On algebraic cryptanalysis of ciphers with low multiplicative complexity

Pavol Zajac

On algebraic cryptanalysis of ciphers with low multiplicative complexity

Pavol Zajac*

Slovak University of Technology in Bratislava,
Ilkovičova 3, SK–812-19 Bratislava, SLOVAKIA,
pavol.zajac@stuba.sk

Abstract. In this article we study the application of multiple right-hand sides (MRHS) equations in algebraic attacks against ciphers with low multiplicative complexity. Each AND gate in the circuit description is converted to a corresponding MRHS equation. The resulting system is transformed into a syndrome decoding problem. The complexity of the decoding problem then depends on the number of AND gates, and on the relative number of known output bits with respect to the number of unknown key bits. This allows us to apply results from coding theory, and explicitly connect the complexity of algebraic cryptanalysis to the multiplicative complexity of the cipher.

Keywords: algebraic cryptanalysis, multiplicative complexity, MRHS

1 Introduction

Multiplicative complexity of a Boolean function is the minimum number of two-input AND gates required to implement the circuit that computes this function. It is possible to compute the multiplicative complexity for small Boolean functions [10]. The multiplicative complexity of a typical cipher with large block or internal state is unknown, but can easily be upper bounded by summing up the multiplicative complexities of (typically small) non-linear building blocks of the cipher (such as S-boxes). We remark that multiplicative complexity of a random Boolean function should scale exponentially in the input size, but for a typical cipher it scales only polynomially due to implementation constraints. This makes low multiplicative complexity a distinguishing property of a practical cipher w.r.t. idealized random model.

Let us suppose that an encryption can be done using a circuit with limited number of two-input AND gates. We start with a set of secret bits, and some known or chosen initialisation vector. Each input of the first AND gate can be computed as a linear combination of secret bits and initialisation vector bits. The inputs of the second AND gate can be computed as a linear combination of secret bits, initialisation bits and the output bit of the first AND gate, and

* This research was supported by grant VEGA 1/0173/13.

so on. The result of the encryption is some final linear combination of the secret bits, initialisation bits, and outputs of all AND-gates. We would like to compute the secret bits, given the output bits and initialisation bits. We can model this problem as a problem of solving a system of non-linear equations over $GF(2)$. There are various algebraic methods that can be applied to this problem.

MRHS equation can represent a non-linear equation over $GF(2)$ in a way, which is particularly useful for constructing equation systems describing ciphers using an S-box as the only means for non-linearity [5]. Similarly, we can adapt this representation to model our problem directly from a AND-XOR circuit description. In recent article [8], we have introduced a new algorithm to solve MRHS equation system. The algorithm converts the system to a group factorization problem, which can be solved either by a global gluing algorithm, or by finding a specific codeword in a linear code. In this article, we explore the application of this algorithm to the algebraic cryptanalysis of ciphers with low multiplicative complexity.

2 Preliminaries

In this section we summarize the notation and the algorithm described in more details in our article [8]. We will use a different notation from [8], as it seems simpler to use row vectors, instead of column vectors. Moreover, we try focus on the algorithm from the coding perspective.

2.1 MRHS equation system

Definition 1. *Let F be a finite field. Multiple-Right-Hand-Sides (MRHS) equation is an expression in the form*

$$x\mathbf{M} = S, \tag{1}$$

where $\mathbf{M} \in F^{(k \times n)}$ is an $(k \times n)$ matrix, and $S \subset F^n$ is a set of n-bit vectors. We say that $x \in F^k$ is a solution of MRHS equation (1), if $x\mathbf{M} \in S$.

A MRHS system $\mathcal{M}$ is a set of m MRHS equations, with the same dimension k, i.e.

$$\mathcal{M} = \big\{x\mathbf{M}_i = S_i; i = 1, 2, \ldots, m\big\},$$

with $\mathbf{M}_i \in F^{(k \times n_i)}$, and $S_i \subset F^{n_i}$, respectively. Vector $x \in F^k$ is a solution of the MRHS system $\mathcal{M}$, if it is a solution of all MRHS equations in $\mathcal{M}$, i.e. $x\mathbf{M}_i \in S_i$ for each $i = 1, 2, \ldots, m$. We denote the set of all solutions of a MRHS system $\mathcal{M}$ by $Sol(\mathcal{M})$. Our goal is to obtain any solution of the system, or to show that no solution of the system exists.

2.2 How to solve MRHS systems with syndrome decoding

Let $\mathcal{M} = \{x\mathbf{M}_1 = S_1, x\mathbf{M}_2 = S_2, \ldots, x\mathbf{M}_m = S_m\}$, where each $\mathbf{M}_i \in F^{(k \times n_i)}$, and let $n = \sum n_i$. Let M denote a single $(k \times n)$ matrix that is composed of individual matrices $\mathbf{M}_i$ in the MRHS system, such that

$$\mathbf{M} = \left(\mathbf{M}_1 | \mathbf{M}_2 | \cdots | \mathbf{M}_m\right).$$

Let us suppose that $n > k$. Than each solution of the MRHS system has a corresponding codeword in (n, k)-code over F generated by $\mathbf{M}$. Let $\mathbf{H}$ denote the $(n-k) \times n$ parity check matrix of this code. For any $c \in F^n$, such that $c\mathbf{H}^T = 0$, there is a unique solution $x \in F^k$ such that $x\mathbf{M} = c$. Moreover, vector x is a solution of the MRHS system if and only if $c \in S = (S_1 \times S_2 \times \cdots \times S_k)$.

The algorithm to solve MRHS equation system works by trying to obtain $c \in S$, such that $c\mathbf{H}^T = 0$. Given c, we can compute x by linear algebra. Vector c can be written as a concatenation of m parts $(c_1, c_2, \ldots, c_m)$, such that $c_i \in S_i$. Let

$$\mathbf{H} = \left(\mathbf{H}_1 | \mathbf{H}_2 | \cdots | \mathbf{H}_m\right),$$

where each $\mathbf{H}_i$ is $(n-k) \times n_i$ matrix. The condition $c\mathbf{H}^T = 0$ can be rewritten as

$$(c_1, c_2, \ldots, c_m) \cdot \begin{pmatrix} \mathbf{H}_1^T \\ \mathbf{H}_2^T \\ \vdots \\ \mathbf{H}_m^T \end{pmatrix} = c_1\mathbf{H}_1^T + c_2\mathbf{H}_2^T + \cdots c_m\mathbf{H}_m^T = 0.$$

Let $S_i\mathbf{H}_i^T$ denote a set of vectors from $F^{(n-k)}$ such that $S_i\mathbf{H}_i^T = \{c_i\mathbf{H}_i^T; c_i \in S_i\}$. We want to choose exactly one vector from each $S_i\mathbf{H}_i^T$ in such a way that all the selected vectors sum to zero, or to show that it is not possible to find such a vector. From the corresponding c_i's we can construct the desired vector c by concatenation, and then obtain x by linear algebra. This corresponds to a group factorisation problem (see [8]), or in a coding theory to a 1-regular decoding problem if the sizes of each S_i are fixed.

Let us reformulate the problem in a different form. Let $r = \sum |S_i|$, and let $\mathbf{R}$ denote a $r \times (n-k)$ matrix with rows composed of vectors from $S_i\mathbf{H}_i^T$. We want to find a solution u of $u\mathbf{R} = 0$, such that $w_H(u) = m$, and if we split u to parts $u = (u_1, u_2, \ldots, u_m)$ of sizes n_i, then each part has $w_H(u_i) = 1$. I.e., we are looking for a vector of a very specific type in the $(r, r-n+k)$-code with parity check matrix $\mathbf{R}^T$. In [8], we append each row of R by m bits that denote to which group $S_i\mathbf{H}_i^T$ the row belongs (i-the additional bit is set to one, and

other $m-1$ bits are set to zero). We get new system

$$u\mathbf{R}' = u \left(\begin{array}{c|l} & 1,0,\ldots,0 \\ & 1,0,\ldots,0 \\ & \vdots \\ \mathbf{R} & 1,0,\ldots,0 \\ & 0,1,\ldots,0 \\ & \vdots \\ & 0,0,\ldots,1 \end{array} \right) = (0,0,\ldots,0,1,1,\ldots,1).$$

Each solution of this system with $w_H(u) = m$ is a solution of the original MRHS system. To find such u, we must solve a syndrome decoding problem for code with parity check matrix $(\mathbf{R}')^T$.

Let us now go in a different direction than in [8]. Let $r_{i,j}$ denote j-th vector in the i-th block of (row) vectors from $\mathbf{R}$. Recall that i-th block of $\mathbf{R}$ corresponds to $S_i\mathbf{H}_i^T$. Let $q = \sum r_{i,n_i}$, and let $\mathbf{Q}$ be a $(r-m) \times (n-k)$ matrix with rows in m blocks $q_{i,1} = r_{i,1} - r_{i,n_i}$, $\ldots$, $q_{i,n_i-1} = r_{i,n_i-1} - r_{i,n_i}$. Let $v\mathbf{Q} = q$. If $w_H(v_i) \leq 1$ for each corresponding $n_i - 1$ long block of v, we can reconstruct u by adding 1 after each part of v where $w_H(v_i) = 0$, and 0 after each part of v where $w_H(v_i) = 1$.

The reformulated problem also corresponds to a classical syndrome decoding problem: Given syndrome q and parity check matrix $\mathbf{Q}^T$, find an error vector v of weight at most m such that $v\mathbf{Q} = q$. We are working with a smaller binary $(r-m, r-m-n+k)$-code. There is a unique solution, if the code can repair up to m errors, i.e., it has a code distance at least $2m+1$. Otherwise, we must search for a specific solution with $w_H(v_i) \leq 1$ for each part of v (if the original MRHS system has a solution, such a vector must exist). The code under consideration is smaller than the one given by $\mathbf{R}'$, so we expect it to be easier to solve.

3 Using MRHS representation to model circuits of low multiplicative complexity

In this section we focus on connection of algebraic cryptanalysis and multiplicative complexity of the circuit. This connection was already spotted by Courtois in [3]. In this section we show how to efficiently model algebraic cryptanalysis of ciphers with low multiplicative complexity via MRHS equations, and in the following section, we will apply our algorithm from [8] and latest general decoding algorithms from [2] to estimate the complexity of attacking such a cipher.

Definition 2. *Let $F : GF(2)^\nu \to GF(2)^\kappa$ be a vectorial Boolean function. Multiplicative complexity of F, denoted by $MC(F)$, is the minimum number of $GF(2)$ multiplications required to compute, using only operations from $GF(2)$, the value of F in arbitrary point $x \in GF(2)^\nu$.*

If function F has multiplicative complexity $MC(F) = \mu$, there exists a computational circuit composed of two-input AND gates and arbitrary number of XOR gates that computes a value of F for any input $x \in GF(2)^\nu$. We can model the computation as a sequence of $\nu + \mu + \kappa$ bits (x_i), $x_i \in GF(2)$. The first ν bits, i.e., x_i for $i = 1, 2, \ldots, \nu$ represent the input bits. The next μ bits are computed as the outputs of two-input AND-gates. Each input of the AND gate is an arbitrary affine function[1] of the previous bits. I.e., $x_{\nu+i} = (a_{i,0} + \bigoplus_{j=1}^{\nu+i-1} a_{i,j}x_j) \cdot (b_{i,0} + \bigoplus_{j=1}^{\nu+i-1} b_{i,j}x_j)$, where $a_{i,j}, b_{i,j} \in GF(2)$, $i = 1, 2, \ldots, \mu$, and the $\bigoplus$ represents the sum over $GF(2)$. The two inputs of the same AND gate must be linearly independent, otherwise we would get an affine function, and the AND gate in question would not be required. Finally, the last κ bits are the outputs of F, which can be computed as any affine function of the previous bits, i.e., $x_{\nu+\mu+i} = c_{i,0} \oplus_{j=1}^{\nu+\mu} c_{i,j}x_j$. Each of the outputs of AND-gates must be used to compute at least one output bit, else the AND-gate would not be required. The output functions should be linearly independent, otherwise some of the outputs would be redundant (they could be computed as a linear combination of other outputs only).

This model can be adapted to represent most of the known stream ciphers, hash functions or block ciphers (we will call them just ciphers for the sake of simplicity), even if their multiplicative complexity (as a whole) is not known. This requires that the cipher under consideration can be written as a sequence of small operations with known multiplicative complexity (such as 4×4 bijective S-boxes analyzed in [10]), or other operations which can be realised by a limited number of two-input AND gates. We can work with a circuit description with possibly higher number of AND-gates than the minimum possible, but still low enough for the attack. It might be possible to reduce the circuit further by some optimization techniques [3].

Due to the requirement of efficient implementation of cipher the number of AND gates for a typical cipher would be very small in comparison of the expected multiplicative complexity of a random function. In the later text, μ will denote the number of two-input AND gates in the circuit representation of the cipher regardless of whether this number is the multiplicative complexity, or just an upper bound of multiplicative complexity obtained from some existing implementation of the cipher.

The (traditional) main problem of the cryptanalysis is to compute the input bits $x_1, x_2, \ldots, x_\nu$, if we are given the output bits $x_{\nu+\mu+1}, x_{\nu+\mu+2}, \ldots, x_{\nu+\mu+\kappa}$ (inverting the one way function). In practice, some of the input bits might be known, e.g., the input block of the block cipher, initialization vector of stream cipher, etc. We can model the known inputs by adding additional outputs that are identically equal to the input values, or add the known bits to the affine constants a_0, b_0, and simplify the circuit where possible.

The circuit representation leads to direct translation of the cryptanalytic problem to a problem of solving a set of non-linear (degree 2) equations over

[1] The bits of a possible initialisation vector from the introduction part become affine constants in this model.

$GF(2)$, which is the domain of the algebraic cryptanalysis. There are many techniques how to solve such a system, such as using some of the Gröbner basis techniques [4], translation to SAT problem [1], and others.

To translate the problem to MRHS representation we do the following:

1. For each of the μ AND-gates, write an MRHS equation using $(\nu + \mu) \times 3$ left-hand side matrix M_i. The first column of M_i contains coefficients $a_{i,j}$, $j = 1, \ldots \nu + i - 1$ (others are zero), the second column of M_i contains coefficients $b_{i,j}$, $j = 1, \ldots \nu + i - 1$, and the third column contains just a single non-zero coefficient at position $\nu + i$ (representing the output bit x_i).
2. The right-hand side of each M_i encodes the AND-gate operation, and the effect of affine constants. If affine constants $a_{i,0} = 0$ and $b_{i,0} = 0$, the right hand side contains four 3-bit vectors $S_i = \{(0,0,0), (0,1,0), (1,0,0), (1,1,1)\}$. The third bit in each vector is the output of the AND gate when the first two bits are its inputs, for all 4 possible input bit combinations. Affine constants are "added" to the input bit prior to using the AND-gate. Thus, the right-hand side with affine constants taken into account is given as $S_i = \{(a_{i,0}, b_{i,0}, 0), (a_{i,0}, b_{i,0} + 1, 0), (a_{i,0} + 1, b_{i,0}, 0), (a_{i,0} + 1, b_{i,0} + 1, 1)\}$.
3. The original equation system has $\nu + \mu$ unknowns. The m known output bits are right hand sides of κ (independent) linear equations (given by coefficients $c_{i,j}$) in these $\nu + \mu$ unknowns. By linear algebra we can express κ of the $\nu + \mu$ unknowns as affine functions of just $\nu + \mu - \kappa$ unknowns.
4. We can simplify a system of μ MRHS equations over $GF(2)^{(\nu+\mu)}$ to a smaller system over $GF(2)^{(\nu+\mu-\kappa)}$ by applying the affine substitutions from the previous step. We add the linear part to the left hand sides of MRHS equations (so we cancel out the substituted variables), and add the constant to the corresponding coordinate of all vectors in the right-hand side sets.

It is possible that some of the new MRHS equations after the last step contain linearly dependent columns. These equations can be simplified by Agreeing algorithm [6], so the final system is even simpler. However, in the following we will suppose the worst case system that cannot be further simplified after the substitution step.

4 On the complexity of algebraic attacks on ciphers with low multiplicative complexity

In section 3 we have constructed an MRHS equation system, whose solution is the solution of the problem of inverting a one way function $F : GF(2)^{\nu} \to GF(2)^{\kappa}$ with multiplicative complexity at most μ. The final MRHS system has $\nu + \mu - \kappa$ unknowns, with μ MRHS equations with $|S_i| = 4$ solutions each. The system matrix M has the size $(\nu + \mu - \kappa) \times (3\mu)$. I.e., $n = 3\mu$, $k = \mu + \nu - \kappa$, $m = \mu$, $r = 4\mu$ in the notation of section 2.2.

Applying the algorithm in section 2.2, we transform the problem of solving the MRHS system into a specific syndrome decoding problem $v\mathbf{Q} = q$. We are

working with a binary $(r-m, r-m-n+k)$-code. Translating back the values, this means that code words have length $r-m = 3\mu$, and the code dimension is $r-m-n+k = 4\mu-\mu-3\mu+\mu+\nu-\kappa = \mu+\nu-\kappa$. Code rate is thus

$$R = \frac{\mu+\nu-\kappa}{3\mu} = 1/3 + \frac{\nu-\kappa}{3\mu}.$$

Furthermore, we want to decode at most $m = \mu$ errors, so we would like the code with distance at least $2\mu+1$. The Singleton bound $k+d \leq n+1$ says that the code distance is less than $n+1-k = 2\mu+1+(\nu-\kappa)$. If $\nu > \kappa$, there are more unknown bits than the number of restrictions (known bits), so we expect that there are more solutions of the inversion problem.

If $\nu = \kappa$, we need an MDS code to provide a unique solution, i.e., every 2μ rows of Q must be linearly independent. This is not possible in the case of binary codes. However, we look for a specific error vector, which is unique, if the original solution of the system is unique. Other (incorrect solutions) must contain some linear combinations of more rows of Q in a single block, corresponding to picking a linear combination of right-hand sides, instead of a single right-hand side. We need a special decoding algorithm that does not accept such solutions. Furthermore, due to the construction of matrix Q, the expected number of errors is lower than the upper bound μ. If there is an independent and equal chance to pick each right hand for each MRHS equation, we expect on average to only get $3/4\mu$ non-zero error bits.

If $\nu < \kappa$, i.e., we have more additional information than unknowns. Thus, we have a higher chance that the code can decode as much as μ (or the expected $3/4\mu$) errors. The code rate R is at most $1/3$, and goes to zero as $\kappa - \nu$ grows relative to the number of AND gates. Thus, the excess bits also lower the code rate, which means the complexity of the syndrome decoding approach is lower. If the number of excess known bits is higher than the number of AND gates, the problem degenerates to a simple linear algebra. On the other hand, if $\kappa - \nu = 0$, the number of AND gates does not influence the code rate, but it still increases the dimension of the problem, so the complexity of the algebraic attack grows exponentially in the multiplicative complexity.

The new results on general decoding algorithms [2] give some (asymptotic) upper bounds on decoding complexity for random linear codes. They estimate the worst case time complexity to be $2^{0.1019n} = 2^{0.3057\mu}$ with space required bounded by $2^{0.0769n} = 2^{0.2307\mu}$. This means that to keep the attack complexity above the expected cost of the brute-force attack 2^{ν}, we need $\nu < 0.3057\mu$, or equivalently $\mu > 3.27\nu$, i.e., at least 3.27 times more AND gates than the number of unknowns/key-bits. A prominent example where this does not hold is the key-stream from state generation for the Trivium stream cipher, where $\nu = \kappa = 288$ (internal state size), and the multiplicative complexity μ is at most 3ν (only 3 AND gates in each clock). Please note, that this does not apply to the key recovery of full Trivium cipher, where $\nu = 80$, and the multiplicative complexity includes all initial rounds.

5 Concluding remarks

The MRHS equations and simple linear algebra can be used to transform an instance of algebraic cryptanalysis to an instance of decoding problem, either a 1-regular decoding problem, or a specific syndrome decoding problem. We can then apply new algorithms from the decoding area to decrypt standard block and stream ciphers, or to invert hash functions. This is especially useful for ciphers with (very) low multiplicative complexity. Moreover, the new results in syndrome decoding (and generalized birthday attacks) can provide us strong bounds on the required multiplicative complexity of ciphers.

Sendrier in [7] analyses the situation when attacker needs to decode only one out of many syndromes in the same code. The algorithm gives a significant advantage to the attacker when the number of errors is small. When applying the algebraic cryptanalysis to obtain a symmetric encryption key we have many possible instances with the same key as unknown, and we need to decode just one of them. However, the instances have different (but related) parity-check matrices. We might ask, whether it is possible to adapt some decoding methods to decode one out of many instances in this specific form, and profit from many possible plaintext-ciphertext pairs in the attack.

Furthermore, the construction of matrix Q removes last vectors from the groups of vectors in R, sums them up and produces the instance to decode. With a single instance of algebraic cryptanalysis, we might construct many instances of decoding problem by random selection of vectors that are taken out of R matrix. Again, we might ask, whether this can speed-up or simplify the decoding algorithm.

MRHS equations are also useful to encode the instances of algebraic cryptanalysis on the S-box level [5]. Furthermore, it is possible to use various local reduction techniques (see e.g.[9]) combined with Gluing to transform the system in the polynomial time to a more compact representation, and only then apply decoding techniques or generalized birthday attacks. It is an interesting question whether we can get any significant asymptotic and/or practical improvements in this way.

References

1. Bard, G.V., Courtois, N.T., Jefferson, C.: Efficient methods for conversion and solution of sparse systems of low-degree multivariate polynomials over GF(2) via SAT-solvers. Cryptology ePrint Archive, Report 2007/024 (2007), http://eprint.iacr.org/
2. Becker, A., Joux, A., May, A., Meurer, A.: Decoding random binary linear codes in $2^{n/20}$: How $1 + 1 = 0$ improves information set decoding. In: Advances in Cryptology–EUROCRYPT 2012, pp. 520–536. Springer (2012)
3. Courtois, N., Hulme, D., Mourouzis, T.: Solving circuit optimisation problems in cryptography and cryptanalysis. Cryptology ePrint Archive, Report 2011/475 (2011)

4. Faugère, J.C.: A new efficient algorithm for computing Gröbner bases without reduction to zero (F5). In: Workshop on Applications of Commutative Algebra, Catania, Italy, 3-6 April 2002. ACM Press (2002)
5. Raddum, H.: MRHS equation systems. In: Selected Areas in Cryptography. pp. 232–245 (2007)
6. Raddum, H., Semaev, I.: Solving multiple right hand sides linear equations. Des. Codes Cryptography 49(1-3), 147–160 (2008)
7. Sendrier, N.: Decoding one out of many. In: Post-quantum cryptography, pp. 51–67. Springer (2011)
8. Zajac, P.: A new method to solve MRHS equation systems and its connection to group factorization. Journal of Mathematical Cryptology 7(4), 367–381 (2013)
9. Zajac, P., Čagala, R.: Local reduction and the algebraic cryptanalysis of the block cipher GOST. Periodica Mathematica Hungarica 65(2), 239–255 (2012)
10. Zajac, P., Jókay, M.: Multiplicative complexity of bijective 4×4 S-boxes. Cryptography and Communications 6(3), 255–277 (2014)