New Polynomials for Strong Algebraic Manipulation Detection Codes

Maksim Alekseev

State University of Aerospace Instrumentation
Saint Petersburg
Russia

alexeev@vu.spb.ru

XV Int. Workshop on Algebraic and Combinatorial Coding Theory
(ACCT2016)
Albena, Bulgaria

June 23, 2016

Overview

Algebraic Manipulations

An algebraic manipulation is a model of an undesirable data modification [Jongsma'08].
Briefly, an additive data distortion is called an algebraic manipulation if its value does not depend on a value-to-be-distorted.

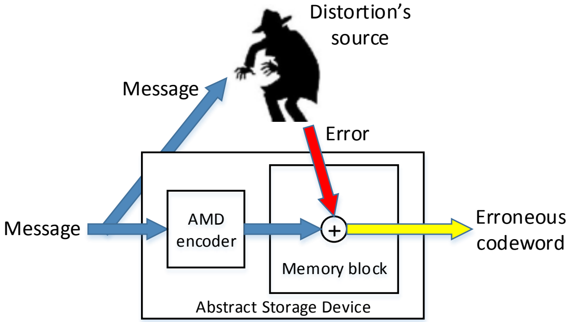
Algebraic manipulation detection (AMD) codes are designed to detect algebraic manipulations.

Types of AMD codes:

- weak AMD codes (also known as robust codes)
- strong AMD codes
    - include "stronger" AMD codes

Only systematic strong AMD codes over $GF(2^n)$ will be examined.

Strong Algebraic Manipulations



A strong algebraic manipulation model assumes:

- An additive error
- An error may be any nonzero element of $GF(2^n)$
- There is a dependency between an error and a message
- The Hamming metric

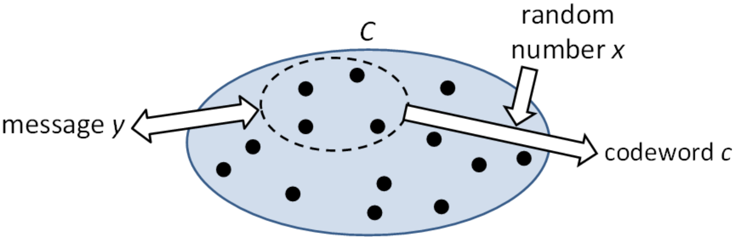
Code Construction

A codeword

$$c = (y, \quad x, \quad f(x,y))$$

consists of 3 parts:

- an informational message $y \in GF(2^k)$,
- a random number $x \in GF(2^m)$,
- a check symbol $f(x,y) \in GF(2^r)$.



Decoding

For a distorted codeword

$$\begin{aligned}\tilde{c} = c + e &= (y + e_y, x + e_x, f(x, y) + e_f) \\ &= (\tilde{y}, \qquad \tilde{x}, \qquad \tilde{f}(x, y)).\end{aligned}$$

compute a syndrome:

$$S(\tilde{c}) = f(\tilde{x}, \tilde{y}) + \tilde{f}(x, y).$$

If $S(\tilde{c}) = 0 \in GF(2^r)$, then no errors detected,
otherwise an error is detected.

The main advantage of AMD codes comparing to classic linear codes is that every q-ary linear code has $q^k - 1$ nonzero undetectable errors, where k is a dimension of a code.
AMD codes have no undetectable errors, each error is detected with some nonzero probability $1 - P_{undet}$.
P_{undet} is a worst–case probability of error masking (an achievable bound).

Applications

Were introduced in 2008 to detect cheaters in linear secret sharing schemes [Cramer et al.'08].

More applications have been found [Cramer et al.'13]:

- Design of secure cryptographic devices (fault-injection attack, ...);
- Fault–tolerant storage devices;
- Robust fuzzy extractors;
- Non-malleable codes;
- Anonymous quantum communication; and others.

Definition of Strong AMD Codes

Definition 1 (Strong AMD Codes)

A code

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

is a systematic strong AMD code if the encoding function $f(x, y)$ *satisfies the following inequality:*

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

Classic strong AMD codes are based on [Cramer et al.'13]:

- Message authentication codes
- Error correction codes
- A multiplication in a finite field, and others.

Definition of Stronger AMD Codes

There is a subset of strong AMD codes that are called *stronger AMD codes*.

Definition 2 (Stronger AMD Codes)

Stronger codes satisfy the equation (1) for all $e \neq 0 \in GF(2^n)$, not only for $e : e_y \neq 0$:

$$P_{undet} \leq \max_{y, e \neq 0} \frac{|\{x : S(\tilde{c}) = 0\}|}{|\{x\}|} < 1. \qquad (2)$$

There are two families of stronger AMD codes [Alekseev'15].
The most efficient one is based on polynomials. Initially, the next encoding polynomial was proposed [Cramer et al.'08]:

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

Examples of Stronger AMD Codes

Karpovsky et al. developed this code into a sophisticated and flexible construction with a variety of encoding polynomials for different parameters [Karpovsky et al.'14].
An encoding function is always a sum of two polynomials:

$$f(x,y) = A(x) + B(y,x).$$

Another example of a stronger AMD code [Karpovsky et al.'14]

The code with $r = 2$ bits,
$k = 4r$ bits $(y \to (y_1, y_2, y_3, y_4))$,
$m = 2r$ (two variables $x \to (x_1, x_2)$),
$x_i, y_i \in GF(2^r)$
has the following encoding polynomial:

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

Proposed family of polynomials

Let $y \in GF(2^{k=ar})$, $y \to (y_1, y_2, \dots, y_a)$, $y_i \in GF(2^r)$, and $x \in GF(2^{m=br})$, $x \to (x_1, x_2, \dots, x_b)$, $x_j \in GF(2^r)$, $a, b, r \geq 1$.
Let us define the following family of polynomial functions:

$$\begin{aligned} f(x,y) &= y_1 x_1^{\alpha_{1,1}} \dots x_b^{\alpha_{1,b}} + y_2 x_1^{\alpha_{2,1}} \dots x_b^{\alpha_{2,b}} + \dots + y_a x_1^{\alpha_{a,1}} \dots x_b^{\alpha_{a,b}} \\ &= \sum_{i=1}^{a} y_i x_1^{\alpha_{i,1}} x_2^{\alpha_{i,2}} \dots x_b^{\alpha_{i,b}} = \sum_{i=1}^{a} y_i \prod_{j=1}^{b} x_j^{\alpha_{i,j}}, \end{aligned} \tag{3}$$

where $\alpha_{i,j} \in \{0, 2^l\}$, $0 \leq l < r$.

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.
A number of available sets of α is limited due to the restriction:

$$\sum_j \alpha_{i,j} < r.$$

Examples of proposed polynomials - I

An example polynomial #1

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

$$\begin{aligned} y_1: &\qquad \alpha_{1,1} = 2^0 &\qquad \to x^1, \\ y_2: &\qquad \alpha_{2,1} = 2^1 &\qquad \to x^2. \end{aligned}$$

The obtained polynomial is:

$$f(x, y) = y_1 x^{2^0} + y_2 x^{2^1} = y_1 x + y_2 x^2.$$

Examples of proposed polynomials - II

An example polynomial #2

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

$$
\begin{array}{lll}
y_1: & \alpha_{1,1} = 2^0, \alpha_{1,2} = 0, \alpha_{1,3} = 0 & \to x_1^1 x_2^0 x_3^0 = x_1, \\
y_2: & \alpha_{2,1} = 0, \alpha_{2,2} = 2^0, \alpha_{2,3} = 0 & \to x_1^0 x_2^1 x_3^0 = x_2, \\
y_3: & \alpha_{3,1} = 0, \alpha_{3,2} = 0, \alpha_{3,3} = 2^0 & \to x_1^0 x_2^0 x_3^1 = x_3.
\end{array}
$$

The following polynomial is constructed:

$$f(x, y) = y_1 x_1 + y_2 x_2 + y_3 x_3.$$

Examples of proposed polynomials - III

An example polynomial #3

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

$$
\begin{aligned}
y_1 : \quad & \alpha_{1,1} = 2^0, \alpha_{1,2} = 0 && \to x_1^1 x_2^0 = x_1, \\
y_2 : \quad & \alpha_{2,1} = 0, \alpha_{2,2} = 2^0 && \to x_1^0 x_2^1 = x_2, \\
y_3 : \quad & \alpha_{3,1} = 2^1, \alpha_{3,2} = 0 && \to x_1^2 x_2^0 = x_1^2, \\
y_4 : \quad & \alpha_{4,1} = 0, \alpha_{4,2} = 2^1 && \to x_1^0 x_2^2 = x_2^2, \\
y_5 : \quad & \alpha_{5,1} = 2^0, \alpha_{5,2} = 2^0 && \to x_1^1 x_2^1 = x_1 x_2, \\
y_6 : \quad & \alpha_{6,1} = 2^0, \alpha_{6,2} = 2^1 && \to x_1^1 x_2^2 = x_1 x_2^2.
\end{aligned}
$$

The constructed polynomial is:

$$f(x, y) = y_1 x_1 + y_2 x_2 + y_3 x_1^2 + y_4 x_2^2 + y_5 x_1 x_2 + y_6 x_1 x_2^2.$$

Code construction

Theorem

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

with an encoding function $f(x,y)$ defined by the equation (3) 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 $p = u(2^r - 1) + v$, $u \leq b$, $v < 2^r - 1$.

The proof is based on the following property of a Galois field $GF(p^m)$:

$$(a+b)^{p^i} = a^{p^i} + b^{p^i}.$$

A code construction defined by the Theorem has the same formula of an error masking probability as stronger codes based on polynomials [Karpovsky et al.'14].

Performance comparison

Let p be a power of a polynomial for a stronger code from [Karpovsky et al.'14] with parameters k, m and r, and p' be a power of a proposed polynomial for same parameters. Then:

- if $p < p - 1$, a proposed code provides lower P_{undet} and lower computational complexity.
- If $p' = p - 1$, a proposed code provides the same P_{undet}, but its polynomial has a lower power and less monomials (thus, lower complexity).
- if $p' \geq p$, a stronger code is more efficient than a proposed one.

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.

Some cases when proposed codes are better - I

In general, a power of a proposed polynomial grows faster than that of stronger codes.
However, for small $a = k/r$ it is possible to construct a strong code with a lower power of a polynomial.

Example 1:

$k = 8$ bits i.e. $y \in GF(2^8)$,
$m = 4$ bits i.e. $x \in GF(2^4)$,
$r = 4$ bits i.e. $f(x, y) \in GF(2^4)$.

Therefore, $a = k/r = 2$, $y \to (y_1, y_2)$,
and $b = m/r = 1$, $x \to x$,
$y_i, x_i \in GF(2^4)$.

Code	$f(x, y)$	P_{undet}
Stronger	$y_1 x + y_2 x^2 + x^5$	0.25
Proposed	$y_1 x + y_2 x^2$	0.125

Some cases when proposed codes are better - II

Example 2:
$k = 12$ bits i.e. $y \in GF(2^{12})$,
$m = 12$ bits i.e. $x \in GF(2^{12})$,
$r = 4$ bits i.e. $f(x, y) \in GF(2^4)$.

Therefore, $a = k/r = 3$, $y \to (y_1, y_2, y_3)$,
and $b = m/r = 3$, $x \to (x_1, x_2, x_3)$,
$y_i, x_i \in GF(2^4)$.

Code	$f(x, y)$	P_{undet}
Stonger	$y_1x_1 + y_2x_2 + y_3x_3 + x_1^3 + x_2^3 + x_3^3$	0.125
Proposed	$y_1x_1 + y_2x_2 + y_3x_3$	0.06

Some cases when proposed codes are better - III

Example 3:
$k = 24$ bits i.e. $y \in GF(2^{24})$,
$m = 8$ bits i.e. $x \in GF(2^8)$,
$r = 4$ bits i.e. $f(x, y) \in GF(2^4)$.

Therefore, $a = k/r = 6$, $y \to (y_1, \ldots, y_6)$,
and $b = m/r = 2$, $x \to (x_1, x_2)$,
$y_i, x_i \in GF(2^4)$.

Code	$f(x,y)$	P_{undet}
Stronger	$y_1x_1 + y_2x_2 + y_3x_1^2 + y_4x_2^2 + y_5x_1x_2 + y_6x_1^3 + x_1x_2^3$	0.188
Proposed	$y_1x_1 + y_2x_2 + y_3x_1^2 + y_4x_2^2 + y_5x_1x_2 + y_6x_1x_2^2$	0.188

Summary:

- A new family of polynomial encoding functions of strong AMD codes is presented.
- 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.
- Efficient encoding and decoding methods based on the Horner scheme can be used for proposed codes [Karpovsky et al.'14].

To do:

- Find a formula to compute a power of $f(x,y)$ (and, thus, P_{undet}) for each set of k, m and r (determine when proposed codes are better).
- Check if codes lay on bounds.

References

E. Jongsma (2008)
Algebraic manipulation detection codes
Bachelorscriptie, Mathematisch Instituut, Universiteit Leiden, 2008.

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

R. Cramer, S. Fehr, C. Padro, (2013)
Algebraic Manipulation Detection Codes
SCIENCE CHINA Mathematics 56, pp. 1349-1358, 2013.

M. G. Karpovsky, Z. Wang, (2014)
Design of Strongly Secure Communication and Computation Channels by Nonlinear Error Detecting Codes
IEEE Trans Computers Nov. 2014.

M. Alekseev, (2015)
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.

Thank you for your attention!