LOOKING INSIDE $\mathcal{AES}$ AND $\mathcal{BES}$

Ilia Toli, Alberto Zanoni
Università degli Studi di Pisa
Dipartimento di Matematica "Leonida Tonelli"
Via F. Buonarroti 2, 56127 Pisa, Italy
{toli, zanoni}@posso.dm.unipi.it

Abstract We analyze an algebraic representation of $\mathcal{AES}$-128 as an embedding in $\mathcal{BES}$, due to Murphy and Robshaw. We present two systems of equations S^* and K^* concerning encryption and key generation processes. After some simple but rather cumbersome substitutions, we should obtain two new systems $\mathcal{C}_1$ and $\mathcal{C}_2$. $\mathcal{C}_1$ has 16 very dense equations of degree up to 255 in each of its 16 variables. With a single pair (p, c), with p a cleartext and c its encryption, its roots give all possible keys that should encrypt p to c. $\mathcal{C}_2$ may be defined using 11 or more pairs (p, c), and has 16 times as many equations in 176 variables. K^* and most of S^* is invariant for all key choices.

Keywords: Advanced Encryption Standard, $\mathcal{AES}$, $\mathcal{BES}$, $\mathcal{DES}$, Cryptography, Gröbner bases, Computer Algebra

Introduction

Rijndael is a block cipher, that encrypts blocks of 128, 192, and 256 bits using symmetric keys of 128, 192, and 256 bits. It was designed with a particular attention to bit-level attacks, such as *linear* and *differential cryptanalysis*. Its resistance to such attacks is the dichotomy between operations in $\mathbf{F} = GF(2^8)$ and $GF(2)$. Since its proposal, many new bit-level attacks, such as *impossible differential* and *truncated differential* have been proposed. Most of them break with some efficiency reduced versions of Rijndael, but they are not much better than exhaustive key search in the general case. In practice they are mainly academic arguments rather than real world threats to the security of $\mathcal{AES}$. The interested reader can find an account and some references about these cryptological tools in [ODR].

Another, new, cryptological tool is the algebraic representation of the cipher [MR; FSW; CP]. In this case, an eavesdropper tries to write the whole set of operations and parameters of the cipher as a system of polynomial equations, which he/she next tries to solve. In general, the systems are enormous. Solving them using general purpose techniques, such as Gröbner bases [CLO] is considered the wrong way to face the problem. However, the systems have sometimes an intrinsic structure, and the task may get easier. Not too much research is done in the topic: in particular, $\mathcal{AES}$ seems to have been designed without considering algebraic cryptanalysis tools.

In this paper we focus on the $\mathcal{BES}$ algebraic approach, due to Murphy and Robshaw [MR]. We present some algebraic aspects of representing $\mathcal{AES}$ as a system of polynomial equations following the $\mathcal{BES}$ approach. By means of successive substitutions, we are able to eliminate all intermediate variables, obtaining two systems $S^\star$ and $K^\star$ whose solution corresponds to code breaking. Actually, they are very complicated: their resolution is not trivial at all.

1. The $\mathcal{AES}$-128 cipher

The $\mathcal{AES}$ encryption algorithm is sketched below:

- Input a cleartext **x**.
  - Initialize **State = x**.
  - perform an operation **AddRoundKey**, in which **RoundKey** is **xor**-ed with the **State**.
- For nine (first to ninth) rounds:
  - perform a substitution operation called **SubBytes** on **State**, using an $\mathcal{S}$-box.
  - perform a permutation **ShiftRows** on **State**.
  - perform an operation **MixColumns** on **State**.
  - perform **AddRoundKey.**
- The tenth (last) round:
  - perform **SubBytes**.
  - perform **ShiftRows**.
  - perform **AddRoundKey**.
- Define the ciphertext **y** to be the **State**.

All $\mathcal{AES}$ operations are byte-oriented. The cleartext, ciphertext, and each output of intermediate steps of encryption and decryption algorithms are thought of as 4×4 matrices of bytes. The operations on each

Figure 1. The **ShiftRows** operation on $\mathcal{AES}$

byte are those of the finite field $\mathbf{F} = GF(2^8)$. The elements are thought of as polynomials with coefficients in $GF(2)$, $\mathbf{mod}\,(m(t))$, the so-called *Rijndael polynomial* :

$$m(t) = t^8 + t^4 + t^3 + t + 1 = \mathtt{11b}\ . \tag{1}$$

They are represented as integers pairs in hexadecimal representation. If interpreted as eight–bit binary strings, we have the $t-$**term** exponents.

The **SubBytes** operation substitutes each of the bytes x with $\mathcal{S}(x)$:

$$\begin{aligned}\mathcal{S}(x) \;=\; & \mathtt{63} + \mathtt{8f}x^{127} + \mathtt{b5}x^{191} + \mathtt{01}x^{223} + \mathtt{f4}x^{239} + \\ & \mathtt{25}x^{247} + \mathtt{f9}x^{251} + \mathtt{09}x^{253} + \mathtt{05}x^{254}\end{aligned}$$

Actually, $\mathcal{S}(x)$ is a permutation polynomial.

The **ShiftRows** operation permutes bytes in each row, see Figure 1.

The **MixColumns** operation performs a permutation of bytes in each column using a matrix in $GL(\mathbf{F}, 4)$, introduced later in Section 2.1. In practice, the columns are considered as polynomials in $\mathbf{F}[x]$, and multiplied mod $(x^4 + 1)$ by the polynomial $a(x)$:

$$a(x) = \mathtt{03}x^3 + \mathtt{01}x^2 + \mathtt{01}x + \mathtt{02}. \tag{2}$$

Now consider the key schedule. The key used in every cipher round is successively obtained by the key of the precedent one. Here is the complete procedure.

- Input a key $\mathbf{h}_0$. Initialize $\mathbf{H}_0 = \mathbf{h}_0$.
- For each round $r = 1, \ldots, 10$, permute **(RotWord)** the sub-vector formed by the last four elements (word) of $\mathbf{H}_{r-1}$, see Figure 2.
- Perform the **Sub Word** ($\mathcal{S}$-**box** on each byte) operation on the obtained result, and add the vector $\mathbf{Rcon}_r = (t^{r-1}, 0, 0, 0)$.
- Define the other elements by means of bitwise **xor** operations in terms of the obtained result and other words from $\mathbf{H}_{r-1}$.
- Define the set of keys to be $\mathbf{h}$ to be $\{\mathbf{H}_r \mid r = 0, \ldots, 10\}$.

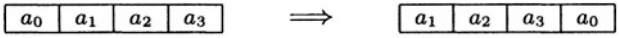

Figure 2. The **RotWord** operation on $\mathcal{AES}$

Consider each vector as a four-words set, indicated with a second index ranging from 0 to 3 indicating single parts. For $\mathbf{y} \in \mathbf{F}^4$ we put $\varphi_A^r(\mathbf{y}) = \mathbf{SubWord}(\mathbf{RotWord}'(\mathbf{y})) + \mathbf{Rcon}_r$. The r^{th} round for $\mathcal{AES}$ key generation scheme is:

$$\mathcal{K}_A = \begin{cases} \mathbf{H}_{r0} = \varphi_A^r(\mathbf{H}_{r-1,3}) \\ \mathbf{H}_{r1} = \mathbf{H}_{r0} + \mathbf{H}_{r-1,1} \\ \mathbf{H}_{r2} = \mathbf{H}_{r1} + \mathbf{H}_{r-1,2} \\ \mathbf{H}_{r3} = \mathbf{H}_{r2} + \mathbf{H}_{r-1,3} \end{cases} \Longrightarrow \mathbf{H}_r = (\mathbf{H}_{r0}, \mathbf{H}_{r1}, \mathbf{H}_{r2}, \mathbf{H}_{r3}) \quad (3)$$

2. The $\mathcal{BES}$ cipher

We start from the $\mathcal{BES}$ cipher, in which $\mathcal{AES}$ is embedded by a "natural" mapping. $\mathcal{BES}$ operations involve only computations in $\mathbf{F}$. This permits to describe $\mathcal{AES}$ using polynomial equation systems. Solving them means to find the key or an alias, and therefore to break the code.

The state spaces of $\mathcal{AES}$ and $\mathcal{BES}$ are respectively $\mathbf{A} = \mathbf{F}^{16}$ and $\mathbf{B} = \mathbf{F}^{128}$. The basic tool for embedding is the *conjugation* ϕ, taking for each value in $\mathbf{F}$ eight successive square powers.

$$\mathbf{F} \ni a \longmapsto \phi(a) = \tilde{\mathbf{a}} = (a^{2^0}, a^{2^1}, ..., a^{2^7}) \in \mathbf{F}^8 \quad (4)$$

$$\mathbf{F}^n \ni \mathbf{a} \longmapsto \phi(\mathbf{a}) = \tilde{\mathbf{a}} = (\phi(a_0), ..., \phi(a_7)) \in \mathbf{F}^{8n} \quad (5)$$

It is easily verified that (with $0^{-1} = 0$)

$$\phi(\mathbf{a} + \mathbf{a}') = \phi(\mathbf{a}) + \phi(\mathbf{a}') \qquad \text{and} \qquad \phi(\mathbf{a}^{-1}) = \phi(\mathbf{a})^{-1} \quad (6)$$

and we define $\mathbf{B}_\mathbf{A} = \phi(\mathbf{A}) \subset \mathbf{B}$ as the subset of $\mathbf{B}$ corresponding to $\mathbf{A}$.

Let $\mathbf{p}, \mathbf{c} \in \mathbf{B}$ be the plaintext and ciphertext, respectively; $\mathbf{w}_i, \mathbf{x}_i \in \mathbf{B}$ $(0 \leq i \leq 9)$ the state vectors before and after the inversion phases, and $\mathbf{h}_i \in \mathbf{B}$ the used keys.

2.1 Correspondence

The matrix $L_A : \mathbf{F} \simeq GF(2)^8 \to GF(2)^8 \simeq \mathbf{F}$ for the one-byte affine transformation in the $\mathcal{S}$-box phase can be represented by the polynomial function $f : \mathbf{F} \to \mathbf{F}$:

$$f(a) = \sum_{k=0}^{7} \lambda_k a^{2^k} \quad (7)$$

with

$$\begin{array}{ll} \lambda_0 = t^2 + 1 & \lambda_4 = t^7 + t^6 + t^5 + t^4 + t^2 \\ \lambda_1 = t^3 + 1 & \lambda_5 = 1 \\ \lambda_2 = t^7 + t^6 + t^5 + t^4 + t^3 + 1 & \lambda_6 = t^7 + t^5 + t^4 + t^2 + 1 \\ \lambda_3 = t^5 + t^2 + 1 & \lambda_7 = t^7 + t^3 + t^2 + t + 1 \end{array} \tag{8}$$

Working in $\mathbf{B}$, $L_B(a) = \phi(L_A(a)) = (f(a)^{2^0}, \ldots, f(a)^{2^7})$. The successive squares of f are needed, and the answer is given by a simple induction with basic step

$$(f(a))^2 = \left(\sum_{k=0}^{7} \lambda_k a^{2^k}\right)^2 = \sum_{k=0}^{7} \lambda_k^2 a^{2^k \cdot 2} = \sum_{k=0}^{7} \lambda_k^2 a^{2^{k+1}} \tag{9}$$

The resulting matrix, still indicated with L_B, is

$$L_B = [l_{ij}]_{i,j=0,\ldots 7} \qquad \text{with} \qquad l_{ij} = \lambda^{2^i}_{(8-i+j) \bmod 8} \tag{10}$$

The global transformation $\mathbf{Lin}_B : \mathbf{F}^{128} \to \mathbf{F}^{128}$ is the block diagonal matrix with 16 blocks equal to L_B.

The $\mathcal{AES}$ $\mathcal{S}$-box constant $c_A = \mathtt{63} = t^6 + t^5 + t + 1 \in \mathbf{F}$ goes into:

$$\begin{aligned} \phi(c_A) \;=\; & (\mathtt{63}, \mathtt{C2}, \mathtt{35}, \mathtt{66}, \mathtt{D3}, \mathtt{2F}, \mathtt{39}, \mathtt{36}) = (t^6 + t^5 + t + 1, t^7 + t^6 + t, \\ & t^5 + t^4 + t^2 + 1, t^6 + t^5 + t^2 + t, t^7 + t^6 + t^4 + t + 1, \\ & t^5 + t^3 + t^2 + t + 1, t^5 + t^4 + t^3 + 1, t^5 + t^4 + t^2 + t) \end{aligned} \tag{11}$$

The corresponding $\mathcal{BES}$ vector $\mathbf{c}_B$ is obtained using sufficient copies

$$\mathbf{c}_B = \phi(\underbrace{c_A, \ldots, c_A}_{16}) = (\underbrace{\phi(c_A), \ldots, \phi(c_A)}_{16}) \qquad [\mathbf{c}_B]_i = [\phi(c_A)]_{i \bmod 8} \tag{12}$$

The $\mathcal{AES}$ **ShiftRows** may be represented by $R_A : \mathbf{F}^{16} \to \mathbf{F}^{16}$.

$$R_A = \left(\begin{array}{cccc|cccc|cccc|cccc} 1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1 \\ \hline 0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0 \\ 0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0 \\ \hline 0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0 \\ \hline 0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0 \end{array}\right) \tag{13}$$

"Expanding" each 1 in R_A with an identity matrix of order 8, I_8, and each 0 with a zero (8 × 8) matrix, we have $R_B : \mathbf{F}^{128} \rightarrow \mathbf{F}^{128}$.

The $\mathcal{AES}$ **MixColumns** may be represented by $C_A : \mathbf{F}^4 \rightarrow \mathbf{F}^4$:

$$C_A = \begin{pmatrix} t & t+1 & 1 & 1 \\ 1 & t & t+1 & 1 \\ 1 & 1 & t & t+1 \\ t+1 & 1 & 1 & t \end{pmatrix} \tag{14}$$

The $\mathcal{AES}$ transformation is given by the $\mathrm{Mix}_A : \mathbf{F}^{16} \rightarrow \mathbf{F}^{16}$ block diagonal matrix having as blocks four copies of C_A. In order to obtain the corresponding matrix we first need to compute $C_B^{(k)}$, for $k = 0, \ldots, 7$:

$$C_B^{(k)} = \begin{pmatrix} t^{2^k} & (t+1)^{2^k} & 1 & 1 \\ 1 & t^{2^k} & (t+1)^{2^k} & 1 \\ 1 & 1 & t^{2^k} & (t+1)^{2^k} \\ (t+1)^{2^k} & 1 & 1 & t^{2^k} \end{pmatrix} \tag{15}$$

where

$$\begin{array}{lll} t^{2^0} = t & t^{2^3} = t^4 + t^3 + t + 1 & t^{2^6} = t^6 + t^3 + t^2 + 1 \\ t^{2^1} = t^2 & t^{2^4} = t^6 + t^4 + t^3 + t^2 + t & t^{2^7} = t^7 + t^6 + t^5 + t^4 + t^3 + t \\ t^{2^2} = t^4 & t^{2^5} = t^7 + t^6 + t^5 + t^2 & \end{array} \tag{16}$$

from which $(t+1)^{2^k} = t^{2^k} + 1$ are immediately obtained.

In an appropriate basis, the resulting matrix $\mathrm{M}_B : \mathbf{F}^{128} \rightarrow \mathbf{F}^{128}$ is a block diagonal one, with four consecutive copies of $C_B^{(k)}$ for all possible k. The change of basis is necessary because of the different positioning of value powers in ϕ's image with respect to our needs. Indeed, if $\mathbf{a} \in \mathbf{F}^{16}$, then:

$$\phi(\mathbf{a}) = (a_0, ..., a_0^{2^7}, a_1, \ldots, a_1^{2^7}, \ldots, a_{15}, \ldots, a_{15}^{2^7}) \tag{17}$$

while to use the block diagonal representation, we would need:

$$\mathbf{a}' = (a_0, ..., a_{15}, a_0^2, \ldots, a_{15}^2, \ldots, a_0^{2^7}, \ldots, a_{15}^{2^7}) \tag{18}$$

This transformation is given by a permutation matrix $\mathrm{Perm}_B : \mathbf{F}^{128} \rightarrow \mathbf{F}^{128}$. To represent it easily, suppose to divide it into (16 × 8) sub-matrices P_{hk}, $h = 0, \ldots, 7, k = 0, \ldots, 15$. Each sub-matrix element (with $i = 0, \ldots, 15$, $j = 0, \ldots, 7$) is:

$$[P_{hk}]_{ij} = \begin{cases} 1 & \text{if } i = k \text{ and } j = h \\ 0 & \text{else} \end{cases} \tag{19}$$

Its inverse matrix $\mathbf{Perm}_B^{(-1)}$ is equally easy to describe: viewing it as composed of (8×16) sub-matrices $P_{hk}^{(-1)}$, with $h = 0, \ldots, 15, k = 0, \ldots, 7$, the generic element $[P_{hk}^{(-1)}]_{ij}$ (with $i = 0, \ldots, 7, j = 0, \ldots, 15$) is defined exactly as $[P_{hk}]_{ij}$ is. We have $\mathbf{Mix}_B = \mathbf{Perm}_B^{-1} \cdot M_B \cdot \mathbf{Perm}_B$.

We can avoid c_A slightly modifying the key generation scheme with respect to the original proposal. If $\mathbf{b}, (\mathbf{h}_B)_i \in \mathbf{B}$ are the state and key vectors for the generic i^{th} round of $\mathcal{BES}$, we have:

$$\begin{aligned} \mathbf{Round}_B(\mathbf{b}, (\mathbf{h}_B)_i) &= \mathbf{Mix}_B(R_B(\mathbf{Lin}_B(\mathbf{b}^{-1}) + \mathbf{c}_B)) + (\mathbf{h}_B)_i \\ &= M_B \cdot (\mathbf{b}^{-1}) + (C_B(\mathbf{c}_B) + (\mathbf{h}_B)_i) \\ &= M_B \cdot (\mathbf{b}^{-1}) + (\mathbf{k}_B)_i \end{aligned} \tag{20}$$

with

$$M_B = \mathbf{Mix}_B \cdot R_B \cdot \mathbf{Lin}_B\ , \quad C_B = \mathbf{Mix}_B \cdot R_B\ , \quad (\mathbf{k}_B)_i = C_B(\mathbf{c}_B) + (\mathbf{h}_B)_i \tag{21}$$

For the last round, being $\mathbf{Mix}_B$ absent, we have

$$(\mathbf{k}_B)_i = R_B(\mathbf{c}_B) + (\mathbf{h}_B)_i \tag{22}$$

but in this particular case we have $C_B(\mathbf{c}_B) = R_B(\mathbf{c}_B)$, and for what concerns this, we can avoid to distinguish the last round from the precedent ones. The change for key generation scheme is simply the addition of a constant vector to each obtained round key, and this will be the form of the system we will work with.

Now we analyze the $\mathcal{BES}$ translation for the key generation scheme.

- The $\mathcal{AES}$ **RotWord** operation is represented by $RW_A : \mathbf{F}^4 \to \mathbf{F}^4$.

$$RW_A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix} \tag{23}$$

  For the $\mathcal{BES}$ version $RW_B : \mathbf{F}^{32} \longrightarrow \mathbf{F}^{32}$, replace the 1's with I_8, and 0's with the (8×8) zero matrix.

- The $\mathcal{S}$-box is here applied only to a part of the whole vector, and therefore the matrix dimension changes. The resulting block diagonal matrix $\mathbf{Lin}_B^k : \mathbf{F}^{32} \longrightarrow \mathbf{F}^{32}$ has four blocks equal to L_B.

- The constant $\mathbf{c}_B^k$ is given by just four copies of $\phi(c_A)$:

$$\mathbf{c}_B^k = \phi(c_A, c_A, c_A, c_A) = (\phi(c_A), \ldots, \phi(c_A))\ , \quad [\mathbf{c}_B^k]_i = [\phi(c_A)]_{i \bmod 8} \tag{24}$$

- The constant vectors $\mathbf{Rcon}_i = (t^{i-1}, 0, 0, 0)$ are mapped into:

$$(\mathbf{Rcon}_B)_i = \phi(\mathbf{Rcon}_i) = (\phi(t^{r-1}), \underbrace{0, \ldots, 0}_{24}) \tag{25}$$

We keep using the matrix notation, but here in a *functional* sense. We *have* here to use constants. If $\varphi_B^i : \mathbf{F}^{32} \rightarrow \mathbf{F}^{32}$ is the $\mathcal{BES}$ i^{th}-round mapping function for a conjugated word $\mathbf{x}$:

$$\varphi_B^i(\mathbf{x}) = \text{Lin}_B^k(RW_B(\mathbf{x}))^{-1} + \mathbf{c}_B^k + (\mathbf{Rcon}_B)_i \tag{26}$$

the generic $\mathcal{AES}$ and $\mathcal{BES}$ key round matrices are MK_A^i and MK_B^i :

$$MK_A^i = \begin{pmatrix} 0 & 0 & 0 & \varphi_A^i \\ 0 & I_4 & 0 & \varphi_A^i \\ 0 & I_4 & I_4 & \varphi_A^i \\ 0 & I_4 & I_4 & I_4 + \varphi_A^i \end{pmatrix}, \; MK_B^i = \begin{pmatrix} 0 & 0 & 0 & \varphi_B^i \\ 0 & I_{32} & 0 & \varphi_B^i \\ 0 & I_{32} & I_{32} & \varphi_B^i \\ 0 & I_{32} & I_{32} & I_{32} + \varphi_B^i \end{pmatrix} \tag{27}$$

A key round is the computation of $\mathbf{h}_i = MK_B^i(\mathbf{h}_{r-1})$.

3. Polynomial Systems

We show how encryption and key generation can be represented by algebraic systems. All variables satisfy the **F**-belonging equation $y^{256} + y = 0$.

3.1 Encryption

Remembering that the last round differs slightly from the other ones, with $M_B^* = R_B \cdot \text{Lin}_B$, the system for codification is [MR] :

$$\begin{cases} \mathbf{w}_0 = \mathbf{p} + \mathbf{k}_0 & \\ \mathbf{x}_i = \mathbf{w}_i{}^{-1} & i = 0, ..., 9 \\ \mathbf{w}_i = M_B \mathbf{x}_{i-1} + \mathbf{k}_i & i = 1, ..., 9 \\ \mathbf{c} = M_B^* \mathbf{x}_9 + \mathbf{k}_{10} & \end{cases} \tag{28}$$

Let (j, m) indicate the $(8j + m)^{\text{th}}$ component of all the vectors, for $j = 0, \ldots, 15$ and $m = 0, \ldots, 7$. If no 0-inversion occurs (true for the 53% of encryptions and 85% of 128-bit keys), it is possible to expand the system as follows, for all possible values of j and m

$$\begin{cases} 0 = w_{0,(j,m)} + p_{(j,m)} + k_{0,(j,m)} & \\ 0 = x_{i,(j,m)} w_{i,(j,m)} + 1 & i = 0, \ldots, 9 \\ 0 = w_{i,(j,m)} + (M_B \mathbf{x}_{i-1})_{(j,m)} + k_{i,(j,m)} & i = 1, \ldots, 9 \\ 0 = c_{(j,m)} + (M_B^* \mathbf{x}_9)_{(j,m)} + k_{10,(j,m)} & \end{cases} \tag{29}$$

Let $\alpha, \beta \in \mathbf{F}$ indicate respectively M_B and M_B^* entries. Everything must be valid for $\mathbf{B_A}$, therefore we have (with $m+1$ considered mod 8)

$$S = \begin{cases} 0 = w_{0,(j,m)} + p_{(j,m)} + k_{0,(j,m)} & \\ 0 = w_{i,(j,m)} + k_{i,(j,m)} + \sum_{(j',m')} \alpha_{(j,m),(j',m')} x_{i-1,(j',m')} & i = 1,\ldots,9 \\ 0 = c_{(j,m)} + k_{10,(j,m)} + \sum_{(j',m')} \beta_{(j,m),(j',m')} x_{9,(j',m')} & \\ & \\ 0 = x_{i,(j,m)} w_{i,(j,m)} + 1 & i = 0,\ldots,9 \\ 0 = x_{i,(j,m)}^2 + x_{i,(j,m+1)} & i = 0,\ldots,9 \\ 0 = w_{i,(j,m)}^2 + w_{i,(j,m+1)} & i = 0,\ldots,9 \end{cases} \tag{30}$$

Let S_ℓ, $\ell = 1,\ldots,6$ be the equations in the ℓ^{th} line of the system for all values of i, j and m, and I_ℓ the ideal they generate. As we see, the system is very sparse, with $S' = \{S_1, S_2, S_3\}$ linear, and the other equations in $S'' = \{S_4, S_5, S_6\}$ quadratic. If $\mathbf{k} = \{\mathbf{k}_i\}$, $\mathbf{w} = \{\mathbf{w}_i\}$, $\mathbf{x} = \{\mathbf{x}_i\}$, we have

Line	*Number of equations*
S_1	$16 \cdot 8 =$ 128
S_2	$9 \cdot 16 \cdot 8 =$ 1152
S_3	$16 \cdot 8 =$ 128
S_4	$10 \cdot 16 \cdot 8 =$ 1280
S_5	$10 \cdot 16 \cdot 8 =$ 1280
S_6	$10 \cdot 16 \cdot 8 =$ 1280
S	**Total** = 5248

Block	*Number of variables*
$\mathbf{k}$	$11 \cdot 16 \cdot 8 =$ 1408
$\mathbf{x}$	$10 \cdot 16 \cdot 8 =$ 1280
$\mathbf{w}$	$10 \cdot 16 \cdot 8 =$ 1280
	Total = 3968

3.2 Key Generation

There is an analogous system for key generation. The equations express all the $h_{i,(j,m)}$ variables in term of the $h_{0,(j,m)}$ ones. The index ranges for the equations are: $i = 1,\ldots,10$, $\tilde{\jmath}, \tilde{\jmath}' = 0,\ldots,3$ and $m, m' = 0,\ldots,7$, and γ are the $\mathbf{Lin}_B^k$ matrix coefficients.

$$\mathcal{K}_B = \begin{cases} \tilde{\mathbf{H}}_{i0} = \varphi_B^i(\tilde{\mathbf{H}}_{i-1,3}) \\ \tilde{\mathbf{H}}_{i1} = \tilde{\mathbf{H}}_{i0} + \tilde{\mathbf{H}}_{i-1,1} \\ \tilde{\mathbf{H}}_{i2} = \tilde{\mathbf{H}}_{i1} + \tilde{\mathbf{H}}_{i-1,2} \\ \tilde{\mathbf{H}}_{i3} = \tilde{\mathbf{H}}_{i2} + \tilde{\mathbf{H}}_{i-1,3} \end{cases} = \tag{31}$$

$$= \begin{cases} z_{i,(\tilde{\jmath},m)} = h_{i-1,(12+[(\tilde{\jmath}+1) \bmod 4],m)}^{254} \\ h_{i,(\tilde{\jmath},m)} = (\mathbf{c}_B^k + (\mathbf{Rcon}_B)_i)_{(\tilde{\jmath},m)} + \sum_{(\tilde{\jmath}',m')} \gamma_{(\tilde{\jmath},m)(\tilde{\jmath}',m')} z_{i,(\tilde{\jmath}',m')} \\ h_{i,(4s+\tilde{\jmath},m)} = h_{i,(4(s-1)+\tilde{\jmath},m)} + h_{i-1,(4s+\tilde{\jmath},m)} \qquad s = 1,2,3 \end{cases}$$

Let $\mathbf{cR}_i = \mathbf{c}_B^k + (\mathbf{Rcon}_B)_i$ be the vector in each round, and its components δ_i. Thanks to the third equivalence of (21), with $t = 0, \ldots, 15$ and the conjugation property, we have:

$$K = \begin{cases} 0 = z_{i,(j,m)} + h^{254}_{i-1,(12+[(j+1) \bmod 4],m)} \\ 0 = h_{i,(j,m)} + \delta_{i,(j,m)} + \sum_{(j',m')} \gamma_{(j,m)(j',m')} z_{i,(j',m')} \\ 0 = h_{i,(4s+j,m)} + h_{i,(4(s-1)+j,m)} + h_{i-1,(4s+j,m)} \qquad s = 1,2,3 \\ 0 = k_{i,(t,m)} + (C_B(\mathbf{c_B}))_{(t,m)} + h_{i,(t,m)} \\ 0 = z^2_{i,(j,m)} + z_{i,(j,m+1)} \\ 0 = h^2_{i,(j,m)} + h_{i,(j,m+1)} \end{cases} \tag{32}$$

4. Resolution

We are interested in obtaining the key out of the systems S and K, that is the original key $\mathbf{h} = \phi^{-1}(\mathbf{k}^\star) = \{h_0, \ldots, h_{15}\}$, where $\mathbf{k}^\star = \{k_{0,(0,m)}, \ldots, k_{0,(15,m)}\}$.

In order to obtain relations among $\mathbf{h}$ ($\mathbf{k}$) components we eliminate all other variables. We do this:

- modifying the way the systems are presented,
- doing some "hand" substitutions, and finally
- performing Gröbner bases computations (more complicated substitutions, expansions and simplifications) to obtain the final systems.

Note that, for each variable $v \in \mathbf{k}, \mathbf{w}, \mathbf{z}, \mathbf{h}$, the conjugation property may be synthesized by the obvious following relations:

$$v_{i,(j,m)} = v^{2^m}_{i,(j,0)} \qquad m = 0, \ldots, 7 \tag{33}$$

4.1 Encryption

We rewrite S: first of all, we remove the imposed restriction about inversion, substituting S_4 with an equation expressing the true definition of the general inversion in $\mathbf{F}$. Then we use (33), to remove all the

variables with index $m > 0$, obtaining:

$$S^\star = \begin{cases} 0 = w^{2^m}_{0,(j,0)} + p^{2^m}_{(j,0)} + k^{2^m}_{0,(j,0)} & \\ 0 = w^{2^m}_{i,(j,0)} + k^{2^m}_{i,(j,0)} + \sum_{(j',m')} \alpha_{(j,m),(j',m')} x^{2^{m'}}_{i-1,(j',0)} & i = 1,\ldots,9 \\ 0 = c^{2^m}_{(j,0)} + k^{2^m}_{10,(j,0)} + \sum_{(j',m')} \beta_{(j,m),(j',m')} x^{2^{m'}}_{9,(j',0)} & \\ 0 = x_{i,(j,0)} + w^{254}_{i,(j,0)} & i = 0,\ldots,9 \end{cases} \tag{34}$$

With the last equation we can remove all the $x_{i,(j,0)}$, and, being each line a set of successive square powers, we keep only the ones with $m = 0$:

$$S^\star = \begin{cases} 0 = w_{0,(j,0)} + p_{(j,0)} + k_{0,(j,0)} & \\ 0 = w_{i,(j,0)} + k_{i,(j,0)} + \sum_{(j',m')} \alpha_{(j,0),(j',m')} w^{254\cdot 2^{m'}}_{i-1,(j',0)} & i = 1,\ldots,9 \\ 0 = c_{(j,0)} + k_{10,(j,0)} + \sum_{(j',m')} \beta_{(j,0),(j',m')} w^{254\cdot 2^{m'}}_{9,(j',0)} & \end{cases} \tag{35}$$

We note that the β coefficients do not depend on j and j', and the values are simply the coefficients of f. To simplify notations even more, we take, mod 255:

$$\omega = (\omega_i) = 254 \cdot (2^0,\ldots,2^7) = (254, 253, 251, 247, 239, 223, 191, 127)\ ,$$
$$\omega' = (\omega'_i) = (\omega_0 - 127,\ldots,\omega_7 - 127) = (127, 126, 124, 120, 112, 96, 64, 0)$$

We can now avoid writing m index:

$$S^\star = \begin{cases} 0 = w_{0,j} + p_j + k_{0,j} & \\ 0 = w_{i,j} + k_{i,j} + \sum_{(j',m')} \alpha_{(j,0),(j',m')} w^{\omega_{m'}}_{i-1,j'} & i = 1,\ldots,9 \\ 0 = k_{10,j} + c_j + \sum_{m'} \lambda_{m'}\, w^{\omega_{m'}}_{9,j'} & \end{cases} \tag{36}$$

The system has 16+9·16+16 = 176 equations in 11·16+10·16 = 336 variables. Obviously, it expresses nothing but a series of successive substitutions, down to the last equation. Considering a block lexicographic (lex) order for which

$$\mathbf{k}_{10} > \mathbf{w}_9 > \mathbf{k}_9 > \cdots > \mathbf{w}_0 > \mathbf{k}_0 \tag{37}$$

we have a (not reduced) Gröbner basis [CLO], and the substitutions may be considered as the complete reduction computation. The resulting set

of the last 16 equations, where all the **w** variables are no more present, is what we are looking for. If q_j^S are the resulting polynomials, we have:

$$k_{10,j} + c_j + q_j^S(\mathbf{k}_0, \ldots, \mathbf{k}_9, p) = 0 \qquad j = 0, \ldots, 15 \qquad (38)$$

4.2 Key Generation

We get more informations analyzing K. We

- substitute **z** variables in the second line equations.
- use the conjugation property,
- note that $C_B(\mathbf{c_B})$ has $c_A = t^6 + t^5 + t + 1$ in the $(j, 0)$ positions, and opportune powers in the other ones. This means that the equations on the fourth line of K, K_4, may be reduced (the other ones being powers of it) to:

$$h_{i,(j,0)} + k_{i,(j,0)} + c_A = 0 \qquad (39)$$

- for the above considerations, express everything directly in term of **k** variables.
- observe that $\mathbf{Lin}_B^k$ is a block diagonal matrix, and therefore just $\tilde{j}' = \tilde{j}$ is "active" for each single equation, and what remains is nothing more than the set of coefficients of the f polynomial.

We define in : $\mathbf{N} \ni n \rightarrow \mathrm{in}(n) = 12 + [(n+1) \bmod 4] \in \mathbf{N}$. After the elaboration, always remembering the **F**-belonging equation, we have the following system (where $i = 1, \ldots, 10$; $s, \tilde{j} = 0, \ldots, 3$ and in the last version we omit m)

$$K^* = \begin{cases} 0 = h_{i,(\tilde{j},0)} + \delta_{i,(\tilde{j},0)} + \sum\limits_{(\tilde{j}',m')} \gamma_{(\tilde{j},0)(\tilde{j}',m')} h_{i-1,(\mathrm{in}(\tilde{j}'),0)}^{254 \cdot 2^{m'}} \\ 0 = h_{i,(4s+\tilde{j},0)} + h_{i,(4(s-1)+\tilde{j},0)} + h_{i-1,(4s+\tilde{j},0)} \\ 0 = h_{i,(\tilde{j},0)} + (k_{i,(\tilde{j},0)} + c_A) \end{cases} =$$

$$\begin{cases} 0 = (k_{i,(\tilde{j},0)} + c_A) + \delta_{i,(\tilde{j},0)} + \sum\limits_{m'} \gamma_{(\tilde{j},0)(\tilde{j},m')} (k_{i-1,(\mathrm{in}(\tilde{j}),0)} + c_A)^{\omega_{m'}} \\ 0 = (k_{i,(4s+\tilde{j},0)} + c_A) + (k_{i,(4(s-1)+\tilde{j},0)} + c_A) + (k_{i-1,(4s+\tilde{j},0)} + c_A) \end{cases} =$$

$$\begin{cases} 0 = k_{i,\tilde{j}} + (c_A + \delta_{i,(\tilde{j},0)}) + (k_{i-1,\mathrm{in}(\tilde{j})} + c_A)^{127} \left(\sum\limits_{m'} \lambda_{m'} (k_{i-1,\mathrm{in}(\tilde{j})} + c_A)^{\omega'_{m'}} \right) \\ 0 = k_{i,4s+\tilde{j}} + k_{i,4(s-1)+\tilde{j}} + k_{i-1,4s+\tilde{j}} + c_A \end{cases}$$

Only **k** variables remain, 160 equations in 176 variables, and by successive substitutions we can express all the ones with $i > 0$ as polynomials

in the "parameters" $\mathbf{k_0}$. The equations are a Gröbner basis for several suitable lex orderings. We may obtain its complete reduction using, e.g.

$$k_{10,15} > \ldots > k_{10,0} > \cdots > k_{0,15} > \ldots > k_{0,0} \tag{40}$$

It is possible to work with $\mathbf{h}$ variables to obtain the equations following the original $\mathcal{AES}$ definition, and use (39) only at the end, in order to obtain the modified key generation scheme. In any case, the result is:

$$k_{i,j} = q^K_{i,j}(\mathbf{k_0}) \qquad i = 1, \ldots, 10 \quad , \quad j = 0, \ldots, 15 \tag{41}$$

In the final phase we merge the results. There are two possibilities, according to how many (p, c) pairs (related by the same key) are known.

One (p, c) pair : We eliminate all intermediate keys, putting together the systems $S^\star$ and $K^\star$, refining (37) with (40). We obtain the entire substitution process once and for all, summarized as follows:

$$\mathcal{C}_1 = \{\ q^K_{10,j}(\mathbf{k_0}) + c_j + q^S_j(\mathbf{k_0}, q^K_{1,j}(\mathbf{k_0}), \ldots, q^K_{9,j}(\mathbf{k_0}), p) = 0 \mid \ j = 0, \ldots, 15\ \} \tag{42}$$

a system of 16 equations in 16 variables, having as roots the desired keys.

More than 10 (p, c) pairs : We use a copy of (38) for each (p, c) pair, to obtain a system in 176 variables with at least 176 equations, whose roots give *all* the keys.

$$\mathcal{C}_2 = \{\ k_{10,j} + c^{(n)}_j + q^S_j(\mathbf{k_0}, \ldots, \mathbf{k_9}, p^{(n)}) = 0 \mid \ n = 1, \ldots, d \quad , \quad j = 0, \ldots, 15\ \} \tag{43}$$

These systems are dense, it is very difficult to write them explicitly, and even more to solve them. Using more than 11 (p, c) the $\mathcal{C}_2$ system becomes overdetermined.

5. Conclusions

$K^\star$ and most of $S^\star$ are invariant for all choices of keys. Actually, the only varying parts of $S^\star$ are the constant terms of the equations 1 to 16, and 161 to 176. Besides, for the equations 1 to 16, it can be chosen, too, if convenient.

When extended, the joint size of $K^\star$ and $S^\star$ is of about 500 Kb. Each of them is a (not reduced) Gröbner basis for several lex orderings, their union is not. Probably there exists some ordering for which the calculus

of a Gröbner basis is easier. If we ever can obtain this with reasonable computational resources, then $\mathcal{AES}$ can be declared broken.

Succeeding to calculate the Hilbert series of $K^\star \cup S^\star$, we should easily obtain the number n_s of its solutions. We suspect that n_s is invariant for all key and (p, c) choices. Furthermore, we expect that n_s expresses the redundancy of the keyspace of $\mathcal{AES}$. That is, it tells us how many key choices will set up the same bijection between the cleartext space and ciphertext space. The number of such bijections is expected to be:

$$\frac{\#(\mathcal{AES}\text{ Keyspace})}{n_s} \tag{44}$$

Probably a reasonably simple canonical representation of such bijections can be found. In this case, if n_s is big enough, probably the right (unique up to the isomorphism) key can be found by means of an exhaustive search.

References

D. A. Cox, J. Little, D. O'Shea. Ideals, Varieties, and Algorithms, An Introduction to Computational Algebraic Geometry and Commutative Algebra. Springer-Verlag, New York, *1992.*

N. Courtois, J. Pieprzyk. *Cryptanalysis of block ciphers with overdefined systems of equations.* IACR eprint server `www.iacr.org`, *2002.*

J. Daemen, V. Rijmen. *AES proposal: Rijndael (Version 2).* NIST AES website: `http://csrc.nist.gov/encryption/aes`, *1999.*

J. Daemen, V. Rijmen. *The design of Rijndael: AES - The Advanced Encryption Standard.* Springer-Verlag, *2002.*

National Institute of Standards and Technology. Advanced Encryption Standard. FIPS *197. 26* November *2001.*

N. Ferguson, R. Schroeppel, D. Whiting. A simple algebraic representation of Rijndael. In *Selected Areas in Cryptography,* Proc. SAC *2001,* Lecture Notes in Computer Science *2259,* pp. *103-111,* Springer Verlag, *2001.*

G.-M. Greuel, G. Pfister, H. Schönemann. SINGULAR *2-0-3.* A Computer Algebra System for Polynomial Computations. Center for Computer Algebra, University of Kaiserslautern, *2003.* `www.singular.uni-kl.de`.

S. Murphy, M.J.B. Robshaw. *Essential Algebraic Structure within the AES.* M. Yung (ed.): CRYPTO *2002,* LNCS *2242,* pp. *1-16,* Springer-Verlag *2002.*

E. Oswald, J. Daemen, and V. Rijmen. *The State of the Art of Rijndael's Security.* Technical report. `www.a-sit.at/technologieb/evaluation/aes_report_e.pdf`

D. R. Stinson. CRYPTOGRAPHY, *Theory and Practice.* Chapman & Hall/CRC, *2002.* Second edition.