being a well-ordering.

An important class of difference monomial orderings can be defined in the following way. Recall that all polynomial algebras $R(t) = K[X(t)]$ $(t \geq 0)$ are in fact isomorphic by means of the shift map. Then, let us consider the same monomial ordering for all such algebras. Since $R = \bigotimes_{t \geq 0} R(t)$, we can define on R the product monomial ordering such that $X(0) \prec X(1) \prec \ldots$. For any choice of a monomial ordering on $R(0)$, this is a difference monomial ordering of R that we call *clock-based*.

From now on, we assume that R is endowed with a difference monomial ordering. Let $f = \sum_i c_i m_i \in R$ with $m_i \in M$ and $0 \neq c_i \in \mathbb{K}$. If $m_k = \max_{\prec}\{m_i\}$, we put $\mathrm{lm}(f) = m_k, \mathrm{lc}(f) = c_k$ and $\mathrm{lt}(f) = c_k m_k$. Since $\prec$ is a difference ordering, one has that $\mathrm{lm}(\sigma(f)) = \sigma(\mathrm{lm}(f))$ and hence $\mathrm{lc}(\sigma(f)) = \mathrm{lc}(f), \mathrm{lt}(\sigma(f)) = \sigma(\mathrm{lt}(f))$. If $G \subset R$, we denote $\mathrm{lm}(G) = \{\mathrm{lm}(f) \mid f \in G, f \neq 0\}$ and we put $\mathrm{LM}(G) = \langle \mathrm{lm}(G) \rangle$. Let I be an ideal of R. A polynomial $f = \sum_i c_i m_i \in R$ is called *normal modulo* I if $m_i \notin \mathrm{LM}(I)$, for all i. Since R is endowed with a monomial ordering, by a reduction process (see [16, 30]) one proves that for each polynomial $f \in R$ there is a unique element $\mathrm{NF}_I(f) \in R$ such that $f - \mathrm{NF}_I(f) \in I$ and $\mathrm{NF}_I(f)$ is a normal polynomial modulo I. In other words, the cosets of the normal monomials modulo I form a $\mathbb{K}$-linear basis of the quotient algebra R/I. We call the polynomial $\mathrm{NF}_I(f)$ the *normal form of f modulo I*.

Proposition 2.9. *Let $G \subset R$. Then $\mathrm{lm}(\Sigma(G)) = \Sigma(\mathrm{lm}(G))$. In particular, if I is a difference ideal of R then $\mathrm{LM}(I)$ is also a difference ideal.*

Proof. Since R is endowed with a difference monomial ordering, one has that $\mathrm{lm}(\sigma(f)) = \sigma(\mathrm{lm}(f))$, for any $f \in R, f \neq 0$. Then, $\Sigma(\mathrm{lm}(I)) = \mathrm{lm}(\Sigma(I)) \subset \mathrm{lm}(I)$ and therefore $\mathrm{LM}(I) = \langle \mathrm{lm}(I) \rangle$ is a difference ideal. □

Definition 2.10. *Let $I \subset R$ be a difference ideal and $G \subset I$. We call G a* difference Gröbner basis *of I if $\mathrm{lm}(G)$ is a difference basis of $\mathrm{LM}(I)$. In other words, $\mathrm{lm}(\Sigma(G)) = \Sigma(\mathrm{lm}(G))$ is a basis of $\mathrm{LM}(I)$, that is, $\Sigma(G)$ is a Gröbner basis of I as an ideal of R.*

For more details about difference Gröbner bases and an optimized version of the Buchberger procedure for these bases, we refer to [16, 30].

Proposition 2.11. *Consider an explicit difference system* (1) *and assume that* R *is endowed with a difference monomial ordering such that* $x_i(r_i) \succ \mathrm{lm}(f_i)$*, for all* $1 \leq i \leq n$*. Then, the set* $G = \{x_1(r_1) - f_1, \ldots, x_n(r_n) - f_n\}$ *is a difference Gröbner basis.*

Proof. From the assumption on the monomial ordering it follows that $x_i(r_i + t) = \mathrm{lm}(x_i(r_i + t) - \sigma^t(f_i))$, for any $1 \leq i \leq n$ and $t \geq 0$. By the linearity of these distinct leading monomials and the Buchberger's Product Criterion (see, for instance, [18]) we conclude that $\Sigma(G)$ is a Gröbner basis, that is, G is a difference Gröbner basis. $\square$

From now on, we assume that $x_i(r_i) \succ \mathrm{lm}(f_i)$, for any i. If $I \subset R$ is the difference ideal generated by the set $G = \{x_1(r_1) - f_1, \ldots, x_n(r_n) - f_n\}$, the above result implies that $\mathrm{LM}(I) = \langle x_1(r_1), \ldots, x_n(r_n) \rangle_\sigma \subset R$. In other words, the set of normal polynomials modulo I is exactly the subalgebra $\bar{R} = \mathbb{K}[\bar{X}]$ where by definition $\bar{X} = \{x_1(0), \ldots, x_1(r_1 - 1), \ldots, x_n(0), \ldots, x_n(r_n - 1)\}$.

Proposition 2.12. *The map* $\eta : R \to \bar{R}, f \mapsto \mathrm{NF}_I(f)$ *is an algebra homomorphism. In other words, one has the algebra isomorphism* $\eta' : R/I \to \bar{R}$ *such that* $f + I \mapsto \mathrm{NF}_I(f)$*.*

Proof. By definition, we have that η is a surjective $\mathbb{K}$-linear map and $\mathrm{Ker}\,\eta = I$. Then, it is sufficient to show that $mm' \notin \mathrm{LM}(I)$, for all monomials $m, m' \notin \mathrm{LM}(I)$. This holds because $\mathrm{LM}(I)$ is an ideal which is generated by variables. $\square$

Theorem 2.13. *Let* $\bar{\mathrm{T}} : \bar{R} \to \bar{R}$ *be the state transition endomorphism of the system* (1) *and consider the algebra endomorphism* $\sigma' : R/I \to R/I$ *such that* $f + I \mapsto \sigma(f) + I$*. Then, one has that* $\bar{\mathrm{T}}\eta' = \eta'\sigma'$*. In particular, for each polynomial* $f \in \bar{R}$ *and for all* $t \geq 0$*, we have that* $\bar{\mathrm{T}}^t(f) = \mathrm{NF}_I(\sigma^t(f))$*.*

Proof. Consider a polynomial $f \in \bar{R}$, that is, $f = \mathrm{NF}_I(f)$. The polynomial $\bar{\mathrm{T}}(f) \in \bar{R}$ is obtained from the polynomial $\sigma(f) \in R$ simply by applying the identities $x_i(r_i) = f_i$ $(1 \leq i \leq n)$. Because $x_i(r_i) - f_i \in I$, we conclude that $\sigma(f) - \bar{\mathrm{T}}(f) \in I$. $\square$

Observe finally that the above result implies that $f_{i,t} = \bar{\mathrm{T}}^t(x_i(0)) = \mathrm{NF}_I(x_i(t))$.

3. Invertible systems

An important class of explicit difference systems are the ones such that a t-state can be obtained from a t'-state also for $t' \geq t$.

Definition 3.1. *For an explicit difference system* (1)*, consider the state transition endomorphism* $\bar{\mathrm{T}} : \bar{R} \to \bar{R}$ *and the corresponding state transition map* $\mathrm{T} : \mathbb{K}^r \to \mathbb{K}^r$ $(r = r_1 + \ldots + r_n)$*. We call the system* invertible *if* $\bar{\mathrm{T}}$ *is an automorphism. In this case,* T *is also a bijective map.*

We state now an algorithmic method to establish if an endomorphism of a polynomial algebra is invertible and to compute its inverse. This important result is due to Arno van den Essen (see [39], Theorem 3.2.1). Recall that a Gröbner basis $G = \{g_1, \ldots, g_r\}$ is called *(completely) reduced* if the polynomial g_i is normal modulo the ideal generated by $G \setminus \{g_i\}$, for all $1 \leq i \leq r$.

Theorem 3.2. *Let $X = \{x_1, \dots, x_r\}, X' = \{x'_1, \dots, x'_r\}$ be two disjoint variable sets and define the polynomial algebras $P = \mathbb{K}[X], P' = \mathbb{K}[X']$ and $Q = \mathbb{K}[X \cup X'] = P \otimes P'$. Consider an algebra endomorphism $\varphi : P \to P$ such that $x_1 \mapsto g_1, \dots, x_r \mapsto g_r$ $(g_i \in P)$ and the corresponding ideal $J \subset Q$ which is generated by the set $\{x'_1 - g_1, \dots, x'_r - g_r\}$. Moreover, we endow the polynomial algebra Q by a product monomial ordering such that $X \succ X'$. Then, the map φ is an automorphism of P if and only if the reduced Gröbner basis of J is of the kind $\{x_1 - g'_1, \dots, x_r - g'_r\}$ where $g'_i \in P'$, for all $1 \leq i \leq r$. In this case, if $\varphi' : P' \to P'$ is the algebra endomorphism such that $x'_1 \mapsto g'_1, \dots, x'_r \mapsto g'_r$ and $\xi : P \to P'$ is the isomorphism $x_1 \mapsto x'_1, \dots, x_r \mapsto x'_r$, we have that $\xi \, \varphi^{-1} = \varphi' \, \xi$.*

For the context of explicit difference systems, the above criterion implies the following results.

Corollary 3.3. *Let $\bar{\mathrm{T}} : \bar{R} \to \bar{R}$ be the state transition automorphism corresponding to an invertible system (1), namely $(1 \leq i \leq n)$*

$$x_i(0) \mapsto x_i(1), \dots, x_i(r_i - 2) \mapsto x_i(r_i - 1), x_i(r_i - 1) \mapsto f_i.$$

Denote $\bar{R}' = \mathbb{K}[\bar{X}']$ where

$$\bar{X}' = \{x'_1(0), \dots, x'_1(r_1 - 1), \dots, x'_n(0), \dots, x'_n(r_n - 1)\}$$

and put $Q = \bar{R} \otimes \bar{R}'$. Consider the ideal $J \subset Q$ that is generated by the following polynomials, for any $i = 1, 2, \dots, n$

$$x'_i(0) - x_i(1), \dots, x'_i(r_i - 2) - x_i(r_i - 1), x'_i(r_i - 1) - f_i.$$

With respect to a product monomial ordering of the algebra Q such that $\bar{X} \succ \bar{X}'$, the reduced Gröbner basis of J has the following form

$$x_i(1) - x'_i(0), \dots, x_i(r_i - 1) - x'_i(r_i - 2), x_i(0) - f'_i$$

where $f'_i \in \bar{R}'$, for all $1 \leq i \leq n$.

Proof. With respect to the considered monomial ordering of Q, we have clearly that the set

$$G = \bigcup_i \{x_i(1) - x'_i(0), \dots, x_i(r_i - 1) - x'_i(r_i - 2)\} \subset J$$

is the reduced Gröbner basis of the ideal of Q that is generated by it. Since $\bar{\mathrm{T}}$ is an automorphism, by the Theorem 3.2 we conclude that there are polynomials $f'_i \in \bar{R}'$ $(1 \leq i \leq n)$ such that the set

$$G \cup \bigcup_i \{x_i(0) - f'_i\}$$

is the reduced Gröbner basis of the ideal $J \subset Q$. □

By the above result, we obtain a sufficient condition to invertibility which is immediate to verify.

Corollary 3.4. *Consider an explicit difference system (1) and assume that $f_i = x_{k_i}(0) + g_i$ where $\{x(0)_{k_1}, \dots, x(0)_{k_n}\} = X(0) = \{x_1(0), \dots, x_n(0)\}$ and the polynomial g_i has all variables in the set $\bar{X} \setminus X(0)$, for all $1 \leq i \leq n$. Then, the system (1) is invertible.*

Proof. With the same notations of Corollary 3.3, consider the set

$$G = \bigcup_i \{x_i(1) - x'_i(0), \ldots, x_i(r_i - 1) - x'_i(r_i - 2)\} \subset J$$

and assume that the algebra $Q = \bar{R} \otimes \bar{R}'$ is endowed with a product monomial ordering such that $\bar{X} \succ \bar{X}'$. Since the variables of g_i are in $\bar{X} \setminus X(0)$, the normal form g'_i modulo the ideal generated by G is a polynomial with variables in the set

$$\bar{X}' \setminus \{x'_1(r_1 - 1), \ldots, x'_n(r_n - 1)\}.$$

Then, the reduced Gröbner basis of the ideal $J \subset Q$ is given by the following polynomials, for any $i = 1, 2, \ldots, n$

$$x_i(1) - x'_i(0), \ldots, x_i(r_i - 1) - x'_i(r_i - 2), x_{k_i}(0) - x'_i(r_i - 1) - g'_i.$$

By Theorem 3.2 we conclude that $\bar{\mathrm{T}}$ is an automorphism, that is, (1) is an invertible system. ☐

Definition 3.5. *For an explicit difference system* (1), *consider the ideal* $J \subset Q = \bar{R} \otimes \bar{R}'$ *which is generated by the following polynomials, for each* $i = 1, 2, \ldots, n$

$$x'_i(0) - x_i(1), \ldots, x'_i(r_i - 2) - x_i(r_i - 1), x'_i(r_i - 1) - f_i.$$

We call J *the* state transition ideal of the system (1).

From now on, we assume that Q is endowed with a product monomial ordering such that $\bar{X} \succ \bar{X}'$.

Definition 3.6. *Consider an invertible system* (1) *and the corresponding state transition ideal* $J \subset Q$. *If the set*

$$G = \bigcup_i \{x_i(1) - x'_i(0), \ldots, x_i(r_i - 1) - x'_i(r_i - 2), x_i(0) - f'_i\}$$

is the reduced Gröbner basis of J, *we denote by* g_i *the image of* f'_i *under the algebra isomorphism* $\bar{R}' \to \bar{R}$ *such that, for any* $i = 1, 2, \ldots, n$

$$x'_i(0) \mapsto x_i(r_i - 1), x'_i(1) \mapsto x_i(r_i - 2), \ldots, x'_i(r_i - 1) \mapsto x_i(0).$$

The inverse of an invertible system (1) *is by definition the following explicit difference system*

$$\left\{ \begin{array}{rcl} x_1(r_1) & = & g_1, \\ & \vdots & \\ x_n(r_n) & = & g_n. \end{array} \right. \tag{3}$$

Let $\bar{\mathrm{T}}, \bar{\mathrm{S}} : \bar{R} \to \bar{R}$ be the state transition endomorphisms of an invertible system (1) and its inverse system (3), respectively. Denote by $\xi : \bar{R} \to \bar{R}$ the algebra automorphism such that

$$x_i(0) \mapsto x_i(r_i - 1), x_i(1) \mapsto x_i(r_i - 2), \ldots, x_i(r_i - 1) \mapsto x_i(0).$$

By Theorem 3.2 and Corollary 3.3, we have that $\xi\bar{\mathrm{S}} = \bar{\mathrm{T}}^{-1}\xi$.

Proposition 3.7. *Let* (3) *be the inverse system of an invertible system* (1). *If* $(a_1, \ldots, a_n)$ *is a* $\mathbb{K}$*-solution of* (1), *consider its* t*-state* ($t \geq 0$)

$$v = (a_1(t), \ldots, a_1(t + r_1 - 1), \ldots, a_n(t), \ldots, a_n(t + r_n - 1)).$$

Denote by $(b_1, \ldots, b_n)$ *the* $\mathbb{K}$*-solution of* (3) *whose initial state is*

$$v' = (a_1(t + r_1 - 1), \ldots, a_1(t), \ldots, a_n(t + r_n - 1), \ldots, a_n(t)).$$

If the t*-state of* $(b_1, \ldots, b_n)$ *is*

$$u' = (b_1(t), \ldots, b_1(t + r_1 - 1), \ldots, b_n(t), \ldots, b_n(t + r_n - 1)),$$

then the initial state of $(a_1, \ldots, a_n)$ *is*

$$u = (b_1(t + r_1 - 1), \ldots, b_1(t), \ldots, b_n(t + r_n - 1), \ldots, b_n(t)).$$

Proof. Denote by $\mathrm{T}, \mathrm{S} : \mathbb{K}^r \to \mathbb{K}^r$ $(r = r_1 + \ldots + r_n)$ the state transition maps of the systems (1),(3), respectively. By definition, we have that $u' = \mathrm{S}^t(v')$. Since $\xi\bar{\mathrm{S}} = \bar{\mathrm{T}}^{-1}\xi$, we conclude that $u = \mathrm{T}^{-t}(v)$. □

Another useful notion is the following one.

Definition 3.8. *An explicit difference system* (1) *is called* reducible *if there is an integer* $0 < m < n$ *such that we have a subsystem*

$$\left\{ \begin{array}{lcl} x_1(r_1) & = & f_1, \\ & \vdots & \\ x_m(r_m) & = & f_m. \end{array} \right. \tag{4}$$

In other words, one has that $f_1, \ldots, f_m \in \bar{R}_m = \mathbb{K}[\bar{X}_m]$ *where by definition* $\bar{X}_m = \{x_1(0), \ldots, x_1(r_1 - 1), \ldots, x_m(0), \ldots, x_m(r_m - 1)\}$*. In this case, the state transition endomorphism and map of* (4) *are just the restrictions of the corresponding functions of* (1) *to the subring* $\bar{R}_m \subset \bar{R}$ *and the subspace* $\mathbb{K}^k \subset \mathbb{K}^r$ $(k = r_1 + \ldots + r_m, r = r_1 + \ldots + r_n)$*, respectively.*

The following result is obtained immediately.

Proposition 3.9. *Let* (1) *be a reducible invertible system. Then, its subsystem* (4) *is also invertible. Moreover, the inverse system of* (1) *is also reducible with a subsystem which is the inverse system of* (4).

4. Periodic systems

Definition 4.1. *For an invertible system* (1)*, consider the state transition map* $\mathrm{T} : \mathbb{K}^r \to \mathbb{K}^r$ $(r = r_1 + \ldots + r_n)$*. We call the system* periodic *if there is an integer* $d > 0$ *such that* $\mathrm{T}^d = \mathrm{id}$*. In this case, the period of the map* T *is called the* period of the system (1).

Proposition 4.2. *Consider a periodic system* (1) *with period* d*. If* $(a_1, \ldots, a_n)$ *is a* $\mathbb{K}$*-solution of* (1)*, then all functions* a_i *are periodic, that is,* $a_i(t) = a_i(t + d)$ *for all clocks* $t \geq 0$.

Proof. If $v \in \mathbb{K}^r$ is the initial state of $(a_1, \ldots, a_n)$, by the argument of Theorem 2.4 we have that $a_1(t)$ is the first coordinate of the vector $\mathrm{T}^t(v) \in \mathbb{K}^r$. Since $\mathrm{T}^t = \mathrm{T}^{t+d}$, one has that $\mathrm{T}^t(v) = \mathrm{T}^{t+d}(v)$ and therefore $a_1(t) = a_1(t + d)$. In a similar way, we also prove that $a_i(t) = a_i(t + d)$ $(1 < i \leq n)$. □

Note that if $\mathbb{K} = \mathrm{GF}(q)$ is a finite field, the symmetric group $\mathbb{S}(\mathbb{K}^r)$ has finite order and therefore all invertible systems are in fact periodic. We also observe that if $\mathbb{K}$ is an infinite field, then the state transition endomorphism $\bar{\mathrm{T}}$ is bijective if and only if the state transition map T is bijective. Moreover, we have that $\bar{\mathrm{T}}$ is periodic if and only if T is periodic and in this case these maps have the same period. Such facts are consequences of the following general result (see for instance [32, 34]).

Proposition 4.3. *Consider a polynomial algebra* $P = \mathbb{K}[x_1, \ldots, x_r]$ *and an algebra endomorphism* $\varphi : P \to P$ *such that* $x_1 \mapsto g_1, \ldots, x_r \mapsto g_r$ $(g_i \in P)$*. Denote by* $\hat{\varphi} : \mathbb{K}^r \to \mathbb{K}^r$ *the corresponding polynomial map, that is, for any* $(\alpha_1, \ldots, \alpha_r) \in \mathbb{K}^r$

$$(\alpha_1, \ldots, \alpha_r) \mapsto (g_1(\alpha_1, \ldots, \alpha_r), \ldots, g_r(\alpha_1, \ldots, \alpha_r)).$$

The map $\varphi \mapsto \hat{\varphi}$ *is a homomorphism from the monoid of algebra endomorphisms of* P *to the monoid of polynomial maps* $\mathbb{K}^r \to \mathbb{K}^r$*. If* $\mathbb{K}$ *is an infinite field, this monoid homomorphism is bijective. Otherwise, if* $\mathbb{K} = \mathrm{GF}(q)$ *then the map* $\varphi \mapsto \hat{\varphi}$ *induces a monoid isomorphism from the monoid of algebra endomorphisms of the quotient algebra* P/L*, where* $L = \langle x_1^q - x_1, \ldots, x_r^q - x_r \rangle \subset P$*. Note that* P *and* P/L *are the coordinate algebras of the affine space* $\mathbb{K}^r$ *for the case that* $\mathbb{K}$ *is an infinite or finite field, respectively.*

An important and difficult task is to compute, or at least bound, the period of a periodic explicit difference system. As usual, the task becomes easy in the linear case.

Definition 4.4. *An explicit difference system* (1) *is called* linear *if all polynomials* f_i $(1 \leq i \leq n)$ *are homogeneous linear ones. In other words, the state transition map* $\mathrm{T} : \mathbb{K}^r \to \mathbb{K}^r$ *is a* $\mathbb{K}$*-linear endomorphism of the vector space* $\mathbb{K}^r$*.*

Restating the Rational (or Frobenius) Canonical Form of a square matrix (see, for instance, [28]) in terms of $\mathbb{K}$-linear endomorphisms, one has the following result.

Proposition 4.5. *Let* $\psi : \mathbb{K}^r \to \mathbb{K}^r$ *be any* $\mathbb{K}$*-linear endomorphism. Then, there is a* $\mathbb{K}$*-linear automorphism* $\xi : \mathbb{K}^r \to \mathbb{K}^r$ *such that* $\psi' = \xi\psi\xi^{-1}$ *can be decomposed as a direct sum* $\psi' = \bigoplus_{1 \leq i \leq n} \psi'_i$ *where* $\psi'_i : \mathbb{K}^{r_i} \to \mathbb{K}^{r_i}$ $(r_1 + \ldots + r_n = r)$ *is a* $\mathbb{K}$*-linear endomorphism such that, for any* $(\alpha_0, \ldots, \alpha_{r_i - 1}) \in \mathbb{K}^{r_i}$

$$\psi'_i(\alpha_0, \ldots, \alpha_{r_i-2}, \alpha_{r_i-1}) = (\alpha_1, \ldots, \alpha_{r_i-1}, g_i(\alpha_0, \ldots, \alpha_{r_i-1}))$$

and g_i *is a homogeneous linear polynomial in* r_i *variables. It follows that if* ψ *is an automorphism of finite period* d*, then* $d = \mathrm{lcm}(d_1, \ldots, d_n)$ *where* d_i *is the period of* ψ'_i*.*

Note that the above result provides that, up to an invertible $\mathbb{K}$-linear change of variables, any linear difference system can be obtained in a canonical form, say (1), where f_i $(1 \leq i \leq n)$ is a linear form which is defined only over the set of variables $\{x_i(0), \ldots, x_i(r_i - 1)\}$. In other words, the system is the join of linear difference equations on disjoint sets of variables. In Cryptography (see, for instance, [17, 38]), a linear difference equation is called a *Linear Feedback Shift Register* or briefly *LFSR*.

For cryptographic applications, to have a periodic difference system with a large period d is a useful property. In the linear case, according to Proposition 4.5, to maximize d one needs that all d_i are coprime so that $d = d_1 \cdots d_n$. Then, the problem reduces to maximize the period of each single periodic linear difference equation. This problem has a well-known solution (see [17]) when $\mathbb{K} = \mathrm{GF}(p) = \mathbb{Z}_p$ with p a prime number. For the purpose of completeness, we provide this result.

Proposition 4.6. *Consider an invertible linear difference equation*

$$x(r) = \sum_{0\leq i\leq r-1} c_i x(i) \ (c_i \in \mathbb{Z}_p). \tag{5}$$

Denote $g = t^r - \sum_i c_i t^i \in \mathbb{Z}_p[t]$ and assume that g is an irreducible polynomial. Consider the finite field $\mathbb{F} = \mathrm{GF}(p^r) = \mathbb{Z}_p[t]/(g)$ and the corresponding multiplicative (cyclic) group $\mathbb{F}^* = \mathbb{F} \setminus \{0\}$. If the element $\alpha = t + (g) \in \mathbb{F}^*$ is a generator of $\mathbb{F}^*$, that is, g is a primitive polynomial *of $\mathbb{Z}_p[t]$, then the equation* (5) *has maximal period $p^r - 1$.*

Proof. The matrix corresponding to the state transition $\mathbb{K}$-linear map $\mathrm{T} : \mathbb{Z}_p^r \to \mathbb{Z}_p^r$ of (5) with respect to the canonical basis of $\mathbb{Z}_p^r$, is indeed the companion matrix A of the polynomial g, that is

$$A = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ c_0 & c_1 & c_2 & \dots & c_{r-1} \end{pmatrix}.$$

Consider the algebra $\mathrm{M}_r(\mathbb{Z}_p)$ of all square matrices of order r with entries in the field $\mathbb{Z}_p$ and denote by $\mathbb{Z}_p[A] \subset \mathrm{M}_r(\mathbb{Z}_p)$ the subalgebra that is generated by the matrix A. It is well-known that the minimal polynomial of the companion matrix A is exactly g and hence $\mathbb{F} = \mathbb{Z}_p[\alpha]$ is isomorphic to $\mathbb{Z}_p[A]$ by the map $\alpha \mapsto A$. Since α is a generator, that is, an element of maximal period in the cyclic group $\mathbb{F}^*$, we obtain that the period of α and A is exactly $p^r - 1$. □

5. Difference stream/block ciphers

From now on, let $\mathbb{K} = \mathrm{GF}(q)$ be a finite field. It is important to note that in this case, by Lagrange interpolation, any function $\mathbb{K}^r \to \mathbb{K}$ is in fact a polynomial one. Moreover, if $f \in \mathbb{K}[x_1, \dots, x_r]$ is the corresponding polynomial, we can assume that f is normal modulo the ideal $L = \langle x_1^q - x_1, \dots, x_r^q - x_r \rangle$, that is, all exponents in the monomials of f are strictly less than q.

Definition 5.1. *A* difference stream cipher *$\mathcal{C}$ is an explicit difference system* (1) *together with a polynomial $f \in \bar{R}$. If $(a_1, \dots, a_n)$ is a $\mathbb{K}$-solution of* (1)*, its initial state is called the* key *of $(a_1, \dots, a_n)$. Moreover, if $v(t) \in \mathbb{K}^r$ $(r = r_1 + \dots + r_n)$ is the t-state of $(a_1, \dots, a_n)$, the function $b : \mathbb{N} \to \mathbb{K}$ such that $b(t) = f(v(t))$ for all $t \geq 0$, is called the* keystream *of $(a_1, \dots, a_n)$. Finally, we call f the* keystream polynomial *of the cipher $\mathcal{C}$.*

If (1) is linear, that is, a system of LFSRs, the polynomial f is required nonlinear and it is usually called a *combining or filtering function.* Observe that a difference stream cipher can also be defined as a special explicit difference system

$$\left\{ \begin{array}{rcl} x_1(r_1) & = & f_1, \\ & \vdots & \\ x_n(r_n) & = & f_n, \\ y(0) & = & f. \end{array} \right.$$

In fact, by a $\mathbb{K}$-solution $(a_1, \dots, a_n, b)$ of such a system one obtains the keystream function $b : \mathbb{N} \to \mathbb{K}$ of the $\mathbb{K}$-solution $(a_1, \dots, a_n)$ of (1).

In Cryptography, a stream cipher (see, for instance, [29]) operates simply by adding and subtracting the keystream to a *stream of plaintexts or ciphertexts*. Such a stream is by definition a function $\mathbb{N} \to \mathbb{K}$. By a *known plaintext attack* we can assume the knowledge of the keystream as the difference between the known ciphertext and plaintext streams. Note that the keystream is usually provided by a stream cipher after a sufficiently high number of clocks in order to prevent cryptanalysis.

Definition 5.2. *Let $\mathcal{C}$ be a difference stream cipher consisting of the system* (1) *and the keystream polynomial f. Let $b : \mathbb{N} \to \mathbb{K}$ be the keystream of a $\mathbb{K}$-solution of* (1) *and fix an integer $T \geq 0$. Consider the ideal*

$$J = \sum_{t \geq T} \langle \sigma^t(f) - b(t) \rangle \subset R$$

and denote by $V_{\mathbb{K}}(J)$ the set of simultaneous $\mathbb{K}$-solutions of all polynomials in J, or equivalently, of its generators. An algebraic attack *to $\mathcal{C}$ by the keystream b* after T clocks *consists in computing the $\mathbb{K}$-solutions $(a_1, \ldots, a_n)$ of the system* (1) *such that $(a_1, \ldots, a_n) \in V_{\mathbb{K}}(J)$. In other words, if we consider the difference ideal corresponding to* (1)*, that is, $I = \langle x_1(r_1) - f_1, \ldots, x_n(r_n) - f_n \rangle_\sigma \subset R$ then one wants to compute $V_{\mathbb{K}}(I + J) = V_{\mathbb{K}}(I) \cap V_{\mathbb{K}}(J)$.*

Since the given function b is the keystream of a $\mathbb{K}$-solution of (1), say $(a_1, \ldots, a_n)$, we have that $(a_1, \ldots, a_n) \in V_{\mathbb{K}}(I + J) \neq \emptyset$. For actual ciphers, we generally have that $V_{\mathbb{K}}(I + J) = \{(a_1, \ldots, a_n)\}$.

Definition 5.3. *With the notation of Definition 5.2, denote by $\bar{V}_{\mathbb{K}}(I + J) \subset \mathbb{K}^r$ the set of keys, that is, initial states of the $\mathbb{K}$-solutions $(a_1, \ldots, a_n) \in V_{\mathbb{K}}(I + J)$. By Theorem 2.4, there is a bijective map $V_{\mathbb{K}}(I + J) \to \bar{V}_{\mathbb{K}}(I + J)$ and we have that $\bar{V}_{\mathbb{K}}(I) = \mathbb{K}^r$.*

Theorem 5.4. *Let $\bar{\mathrm{T}} : \bar{R} \to \bar{R}$ be the state transition endomorphism of the system* (1) *and put $f'_t = \bar{\mathrm{T}}^t(f) \in \bar{R}$, for all $t \geq 0$. Moreover, define the ideal*

$$J' = \sum_{t \geq T} \langle f'_t - b(t) \rangle \subset \bar{R}.$$

Then, we have that $\bar{V}_{\mathbb{K}}(I + J) = V_{\mathbb{K}}(J')$.

Proof. Let $(a_1, \ldots, a_n)$ be a $\mathbb{K}$-solution of (1) and denote by $v(t)$ its t-state ($t \geq 0$). By the identity (2), one obtains that

$$f'_t(v(0)) = \bar{\mathrm{T}}^t(f)(v(0)) = f(\mathrm{T}^t(v(0))) = f(v(t)).$$

We conclude that the condition $f(v(t)) = b(t)$ ($t \geq T$) is equivalent to the condition $f'_t(v(0)) = b(t)$. □

By assuming that $x_i(r_i) \succ \mathrm{lm}(f_i)$ for all $1 \leq i \leq n$, that is, $G = \{x_1(r_1) - f_1, \ldots, x_n(r_n) - f_n\}$ is a difference Gröbner basis of the difference ideal $I = \langle G \rangle_\sigma$, by Theorem 2.13 one obtains that

$$f'_t = \bar{\mathrm{T}}^t(f) = \mathrm{NF}_I(\sigma^t(f)).$$

In actual algebraic attacks, we are given a finite number of values of the keystream b, that is, for a fixed integer bound $B \geq T$, we consider the finitely generated ideal

$$J'_B = \sum_{T \leq t \leq B} \langle f'_t - b(t) \rangle \subset \bar{R}.$$

We have that $J' = \bigcup_{B \geq T} J'_B$ where $J'_B \subset J'_{B+1}$. Since the polynomial algebra $\bar{R}$ is finitely generated and hence Noetherian, one has that $J'_B = J'$ for some $B \geq T$. In other words, we don't lose any equation satisfied by the keys if a sufficiently large number of keystream values is provided for the attack.

To compute the set $V_{\mathbb{K}}(J'_B) \subset \mathbb{K}^r$ one can use essentially Gröbner bases or SAT solvers when $\mathbb{K} = \mathrm{GF}(2)$ (see, for instance, [3]). For real ciphers, one generally has that $V_{\mathbb{K}}(J'_B) = V_{\mathbb{K}}(J')$ contains a single $\mathbb{K}$-solution, that is, a single key. The Nullstellensatz over finite fields (see, for instance [24]) implies the following result.

Proposition 5.5. *Let* $\mathbb{K} = \mathrm{GF}(q)$ *be a finite field and consider the polynomial algebra* $P = \mathbb{K}[x_1, \ldots, x_r]$ *and the ideal* $L = \langle x_1^q - x_1, \ldots, x_r^q - x_r \rangle \subset P$. *Moreover, let* $J \subset P$ *be any ideal and denote by* $V(J)$ *the set of* $\bar{\mathbb{K}}$*-solutions of all polynomials* $f \in J$ *where the field* $\bar{\mathbb{K}}$ *is the algebraic closure of* $\mathbb{K}$. *We have that* $V(L) = \mathbb{K}^r$ *and* $V_{\mathbb{K}}(J) = V(J) \cap \mathbb{K}^r = V(J + L)$ *where* $J + L \subset P$ *is a radical ideal. Moreover, if* $V_{\mathbb{K}}(J) = \{(\alpha_1, \ldots, \alpha_r)\}$ *then* $G = \{x_1 - \alpha_1, \ldots, x_r - \alpha_r\}$ *is the (reduced) universal Gröbner basis of* $J + L$, *that is, its Gröbner basis with respect to all monomial orderings of* P.

The above result is very useful for the algebraic attacks because Gröbner bases computations are very sensitive to the monomial orderings and we are free here to choose the most efficient orderings such as DegRevLex. Another possible optimization when performing the Buchberger algorithm on the ideal $J + L \subset P$ consists in skipping all remaining S-polynomials once each variable x_i $(1 \leq i \leq r)$ has been obtained as the leading monomial of an element in the current Gröbner basis. In some cases, this trick speeds up the computation in a significant way.

Let $\mathcal{C}$ be a difference stream cipher which is given by the system (1) and the keystream polynomial f. The polynomial $f'_t = \bar{\mathrm{T}}^t(f) \in \bar{R}$ generally has a high degree if T is large with respect to 0. This is usually the case in actual ciphers where a high number of clocks is required before the keystream appears. Nevertheless, if the system (1) is invertible we can always assume that $T = 0$. In fact, by means of the notion of inverse system in Definition 3.6, to compute the T-state is completely equivalent to compute the initial state, that is, the key of a $\mathbb{K}$-solution of (1). This is a very effective optimization because it drastically reduces the degrees of the generators of the ideal $J'_B = \sum_{T \leq t \leq B} \langle f'_t - b(t) \rangle$ to those of the generators of the ideal $J''_B = \sum_{0 \leq t \leq B-T} \langle f'_t - b(T + t) \rangle$. Recall that we have to compute a Gröbner basis for obtaining $\mathbb{K}$-solutions and such computations are very sensitive to the degree of the generators. We apply this strategy when attacking the stream cipher Bivium in Section 6. If the polynomials f'_t have still high degrees and they are even difficult to compute, an alternative strategy consists in computing directly the $\mathbb{K}$-solutions of the system (1) which are also solutions of the fixed degree polynomials $\sigma^t(f) - b(T + t)$ $(0 \leq t \leq B - T)$. Even though the clocks of the variables in X can be bounded, this strategy has the main drawback that one has to compute a Gröbner basis over a generally high number of variables.

We have just observed that difference stream ciphers that are defined by invertible systems show some lack of security with respect to algebraic attacks. On the other hand, invertible systems can be used to define block ciphers.

Definition 5.6. *A* difference block cipher $\mathcal{C}$ *is a reducible invertible system* (1) *together with an integer* $T \geq 0$. *If* (4) *is the subsystem of* (1), *we put* $k = r_1 + \ldots + r_m$ *and* $l = r_{m+1} + \ldots + r_n$. *If a t-state of a* $\mathbb{K}$*-solution* $(a_1, \ldots, a_n)$ *of* (1) *is denoted* *as the pair* $(u(t), v(t)) \in \mathbb{K}^k \times \mathbb{K}^l = \mathbb{K}^r$*, we call* $u(0)$ *the* key*,* $v(0)$ *the* plaintext *and* $v(T)$ *the* ciphertext of $(a_1, \ldots, a_n)$*. Moreover, we call* $(u(T), v(T))$ *the* final state of $(a_1, \ldots, a_n)$ *and* (4) *the* key subsystem of the cipher $\mathcal{C}$*.*

With the language of Cryptography, the *encryption function* $E_{u(0)} : \mathbb{K}^l \to \mathbb{K}^l$ of the difference block cipher $\mathcal{C}$ is given by the map $v(0) \mapsto v(T)$, where the pair $(u(0), v(0))$ varies in the affine space $\mathbb{K}^k \times \mathbb{K}^l$ of all initial states of the $\mathbb{K}$-solutions of (1). To provide the decryption function we introduce the following notion.

Definition 5.7. *Let* $\mathcal{C}$ *be a difference block cipher consisting of a reducible invertible system* (1) *and a clock* $T \geq 0$*. The* inverse cipher of $\mathcal{C}$ *is by definition the inverse system of* (1) *together with* T*.*

Let $\mathcal{C}'$ be the inverse cipher of $\mathcal{C}$ where (3) is the inverse system of (1). Consider also the key subsystem (4) of $\mathcal{C}$. If $u(0)$ is the key of a solution of $(a_1, \ldots, a_n)$ of (1), we can compute $u(T)$ by means of (4) without knowing $v(0)$. If we are given the ciphertext $v(T)$, we have hence the final state $(u(T), v(T))$ of $(a_1, \ldots, a_n)$. By Proposition 3.7, the inverse system (3) provides the computation of the initial state $(u(0), v(0))$ of $(a_1, \ldots, a_n)$ and in particular of the plaintext $v(0)$. In other words, the decryption function $D_{u(0)} : \mathbb{K}^l \to \mathbb{K}^l$ is obtained as the map $v(T) \mapsto v(0)$ which is computable by means of the systems (3), (4).

Definition 5.8. *Let* $\mathcal{C}$ *be a difference block cipher given by a reducible invertible system* (1) *and a clock* $T \geq 0$*. For all* $t \geq 0$*, let* $(u(t), v(t)) \in \mathbb{K}^k \times \mathbb{K}^l$ *be the* t*-state of a* $\mathbb{K}$*-solution of* (1) *where we denote*

$$v(t) = (a_{m+1}(t), \ldots, a_{m+1}(t + r_{m+1} - 1), \ldots, a_n(t), \ldots, a_n(t + r_n - 1)).$$

Consider the corresponding linear ideal

$$J(t) = \sum_{m+1 \leq i \leq n} \langle x_i(t) - a_i(t), \ldots, x_i(t + r_i - 1) - a_i(t + r_i - 1) \rangle \subset R$$

and put $J = J(0) + J(T)$*. An* algebraic attack to $\mathcal{C}$ by the plaintext-ciphertext pair $(v(0), v(T))$ *consists in computing the* $\mathbb{K}$*-solutions* $(a_1, \ldots, a_n)$ *of the system* (1) *such that* $(a_1, \ldots, a_n) \in V_{\mathbb{K}}(J)$*. If* $I = \langle x_1(r_1) - f_1, \ldots, x_n(r_n) - f_n \rangle_\sigma \subset R$*, this is equivalent to compute* $V_{\mathbb{K}}(I + J) = V_{\mathbb{K}}(I) \cap V_{\mathbb{K}}(J)$*.*

Note that the above attack belongs to the class of known plaintext attacks. Since the given pair $(v(0), v(T))$ is obtained by the states $(u(t), v(t))$ of a $\mathbb{K}$-solution of (1), say $(a_1, \ldots, a_n)$, we have that $(a_1, \ldots, a_n) \in V_{\mathbb{K}}(I + J) \neq \emptyset$. For actual ciphers, one generally has that the set $V_{\mathbb{K}}(I+J)$ contains more than one $\mathbb{K}$-solution. Since computing a unique solution by a single DegRevLex-Gröbner basis can be faster than calculating multiple solutions via conversion to an elimination ordering (FGLM algorithm [23]), we prefer to obtain uniqueness by attacking with multiple plaintext-ciphertext pairs.

Precisely, fix an integer $s > 1$ and let $(u(t), v^{(i)}(t))$ $(1 \leq i \leq s)$ be the t-state of a $\mathbb{K}$-solution $(a_1, \ldots, a_m, a^{(i)}_{m+1}, \ldots, a^{(i)}_n)$ of the system (1) where $(a_1, \ldots, a_m)$ is some fixed $\mathbb{K}$-solution of the key subsystem (4). In other words, we consider some plaintext-ciphertext pairs $(v^{(i)}(0), v^{(i)}(T))$ $(1 \leq i \leq s)$ which are obtained by the same key $u(0)$. To describe properly a multiple pairs attack, we also need the following notations.

For each $i = 1, 2, \ldots, s$ and $t \geq 0$, consider the set of variables $X^{(i)}(t) = \{x_1(t), \ldots, x_m(t), x^{(i)}_{m+1}(t), \ldots, x^{(i)}_n(t)\}$ where $\bigcap_i X^{(i)}(t) = \{x_1(t), \ldots, x_m(t)\}$. We put $X^{(i)} = \bigcup_{t\geq 0} X^{(i)}(t)$ and $R^{(i)} = \mathbb{K}[X^{(i)}]$. The polynomial algebra $R^{(i)}$ is clearly isomorphic to R and we denote by $I^{(i)} \subset R^{(i)}$ the ideal which is isomorphic to $I \subset R$. For all $1 \leq i \leq s$, consider (1) as written in the variables $X^{(i)}$ and let $J^{(i)}(t) \subset R^{(i)}$ $(t \geq 0)$ be the linear ideal corresponding to the t-state of the $\mathbb{K}$-solution $(a_1, \ldots, a_m, a^{(i)}_{m+1}, \ldots, a^{(i)}_n)$ of the system (1). Moreover, we put $X' = \bigcup_i X^{(i)}, R' = \mathbb{K}[X']$ and we denote by $I', J'(t)$ the ideals of R' that are generated by $\sum_i I^{(i)}, \sum_i J^{(i)}(t)$, respectively. Finally, we put $J' = J'(0) + J'(T)$.

Definition 5.9. *Let $\mathcal{C}$ be a difference block cipher given by a reducible invertible system* (1) *and a clock $T \geq 0$. An* algebraic attack to $\mathcal{C}$ by the multiple plaintext-ciphertext pairs $(v^{(i)}(0), v^{(i)}(T))$ $(1 \leq i \leq s)$ *consists in computing the $\mathbb{K}$-solutions*

$$(a_1, \ldots, a_m, a^{(1)}_{m+1}, \ldots, a^{(s)}_{m+1}, \ldots, a^{(1)}_n, \ldots, a^{(s)}_n) \in V_{\mathbb{K}}(I' + J').$$

For real ciphers, a sufficiently large number of pairs implies that we have $V_{\mathbb{K}}(I' + J') = \{(a_1, \ldots, a_m, a^{(1)}_{m+1}, \ldots, a^{(s)}_{m+1}, \ldots, a^{(1)}_n, \ldots, a^{(s)}_n)\}$. By bounding the clocks of the variables in X', one can compute this unique $\mathbb{K}$-solution and hence its key $u(0) = (a_1(0), \ldots, a_1(r_1 - 1), \ldots, a_m(0), \ldots, a_m(r_m - 1))$ using a Gröbner basis computation as in Proposition 5.5. Alternatively, for $\mathbb{K} = \mathrm{GF}(2)$ one can use SAT solvers or other methods. In Section 7 we make use of multiple pairs when attacking the block cipher KeeLoq.

Since the final clock T is usually chosen a large one, the main drawback of this approach is the high number of variables. Despite this, such a strategy generally appears to be more viable than elimination techniques. Indeed, the normal forms of the generators of $J'(T)$ modulo $I' + J'(0)$ belong to $\bar{R}_m$ but they may have very high degrees because of the large clock T. As for stream ciphers, the main problem is hence to reduce somehow the final clock T. Even though the system (1) of the block cipher $\mathcal{C}$ is invertible, note that we cannot attack an internal state instead of the initial one because the set $V_{\mathbb{K}}(I' + J'(t))$ $(t \leq T)$ generally contains too many solutions. In other words, a ciphertext only attack is generally too weak for difference block ciphers.

A better strategy is possible when the period of the key subsystem (4), say d, is sufficiently small. This technique has been introduced in [9, 10] to attack KeeLoq. To simplify its description, let us assume that the final state T is a multiple of d. Consider the state transition map $\mathrm{T} : \mathbb{K}^r \to \mathbb{K}^r$ of the explicit difference system (1) and denote by $\mathrm{S} : \mathbb{K}^k \to \mathbb{K}^k$ the state transition map of the subsystem (4). Recall that S is just the restriction of the map T to the subspace $\mathbb{K}^k \subset \mathbb{K}^r$ $(k = r_1 + \ldots + r_m, r = r_1 + \ldots + r_n)$. By definition of period, one has that $\mathrm{S}^d = \mathrm{id}$ and therefore $\mathrm{S}^d(u) = u$, for all $u \in \mathbb{K}^k$. For any $t \geq 0$, denote by $(u(t), v(t)) \in \mathbb{K}^k \times \mathbb{K}^l = \mathbb{K}^r$ the t-state of a $\mathbb{K}$-solution $(a_1, \ldots, a_n)$ of the system (1). Then, the encryption function $E_{u(0)} : \mathbb{K}^l \to \mathbb{K}^l$ corresponding to the key $u(0) \in \mathbb{K}^k$ is the map $v(0) \mapsto v(T)$. By a chosen plaintext attack, we can assume the knowledge of the bijection $E_{u(0)}$. If $\mathbb{K}^l$ is a large space $(l \to \infty)$, one has a probability equal to $1 - 1/e \approx 0.63$ (see [9], Section 4.1) that $E_{u(0)}$ has one or more fixed points $v(0) = v(T)$. Observe now that $v(0) = v(d)$ implies that $v(0) = v(T)$. In fact, by definition $(u(d), v(d)) = \mathrm{T}^d(u(0), v(0))$ and we have that $u(d) = \mathrm{S}^d(u(0)) = u(0)$. Then, from $v(0) = v(d)$ it follows that $(u(0), v(0)) = \mathrm{T}^d(u(0), v(0))$ and hence $(u(0), v(0)) = \mathrm{T}^T(u(0), v(0))$ because T is a multiple of d. We conclude that among the fixed points of the encryption function $E_{u(0)}$ one has the fixed points of the map $v(0) \mapsto v(d)$. If $v(0) = v(d)$ is such a fixed point, we have that $(v(0), v(0)) = (v(0), v(d)) = (v(0), v(T))$, that is, $(v(0), v(0))$ is a plaintext-ciphertext pair for the final clocks d and T. In other words, by means of such pairs we can perform an algebraic attack to the difference block cipher $\mathcal{C}$ assuming that the final clock is just the period of the key subsystem.

Let us conclude this section with a final general observation. When we apply Proposition 5.5 for solving polynomial systems, an essential trick consists in adding some linear polynomials to the considered ideal $J + L \subset \mathbb{K}[x_1, \ldots, x_r]$ ($L = \langle x_1^q - x_1, \ldots, x_r^q - x_r \rangle$) in order to speed up the Gröbner basis computation. Such linear polynomials are either elements of J which are given or computed ones, or they correspond to the evaluations of some subset of variables $\{x_{i_1}, \ldots, x_{i_s}\} \subset \{x_1, \ldots, x_r\}$. If some of these evaluations, say $x_{i_k} = \alpha_{i_k}$ ($\alpha_{i_k} \in \mathbb{K}$), is wrong and $V_{\mathbb{K}}(J)$ contains a unique solution, one has that

$$J + L + \langle x_{i_1} - \alpha_{i_1}, \ldots, x_{i_s} - \alpha_{i_s} \rangle = \langle 1 \rangle$$

and the Gröbner basis computation stops as soon as the element 1 is obtained. Note that using instead a SAT solver, the answer "UNSAT" essentially arrives when the full space $\mathbb{K}^r$ ($\mathbb{K} = \mathrm{GF}(2)$) has been examined. This means that for wrong evaluations, which are all except that one, Gröbner basis solving is generally faster than SAT solving. We have evidence of this in practice in Section 6.

Note that solving after the evaluation of some bunch of variables is usually called a *guess-and-determine strategy* (see, for instance, [20]) or a *hybrid strategy* (see [5]) in the case of (semi-)regular polynomial systems. The latter case cannot be generally assumed for algebraic attacks to difference ciphers because the polynomials we obtain by means of elimination techniques such as Theorem 5.4 do not seem random at all. For this reason, as in the paper [20], we prefer to consider the experimental running time of a guess-and-determine strategy as the product $a \cdot q^s$ where a is the average solving time for a single guess and q^s is the number of guesses of s variables when $\mathbb{K} = \mathrm{GF}(q)$. In other words, the complexity of such a strategy is $\mathcal{O}(q^s)$ where q^s is the total number of solving processes to be performed in a reasonable time.

Of course, by assuming with probability $\geq 1/2$ that the correct guess is found in half of the space $\mathbb{K}^q$, we obtain that the average running time is reduced to $a \cdot q^s/2$. We conclude by observing that the choice of the variables to be evaluated is a very important issue to optimize a guess-and-determine strategy. The parallelization of the computation which can be obtained simply by dividing the guess space in different subsets is also a viable way to scale down the complexity.

6. Attacking Bivium

Aiming to illustrate by a concrete example how to perform in practice an algebraic attack to a difference stream cipher as described in the previous section, we start considering Trivium which is a well-known stream cipher designed in 2003 by De Cannière and Preneel as a submission to European project eSTREAM [14]. In fact, Trivium was one of the winners of the project for the category of hardware-oriented ciphers. Even though it has been widely cryptanalysed, no critical attacks are known up to date. The system of explicit difference equations describing Trivium looks quite simple since it consists only of three quadratic equations over the base field $\mathbb{K} = \mathrm{GF}(2)$, namely

$$
(6) \qquad \begin{cases} x(93) &= z(0) + x(24) + z(45) + z(1)z(2), \\ y(84) &= x(0) + y(6) + x(27) + x(1)x(2), \\ z(111) &= y(0) + y(15) + z(24) + y(1)y(2). \end{cases}
$$

Its keystream polynomial is a homogeneous linear one

$$
f = x(0) + x(27) + y(0) + y(15) + z(0) + z(45).
$$

Therefore, a t-state consists of $288 = 93 + 84 + 111$ bits, for any clock $t \geq 0$. The keystream bits are known by the attackers starting with clock $T = 4 \cdot 288 = 1152$. The key and the initial vector of Trivium are 80 bit vectors and they form together 160 bits of an initial state. The remaining 128 bits are fixed ones.

By Corollary 3.4, we obtain that the system (6) is invertible with inverse system

$$
\begin{cases} x(93) &= y(0) + x(66) + y(78) + x(91)x(92), \\ y(84) &= z(0) + y(69) + z(87) + y(82)y(83), \\ z(111) &= x(0) + z(66) + x(69) + z(109)z(110). \end{cases}
$$

This allows an algebraic attack to the T-state instead of the initial state containing the key and the initial vector. The problem with such an attack is the high number (288) of variables in the set

$$
\bar{X} = \{x(0), \ldots, x(92), y(0), \ldots, y(83), z(0), \ldots, z(100)\}.
$$

Indeed, to solve the polynomial system obtained by means of Theorem 5.4 using a guess-and-determine strategy, one has to evaluate a number of variables that exceeds the length of the key which is 80 bit. In other words, providing that all solving computations for any guess can actually be performed in a reasonable time, one has a complexity which is greater than the key recovery by exhaustive search. An experimental evidence of this is contained, for instance, in [20, 26]. We also tried ourselves with a time limit of one hour for the Gröbner bases computations. We plan to further improve our guess-and-determine strategies to tackle Trivium.

Therefore, we present here all optimizations and computational data that we have obtained for a well-studied simplified version of Trivium cipher which is called Bivium. For this cipher we obtain a running time for an algebraic attack which improves a previous one [20] and it is much better than brute force.

The explicit difference system defining Bivium are the following two quadratic equations

$$
(7) \qquad \begin{cases} x(93) &= y(0) + y(15) + x(24) + y(1)y(2), \\ y(84) &= x(0) + y(6) + x(27) + x(1)x(2), \end{cases}
$$

and its keystream polynomial is

$$
f = x(0) + x(27) + y(0) + y(15).
$$

In this case, the t-states are vectors of $93 + 84 = 177$ bits and the keystream starts at clock $T = 4 \cdot 177 = 708$. Again, the key and the initial vector are $80 + 80 = 160$ bits of the initial state. Corollary 3.4 implies that the system (7) is invertible with inverse

$$
\begin{cases} x(93) &= y(0) + x(66) + y(78) + x(91)x(92), \\ y(84) &= x(0) + x(69) + y(69) + y(82)y(83). \end{cases}
$$

Consider the polynomial algebras $R = \mathbb{K}[X]$ where $X = \bigcup_{t\geq 0}\{x(t), y(t)\}$ and $\bar{R} = \mathbb{K}[\bar{X}]$ where $\bar{X} = \{x(0), \ldots, x(92), y(0), \ldots, y(83)\}$. If R is endowed with a clock-based monomial ordering, we have that

$$G = \{x(93) + y(0) + y(15) + x(24) + y(1)y(2),\\ y(84) + x(0) + y(6) + x(27) + x(1)x(2)\}$$

is a difference Gröbner basis of the difference ideal $I = \langle G\rangle_\sigma \subset R$. Consider the polynomials $f'_t = \bar{\mathrm{T}}^t(f) = \mathrm{NF}_I(\sigma^t(f)) \in \bar{R}$ ($t \geq 0$) and let $b : \mathbb{N} \to \mathbb{K}$ be a keystream. Since the system (7) is invertible, we can attack the T-state by the ideal

$$J''_B = \sum_{0\leq t\leq B-T} \langle f'_t + b(T+t)\rangle \subset \bar{R}.$$

We have found experimentally that if the number $B - T + 1$ of known values of the keystream is approximately 180, one has a unique $\mathbb{K}$-solution in $V_\mathbb{K}(J''_B)$. In this case, the maximal degree of the generators of J''_B is 3. Even though we use a DegRevLex monomial ordering on $\bar{R}$, a Gröbner basis of the ideal $J''_B + L \subset \bar{R}$ where

$$L = \langle x(0)^2 + x(0), \ldots, x(92)^2 + x(92), y(0)^2 + y(0), \ldots, y(83)^2 + y(83)\rangle$$

seems to be hard to compute. Indeed, we have experimented that the number of S-polynomials for such a computation increases in a very fast way affecting time and space complexity. We therefore use a guess-and-determine strategy to obtain solving times that we can actually determine.

Since all shifts $\sigma^t(f)$ ($0 \leq t \leq 65$) of the keystream polynomial $f = x(0)+x(27)+y(0)+y(15)$ are normal modulo I, we have 66 linear polynomials $\sigma^t(f) - b(T+t) = f'_t - b(T+t) \in J''_B$. Because the clocks of the variables in f are all multiples of 3, we can divide these polynomials into 3 sets of 22 linear polynomials, namely

$$S_i = \{\sigma^t(f) - b(T+t) \mid 0 \leq t \leq 65, t \equiv i \bmod 3\}\ (0 \leq i \leq 2).$$

By performing Gaussian elimination over S_i, we obtain 22 pivot variables and 36 free variables. In other words, for any set S_i the evaluation of 36 variables implies the evaluation of $36+22 = 58$ variables. This is a good trick that was first observed in [33]. In our computations, we choose the set S_2, that is, we guess the following 36 free variables

$$x(68), x(71), \ldots, x(92), y(2), y(5), \ldots, y(80)$$

and we obtain the evaluation of the 22 pivot variables

$$x(2), x(5), \ldots, x(65).$$

Moreover, note that one has the polynomial $f'_{68} - b(T+t) \in J''_B$ where

$$f'_{68} = y(83) + x(68) + y(68) + x(26) + y(17) + y(4)y(3) + y(2).$$

By guessing the variables $y(3), y(4)$ together with the previous 36 ones, one obtains in fact the evaluation of the variable $y(83)$, that is, a total of 61 evaluations out of the 177 variables of the algebra $\bar{R}$. This is enough to have Gröbner bases computations that last only a few tenths of seconds. Precisely, our guess-and-determine strategy for Bivium consists in computing the Gröbner bases of all ideals

$J''_B + L + E_{\alpha_1,\ldots,\alpha_{38}} \subset \bar{R}$, where

$$E_{\alpha_1,\ldots,\alpha_{38}} = \langle x(68) + \alpha_1, x(71) + \alpha_2, \ldots, x(92) + \alpha_9, y(2) + \alpha_{10}, \\ y(5) + \alpha_{11}, \ldots, y(80) + \alpha_{36}, y(3) + \alpha_{37}, y(4) + \alpha_{38}\rangle \tag{8}$$

and the vector $(\alpha_1, \ldots, \alpha_{38})$ ranges in the space $\mathbb{K}^{38}$.

We propose now tables where we compare the solving time to obtain the set $V_{\mathbb{K}}(J''_B + E_{\alpha_1,\ldots,\alpha_{38}})$ by using Gröbner bases and SAT solvers. For Gröbner bases, we make use of two main implementations of the Buchberger algorithm that are available in the computer algebra system Singular [15], namely std and slimgb. We have decided to use this free and open-source system because of our long experience with it, to compare with a previous attack [20] based on the same Gröbner bases implementations and finally because different implementations would affect the average running time $a \cdot 2^{37}$ only by the factor a corresponding to average solving time. The considered SAT solvers are minisat [19] and cryptominisat [37] which are widely used in cryptanalysis.

We have carried out the computations on a server: Intel(R) Core(TM) $i7-8700$ CPU @ 3.20GHz, 6 Cores, 12 Threads, 32 GB RAM with a Debian based Linux operating system. In our tables, we abbreviate milliseconds and seconds by ms and s, respectively.

Table 1. 90% Confidence Interval for timings with random guesses

# ks bits	slimgb (ms)	std (ms)	MiniSat (s)	CrMiniSat (s)
180	(160, 195)	(326, 397)	(9.33, 84.03)	(9.61, 40.39)
185	(159, 170)	(332, 367)	(6.88, 60.24)	(7.33, 28.41)
190	(119, 134)	(353, 411)	(6.94, 63.98)	(9.59, 38.26)
195	(123, 138)	(342, 397)	(6.55, 67.57)	(7.69, 25.26)

Table 2. 90% Confidence Interval for timings with correct guess

# ks bits	slimgb (ms)	std (ms)	MiniSat (s)	CrMiniSat (s)
180	(172, 187)	(330, 350)	(1.21, 53.81)	(2.71, 33.32)
185	(178, 191)	(352, 388)	(0.15, 49.59)	(0.24, 15.85)
190	(127, 145)	(351, 411)	(0.81, 24.56)	(0.23, 31.52)
195	(122, 135)	(328, 348)	(1.08, 36.50)	(6.72, 19.92)

In both tables, the rows correspond to different choices of the number of keystream bits that are used for the attack. The second and third columns present the 90% confidence intervals for Gröbner bases timings that are obtained by slimgb and std. The fourth and fifth columns provide the intervals for SAT solvers timings corresponding to minisat and cryptominisat.

In Table 1, the confidence intervals for Gröbner bases are computed for 2^4 different random (key, iv)-pairs and 2^{10} different random guesses of the 38 variables in (8) for each (key, iv)-pair. In other words, the confidence intervals contain 90% of the timings that are obtained by a total of 2^{14} computations. The intervals for SAT solvers are computed for the same set of 2^4 (key, iv)-pairs and with a subset of 2^4 different random guesses from the set that we have considered for Gröbner bases. The motivation of such reduction is larger total computing times for SAT solving.

Similarly, in Table 2 the confidence intervals are computed for the same 2^4 (key, iv)-pairs of Table 1 and the correct guess of the 38 variables corresponding to each (key, iv)-pair.

For Bivium attack, the tables show that the procedure slimgb is faster than std. This happens because "slim", that is, compact elements in the resulting Gröbner bases imply faster S-polynomial reductions which are the most expensive component of these computations. Moreover, we have that Gröbner bases perform better than SAT solvers for computing solutions of the polynomial systems involved in the Bivium attack. This is especially true for the UNSAT case which is dominant in complexity. About 190 keystream bits are the best choice for our attack and we conclude that its average running time is $0.12 \cdot 2^{37}\, s \sim 2^{34}\, s$.

This result improves the timing $2^{39}\, s$ of an algebraic attack to Bivium which is presented in [20]. This attack uses a guess-and-determine strategy based on the exhaustive evaluation of 42 variables and Gröbner bases computations which are obtained by the same Singular routines std and slimgb running on a comparable CPU.

7. Attacking KeeLoq

We present now an illustrative example of an algebraic attack to a concrete difference block cipher. A well-known small size block cipher is KeeLoq which has important applications in remote keyless entry systems which are used, for instance, by the automotive industry. KeeLoq is a proprietary cipher [6] whose cryptographic algorithm was created by Gideon Kuhn at the University of Pretoria in the mid-1980s. Starting from the mid-1990s, the cipher was widely used by car manufactures but it has begun to be cryptanalysed only in 2007. In particular, we mention the papers [9, 10] where there are algebraic attacks to KeeLoq on which this section is based. For another important class of "meet-in-the-middle attacks", see [27].

The block cipher KeeLoq is defined by a reducible invertible difference system over the base field $\mathbb{K} = \mathrm{GF}(2)$, where the key subsystem consists of a single homogeneous linear equation (LFSR). In fact, its state transition $\mathbb{K}$-linear map corresponds to a cyclic permutation matrix of period 64. In addition to the key equation, the invertible system consists of an explicit difference cubic equation involving a single key variable. Precisely, the KeeLoq system is the following one

$$
(9) \qquad \left\{
\begin{array}{rcl}
k(64) & = & k(0), \\
x(32) & = & x(0) + x(16) + x(9) + x(1) + x(20)x(31) \\
 & & + \, x(1)x(31) + x(20)x(26) + x(1)x(26) + x(9)x(20) \\
 & & + \, x(1)x(9) + x(1)x(9)x(31) + x(1)x(20)x(31) \\
 & & + \, x(9)x(26)x(31) + x(20)x(26)x(31) + k(0).
\end{array}
\right.
$$

The key, that is, the initial state of the key equation, consists therefore of 64 bits and the plaintext and ciphertext are 32 bits vectors. In other words, any t-state of KeeLoq consists of $64 + 32 = 96$ bits. The final clock of this difference block cipher is defined as the clock $T = 8 \cdot 64 + 16 = 528$. Theorem 3.2 provides that the system (9) is invertible with the following inverse system

$$
(10) \qquad \begin{cases} k(64) & = & k(0), \\ x(32) & = & x(0) + x(31) + x(23) + x(16) + x(23)x(31) \\ & & + x(6)x(31) + x(1)x(31) + x(12)x(23) + x(6)x(12) \\ & & + x(1)x(12) + x(1)x(23)x(31) + x(1)x(12)x(31) \\ & & + x(1)x(6)x(23) + x(1)x(6)x(12) + k(0). \end{cases}
$$

To describe a multiple plaintext-ciphertext pairs attack to KeeLoq, consider two such pairs $(v', v''), (w', w'') \in \mathbb{K}^{32} \times \mathbb{K}^{32}$, where $v' = (\alpha'_0, \ldots, \alpha'_{31}), v'' = (\alpha''_0, \ldots, \alpha''_{31})$ and $w' = (\beta'_0, \ldots, \beta'_{31}), w'' = (\beta''_0, \ldots, \beta''_{31})$. Then, define the polynomial algebra $R' = \mathbb{K}[X']$ where $X' = \bigcup_{t \geq 0} \{k(t), x(t), y(t)\}$ and consider the following set

$$
\begin{gathered}
G' = \{k(64) + k(0), \\
x(32) + x(0) + x(31) + x(23) + x(16) + x(23)x(31) + x(6)x(31) \\
+ x(1)x(31) + x(12)x(23) + x(6)x(12) + x(1)x(12) + x(1)x(23)x(31) \\
+ x(1)x(12)x(31) + x(1)x(6)x(23) + x(1)x(6)x(12) + k(0), \\
y(32) + y(0) + y(31) + y(23) + y(16) + y(23)y(31) + y(6)y(31) \\
+ y(1)y(31) + y(12)y(23) + y(6)y(12) + y(1)y(12) + y(1)y(23)y(31) \\
+ y(1)y(12)y(31) + y(1)y(6)y(23) + y(1)y(6)y(12) + k(0)\}.
\end{gathered}
$$

We define the difference ideal $I' = \langle G' \rangle_\sigma \subset R'$ and the linear ideal $J' = J'(0) + J'(T) \subset R'$ where

$$
\begin{gathered}
J'(0) = \langle x(0) + \alpha'_0, \ldots, x(31) + \alpha'_{31}, y(0) + \beta'_0, \ldots, y(31) + \beta'_{31} \rangle, \\
J'(T) = \langle x(T) + \alpha''_0, \ldots, x(T+31) + \alpha''_{31}, y(T) + \beta''_0, \ldots, y(T+31) + \beta''_{31} \rangle.
\end{gathered}
$$

An algebraic attack to KeeLoq by the plaintext-ciphertext pairs (v', v'') and (w', w'') consists in computing $V_{\mathbb{K}}(I' + J')$. Note that we can indeed assume that the variables clocks are bounded by $T + 31$, that is, we solve over the finite set of variables $\bigcup_{0 \leq t \leq T+31} \{k(t), x(t), y(t)\}$. Actually, the computation of $V_{\mathbb{K}}(I' + J')$ is unfeasible for $T = 528$. If we would assume that $T = 512 = 8 \cdot 64$, we could use the trick of fixed pairs which is described at the end of Section 5. Briefly, if $(u(t), v(t)) \in \mathbb{K}^{64} \times \mathbb{K}^{32}$ denotes the t-state of the system (9), the trick consists in computing enough fixed points $v(0) = v(512)$ by the knowledge of the encryption function and to assume that some of them are in fact fixed points $v(0) = v(64)$. By means of a couple of such plaintext-ciphertext pairs $(v(0), v(0))$, we are reduced to compute $V_{\mathbb{K}}(I' + J')$ for $T = 64$.

The problem now is how to compute $v(512)$ from the ciphertext $v(528)$. If we assume that the variables $k(0), \ldots, k(15)$ are evaluated by the correct corresponding key bits, we can apply the inverse cipher (10) since these bits are the only ones that are involved in the computation of $v(512)$ from $v(528)$. Indeed, the authors of [9, 10] have studied a property based on the disjoint cycles decomposition which is able to distinguish a generic permutation of the set $\mathbb{K}^{32}$ from the KeeLoq encryption function reduced to $T = 512$ clocks. By means of this method, the cost of the search of the correct values of the variables $k(0), \ldots, k(15)$ is assumed to be 2^{52} CPU clocks in the worst case. With an optimized implementation on our CPU @ 3.20GHz, this method should then take $a = 2^{21}$ sec.

Once obtained the correct values $\alpha_1, \ldots, \alpha_{15} \in \mathbb{K}$ of the variables $k(0), \ldots, k(15)$, we have experimented that to compute each set $V_{\mathbb{K}}(I' + J' + E_{\alpha_0,\ldots,\alpha_{15}})$ where $T = 64$ and

$$E_{\alpha_0,\ldots,\alpha_{15}} = \langle k(0) + \alpha_0, \ldots, k(15) + \alpha_{15} \rangle$$

one needs just few tens of milliseconds by using Gröbner bases or SAT solvers. Note that this improves solving times obtained in [9, 10] which are hundreds of milliseconds on a similar CPU.

We conclude that the total running time of this algebraic attack to KeeLoq can be described by the formula

$$a + b \cdot c \cdot 2^{32} + d \tag{11}$$

where b is the average encryption time for $T = 528$ clocks, c is the average percentage of the plaintext space $\mathbb{K}^{32}$ containing enough fixed points $v(0) = v(512)$ and d is the average computing time to obtain the sets $V_{\mathbb{K}}(I' + J' + E_{\alpha_0,\ldots,\alpha_{15}})$ for each couple of such fixed points. The authors of [9, 10] have experimented that there are 26% of keys such that $c = 60\%$. Among the computed fixed points $v(0) = v(512)$, they also assume a good chance that at least one couple $v'(0), v''(0)$ of them is such that $v'(0) = v'(64), v''(0) = v''(64)$. In our experiments we make use of 4 distinct random such "weak keys".

In the following table, we present then statistics of the values b, d. Similarly to Section 6, the solving time d is provided for Gröbner basis algorithms and SAT solvers and it appears in the columns corresponding to SLIMGB, STD, MINISAT, CRYPTOMINISAT. Recall that d is the total computing time for solving the polynomial systems corresponding to all couples that are obtained by computed fixed points $v(0) = v(512)$.

Table 3. 90% Confidence Interval for timings

b (ms)	slimgb (ms)	std (ms)	MiniSat (ms)	CrMiniSat (ms)
$(2.9,\ 3.2) \cdot 2^{-10}$	(93, 183)	(27, 93)	(12, 27)	(16, 32)

In Table 3, we present 90% confidence intervals corresponding to 4 weak key and the correct guess of the 16 variables corresponding to each key. The total encryption time $b \cdot c \cdot 2^{32}$ when $c = 0.6$ is about 2.5 hours by an executable C file. The confidence interval of b, that is, the timing for a single encryption is obtained by the corresponding interval of this total time.

For the polynomial systems involved in the KeeLoq attack, the SAT solvers seem to be the best option. Nevertheless, note that in the tables of Section 6 and 7, the time for computing ANF-to-CNF conversion (see, for instance, [3]) is not considered. In particular, for KeeLoq attack this would imply that Gröbner bases (STD method) and SAT solvers have comparable timings. In any case, for the total running time (11) of this algebraic attack to KeeLoq, the dominant timing is clearly $a = 582$ hours which confirms the results in [9, 10].

8. Conclusions

We have shown in this paper that the notion of system of (ordinary) explicit difference equations over a finite field is useful for modeling the class of "stream and block difference ciphers" which many ciphers of application interest belong to. The appropriate algebraic formalization of such systems and corresponding ciphers requires the theory of difference algebras and ideals, as well the methods of difference Gröbner bases. This formalization allows the study of general properties of the difference ciphers such as their invertibility and periodicity. This study is essential to assess their security by means of suitably defined algebraic attacks. We have illustrated this in practice using two well-known difference ciphers, Bivium and KeeLoq, where Gröbner bases and SAT solvers are also compared. We plan to include different methods from the formal theory of Ordinary Difference Equations for further improving the cryptanalysis of difference ciphers. We believe therefore that the proposed modeling and the corresponding methods will be useful for the development of new applicable ciphers.

9. Acknowledgements

We would like to thank Arcangelo La Bianca, not only for providing computational resources for our experimental sessions, but also for having patiently supported the first author erratic paths in problem solving. We also thank the anonymous referees for the careful reading of the manuscript. We have sincerely appreciated all valuable comments and suggestions as they have significantly improved the readability of the paper.

References

[1] Agarwal, R.P., Difference equations and inequalities. Theory, methods, and applications. Second edition. Monographs and Textbooks in Pure and Applied Mathematics, 228. Marcel Dekker, New York, 2000.

[2] Aschenbrenner, M.; Hillar, C.J., Finite generation of symmetric ideals. Trans. Amer. Math. Soc., 359 (2007), 5171–5192.

[3] Bard, G.V., Algebraic cryptanalysis. Springer, Dordrecht, 2009.

[4] Bardet, M.; Faugère, J.-C.; Salvy, B., On the complexity of the F5 Gröbner basis algorithm, J. Symbolic Comput. 70 (2015), 49 – 70.

[5] Bettale, L., Faugère, J.-C., Perret, L., Hybrid approach for solving multivariate systems over finite fields. J. Math. Cryptol., 3, (2009), 177 – 197.

[6] Bruwer, F.J.; Smit, W.; Kuhn, G.J., US patent 5517187, Microchips and remote control devices comprising same, issued 1996-05-14, assigned to Microchip Technology Inc.

[7] Chen, M.-s.; Ding, J.; Kannwischer, M.; Patarin, J.; Petzoldt, A.; Schmidt, D.; Yang, B.-Y., Rainbow Signature: One of the Three NIST Post-quantum Signature Finalists (2020). https://www.pqcrainbow.org.

[8] Cohn, R.M., Difference algebra. Interscience Publishers John Wiley & Sons, New York-London-Sydney, 1965.

[9] Courtois, N.T.; Bard, G.V.; Bogdanov, A., Periodic ciphers with small blocks and cryptanalysis of KeeLoq. TATRACRYPT '07. Tatra Mt. Math. Publ., 41 (2008), 167–188.

[10] Courtois, N.T.; Bard, G.V.; Wagner, D., Algebraic and slide attacks on KeeLoq. Fast Software Encryption - FSE '08, Lecture Notes in Comput. Sci., 5086, Springer, 2008, 97–115.

[11] Courtois, N.; Klimov, A.; Patarin, J.; Shamir, A., Efficient algorithms for solving overdefined systems of multivariate polynomial equations. Advances in cryptology - EUROCRYPT 2000, 392 – 407, Lecture Notes in Comput. Sci., 1807, Springer, 2000.

[12] Courtois N.T, Meier W., Algebraic attacks on stream ciphers with linear feedback, Advances in Cryptology - EUROCRYPT 2003, Lecture Notes in Comput. Sci., 2656, Springer, 2003, 345 – 359.

[13] Courtois N.T., Pieprzyk J., Cryptanalysis of block ciphers with overdefined systems of equations, Advances in Cryptology - ASIACRYPT 2002, Lecture Notes in Comput. Sci., 2501, Springer, 2002, 267 – 287.

[14] De Cannière, C.; Preneel, B.: TRIVIUM - a stream cipher construction inspired by block cipher design principles. eSTREAM, ECRYPT Stream Cipher Project, Report 2006/021. http://www.ecrypt.eu.org/stream/trivium.html.

[15] Decker, W.; Greuel, G.-M.; Pfister, G.; Schönemann, H.: Singular 4-1-2 — A computer algebra system for polynomial computations (2019). http://www.singular.uni-kl.de.

[16] Gerdt, V.; La Scala, R., Noetherian quotients of the algebra of partial difference polynomials and Gröbner bases of symmetric ideals. J. Algebra, 423 (2015), 1233–1261.

[17] Golomb, S.W., Shift register sequences. Secure and limited-access code generators, efficiency code generators, prescribed property generators, mathematical models. 3rd revised edition. Hackensack, NJ: World Scientific, (2017).

[18] Greuel, G.-M.; Pfister, G., A Singular introduction to commutative algebra. Second, extended edition. With contributions by O. Bachmann, C. Lossen and H. Schönemann. Springer, Berlin, 2008.

[19] Eén, N.; Sörensson, N., MiniSat 2. An open-source SAT solver package. Available from http://minisat.se.

[20] Eibach, T.; Völkel, G.; Pilz, E., Optimising Gröbner Bases on Bivium, Math. Comput. Sci., 3, (2010), 159–172

[21] Faugère, J.-C., A new efficient algorithm for computing Gröbner bases (F4). Effective methods in algebraic geometry (Saint-Malo, 1998). J. Pure Appl. Algebra 139 (1999), 61 – 88.

[22] Faugère, J.-C., A new efficient algorithm for computing Gröbner bases without reduction to zero (F5). Proceedings of the 2002 International Symposium on Symbolic and Algebraic Computation, 2002, 75 – 83

[23] Faugère, J. C.; Gianni, P.; Lazard, D.; Mora, T., Efficient computation of zero-dimensional Gröbner bases by change of ordering. J. Symbolic Comput. 16 (1993), 329 – 344.

[24] Ghorpade, S.R., A note on Nullstellensatz over finite fields, Contributions in Algebra and Algebraic Geometry, Contemporary Mathematics, 738, Amer. Math. Soc., Providence, RI, 2019, 23 – 32.

[25] Higman, G., Ordering by divisibility in abstract algebras. Proc. London Math. Soc. (3), 2 (1952), 326–336.

[26] Huang, Z.; Lin, D., Attacking Bivium and Trivium with the characteristic set method. Progress in cryptology - AFRICACRYPT 2011, Lecture Notes in Comput. Sci., 6737, Springer, 2011, 77–91.

[27] Indesteege S.; Keller N.; Dunkelman, O; , Biham, E.; Preneel B., A Practical Attack on KeeLoq, EUROCRYPT 2008, Lecture Notes in Comput. Sci., 4965, Springer, 2008, 1 – 18.

[28] Jacobson N., Basic algebra. I. Second edition. W. H. Freeman and Company, New York, 1985.

[29] Klein, A., Stream ciphers. Springer, London, 2013.

[30] La Scala, R., Gröbner bases and gradings for partial difference ideals. Math. Comp., 84 (2015), 959–985.

[31] Levin, A., Difference algebra. Algebra and Applications, 8. Springer, New York, 2008.

[32] Maubach, S., Polynomial automorphisms over finite fields, Sedica Math. J., 27 (2001), 343–350

[33] Maximov, A.; Biryukov, A., Two trivial attacks on Trivium. Select. Areas Cryptogr. vol. 4876, 36–55 (2007)

[34] Reid, M., Undergraduate algebraic geometry. London Mathematical Society Student Texts, 12. Cambridge University Press, Cambridge, 1988.

[35] Ritt, J.F.; Raudenbush, H.W., Ideal theory and algebraic difference equations. Trans. Amer. Math. Soc., 46 (1939), 445–453.

[36] Shannon, C.E., Communication theory of secrecy systems. Bell System Tech. J., 28 (1949), 656–715.

[37] Soos, M., CryptoMiniSat 5. An advanced SAT solver. Available from https://www.msoos.org/cryptominisat5.

[38] Stinson, D.R.; Paterson, M.B. Cryptography. Theory and practice. Fourth edition. Textbooks in Mathematics. CRC Press, Boca Raton, FL, 2019.

[39] van den Essen, A., Polynomial Automorphisms and the Jacobian Conjecture. Progress in Mathematics, 190. Birkäuser Verlag, Basel, 2000.

[40] Wibmer, M., Algebraic Difference Equations. Lecture Notes available from Internet.

[41] Wibmer, M., On the dimension of systems of algebraic difference equations. Adv. in Appl. Math., 123 (2021), 102136, 34 pp.

* Dipartimento di Matematica, Università di Bari, Via Orabona 4, 70125 Bari, Italy
Email address: roberto.lascala@uniba.it

** Scientific Analysis Group, Defence Research & Development Organization, Metcalfe House, Delhi-110054, India
Email address: shrawant@gmail.com