Essential Algebraic Structure Within the AES

Sean Murphy and Matthew J.B. Robshaw

Information Security Group,
Royal Holloway, University of London,
Egham, Surrey, TW20 0EX, U.K.
s.murphy@rhul.ac.uk m.robshaw@rhul.ac.uk
mrobshaw@supanet.com

Abstract. One difficulty in the cryptanalysis of the Advanced Encryption Standard *AES* is the tension between operations in the two fields $GF(2^8)$ and $GF(2)$. This paper outlines a new approach that avoids this conflict. We define a new block cipher, the *BES*, that uses only simple algebraic operations in $GF(2^8)$. Yet the AES can be regarded as being identical to the BES with a restricted message space and key space, thus enabling the AES to be realised solely using simple algebraic operations in one field $GF(2^8)$. This permits the exploration of the AES within a broad and rich setting. One consequence is that AES encryption can be described by an extremely sparse overdetermined multivariate quadratic system over $GF(2^8)$, whose solution would recover an AES key.

Keywords: Advanced Encryption Standard, AES, Rijndael, BES, Algebraic Structure, (Finite) Galois Field, (Field) Conjugate, Multivariate Quadratic (MQ) Equations.

1 Introduction

Rijndael [7, 8] was chosen as the Advanced Encryption Standard (AES) and published as FIPS 197 [21] on 26 November 2001. The AES is carefully designed to resist standard block cipher attacks [1, 18]. Here we move our attention to a cipher that is an extension of AES, but which offers one particular advantage. All of the operations in this new cipher, the *BES*, are entirely described using very simple operations in $GF(2^8)$. Thus while the AES is embedded within the BES, and while the BES fully respects encryption with the AES, there are no $GF(2)^8$ operations.

The properties of this new cipher are intimately related to the properties of the AES, as the AES is essentially the BES with a restricted message and key space. The AES is, in essence, woven into the fabric of the BES. Yet, in many ways, the new cipher is easier to analyse. It is certainly easier to describe; one round of the cipher consists exclusively of inversion in $GF(2^8)$, matrix multiplication in $GF(2^8)$, and key addition in $GF(2^8)$.

By recasting the AES in this way we highlight some important structural features of the AES. We illustrate this with a differential-type effect in the BES that seems surprising given the design principles of the AES. Furthermore, we show that the AES preserves algebraic curves and that it can be expressed as a very simple system of multivariate quadratic equations over $GF(2^8)$. It is entirely possible that such a new approach might offer significant improvements to the cryptanalysis of the AES.

2 Previous Work and Notation

Throughout the AES process, Rijndael (the eventual AES) received considerable cryptanalytic attention [10, 12, 17]. The simplicity of Rijndael was emphasized by its designers [7, 8], and much work has concentrated on the structural properties of the cipher [9, 11, 15, 19, 20, 23, 24].

In this paper we introduce a new technique which further simplifies analysis of the AES. While the AES encryption process is typically described using operations on an array of bytes, we represent the data as column vectors, so matrix multiplication of such a column vector occurs on the left. We regard a byte as an element of the binary field defined by the irreducible "Rijndael" polynomial $X^8 + X^4 + X^3 + X + 1$. We denote this field by $\mathbf{F}$ and a root of this polynomial by θ, so

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

Each byte therefore represents a polynomial in θ and we adopt the convention that the most significant bit in a byte (the θ^7 term) is represented by the left-most, and most significant, bit of the hexadecimal representation of a byte.

The version of the AES we consider has a 128-bit or 16-byte message and key space, though our comments are more generally applicable. The new cipher BES has a 128-byte message and key space. We later define a restriction of the BES spaces to a subset of size 2^{128} that corresponds to the AES. We denote these three sets by $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{B_A}$ respectively, so

$\mathbf{A}$	State space of the AES	Vector space $\mathbf{F}^{16}$
$\mathbf{B}$	State space of the BES	Vector space $\mathbf{F}^{128}$
$\mathbf{B_A}$	Subset of $\mathbf{B}$ corresponding to $\mathbf{A}$	Subset of $\mathbf{F}^{128}$.

3 The Basic Structure of the AES

We refer to FIPS 197 [21] for a full description of the cipher, but we list the significant steps here. We concentrate our attentions on a typical round; the first and last rounds have a different (but related) form that is easily assimilated. We consider the basic version of the AES, which encrypts a 16-byte block using a 16-byte key with 10 encryption rounds.

The input to the AES round function can be viewed as a rectangular array of bytes or, equivalently, as a column vector of bytes. Throughout the encryption process this byte-structure is fully respected. The AES specification defines a round in terms of the following three transformations.

1. **The AES S-Box**. The value of each byte in the array is substituted according to a table look-up. This table look up $S[\cdot]$ is the combination of three transformations.
   (a) The input w is mapped to $x = w^{(-1)}$ where $w^{(-1)}$ is defined by

   $$w^{(-1)} = w^{254} = \begin{cases} w^{-1} & w \neq 0 \\ 0 & w = 0 \end{cases}$$

   Thus "AES inversion" is identical to standard field inversion in $\mathbf{F}$ for non-zero field elements with $0^{(-1)} = 0$.
   (b) The intermediate value x is regarded as a $GF(2)$-vector of dimension 8 and transformed using an (8×8) $GF(2)$-matrix L_A. The transformed vector $L_A \cdot x$ is then regarded in the natural way as an element of $\mathbf{F}$.
   (c) The output of the AES S-Box is $(L_A \cdot x) + \mathtt{63}$, where addition is with respect to $GF(2)$.
2. **The AES linear diffusion (mixing) layer.**
   (a) Each row of the array is rotated by a certain number of byte positions.
   (b) Each column of the array is considered to be an $\mathbf{F}$-vector, and a column $\mathbf{y}$ is transformed to the column $C \cdot \mathbf{y}$, where C is a (4×4) $\mathbf{F}$-matrix.
3. **The AES subkey addition.** Each byte of the array is added (with respect to $GF(2)$) to a byte from the corresponding array of round subkeys.

The additive constant (63) in the AES S-box can be removed by incorporating it within a (slightly) modified key schedule [19]. For simplicity, we use this description of the AES in this paper.

4 The Big Encryption System (BES)

We introduce a new iterated block cipher, the *Big Encryption System (BES)*, which operates on 128-byte blocks with a 16-byte key. Both the AES and the BES are defined in terms of bytes and we now describe the common mathematical framework for both ciphers.

Both the AES and the BES use a *state vector* of bytes, which is transformed by the basic operations within a round. In both cases, the plaintext is the input state vector while the ciphertext is the output state vector. As described in Section 2, the state spaces of the AES and the BES are the vector spaces $\mathbf{A} = \mathbf{F}^{16}$ and $\mathbf{B} = \mathbf{F}^{128}$ respectively. We now describe the basic techniques required to establish the relationship between the AES and the BES.

Inversion. The inversion operation is easily described. For $a \in \mathbf{F}$, it is identical to standard field inversion for non-zero field elements with $0^{(-1)} = 0$. For an n-dimensional vector $\mathbf{a} = (a_0, \ldots, a_{n-1}) \in \mathbf{F}^n$, we view inversion as a componentwise operation and set

$$\mathbf{a}^{(-1)} = (a_0^{(-1)}, \ldots, a_{n-1}^{(-1)}).$$

Vector conjugates. For any element $a \in \mathbf{F}$ we can define the *vector conjugate* of a, $\widetilde{\mathbf{a}}$, as the vector of the eight $GF(2)$-conjugates of a, so

$$\widetilde{\mathbf{a}} = \left(a^{2^0}, a^{2^1}, a^{2^2}, a^{2^3}, a^{2^4}, a^{2^5}, a^{2^6}, a^{2^7} \right).$$

We use a *vector conjugate mapping* ϕ from $\mathbf{F}^n$ to a subset of $\mathbf{F}^{8n}$. For $n = 1$ and $a \in \mathbf{F}$, we have

$$\widetilde{\mathbf{a}} = \phi(a) = \left(a^{2^0}, a^{2^1}, a^{2^2}, a^{2^3}, a^{2^4}, a^{2^5}, a^{2^6}, a^{2^7} \right).$$

This definition extends in the obvious way to a vector conjugate mapping ϕ from $\mathbf{F}^n$ to a subset of $\mathbf{F}^{8n}$. The n-dimensional vector $\mathbf{a} = (a_0, \ldots, a_{n-1}) \in \mathbf{F}^n$ is mapped to

$$\widetilde{\mathbf{a}} = \phi(\mathbf{a}) = (\phi(a_0), \ldots, \phi(a_{n-1})).$$

The vector conjugate mapping ϕ has desirable algebraic properties, namely that it is additive and preserves inverses, so

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

When each successive set of eight components in $\mathbf{a} \in \mathbf{F}^{8n}$ form an ordered set of $GF(2)$-conjugates, we say that $\mathbf{a}$ has the *conjugacy property.* Such vectors lie in $Im(\phi)$, and we can consider $\phi^{-1} : Im(\phi) \to \mathbf{F}^n$ as an *extraction mapping* which recovers the basic vector from a vector conjugate.

Embedding the AES state space in the BES state space. Any plaintext, ciphertext, intermediate text, or subkey for the AES is an element of the state space $\mathbf{A}$. Similarly, any plaintext, ciphertext, intermediate text, or subkey for the BES is an element of the state space $\mathbf{B}$.

We can use the vector conjugate map ϕ to embed any element of the AES state space $\mathbf{A}$ into the BES state space $\mathbf{B}$. We define

$$\mathbf{B_A} = \phi(\mathbf{A}) \subset \mathbf{B} \text{ to be the } \textit{AES subset of BES},$$

that is the embedded image of the AES state space in the BES state space. Elements of $\mathbf{B_A}$, that is embedded images of AES states or subkeys, have the vector conjugacy property. Furthermore, $\mathbf{B_A}$ is an additively closed set that also preserves inverses.

In the following sections we describe the cipher BES. This is done in such a way that the "commuting" diagram in Figure 1 is fully respected.

4.1 AES and BES

As previously described, we regard a state vector of the AES to be an element $\mathbf{a} \in \mathbf{A}$. We further regard each round subkey as an element $\mathbf{k}_i \in \mathbf{A}$. We do

$$\begin{array}{ccccc} & \mathbf{A} & \xrightarrow{\phi} & \mathbf{B_A} & \\ & \downarrow & & \downarrow & \\ k \rightarrow & \boxed{\text{AES}} & & \boxed{\text{BES}} & \leftarrow \phi(k) \\ & \downarrow & & \downarrow & \\ & \mathbf{A} & \xleftarrow{\phi^{-1}} & \mathbf{B_A} & \end{array}$$

Fig. 1. The relationship between the AES and the BES. The important feature of the BES is that it is defined exclusively using simple operations in one field, $GF(2^8)$.

not use the standard AES way of representing an element $\mathbf{a}$ as a square array. Instead we view the state vector $\mathbf{a}$ as a column vector, where

$$\mathbf{a} = \begin{array}{|c|c|c|c|} \hline a_{00} & a_{01} & a_{02} & a_{03} \\ \hline a_{10} & a_{11} & a_{12} & a_{13} \\ \hline a_{20} & a_{21} & a_{22} & a_{23} \\ \hline a_{30} & a_{31} & a_{32} & a_{33} \\ \hline \end{array} = (a_{00}, \ldots, a_{30}, a_{01}, \ldots, a_{31}, \ldots, a_{33})^T.$$

For the BES, we also view the state vector $\mathbf{b} \in \mathbf{B}$ as a column vector where

$$\mathbf{b} = (b_{000}, \ldots, b_{007}, b_{100}, \ldots, b_{107}, \ldots, \ldots, b_{330}, \ldots, b_{337})^T.$$

It should be obvious how we intend to use the embedding mapping ϕ. We set

$$\phi(a_{ij}) = (b_{ij0}, \ldots, b_{ij7}).$$

Each basic operation in a round of the AES describes a bijective mapping on $\mathbf{A}$. These can be readily replaced with similar operations in the BES. Our aim in doing this is to ensure that every operation (including the $GF(2)$-linear map from the AES S-box) is expressed using simple algebraic operations over $\mathbf{F}$.

Subkey addition. This is obvious for both the AES and the BES. For the AES we combine the state vector $\mathbf{a} \in \mathbf{A}$ with an AES subkey $(\mathbf{k}_A)_i \in \mathbf{A}$ by $\mathbf{a} \mapsto \mathbf{a} + (\mathbf{k}_A)_i$. We do exactly the same in BES and we combine the state vector $\mathbf{b} \in \mathbf{B}$ with a subkey $(\mathbf{k}_B)_i \in \mathbf{B}$ by $\mathbf{b} \mapsto \mathbf{b} + (\mathbf{k}_B)_i$. We consider the generation of the BES subkeys below.

S-box inversion. As inversion operates componentwise on bytes, it is just as easy to describe in the BES as the AES. In the AES, inversion can be viewed as a componentwise vector inversion of the state vector $\mathbf{a} \in \mathbf{A}$. Thus the AES inversion operation is given by $\mathbf{a} \mapsto \mathbf{a}^{(-1)}$. This can be translated in the obvious manner, and for $\mathbf{b} \in \mathbf{B}$, inversion in the BES is given by $\mathbf{b} \mapsto \mathbf{b}^{(-1)}$.

Row operation. The AES `RowShift` operation permutes the bytes in the array. Clearly this process can be considered as a transformation of the components of a column vector $\mathbf{a} \in \mathbf{A}$. It is straightforward to represent this transformation as multiplication of the state vector $\mathbf{a} \in \mathbf{A}$ by a (16×16) $\mathbf{F}$-matrix R_A. Consider the equivalent operation in the BES. It is equally straightforward to represent this transformation as multiplication of the state vector $\mathbf{b} \in \mathbf{B}$ by a (128×128) $\mathbf{F}$-matrix R_B. In moving from R_A to R_B we only need ensure that vector conjugates are moved as a single entity.

Column operation. The AES `MixColumn` operation is defined using a (4×4) $\mathbf{F}$-matrix C_A. A column $\mathbf{y} \in \mathbf{F}^4$ of the conceptual state array is transformed into a replacement column $\mathbf{z} \in \mathbf{F}^4$ by

$$\mathbf{z} = C_A \cdot \mathbf{y} = \begin{pmatrix} \theta & (\theta+1) & 1 & 1 \\ 1 & \theta & (\theta+1) & 1 \\ 1 & 1 & \theta & (\theta+1) \\ (\theta+1) & 1 & 1 & \theta \end{pmatrix} \cdot \mathbf{y} = \begin{pmatrix} \mathtt{02\ 03\ 01\ 01} \\ \mathtt{01\ 02\ 03\ 01} \\ \mathtt{01\ 01\ 02\ 03} \\ \mathtt{03\ 01\ 01\ 02} \end{pmatrix} \cdot \mathbf{y}.$$

We can readily view this as a transformation of the AES state space $\mathbf{A}$ by the (16×16) $\mathbf{F}$-matrix transformation Mix_A, where Mix_A is a block diagonal matrix with 4 identical blocks C_A, so $\mathrm{Mix}_A = Diag_4(C_A)$. Consider now the equivalent transformation within the BES. Our aim is to replicate the actions of the AES, but to maintain the condition that each byte in the AES is represented by a conjugate vector in the BES. To do this we consider eight versions of the matrix C_A. These versions are denoted by $C_B^{(k)}$ and they are defined as

$$C_B^{(k)} = \begin{pmatrix} \theta^{2^k} & (\theta+1)^{2^k} & 1 & 1 \\ 1 & \theta^{2^k} & (\theta+1)^{2^k} & 1 \\ 1 & 1 & \theta^{2^k} & (\theta+1)^{2^k} \\ (\theta+1)^{2^k} & 1 & 1 & \theta^{2^k} \end{pmatrix} \text{ for } k = 0, \ldots, 7,$$

so $C_B^{(0)} = C_A$. We note that $C_B^{(k)}$ is an MDS matrix, thereby offering certain diffusion properties [7, 8], and that if

$$\begin{gathered} (z_0, z_1, z_2, z_3)^T = C_A \cdot (y_0, y_1, y_2, y_3)^T \text{ then} \\ (z_0^{2^k}, z_1^{2^k}, z_2^{2^k}, z_3^{2^k})^T = C_B^{(k)} \cdot \left(y_0^{2^k}, y_1^{2^k}, y_2^{2^k}, y_3^{2^k}\right)^T. \end{gathered}$$

This provides a way of preserving the conjugacy property through the `MixColumn` transformation in the BES. The matrices $C_B^{(k)}$ can be used to define the (128×128) $\mathbf{F}$-matrix Mix_B that respects the vector conjugate embedding mapping $\phi : \mathbf{A} \to \mathbf{B}_A$, so the action of `MixColumn` on bytes in the AES is replicated by the action of Mix_B on vector conjugates in the BES. Under a simple basis re-ordering, Mix_B is a block diagonal matrix comprising 32 (4×4) MDS matrices.

The S-box $\boldsymbol{GF(2)}$-linear operation. In the AES, there is no easy way to represent this transformation of the state space $\mathbf{A}$ as a matrix multiplication However, in the BES there is a simple matrix representation of this operation.

The AES $GF(2)$-linear operation $\sigma_A : \mathbf{F}^{16} \to \mathbf{F}^{16}$ is defined using a function $f : \mathbf{F} \to \mathbf{F}$ that operates on each component of the state vector $\mathbf{a}$, so

$$\mathbf{a} = (a_{00}, \ldots, a_{33}) \mapsto \sigma_A(\mathbf{a}) = (f(a_{00}), \ldots, f(a_{33})).$$

In the AES specification, f is defined by considering $\mathbf{F} = GF(2^8)$ as the vector space $GF(2)^8$. The transformation f is then represented by the action of an (8×8) $GF(2)$ matrix L_A where

$$L_A = \begin{pmatrix} 1\,0\,0\,0\,1\,1\,1\,1 \\ 1\,1\,0\,0\,0\,1\,1\,1 \\ 1\,1\,1\,0\,0\,0\,1\,1 \\ 1\,1\,1\,1\,0\,0\,0\,1 \\ 1\,1\,1\,1\,1\,0\,0\,0 \\ 0\,1\,1\,1\,1\,1\,0\,0 \\ 0\,0\,1\,1\,1\,1\,1\,0 \\ 0\,0\,0\,1\,1\,1\,1\,1 \end{pmatrix}.$$

To accomplish the change from $GF(2^8)$ to $GF(2)^8$, the natural mapping ψ : $\mathrm{GF}(2^8) \to \mathrm{GF}(2)^8$ is used in the AES. The componentwise AES $GF(2)$-linear operation $f : \mathbf{F} \to \mathbf{F}$ is then defined by $f(a) = \psi^{-1}(L_A(\psi(a)))$ for $a \in \mathbf{F}$. It is the need for the maps ψ and ψ^{-1} that complicates analysis of the AES.

However, there exists a polynomial with co-efficients in $\mathbf{F}$ which interpolates $f : \mathbf{F} \to \mathbf{F}$. This polynomial may be regarded as an equivalent definition of f. Further, since f is an *additive* or *linearized* polynomial [16] on $\mathbf{F}$, it is necessarily described by a linear combination of conjugates. Thus we obtain

$$f(a) = \sum_{k=0}^{7} \lambda_k a^{2^k} \text{ for } a \in \mathbf{F},$$

$$\text{where } (\lambda_0, \lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6, \lambda_7) = (\mathtt{05}, \mathtt{09}, \mathtt{f9}, \mathtt{25}, \mathtt{f4}, \mathtt{01}, \mathtt{b5}, \mathtt{8f}).$$

This polynomial is essentially given in [8] as part of the derivation of the related S-Box interpolation polynomial [7, 8]. However, our interest is in separating out the $\mathbf{F}$-inversions from the rest of the $\mathbf{F}$-linear round function, since this separation seems algebraically the most natural.

The $GF(2)$-linear operation from the AES S-box can now be defined in the BES using an (8×8) $\mathbf{F}$-matrix. This matrix replicates the (AES) action of the $GF(2)$-linear map on the first byte of a vector conjugate set while ensuring that the property of *vector conjugacy* is preserved on the remaining bytes. We set

$$L_B = \begin{pmatrix}
(\lambda_0)^{2^0} & (\lambda_1)^{2^0} & (\lambda_2)^{2^0} & (\lambda_3)^{2^0} & (\lambda_4)^{2^0} & (\lambda_5)^{2^0} & (\lambda_6)^{2^0} & (\lambda_7)^{2^0} \\
(\lambda_7)^{2^1} & (\lambda_0)^{2^1} & (\lambda_1)^{2^1} & (\lambda_2)^{2^1} & (\lambda_3)^{2^1} & (\lambda_4)^{2^1} & (\lambda_5)^{2^1} & (\lambda_6)^{2^1} \\
(\lambda_6)^{2^2} & (\lambda_7)^{2^2} & (\lambda_0)^{2^2} & (\lambda_1)^{2^2} & (\lambda_2)^{2^2} & (\lambda_3)^{2^2} & (\lambda_4)^{2^2} & (\lambda_5)^{2^2} \\
(\lambda_5)^{2^3} & (\lambda_6)^{2^3} & (\lambda_7)^{2^3} & (\lambda_0)^{2^3} & (\lambda_1)^{2^3} & (\lambda_2)^{2^3} & (\lambda_3)^{2^3} & (\lambda_4)^{2^3} \\
(\lambda_4)^{2^4} & (\lambda_5)^{2^4} & (\lambda_6)^{2^4} & (\lambda_7)^{2^4} & (\lambda_0)^{2^4} & (\lambda_1)^{2^4} & (\lambda_2)^{2^4} & (\lambda_3)^{2^4} \\
(\lambda_3)^{2^5} & (\lambda_4)^{2^5} & (\lambda_5)^{2^5} & (\lambda_6)^{2^5} & (\lambda_7)^{2^5} & (\lambda_0)^{2^5} & (\lambda_1)^{2^5} & (\lambda_2)^{2^5} \\
(\lambda_2)^{2^6} & (\lambda_3)^{2^6} & (\lambda_4)^{2^6} & (\lambda_5)^{2^6} & (\lambda_6)^{2^6} & (\lambda_7)^{2^6} & (\lambda_0)^{2^6} & (\lambda_1)^{2^6} \\
(\lambda_1)^{2^7} & (\lambda_2)^{2^7} & (\lambda_3)^{2^7} & (\lambda_4)^{2^7} & (\lambda_5)^{2^7} & (\lambda_6)^{2^7} & (\lambda_7)^{2^7} & (\lambda_0)^{2^7}
\end{pmatrix}$$

We can now represent the entire set of $GF(2)$-linear operations in the AES with a (128×128) $\mathbf{F}$-matrix in the BES, Lin_B. Thus Lin_B is a block diagonal matrix with 16 identical blocks L_B, so $\mathrm{Lin}_B = Diag_{16}(L_B)$.

Key schedule. We can use the techniques from previous sections to describe the key schedule for the BES. This effectively replicates the actions of the AES key schedule, in which a 16-byte AES key $\mathbf{k}_A$ provides eleven subkeys, each in $\mathbf{A}$. In the BES, a 128-byte BES key $\mathbf{k}_B$ provides eleven subkeys, each in $\mathbf{B}$.

The key schedule in the AES uses the same operations as the AES encryption process, namely the $GF(2)$-linear map, componentwise inversion, byte rotation, and addition. Thus the key schedule can also be described using the same simple algebraic operations over $\mathbf{F}$. Whenever a constant is required in the AES, we use the embedded image of that constant in the BES. Whenever a byte in the AES has to be moved to a different position, we ensure that the corresponding vector conjugate is moved as a single entity in the BES. In this way, we ensure that if a BES key the conjugacy property, then so do all its derived subkeys. If the embedded image of the AES key $\mathbf{k}_A$ is the BES key $\mathbf{k}_B = \phi(\mathbf{k}_A)$, then $(\mathbf{k}_B)_i = \phi((\mathbf{k}_A)_i)$ for every round subkey, so the embedded images of an AES subkey sequence form a BES subkey sequence.

Round function of BES. We have now completely described a round of BES. If the inputs to the BES round function are $\mathbf{b} \in \mathbf{B}$ and subkey $(\mathbf{k}_B)_i \in \mathbf{B}$, then the BES round function is given by

$$\begin{aligned}\mathrm{Round}_B(\mathbf{b}, (\mathbf{k}_B)_i) &= \mathrm{Mix}_B\left(R_B\left(\mathrm{Lin}_B\left(\mathbf{b}^{(-1)}\right)\right)\right) + (\mathbf{k}_B)_i \\ &= M_B \cdot (\mathbf{b}^{(-1)}) + (\mathbf{k}_B)_i,\end{aligned}$$

where M_B is a (128×128) $\mathbf{F}$-matrix performing linear diffusion within the BES. Furthermore, if the inputs to the AES round function are $\mathbf{a} \in \mathbf{A}$ and subkey $(\mathbf{k}_A)_i \in \mathbf{A}$, then we have

$$\mathrm{Round}_A(\mathbf{a}, (\mathbf{k}_A)_i) = \phi^{-1}\left(\,\mathrm{Round}_B\left(\,\phi(\mathbf{a}), \phi((\mathbf{k}_A)_i)\,\right)\,\right).$$

4.2 The relationship between the AES and the BES

The BES is a 128-byte block cipher, which consists entirely of simple algebraic operations over $\mathbf{F}$. It has the property that $\mathbf{B_A}$, the set of embedded images of AES vectors, or equivalently the set of all BES inputs with the conjugacy property, is closed under the action of the BES round function. Furthermore, encryption in the BES fully respects encryption in the AES and the commuting diagram given in Figure 1 holds. Thus the BES restricted to $\mathbf{B_A}$ provides an alternative description of the AES and analysis of the BES may well provide additional insight into the AES.

5 Algebraic Observations on the BES

The round function of the BES, and hence essentially the AES, is given by

$$\mathbf{b} \mapsto M_B \cdot \mathbf{b}^{(-1)} + (\mathbf{k}_B)_i.$$

Thus a round of the AES is simply componentwise inversion and an affine transformation with respect to the *same* field $\mathbf{F} = GF(2^8)$. This suggests many possible areas for future investigation. We offer some preliminary observations.

5.1 Linear diffusion in BES

The linear diffusion $\mathbf{F}$-matrix M_B of the BES is a sparse matrix and can be analysed using similar techniques to those used in [19]. These were originally used to analyse the related linear diffusion $GF(2)$-matrix (denoted by M in [19]). However, this linear diffusion matrix (M) and the AES inversion are with respect to *different* fields.

The minimum polynomial of M_B is $(X+1)^{15}$, effectively the same as the minimum polynomial of M. In some sense, the BES is structurally no more complicated than the AES. Following [19], we find an $\mathbf{F}$-matrix P_B such that

$$R_B = P_B^{-1} \cdot M_B \cdot P_B,$$

where R_B is essentially the *Jordan* form of M_B. The matrix R_B has 112 rows with two ones and 16 rows with a single one while all other entries are zero. It is effectively the simple matrix R given in [19], and has similar interesting properties. The significant change here is that the properties of M_B are properties in $\mathbf{F}$ and not $GF(2)$. Such properties have the potential to interact directly with the inversion operation. Many of these properties involve *linear functionals* or *parity equations.* A *parity equation* is a row vector $\mathbf{e}^T$, and the parity of a vector $\mathbf{b} \in \mathbf{B}$ is $\mathbf{e}^T \cdot \mathbf{b} = \sum_{i=0}^{127} e_i \cdot b_i$. We note a few interesting properties.

- M_B has order 16.
- The columns of P_B form a basis for $\mathbf{B}$. In this basis, the action of the linear diffusion layer is given by the very simple matrix M_B.
- In particular, M_B fixes a subspace of $\mathbf{B}$ of dimension 16. The intersection with $\mathbf{B_A}$, the embedded AES state space, has 2^{16} elements.
- The rows of P_B^{-1} form linear functionals or parity equations (defined above) that always evaluate to 0 or 1 on $\mathbf{B_A}$ (by considering dual spaces).
- The set of parity equations whose value is fixed by M_B form a 16-dimensional vector subspace over $\mathbf{F}$.

These observations may seem somewhat abstract, but they do have important consequences. We discuss an example below in which these observations can be used to illustrate certain differential properties of the BES.

5.2 Related encryptions in the BES.

As noted in Section 5.1, it is possible to find parity equations whose values are fixed by M_B, the linear diffusion layer of the BES. One example is

$$\mathbf{e} = (\overbrace{\mathtt{b4}, \mathtt{fd}, \mathtt{17}, \mathtt{0e}, \mathtt{54}, \mathtt{a0}, \mathtt{f6}, \mathtt{52}}^{\text{repeat 16 times}}, \ldots)^T,$$

for which $\mathbf{e}^T = \mathbf{e}^T \cdot M_B$, so $\mathbf{e}^T \cdot \mathbf{b} = \mathbf{e}^T \cdot (M_B \cdot \mathbf{b})$. We now describe some interesting properties relating two plaintext-ciphertext pairs generated under related subkey sequences. These properties hold with probability one and so they can be appropriately extended to any number of rounds.

Suppose $\mathbf{p}$ has parity $p_e = \mathbf{e}^T \cdot \mathbf{p}$ under parity equation $\mathbf{e}^T$, so $t \cdot \mathbf{p}$ has parity tp_e for any $t \in \mathbf{F}$. Consider two state and subkey pairs $\mathbf{p}, \mathbf{k}_i \in \mathbf{B}$ and $t\mathbf{p}, t^{(-1)}\mathbf{k}_i \in \mathbf{B}$ $(t \neq 0, 1)$. A typical BES round function is given by

$$\begin{aligned} \mathbf{p} &\mapsto M_B \cdot \mathbf{p}^{(-1)} + \mathbf{k}_i, \text{ and} \\ t \cdot \mathbf{p} &\mapsto t^{-1} M_B \cdot \mathbf{p}^{(-1)} + t^{-1}\mathbf{k}_i. \end{aligned}$$

When we consider the effect of the BES round function on the parities, we obtain

$$\begin{aligned} p_e = (\mathbf{e}^T \cdot \mathbf{p}) &\mapsto \mathbf{e}^T \cdot M_B \cdot \mathbf{p}^{(-1)} + \mathbf{e}^T \cdot \mathbf{k}_i &&= \mathbf{e}^T \cdot \mathbf{p}^{(-1)} + \mathbf{e}^T \cdot \mathbf{k}_i \\ & &&= \mathbf{e}^T \cdot (\mathbf{p}^{(-1)} + \mathbf{k}_i), \\ tp_e = (\mathbf{e}^T \cdot t\mathbf{p}) &\mapsto \mathbf{e}^T \cdot t^{-1} M_B \cdot \mathbf{p}^{(-1)} + \mathbf{e}^T \cdot t^{-1}\mathbf{k}_i &&= \mathbf{e}^T \cdot t^{-1}\mathbf{p}^{(-1)} + \mathbf{e}^T \cdot t^{-1}\mathbf{k}_i \\ & &&= t^{-1}\mathbf{e}^T \cdot (\mathbf{p}^{(-1)} + \mathbf{k}_i). \end{aligned}$$

Thus if (p_e, tp_e) are the parities under $\mathbf{e}^T$, then after one round using subkeys $\mathbf{k}_i$ and $t^{-1}\mathbf{k}_i$ respectively, the respective parities are $(p'_e, t^{-1}p'_e)$ for some p'_e. Hence if we encrypt two plaintexts $(\mathbf{p}, t\mathbf{p})$ under different sets of related subkey sequences as detailed below, then we obtain two ciphertexts that are related by their parities c_e and tc_e.

Plaintext Parity	Subkey sequence	Ciphertext Parity
p_e	$\mathbf{k}_0, \mathbf{k}_1, \mathbf{k}_2, \mathbf{k}_3 \cdots, \mathbf{k}_9\mathbf{k}_{10}$	c_e
tp_e	$t\mathbf{k}_0, t^{-1}\mathbf{k}_1, t\mathbf{k}_2, t^{-1}\mathbf{k}_3, \cdots, t^{-1}\mathbf{k}_9, t\mathbf{k}_{10}$	tc_e

Differential-type effect in BES. We can increase the sophistication slightly and consider two pairs of plaintext $\mathbf{p}_0, \mathbf{p}_1 \in \mathbf{B}$ and $t\mathbf{p}_0, t\mathbf{p}_1 \in \mathbf{B}$. The difference in the first pair is $\mathbf{p}_0 + \mathbf{p}_1$ with parity $p_e = \mathbf{e}^T \cdot (\mathbf{p}_0 + \mathbf{p}_1)$, and similarly the parity of the difference in the second pair is tp_e.

Plaintext Difference Parity	Subkey sequence	Ciphertext Difference Parity
$\mathbf{e}^T \cdot (\mathbf{p}_0 + \mathbf{p}_1) = p_e$	$\mathbf{k}_0, \mathbf{k}_1, \cdots, \mathbf{k}_{10}$	c_e
$\mathbf{e}^T \cdot (t\mathbf{p}_0 + t\mathbf{p}_1) = tp_e$	$t\mathbf{k}_0, t^{-1}\mathbf{k}_1, \cdots, t\mathbf{k}_{10}$	tc_e

Suppose we encrypt the two pairs of plaintexts under two sets of related subkey sequences as detailed in the above table, then the plaintext and ciphertext difference parities have the same relationship, as shown in the above table. This relationship holds with probability one, so would be applicable for any number of rounds. Thus there exists a probability one differential effect under related subkey sequences in the BES in which every S-Box is active.

Relevance of these BES observations to the AES. These preliminary observations do not apply when specific details of the key schedule are considered. Even if they did, they would not apply directly to the AES for a rather subtle reason. If $(\mathbf{p}, t\mathbf{p}) \in \mathbf{B} \times \mathbf{B}$, then $(\mathbf{p}, t\mathbf{p}) \notin \mathbf{B_A} \times \mathbf{B_A}$; that is if $\mathbf{p}$ has the conjugacy property, then $t\mathbf{p}$ cannot have the conjugacy property ($t \neq 0, 1$). Thus, if $\mathbf{p}$ is an embedded AES plaintext, then $t\mathbf{p}$ cannot be an embedded AES plaintext.

However, these observations are very interesting for the light they shed on the AES design philosophy [7, 8]. As far as linear and differential cryptanalysis are concerned, the BES would be expected to have similar properties to the AES. In particular, the diffusion in both has the same reliance on MDS matrices. However in the BES, which is intricately entwined with the AES, we have exhibited a differential-like property that occurs with certainty even though every S-Box is active.

5.3 Preservation of algebraic curves

Each of the BES operations, namely "inversions" (ignoring 0-inversion for the moment) and affine transformations over $\mathbf{F}$, are simple algebraic transformations of $\mathbf{B}$. Thus each BES operation maps an algebraic curve defined on $\mathbf{B} = \mathbf{F}^{128}$ to an isomorphic algebraic curve. For a given key 128-bit key $\mathbf{k}$, more than half (about 53%) of AES plaintexts are encrypted without "inverting" 0 (since 160 inversions are performed). Let $\mathbf{A_k} \subset \mathbf{A}$ denote this set of AES plaintexts for key $\mathbf{k}$. If embedded plaintexts from $\mathbf{A_k}$ lie on a curve, then the corresponding embedded ciphertexts lie on an isomorphic curve over $\mathbf{F}$. Thus, the AES and the BES can be considered to preserve algebraically simple curves over $\mathbf{F}$ with a reasonable probability. In particular, the inversion and the affine transformation of the BES round function map quadratic forms over $\mathbf{F}$ to quadratic forms over $\mathbf{F}$, so the AES can be described using a very simple system of multivariate quadratic equations over $\mathbf{F}$. We consider the consequences of this observation below.

6 Multivariate Quadratic Equations

We now demonstrate that recovering an AES key is equivalent to solving particular systems of extremely sparse multivariate quadratic equations by expressing a BES (and hence an AES) encryption as such a system. The problem of solving such systems of equations lies at the heart of several public key cryptosystems [3, 22], and there has been some progress in providing solutions to such problems [4, 5, 14]. Recently, Courtois and Pieprzyk [6] have suggested the use of a system of multivariate quadratic equations over $GF(2)$ to analyse the AES. However, such a $GF(2)$-system derived directly from the AES is far more complicated than the $\mathbf{F}$-system derived from the BES.

6.1 A simple multivariate quadratic system for the AES

We first establish the notation that we need. We denote the plaintext and ciphertext by $\mathbf{p} \in \mathbf{B}$ and $\mathbf{c} \in \mathbf{B}$ respectively, and the state vectors before and after the i^{th} invocation of the inversion layer by $\mathbf{w}_i \in \mathbf{B}$ and $\mathbf{x}_i \in \mathbf{B}$ $(0 \le i \le 9)$ respectively. A BES encryption is then described by the following system of equations:

$$\begin{aligned}
\mathbf{w}_0 &= \mathbf{p} + \mathbf{k}_0, & \\
\mathbf{x}_i &= \mathbf{w}_i^{(-1)} & \text{for } i = 0, \ldots, 9, \\
\mathbf{w}_i &= M_B \mathbf{x}_{i-1} + \mathbf{k}_i & \text{for } i = 1, \ldots, 9, \\
\mathbf{c} &= M_B^* \mathbf{x}_9 + \mathbf{k}_{10}, &
\end{aligned}$$

where $M_B^* = R_B \cdot Lin_B = Mix_B^{-1} \cdot M_B$, since the final round in the BES (equivalently the AES) does not use the `MixColumn` operation.

We now consider these equations componentwise. We first denote the matrix M_B by (α) and the matrix M_B^* by (β). We represent the $(8j+m)^{\text{th}}$ component of $\mathbf{x}_i$, $\mathbf{w}_i$ and $\mathbf{k}_i$ by $x_{i,(j,m)}$, $w_{i,(j,m)}$ and $k_{i,(j,m)}$ respectively. We can now express the previous set of equations in the following way:

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

We assume that 0-inversion does not occur as part of the encryption or the key schedule. This assumption is true for 53% of encryptions and 85% of 128-bit keys, and even if the assumption is invalid, only a very few of the following equations are incorrect. Under the stated assumption, the system of equations can be written as:

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

We thus obtain a collection of simultaneous multivariate quadratic equations which fully describe a BES encryption. These are given for $j = 0, \ldots, 15$ and $m = 0, \ldots, 7$ by:

$$\begin{aligned}
0 &= w_{0,(j,m)} + p_{(j,m)} + k_{0,(j,m)}, & \\
0 &= x_{i,(j,m)} w_{i,(j,m)} + 1 & \text{for } i = 0, \ldots, 9, \\
0 &= w_{i,(j,m)} + k_{i,(j,m)} + \textstyle\sum_{(j',m')} \alpha_{(j,m),(j',m')} x_{i-1,(j',m')} & \text{for } i = 1, \ldots, 9, \\
0 &= c_{(j,m)} + k_{10,(j,m)} + \textstyle\sum_{(j',m')} \beta_{(j,m),(j',m')} x_{9,(j',m')}. &
\end{aligned}$$

A BES encryption can therefore be described as a multivariate quadratic system using 2688 equations over $\mathbf{F}$, of which 1280 are (extremely sparse) quadratic equations and 1408 are linear (diffusion) equations. These equations comprise 5248 terms, made from 2560 state variables and 1408 key variables.

When we consider an AES encryption embedded in the BES framework, we obtain more multivariate quadratic equations because the embedded state variables of an AES encryption are in $\mathbf{B_A}$ and possess the conjugacy property. We thus obtain the following very simple multivariate quadratic equations for $j = 0, \ldots, 15$ and $m = 0, \ldots, 7$ (where $m+1$ is interpreted modulo 8). We divide these equations into linear equations and multivariate quadratic equations.

$$\begin{array}{ll}
0 = w_{0,(j,m)} + p_{(j,m)} + k_{0,(j,m)}; & \\
0 = w_{i,(j,m)} + k_{i,(j,m)} + \sum_{(j',m')} \alpha_{(j,m),(j',m')} x_{i-1,(j',m')} & \text{for } i = 1, \ldots, 9, \\
0 = c_{(j,m)} + k_{10,(j,m)} + \sum_{(j',m')} \beta_{(j,m),(j',m')} x_{9,(j',m')}. &
\end{array}$$

$$\begin{array}{ll}
0 = x_{i,(j,m)} w_{i,(j,m)} + 1 & \text{for } i = 0, \ldots, 9, \\
0 = x^2_{i,(j,m)} + x_{i,(j,m+1)} & \text{for } i = 0, \ldots, 9, \\
0 = w^2_{i,(j,m)} + w_{i,(j,m+1)} & \text{for } i = 0, \ldots, 9.
\end{array}$$

An AES encryption can therefore be described as an overdetermined multivariate quadratic system using 5248 equations over $\mathbf{F}$, of which 3840 are (extremely sparse) quadratic equations and 1408 are linear equations. These encryption equations comprise 7808 terms, made from 2560 state variables and 1408 key variables. Furthermore, the AES key schedule can be expressed as a similar multivariate quadratic system. In its most sparse form, the key schedule system uses 2560 equations over $\mathbf{F}$, of which 960 are (extremely sparse) quadratic equations and 1600 are linear equations. These key schedule equations comprise 2368 terms made from the 2048 variables, of which 1408 are basic key variables and 640 are auxiliary variables. We can, of course, immediately reduce the sizes of these multivariate quadratic systems by using the linear equations to substitute for state and key variables, though the resulting system is slightly less sparse.

6.2 Potential attack techniques

It is clear that an efficient method for the solution of this type of multivariate quadratic system would give a cryptanalysis of the AES with potentially very few plaintext-ciphertext pairs. While there is some connection to work on interpolation attacks [13], techniques such as *relinearisation* [14] or the *extended linearisation* or *XL* algorithm [5] have been specifically developed for the solution of such systems. A simple overview of these techniques is given below.

- Generate equations of higher degree from the original equations by multiplying the original equations by certain other terms or equations.
- Regard the generated system of equations of higher degree as linear combinations of formal terms.
- If there are more linearly independent equations than terms, solve the linear system.

The recently proposed *extended sparse linearisation* or *XSL* algorithm [6] is a modification of the XL algorithm that attempts to solve the types of multivariate quadratic systems that can occur in iterated block ciphers. A discussion of the use of the XSL algorithm on the AES multivariate quadratic $GF(2)$-system is given in [6]. The AES **F**-system derived from the BES is far simpler, which would suggest that the XSL algorithm would solve this **F**-system far faster (2^{100} AES encryptions) than the $GF(2)$-system. However, the estimate given for the number of linearly independent equations generated by the XSL technique [6] appears to be inaccurate [2].

It is obvious that much urgent research is required on the solution of AES multivariate quadratic systems over **F** to see what new cryptanalytic approaches and attacks are possible. In particular, refinements to XL-type techniques and the applicability of sparse matrix techniques seem to be important topics for future work. It is certainly important to know the degree and size of linearly soluble systems generated from the AES multivariate quadratic systems. If the degree and size of such a generated system is too small, then attacks on the AES might be possible. We note that the BES representation of the AES gives other simple quadratic equations over **F**, such as $x_{i,(j,m+1)}w_{i,(j,m)} = x_{i,(j,m)}$ or $x_{i,(j,m+2)}w_{i,(j,m)} = x_{i,(j,m+1)}x_{i,(j,m)}$. These can be used to build other simple multivariate quadratic systems over **F** for the AES. Indeed, the first of these equations is essentially used to construct the $GF(2)$ system for the AES given in [6]. We can also use simple higher degree equations over **F** to build other simple multivariate systems for the AES. It is clear from this brief discussion that many aspects of the AES representation over **F** remain to be investigated.

6.3 Implications for the AES

The cryptanalysis of the AES is equivalent to the solution of some particular system of extremely sparse multivariate quadratic equations over **F**. The analysis of the AES as a complicated multivariate quadratic system over $GF(2)$ by Courtois and Pieprzyk [6] is related to the problem of finding such a solution. Most of the other published security results on the AES are concerned with demonstrating that bit-level linear and differential techniques do not compromise the AES. However, from an algebraic viewpoint, such techniques are trace ($\mathbf{F} \rightarrow GF(2)$) function techniques, and trace function techniques are not normally employed in the solution of multivariate systems. It is arguable that an important aspect of the security of the AES, namely the solubility of an extremely sparse multivariate quadratic system over **F**, is yet to be explored.

7 Conclusions

In this paper we have introduced a novel interpretation of the AES as being embedded in a new cipher, the BES. However, the BES does not necessarily inherit security properties we might have expected from the AES. Furthermore, the BES has a simple algebraic round function consisting solely of a componentwise inversion and and a highly structured affine transformation over the same field $GF(2^8)$. Indeed, this alternative description of the AES is mathematically much simpler than the original specification. One consequence is that the security of the AES is equivalent to the solubility of certain extremely sparse multivariate quadratic systems over $GF(2^8)$.

Acknowledgements

We would like to thank Fred Piper, Simon Blackburn and Don Coppersmith for some interesting and useful discussions about this paper.

References

1. E. Biham and A. Shamir. *Differential Cryptanalysis of the Data Encryption Standard*. Springer–Verlag, New York, 1993.
2. D. Coppersmith. Personal communication, 30 April 2002.
3. N. Courtois, L. Goubin, and J. Patarin. Quartz, 128-bit long digital signatures. In D. Naccache, editor, Proceedings of *Cryptographers' Track RSA Conference 2001*, LNCS 2020, pages 282–297, Springer–Verlag, 2001.
4. N. Courtois, L. Goubin, W. Meier, and J. Tacier. Solving underdefined systems of multivariate quadratic equations. In D. Paillier, editor, Proceedings of *Public Key Cryptography 2002*, LNCS 2274, pages 211-227, Springer-Verlag, 2002.
5. N. Courtois, A. Klimov, J. Patarin, and A. Shamir. Efficient algorithms for solving overdefined systems of multivariate polynomial equations. In B. Preneel, editor, Proceedings of *Eurocrypt 2000*, LNCS 1807, pages 392–407, Springer-Verlag, 2000.
6. N. Courtois and J. Pieprzyk. Cryptanalysis of block ciphers with overdefined systems of equations. IACR eprint server `www.iacr.org`, April 2002.
7. J. Daemen and V. Rijmen. AES Proposal: Rijndael (Version 2). NIST AES website `csrc.nist.gov/encryption/aes`, 1999.
8. J. Daemen and V. Rijmen. *The Design of Rijndael: AES—The Advanced Encryption Standard*. Springer–Verlag, 2002.
9. J. Daemen and V. Rijmen. Answers to "New Observations on Rijndael". NIST AES website `csrc.nist.gov/encryption/aes`, August 2000.
10. N. Ferguson, J. Kelsey, B. Schneier, M. Stay, D. Wagner, and D. Whiting. Improved cryptanalysis of Rijndael. In B. Schneier, editor, Proceedings of *Fast Software Encryption 2000*, LNCS , pages 213–230, Springer-Verlag, 2000.
11. N. Ferguson, R. Shroeppel, and D. Whiting. A simple algebraic representation of Rijndael. In S. Vaudenay and A. Youssef, editors, Proceedings of *Selected Areas in Cryptography*, LNCS, pages 103–111, Springer-Verlag, 2001.
12. H. Gilbert and M. Minier. A collision attack on seven rounds of Rijndael. *Third AES Conference*, NIST AES website `csrc.nist.gov/encryption/aes`, April 2000.
13. T. Jakobsen and L.R. Knudsen. The interpolation attack on block ciphers. In E. Biham, editor, Proceedings of *Fast Software Encryption 1997*, LNCS 1267, pages 28–40, Springer-Verlag, 1997.
14. A. Kipnis and A. Shamir. Cryptanalysis of the HFE Public Key Cryptosystem be Relinearization. In M.Wiener, editor, Proceedings of *Crypto '99*, LNCS 1666, pages 19–30, Springer-Verlag, 1999.
15. L. Knudsen and H. Raddum. Recommendation to NIST for the AES. *NIST second round comment*, NIST AES website `csrc.nist.gov/encryption/aes/`, 2000.
16. R. Lidl and H. Niederreiter. *Introduction to Finite Fields and Their Applications*. Cambridge University Press, 1984.

17. S. Lucks. Attacking seven rounds of Rijndael under 192-bit and 256-bit keys. In Proceedings of *Third AES Conference* and also via NIST AES website `csrc.nist.gov/encryption/aes`, April 2000.
18. M. Matsui. Linear cryptanalysis method for DES cipher. In T. Helleseth, editor, Proceedings of *Eurocrypt '93*, LNCS 765, pages 386–397, Springer-Verlag, 1994.
19. S. Murphy and M.J.B. Robshaw. New observations on Rijndael. NIST AES website `csrc.nist.gov/encryption/aes`, August 2000.
20. S. Murphy and M.J.B. Robshaw. Further comments on the structure of Rijndael. NIST AES website `csrc.nist.gov/encryption/aes`, August 2000.
21. National Institute of Standards and Technology. Advanced Encryption Standard. FIPS 197. 26 November 2001.
22. J. Patarin. Hidden field equations (HFE) and isomorphisms of polynomials (IP): Two new families of asymmetric algorithms. In U. Maurer, editor, Proceedings of *Eurocrypt '96*, LNCS 1070, pages 33–48, Springer-Verlag, 1996.
23. R. Schroeppel. Second round comments to NIST. *NIST second round comment*, NIST AES website `csrc.nist.gov/encryption/aes/`, 2000.
24. R. Wernsdorf. The round functions of Rijndael generate the alternating group. In V. Rijmen, editor, Proceedings of *Fast Software Encryption*, LNCS, Springer–Verlag, to appear.