Extended Linear Cryptanalysis and Extended Piling-up Lemma

Qin Li and Serdar Boztas

Abstract. **In this paper, we extend the idea of piling-up lemma and linear cryptanalysis applied to symmetric-key block ciphers. We also examine this new method of Extended Linear Cryptanalysis on two-round Rijndael, which is designed to be immune to linear cryptanalysis attack. Even though our results do not show much surprise on two-round Rijndael, the effects on other block ciphers remain open questions.**

I. Introduction

Linear cryptanalysis was introduced by Matsui [1] in 1994. It utilises the highly unbalanced probability distribution of binary linear approximation expressions involving plaintext bits and ciphertext bits of iterated ciphers. The unbalanced probability distribution of the binary linear approximation function is generated by firstly seeking unevenly distributed binary linear approximation function involving an individual S-box, the only non-linear component of the cipher. Then those approximation functions are concatenated together over all rounds of the cipher such that all terms involving intermediate bits can be eliminated and the final linear approximation function involves only plaintext and the ciphertext bits. Moreover, it must still be unbalanced for this attack to work. This attack has been generalised in many ways such as in [3,4]. In this paper we extend linear cryptanalysis from an algebraic perspective.

Traditionally, it is over Galois Field GF(2) that linear cryptanalysis is performed. However, we have managed to extend the attack to extension fields of GF(2). To sum up, *Extended Linear Cryptanalysis* seeks linear functions over extension fields of GF(2).

Qin Li is with the School of Mathematical and Geospatial Sciences RMIT University, Melbourne, VIC 3001 Australia

Serdar Boztas is with the School of Mathematical and Geospatial Sciences RMIT University, Melbourne, VIC 3001 Australia

I. Description of Classical Linear Cryptanalysis

A. Analyze an individual S-box

Linear cryptanalysis has to firstly analyze the linearity of the only non-linear component, S-box. An S-box is typically a non-linear bijective mapping with n bits input and n bits output. Let $S:\{0,1\}^n \to \{0,1\}^n$ be an n-bit S-box. Linear cryptanalysis essentially seeks a desirable binary linear approximation function

$$f(\boldsymbol{x}) = \boldsymbol{a} \bullet \boldsymbol{x} \oplus \boldsymbol{b} \bullet S(\boldsymbol{x}) \quad (1)$$

where $\boldsymbol{a}$, $\boldsymbol{b}$ are constant vectors which are regarded as input and output masks and $\boldsymbol{x}$ is a vector which represents the input of the S-box, and $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{x} \in \{0,1\}^n$. The operation $\bullet$ represents dot product of two vectors and $\oplus$ denotes the XOR operation.

A desirable binary linear approximation function $f(\boldsymbol{x})$ is one where the probability that $f(\boldsymbol{x}) = 0$ is as far from ½ as possible. The difference between this probability and 1/2 is referred to as probability bias or bias and denoted as ε, i.e.,

$$\Pr[f(\mathbf{x}) = 0] = 1/2 + \varepsilon \quad (2)$$

B. Piling-Up Lemma

Once multiple binary linear approximation functions over all rounds with their probability distributions are discovered, they are combined together. Additionally, Matsui employed a useful tool [1] to calculate the bias of the combined binary linear approximation function.

For n independent random binary variables $X_1, X_2, \cdots, X_n$ whose corresponding probability biases for $X_i = 0$ (i = 1,2, … , n) are $\varepsilon_1, \varepsilon_2, \cdots, \varepsilon_n$,

$$\Pr[X_1 \oplus X_2 \oplus \cdots \oplus X_n = 0] = 1/2 + 2^{n-1}\varepsilon_1\varepsilon_2\cdots\varepsilon_n \quad (3)$$

Or equivalently, the bias of the combined binary function

$X_1 \oplus X_2 \oplus \cdots \oplus X_n = 0$ equals $2^{n-1}\varepsilon_1\varepsilon_2\cdots\varepsilon_n$.

C. Combine binary linear approximation function over whole cipher and extract partial key bits

Now we are able to concatenate multiple binary linear approximation functions over different rounds and to calculate the probability bias of the new binary linear approximation function. When the combined binary linear function over whole cipher is generated, it can be represented in the form

$$f_1(\boldsymbol{x}) \oplus f_2(\boldsymbol{x}) \oplus \cdots \oplus f_n(\boldsymbol{x}) = \boldsymbol{a} \bullet \boldsymbol{x} \oplus \boldsymbol{b} \bullet E_K(\boldsymbol{x}) \quad (4)$$

where $\boldsymbol{a}$, $\boldsymbol{b}$ are constant vectors which are regarded as input and output masks and $\boldsymbol{x}$ is a vector represents the input of the block cipher, and $E_K(\boldsymbol{x})$, in the form of vector, is the output of the second last round of the cipher encrypted using key K. The operation $\bullet$ and $\oplus$ represent dot product and XOR respectively. All other intermediate values are eliminated.

By applying piling-up lemma, (4) has probability bias of $\pm 2^{n-1}\varepsilon_1\varepsilon_2\cdots\varepsilon_n$. This bias specifies the vulnerability of the cipher. It can be exploited to find partial key bits of the cipher.

II. Extended Linear Cryptanalysis

It can be seen from previous description of linear cryptanalysis that binary linear cryptanalysis is performed over vector spaces $\{0,1\}^n$, where n is the number of the input/output bits of the cipher being analyzed. See Fig. 1.

The input bits, $\boldsymbol{x}$, and the output bits, $S(\boldsymbol{x})$, are represented as vectors, as well as the constant masks values $\boldsymbol{a}$ and $\boldsymbol{b}$. The binary linear approximation function is indeed a sum of a dot product of $\boldsymbol{a}$ and $\boldsymbol{x}$ and another dot product of $\boldsymbol{b}$ and $S(\boldsymbol{x})$.

On the contrary, Extended Linear Cryptanalysis is carried out over GF(2^n), where n denotes the number of the input/output bits of the S-box. The linear approximation function is computed in finite fields instead of in vector spaces over GF(2). The input and the output of the S-box, as well as the constant masks a and b, are represented as elements of the finite field of GF(2^n). The probability distribution, of course, can be computed over any subfield of GF(2^n).

A. Trace function

In this paper, we apply the trace function over finite fields as the linear approximation function—note that by using a pair of dual bases, this can be reduced to an inner product over the relevant extension field.

Definition 1 *Let K be a finite field with q elements and F be an extension field of K with q^m elements. For $a \in F$, the trace $Tr_{F/K}(a)$ of a over K is defined by*

$$Tr_{F/K}(a) = a + a^q + \cdots + a^{q^{m-1}} \quad (5)$$

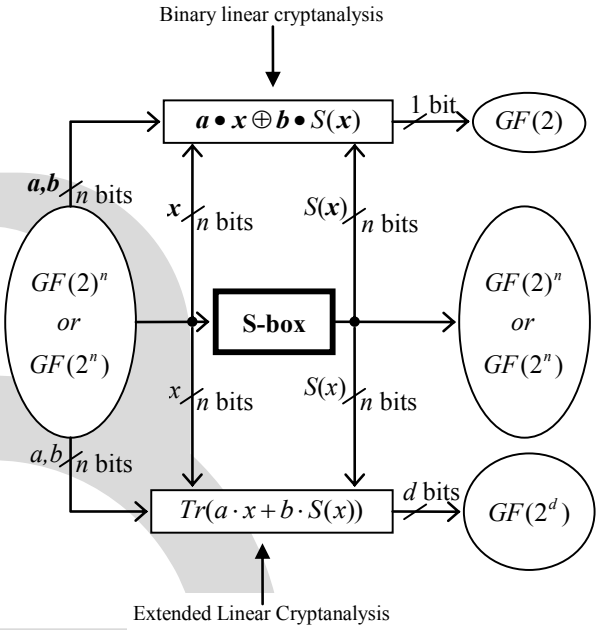

Fig.1. Difference between binary and Extended Linear Cryptanalysis

The trace function has the following well-known properties (see [5]) which we shall use in this paper:

$Tr_{F/K}(a) \in K$ for all $a \in F$ (6)

$Tr_{F/K}(a+b) = Tr_{F/K}(a) + Tr_{F/K}(b)$ for all $a, b \in F$ (7)

B. Analyzing individual S-box of Rijndael

In this paper, we use Rijndael [2] to demonstrate Extended Linear Cryptanalysis. In Rijndael, there is only one S-box. it has eight bits input and output. Like the Rijndael description [2], the input and output of the S-box are represented as elements of finite field GF(2^8) in this paper. The input bit string $x_7\,x_6\,x_5\,x_4\,x_3\,x_2\,x_1\,x_0$ represents the input byte x where $x = x_7\alpha^7 + x_6\alpha^6 + x_5\alpha^5 + x_4\alpha^4 + x_3\alpha^3 + x_2\alpha^2 + x_1\alpha + x_0$,

and the output bit string $y_7\, y_6\, y_5\, y_4\, y_3\, y_2\, y_1\, y_0$ represents the output bytes y where $y = y_7\alpha^7 + y_6\alpha^6 + y_5\alpha^5 + y_4\alpha^4 + y_3\alpha^3 + y_2\alpha^2 + y_1\alpha + y_0$. Both x and y are the elements of the finite field GF(256) with the irreducible polynomial $\alpha^8+\alpha^4+\alpha^3+\alpha+1$ as the reduction polynomial.

Because finite field GF(256) has three proper subfields, namely GF(2), GF(4) and GF(16), we will analyse the probability distributions of trace functions over the three subfields:

$$Tr_{GF(256)/GF(2)}(a \cdot x + b \cdot y) \quad (8)$$

$$Tr_{GF(256)/GF(4)}(a \cdot x + b \cdot y) \quad (9)$$

$$Tr_{GF(256)/GF(16)}(a \cdot x + b \cdot y) \quad (10)$$

where a, b are constant input and output mask values and x, y are the input and its corresponding output of S-box of Rijndael, respectively. a, b, x and y are all elements of GF(256) and the addition and multiplication operations are also under GF(256).

For each of them, we are going to search for the best mask values (a, b) such that the probability distribution of each trace function is the most unbalanced.

The probability distribution of a trace function with a fixed (a, b) pair can be computed by replacing x and y with the input and corresponding output value of S-box in the form of elements of GF(256) and then going through all 2^8=256 input/output values. For each trace function among (8), (9) and (10), there are 2^{16} possible options for the constant mask value pairs (a,b). We need figure out the best (a,b) mask pairs with which the trace function produces the highest unbalanced probability distribution. The results of this paper were obtained by using *Magma* [6] as a software platform to search for desirable (a, b) mask pairs.

When probability distribution of the trace function is over extension field of GF(2), such as GF(4) and GF(16), however, to determine the probability bias becomes more complex because the domain of the trace function has more than two values. In this paper, we simply refer to the maximum amount by which the probability of the trace function deviates from the uniform probability as the probability bias.

Our experiments show that there are 1275 different (a,b) mask pairs with which the trace function (8) produces the most unbalanced probability over GF(2). The absolute value of the highest bias is $16/256=2^{-4}$. There are three (a,b) constant mask pairs with which the trace function (9) produces the most unbalanced probability over GF(4). The absolute value of the highest bias is $24/256=1.5\times2^{-4}$. There are 15 (a,b) constant mask pairs with which the trace function (10) produces the most unbalanced probability over GF(16). The absolute value of the highest bias is $24/256\approx1.1\times2^{-4}$.

By comparing the three results it is found that when the Extended Linear Cryptanalysis is performed over GF(4) and GF(16), the biases we obtain are higher than over GF(2).

C. *Extended Piling-up Lemma*

After analyzing the probability biases of the trace function in the form of Tr(ax+by) based on the input and the corresponding output of an individual S-box, we will show (when multiple trace functions are added together) how the probability distribution of a new trace function is obtained. Here we introduce a new mathematical model–*The Extended Piling-Up Lemma*–to compute the probability distribution of a combined trace function. The Extended Piling-Up Lemma is essentially a generalized piling-up Lemma [1]. We will introduce this model starting with the following straightforward example:

Let U_1, U_2 be random variables whose sample space is GF(4). Alternatively, the domain of random variables U_1, U_2 are {0, 1, z, z+1} where 0, 1, z and z+1 are all the entire elements of GF(4). Let

$$U_1 \sim (\Pr_1(0), \Pr_1(1), \Pr_1(z), \Pr_1(z+1))$$
$$U_2 \sim (\Pr_2(0), \Pr_2(1), \Pr_2(z), \Pr_2(z+1))$$

denote the probability distributions of the random variables U_1 and U_2 respectively, where $\Pr_i(X)$ is the probability that U_i=X.

Applying our knowledge of finite fields and probability theory, we get the following result (essentially a convolution of probability distributions):

$$\Pr(U_1+U_2=0) = \Pr_1(0)\times\Pr_2(0) + \Pr_1(1)\times\Pr_2(1) + \Pr_1(z)\times\Pr_2(z) + \Pr_1(z+1)\times\Pr_2(z+1) \quad (11)$$

$$\Pr(U_1+U_2=1) = \Pr_1(0)\times\Pr_2(1) + \Pr_1(1)\times\Pr_2(0) + \Pr_1(z)\times\Pr_2(z+1) + \Pr_1(z+1)\times\Pr_2(z) \quad (12)$$

$$\Pr(U_1+U_2=z) = \Pr_1(0)\times\Pr_2(z) + \Pr_1(1)\times\Pr_2(z+1) + \Pr_1(z)\times\Pr_2(0) + \Pr_1(z+1)\times\Pr_2(1) \quad (13)$$

$$\Pr(U_1+U_2=z+1)=\Pr_1(0)\times\Pr_2(z+1)+\Pr_1(1)\times\Pr_2(z)+\Pr_1(z)\times\Pr_2(1)+\Pr_1(z+1)\times\Pr_2(0) \quad (14)$$

The above computation can be represented as matrix multiplication as follows. Let the 1×4 matrix

$(\Pr_{1+2}(0),\Pr_{1+2}(1),\Pr_{1+2}(z),\Pr_{1+2}(z+1))$

denote the probability distribution of U_1+U_2, then

$$\begin{pmatrix}\Pr_{1+2}(0) & \Pr_{1+2}(1) & \Pr_{1+2}(z) & \Pr_{1+2}(z+1)\end{pmatrix}= \begin{pmatrix}\Pr_1(0) & \Pr_1(1) & \Pr_1(z) & \Pr_1(z+1)\\ \Pr_1(1) & \Pr_1(0) & \Pr_1(z+1) & \Pr_1(z)\\ \Pr_1(z) & \Pr_1(z+1) & \Pr_1(0) & \Pr_1(1)\\ \Pr_1(z+1) & \Pr_1(z) & \Pr_1(1) & \Pr_1(0)\end{pmatrix}\times\begin{pmatrix}\Pr_2(0)\\ \Pr_2(1)\\ \Pr_2(z)\\ \Pr_2(z+1)\end{pmatrix} \quad (15)$$

where $\begin{pmatrix}\Pr_1(0) & \Pr_1(1) & \Pr_1(z) & \Pr_1(z+1)\\ \Pr_1(1) & \Pr_1(0) & \Pr_1(z+1) & \Pr_1(z)\\ \Pr_1(z) & \Pr_1(z+1) & \Pr_1(0) & \Pr_1(1)\\ \Pr_1(z+1) & \Pr_1(z) & \Pr_1(1) & \Pr_1(0)\end{pmatrix}$ is a square matrix generated from the row vector $\begin{pmatrix}\Pr_1(0) & \Pr_1(1) & \Pr_1(z) & \Pr_1(z+1)\end{pmatrix}$ by using the Cayley table of the additive group of GF(4).

The general scheme to create such $2^n\times2^n$ matrix from a 2^n elements row vector can also be described recursively as:

1. Pair up two consecutive elements of the row vector from one side to another;
2. Generate square matrices for every pair by using the following rule (Cayley table of (GF(2),+)):

   $\begin{pmatrix}a & b\end{pmatrix}\mapsto\begin{pmatrix}a & b\\ b & a\end{pmatrix}$;
3. Concatenate the new generated square matrices from last step in order that the first row of the matrix concatenated is identical as the initial row vector;
4. Treat the new concatenated matrix as a row vector whose elements are the square matrices from the last step;
5. Repeat step 1 to 4 until the new concatenated matrix itself is a square matrix.

In the following section of this paper, all square matrices generated from row vectors are created by using this scheme.

Now, we can extend this example to compute the probability distribution of the sum of finitely many random variables:

Let $U_i\sim(\Pr_i(0), \Pr_i(1), \Pr_i(z), \Pr_i(z+1))$ be the probability distributions of the random variables U_i whose sample space is GF(4), where $i\in\{0,1,\cdots,n\}$.

Let the 1×4 matrix

$(\Pr_{1+2+\ldots+n}(0), \Pr_{1+2+\ldots+n}(1), \Pr_{1+2+\ldots+n}(z), \Pr_{1+2+\ldots+n}(z+1))$

be the probability distribution of the sum of the random variables $U_1+U_2+\ldots+U_n$, then

$$\begin{pmatrix}\Pr_{1+2+\ldots+n}(0) & \Pr_{1+2+\ldots+n}(1) & \Pr_{1+2+\ldots+n}(z) & \Pr_{1+2+\ldots+n}(z+1)\end{pmatrix}= \begin{pmatrix}\Pr_1(0) & \Pr_1(1) & \Pr_1(z) & \Pr_1(z+1)\\ \Pr_1(1) & \Pr_1(0) & \Pr_1(z+1) & \Pr_1(z)\\ \Pr_1(z) & \Pr_1(z+1) & \Pr_1(0) & \Pr_1(1)\\ \Pr_1(z+1) & \Pr_1(z) & \Pr_1(1) & \Pr_1(0)\end{pmatrix}\times \begin{pmatrix}\Pr_2(0) & \Pr_2(1) & \Pr_2(z) & \Pr_2(z+1)\\ \Pr_2(1) & \Pr_2(0) & \Pr_2(z+1) & \Pr_2(z)\\ \Pr_2(z) & \Pr_2(z+1) & \Pr_2(0) & \Pr_2(1)\\ \Pr_2(z+1) & \Pr_2(z) & \Pr_2(1) & \Pr_2(0)\end{pmatrix}\times\cdots\times\begin{pmatrix}\Pr_n(0)\\ \Pr_n(1)\\ \Pr_n(z)\\ \Pr_n(z+1)\end{pmatrix} \quad (16)$$

When the sample space of the random variables is GF(16), the probability distribution of the sum of the variables U_i where $i\in\{0,1,\cdots,n\}$ is

$$(\Pr_{1+2+\cdots+n}(1),\Pr_{1+2+\cdots+n}(2),\cdots,\Pr_{1+2+\cdots+n}(z^3+z^2+z+1))= \begin{pmatrix}\Pr_1(0) & \Pr_1(1) & \cdots & \Pr_1(z^3+z^2+z+1)\\ \Pr_1(1) & \Pr_1(0) & \cdots & \Pr_1(z^3+z^2+z)\\ \vdots & \vdots & \ddots & \vdots\\ \Pr_1(z^3+z^2+z+1) & \Pr_1(z^3+z^2+z) & \cdots & \Pr_1(0)\end{pmatrix}\times \begin{pmatrix}\Pr_2(0) & \Pr_2(1) & \cdots & \Pr_2(z^3+z^2+z+1)\\ \Pr_2(1) & \Pr_2(0) & \cdots & \Pr_2(z^3+z^2+z)\\ \vdots & \vdots & \ddots & \vdots\\ \Pr_2(z^3+z^2+z+1) & \Pr_2(z^3+z^2+z) & \cdots & \Pr_2(0)\end{pmatrix}\times \cdots\times\begin{pmatrix}\Pr_n(0)\\ \Pr_n(1)\\ \vdots\\ \Pr_n(z^3+z^2+z+1)\end{pmatrix} \quad (17)$$

Using the same systematic method, we can compute the probability distribution of the sum of any finite number of random variables U_i whose sample space is a finite field F, where $i\in\{0,1,\cdots,n\}$,. However, we do not pursue this further here.

D. Combine linear approximation trace functions over two-round Rijndael variant

Similar to binary linear cryptanalysis, Extended Linear Cryptanalysis is also very sensitive to the structure of the block cipher. Here we use a two-round *Rijndael* variant as an example to show how to concatenate multiple approximation trace functions over the whole cipher. The structure of two-round Rijndael variant is shown in the two-dimensional graphical demonstration of Fig. 2, where

$\bigoplus$ represents *AddRoundKey* operation applied on one byte. Equivalently, it is a bitwise XOR operation applied on two 8-bit inputs, generating one 8-bit output.

S represents *SubBytes* operation applied on one byte. Equivalently, it is an S-box with 8-bit input and output.

M represents *MixColumns* operation applied on four bytes. It is sometimes called a D-box with 32-bit input and output.

↓ represents flow of one byte (8 bits) of plaintext, ciphertext, sub-key or intermediate state. Moreover, *ShiftRows* is also implemented by crossover of lines with arrows.

$P_{j,k}$, $C_{j,k}$ correspondingly denote one plaintext or ciphertext byte (8 bits) at the *j*-th row and the *k*-th column of the state. $K_{i,j,k}$ represents one round-key byte (8 bits) at the *j*-th row and the *k*-th column of the state in the *i*-th round. Note that $K_{0,j,k}$ represents the initial key byte preceding the two-round. $X_{i,j,k}$ and $Y_{i,j,k}$ correspondingly denotes an input and an output byte of an S-box where subscript *i*, *j* and *k* represents the byte is at the *j*-th row and the *k*-th column of the state in the *i*-th round. Similarly, $Y_{i,j,k}$ and $Z_{i,j,k}$ correspondingly denotes an input and an output byte of a D-box where subscript *i*, *j* and *k* represents the byte is at the *j*-th row and the *k*-th column of the state in the *i*-th round.

It is clear from Fig. 2 that we combine approximation trace functions based on the inputs and outputs of S-boxes $S_{1,1,1}$, $S_{2,1,1}$, $S_{2,2,1}$, $S_{2,3,1}$ and $S_{2,4,1}$ to create a new trace function. The new combined trace function will involve only the first byte of the plaintext and all bytes of the ciphertext.

For each one of the involving S-boxes, we can get a best linear approximation trace function over each of GF(2), GF(4) and GF(16). Then we can concatenate these five trace functions as following:

$$\begin{aligned}&Tr_{1,1,1}(a_{1,1,1}X_{1,1,1}+b_{1,1,1}Y_{1,1,1})+Tr_{2,1,1}(a_{2,1,1}X_{2,1,1}+b_{2,1,1}Y_{2,1,1})+\\&Tr_{2,2,1}(a_{2,2,1}X_{2,2,1}+b_{2,2,1}Y_{2,2,1})+Tr_{2,3,1}(a_{2,3,1}X_{2,3,1}+b_{2,3,1}Y_{2,3,1})+\\&Tr_{2,4,1}(a_{2,4,1}X_{2,4,1}+b_{2,4,1}Y_{2,4,1})=\\&Tr_{1,1,1+2,1,1+2,2,1+2,3,1+2,4,1}(a_{1,1,1}X_{1,1,1}+b_{1,1,1}Y_{1,1,1}+\\&a_{2,1,1}X_{2,1,1}+b_{2,1,1}Y_{2,1,1}+a_{2,2,1}X_{2,2,1}+b_{2,2,1}Y_{2,2,1}+\\&a_{2,3,1}X_{2,3,1}+b_{2,3,1}Y_{2,3,1}+a_{2,4,1}X_{2,4,1}+b_{2,4,1}Y_{2,4,1})\end{aligned}\quad(18)$$

From the description of Rijndael [2] and Fig. 2, we know:

$$\begin{aligned}&X_{1,1,1}=K_{0,1,1}+P_{1,1}\\&Y_{1,1,1}=0E\cdot Z_{1,1,1}+0B\cdot Z_{1,2,1}+0D\cdot Z_{1,3,1}+09\cdot Z_{1,4,1}\\&Z_{1,1,1}=X_{2,1,1}+K_{1,1,1},\quad Z_{1,2,1}=X_{2,2,1}+K_{1,2,1}\\&Z_{1,3,1}=X_{2,3,1}+K_{1,3,1},\quad Z_{1,4,1}=X_{2,4,1}+K_{1,4,1}\\&Y_{2,1,1}=0E\cdot Z_{2,1,1}+0B\cdot Z_{2,2,1}+0D\cdot Z_{2,3,1}+09\cdot Z_{2,4,1}\\&Z_{2,1,1}=C_{1,1}+K_{2,1,1},\quad Z_{2,2,1}=C_{2,1}+K_{2,2,1}\\&Z_{2,3,1}=C_{3,1}+K_{2,3,1},\quad Z_{2,4,1}=C_{4,1}+K_{2,4,1}\\&Y_{2,2,1}=09\cdot Z_{2,1,4}+0E\cdot Z_{2,2,4}+0B\cdot Z_{2,3,4}+0D\cdot Z_{2,4,4}\\&Z_{2,1,4}=C_{1,4}+K_{2,1,4},\quad Z_{2,2,4}=C_{2,4}+K_{2,2,4}\\&Z_{2,3,4}=C_{3,4}+K_{2,3,4},\quad Z_{2,4,4}=C_{4,4}+K_{2,4,4}\\&Y_{2,3,1}=0D\cdot Z_{2,1,3}+09\cdot Z_{2,2,3}+0E\cdot Z_{2,3,3}+0B\cdot Z_{2,4,3}\\&Z_{2,1,3}=C_{1,3}+K_{2,1,3},\quad Z_{2,2,3}=C_{2,3}+K_{2,2,3}\\&Z_{2,3,3}=C_{3,3}+K_{2,3,3},\quad Z_{2,4,3}=C_{4,3}+K_{2,4,3}\\&Y_{2,4,1}=0D\cdot Z_{2,1,2}+09\cdot Z_{2,2,2}+0E\cdot Z_{2,3,2}+0B\cdot Z_{2,4,2}\\&Z_{2,1,2}=C_{1,2}+K_{2,1,2},\quad Z_{2,2,2}=C_{2,2}+K_{2,2,2}\\&Z_{2,3,2}=C_{3,2}+K_{2,3,2},\quad Z_{2,4,2}=C_{4,2}+K_{2,4,2}\end{aligned}\quad(19)$$

where 0E, 0B, 0D, 09 are hexadecimal representations of the constant elements in GF(256). They are respectively:

$$\begin{aligned}0E&=\alpha^3+\alpha^2+\alpha\\0B&=\alpha^3+\alpha+1\\0D&=\alpha^3+\alpha^2+1\\09&=\alpha^3+1\end{aligned}\quad(20)$$

If

$$\begin{aligned}a_{2,1,1}&=b_{1,1,1}\cdot 0E\\a_{2,2,1}&=b_{1,1,1}\cdot 0B\\a_{2,3,1}&=b_{1,1,1}\cdot 0D\\a_{2,4,1}&=b_{1,1,1}\cdot 09\end{aligned}\quad(21)$$

then the combined trace function (21) equals

$$\begin{aligned}&Tr_{1,1,1+2,1,1+2,2,1+2,3,1+2,4,1}(a_{1,1,1}P_{1,1}\\&+b_{2,1,1}\cdot 0E\cdot C_{1,1}+b_{2,1,1}\cdot 0B\cdot C_{2,1}+b_{2,1,1}\cdot 0D\cdot C_{3,1}+b_{2,1,1}\cdot 09\cdot C_{4,1}\\&+b_{2,2,1}\cdot 09\cdot C_{1,4}+b_{2,2,1}\cdot 0E\cdot C_{2,4}+b_{2,2,1}\cdot 0B\cdot C_{3,4}+b_{2,2,1}\cdot 0D\cdot C_{4,4}\\&+b_{2,3,1}\cdot 0D\cdot C_{1,3}+b_{2,3,1}\cdot 09\cdot C_{2,3}+b_{2,3,1}\cdot 0E\cdot C_{3,3}+b_{2,3,1}\cdot 0B\cdot C_{4,3}\\&+b_{2,4,1}\cdot 0B\cdot C_{1,2}+b_{2,4,1}\cdot 0D\cdot C_{2,2}+b_{2,4,1}\cdot 09\cdot C_{3,2}+b_{2,4,1}\cdot 0E\cdot C_{4,2}\\&+b_{2,1,1}\cdot 0E\cdot K_{2,1,1}+b_{2,1,1}\cdot 0B\cdot K_{2,2,1}+b_{2,1,1}\cdot 0D\cdot K_{2,3,1}+b_{2,1,1}\cdot 09\cdot K_{2,4,1}\\&+b_{2,2,1}\cdot 09\cdot K_{2,1,4}+b_{2,2,1}\cdot 0E\cdot K_{2,2,4}+b_{2,2,1}\cdot 0B\cdot K_{2,3,4}+b_{2,2,1}\cdot 0D\cdot K_{2,4,4}\\&+b_{2,3,1}\cdot 0D\cdot K_{2,1,3}+b_{2,3,1}\cdot 09\cdot K_{2,2,3}+b_{2,3,1}\cdot 0E\cdot K_{2,3,3}+b_{2,3,1}\cdot 0B\cdot K_{2,4,3}\\&+b_{2,4,1}\cdot 0B\cdot K_{2,1,2}+b_{2,4,1}\cdot 0D\cdot K_{2,2,2}+b_{2,4,1}\cdot 09\cdot K_{2,3,2}+b_{2,4,1}\cdot 0E\cdot K_{2,4,2})\end{aligned}\quad(22)$$

It is worth noting that the combined trace function (22) involves only the first byte of the plaintext and all bytes of the ciphertext of two-core-round Rijndael. All other terms that are left are merely constants, given the initial key of the cipher is fixed.

Now we can make use of previous results of probability distribution of individual S-box and Extended Piling-up Lemma to compute the probability bias, over GF(2), GF(4) and GF(16), of (22), which is an extended linear approximation function for the two-round Rijndael variant.

The results show that the possible highest biases of the combined trace function (22) over GF(2), GF(4) and GF(16) are 2^{-16}, 1.5×2^{-16} and $2^{-17.22}$ correspondingly. Obviously, unlike binary linear cryptanalysis, we have more choices to approximate a cipher using Extended Linear Cryptanalysis.

In this paper, we have introduced Extended Linear Cryptanalysis, as well as Extended Piling-up Lemma. It is of interest to apply this idea on other block ciphers, or to improve the result on Rijndael.

CONCLUSIONS AND POSSIBLE FUTURE DIRECTIONS

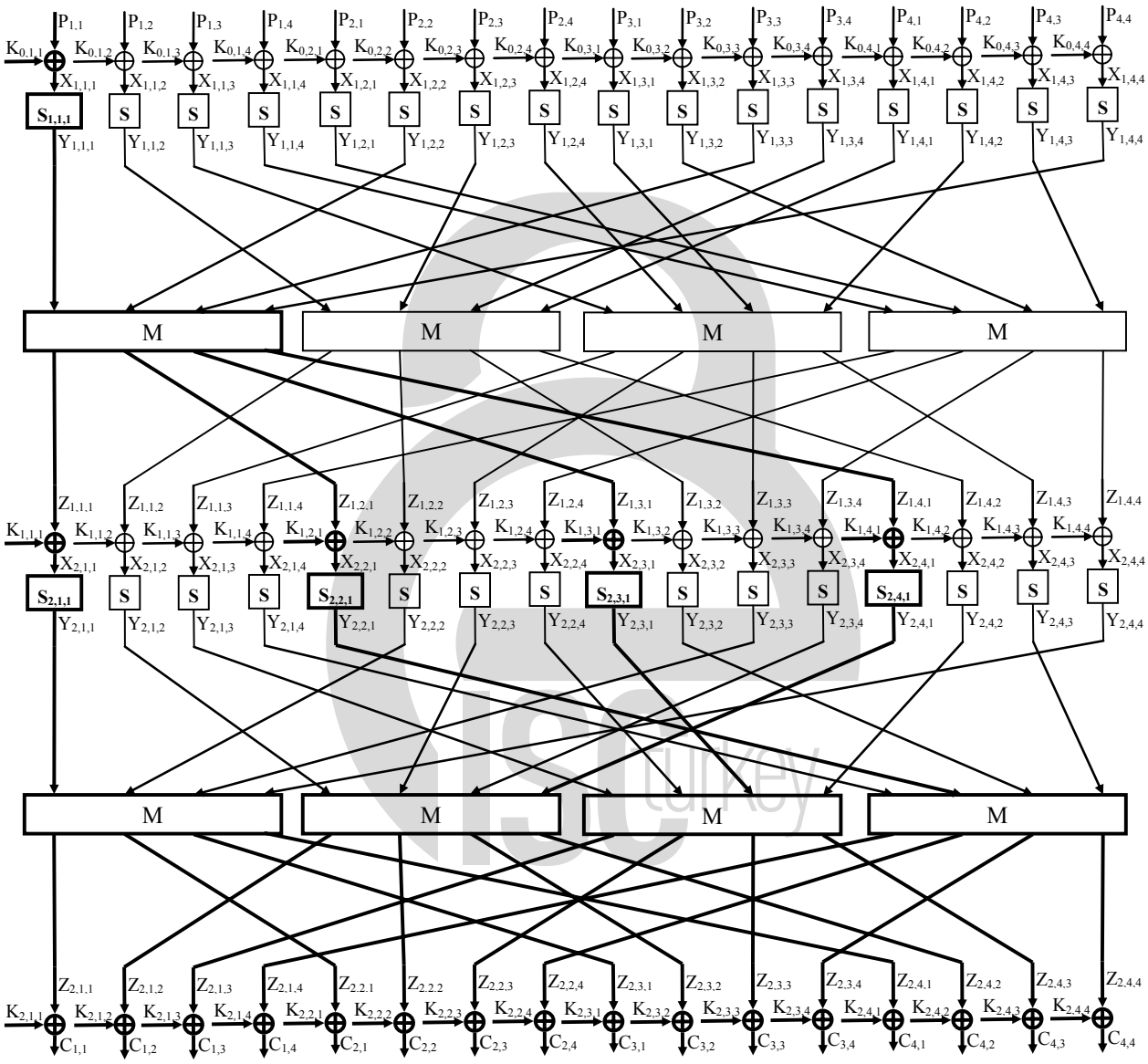

Fig. 2 Two dimensional graphical demonstration of two-round Rijndael variant

Besides, how best to measure the probability bias of a random variable whose sample space has more than two elements is an interesting problem. Some recent work in [7] has investigated this question in general.

REFERENCES

[1] M. Matsui, "Linear Cryptanalysis Method for DES Cipher," *EUROCRYPT '93, LNCS, Vol.* 765, Springer-Verlag, pp.386-397, 1994

[2] Joan Daemen and Vincent Rijmen. *The design of Rijndael: AES – The Advanced Encryption Standard*. Springer, 2002

[3] C. Harpes, G. Kramer and J. Massey, "A Generalization of Linear Cryptanalysis and the Applicability of Matsui's Piling-up Lemma" *EUROCRYPT '95*, *LNCS*, vol.921, Springer-Verlag, pp.24-38, 1995

[4] C. Harpes and J.Massey, "Partitioning Cryptanalysis", Fast Software Encryption 4, LNCS, vol.1267, Springer-Verlag, pp.13-27, 1997

[5] R. Lidl and H. Niederreiter, *Finite Fields*, Cambridge University Press, 1997

[6] Magma, http://magma.maths.usyd.edu.au/magma/, 2007

[7] T. Baignères, P. Junod, and S. Vaudenay, "How Far Can We Go Beyond Linear Cryptanalysis?" *ASIACRYPT 2004, LNCS*, vol. 3329, pp. 432-450, 2004.