Small Scale Variants of the AES

C. Cid*, S. Murphy, and M.J.B. Robshaw

Information Security Group,
Royal Holloway, University of London,
Egham, Surrey, TW20 0EX, U.K.
{Carlos.Cid,S.Murphy,M.Robshaw}@rhul.ac.uk

Abstract. In this paper we define small scale variants of the AES. These variants inherit the design features of the AES and provide a suitable framework for comparing different cryptanalytic methods. In particular, we provide some preliminary results and insights when using off-the-shelf computational algebra techniques to solve the systems of equations arising from these small scale variants.

1 Introduction

The potential for algebraic attacks [1, 2, 8] on the AES [4, 11] has been the source of recent speculation. Two important (and complementary) approaches to the algebraic analysis of the AES were provided in [2] and [8]. In [2] it was shown how recovering an AES encryption key could be viewed as solving a set of sparse overdefined multivariate quadratic equations over $GF(2)$ and a method—the XSL method—for solving this set of AES-specific equations was proposed. In [8] the AES was embedded in a related cipher (called the BES) and it was shown how recovering an AES encryption key could be viewed as solving a similar set of equations over $GF(2^8)$. The highly structured equation systems that result from this approach may well be more tractable than those arising from a $GF(2)$ perspective [8, 9].

Currently, however, it is unknown whether the XSL—or any other proposed method of solution—works on the AES. For most types of cryptanalysis it is straightforward to perform experiments on reduced versions of the cipher to understand how an attack might perform. However this is not so easy for the AES, and while some experiments have been conducted [2], the equation systems used were very different from those that might actually arise from the AES.

With this goal in mind, we specify a family of small scale variants of the AES. Previous variants have been used before as an educational tool [10, 12], but our aim is to provide a fully parameterised framework for the analysis of AES equation systems. We describe how to construct the equation systems corresponding to these small scale variants of the AES, and give an example of such a system derived using the BES-style embedding. We report on some preliminary analysis of a number of small scale variants and provide the first experimental insight into the behaviour of algebraic cryptanalysis on AES-like ciphers.

* This author was supported by EPSRC Grant GR/S42637.

2 Small Variants of the AES

We define two sets of small scale variants of the AES; these differ in the form of the final round. These two sets of variants will be denoted by $\mathrm{SR}(n, r, c, e)$ and $\mathrm{SR}^*(n, r, c, e)$.

2.1 Small Scale AES Parameters

Both $\mathrm{SR}(n, r, c, e)$ and $\mathrm{SR}^*(n, r, c, e)$ are parameterised in the following way:

- n is the number of (encryption) rounds;
- r is the number of "rows" in the rectangular arrangement of the input;
- c is the number of "columns" in the rectangular arrangement of the input;
- e is the size (in bits) of a word.

$\mathrm{SR}(n, r, c, e)$ and $\mathrm{SR}^*(n, r, c, e)$ both have n rounds and a block size of rce bits, where a data block is viewed as an array of $(r \times c)$ "words" of e bits. We will see that the full AES is equivalent to $\mathrm{SR}^*(10, 4, 4, 8)$.

Number of Rounds n. The AES is an iterated block cipher consisting of 10 rounds. The typical round uses four different operations. The small scale variants $\mathrm{SR}(n, r, c, e)$ and $\mathrm{SR}^*(n, r, c, e)$ consist of n rounds, with $1 \leq n \leq 10$, using small scale variants of these operations. These operations are specified in Section 2.2.

Data Block Array Size $(r \times c)$. Each element of the data array is a word of size e bits. The array itself has r rows and c columns. We consider small scale variants of the AES with both r and c restricted to $1, 2,$ or 4. Some examples are given below. Note that we adopt the AES-style of numbering "words" within an array and work by column first.

$$\begin{array}{|c|}\hline 0\\\hline\end{array}\qquad \begin{array}{|c|}\hline 0\\\hline 1\\\hline\end{array}\qquad \begin{array}{|c|c|}\hline 0&2\\\hline 1&3\\\hline\end{array}\qquad \begin{array}{|c|c|}\hline 0&4\\\hline 1&5\\\hline 2&6\\\hline 3&7\\\hline\end{array}\qquad \begin{array}{|c|c|c|c|}\hline 0&2&4&6\\\hline 1&3&5&7\\\hline\end{array}\qquad \begin{array}{|c|c|c|c|}\hline 0&4&8&12\\\hline 1&5&9&13\\\hline 2&6&10&14\\\hline 3&7&11&15\\\hline\end{array}$$

Word Size e. We define small scale variants of the AES for word sizes $e = 4$ and $e = 8$. It is natural within the context of the AES to regard a word of size e as an element of the field $GF(2^e)$. Thus we define small scale variants of the AES with respect to the two fields $GF(2^4)$ and $GF(2^8)$.

The small scale variants $\mathrm{SR}(n, r, c, 4)$ and $\mathrm{SR}^*(n, r, c, 4)$ use the field $GF(2^4)$. We use the primitive polynomial $X^4 + X + 1$ over $GF(2)$ to define this field. We let ρ be a root of this polynomial, so

$$GF(2^4) = \frac{GF(2)[X]}{(X^4 + X + 1)} = GF(2)(\rho).$$

When referring to elements of $GF(2^4)$, we sometimes use hexadecimal notation, so that $\mathtt{D} = \rho^3 + \rho^2 + 1$ and so on.

The small scale variants of the AES with word size 8, $\mathrm{SR}(n, r, c, 8)$ and $\mathrm{SR}^*(n, r, c, 8)$, use the field $GF(2^8)$. The Rijndael polynomial $X^8 + X^4 + X^3 + X + 1$ over $GF(2)$ is used to define this field. We let θ be a root of this polynomial, so

$$GF(2^8) = \frac{GF(2)[X]}{(X^8 + X^4 + X^3 + X + 1)} = GF(2)(\theta).$$

When referring to elements of $GF(2^8)$, we again sometimes use hexadecimal notation, so that $\mathtt{D1} = \theta^7 + \theta^6 + \theta^4 + 1$ and so on.

2.2 Small Scale Round Operations

Each round of the AES consists of some combination of the following operations:

1. `SubBytes`
2. `ShiftRows`
3. `MixColumns`
4. `AddRoundKey`

A round of the small scale variants of the AES consists of small scale variants of these operations. For the last round of the AES, the operation `MixColumns` is omitted. Similarly, for $\mathrm{SR}^*(n, r, c, e)$ the final round does not use `MixColumns`, whereas `MixColumns` is retained for the final round of $\mathrm{SR}(n, r, c, e)$. The AES is thus identical to $\mathrm{SR}^*(10, 4, 4, 8)$.

Note that the two ciphertexts produced by $\mathrm{SR}(n, r, c, e)$ and $\mathrm{SR}^*(n, r, c, e)$ when encrypting the same plaintext under the same key are related by an affine mapping. A solution of the system of equations for one cipher would immediately give a solution for the other and so, without loss of generality, we only consider $\mathrm{SR}(n, r, c, e)$ for the remainder of this paper.

`SubBytes`. The operation `SubBytes` uses an S-Box and is defined to be the simultaneous application of the S-Box to each element of the data array. For the small scale variants $\mathrm{SR}(n, r, c, 4)$ based on the field $GF(2^4)$, we define the S-Box by analogy with the AES. Thus this S-box consists of the following three (sequential) operations.

1. *Inversion.* The first operation of the S-Box is an inversion in the field $GF(2^4)$ (with $0 \mapsto 0$), using the representation defined in Section 2.1. The look-up table for this inversion map is given below.

Inversion in $GF(2^4)$																
Input	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Output	0	1	9	E	D	B	7	6	F	2	C	5	A	4	3	8

2. *GF*(2)*-linear map.* The output of the inversion is the input to a $GF(2)$-linear map. This $GF(2)$-linear map is given by (the pre-multiplication by) the circulant $GF(2)$-matrix

$$\begin{pmatrix} 1\,1\,1\,0 \\ 0\,1\,1\,1 \\ 1\,0\,1\,1 \\ 1\,1\,0\,1 \end{pmatrix}$$

with respect to the "FIPS component ordering" [11]. The look-up table for the $GF(2)$-linear map is given below.

$GF(2)$-linear map in $GF(2^4)$																
Input	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Output	0	D	B	6	7	A	C	1	E	3	5	8	9	4	2	F

Note that this $GF(2)$-linear map can also be expressed as the linearised polynomial $f(X) = \lambda_0 X^{2^0} + \lambda_1 X^{2^1} + \lambda_2 X^{2^2} + \lambda_3 X^{2^3}$, where $(\lambda_0, \lambda_1, \lambda_2, \lambda_3) = (\mathtt{5}, \mathtt{1}, \mathtt{C}, \mathtt{5})$. Thus we have

$$\begin{aligned} f(X) &= (\rho^2+1)X + X^2 + (\rho^3+\rho^2)X^4 + (\rho^2+1)X^8 \\ &= \mathtt{5}X + \mathtt{1}X^2 + \mathtt{C}X^4 + \mathtt{5}X^8 \end{aligned}$$

3. *S-Box constant.* The S-Box constant 6 (or equivalently $\rho^2+\rho$) is added (as an element of $GF(2^4)$) to the output of the $GF(2)$-linear map. This result is the output of the S-Box.

The look-up table for the entire S-Box is given below.

S-Box over $GF(2^4)$																
Input	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Output	6	B	5	4	2	E	7	A	9	D	F	C	3	1	0	8

For the small scale variants SR$(n, r, c, 8)$ we use the AES S-Box. The values of the S-box operation over $GF(2^8)$ are available in the AES specification [11].

S-Box Summary	$GF(2^4)$	$GF(2^8)$
Irreducible polynomial	X^4+X+1	$X^8+X^4+X^3+X+1$
GF(2)*-linear map*	$\begin{pmatrix} 1\,1\,1\,0 \\ 0\,1\,1\,1 \\ 1\,0\,1\,1 \\ 1\,1\,0\,1 \end{pmatrix}$	$\begin{pmatrix} 1\,0\,0\,0\,1\,1\,1\,1 \\ 1\,1\,0\,0\,0\,1\,1\,1 \\ 1\,1\,1\,0\,0\,0\,1\,1 \\ 1\,1\,1\,1\,0\,0\,0\,1 \\ 1\,1\,1\,1\,1\,0\,0\,0 \\ 0\,1\,1\,1\,1\,1\,0\,0 \\ 0\,0\,1\,1\,1\,1\,1\,0 \\ 0\,0\,0\,1\,1\,1\,1\,1 \end{pmatrix}$
Constant	6	63

`ShiftRows`. The `ShiftRows` operation is defined to be the simultaneous (left) rotation of the row i of the data array, $0 \le i \le r-1$, by i positions. This is independent of the number of columns and the top row is fixed by this operation.

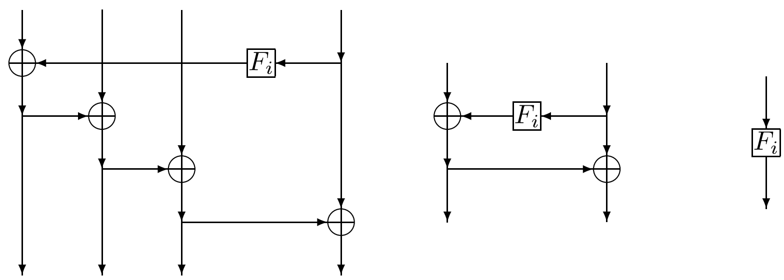
`MixColumns`. The `MixColumns` operation pre-multiplies each column of the data array by an invertible circulant $GF(2^e)$-matrix with row (and column) sum 1. These matrices are all MDS matrices (see [4]) and the choice of matrix in a small scale variant depends on the number of rows in the data array.

Number of Rows	$GF(2^4)$	$GF(2^8)$
$r = 1$	$\begin{pmatrix} 1 \end{pmatrix}$	$\begin{pmatrix} 1 \end{pmatrix}$
$r = 2$	$\begin{pmatrix} \rho+1 & \rho \\ \rho & \rho+1 \end{pmatrix}$	$\begin{pmatrix} \theta+1 & \theta \\ \theta & \theta+1 \end{pmatrix}$
$r = 4$	$\begin{pmatrix} \rho & \rho+1 & 1 & 1 \\ 1 & \rho & \rho+1 & 1 \\ 1 & 1 & \rho & \rho+1 \\ \rho+1 & 1 & 1 & \rho \end{pmatrix}$	$\begin{pmatrix} \theta & \theta+1 & 1 & 1 \\ 1 & \theta & \theta+1 & 1 \\ 1 & 1 & \theta & \theta+1 \\ \theta+1 & 1 & 1 & \theta \end{pmatrix}$

`AddRoundKey`. The key schedule (described in Section 2.3) for an n-round small scale variant of the AES produces $n + 1$ subkey blocks. `AddRoundKey` simultaneously adds (as elements of $GF(2^e)$) each element of the subkey block to some intermediate data block. Since the AES begins with an initial `AddRoundKey`, the small scale variants $\mathrm{SR}(n, r, c, e)$ also begin with this operation.

2.3 Small Scale Key Schedule

The structure of one round of the AES key schedule is illustrated below left (we only consider equal block and key sizes). Each vertical line represents one column of bytes, and F_i is the non-linear key schedule function for round i applied to one column of the array holding the previous round key. The function F_i consists of the application of the AES S-Box to all r components of the column, along with a word-based rotation and addition of a constant. When considering small scale analogues, we use this standard key schedule structure for four columns ($c = 4$). For two columns or one column ($c = 2$ or $c = 1$) we use the key schedule structures given by the diagrams below center and below right respectively.



Using this framework we can define key schedules for the small scale variants. $\mathrm{SR}(n, r, c, e)$ has a user-provided key of size rce bits, which is considered to be an array of $(r \times c)$ e-bits words. This key forms the initial subkey. Each subkey is then used to define the succeeding subkeys. We provide a full description of the small scale key schedules in Appendix A.

3 Multivariate Quadratic Equation Systems

The existence of a sparse multivariate quadratic equation system over $GF(2^8)$ for an AES encryption was shown by defining a new block cipher, the *Big Encryption System (BES)*, as well as a "BES"-embedding of the AES [8]. The main idea of the BES-embedding is to use the vector conjugate mapping ϕ to embed the AES into the larger cipher BES [8].

This technique can be used to derive sparse multivariate quadratic equation systems over $GF(2^e)$ for the small scale variants of the AES. This is based on the vector conjugate mapping ϕ for $GF(2^4)$ defined by

$$\phi(a) = \left(a^{2^0}, a^{2^1}, a^{2^2}, a^{2^3}\right)^T .$$

Any element $a \in GF(2^4)$ can be embedded as an element $(a, a^2, a^4, a^8)^T \in GF(2^4)^4$ under ϕ. We now describe how operations on a data block $\mathbf{a}$ in small scale variants $\mathrm{SR}(n, r, c, 4)$ can be replicated by operations on the vector conjugate $\phi(\mathbf{a})$. Further details and the justification for these operations are given in [8].

`SubBytes`. Consider the three component operations of the S-box separately.

1. *Inversion.* The operation of inversion can be replicated by the componentwise inversion of the vector conjugate.
2. *$GF(2)$-linear map.* The effect of the $GF(2)$-linear map can by replicated by pre-multiplying the (column) vector conjugate by the matrix

$$\begin{pmatrix} \lambda_0 & \lambda_1 & \lambda_2 & \lambda_3 \\ \lambda_3^2 & \lambda_0^2 & \lambda_1^2 & \lambda_2^2 \\ \lambda_2^4 & \lambda_3^4 & \lambda_0^4 & \lambda_1^4 \\ \lambda_1^8 & \lambda_2^8 & \lambda_3^8 & \lambda_0^8 \end{pmatrix} = \begin{pmatrix} \mathtt{5} & \mathtt{1} & \mathtt{C} & \mathtt{5} \\ \mathtt{2} & \mathtt{2} & \mathtt{1} & \mathtt{F} \\ \mathtt{A} & \mathtt{4} & \mathtt{4} & \mathtt{1} \\ \mathtt{1} & \mathtt{8} & \mathtt{3} & \mathtt{3} \end{pmatrix},$$

   where $(\lambda_0, \lambda_1, \lambda_2, \lambda_3)$ are the coefficients of the linearised polynomial given previously.
3. *S-Box constant.* The effect of adding $\mathtt{6}$ to a data array element can be replicated by adding $(\mathtt{6}, \mathtt{7}, \mathtt{6}, \mathtt{7})^T$ to its vector conjugate.

`ShiftRows`. The effect of `ShiftRows` on the conjugate embedding can be easily replicated for the small variants $\mathrm{SR}(n, r, c, 4)$. For example, the operation on $\mathrm{SR}(n, 1, 1, 4)$, $\mathrm{SR}(n, 2, 1, 4)$ and $\mathrm{SR}(n, 2, 2, 4)$ is given by the following matrices, where I_4 denotes the 4×4 identity matrix over $GF(2^4)$:

$$\begin{pmatrix} I_4 \end{pmatrix}, \quad \begin{pmatrix} I_4 & 0 \\ 0 & I_4 \end{pmatrix}, \quad \text{and} \quad \begin{pmatrix} I_4 & 0 & 0 & 0 \\ 0 & 0 & 0 & I_4 \\ 0 & 0 & I_4 & 0 \\ 0 & I_4 & 0 & 0 \end{pmatrix}.$$

`MixColumns.` The effect of multiplying a field element by some other field element z can be replicated by pre-multiplying its (column) vector conjugate by the diagonal matrix

$$D_z = \begin{pmatrix} z & 0 & 0 & 0 \\ 0 & z^2 & 0 & 0 \\ 0 & 0 & z^4 & 0 \\ 0 & 0 & 0 & z^8 \end{pmatrix}.$$

Clearly `MixColumns` is a trivial operation when $r = 1$. For $r = 2$ and $r = 4$ the effect of `MixColumns` can be replicated by pre-multiplying the vector conjugates of the column of the corresponding data array by the matrices

$$\left(\begin{array}{c|c} D_{\rho+1} & D_\rho \\ \hline D_\rho & D_{\rho+1} \end{array}\right) \text{ and } \left(\begin{array}{c|c|c|c} D_\rho & D_{\rho+1} & 1 & 1 \\ \hline 1 & D_\rho & D_{\rho+1} & 1 \\ \hline 1 & 1 & D_\rho & D_{\rho+1} \\ \hline D_{\rho+1} & 1 & 1 & D_\rho \end{array}\right) \text{ respectively.}$$

`AddRoundKey.` This operation can be replicated by adding the appropriate vector conjugates. As the key schedule essentially uses the same operations as the encryption process, it can also be easily replicated using vector conjugates.

Since inversion is the only non-linear part of the round function, we can move the S-Box constant into a slightly modified key schedule and construct an augmented linear diffusion layer consisting of the $GF(2)$-linear map, `ShiftRows` and `MixColumns` [7]. This augmented linear diffusion layer is given by an $(rc\times rc)$ matrix; examples of this useful representation are given in Appendix B.

In Appendix C, we give an example of a multivariate quadratic equation system for $\mathrm{SR}(2, 2, 2, 4)$, as well as a link to a website where systems for other small scale variants can be downloaded from. The given systems are constructed using the "BES-style" embedding. If the plaintext and ciphertext are known, which we assume, then the given equations are sufficient. If they are unknown, then the plaintext and ciphertext can be treated as variables. We note that whilst the system of equations is systematic, it does not form a minimal system. Furthermore, the systems of equations presented are correct only if no 0-inversion is performed either in the key schedule or in the encryption rounds. The probability of any particular inversion being a 0-inversion is 2^{-e}, so the probability that the entire equation system is free from 0-inversions can be easily estimated. In general, equations for the small scale variants can be easily established and the number of equations and variables for different variants are given here:

	Equations	State Variables	Key Variables
Encryption	$(4n+1)rce$	$2nrce$	$(n+1)rce$
Key Schedule	$2nrce + 2nre + rce$	-	nre (additional)
Total	$(6n+1)rce + 2nre + rce$	$2nrce$	$(n+1)rce + nre$

An alternative approach would be to work with equation systems over $GF(2)$, as originally proposed in [2]. If $w, x \in GF(2^e)$ are the input and output respectively of the "AES inversion", the relation $wx = 1$ gives rise to e bilinear expressions over $GF(2)$. Furthermore, we also have the "associated inversion" relations $w^2x = w$ and $wx^2 = x$, each of which give e further independent relations at the bit level. While there are also further relations of the form $w^4x = w^2 \cdot w$ and $wx^4 = x \cdot x^2$, we do not consider these. Using these expressions and those derived from the linear layer, we can construct a system of multivariate quadratic equations over $GF(2)$ in a similar manner to that given by [2]. In Section 4 we present the results of some experiments using both the $GF(2)$ and the BES-style representations.

4 Experimental Results

In this section we describe some experimental results concerning the solution of the equation systems that arise for these small scale variants of AES. These are basic timing experiments for the solution of the relevant system of equations by computing the Gröbner basis of the related polynomial ideal. The computations were made using the MAGMA 2.11-1 computer algebra package [6], which includes a highly efficient (particularly for $GF(2)$) implementation of Faugère's F4 algorithm [5]. In general, the Gröbner bases were computed with respect to the graded reverse lexicographic monomial ordering. All experiments were performed on a HP workstation, with Pentium 4 - 3GHz processor, 1 GB RAM, running Windows XP.

We are aware of the limitations of performing simple timings experiments using off-the-shelf software with limited computing resources. However we believe that such experiments can still be helpful in a preliminary assessment of algebraic attacks as cryptanalytic techniques. And while particularly degenerate small scale variants might not exhibit all the features of the AES, a comparison of attacks on such variants will help to provide an understanding of how various components and representations of the cipher contribute to the complexity of algebraic attacks.

4.1 $\mathbf{SR}(n, 1, 1, e)$

We ran experiments with the simple variants SR$(n, 1, 1, 4)$ and SR$(n, 1, 1, 8)$ for different number of rounds. We performed computations using the BES-style equation system over $GF(2^e)$, as well as the equation system over $GF(2)$. The $GF(2)$ equation systems are similar to that given in [2] with the addition of all field polynomials of the form $z^2 + z$. Table 1 shows the results for the computations; timings are given in seconds and N/A denotes insufficient memory to complete the computation.

In view of their simple form, we would expect to solve such equation systems for many rounds. This happened for SR$(n, 1, 1, 4)$, where we ran tests for up to 10 rounds. However the time required varied greatly when we changed the ordering of variables. When working with the cipher SR$(n, 1, 1, 8)$, we had problems with insufficient memory as early as three rounds. In particular, we note that

Cipher	Variables	Equations $GF(2^e)$	Monomials $GF(2^e)$	Time	Equations $GF(2)$	Monomials $GF(2)$	Time
SR(2,1,1,4)	36	72	89	0.11	104	137	0.03
SR(3,1,1,4)	52	104	129	0.75	152	201	0.11
SR(4,1,1,4)	68	136	169	2.02	200	265	0.28
SR(5,1,1,4)	84	168	209	7.47	248	339	0.97
SR(6,1,1,4)	100	200	249	23.71	296	393	4.30
SR(7,1,1,4)	116	232	289	56.74	344	457	11.26
SR(8,1,1,4)	132	264	329	43.70	392	521	16.56
SR(9,1,1,4)	148	296	369	219.38	440	585	46.05
SR(10,1,1,4)	164	328	409	340.31	488	649	74.06
SR(2,1,1,8)	72	144	177	43.55	172	365	118.45
SR(3,1,1,8)	104	208	257	N/A	252	541	N/A

Table 1. Time (in seconds) for Gröbner Basis computation of the $GF(2^e)$ and $GF(2)$ equation systems that arise from $\mathrm{SR}(n, 1, 1, e)$ (using graded reverse lexicographic monomial ordering).

the system for $\mathrm{SR}(3, 1, 1, 8)$ has a similar number of variables, monomials and equations as the system for $\mathrm{SR}(6, 1, 1, 4)$. Thus we might expect a similar performance for these two systems. However, our results show that this is not the case. This suggests that the underlying field equations, which are implicitly included in the BES-style equations, may play an important role in the computations for solving the system. However, this is yet to be established.

By comparing the results in Table 1, it is clear that the timings of computations for $\mathrm{SR}(n, 1, 1, 4)$ over $GF(2)$ are much better than those over $GF(2^4)$. However it is not clear whether this means that bit-level equations offer a better representation than BES-style equations in general, since MAGMA's implementation of the F4 algorithm is heavily optimised for operations over $GF(2)$ [13]. (In fact we see the opposite behaviour occurring for the cipher $\mathrm{SR}(n, 1, 1, 8)$.) Given the highly structured form of the BES-style systems we would expect computations using equation sets over $GF(2^e)$ to be generally more efficient than those over $GF(2)$.

4.2 SR$(n, 2, 1, 4)$ and SR$(n, 2, 2, 4)$

Some basic timing experiments with the systems derived from the variants $\mathrm{SR}(n, 2, 1, 4)$ and $\mathrm{SR}(n, 2, 2, 4)$ are given below. For these two variants, we also used MAGMA's implementation of Buchberger's algorithm in addition to computations of the Gröbner bases using the F4 algorithm. While we would expect Buchberger's algorithm to be slower, it should require less memory than the F4 algorithm. As before, timings are given in seconds, with N/A meaning insufficient memory to complete the computation (see Table 2).

By comparing the results in Table 2, we note that the equation system derived from $\mathrm{SR}(4, 2, 1, 4)$ has a similar number of variables, monomials and equations as the equation system arising from $\mathrm{SR}(2, 2, 2, 4)$. Therefore we might expect

Cipher	Variables	Equations	Monomials	Time F4	Time Buchberger
SR(1,2,1,4)	40	80	97	0.22	1.11
SR(2,2,1,4)	72	144	177	24.55	40.58
SR(3,2,1,4)	104	208	257	519.92	2649.90
SR(4,2,1,4)	136	272	337	N/A	28999.41
SR(1,2,2,4)	72	144	169	27.73	444.07
SR(2,2,2,4)	128	256	305	N/A	N/A

Table 2. Time (in seconds) for Gröbner Basis computation of $GF(2^4)$ equation systems with F4 and Buchberger's algorithm (using graded reverse lexicographic monomial ordering).

a similar performance in the computation for these two systems. However, our results confirm the important role played by the inter-word diffusion in the complexity of the computations. The diffusion of $\mathrm{SR}(n, 2, 1, 4)$ is limited, whereas $\mathrm{SR}(n, 2, 2, 4)$ has a similar diffusion pattern to that seen in the AES.

4.3 *Meet-in-the-Middle* Approach

Our experiments used the exact equation systems discussed in this paper; no pre-computation was performed and we did not explore any special structure. However it is well-known that the equation systems derived from the AES are highly structured, especially when represented as the set of BES-style equations over $GF(2^e)$. In particular, these systems might be viewed as "iterated" systems of equations, with similar blocks of multivariate quadratic equations repeated for every round. These blocks are connected to each other via the input and output variables, as well as the key schedule. When working with systems with such structure, a promising technique to find the overall solution is, in effect, a *meet-in-the-middle* approach: rather than attempting to solve the full system of equations for n rounds (we assume that n is even), we can try to solve two subsystems with $\frac{n}{2}$ rounds, by considering the output of round $\frac{n}{2}$ (which is also the input of round $\frac{n}{2}+1$) as variables. By choosing an appropriate monomial ordering we obtain two sets of equations (each covering half of the encryption operation) that relate these variables with the round subkeys. These two systems can then be combined along with some other equations relating the round subkeys. This gives a third smaller system which can be solved to obtain the encryption key.

We have tried this approach with some of the AES variants and compared the results with the timings obtained earlier. Our experiments suggest that this approach may be more efficient. For example, we were able to solve the system for $\mathrm{SR}(10, 1, 1, 4)$ using this approach in 42 seconds compared with 340 seconds using the naive approach. We also obtained better results for $\mathrm{SR}(4, 1, 1, 8)$ and $\mathrm{SR}(4, 2, 1, 4)$ using this approach (see Table 3).

This technique is cryptographically intuitive and is in fact a simple application of Elimination Theory [3], in which the Gröbner bases are computed with

Cipher	Variables	Equations	Monomials	Time
SR(10,1,1,4) - 5 rounds ↓	88	172	217	19.22
SR(10,1,1,4) - 5 rounds ↑	76	148	189	22.41
Solve	16	40	52	0.02
			Total:	41.65
SR(4,1,1,8) - 2 rounds ↓	80	152	193	15466.37
SR(4,1,1,8) - 2 rounds ↑	56	104	137	4603.89
Solve	32	80	576	215.92
			Total:	20286.18
SR(4,2,1,4) - 2 rounds ↓	80	152	193	667.17
SR(4,2,1,4) - 2 rounds ↑	56	104	137	2722.43
Solve	80	176	524	14.87
			Total:	3404.47

Table 3. Time (in seconds) for the *meet-in-the-middle* approach using F4 Gröbner Basis computation of equation systems arising from SR(10, 1, 1, 4), SR(4, 1, 1, 8) and SR(4, 2, 1, 4) using lexicographic ordering.

respect to the appropriate monomial ordering to eliminate the variables that do not appear in rounds $\frac{n}{2}$ and $\frac{n}{2}+1$. One problem with this approach is that computations using elimination orderings (such as lexicographic) are usually less efficient than those with degree orderings (such as graded reverse lexicographic). Thus, for more complex systems, we might expect that using lexicographic ordering in the two main subsystems would yield only limited benefit when compared with graded reverse lexicographic ordering for the full system. As an alternative, we could simply compute the Gröbner bases for the two subsystems (using the most efficient ordering) and combine both results to compute the solution of the full set equations. While this approach was more expensive for the variant SR(10, 1, 1, 4), it was more efficient for the cipher SR(4, 2, 1, 4) (see Table 4).

Cipher	Variables	Equations	Monomials	Time
SR(4,2,1,4) - 2 rounds ↓	112	216	273	553.63
SR(4,2,1,4) - 2 rounds ↑	104	200	257	1501.41
Solve	136	1197	918	12.68
			Total:	2067.72

Table 4. Time (in seconds) for the *meet-in-the-middle* approach using F4 Gröbner Basis computation of equation systems arising from SR(4, 2, 1, 4) using graded reverse lexicographic monomial ordering.

These results suggest the applicability of a more general *divide-and-conquer* approach to this problem, in which some form of (perhaps largely symbolic) pre-computation could be performed and then combined to produce the solution of the full system. This might be a promising direction and more research will assess whether this approach might increase the efficiency of algebraic attacks against the AES and related ciphers.

5 Conclusions

We have defined a family of small scale variants of the AES. This provides a common framework for the analysis of AES-like equation systems. We also present some basic experimental results when using off-the-shelf computational algebra techniques to solve these systems. These provide some preliminary insight into the behavior of algebraic attacks and future work can now take place within a framework for the systematic analysis of small scale AES variants.

References

1. C. Cid, S. Murphy, and M.J.B. Robshaw. Computational and Algebraic Aspects of the Advanced Encryption Standard. In V. Ganzha *et al.*, editors, Proceedings of the *Seventh International Workshop on Computer Algebra in Scientific Computing - CASC 2004*, St. Petersburg, Russia, pages 93–103, Technische Universität München. 2004.
2. N. Courtois and J. Pieprzyk. Cryptanalysis of block ciphers with overdefined systems of equations. In Y. Zheng, editor, Proceedings of *Asiacrypt 2002*, LNCS 2501, pages 267–287, Springer-Verlag, 2002.
3. D. Cox, J. Little and D. O'Shea. Ideals, Varieties, and Algorithms. Undergraduate Texts in Mathematics, Second Edition, Springer-Verlag, 1997.
4. J. Daemen and V. Rijmen. *The Design of Rijndael: AES – The Advanced Encryption Standard*, Springer-Verlag, 2002.
5. J.-C. Faugère. A new efficient algorithm for computing Gröbner bases (F4). Journal of Pure and Applied Algebra, 139, pages 61–88, 1999.
6. Magma V2.11-1, Computational Algebra Group, School of Mathematics and Statistics, University of Sydney. Website: `http://magma.maths.usyd.edu.au`. 2004.
7. S. Murphy and M.J.B. Robshaw. New Observations on Rijndael. Submitted to NIST. Available via `csrc.nist.gov`. 7 August 2000.
8. S. Murphy and M.J.B. Robshaw. Essential Algebraic Structure within the AES. In M. Yung, editor, Proceedings of *CRYPTO 2002*, LNCS 2442, pages 11–16, Springer-Verlag, 2002.
9. S. Murphy and M.J.B. Robshaw. Comments on the Security of the AES and the XSL Technique. *Electronics Letters* Vol. 39, pp 36-38, 2002.
10. M.A. Musa, E.F. Schaefer, and S. Wedig. A simplified AES algorithm and its linear and differential cryptanalysis. Cryptologia, Vol. XXVII (2), pages 148-177, 2003.
11. National Institute of Standards and Technology. Advanced Encryption Standard. FIPS 197. November 26, 2001.
12. R.C.-W. Phan. Mini Advanced Encryption Standard (Mini-AES): A Testbed for Cryptanalysis Students, Cryptologia, Vol. XXVI (4), pages 283-306, 2002.
13. A. Steel. (Magma Development Team). Personal communication, October 2004.

Appendix A: Key Schedule Equations

SR(n, r, c, e) has a user-provided key of rce bits, which is considered to be an array of $(r \times c)$ e-bits words. This key forms the initial subkey. Each subkey is then used to define the next subkey as described below. This description uses constants and functions which depend on the field $GF(2^e)$. All constants and functions (apart from the round constant κ_i) have been discussed elsewhere and are summarised in the following table.

		$GF(2^4)$	$GF(2^8)$
Round constant	κ_i	$\rho^{(i-1)}$	$\theta^{(i-1)}$
S-Box constant	d	6	63
Inversion	$z \mapsto z^{(-1)}$	Inversion in $GF(2^4)$	Inversion in $GF(2^8)$
$GF(2)$-linear map	$z \mapsto L(z)$	L for $GF(2^4)$	L for $GF(2^8)$

We regard each round subkey as a column $GF(2^e)$-vector of length rc. In order to define the key schedule, we effectively divide the round subkey vector into c subvectors of length r. Thus the subkey vectors are given below.

$$\begin{array}{cc}
\text{Initial Subkey} & \left(k_{0,0}, \ldots, k_{0,r-1}, \ldots, k_{0,r(c-1)}, \ldots, k_{0,rc-1}\right)^T \\
\text{Round 1 Subkey} & \left(k_{1,0}, \ldots, k_{1,r-1}, \ldots, k_{1,r(c-1)}, \ldots, k_{1,rc-1}\right)^T \\
\vdots & \vdots \\
\text{Round } n \text{ Subkey} & \left(k_{n,0}, \ldots, k_{n,r-1}, \ldots, k_{n,r(c-1)}, \ldots, k_{n,rc-1}\right)^T
\end{array}$$

The definition of the round subkeys now depends on the number of rows (r) and columns (c) in the array. The round subkeys are defined recursively for each round $1 \leq i \leq n$.

Key Schedule for One Row ($r = 1$).

$$s_0 = k^{(-1)}_{i-1,c-1}.$$

– One column ($r = 1, c = 1$).

$$(k_{i,0}) = (L(s_0)) + (d) + (\kappa_i)\,.$$

– More than one column ($r = 1, c > 1$). For $0 \leq q \leq c - 1$

$$(k_{i,q}) = (L(s_0)) + (d) + (\kappa_i) \ + \sum_{t=0}^{q} (k_{i-1,t}).$$

Key Schedule for Two Rows ($r = 2$).

$$s_0 = k^{(-1)}_{i-1,2c-1}, \qquad s_1 = k^{(-1)}_{i-1,2c-2}.$$

– One column ($r = 2, c = 1$).

$$\begin{pmatrix} k_{i,0} \\ k_{i,1} \end{pmatrix} = \begin{pmatrix} L(s_0) \\ L(s_1) \end{pmatrix} + \begin{pmatrix} d \\ d \end{pmatrix} + \begin{pmatrix} \kappa_i \\ 0 \end{pmatrix}.$$

– More than one column ($r = 2, c > 1$). For $0 \leq q \leq c - 1$

$$\begin{pmatrix} k_{i,rq} \\ k_{i,rq+1} \end{pmatrix} = \begin{pmatrix} L(s_0) \\ L(s_1) \end{pmatrix} + \begin{pmatrix} d \\ d \end{pmatrix} + \begin{pmatrix} \kappa_i \\ 0 \end{pmatrix} + \sum_{t=0}^{q} \begin{pmatrix} k_{i-1,rt} \\ k_{i-1,rt+1} \end{pmatrix}.$$

Key Schedule for Four Rows ($r = 4$).

$$s_0 = k_{i-1,4c-1}^{(-1)}, \quad s_1 = k_{i-1,4c-2}^{(-1)}, \quad s_2 = k_{i-1,4c-3}^{(-1)}, \quad s_3 = k_{i-1,4c-4}^{(-1)}.$$

– One column ($r = 4, c = 1$).

$$\begin{pmatrix} k_{i,0} \\ k_{i,1} \\ k_{i,2} \\ k_{i,3} \end{pmatrix} = \begin{pmatrix} L(s_0) \\ L(s_1) \\ L(s_2) \\ L(s_3) \end{pmatrix} + \begin{pmatrix} d \\ d \\ d \\ d \end{pmatrix} + \begin{pmatrix} \kappa_i \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

– More than one column ($r = 4, c > 1$). For $0 \leq q \leq c - 1$

$$\begin{pmatrix} k_{i,rq} \\ k_{i,rq+1} \\ k_{i,rq+2} \\ k_{i,rq+3} \end{pmatrix} = \begin{pmatrix} L(s_0) \\ L(s_1) \\ L(s_2) \\ L(s_3) \end{pmatrix} + \begin{pmatrix} d \\ d \\ d \\ d \end{pmatrix} + \begin{pmatrix} \kappa_i \\ 0 \\ 0 \\ 0 \end{pmatrix} + \sum_{t=0}^{q} \begin{pmatrix} k_{i-1,rt} \\ k_{i-1,rt+1} \\ k_{i-1,rt+2} \\ k_{i-1,rt+3} \end{pmatrix}.$$

Appendix B: Augmented Linear Diffusion Layer

For small scale variants of the AES, we can construct an augmented linear diffusion layer that consists of the $GF(2)$-linear map, `ShiftRows` and `MixColumns` [7]. This helps to provide a natural set of equations. The augmented linear diffusion layer can be represented by an $(rc \times rc)$ matrix. If we replace every entry z of this matrix by D_z given earlier, we obtain an $(rce \times rce)$ matrix which replicates the augmented linear diffusion layer for vector conjugates. We provide these matrices for different array sizes over $GF(2^4)$. For array sizes (1×1) and (2×1), the augmented linear diffusion layers for vector conjugates are given by the matrices

$$\begin{pmatrix} \mathtt{5} & \mathtt{1} & \mathtt{C} & \mathtt{5} \\ \mathtt{2} & \mathtt{2} & \mathtt{1} & \mathtt{F} \\ \mathtt{A} & \mathtt{4} & \mathtt{4} & \mathtt{1} \\ \mathtt{1} & \mathtt{8} & \mathtt{3} & \mathtt{3} \end{pmatrix} \text{ and } \left(\begin{array}{cccc|cccc} \mathtt{F} & \mathtt{3} & \mathtt{7} & \mathtt{F} & \mathtt{A} & \mathtt{2} & \mathtt{B} & \mathtt{A} \\ \mathtt{A} & \mathtt{A} & \mathtt{5} & \mathtt{6} & \mathtt{8} & \mathtt{8} & \mathtt{4} & \mathtt{9} \\ \mathtt{7} & \mathtt{8} & \mathtt{8} & \mathtt{2} & \mathtt{D} & \mathtt{C} & \mathtt{C} & \mathtt{3} \\ \mathtt{4} & \mathtt{6} & \mathtt{C} & \mathtt{C} & \mathtt{5} & \mathtt{E} & \mathtt{F} & \mathtt{F} \\ \hline \mathtt{A} & \mathtt{2} & \mathtt{B} & \mathtt{A} & \mathtt{F} & \mathtt{3} & \mathtt{7} & \mathtt{F} \\ \mathtt{8} & \mathtt{8} & \mathtt{4} & \mathtt{9} & \mathtt{A} & \mathtt{A} & \mathtt{5} & \mathtt{6} \\ \mathtt{D} & \mathtt{C} & \mathtt{C} & \mathtt{3} & \mathtt{7} & \mathtt{8} & \mathtt{8} & \mathtt{2} \\ \mathtt{5} & \mathtt{E} & \mathtt{F} & \mathtt{F} & \mathtt{4} & \mathtt{6} & \mathtt{C} & \mathtt{C} \end{array}\right), \text{ respectively.}$$

For the (2×2)-array, the augmented diffusion layer is given by

$$
\left(\begin{array}{cccc|cccc|cccc|cccc}
F&3&7&F&0&0&0&0&0&0&0&0&A&2&B&A\\
A&A&5&6&0&0&0&0&0&0&0&0&8&8&4&9\\
7&8&8&2&0&0&0&0&0&0&0&0&D&C&C&3\\
4&6&C&C&0&0&0&0&0&0&0&0&5&E&F&F\\
\hline
A&2&B&A&0&0&0&0&0&0&0&0&F&3&7&F\\
8&8&4&9&0&0&0&0&0&0&0&0&A&A&5&6\\
D&C&C&3&0&0&0&0&0&0&0&0&7&8&8&2\\
5&E&F&F&0&0&0&0&0&0&0&0&4&6&C&C\\
\hline
0&0&0&0&A&2&B&A&F&3&7&F&0&0&0&0\\
0&0&0&0&8&8&4&9&A&A&5&6&0&0&0&0\\
0&0&0&0&D&C&C&3&7&8&8&2&0&0&0&0\\
0&0&0&0&5&E&F&F&4&6&C&C&0&0&0&0\\
\hline
0&0&0&0&F&3&7&F&A&2&B&A&0&0&0&0\\
0&0&0&0&A&A&5&6&8&8&4&9&0&0&0&0\\
0&0&0&0&7&8&8&2&D&C&C&3&0&0&0&0\\
0&0&0&0&4&6&C&C&5&E&F&F&0&0&0&0
\end{array}\right).
$$

Appendix C: Equation System for SR(2, 2, 2, 4)

We illustrate the kind of equation systems that arise by listing the BES-style relations between variables in an SR(2, 2, 2, 4) encryption and key schedule. If neither the encryption nor the key schedule require a 0-inversion, then each of these relations is identically 0. Under this assumption, the following relations give a multivariate quadratic equation system for SR(2, 2, 2, 4) over $GF(2^4)$. The probability that the encryption rounds do not require any 0-inversions is about $\left(\frac{15}{16}\right)^8 \approx 0.60$. The probability that the key schedule does not require any 0-inversions is about $\left(\frac{15}{16}\right)^4 \approx 0.77$. Systems for other small scale variants can be downloaded from the site `http://www.isg.rhul.ac.uk/aes/index.html`.

Component j and conjugate l for the plaintext, ciphertext and the key (also used as the initial subkey) are denoted by p_{jl}, c_{jl} and k_{0jl} respectively. We regard the two rounds as round one and round two. We denote the input and output of the inversion and the subkey used in round i for component j and conjugate l by w_{ijl}, x_{ijl} and k_{ijl} respectively.

	$GF(2^4)$ Variable	Round i	Component j	Conjugate l
Plaintext	p_{jl}		0, 1, 2, 3	0, 1, 2, 3
Ciphertext	c_{jl}		0, 1, 2, 3	0, 1, 2, 3
State				
Inversion Input	w_{ijl}	1, 2	0, 1, 2, 3	0, 1, 2, 3
Inversion Output	x_{ijl}	1, 2	0, 1, 2, 3	0, 1, 2, 3
Key				
Subkey	k_{ijl}	0, 1, 2	0, 1, 2, 3	0, 1, 2, 3
Dummy	s_{ijl}	0, 1	0, 1	0, 1, 2, 3

Initial Subkey Relations

$$
\begin{array}{llll}
w_{100}+p_{00}+k_{000} & w_{110}+p_{10}+k_{010} & w_{120}+p_{20}+k_{020} & w_{130}+p_{30}+k_{030} \\
w_{101}+p_{01}+k_{001} & w_{111}+p_{11}+k_{011} & w_{121}+p_{21}+k_{021} & w_{131}+p_{31}+k_{031} \\
w_{102}+p_{02}+k_{002} & w_{112}+p_{12}+k_{012} & w_{122}+p_{22}+k_{022} & w_{132}+p_{32}+k_{032} \\
w_{103}+p_{03}+k_{003} & w_{113}+p_{13}+k_{013} & w_{123}+p_{23}+k_{023} & w_{133}+p_{33}+k_{033}
\end{array}
$$

Inversion and Conjugacy Relations: Rounds 1 and 2

$$
\begin{array}{llllll}
w_{100}^2+w_{101} & w_{100}x_{100}+1 & x_{100}^2+x_{101} & w_{200}^2+w_{201} & w_{200}x_{200}+1 & x_{200}^2+x_{201} \\
w_{101}^2+w_{102} & w_{101}x_{101}+1 & x_{101}^2+x_{102} & w_{201}^2+w_{202} & w_{201}x_{201}+1 & x_{201}^2+x_{202} \\
w_{102}^2+w_{103} & w_{102}x_{102}+1 & x_{102}^2+x_{103} & w_{202}^2+w_{203} & w_{202}x_{202}+1 & x_{202}^2+x_{203} \\
w_{103}^2+w_{100} & w_{103}x_{103}+1 & x_{103}^2+x_{100} & w_{203}^2+w_{200} & w_{203}x_{203}+1 & x_{203}^2+x_{200} \\
w_{110}^2+w_{111} & w_{110}x_{110}+1 & x_{110}^2+x_{111} & w_{210}^2+w_{211} & w_{210}x_{210}+1 & x_{210}^2+x_{211} \\
w_{111}^2+w_{112} & w_{111}x_{111}+1 & x_{111}^2+x_{112} & w_{211}^2+w_{212} & w_{211}x_{211}+1 & x_{211}^2+x_{212} \\
w_{112}^2+w_{113} & w_{112}x_{112}+1 & x_{112}^2+x_{113} & w_{212}^2+w_{213} & w_{212}x_{212}+1 & x_{212}^2+x_{213} \\
w_{113}^2+w_{110} & w_{113}x_{113}+1 & x_{113}^2+x_{110} & w_{213}^2+w_{210} & w_{213}x_{213}+1 & x_{213}^2+x_{210} \\
w_{120}^2+w_{121} & w_{120}x_{120}+1 & x_{120}^2+x_{121} & w_{220}^2+w_{221} & w_{220}x_{220}+1 & x_{220}^2+x_{221} \\
w_{121}^2+w_{122} & w_{121}x_{121}+1 & x_{121}^2+x_{122} & w_{221}^2+w_{222} & w_{221}x_{221}+1 & x_{221}^2+x_{222} \\
w_{122}^2+w_{123} & w_{122}x_{122}+1 & x_{122}^2+x_{123} & w_{222}^2+w_{223} & w_{222}x_{222}+1 & x_{222}^2+x_{223} \\
w_{123}^2+w_{120} & w_{123}x_{123}+1 & x_{123}^2+x_{120} & w_{223}^2+w_{220} & w_{223}x_{223}+1 & x_{223}^2+x_{220} \\
w_{130}^2+w_{131} & w_{130}x_{130}+1 & x_{130}^2+x_{131} & w_{230}^2+w_{231} & w_{230}x_{230}+1 & x_{230}^2+x_{231} \\
w_{131}^2+w_{132} & w_{131}x_{131}+1 & x_{131}^2+x_{132} & w_{231}^2+w_{232} & w_{231}x_{231}+1 & x_{231}^2+x_{232} \\
w_{132}^2+w_{133} & w_{132}x_{132}+1 & x_{132}^2+x_{133} & w_{232}^2+w_{233} & w_{232}x_{232}+1 & x_{232}^2+x_{233} \\
w_{133}^2+w_{130} & w_{133}x_{133}+1 & x_{133}^2+x_{130} & w_{233}^2+w_{230} & w_{233}x_{233}+1 & x_{233}^2+x_{230}
\end{array}
$$

Diffusion Relations: Rounds 1 and 2

$$
\begin{array}{l}
w_{200}+\mathtt{F}x_{100}+\mathtt{3}x_{101}+\mathtt{7}x_{102}+\mathtt{F}x_{103}+\mathtt{A}x_{130}+\mathtt{2}x_{131}+\mathtt{B}x_{132}+\mathtt{A}x_{133}+k_{100}+\mathtt{6} \\
w_{201}+\mathtt{A}x_{100}+\mathtt{A}x_{101}+\mathtt{5}x_{102}+\mathtt{6}x_{103}+\mathtt{8}x_{130}+\mathtt{8}x_{131}+\mathtt{4}x_{132}+\mathtt{9}x_{133}+k_{101}+\mathtt{7} \\
w_{202}+\mathtt{7}x_{100}+\mathtt{8}x_{101}+\mathtt{8}x_{102}+\mathtt{2}x_{103}+\mathtt{D}x_{130}+\mathtt{C}x_{131}+\mathtt{C}x_{132}+\mathtt{3}x_{133}+k_{102}+\mathtt{6} \\
w_{203}+\mathtt{4}x_{100}+\mathtt{6}x_{101}+\mathtt{C}x_{102}+\mathtt{C}x_{103}+\mathtt{5}x_{130}+\mathtt{E}x_{131}+\mathtt{F}x_{132}+\mathtt{F}x_{133}+k_{103}+\mathtt{7} \\
w_{210}+\mathtt{A}x_{100}+\mathtt{2}x_{101}+\mathtt{B}x_{102}+\mathtt{A}x_{103}+\mathtt{F}x_{130}+\mathtt{3}x_{131}+\mathtt{7}x_{132}+\mathtt{F}x_{133}+k_{110}+\mathtt{6} \\
w_{211}+\mathtt{8}x_{100}+\mathtt{8}x_{101}+\mathtt{4}x_{102}+\mathtt{9}x_{103}+\mathtt{A}x_{130}+\mathtt{A}x_{131}+\mathtt{5}x_{132}+\mathtt{6}x_{133}+k_{111}+\mathtt{7} \\
w_{212}+\mathtt{D}x_{100}+\mathtt{C}x_{101}+\mathtt{C}x_{102}+\mathtt{3}x_{103}+\mathtt{7}x_{130}+\mathtt{8}x_{131}+\mathtt{8}x_{132}+\mathtt{2}x_{133}+k_{112}+\mathtt{6} \\
w_{213}+\mathtt{5}x_{100}+\mathtt{E}x_{101}+\mathtt{F}x_{102}+\mathtt{F}x_{103}+\mathtt{4}x_{130}+\mathtt{6}x_{131}+\mathtt{C}x_{132}+\mathtt{C}x_{133}+k_{113}+\mathtt{7} \\
w_{220}+\mathtt{A}x_{110}+\mathtt{2}x_{111}+\mathtt{B}x_{112}+\mathtt{A}x_{113}+\mathtt{F}x_{120}+\mathtt{3}x_{121}+\mathtt{7}x_{122}+\mathtt{F}x_{123}+k_{120}+\mathtt{6} \\
w_{221}+\mathtt{8}x_{110}+\mathtt{8}x_{111}+\mathtt{4}x_{112}+\mathtt{9}x_{113}+\mathtt{A}x_{120}+\mathtt{A}x_{121}+\mathtt{5}x_{122}+\mathtt{6}x_{123}+k_{121}+\mathtt{7} \\
w_{222}+\mathtt{D}x_{110}+\mathtt{C}x_{111}+\mathtt{C}x_{112}+\mathtt{3}x_{113}+\mathtt{7}x_{120}+\mathtt{8}x_{121}+\mathtt{8}x_{122}+\mathtt{2}x_{123}+k_{122}+\mathtt{6} \\
w_{223}+\mathtt{5}x_{110}+\mathtt{E}x_{111}+\mathtt{F}x_{112}+\mathtt{F}x_{113}+\mathtt{4}x_{120}+\mathtt{6}x_{121}+\mathtt{C}x_{122}+\mathtt{C}x_{123}+k_{123}+\mathtt{7} \\
w_{230}+\mathtt{F}x_{110}+\mathtt{3}x_{111}+\mathtt{7}x_{112}+\mathtt{F}x_{113}+\mathtt{A}x_{120}+\mathtt{2}x_{121}+\mathtt{B}x_{122}+\mathtt{A}x_{123}+k_{130}+\mathtt{6} \\
w_{231}+\mathtt{A}x_{110}+\mathtt{A}x_{111}+\mathtt{5}x_{112}+\mathtt{6}x_{113}+\mathtt{8}x_{120}+\mathtt{8}x_{121}+\mathtt{4}x_{122}+\mathtt{9}x_{123}+k_{131}+\mathtt{7} \\
w_{232}+\mathtt{7}x_{110}+\mathtt{8}x_{111}+\mathtt{8}x_{112}+\mathtt{2}x_{113}+\mathtt{D}x_{120}+\mathtt{C}x_{121}+\mathtt{C}x_{122}+\mathtt{3}x_{123}+k_{132}+\mathtt{6} \\
w_{233}+\mathtt{4}x_{110}+\mathtt{6}x_{111}+\mathtt{C}x_{112}+\mathtt{C}x_{113}+\mathtt{5}x_{120}+\mathtt{E}x_{121}+\mathtt{F}x_{122}+\mathtt{F}x_{123}+k_{133}+\mathtt{7}
\end{array}
$$

$$c_{00} + \mathtt{F}x_{200} + \mathtt{3}x_{201} + \mathtt{7}x_{202} + \mathtt{F}x_{203} + \mathtt{A}x_{230} + \mathtt{2}x_{231} + \mathtt{B}x_{232} + \mathtt{A}x_{233} + k_{200} + \mathtt{6}$$
$$c_{01} + \mathtt{A}x_{200} + \mathtt{A}x_{201} + \mathtt{5}x_{202} + \mathtt{6}x_{203} + \mathtt{8}x_{230} + \mathtt{8}x_{231} + \mathtt{4}x_{232} + \mathtt{9}x_{233} + k_{201} + \mathtt{7}$$
$$c_{02} + \mathtt{7}x_{200} + \mathtt{8}x_{201} + \mathtt{8}x_{202} + \mathtt{2}x_{203} + \mathtt{D}x_{230} + \mathtt{C}x_{231} + \mathtt{C}x_{232} + \mathtt{3}x_{233} + k_{202} + \mathtt{6}$$
$$c_{03} + \mathtt{4}x_{200} + \mathtt{6}x_{201} + \mathtt{C}x_{202} + \mathtt{C}x_{203} + \mathtt{5}x_{230} + \mathtt{E}x_{231} + \mathtt{F}x_{232} + \mathtt{F}x_{233} + k_{203} + \mathtt{7}$$
$$c_{10} + \mathtt{A}x_{200} + \mathtt{2}x_{201} + \mathtt{B}x_{202} + \mathtt{A}x_{203} + \mathtt{F}x_{230} + \mathtt{3}x_{231} + \mathtt{7}x_{232} + \mathtt{F}x_{233} + k_{210} + \mathtt{6}$$
$$c_{11} + \mathtt{8}x_{200} + \mathtt{8}x_{201} + \mathtt{4}x_{202} + \mathtt{9}x_{203} + \mathtt{A}x_{230} + \mathtt{A}x_{231} + \mathtt{5}x_{232} + \mathtt{6}x_{233} + k_{211} + \mathtt{7}$$
$$c_{12} + \mathtt{D}x_{200} + \mathtt{C}x_{201} + \mathtt{C}x_{202} + \mathtt{3}x_{203} + \mathtt{7}x_{230} + \mathtt{8}x_{231} + \mathtt{8}x_{232} + \mathtt{2}x_{233} + k_{212} + \mathtt{6}$$
$$c_{13} + \mathtt{5}x_{200} + \mathtt{E}x_{201} + \mathtt{F}x_{202} + \mathtt{F}x_{203} + \mathtt{4}x_{230} + \mathtt{6}x_{231} + \mathtt{C}x_{232} + \mathtt{C}x_{233} + k_{213} + \mathtt{7}$$
$$c_{20} + \mathtt{A}x_{210} + \mathtt{2}x_{211} + \mathtt{B}x_{212} + \mathtt{A}x_{213} + \mathtt{F}x_{220} + \mathtt{3}x_{221} + \mathtt{7}x_{222} + \mathtt{F}x_{223} + k_{220} + \mathtt{6}$$
$$c_{21} + \mathtt{8}x_{210} + \mathtt{8}x_{211} + \mathtt{4}x_{212} + \mathtt{9}x_{213} + \mathtt{A}x_{220} + \mathtt{A}x_{221} + \mathtt{5}x_{222} + \mathtt{6}x_{223} + k_{221} + \mathtt{7}$$
$$c_{22} + \mathtt{D}x_{210} + \mathtt{C}x_{211} + \mathtt{C}x_{212} + \mathtt{3}x_{213} + \mathtt{7}x_{220} + \mathtt{8}x_{221} + \mathtt{8}x_{222} + \mathtt{2}x_{223} + k_{222} + \mathtt{6}$$
$$c_{23} + \mathtt{5}x_{210} + \mathtt{E}x_{211} + \mathtt{F}x_{212} + \mathtt{F}x_{213} + \mathtt{4}x_{220} + \mathtt{6}x_{221} + \mathtt{C}x_{222} + \mathtt{C}x_{223} + k_{223} + \mathtt{7}$$
$$c_{30} + \mathtt{F}x_{210} + \mathtt{3}x_{211} + \mathtt{7}x_{212} + \mathtt{F}x_{213} + \mathtt{A}x_{220} + \mathtt{2}x_{221} + \mathtt{B}x_{222} + \mathtt{A}x_{223} + k_{230} + \mathtt{6}$$
$$c_{31} + \mathtt{A}x_{210} + \mathtt{A}x_{211} + \mathtt{5}x_{212} + \mathtt{6}x_{213} + \mathtt{8}x_{220} + \mathtt{8}x_{221} + \mathtt{4}x_{222} + \mathtt{9}x_{223} + k_{231} + \mathtt{7}$$
$$c_{32} + \mathtt{7}x_{210} + \mathtt{8}x_{211} + \mathtt{8}x_{212} + \mathtt{2}x_{213} + \mathtt{D}x_{220} + \mathtt{C}x_{221} + \mathtt{C}x_{222} + \mathtt{3}x_{223} + k_{232} + \mathtt{6}$$
$$c_{33} + \mathtt{4}x_{210} + \mathtt{6}x_{211} + \mathtt{C}x_{212} + \mathtt{C}x_{213} + \mathtt{5}x_{220} + \mathtt{E}x_{221} + \mathtt{F}x_{222} + \mathtt{F}x_{223} + k_{233} + \mathtt{7}$$

Key Schedule Conjugacy Relations

$k_{000}^2 + k_{001}$	$k_{100}^2 + k_{101}$	$k_{200}^2 + k_{201}$
$k_{001}^2 + k_{002}$	$k_{101}^2 + k_{102}$	$k_{201}^2 + k_{202}$
$k_{002}^2 + k_{003}$	$k_{102}^2 + k_{103}$	$k_{202}^2 + k_{203}$
$k_{003}^2 + k_{000}$	$k_{103}^2 + k_{100}$	$k_{203}^2 + k_{200}$
$k_{010}^2 + k_{011}$	$k_{110}^2 + k_{111}$	$k_{210}^2 + k_{211}$
$k_{011}^2 + k_{012}$	$k_{111}^2 + k_{112}$	$k_{211}^2 + k_{212}$
$k_{012}^2 + k_{013}$	$k_{112}^2 + k_{113}$	$k_{212}^2 + k_{213}$
$k_{013}^2 + k_{010}$	$k_{113}^2 + k_{110}$	$k_{213}^2 + k_{210}$
$k_{020}^2 + k_{021}$	$k_{120}^2 + k_{121}$	$k_{220}^2 + k_{221}$
$k_{021}^2 + k_{022}$	$k_{121}^2 + k_{122}$	$k_{221}^2 + k_{222}$
$k_{022}^2 + k_{023}$	$k_{122}^2 + k_{123}$	$k_{222}^2 + k_{223}$
$k_{023}^2 + k_{020}$	$k_{123}^2 + k_{120}$	$k_{223}^2 + k_{220}$
$k_{030}^2 + k_{031}$	$k_{130}^2 + k_{131}$	$k_{230}^2 + k_{231}$
$k_{031}^2 + k_{032}$	$k_{131}^2 + k_{132}$	$k_{231}^2 + k_{232}$
$k_{032}^2 + k_{033}$	$k_{132}^2 + k_{133}$	$k_{232}^2 + k_{233}$
$k_{033}^2 + k_{030}$	$k_{133}^2 + k_{130}$	$k_{233}^2 + k_{230}$

Key Schedule Inversion and Conjugacy Relations

$k_{030}s_{000} + 1$	$s_{000}^2 + s_{001}$	$k_{130}s_{100} + 1$	$s_{100}^2 + s_{101}$
$k_{031}s_{001} + 1$	$s_{001}^2 + s_{002}$	$k_{131}s_{101} + 1$	$s_{101}^2 + s_{102}$
$k_{032}s_{002} + 1$	$s_{002}^2 + s_{003}$	$k_{132}s_{102} + 1$	$s_{102}^2 + s_{103}$
$k_{033}s_{003} + 1$	$s_{003}^2 + s_{000}$	$k_{133}s_{103} + 1$	$s_{103}^2 + s_{100}$
$k_{020}s_{010} + 1$	$s_{010}^2 + s_{011}$	$k_{120}s_{110} + 1$	$s_{110}^2 + s_{111}$
$k_{021}s_{011} + 1$	$s_{011}^2 + s_{012}$	$k_{121}s_{111} + 1$	$s_{111}^2 + s_{112}$
$k_{022}s_{012} + 1$	$s_{012}^2 + s_{013}$	$k_{122}s_{112} + 1$	$s_{112}^2 + s_{113}$
$k_{023}s_{013} + 1$	$s_{013}^2 + s_{010}$	$k_{023}s_{113} + 1$	$s_{113}^2 + s_{110}$

Key Schedule Diffusion Relations: Round 1

$$
\begin{array}{ll}
k_{100} + k_{000} & +\mathtt{5}s_{000} + \mathtt{1}s_{001} + \mathtt{C}s_{002} + \mathtt{5}s_{003} + \mathtt{7} \\
k_{101} + k_{001} & +\mathtt{2}s_{000} + \mathtt{2}s_{001} + \mathtt{1}s_{002} + \mathtt{F}s_{003} + \mathtt{6} \\
k_{102} + k_{002} & +\mathtt{A}s_{000} + \mathtt{4}s_{001} + \mathtt{4}s_{002} + \mathtt{1}s_{003} + \mathtt{7} \\
k_{103} + k_{003} & +\mathtt{1}s_{000} + \mathtt{8}s_{001} + \mathtt{3}s_{002} + \mathtt{3}s_{003} + \mathtt{6} \\
k_{110} + k_{010} & +\mathtt{5}s_{010} + \mathtt{1}s_{011} + \mathtt{C}s_{012} + \mathtt{5}s_{013} + \mathtt{6} \\
k_{111} + k_{011} & +\mathtt{2}s_{010} + \mathtt{2}s_{011} + \mathtt{1}s_{012} + \mathtt{F}s_{013} + \mathtt{7} \\
k_{112} + k_{012} & +\mathtt{A}s_{010} + \mathtt{4}s_{011} + \mathtt{4}s_{012} + \mathtt{1}s_{013} + \mathtt{6} \\
k_{113} + k_{013} & +\mathtt{1}s_{010} + \mathtt{8}s_{011} + \mathtt{3}s_{012} + \mathtt{3}s_{013} + \mathtt{7} \\
k_{120} + k_{020} + k_{000} & +\mathtt{5}s_{000} + \mathtt{1}s_{001} + \mathtt{C}s_{002} + \mathtt{5}s_{003} + \mathtt{7} \\
k_{121} + k_{021} + k_{001} & +\mathtt{2}s_{000} + \mathtt{2}s_{001} + \mathtt{1}s_{002} + \mathtt{F}s_{003} + \mathtt{6} \\
k_{122} + k_{022} + k_{002} & +\mathtt{A}s_{000} + \mathtt{4}s_{001} + \mathtt{4}s_{002} + \mathtt{1}s_{003} + \mathtt{7} \\
k_{123} + k_{023} + k_{003} & +\mathtt{1}s_{000} + \mathtt{8}s_{001} + \mathtt{3}s_{002} + \mathtt{3}s_{003} + \mathtt{6} \\
k_{130} + k_{030} + k_{010} & +\mathtt{5}s_{010} + \mathtt{1}s_{011} + \mathtt{C}s_{012} + \mathtt{5}s_{013} + \mathtt{6} \\
k_{131} + k_{031} + k_{011} & +\mathtt{2}s_{010} + \mathtt{2}s_{011} + \mathtt{1}s_{012} + \mathtt{F}s_{013} + \mathtt{7} \\
k_{132} + k_{032} + k_{012} & +\mathtt{A}s_{010} + \mathtt{4}s_{011} + \mathtt{4}s_{012} + \mathtt{1}s_{013} + \mathtt{6} \\
k_{133} + k_{033} + k_{013} & +\mathtt{1}s_{010} + \mathtt{8}s_{011} + \mathtt{3}s_{012} + \mathtt{3}s_{013} + \mathtt{7}
\end{array}
$$

Key Schedule Diffusion Relations: Round 2

$$
\begin{array}{ll}
k_{200} + k_{100} & +\mathtt{5}s_{100} + \mathtt{1}s_{101} + \mathtt{C}s_{102} + \mathtt{5}s_{103} + \mathtt{4} \\
k_{201} + k_{101} & +\mathtt{2}s_{100} + \mathtt{2}s_{101} + \mathtt{1}s_{102} + \mathtt{F}s_{103} + \mathtt{3} \\
k_{202} + k_{102} & +\mathtt{A}s_{100} + \mathtt{4}s_{101} + \mathtt{4}s_{102} + \mathtt{1}s_{103} + \mathtt{5} \\
k_{203} + k_{103} & +\mathtt{1}s_{100} + \mathtt{8}s_{101} + \mathtt{3}s_{102} + \mathtt{3}s_{103} + \mathtt{2} \\
k_{210} + k_{110} & +\mathtt{5}s_{110} + \mathtt{1}s_{111} + \mathtt{C}s_{112} + \mathtt{5}s_{113} + \mathtt{6} \\
k_{211} + k_{111} & +\mathtt{2}s_{110} + \mathtt{2}s_{111} + \mathtt{1}s_{112} + \mathtt{F}s_{113} + \mathtt{7} \\
k_{212} + k_{112} & +\mathtt{A}s_{110} + \mathtt{4}s_{111} + \mathtt{4}s_{112} + \mathtt{1}s_{113} + \mathtt{6} \\
k_{213} + k_{113} & +\mathtt{1}s_{110} + \mathtt{8}s_{111} + \mathtt{3}s_{112} + \mathtt{3}s_{113} + \mathtt{7} \\
k_{220} + k_{120} + k_{100} & +\mathtt{5}s_{100} + \mathtt{1}s_{101} + \mathtt{C}s_{102} + \mathtt{5}s_{103} + \mathtt{4} \\
k_{221} + k_{121} + k_{101} & +\mathtt{2}s_{100} + \mathtt{2}s_{101} + \mathtt{1}s_{102} + \mathtt{F}s_{103} + \mathtt{3} \\
k_{222} + k_{122} + k_{102} & +\mathtt{A}s_{100} + \mathtt{4}s_{101} + \mathtt{4}s_{102} + \mathtt{1}s_{103} + \mathtt{5} \\
k_{223} + k_{123} + k_{103} & +\mathtt{1}s_{100} + \mathtt{8}s_{101} + \mathtt{3}s_{102} + \mathtt{3}s_{103} + \mathtt{2} \\
k_{230} + k_{130} + k_{110} & +\mathtt{5}s_{110} + \mathtt{1}s_{111} + \mathtt{C}s_{112} + \mathtt{5}s_{113} + \mathtt{6} \\
k_{231} + k_{131} + k_{111} & +\mathtt{2}s_{110} + \mathtt{2}s_{111} + \mathtt{1}s_{112} + \mathtt{F}s_{113} + \mathtt{7} \\
k_{232} + k_{132} + k_{112} & +\mathtt{A}s_{110} + \mathtt{4}s_{111} + \mathtt{4}s_{112} + \mathtt{1}s_{113} + \mathtt{6} \\
k_{233} + k_{133} + k_{113} & +\mathtt{1}s_{110} + \mathtt{8}s_{111} + \mathtt{3}s_{112} + \mathtt{3}s_{113} + \mathtt{7}
\end{array}
$$