Permutations Preserving Divisibility

Robert J. McEliece, Fellow, IEEE, Claude Le Dantec, Member, IEEE, and Philippe M. Piret, Member, IEEE

Abstract—We give a proof of a theorem on the common divisibility of polynomials and permuted polynomials (over GF(2)) by a polynomial \( g(x) \).

Index Terms—Divisibility, permuted polynomials, self-termination, trellis, turbo codes.

I. INTRODUCTION

Let

\[ g(x) = 1 + \sum_{i=0}^{m-1} a_i x^i \]

be a fixed polynomial over GF(2) and let \( A_n(g) \) be the set of polynomials

\[ a(x) = \sum_{i=0}^{n-1} a_i x^i \]

of formal degree \( n - 1 \) over GF(2) that are divisible by \( g(x) \). Let also \( \pi : \mathbb{Z} \to \mathbb{Z} : i \mapsto \pi(i) \) be a permutation and, for any \( a(x) \in A_n(g) \), define

\[ a^\pi(x) = \sum_{i=0}^{n-1} a_i x^{\pi(i)}. \]  

(1)

In [1], a theorem (which generalizes [2], [3]) is given without proof, that characterizes those permutations \( \pi \) such that \( a^\pi(x) \in A_n(g) \) for all \( a(x) \in A_n(g) \). In this correspondence, we give a proof of this statement.

II. PERMUTATIONS AND DIVISIBILITY

For any \( g(x) \) with a nonzero constant term, it is well known that there exists some \( N_0(g) = N_0 \), such that \( g(x) \) divides \( x^{N_0} - 1 \) if and only if \( N \) is a multiple of \( N_0 \). Let us first assume that \( N \) is a multiple of \( N_0 : n = MN_0 \). We associate to any \( a(x) \) of formal degree \( n - 1 \), the two-variable polynomial

\[ b(y, z) = \sum_{j=0}^{N_0-1} b^j(y) z^j \]

with

\[ b^j(y) = \sum_{i=0}^{M-1} a_{N+ji} y^i. \]

Obviously, one has \( a(x) = b(x^{N_0}, x) \).

Lemma 1: The polynomial \( a(x) \) is in \( A_n(g) \) if and only if \( g(x) \) is a divisor of \( b(1, x) \).

Proof: \( b(1, x) \) is the residue of \( a(x) \bmod x^{N_0} - 1 \), and \( g(x) \) is a factor of \( x^{N_0} - 1 \).

Lemma 2: If \( i \equiv j \bmod N_0 \), then any \( \pi \) that preserves \( A_n(g) \) (for (1)) satisfies \( \pi(i) \equiv \pi(j) \bmod N_0 \).

Proof: \( x^i + x^j \) (resp., \( x^{\pi(i)} + x^{\pi(j)} \)) is divisible by \( g(x) \) if and only if \( i \equiv j \bmod N_0 \).

For \( j = 0, \ldots, N_0 - 1 \), let \( \sigma_j : i \mapsto \sigma_j(i) \) be an arbitrary permutation of the coefficients of \( b^j(y) \) and for \( \sigma = (\sigma_0, \ldots, \sigma_{N_0-1}) \), denote by \( a(x) \mapsto a^\sigma(x) \) the permutation of the coefficients of \( a(x) = b(x^{N_0}, x) \) induced by the action of those \( N_0 \) permutations \( \sigma_j \) on the corresponding \( N_0 \) polynomials \( b^j(y) \).

Lemma 3: Any such \( \sigma \) preserves \( A_n(g) \):

\[ a(x) \in A_n(g) \Rightarrow a^\sigma(x) \in A_n(g). \]

Proof: Check that such a \( \sigma \) does not modify \( b(1, x) \) and apply Lemma 1.

The set of all those \( \sigma \) is a group (denoted by \( S \)).

Let \( \rho : j \mapsto \rho(j) \) be an arbitrary permutation of \( \{0, \ldots, N_0 - 1\} \) and let it permute the coefficients \( b^j(y) \) of \( b(y, z) \):

\[ b(y, z) \mapsto b^\rho(y, z) = \sum_j b^{\rho(j)}(y) z^j. \]

Denote by \( a(x) \mapsto a^\rho(x) \) the permutation of the coefficients of \( a(x) = b(x^{N_0}, x) \) induced by this \( \rho \), and by \( \text{Aut}(g, N_0) \) the automorphism (permutation) group of the binary cyclic code \( C \) of length \( N_0 \) generated by \( g(x) \).

Lemma 4: Any such \( \rho \) preserves \( A_n(g) \) if and only if it is an element of \( \text{Aut}(g, N_0) \).

Proof: By definition, if \( \rho \notin \text{Aut}(g, N_0) \), there exists a polynomial \( a(x) \) of degree \( \leq N_0 - 1 \) that is a multiple of \( g(x) \) while \( a^\rho(x) \) is not a multiple of \( g(x) \). Conversely, any \( \rho \in \text{Aut}(g, N_0) \) preserves the divisibility of \( b(1, x) \) by \( g(x) \). Then apply Lemma 1.

In the sequel, the group \( \text{Aut}(g, N_0) \) is denoted by \( R \).

An easy consequence, one obtains the following theorem.

Theorem 5: Any permutation \( \pi \) of \( \{0, \ldots, n - 1\} \) leaves \( A_n(g) \) invariant if and only if it can be written as a finite product \( \pi = \sigma_1 \rho_1 \sigma_2 \rho_2 \sigma_3 \rho_3 \cdots \), all \( \sigma_i \) are in \( S \) and all \( \rho_i \) are in \( R \).

It is obvious that \( R \) contains only the identity and that for any \( (\rho, \sigma) \) there is some \( (\rho', \sigma') \) such that \( a^{\sigma' \rho'}(x) = a^{\rho \sigma}(x) \). Hence any \( \pi \) in Theorem 5 may be written in a unique way as \( \pi = \rho \sigma \) (or as \( \pi = \sigma' \rho' \)). The set of those products \( \pi = \rho \sigma \) is a group which is often called the semidirect product of \( S \) and \( R \). See [4, pp. 20-25] for further comments.

In the more general case, where \( n \) is not a multiple of \( N_0 (n = MN_0 + r \) with \( 1 \leq r \leq N_0 - 1 \), the first \( r \) polynomials \( b^j(y) \) have formal degree \( M \) and the last \( N_0 - r \) ones have formal degree \( M - 1 \). Define then \( S' \) as the set of all \( \sigma = (\sigma_0, \ldots, \sigma_{N_0-1}) \) where, for \( j \leq r - 1 \), \( \sigma_j \) acts on polynomials of degree \( M \), and for \( j \geq r \), \( \sigma_j \) acts on polynomials of degree \( M - 1 \). Define also \( R' \) as the subset of the elements of \( \text{Aut}(g, N_0) \) that preserve (as a set) the first components of \( C \). For example, with \( g(x) = 1 + x^2 + x^3 \) and \( r = 5 \), the permutations (04)(12) and (0421)(36) are in the subset \( R' \) of \( R \), while (051)(324) is in \( R \) but not in \( R' \). The detailed proof of the following corollary is omitted.
III. APPLICATION TO TURBO CODES

Theorem 5 and Corollary 6 characterize which turbo code interleavers are spontaneously “self-terminating.” (see also [6]). Some of them seem to be very good [5] but it is not yet clear whether this self-terminating property does imply some loss in the performances at high signal-to-noise ratio.

ACKNOWLEDGMENT

McEliece’s contribution to this correspondence was performed at the Sony Corporation in Tokyo, Japan, while he was a holder of a Sony Sabbatical Chair. Thanks are due to the referees for their useful comments, and the suggestion to include Corollary 6 in the correspondence.

REFERENCES


Weak Keys in the McEliece Public-Key Cryptosystem

Pierre Loidreau and Nicolas Sendrier, Member, IEEE

Abstract—We show that it is possible to know whether the secret Goppa code of an instance of the McEliece public-key cryptosystem was chosen with a binary generator polynomial. Furthermore, whenever such a weak key is used, we present an attack which can be completed, for codes of length 1024 and dimension 524, with a large, but feasible amount of computation.

Index Terms—Automorphism group of a code, Goppa codes, McEliece cryptosystem, support splitting algorithm.

I. INTRODUCTION

In this correspondence, we consider the security of the McEliece public-key cryptosystem [1]. In this system, the public key is a generator matrix of a linear code. The encryption consists in choosing a codeword in this code to which an error vector of a given weight is added. The decryption is the decoding of these errors. The trap is the knowledge of a decoder for the public code. The security of the cryptosystem lies in the following two assumptions:

- the parameters of the public code are large enough to avoid decoding by a general purpose decoder;
- it is difficult to build a fast (polynomial-time) decoder from the knowledge of the public code alone.

The issues regarding the first assumption were investigated at length in [2]–[4]. Here we deal with attacks related to the second assumption. In the original construction of the McEliece system, the secret code $\Gamma$ is picked in a family of binary Goppa codes of length $n = 2^m$ and error-correcting capability $t$ where $m = 10$ and $t = 50$. The public code is obtained by permuting the coordinates of $\Gamma$.

The support splitting algorithm [5] allows the computation of the permutation between two equivalent binary linear codes. Hence, this algorithm can be used to derive an attack by enumerating all the Goppa codes with suitable parameters. Because of the huge number of Goppa codes, this attack remains unrealistic (for McEliece parameters it can be roughly estimated at $10^{130}$ years on a workstation). However, subfamilies of Goppa codes can be recognized—the weak keys—thanks to their particular structure. Namely, by applying the support splitting algorithm to Goppa codes with a binary generator polynomial one detects their nontrivial automorphism group. This allows an attack by enumerating the Goppa codes with such a property. Once again, this attack remains infeasible since it would require an unreasonable computation time (about $10^7$ years on a workstation). Still, there is a way to greatly reduce its complexity by constructing the much shorter (length about $n/m$) projected idempotent subcode. We present a nontrivial lower bound for this subcode. From this bound we deduce the nontriviality of the code whenever the generator polynomial is binary. Finally, we show how to modify the attack by using the properties of the projected idempotent subcode. With half size parameters ($m = 9$, $t = 28$) our implementation of the attack ran 15 min on a standard workstation. For