Two Algebraic Manipulation Detection Codes Based on a Scalar Product Operation

Maksim Alekseev

Two Algebraic Manipulation Detection Codes Based on a Scalar Product Operation*

Maksim Alekseev

St.Petersburg State University of Aerospace Instrumentation, St.Petersburg, Russia
alexeev@vu.spb.ru

Abstract. Algebraic manipulation detection codes were introduced in 2008 as a primitive for providing data integrity in the case of a special attack model — an algebraic manipulation. In this paper two constructions of strongly secure algebraic manipulation detection codes are researched. These codes are both based on a scalar product operation in a finite field and guarantee flexibility, low complexity and high detection probability.

Keywords: Algebraic manipulation, secure hardware, scalar product, nonlinear codes.

1 Introduction

Several applications have been found for algebraic manipulation detection (AMD) codes. In the paper a concept of AMD codes is described in accordance to a design of secure cryptographic devices. More detailed introduction to this field can be found in [1, 2]

It is well known that cryptographic algorithms' implementations are vulnerable to side–channel attacks [3]. One of the most efficient attacks is a fault-injection attack. The attack is based on an analysis of a cryptographic device's functioning in a presence of faults and computational errors. Almost all popular ciphers including RSA, DES and AES are recognized to be vulnerable to the fault–injection attack [4].

As a model of injected faults an algebraic manipulation model is considered [5, 6]. The algebraic manipulation is the attack when an attacker is able to modify data being processed or stored (distort it by injecting a fault) by some device, but not able to obtain any information on the value of the data. Two attacker models are distinguished: a weak attacker model and a strong attacker model. The weak attacker is not able to control an input to the device–under–attack. In other words, an informational message input into the device is considered to be uniformly distributed. The strong attacker model is used for situations when the attacker is able to control the input to the device.

To protect cryptographic devices against fault–injection attacks, several methods based on redundancy are often used. The most common is to use linear error detecting codes such as duplication codes, parity check codes and Hamming

* This work was supported by the Ministry of Education and Science of the Russian Federation under grant agreement N 2.2716.2014/K from 17.07.2014

codes. But in cases when an attacker is able to inject specific error patterns, linear codes are not effective. Every q–ary linear code of a dimension k has $q^k - 1$ error patterns (corresponding to non–zero codewords) that are undetectable by this code. Therefore, the attacker can successfully inject one of the codewords as an error into the device.

To protect devices against the weak attacker model nonlinear weakly secure algebraic manipulation detection codes were proposed (these codes are also called robust codes)[7]. The codes are able to detect every error pattern with some non–zero probability. Thus, weakly secure AMD codes provide robust protection against all error patterns even if the attacker is able to inject specific errors. But in the case of the strong attacker model these codes are not effective. The encoding procedure is deterministic that implies that the attacker is able to predict the codeword on the basis of the known input to the device. So the error which can be successfully injected is a difference between some codeword and the predicted one.

Nonlinear strongly secure AMD codes were proposed for the protection against the strong attacker model. Its encoding procedure is probabilistic and controlled by the random number that is located inside the device is and not accessible by the attacker (but may be distorted by him). 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. Strongly secure AMD codes are constructed in such a way that even if the attacker knows the informational message, he is not able to choose an error that will be undetected for all values of the random number. In other words, strongly secure AMD codes have no undetectable errors under the strong attacker model.

Two constructions of strongly secure AMD codes are explored in the paper. In this paper only systematic codes over $GF(2^n)$ are explored since they are more practical. The rest part of the paper is organized as follows. In Section 2, we give definitions of two types of strongly secure AMD codes: a narrow–sense and a wide–sense code. References to main code constructions are presented. In Section 3 a narrow–sense AMD code based on a scalar product operation is examined. A new wide–sense AMD code also based on a scalar product operation is described in Section 4. Both Sections 3 and 4 contain a comparison of the explored construction with the state–of–the–art.

2 Definitions of AMD codes

Definition 1. *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 strongly secure AMD code if the encoding function $f(x, y) \in GF(2^r)$ satisfies the following inequality:

$$\max_{y, e \neq 0} \frac{|\{x : S = 0\}|}{|\{x\}|} < 1, \tag{1}$$

where $|\cdot|$ is the number of entries, the error $e = (e_y \in GF(2^k), e_x \in GF(2^m), e_f \in GF(2^r))$, and the syndrome is $S = f(x, y) + f(x + e_x, y + e_y) + e_f$.

In other words, there are no pairs of y and $e \neq 0$ such that the syndrome S will be equal to zero (meaning the error is undetected) at all values of the random variable x.

In practice it is often required to maintain a data integrity only for an informational message and not for a redundant part of a codeword. Let's introduce the next definition.

Definition 2. *A narrow–sense AMD code is a code that satisfies the inequality (1) only for errors e with a non–zero informational part of the error $e_y \neq 0 \in GF(2^k)$:*

$$\max_{y,e:e_y \neq 0} \frac{|\{x : S = 0\}|}{|\{x\}|} < 1.$$

In opposite, codes defined by the definition 1 will be called wide–sense AMD codes (stronger AMD codes in [8]). In general, wide–sense codes have more complicated constructions and provide a lower detecting probability than narrow–sense ones.

It is easy to see that the upper bound on the error masking probability P follows from the definition of an AMD code:

$$P \leq \max_{y,e \neq 0} \frac{|\{x : S = 0\}|}{|\{x\}|} < 1. \tag{2}$$

In other words, the error masking probability depends on the number of solutions for x to the syndrome equation. A ratio of the maximum number of solutions to the number of all possible values of x gives us the worst–case error masking probability.

An attacker is assumed to have an adaptive behavior meaning he is able to select the most suitable and effective errors to be injected. In that case the upper bound on the error masking probability P is a reasonable measure for a detection capability of a code.

The first AMD code described in 2008 has the following construction [5]:

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

where $y = (y_1, y_2, \ldots, y_t)$, $y_i \in GF(2^r)$, and the encoding function is a polynomial:

$$f(x, y) = y_1 x + y_2 x^2 + \ldots + y_t x^t + x^{t+2}. \tag{3}$$

The code is a wide–sense AMD code with the error masking probability $P \leq (t + 1)2^{-r}$. Karpovsky et al. have developed this code into a sophisticated and flexible construction called the AMD codes based on a generalized Reed—Muller codes.

Further goes a list of existing systematic AMD constructions. There are narrow–sense AMD codes based on: multiplication in finite field [5], message

authentication codes (MAC) [6, 8], error correcting codes (ECC) [6, 8], and others. The only known class of wide–sense AMD codes is a construction based on polynomials [5, 6, 1, 2] (see the encoding function (3)).

A secure memory block protected with an AMD code is demonstrated in the Fig. 1. More protection schemes for different elements of cryptographic devices are presented in [2].

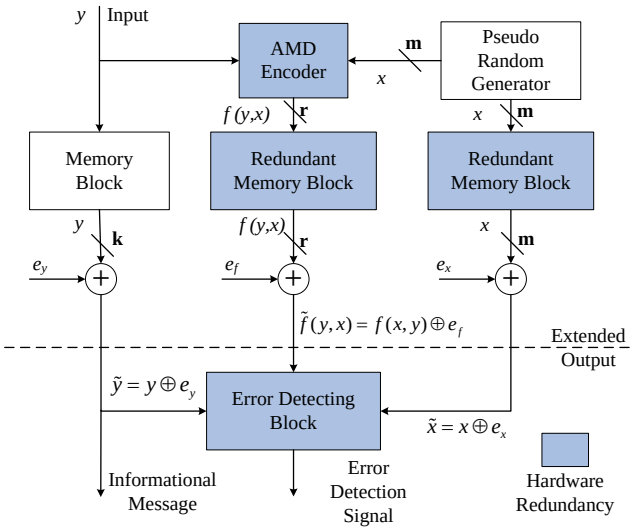

Fig. 1. A principle of a secure memory block protection using an AMD code

In this paper two AMD codes based on a scalar product operation in a finite field are presented. These codes are flexible in terms of the achieved code rate and the error masking probability. The encoding function is a linear polynomial so its computational complexity is quite low.

3 Analysis of a narrow–sense AMD code based on a scalar product operation

3.1 Construction

Let's examine the following code:

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

where $y = (y_1, y_2, \ldots, y_t)$, $x = (x_1, x_2, \ldots, x_t)$, $x_i, y_i \in GF(2^r)$, and an encoding function:

$$f(x,y) = x_1y_1 + x_2y_2 + \ldots + x_ty_t = \sum_{i=1}^{t} x_iy_i.$$

Thereby a scalar product of the informational vector and the random vector is used as the encoding function. Vectors' components x_i, y_i are considered as elements of the finite field $GF(2^r)$. This code was briefly mentioned in [1, 2]. The following analysis of this code shows that it is undeservedly missing in review papers like [6, 8].

Theorem 1. *The code is a narrow–sense AMD code with the error masking probability* $P = 2^{-r}$.

Proof. As it was mentioned above the error masking probability depends on the syndrome's maximum number of roots x for all messages y and errors e. The syndrome of an arbitrary codeword is:

$$\begin{aligned} S &= f(x,y) + f(x+e_x, y+e_y) + e_f \\ &= \sum_{i=1}^{t} x_i y_i + \sum_{i=1}^{t} (x_i + e_{x_i})(y_i + e_{y_i}) + e_f \\ &= \sum_{i=1}^{t} (x_i e_{y_i} + y_i e_{x_i} + e_{y_i} e_{x_i}) + e_f. \end{aligned}$$

By equating the syndrome to zero we get:

$$\sum_{i=1}^{t} x_i e_{y_i} = \sum_{j=1}^{t} (y_j e_{x_j} + e_{y_j} e_{x_j}) + e_f. \tag{4}$$

The syndrome equation is a linear polynomial of, in general, t variables $x_i \in GF(2^r)$. The actual number of x_i that affect the syndrome's value depends on the number of non-zero e_{y_i}. Let's denote this actual number w.

For fixed y and e the RHS of (4) is a constant. It is easy to see that the LHS could be equal to that constant at all values of x only if the constant is zero and all $e_{y_i} = 0$. In that case the syndrome equation does not depend on the random number x.

But the code was claimed to be a narrow–sense AMD code. Therefore the code must detect every error e with the non–zero informational part of the error $e_y \neq 0 \in GF(2^{tr})$ with the probability P stated above. It means that at least one $e_{y_i} \neq 0$, hence $1 \leq w \leq t$. So the LHS of (4) cannot be equal to the RHS at all x. Consequently, there are no pairs of y and $e : e_y \neq 0$ such that equation (4) will be correct at all values of x. That means that by definition 2 the code is a narrow–sense AMD code.

Now let's examine the number of solutions (roots) to the syndrome equation (4). It is well–known that a linear polynomial of w q–ary variables ($q = 2^r$) has q^{w-1} roots. Hence, at $2^{r(w-1)}$ values of the random variable x the syndrome will be equal to zero and an error will not be detected. From (2) follows that the error masking probability is

$$P = \max_{y, e:e_y \neq 0} \frac{|\{x : S = 0\}|}{|\{x\}|} = \frac{2^{r(w-1)}}{2^{rw}} = 2^{-r}. \square$$

Example 1. Let $y \in GF(2^{32})$, $x \in GF(2^{32})$, $r = 8$ bits. Then $y = (y_1, y_2, y_3, y_4)$, $x = (x_1, x_2, x_3, x_4)$, $y_i, x_i \in GF(2^8)$. The encoding function is $f(x, y) = x_1y_1 + x_2y_2 + x_3y_3 + x_4y_4$. The code rate $R = k/(k + m + r) \approx 0.44$. The usage of this code leads to the error masking probability $P = 2^{-8} \approx 4 \cdot 10^{-3}$.

It is worth mentioning that the random variable x could be an element of a smaller field $GF(2^{th})$, $h < r$. In that case we also divide x into t parts: $x = (x_1, \ldots, x_t)$, $x_i \in GF(2^h)$. Each monomial y_ix_i of the encoding function is computed with an assumption that x_i is an element of the bigger filed $GF(2^r)$. That means that we use an element from $GF(2^r)$ with the same decimal representation. With this manipulation we get a code with a reduced random part:

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

where $h < r$. The code provides $P \leq 2^{-h}$. Thus the required size of the random number x is not strictly fixed and can be reduced from tr to th, $1 \leq h \leq r$.

Example 2. Let's modify the code from the previous example in this way. The informational message y is still in $GF(2^{32})$. Let h be equal to 4 bits, $x \in GF(2^{20})$. The required size of the random number x is reduced by 12 bits. It leads to the code rate increase from 0.44 to 0.53 in exchange for the error masking probability decrease to $P \leq 2^{-5} \approx 3 \cdot 10^{-2}$.

3.2 Comparison with other AMD codes

Next let's compare the described code with other constructions.

Comparing to narrow–sense codes the presented one provides more flexibility guaranteeing the same error detecting probability. The size r of the redundant check symbol $f(x, y)$ can be selected from dividers of the informational part's size k. Also, it was shown that the size m of the random number can be reduced from k bits to a smaller value. Described in [5] a narrow–sense AMD code based on the multiplication in a finite field is a special case of the presented code with $t = 1$. For that code the size of an informational message determines all other parameters: $m = r = k$, $P = 2^{-k}$ and $R = 1/3$. An example of the AMD code based on error correcting codes from [6] also has fixed parameters and the code rate equal to $1/3$. The code based on message authentication codes presented in [6] requires at least two times bigger random number x demonstrating the same error masking probability as the presented code with $t = 1$. Also, it should be mentioned that the encoding function of the code described in the section 3.1 generally has lower computational complexity. The encoding polynomial is linear and only t multiplications in a finite field should be done. The size of the field is at most the same as for other constructions ($t = 1$), but mainly the size is t times smaller. It is known that the complexity of a $GF(2^n)$ multiplication implementation is $\mathcal{O}(n \log n \log \log n)$. It means that often t multiplications in $GF(2^{r=k/t})$ require less hardware overhead than one multiplication in $GF(2^k)$.

A comparison of the code based on a scalar product and a wide–sense AMD code based on polynomials demonstrates that the first one requires bigger random number x, but its computational complexity is much lower. The code based

on polynomials uses an encoding polynomial of a degree $t+2$ (at least 3), see equation (3). It has the equal number of multiplications in the same field $GF(2^r)$, but also requires a computation of $x^2, x^3, \ldots, x^t$ and x^{t+2}. Also, the code based on a scalar product provides up to $t+1$ times the lower error masking probability P. For applications like secure hardware design where computational resources are limited it may be more suitable to use the presented code than the one based on polynomials. In that case we lose an ability to detect errors with a zero informational part e_y in exchange for the bigger required random number and lower complexity and error masking probability.

4 A new wide–sense AMD code based on a scalar product operation

4.1 Construction

A scalar product operation can be used to construct also a wide–sense AMD code.
The next construction is proposed by the author:

$$C = \{(y \in GF(2^{ta}), x \in GF(2^{tb}), f(x,y) \in GF(2^{r=a+b}))\},$$

where $y = (y_1, y_2, \ldots, y_t)$ is an informational message, $y_i \in GF(2^a)$, $x = (x_1, x_2, \ldots, x_t)$ is a random number, $x_i \in GF(2^b)$, and t is even. The encoding function is again a scalar product of two vectors, but each vector now is a combination of y_i and x_i:

$v_1 = ((y_1, x_1), (y_3, x_3), \ldots, (y_{t-1}, x_{t-1}))$ and $v_2 = ((y_2, x_2), (y_4, x_4), \ldots, (y_t, x_t))$.

It is assumed that components of vectors are $(y_i, x_i) \in GF(2^{a+b})$. Thus, a computation of the scalar product of these two vectors v_1 and v_2 leads to the following encoding function:

$$f(x,y) = (y_1, x_1)(y_2, x_2) + \ldots + (y_{t-1}, x_{t-1})(y_t, x_t) = \sum_{i=1,3,\ldots}^{t-1} (y_i, x_i)(y_{i+1}, x_{i+1}).$$

Theorem 2. *The code is a wide–sense AMD code with the error masking probability* $P = 2^{-b}$.

Proof. Let's denote each element (y_i, x_i) as z_i and (e_{y_i}, e_{x_i}) as e_{z_i}. Then the encoding polynomial is the following:

$$f(z) = z_1 z_2 + z_3 z_4 + \ldots + z_{t-1} z_t = \sum_{i=1,3,\ldots}^{t-1} z_i z_{i+1}.$$

Similarly to the previous theorem, we get the following syndrome equation:

$$\sum_{i=1,3,\ldots}^{t-1} z_i e_{z_{i+1}} + z_{i+1} e_{z_i} = \sum_{i=1,3,\ldots}^{t-1} e_{z_i} e_{z_{i+1}} + e_f. \tag{5}$$

Thus we obtain a linear polynomial with t q–ary variables z_i, where $q = 2^{r=a+b}$.

The only way for the attacker to eliminate all z_i (and all x_i) from the equation (5) is to inject errors $e_{z_i} = 0$ for all $i = 1, \ldots, t$. Then the syndrome equation will not depend on the random number x:

$$S = e_f.$$

But in this case e_f should also be equal to zero that leads to the all–zero error e. That means there is no non–zero error $e \in GF(2^{ta+tb+a+b})$ that will be masked for all values of z.

Similarly to the previous theorem, the syndrome equation has $q^{w-1} = 2^{r(w-1)}$ solutions for z, where $1 \leq w \leq t$ is the number of non–zero e_{z_i}. It means there are $2^{r(w-1)}$ combinations of vectors (y_i, x_i), $i = 1, \ldots, t$, at which the syndrome equation is equal to zero. But in a strong attack model the informational part y and the error e are fixed, then each $z_i = (y_i, x_i)$ takes only 2^b values instead of $q = 2^{r=a+b}$. Therefore, the maximum number of solutions for x to the syndrome equation is upper bounded by $2^{b(w-1)}$. By definition 1 the code is a wide–sense AMD code with the error masking probability

$$P \leq \max_{y,e \neq 0} \frac{|x : S = 0|}{|x|} = \frac{2^{b(w-1)}}{2^{bw}} = 2^{-b}.\square$$

Corollary 1. *In the case of a weak attack model (when the encoded informational message y is unknown to an attacker and considered to be uniformly distributed) the code provides $P = 2^{-r=-(a+b)}$.*

Proof. It follows from the fact that z_i from the previous theorem now takes $2^{r=a+b}$ possible values because y is not fixed in a weak attack model. Therefore, the syndrome equation has $2^{r(w-1)}$ solutions for z. According to the equation (2) from [7] we get the following expression for the error masking probability in the case of a weak attack model:

$$P_{weak} = \max_{e \neq 0} \frac{|\{(y, x) : S = 0\}|}{|\{y\}| \cdot |\{x\}|} = \frac{2^{r(w-1)}}{2^{rw}} = 2^{-r}.\square$$

Example 3. Let $y \in GF(2^{48})$ be an informational message and $r = 13$ bits. Let $y = (y_1, y_2, y_3, y_4, y_5, y_6)$, $y_i \in GF(2^8)$, $t = 6$. Then $x \in GF(2^{t(r-a)=30})$ and $x = (x_1, x_2, x_3, x_4, x_5, x_6)$, $x_i \in GF(2^{b=5})$. The following wide–sense AMD code can be constructed:

$$C = \{(y \in GF(2^{48}), x \in GF(2^{30}), f(x, y) \in GF(2^{13}))\}$$

using the encoding function

$$f(x, y) = (y_1, x_1)(y_2, x_2) + (y_3, x_3)(y_4, x_4) + (y_5, x_5)(y_6, x_6).$$

The code rate is $R \approx 0.53$, the error masking probability is $P = 2^{-5} \approx 3 \cdot 10^{-2}$. In the case of the weak attacker $P_{weak} = 2^{-13} \approx 10^{-4}$.

It is easy to see that for codes based on polynomials $P = P_{weak}$. It should be mentioned that other strongly secure AMD codes also look likely to provide the same error masking probability for both weak and strong attacker models.

4.2 Comparison with other AMD codes

In comparison with the first presented code based on a scalar product, this one is a wide–sense AMD code. It guarantees the error detecting probability $1 - P$ for all errors while the other one detects only those with a non–zero informational part e_y. For the same t the wide–sense code requires twice less multiplications, but, in general, in a bigger field. On the whole, a wide–sense code has the lower code rate. To get the same error masking probability as the first code does, the wide–sense code requires b more bits for the redundant part (for two codes with fixed $k = tr$ and $m = tb$). However, in that case the wide–sense code has the 2^r times lower error masking probability of the weak algebraic manipulations. In general, these observations are applicable when comparing the code with other narrow-sense AMD codes.

Now let's compare a wide–sense AMD code based on a scalar product to the wide–sense construction based on polynomials. The second one requires a smaller random number's size and check symbol's size resulting in a higher code rate. The first one can reach the at least two times lower error masking probability. In the case of the weak attacker model the first code demonstrates the even lower masking probability. The main point of the wide–sense AMD code based on a scalar product is that it has the simple encoding function that is a linear polynomial. In applications where the hardware complexity is critical the proposed wide–sense code may be chosen. In that case the code rate decreases in exchange for the lower computational complexity.

All discussed AMD constructions are presented in the Table 1: «N» is for the «Narrow–sense», «W» is for the «Wide–sense», «c/c» is for the «computational complexity». For codes based on polynomials, parameters and the error masking probability shown are correct for the special case of the construction (the code with the minimum P among all variants).

Table 1. AMD codes

Construction	Type	Parameters	P	Features
Multiplication [5]	N	$k = m = r$	2^{-k}	inflexible; average c/c;
ECC [6]	N	$k = m = r$	2^{-k}	inflexible; average c/c;
MAC [6]	N	$k = r$, $m = 2k$	2^{-k}	inflexible; require a large x; average c/c;
Polynomials [1, 2, 5, 6]	W	$k = tr$, $m = r$	$(t+1)2^{-r}$	flexible; high or average c/c;
Scalar product (Sect. 3.1)	N	$k = tr$, $m = th$, $h \leq r$	2^{-h}	flexible; low c/c;
Scalar product (Sect. 4.1)	W	$k = ta$, $m = tb$, $r = a + b$	$P = 2^{-b}$ $P_{weak} = 2^{-r}$	flexible; high or average c/c.

5 Conclusion

In this paper, two strongly secure algebraic manipulation detection codes based on a scalar product operation in a finite field were researched. The first one was briefly mentioned in [2], but wasn't examined in details and compared to other codes. The second one is a new wide–sense AMD code proposed by the author. Both code constructions provide the lowest error masking probability and relatively low computational complexity. Codes can be applied to many applications such as robust secret sharing schemes, robust fuzzy extractors, anonymous quantum communication and secure cryptographic devices resistant to fault injection attacks.

References

1. Z. Wang and M.G. Karpovsky, "Algebraic Manipulation Detection Codes and Their Application for Design of Secure Cryptographic Devices", Proc of Int. Symp. on On-Line Testing, 2011.
2. M.G. Karpovsky and Z. Wang, "Design of Strongly Secure Communication and Computation Channels by Nonlinear Error Detecting Codes", IEEE Trans Computers Nov. 2014.
3. Zhou Y., Feng D., "Side–Channel Attacks: Ten Years After Its Publication and the Impacts on Cryptographic Module Security Testing", IACR Cryptology ePrint Archive, 2005.
4. E.S. Abuelyaman, B. Devadoss, "Differential Fault Analysis", International Conference on Internet Computing 2005, pp. 535-544.
5. 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.
6. E. Jongsma, "Algebraic Manipulation Detection Codes", Bachelorscriptie, 6 maart 2008.
7. K.D. Akdemir, Z. Wang, M. G. Karpovsky, and B. Sunar, "Design of Cryptographic Devices Resilient to Fault Injection Attacks Using Nonlinear Robust Codes", Fault Analysis in Cryptography, M. Joye Editor, 2011.
8. R.J.F. Cramer, S. Fehr, C. Padro, "Algebraic Manipulation Detection Codes", SCIENCE CHINA Mathematics 56, 1349-1358, 2013.