Fifteenth International Workshop on Algebraic and Combinatorial Coding Theory
June 18-24, 2016, Albena, Bulgaria *pp. 7–12*

New polynomials for strong algebraic manipulation detection codes [1]

Maksim Alekseev alexeev@vu.spb.ru
State University of Aerospace Instrumentation, St.Petersburg, B. Morskaya str., 67, 190000, Russia

Abstract. Algebraic manipulation detection codes were introduced in 2008 to protect data against a special type of a modification: algebraic manipulation. There are three classes of codes: weak, strong and stronger ones. One of the most effective ways of constructing strong and stronger codes is based on polynomials. In this paper a new family of polynomial functions for strong codes is proposed, which may lead to higher detection capabilities and lower computational complexity of encoding and decoding procedures.

1 Introduction

An algebraic manipulation is a model of an undesirable data modification [1, 2]. For a detailed description of the model, [2] is recommended. Briefly, an additive data distortion is called an algebraic manipulation if its value does not depend on a value-to-be-distorted (a distortion's source has no knowledge about a value-to-be-distorted).

Algebraic manipulation detection (AMD) codes were proposed to guarantee a high level of data integrity in case of algebraic manipulations. The data-to-be-protected is firstly encoded using an AMD code, and then an obtained codeword c is processed. If a codeword c is distorted with an error e, then this will be detected, although with some small error masking probability P_{undet}. It should be mentioned that no external secret key is used: an information in a codeword c is enough to recover the original value if no error happened. The main advantage of AMD codes comparing to classic linear codes is that every q-ary linear code has $q^k - 1$ undetectable errors, where k is a dimension of a code. Thus, distortions corresponding to codewords are undetectable. AMD codes are able to detect any distortion with some nonzero probability $1 - P_{undet}$ because of their nonlinearity.

Strong AMD codes, which are examined in this paper, were proposed for the protection against a strong manipulation model. A strong manipulation model describes situations when a value-to-be-protected is known to the distortion's

[1]The research of the author is supported by the Ministry of Education and Science of the Russian Federation under grant agreement N 2.2716.2014/K from 17.07.2014.

source (but not a codeword c). It may lead to a more sophisticated choice of a distortion's value by the source.

Although originally AMD codes were proposed for providing data integrity in linear secret sharing schemes and fuzzy extractors [1], several new applications were found. AMD codes are utilized in fields such as: design of secure cryptographic devices resistant to injected faults, fault–tolerant storage devices, public key encryption against related key attacks, anonymous quantum communication, and others [3, 4].

This paper will consider systematic AMD codes over $GF(2^n)$, which are the most practical for hardware implementation.

2 Strong AMD codes

Nonlinear strong AMD codes were proposed for the protection against the strong algebraic manipulations. Their encoding procedure is probabilistic and controlled by a random number that is located inside the device is and not observable or accessible (but also may be distorted). Therefore, for each informational message there are several possible codewords and the encoding result is chosen between them on the basis of the random number's value.

Definition. *Let $y \in GF(2^k)$ be an informational message to be encoded, $x \in GF(2^m)$ be a random number. A code*

$$C = \{(y, x, f(x, y))\}$$

is a systematic strong AMD code if the encoding function $f(x, y) \in GF(2^r)$ satisfies the following inequality:

$$P_{undet} \leq \max_{y,e:e_y \neq 0} \frac{|\{x : S(\tilde{c}) = 0\}|}{|\{x\}|} < 1, \tag{1}$$

where the error $e = (e_y \in GF(2^k), e_x \in GF(2^m), e_f \in GF(2^r))$, the distorted codeword $\tilde{c} = c+e = (\tilde{y}, \tilde{x}, \tilde{f}(y)) = (y+e_y, x+e_x, f(x, y)+e_f)$, and the syndrome is $S(\tilde{c}) = f(\tilde{x}, \tilde{y}) + \tilde{f}(x, y)$.

In other words, there are no pairs of y and e with non–zero e_y such that the syndrome $S(\tilde{c})$ will be equal to zero (meaning the error is undetected) at all values of the random variable x. The detection capability of all AMD codes depends on the maximum number of the syndrome's roots. For strong codes, the syndrome equation's roots are values of a random number x for which $S(\tilde{c}) = 0$ for fixed pairs of messages y and errors e. Strong AMD codes provide a detection of errors that distort an informational part y of a codeword, and can also have the same effect on other parts (x and $f(x, y)$). It should be noted that P_{undet} is a worst–case probability of error masking (an achievable bound).

Classic strong AMD codes are based on error correcting codes [2, 4], a multiplication in finite fields [2, 4], message authentication codes [2, 4], a scalar product operation [5], and others.

There is a subset of strong AMD codes that are called *stronger AMD codes*. Stronger codes satisfy the equation (1) for all $e \neq 0$, not only for $e : e_y \neq 0$. Otherwise stated, stronger AMD codes are capable of detecting all error patterns, even if they do not disturb an informational part y of a codeword (only x or/and $f(x, y)$).

There is only one family of systematic stronger AMD codes proposed for the moment. The construction is based on polynomials. Initially, the next encoding polynomial was proposed:

$$f(x, y) = y_1 x + y_2 x^2 + \ldots + y_t x^t + x^{t+2}, \tag{2}$$

$y = (y_1, \ldots, y_t)$, $x, y_i \in GF(2^r)$. Later, Karpovsky et al. developed this code into a sophisticated and flexible construction with a variety of encoding polynomials for different parameters [3]. An encoding function is always a sum of two polynomials: $f(x, y) = A(x) + B(y, x)$. For example, as stated above $A(x) = x^{t+2}$. A power of $A(x)$ is greater than that of $B(y, x)$.

Let us consider another example of an AMD code based on polynomials. The code with $r = 2$ bits, $k = 4r$ bits ($y = (y_1, y_2, y_3, y_4)$), $m = 2r$ (two variables $x = (x_1, x_2)$), $x_i, y_i \in GF(2^r)$ has the following encoding polynomial:

$$f(x, y) = A(x) + B(y, x) = \left(x_1 x_2^3 + x_1^3 x_2\right) + \left(y_1 x_1 + y_2 x_1^2 + y_3 x_2 + y_4 x_2^2\right). \tag{3}$$

The construction based on polynomials is optimal and close to optimal for many sets of parameters [3]. In many applications it is more suitable to use stronger codes based on polynomials than strong codes, even if error detection only in an information part y is required.

However, there is another type of polynomials that can be used as an encoding function for strong AMD codes.

3 Proposed code construction

Let $y \in GF(2^{k=ar})$ and $x \in GF(2^{m=br})$ bits, $a, b, r \geq 1$. Let us define the following family of polynomial functions:

$$f(x, y) = \sum_{i=1}^{a} y_i x_1^{\alpha_{i,1}} x_2^{\alpha_{i,2}} \ldots x_b^{\alpha_{i,b}} = \sum_{i=1}^{a} y_i \prod_{j=1}^{b} x_j^{\alpha_{i,j}}, \tag{4}$$

where y_i, $x_j \in GF(2^r)$, $\alpha_{i,j} \in \{0, 2^l\}$, $0 \leq l < r$. For each consecutive i, a new set of $\alpha_{i,j}$ is selected in order to minimize the sum $\sum_j \alpha_{i,j}$, and the set of all

zeros is prohibited. Also, a number of available sets of α is limited due to the restriction: $\sum_j \alpha_{i,j} < r$.

Example 1.1. Let $a = 2$, $b = 1$, thus, $y = (y_1, y_2)$ and there is one variable x. Then the next sets of α are chosen:

$$\alpha_{1,1} = 2^0,$$
$$\alpha_{2,1} = 2^1.$$

The obtained polynomial is:

$$f(x, y) = y_1 x^{2^0} + y_2 x^{2^1} = y_1 x + y_2 x^2. \quad \square \tag{5}$$

Example 2.1. Let $a = 3$, $b = 3$, thus, $y = (y_1, y_2, y_3)$ and $x = (x_1, x_2, x_3)$. The next sets of α are chosen:

$$\alpha_{1,1} = 2^0,\ \alpha_{1,2} = 0,\ \alpha_{1,3} = 0,$$
$$\alpha_{2,1} = 0,\ \alpha_{2,2} = 2^0,\ \alpha_{2,3} = 0,$$
$$\alpha_{3,1} = 0,\ \alpha_{3,2} = 0,\ \alpha_{3,3} = 2^0.$$

The following polynomial is constructed:

$$f(x, y) = y_1 x_1 + y_2 x_2 + y_3 x_3. \quad \square \tag{6}$$

Example 3.1. Let $a = 6$, $b = 2$, therefore, $y = (y_1, \ldots, y_6)$ and $x = (x_1, x_2)$. Then the next sets of α are chosen:

$$\alpha_{1,1} = 2^0, \quad \alpha_{1,2} = 0,$$
$$\alpha_{2,1} = 0, \quad \alpha_{2,2} = 2^0,$$
$$\alpha_{3,1} = 2^1, \quad \alpha_{3,2} = 0,$$
$$\alpha_{4,1} = 0, \quad \alpha_{4,2} = 2^1,$$
$$\alpha_{5,1} = 2^0, \quad \alpha_{5,2} = 2^0,$$
$$\alpha_{6,1} = 2^0, \quad \alpha_{6,2} = 2^1,$$

The constructed polynomial is:

$$f(x, y) = y_1 x_1 + y_2 x_2 + y_3 x_1^2 + y_4 x_2^2 + y_5 x_1 x_2 + y_6 x_1 x_2^2. \quad \square \tag{7}$$

Theorem. *A code*

$$C = \{y \in GF(2^{ar}), x \in GF(2^{br}), f(x, y) \in GF(2^r)\}$$

with an encoding function $f(x, y)$ defined by the equation (4) *is a strong AMD code providing an error masking probability*

$$P_{undet} \leq 1 - (2^r - v)2^{-(u+1)r},$$

where p is the power of the encoding polynomial, and $p = u(2^r - 1) + v$, $u \leq b$, $v < 2^r - 1$.

From the equation (6), we can see that the proposed code is a generalization of the previously presented strong AMD code based on a scalar product operation, since the family of polynomials (4) includes its encoding function [5].

A code construction defined by the Theorem has the same formula of an error masking probability (that depends on a power of a polynomial) as stronger codes based on polynomials [3]. Let p be a power of a polynomial for a stronger code with parameters k, m and r, and $p^\dagger$ be a power of a proposed polynomial for same parameters. Then if $p^\dagger < p - 1$, a proposed code provides lower P_{undet} and lower computational complexity. If $p^\dagger = p - 1$, a proposed code provides the same P_{undet}, but its polynomial has a lower power and less monomials (thus, lower complexity). When $p^\dagger \geq p$, a stronger code is more efficient than a proposed one. Although a power of an encoding polynomial from Theorem grows faster than that of stronger codes, for small $a = k/r$ it is possible to construct a strong code with a lower power of a polynomial.

In conformity with code definitions, a replacement of stronger AMD codes with proposed strong ones is feasible only in cases when it is sufficient to provide error detection in informational parts y of codewords (not in all parts). However, this requirement seems to be adequate for most applications.

Let us demonstrate several examples when stronger AMD codes can be effectively replaced with proposed strong AMD codes.

Example 1.2. Let $k = 8$ bits, $m = 4$ bits, $r = 4$ bits. Then $a = k/r = 2$, $b = m/r = 2$, $y = (y_1, y_2)$, $y_i, x \in GF(2^4)$ (similar to the Example 1.1). A stronger code from [3] uses the next polynomial as an encoding one:

$$f_1(x, y) = y_1 x + y_2 x^2 + x^5.$$

The code provides $P_{undet} \leq 4/2^4 = 0.25$.

The encoding polynomial for the proposed code with same parameters is presented in the Example 1.1 by the equation (5). It is easy to see that the encoding polynomial has the lower power and requires less computations. The code provides $P_{undet} \leq 2/2^4 = 0.125$. □

Example 2.2. Let $k = m = 12$ bits, $r = 4$ bits. Thus, $a = k/r = 3$, $b = m/r = 3$, $y = (y_1, y_2, y_3)$ and $x = (x_1, x_2, x_3)$, $y_i, x_i \in GF(2^4)$ (similar to the Example 2.1). The polynomial to construct a stronger code from [3] is:

$$f_1(x, y) = y_1 x_1 + y_2 x_2 + y_3 x_3 + x_1^3 + x_2^3 + x_3^3.$$

The code provides $P_{undet} \leq 2/2^4 = 0.125$.

The encoding polynomial for the proposed code with same parameters is presented in the Example 2.1 by the equation (6). This encoding polynomial is linear and, thus, has significantly lower computational complexity. The code provides two times lower $P_{undet} \leq 1/2^4 \approx 0.06$. □

Example 3.2. Let $k = 24$ bits, $m = 8$ bits, $r = 4$ bits. Therefore, $a = k/r = 6$, $b = m/r = 2$, $y = (y_1, \ldots, y_6)$ and $x = (x_1, x_2)$, $y_i, x_i \in GF(2^4)$ (similar to the Example 3.1). The encoding polynomial of a stronger code from [3] is:

$$f_1(x, y) = y_1x_1 + y_2x_2 + y_3x_1^2 + y_4x_2^2 + y_5x_1x_2 + y_6x_1^3 + x_1x_2^3.$$

The encoding polynomial for the proposed code with same parameters is presented in the Example 3.1 by the equation (7). Both codes provide the same $P_{undet} \leq 3/2^4 \approx 0.188$. It is easy to see that the proposed polynomial has a lower power and less monomials and, thus, requires less computations. □

4 Summary

A new family of polynomial encoding functions of strong AMD codes is presented in this paper. Comparing to polynomials used in a stronger AMD code construction [3], in some cases proposed ones have less monomials (in fact, a part $A(x)$ of $f(x, y)$ is omitted) and a lower power. This leads to a lower error masking probability and lower computational complexity, that can be critical for many modern applications. Similar efficient encoding and decoding methods based on the Horner scheme described in [3] can be used for proposed codes. Furthermore, since powers of monomials are sums of $\{0, 2^l\}$, $l \geq 0$, a normal basis of a finite field can be utilized for squaring.

References

[1] R. Cramer, Y. Dodis, S. Fehr, C. Padro, D. Wichs, "Detection of algebraic manipulation with applications to robust secret sharing and fuzzy extractors", *Advances in Cryptology - EUROCRYPT 2008*, pp. 471-488.

[2] E. Jongsma, "Algebraic manipulation detection codes", *Bachelorscriptie*, Mathematisch Instituut, Universiteit Leiden, 2008.

[3] M.G. Karpovsky, Z. Wang, "Design of Strongly Secure Communication and Computation Channels by Nonlinear Error Detecting Codes", *IEEE Trans Computers*, Nov. 2014.

[4] R. Cramer, S. Fehr, C. Padro, "Algebraic Manipulation Detection Codes", *SCIENCE CHINA Mathematics* 56, pp. 1349-1358, 2013.

[5] M. Alekseev, "Two Algebraic Manipulation Detection Codes Based on a Scalar Product Operation", *Proceedings of the 9th International Workshop on Coding and Cryptography 2015 - WCC2015*, Paris, France, April 2015.