2-1-2005

Topics in the Theory and Practice of the AES Algorithm

Nelson A. Carella
Pace University, Computer Science and Information Systems Department

SCHOOL OF COMPUTER SCIENCE AND INFORMATION SYSTEMS

TECHNICAL REPORT

Number 211, February 2005

Topics in the Theory and Practice of the AES Algorithm

Nelson A. Carella

PACE UNIVERSITY

Nelson A. Carella, a mathematician who hails from the City University of New York, adjuncts in the Information Systems Department at Pace University in Manhattan.

Topics in the Theory and Practice of the AES Algorithm

by

Nelson A. Carella

Table of Contents

Appendices

1. Introduction

This note is an introduction to the theory and practice of the Advanced Encryption Standard (the AES algorithm) and related algebraic block ciphers. The symmetric block cipher AES, called Rijndael, was certified as FIPS-PUB 197 in October, 2000. The new standard is intended to replace the block cipher DES specified in the standard FIPS PUB 46-1, 2, 1977, 1988, which was decommissioned in 2004. The cryptosystem AES-32b was designed to resist all forms of known cryptoanalysis such as linear, differential, and cyclic analysis. It is expected to have a lifetime of about 30 years as its predecessor. Further it was designed to have efficient implementations on various microprocessor platforms.

The first section introduces preliminary information and a block diagram of the AES algorithm. Subsequent sections provide specific descriptions of the AES algorithm, and its components. Theoretical and practical material used in the analysis and design of AES algorithm and block ciphers is provided in the Appendix. Each section in the Appendix is essentially independent of the rest and it is geared to supply deeper coverage of specific topic. The material in the Appendix is useful in the analysis of the current AES cipher, related ciphers, and future designs.

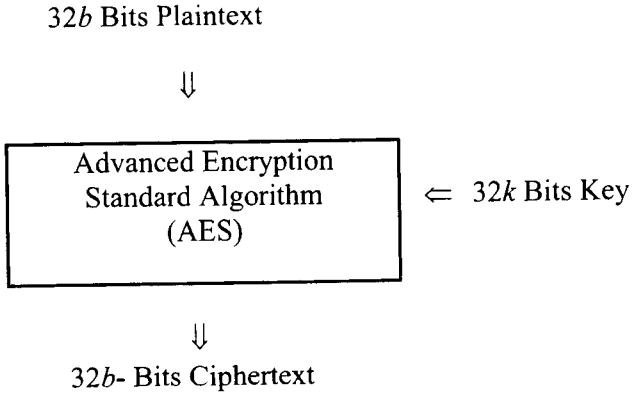
2. The AES Algorithm

The plaintext length, ciphertext length and key length of the AES-32b algorithm are

$$32b \text{ bits plaintext, } 32b \text{ bits ciphertext, and } 32k \text{ bits key}$$

respectively, and the parameters $b, k \in \{ 4, 5, 6, 7, 8 \}$. The standard FIPS-PUB 197 specifies only three versions AES-128, AES-192, and AES-256. These versions have plaintext and key of lengths 128, 192 and 256, a total of nine combinations are possible.

The diagram of the encryption function is the following.



The descriptions and analysis of the AES-32*b* algorithm given here are for the smallest plain text length and key length of 128 bits. This corresponds to the parameters $b = k = 4$. However, the analysis of all cases (with different values of the parameters b, k) are essentially the same mutatis mutandis.

The software and hardware implementations of AES algorithm are series of bit/byte operations and look-up table substitutions. An implementation can be realized at the bit level and there is no need for any finite field arithmetic of any sort. The importance of transferring the bit/byte level description of AES algorithm to an algebraic description has to do with the optimization, analysis, and further development of the algorithm. Once an algebraic description is on hand, the vast knowledge of finite rings and finite fields theory is immediately available to analyze it and improve it. This is analogous to machine level programming as oppose to high level language programming, (it is far easier to analyze and debug a program written in a high level language than a program written in machine language).

Several algebraic descriptions of the AES-32*b* algorithm are already known. Among these are
(1) Matrix Description,
(2) Polynomial Description,
(3) Isomorphic Description.

A few of these descriptions will be introduced here.

3. Matrix Description of the AES Algorithm

The matrix description of the AES algorithm is given in the original source [DR01]. This description is presented here in slightly different form as vector/matrix equation. A generalization of the matrix description to a larger encryption system appears in [MY02].

Bytes Space

The *bytes space* is a direct product of 8 copies of the smallest finite field $\mathbf{F}_2 = \{ 0, 1 \}$. A byte is viewed as a vector $b = (b_7, b_6, b_5, b_4, b_3, b_2, b_1, b_0)$ of eight bits in the bytes space

$$\mathbf{F}_2^8 = \mathbf{F}_2 \times \mathbf{F}_2 \times \mathbf{F}_2 \times \mathbf{F}_2 \times \mathbf{F}_2 \times \mathbf{F}_2 \times \mathbf{F}_2 \times \mathbf{F}_2 .$$

Equivalently, a byte is also viewed as an element of the finite field $\mathbf{F}_{2^8}$. More generally, any 8-bit arrangement will be used as equivalent.

State Space

The *states space* is a direct product of 16 byte spaces $\mathbf{F}_2^{8\times 16} = \mathbf{F}_2^8 \times \cdots \times \mathbf{F}_2^8$. A state is viewed as a 16-byte vector

$$x = (x_{15}, x_{14}, x_{13}, x_{12}, x_{11}, x_{10}, x_9, x_8, x_7, x_6, x_5, x_4, x_3, x_2, x_1, x_0),$$

where $x_i \in \mathbf{F}_{2^8}$. The equivalent byte spaces is $\mathbf{F}_{2^8}^{16} = \mathbf{F}_{2^8} \times \cdots \times \mathbf{F}_{2^8}$. The original manuscript identifies a 16-byte block with a 4 × 4 array of bytes

$$A = \begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} x_0 & x_4 & x_8 & x_{12} \\ x_1 & x_5 & x_9 & x_{13} \\ x_2 & x_6 & x_{10} & x_{14} \\ x_3 & x_7 & x_{11} & x_{15} \end{bmatrix}. \quad (1)$$

However, the vector notation is more common in the mathematical analysis of this algorithm. All these arrangements of a 128-bit block will be used interchangeably.

3.1 S-Box

The S-box used in the AES algorithm is a function $\sigma : \mathbf{F}_{2^8} \to \mathbf{F}_{2^8}$ (or $\sigma : \mathbf{F}_2^8 \to \mathbf{F}_2^8$) acting on the individual coordinate of the vector $\boldsymbol{x} = (x_{15}, \ldots, x_0)$. Some of the essential properties of an s-box are:

(1) Invertible,
(2) Nonlinear,
(3) Correlation resistance,
(4) Large cycle lengths.

These properties are derived from the current knowledge of functions on discrete structures, see the section on *Functions in Finite Fields* in the Appendix. The S-box is defined by $\sigma(x) = g \circ f(x) = ax^{-1} + b$. This is a composition of the nonlinear function $f(x) = x^{-1}$, and a linear function $g(x) = ax + b$ over the finite field $\mathbf{F}_{2^8}$, (or bytes space $\mathbf{F}_2^8$). This combination is intended to resist linear, differential, and cyclic analysis, etc.

Polynomial Description of the S-Box.

The evaluation of $\sigma(x)$ takes place in the finite field $\mathbf{F}_{2^8}$ and the finite ring $\mathbf{F}_2[z]/(z^8 + 1)$. Specifically (2)

$$\sigma(x) = [(z^4 + z^3 + z^2 + z + 1)(x^{254} \bmod (z^8 + z^4 + z^3 + z + 1)) + z^6 + z^5 + z + 1] \bmod (z^8 + 1).$$

A nonlinear operation in the finite field $\mathbf{F}_2[z]/(f(z))$ follows by a linear operation in the finite ring $\mathbf{F}_2[z]/(z^8 + 1)$. The constants are $a = z^4 + z^3 + z^2 + z + 1$, $b = z^6 + z^5 + z + 1 \in \mathbf{F}_{2^8}$ or $a =$ 1F, $b = 63$ in hexadecimal notation.

The inverse function is usually computed via either the power map $f(x) = x^{-1} = x^{q-2}$, or the Euclidean algorithm. For the finite field of $q = 2^8$ elements, the former case is given by $x^{-1} = x^{254}$.

Example of S-Box Calculation. Compute the value of $\sigma(x)$ at $x = z^7 + z^5 + z^3 \in \mathbf{F}_{2^8}$, (or $x =$ A8 in hexadecimal notation).

Since $(z^7 + z^5 + z^3)^{-1} = z^5 + z^2 + z$ in $\mathbf{F}_{2^8}$, the evaluation gives

$$\begin{aligned}\sigma(x) = ax^{-1} + b &= [(z^4 + z^3 + z^2 + z + 1)(z^7 + z^5 + z^3)^{-1} + (z^6 + z^5 + z + 1)] \bmod (z^8 + 1) \\ &= [(z^4 + z^3 + z^2 + z + 1)(z^5 + z^2 + z) + (z^6 + z^5 + z + 1)] \bmod (z^8 + 1) \\ &= z^7 + z^6 + z.\end{aligned}$$

In hexadecimal notation, this is $z^7 + z^6 + z =$ C2. All the values of the s-box are listed on the 16×16 substitution table.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	Ca	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

S-Box Substitution Table 1.

In general, an s-box is a nonlinear function $\sigma : \mathbf{F}_q \to \mathbf{F}_q$ acting on the coordinates of the state vectors. It is selected based on certain cryptographic criteria such as nonlinearity, high degree, diffusion, etc. Distinct s-boxes can be used in each coordinates of the state vector $\boldsymbol{x} = (x_n, \ldots, x_0)$. For example, the AES algorithm use the same s-box in each coordinates, but the DES algorithm uses distinct s-box in each coordinate.

Partial Matrix Description of the S-Box

The nonlinear map $f(x) = x^{-1}$ as any nonlinear function does not have a matrix representation. But the linear part $g(x) = ax + b$ does has a matrix component. Given $x = a_7z^7 + \cdots + a_1z + a_0 \in \mathbf{F}_{2^8}$, the inverse is $x^{-1} = (a_7z^7 + \cdots + a_1z + a_0)^{-1} = b_7z^7 + \cdots + b_1z + b_0 \in \mathbf{F}_{2^8}$. Thus the composition $\sigma(x) = ax^{-1} + b$ has the partial matrix representation

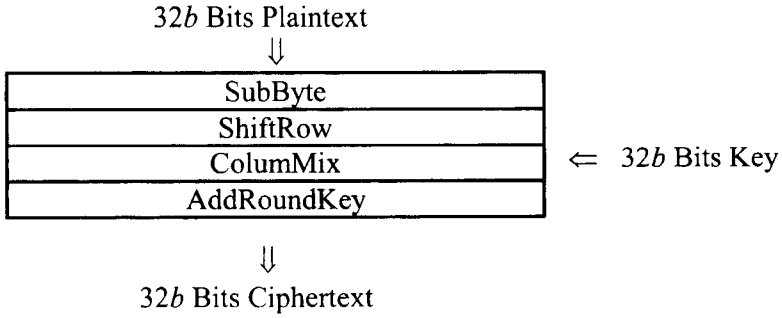
$$\begin{bmatrix} c_7 \\ c_6 \\ c_5 \\ c_4 \\ c_3 \\ c_2 \\ c_1 \\ c_0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} b_7 \\ b_6 \\ b_5 \\ b_4 \\ b_3 \\ b_2 \\ b_1 \\ b_0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}, \tag{2}$$

where the 8×8 matrix corresponds to the linear function $g(x) = ax + b$. Note that this is a circulant matrix $C = (1, 1, 1, 1, 1, 0, 0, 0)$, which is generated by a single row, and its inverse $C^{-1} = (0, 1, 0, 1, 0, 0, 1, 0)$ is also a circulant.

3.2 Round Function

The encryption process in the AES algorithm is a series of repetition of a simpler process called a round function. The round function consists of four basic steps.

(1) SubByte : This step performs component-wise substitution of the 4×4 array A of bytes.
(2) ShiftRow : This step permutes the rows of the 4×4 array A of bytes.
(3) MixColumn : This step scales and permutes the column of the 4×4 array A of bytes.
(4) AddRoundKey : This step adds a round key vector to the 4×4 array A of bytes.



Finer analytical details on each step are given below.

SubByte Step

The *SubByte* step performs a vector substitution. The vector s-box is computed component-wise on each coordinates using a substitution look-up table. Other methods of implementing

this function are also used in the actual implementation. In vector notation, the *SubByte* operation is given by

$$(x_{15}, x_{14}, x_{13}, x_{12}, x_{11}, x_{10}, x_9, x_8, x_7, x_6, x_5, x_4, x_3, x_2, x_1, x_0) \tag{4}$$

$$\xrightarrow{\text{SubByte}} (\sigma(x_{15}), \sigma(x_{14}), \sigma(x_{13}), \ldots, \sigma(x_4), \sigma(x_3), \sigma(x_2), \sigma(x_1), \sigma(x_0)),$$

where $\sigma(x_i) = ax_i^{-1} + b$, $x_i \in \mathbf{F}_{2^8}$.

ShiftRow Step

The *ShiftRow* step performs a series of left shift of the rows in a state array. The right cyclic shift increment R_i depends on the parameter b, and the ith position of the row in the array. The precise amount of shift is tabulated below.

b	R_0	R_1	R_2	R_3
4	0	1	2	3
5	0	1	2	3
6	0	1	2	3
7	0	1	2	4
8	0	1	3	4

The state diagram for $b = 4$ is given here: the first row remains fixed, the second row is right cyclic shifted by one place, the third row is right cyclic shifted by two places, and the fourth row is right cyclic shifted by three places.

$$\begin{bmatrix} x_0 & x_4 & x_8 & x_{12} \\ x_1 & x_5 & x_9 & x_{13} \\ x_2 & x_6 & x_{10} & x_{14} \\ x_3 & x_7 & x_{11} & x_{15} \end{bmatrix} \xrightarrow{\text{ShiftRow}} \begin{bmatrix} x_0 & x_4 & x_8 & x_{12} \\ x_{13} & x_1 & x_5 & x_9 \\ x_{10} & x_{14} & x_2 & x_6 \\ x_7 & x_{11} & x_{15} & x_3 \end{bmatrix}. \tag{5}$$

The *ShiftRow* step is realized by a 16×16 permutation matrix, it can be obtained by inspection of the state diagram. In vector notation, the *ShiftRow* operation is given by

$$(x_{15}, x_{14}, x_{13}, x_{12}, x_{11}, x_{10}, x_9, x_8, x_7, x_6, x_5, x_4, x_3, x_2, x_1, x_0) \tag{6}$$

$$\xrightarrow{\text{ShiftRow}} (x_3, x_6, x_9, x_{12}, x_{15}, x_2, x_5, x_8, x_{11}, x_{14}, x_1, x_4, x_7, x_{10}, x_{13}, x_0).$$

MixColumn Step

The *MixColumn* step scales and permutes the columns of a state array. This step is realized by a 16×16 diagonal matrix $D = \text{diag}(D_0, D_1, D_2, D_3)$, here each diagonal entry is a 4 × 4 submatrix, and $D_0 = D_1 = D_2 = D_3$. The submatrix and its inverse are given by

$$D_i = \begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \text{ and } D_i^{-1} = \begin{bmatrix} \text{E} & \text{B} & \text{D} & 9 \\ 9 & \text{E} & \text{B} & \text{D} \\ \text{D} & 9 & \text{E} & \text{B} \\ \text{B} & \text{D} & 9 & \text{E} \end{bmatrix}. \tag{7}$$

These pairs of circulant submatrices have entries from $\mathbf{F}_{2^8}$, but are commonly given in hexadecimal notation. In vector notation, the *MixColumn* operation is given by

$$(x_{15}, x_{14}, x_{13}, x_{12}, \ldots, x_3, x_2, x_1, x_0) \xrightarrow{\text{MixColumn}} (y_{15}, y_{14}, y_{13}, y_{12}, \ldots, y_3, y_2, y_1, y_0), \tag{8}$$

where $(c_3, c_2, c_1, c_0) = (2, 3, 1, 1)$, and

$$\begin{aligned}
y_{15} &= c_3x_{15} + c_2x_{14} + c_1x_{13} + c_0x_{12}, \\
y_{14} &= c_0x_{15} + c_3x_{14} + c_2x_{13} + c_1x_{12}, \\
y_{13} &= c_1x_{15} + c_0x_{14} + c_3x_{13} + c_2x_{12}, \\
y_{12} &= c_2x_{15} + c_1x_{14} + c_0x_{13} + c_3x_{12}, \\
&\vdots \qquad\qquad \vdots \qquad\qquad \vdots \\
y_3 &= c_3x_3 + c_2x_2 + c_1x_1 + c_0x_0, \\
y_2 &= c_0x_3 + c_3x_2 + c_2x_1 + c_1x_0, \\
y_1 &= c_1x_3 + c_0x_2 + c_3x_1 + c_2x_0, \\
y_0 &= c_2x_3 + c_1x_2 + c_0x_1 + c_3x_0.
\end{aligned}$$

For example, the first byte , x_0 is replaced by $y_0 = 3x_3 + x_2 + x_1 + 2x_0$.

The Key Schedule

The key schedule generates a sequence of key vectors from the initial key vector. The ith key vectors are given by $\boldsymbol{k}_i = (k_{16i,15}, k_{16i,14}, k_{16i,13}, k_{16i,12}, \ldots, k_{16i,3}, k_{16i,2}, k_{16i,1}, k_{16i,0})$, $i = 0, 1, 2, 3, \ldots$, where

$$k_{i,j} = \begin{cases} k_{i,j-1} + k_{i,j-k} & \text{if } j \neq 0 \bmod 4, \\ \sigma(k_{i,j+1}) + k_{i,j-k} + \widetilde{k}_{i,j} & \text{if } j \equiv 0 \bmod 4, \end{cases} \tag{9}$$

$\widetilde{k}_{i,j} \equiv y^{(j-k)/4} \bmod (y^4 + 1)$ is the round key constant, and the key vector $\boldsymbol{k}_0$ is the initial key.

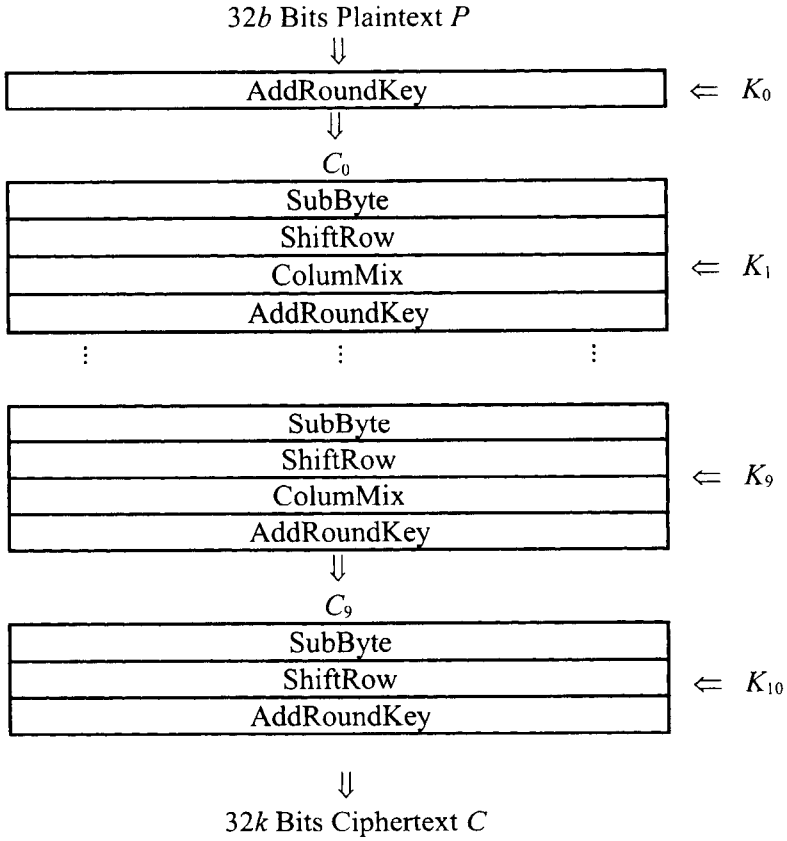
Currently block ciphers use a static key sequence, that is, only the first few vectors of the key sequence are actually used in the encryption process, AES uses less than $i < 15$. On the other hand, stream ciphers use dynamic key sequences, that is, all the vectors of the key sequence are actually used in the encryption process.

3.3 Encryption Algorithm

The round function is the crux of the AES algorithm, it is repeated a number of times depending on the security criteria. The number of round N_r = 10 to 14 in the AES-32b algorithm varies according to the plaintext length and key length. The AES-128 uses 10 to 14 full rounds and 2 shorter rounds, AES-192 uses 12 to 14 full rounds and 2 shorter rounds, and the AES-256 uses 14 full rounds and 2 shorter rounds. The small table below catalogs all the possibilities.

$N_R \backslash N_B$	4	5	6	7	8
4	10	11	12	13	14
5	11	11	12	13	14
6	12	12	12	13	14
7	13	13	13	13	14
8	14	14	14	14	14

The arrangement of the initial round, inner rounds and the final round is depicted in the diagram. Here P denotes a plaintext block, C_i denotes the ith round ciphertext block, and K_i denotes the ith round key vector.



4. Polynomial Description of the AES Algorithm

A polynomial description of the block cipher AES is essentially given in the original source [DR01]. A few other authors have also developed polynomial descriptions of the AES algorithm, see [LA02], [RL03]. The polynomial description given here is a synthesis of the descriptions given by these three authors, and uses the notations of the last two authors.
As before the plaintext data and key lengths are $32b$ bits and $32k$ bits respectively, and the parameters b, $k = 4, 5, 6, 7, 8$. The smallest plain text and key lengths are 128 bits for $b = k = 4$.

Bytes Space

The bytes space is identified with the finite field $\mathbf{F}_{2^8}$ of 2^8 elements

$$\mathbf{F}_{2^8} = \mathbf{F}_2[z]/(f(z)) = \{a(z) \in \mathbf{F}_2[z] : \deg(a(z)) < 8\}, \tag{12}$$

where $f(z) = z^8 + z^4 + z^3 + z + 1$. A byte $b = (b_7, b_6, b_5, b_4, b_3, b_2, b_1, b_0)$ of eight bits is viewed as polynomial $b(z) = b_7z^7 + \cdots + b_1z + b_0$ in $\mathbf{F}_2[z]/(f(z))$.

State Space

The states space is identified with the finite ring

$$\mathbf{F}_2[x,y,z]/(x^4+1, y^4+1, f(z)) = \{v(x,y) \in \mathbf{F}_{2^8}[x,y,z] : \deg(v(x,y)) < 4\}. \tag{13}$$

A state is viewed as polynomial in x, y over $\mathbf{F}_{2^8}$, (or x, y, z over $\mathbf{F}_2$), namely,

$$P = \sum_{i,j=0}^{3} a_{i,j} x^i y^j = \sum_{i,j=0}^{3} \sum_{k=0}^{7} a_{i,j,k} x^i y^j z^k , \tag{14}$$

where $a_{i,j} \in \mathbf{F}_{2^8}$, (or $a_{i,j,k} \in \mathbf{F}_2$).

The Key Schedule

The key stream is a sequence of polynomials $\{ k_i(x,y) = k_{i,3}x^3 + k_{i,2}x^2y + k_{i,1}xy^2 + \cdots + k_{i,0}y^3 \in \mathbf{F}_{2^8}[x,y] : k_{i,j}(z) \in \mathbf{F}_2[z]/(f(z)) \}$, where $k_{i,j}(z)$ is defined by

$$k_{i,j}(z) = \begin{cases} k_{i,j-1} + k_{i,j-k} & \text{if } j \neq 0 \bmod 4, \\ \sigma(k_{i,j+1}) + k_{i,j-k} + \tilde{k}_{i,j} & \text{if } j \equiv 0 \bmod 4, \end{cases} \tag{15}$$

where $\widetilde{k}_{i,j} \equiv x^{(j-k)/4} \bmod(f(z), x^4+1)$ is the round key constant, and the polynomial $k_0(z) = k_{0,3} z^3 + k_{0,2}z^2 + k_{0,1}z + k_{0,0}$ is the initial key. The ith round key is the polynomial $k_i(z) = k_{i,3}z^3 + k_{i,2}z^2 + k_{i,1}z + k_{i,0}$, $i = 0, 1, 2, 3, \ldots, k = 4, 5, 6, 7, 8$.

S-Box

The s-box is the same polynomial representation as in (2).

SubByte Step

The *SubByte* step performs the substitution $a_{i,j}(z) \to a(z) \cdot a_{i,j}^{-1}(z) + b(z) \bmod(z^8 + 1)$ of each coefficient $a_{i,j}(z) \in \mathbf{F}_{2^8}$ of the plaintext polynomial. Specifically this is given by

$$P = \sum_{i,j=0}^{3} a_{i,j} x^i y^j \quad \to \quad \sigma(P) = \sum_{i,j=0}^{3} \sigma(a_{i,j}) x^i y^j . \tag{16}$$

Each coefficient is a polynomial $a_{i,j}(z) = a_{i,j,7}z^7 + a_{i,j,6}z^6 + \cdots + a_{i,j,1}z + a_{i,j,0}$ in $\mathbf{F}_2[z]/(f(z))$, with $a_{i,j,k} \in \mathbf{F}_2$. The inverse $a_{i,j}^{-1}(z)$ is also a polynomial of degree $\deg(a_{i,j}^{-1}(z)) < 8$.

ShiftRow Step

The *ShiftRow* map shifts the degree of one of the variable of the plaintext polynomial P. This is given by

$$P = \sum_{i,j=0}^{3} a_{i,j} x^i y^j \quad \to \quad \rho(P) \equiv \sum_{i,j=0}^{3} a_{i,j} x^i y^{3i+j} \mod(y^4+1). \tag{17}$$

Here the polynomial $\rho(P)$ is reduced modulo $y^4 + 1$, or equivalently the exponent $3i + j$ is computed modulo 4.

MixColumn Step

The *MixColumn* map corresponds to multiplication by $\alpha = (z + 1)x^3 + x^2 + x + z$, that is,

$$P = \sum_{i,j=0}^{3} a_{i,j} x^i y^j \quad \to \quad \mu(P) \equiv \alpha \sum_{i,j=0}^{3} a_{i,j} x^i y^j \mod(x^4+1). \tag{18}$$

After multiplication by α, the resulting polynomial $\mu(P)$ is reduced modulo $x^4 + 1$.

AddRoundKey Step

The AddRoundKey consists of a polynomial addition:

$$P = \sum_{i,j=0}^{3} a_{i,j} x^i y^j \quad \to \quad \tau_k(P) = \sum_{i,j=0}^{3} a_{i,j} x^i y^j + \sum_{i,j=0}^{3} k_{i,j} x^i y^j = \sum_{i,j=0}^{3} (a_{i,j} + k_{i,j}) x^i y^j , \tag{19}$$

K is the key polynomial.

4.1 Encryption And Decryption Process

Let $K \in \mathsf{K} = \mathbf{F}_2^{32k}$, $P \in \mathcal{E} = \mathbf{F}_2^{32b}$, and $C \in \mathcal{F} = \mathbf{F}_2^{32b}$ be a key, plaintext, and ciphertext vectors respectively from the key, plaintext, and ciphertext spaces. The encryption function $\varepsilon_K : \mathbf{F}_2^{32b} \times \mathbf{F}_2^{32k} \to \mathbf{F}_2^{32b}$ is defined by a chain of compositions:

$$\varepsilon_k = \tau_{k_r} \rho\sigma\tau_{k_{r-1}} \mu\rho\sigma\tau_{k_{r-2}} \mu\rho\sigma\tau_{k_{r-3}} \mu\rho\sigma \cdots \tau_{k_3} \mu\rho\sigma\tau_{k_2} \mu\rho\sigma\tau_{k_1} \mu\rho\sigma\tau_{k_0}, \tag{20}$$

where the parameters $4 \le b, k \le 8$ specify the key, plaintext block lengths respectively, and $10 \le r \le 14$ specifies the number r of rounds.
The round function of the AES algorithm is defined an 4-tuple

$$\tau_{k_n} \mu\rho\sigma \ , \ 1 \le n \le 14. \tag{21}$$

A step by step evaluation of a round function is as follows: Given the nth round key and plaintext polynomials

$$K = \sum_{i,j=0}^{3} k_{i,j} x^i y^j, \quad P = \sum_{i,j=0}^{3} a_{i,j} x^i y^j, \tag{22}$$

the nth round ciphertext $C = \tau_{k_n} \mu\rho\sigma(P)$ is computed as follows:

$$\begin{aligned}
\tau_K \mu\rho\sigma(P) &= \tau_K \mu\rho\sigma\Big(\sum_{i,j=0}^{3} a_{i,j} x^i y^j\Big) \\
&= \tau_K \mu\rho\Big(\sum_{i,j=0}^{3} \sigma(a_{i,j}) x^i y^j\Big) \quad \mathrm{mod}(z^8 + z^4 + z^3 + z + 1) \\
&= \tau_K \mu\Big(\sum_{i,j=0}^{3} \sigma(a_{i,j}) x^i y^{3i+j}\Big) \quad \mathrm{mod}(y^4 + 1) \\
&= \tau_K \Big(\alpha \sum_{i,j=0}^{3} \sigma(a_{i,j}) x^i y^{3i+j}\Big) \quad \mathrm{mod}(x^4 + 1, z^8 + z^4 + z^3 + z + 1) \\
&= \sum_{i,j=0}^{3} (\alpha\sigma(a_{i,j}) y^{3i} + k_{i,j}) x^i y^j \quad \mathrm{mod}(x^4 + 1, y^4 + 1, z^8 + z^4 + z^3 + z + 1).
\end{aligned} \tag{23}$$

Line 1 corresponds to the *SubByte* step, line 2 corresponds to the *ShiftRow* step, line 3 corresponds to the *MixColumn* step, and line 4 corresponds to the *AddRoundKey* step.

This clearly shows the commutative property of a round function $\tau_{k_i} \mu\rho\sigma$, see Lemma 1 for the matrix/vector equation.

Similarly the decryption function $\delta_K : \mathbf{F}_2^{32b} \times \mathbf{F}_2^{32k} \to \mathbf{F}_2^{32b}$ is defined by a chain of compositions:

$$\delta_k = \tau_{k_0}^{-1}\sigma^{-1}\rho^{-1}\mu^{-1}\tau_{k_1}^{-1}\sigma^{-1}\rho^{-1}\mu^{-1}\tau_{k_2}^{-1}\cdots\sigma^{-1}\rho^{-1}\mu^{-1}\tau_{k_{r-2}}^{-1}\sigma^{-1}\rho^{-1}\mu^{-1}\tau_{k_{r-1}}^{-1}\sigma^{-1}\rho^{-1}\tau_{k_r}^{-1}. \tag{24}$$

Naturally the composition $\delta_K \circ \varepsilon_K = \iota$ of ε_K and δ_K is the identity function on $\mathbf{F}_2^{32b}$. For example, $\delta_K \circ \varepsilon_K (P) = \delta_K(\varepsilon_K(P)) = \delta_K(C) = P$.

5. Isomorphic AES Algorithms

An *isomorphic* AES algorithm is an exact image of the AES algorithm but in a different form. Any isomorphic AES algorithm is mapped to the AES algorithm by a linear map ω. The relationship is depicted in the diagram.

$$\begin{array}{ccc} \mathbf{F}_2^{32b} \times \mathbf{F}_2^{32k} & \xrightarrow{\ E\ } & \mathbf{F}_2^{32b} \\ \downarrow \iota & & \downarrow \omega \\ \mathbf{F}_2^{32b} \times \mathbf{F}_2^{32k} & \xrightarrow{\ \omega(E)\ } & \mathbf{F}_2^{32b} \end{array} \tag{25}$$

Here E denotes encryption by the standard AES algorithm, and ω(E) denotes encryption by the isomorphic AES algorithm, (ι is the identity map).

Two of the mechanisms used to produce isomorphic algorithms are as follow:
(i) Finite Field Conjugation. In the case of $\mathbf{F}_{2^8}$, there are eight possible conjugates,

$$\phi(x) = x^2,\ \phi^2(x) = x^4,\ \phi^3(x) = x^8,\ \ldots,\ \phi^7(x) = x^{128}. \tag{26}$$

(ii) Composite Representations of the Finite Field $\mathbf{F}_{2^8}$. There are exactly 450 composite representations of $\mathbf{F}_{2^8}$ using two subfields, see the section on *Irreducible Polynomials* in the Appendix for more details.

This technique (isomorphic AES algorithm) has several applications, among these are:

(a) *Fast implementation of the AES algorithm.* This is already proven to be very effective, see [RA01], and [WR02] etc.

(b) *Cryptoanalysis of the AES algorithm.* The probable advantage of using isomorphic AES algorithm is the possibility of obtaining simpler but equivalent round function, for instance, with diagonal submatrix in the *MixColumn* step. There is no result in this area yet.

Fact: The AES algorithm has over 3600 isomorphic AES algorithms.

Reason: Each isomorphic algorithm has eight conjugate algorithms, and there are at least 450 composite representations of the finite fields.

Example 1. The conjugate map $\phi(x) = x^2$ generates one of the simplest isomorphic AES algorithm ϕ(E). In this case the steps SubByte, ShiftRow, ColumMix, and AddRoundKey of the standard AES algorithm are conjugated (squared). As an illustration, the s-box (2) becomes

$$\begin{aligned}\phi\circ\sigma(x) &= [(z^6 + z^3 + z^2 + z)(x^{253} \bmod (z^8 + z^4 + z^3 + z + 1)) + z^7 + z^6 + z^2 + z + 1] \bmod (z^8 + 1).\\ &= [(z^6 + z^3 + z^2 + z)(x^{-2} \bmod (z^8 + z^4 + z^3 + z + 1)) + z^7 + z^6 + z^2 + z + 1] \bmod (z^8 + 1).\end{aligned}$$

6. Properties of the S-Box and the Round Function

The matrix/vector equation form of the round function consists of four operations shown in the diagram:

$$\boldsymbol{x} \rightarrow \boxed{PD\sigma(\boldsymbol{x}) + \boldsymbol{k}} \rightarrow \boldsymbol{y}, \quad \boldsymbol{k} \downarrow$$

where $\boldsymbol{k} = (k_{15}, \ldots, k_1, k_0)$, $\boldsymbol{x} = (x_{15}, \ldots, x_1, x_0)$, $\boldsymbol{y} = (y_{15}, \ldots, y_1, y_0)$, are key, plaintext, and ciphertext vectors respectively, and $D = (d_{i,j})$ and $P = (p_{i,j})$ are 16×16 matrices. All the entries $k_i, x_i, y_i, d_{i,j}, p_{i,j} \in \mathbf{F}_{2^8}$. Some observations on the function $f(\boldsymbol{x}) = \boldsymbol{y}$ are recorded here.

***Lemma* 1.** Let $\boldsymbol{x}$ and $\boldsymbol{k}$ be the nth state vector and key vector. Then the $(n+1)$th state vector

$$\boldsymbol{y} = PD\sigma(\boldsymbol{x}) + \boldsymbol{k} = DP\sigma(\boldsymbol{x}) + \boldsymbol{k} = \cdots = D\sigma(P\boldsymbol{x}) + \boldsymbol{k},$$

,

where $\sigma(\boldsymbol{x}) = (\sigma(x_{15}), \sigma(x_{14}), \ldots, \sigma(x_1), \sigma(x_0))$, and $\sigma(x) = ax^{-1} + b$, $n \geq 1$.

This shows that most of the steps of the round function commutes, and almost any two orderings of the steps in a round function are equivalent.

6.1 Invariant Sets of the AES Algorithm

A subset S is called an *invariant* subset of a map f if the image $f(S) = \{ f(s) : s \in S \}$ is contained in the subset, that it, $f(S) \subseteq S$. Define the subsets of state vectors

$S_0 = \{ \boldsymbol{x} = (x, x, x, x, x, x, x, x, x, x, x, x, x, x, x, x) : x \in \mathbf{F}_{2^8} \}$,

$S_1 = \{ \boldsymbol{x} = (x, y, x, y, x, y, x, y, x, y, x, y, x, y, x, y) : x, y \in \mathbf{F}_{2^8} \}$, and

$S_2 = \{ \boldsymbol{x} = (x, y, z, w, x, y, z, w, x, y, z, w, x, y, z, w) : x, y, z, w \in \mathbf{F}_{2^8} \}$.

***Lemma* 2.** Suppose that the key vector $\boldsymbol{k} \in S_i$, $i = 0, 1, 2$. Then the subsets S_0, S_1, and S_2 are invariant subsets of the nonlinear function $f(\boldsymbol{x}) = PD\sigma(\boldsymbol{x}) + \boldsymbol{k}$.

Proof: The matrix PD is a product of two 16×16 matrices, where P is a permutation composed of four 4×4 submatrices, and diagonal matrix $D = \text{diag}(D_0, D_1, D_2, D_3)$ is composed of four identical 4×4 submatrices. Let vector $\boldsymbol{x} \in S_0$. The action of the matrix P leaves the vector x fixed, and the action of each submatrix D_i on a subvector of $\boldsymbol{x}$ is given by

$$\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \begin{bmatrix} x \\ x \\ x \\ x \end{bmatrix} = \begin{bmatrix} x \\ x \\ x \\ x \end{bmatrix}, \tag{27}$$

which also fixes $\boldsymbol{x}$. Thus for any vector $\boldsymbol{x} \in S_0$, the image of the function is

$$f(\boldsymbol{x}) = PD\sigma(\boldsymbol{x}) + \boldsymbol{k} = (y, y, \ldots, y, y) \tag{28}$$

is again a vector in S_0. The verifications of the other invariants subsets are similar. ■

Recall that the entries on the submatrix are in hexadecimal notation, for instance, x01 = 1, x02 = z, x03 = $z + 1$, and $x = x_7z^7 + \cdots + x_1z + x_0$ in $\mathbf{F}_{2^8}$.

The invariants of the AES algorithm are used in the cyclic analysis of this encryption system, see [DT02]. This is related to the collision analysis of a function. Collision analysis seeks repetitions in the sequence of values $f(x_1), f(x_2), f(x_3), \ldots, f(x_N)$ of a function on a finite set of cardinality N. A repeated value can be used to obtain the argument x of $f(x)$ or some other related information. The expected number of distinct values $f(\boldsymbol{x})$ of a function on a set of cardinality N before it repeats itself is about $\sqrt{\pi N/2}$, see [MS96]. Usually the collision analysis, based on the birthday paradox, is intended for functions that are not one-to-one, such as Hash functions. Nevertheless it can be used to analyze any function.

Another possible application is chosen plaintext analysis to recover the key $\boldsymbol{k} \in S_0 \cup S_1 \cup S_2$.

6.1 Trace Description of the Round Function

The trace analysis provides a straight forward and important description of the round function of the AES algorithm. The reader should consult the sections on *Functions of n Variables*, and *Sequences* in the Appendix for background materials.

The trace description of the round function of the AES algorithm expresses the vector round function of the encryption function as a multivariable polynomial $f : \mathbf{F}_q^n \to \mathbf{F}_2^{8n}$ over the ground field $\mathbf{F} = \mathbf{F}_{2^8}$. This is defined by

$$(x_{n-1}, \ldots, x_1, x_0) \to (f_{8n-1}(x_{n-1} \ldots x_0), \ldots., f_0(x_{n-1} \ldots x_0)), \tag{29}$$

where $n = 4b$, and $x_i \in \mathbf{F}_{2^8}$. A more general ith coordinate function $f_i : \mathbf{F}_q^n \to \mathbf{F}_2$ is given by a function of n-variables of the form

$$f_i(x_{n-1} \cdots x_0) = Tr(\beta^{a_{n-1}} x_{n-1}^{-1}) + Tr(\beta^{a_{n-2}} x_{n-2}^{-1}) + \cdots + Tr(\beta^{a_0} x_0^{-1}) \tag{30}$$

where Tr is the absolute trace from $\mathbf{F}_q$ to $\mathbf{F}_2$, $\beta \in \mathbf{F}$ is a primitive element, and the a_i are fixed integers. In the case of the AES algorithm, each coordinate function is actually much simpler, just 4 of the 16 variables $x_{15}, \ldots, x_1, x_0$ appear in any one of them.

***Lemma* 3.** The coordinates of the vector round function of the AES algorithm functions are given by

$$\begin{aligned} f_0(x_{15}x_{10}x_5x_0) &= Tr(\beta^{166} x_{15}^{-1}) + Tr(\beta^{166} x_{10}^{-1}) + Tr(\beta^{154} x_5^{-1}) + Tr(\beta^{26} x_0^{-1}), \\ &\vdots \qquad\qquad \vdots \qquad\qquad \vdots \\ f_{127}(x_{15}x_{10}x_5x_0) &= Tr(\beta^{27} x_{12}^{-1}) + Tr(\beta^{51} x_{11}^{-1}) + Tr(\beta^{26} x_6^{-1}) + Tr(\beta^{26} x_1^{-1}), \end{aligned} \tag{31}$$

respectively. The complete set of 128 coordinate functions used to describe AES algorithm are derived in [YT03]. Note that the key vector $\boldsymbol{k} = (k_{127}, \ldots, k_0)$, $k_i \in \mathbf{F}_2$, have to be added to the coordinate functions to get the round function state vector $(f_{127} + k_{127}, \ldots, f_0 + k_0)$.

6.2 Maximal Sequence Description of the Round Function

The maximal sequence description of the AES algorithm is a continuation of the analysis used to derive the trace description. As previously stated the coordinate functions $f_i : \mathbf{F}_2^{32b} \to \mathbf{F}_2$ are from the plaintext space to the ground field, $0 \le i \le 127$. Consult the section on *Sequences* in the Appendix for background material.

***Lemma* 4.** (1) Each coordinate function f_i is a maximal sequence of period $q - 1 = 255$.
(2) The maximal sequences defined by the coordinate functions are in the same equivalent class.

Proof: Let $\beta \in \mathbf{F}_q$ be a primitive element, and for each $0 \neq x_i \in \mathbf{F}_q$, write variable $x_i = \beta^{s_i}$, $0 \le s_i < q - 1$. Then

$$Tr(\beta^{a_i} x_i^{-1}) = Tr(\beta^{a_i} \beta^{-s_i}) = Tr(\beta^{(q-2)s_i + a_i}). \quad (32)$$

Moreover, $(q-2)s_i + a_i$ is a linear function of s_i, so the argument of trace Tr runs over all the elements $0 \neq x_i \in \mathbf{F}_q$ as s_i varies over the integers. Hence $Tr(\beta^{a_i} x_i^{-1})$ is a maximal sequence over $\mathbf{F}_q$. Since the sum of maximal sequences with the same minimal polynomial is again a maximal sequence, it proves the claim. ■

The first statement implies that the output of a round function of the AES-128 algorithm, (when viewed as a compound pseudo random sequence), has a period of $(q-1)^{32b} = (2^8 - 1)^{128} < 2^{1024}$. This is better than a maximal sequence in $\mathbf{F}_{2^{128}}$ of period $2^{128} - 1$. However, the second statement claims that there is a linear relationship between the output of any two coordinate functions $f_i(x)$ and $f_j(x)$. Thus it probably does not have such a large period.

Observe that the structure of the sequence (3) is simpler to handle than an arbitrary coupled sequence of four variables, for instance, (33)

$$g(x,y,z,w) = Tr(\beta^{166} x^{-1} y^{-1} zw) + Tr(\beta^{166} x^{-1} yz^{67} w^{-1}) + Tr(\beta^{154} xy^{-1} zw^5) + Tr(\beta^{26} x^{54} y^{89} zw^{-1}).$$

6.3 The S-Box And Inversive Pseudorandom Number Method

Merging information from several area of analysis, for instance, matrix algebra, combinatoric of functions, and inversive pseudorandom number methods could be very productive in the cryptoanalysis of the round function of the AES algorithm.

For each fixed key, an iterated block cipher has many similarities to an iterated pseudorandom number generator. For example, the iterated AES-128 can be viewed as a pseudorandom number generator over the finite field $\mathbf{F}_{2^8}$. It is expected that the linear complexity is very large in order to avoid prediction analysis such as the Massey-Berlekamp algorithm.

The inversive pseudorandom number method can be utilized as a reference to compare the statistical and other properties of the round function of the AES algorithm against the statistical and other properties of inversive pseudorandom number generators. The round function of the AES algorithm is almost identical to an inversive pseudorandom number generator. The marked difference is in the calculation steps: An iterated round function is computed as (34)

$$s_{n+1} = [(z^4 + z^3 + z^2 + z + 1)(s_n^{254} \bmod (z^8 + z^4 + z^3 + z + 1)) + z^6 + z^5 + z + 1] \bmod (z^8 + 1),$$

in the both the finite field $\mathbf{F}_2[z]/(z^8 + z^4 + z^3 + z + 1)$ and the finite ring $\mathbf{F}_2[z]/(z^8 + 1)$. This accounts for the richer cycles structure and the fairly large period 277182 of the sequence s_0, $s_1, s_2, \ldots$. The orbits and cycles are the following;

$\mathrm{Orb}(s_0 = z^5 + 1) = \{ s_0, s_1, \ldots, s_{57}, s_{58} \}$, $\mathrm{Orb}(s_0 = (z^5 + 1)^2) = \{ s_0, s_1, \ldots, s_{79}, s_{80} \}$,

$\mathrm{Orb}(s_0 = (z^5 + 1)^4) = \{ s_0, s_1, \ldots, s_{85}, s_{86} \}$, $\mathrm{Orb}(s_0 = (z^5 + 1)^{22}) = \{ s_0, s_1, \ldots, s_{25}, s_{26} \}$,

$\mathrm{Orb}(s_0 = (z^5 + 1)^{38}) = \{ s_0, s_1 \}$,

and

$\#\mathrm{Orb}(z^5 + 1) = 59$, $\#\mathrm{Orb}((z^5 + 1)^2) = 81$, $\#\mathrm{Orb}((z^5 + 1)^{22}) = 87$,

$\#\mathrm{Orb}((z^5 + 1)^{22}) = 27$ $\#\mathrm{Orb}((z^5 + 1)^{38}) = 2$,

respectively. The period is computed via the lowest common multiple of the individual cycles:

$$\mathrm{lcm}(59, 81, 87, 27, 2) = 277182.$$

Note that the maximal order of a permutation on 256 items is 451,129,701,092,070. These statistics of the iterated s-box, which can be computed using MAPLE or similar software, have been verified by several authors, see [LA02], [RL03].

In contrast, the inversive pseudorandom number generator is simply

$$s_{n+1} = [(z^4 + z^3 + z^2 + z + 1)\, s_n^{254} + z^6 + z^5 + z + 1] \bmod (z^8 + z^4 + z^3 + z + 1), \qquad (35)$$

where all calculations take place in the finite field. The generator has the maximal period of $q = 256$, and there is a single orbit

$$\mathrm{Orb}(s_0) = \{ s_0, s_1, \ldots, s_{57}, s_{255} \} = \mathbf{F}_{2^8},$$

Of cardinality $\#\mathrm{Orb}(s_0) = 256$ for any $s_0 \in \mathbf{F}_{2^8}$.These statistics can be verified theoretically, see *Inversive Pseudorandom Method* in the Appendix, or using machine calculations.

A. Introduction to Finite Fields

The background material in finite field analysis provided here is substantially more than it is required to understand and implement the AES algorithm. However it is just a bit of the extensive and expanding knowledge employed in the design and cryptoanalysis of the AES algorithm and similar class of block ciphers. Many selected and useful results are included, but only the proofs of those with elementary proofs are given.

The ideas and concepts encountered here range from the very simple to the very difficult and some open problems.

***Definition* 1.** A finite ring is a finite set with two operations + and × defined on it. The ring is said to be a finite field if every element x has an additive inverse $-x$, and every nonzero element x has a multiplicative inverse x^{-1}.

***Definition* 2.** The characteristic $p > 0$ of a field (or ring) is the least integer p such that $1 + 1 + \cdots + 1 = 1 \cdot p = 0$. If there is no such integer, then $p = \infty$.

The smallest finite field $\mathbf{F}_p = \{ 1,\cdot 2,\cdot 3, \ldots, p - 1 \}$ of characteristic p is called the prime field.

***Theorem* 3.** Every finite field has $q = p^n$ elements, some prime p and integer $n \geq 1$.

A finite field $\mathbf{F}_q = \{ (a_{n-1}, a_{n-2}, \ldots, a_1, a_0)\colon a_i \in \mathbf{F}_p \}$ of $q = p^n$ elements is an extension of the prime field $\mathbf{F}_p$ of degree $n > 1$. As a set it consists of all the p-adic sequences of length n.

***Theorem* 4.** (*Binomial Theorem*) Let $q = p^n$ be a prime power. Then $(x + y)^q = x^q + y^q$ in $\mathbf{F}_q$.

***Theorem* 5.** (*Lucas Theorem*) Let n, k be a pair of integers and $\mathbf{F}_q$ a finite field of p^m elements. Then the binomial coefficient satisfies the congruence

$$\binom{n}{k} \equiv \prod_{i=0}^{r} \binom{n_i}{k_i} \bmod p\,, \tag{1}$$

where $n = n_r p^r + \cdots + n_1 p + n_0$, and $k = k_r p^r + \cdots + k_1 p + k_0$, $k_i, n_i \in \mathbf{F}_p$.

These results give a characterization of the binomial coefficients in the expansion of $(x + y)^n$ modulo p. However these results do not hold in finite ring, for instance, $(x + y)^6 \not\equiv x^6 + y^6 \bmod 6$.

***Theorem* 6.** The product of all the nonzero elements in a finite field equals -1. This generalizes the well known identity $1 \cdot 2 \cdot 3 \cdots (p - 1) \equiv -1 \bmod p$, called Wilson's theorem.

***Theorem* 7.** (*Fermat Little Theorem*) For every nonzero element $\alpha \in \mathbf{F}_q$, the relation $\alpha^{q-1} = 1$ holds.

Automorphism And Conjugation

An automorphism $\sigma : \mathbf{F}_q \rightarrow \mathbf{F}_q$ of a finite field is a map with the following properties:
(1) $\sigma(0) = 0$, $\sigma(1) = 1$, preserves the identities.
(2) $\sigma(\alpha+\beta) = \sigma(\alpha) + \sigma(\beta)$, additive,
(3) $\sigma(\alpha\beta) = \sigma(\alpha)\sigma(\beta)$, multiplicative.

The galois group of a finite field extension $\mathbf{F}_{q^n}$ of $\mathbf{F}_q$ is the set of all automorphisms:

$$\mathrm{Gal}(\mathbf{F}_{q^n}/\mathbf{F}_q) = \{ 1 = \sigma^0, \sigma^1, \sigma^2, \ldots, \sigma^{n-1} \}. \tag{2}$$

The generating map is defined by $\sigma(\alpha) = \alpha^q$. The ground field $\mathbf{F}_q$ is an invariant set (also called fixed field) of the galois group, in other words, $\sigma(\alpha) = \alpha$ whenever $\alpha \in \mathbf{F}_q$.

The set of *conjugates* of an element $\alpha \in \mathbf{F}_{q^n}$ is the orbit of the galois group:

$$\alpha, \sigma(\alpha) = \alpha^q, \sigma^2(\alpha) = \alpha^{q^2}, \ \ldots, \sigma^{n-1}(\alpha) = \alpha^{q^{n-1}}. \tag{3}$$

The Trace and Norm Functions

The trace and norm are maps defined by the sum

$$Tr(x) = x + x^q + x^{q^2} + \cdots + x^{q^{n-1}} \tag{4}$$

and the product

$$N(x) = x \cdot x^q \cdot x^{q^2} \cdots x^{q^{n-1}}. \tag{5}$$

More generally, These are maps from a field extension $\mathbf{F}_{q^n}$ to a subfield $\mathbf{F}_{q^d}$, $d \mid n$. The case $d = 1$ given above is called the absolute trace from $\mathbf{F}_{q^n}$ to $\mathbf{F}_q$. The composition $Tr_{n:d}(x) \circ Tr_{d:1}(x) = Tr_{n:d}(Tr_{d:1}(x))$ is computed by computing the sub traces

$$Tr_{d:1} : \mathbf{F}_{q^d} \rightarrow \mathbf{F}_q, \text{ where } Tr_{d:1}(x) = x + x^q + x^{q^2} + \cdots + x^{q^{d-1}} \tag{6}$$

and

$$Tr_{n:d} : \mathbf{F}_{q^n} \rightarrow \mathbf{F}_{q^d}, \text{ where } Tr_{n:d}(x) = x + x^{q^d} + x^{q^{2d}} + \cdots + x^{q^{d(e-1)}}.$$

Similar rule applies to the composition of the norm function.

***Theorem* 8.** Properties of the Trace Function

(1) $Tr(ax) = aTr(x)$, for all $a \in \mathbf{F}_q$, and $x \in \mathbf{F}_{q^n}$.

(2) $Tr(ax + by) = aTr(x) + bTr(y)$, for all $a, b \in \mathbf{F}_q$.

(3) $Tr_{n:1}(x) = Tr_{n:d}(x) \circ Tr_{d:1}(x)$, for any divisor $d \mid n$.

(4) The trace equation $Tr(x) = a$ has q^{n-1} solution, and it has the factorization

$$Tr(x) = \prod_{\alpha \in F_q} (x - \alpha)^{q^{n-1}}.$$

***Theorem* 9.** Properties of the Norm Function

(1) $N(ax) = a^n N(x)$, for all $0 \neq a \in \mathbf{F}_q$, and $x \in \mathbf{F}_{q^n}$.

(2) $N(xy) = N(x)N(y)$, for all $x, y \in \mathbf{F}_{q^n}$.

(3) $N_{n:1}(x) = N_{n:d}(x) \circ N_{d:1}(x)$, for any divisor $d \mid n$.

(4) The equation $N(x) = 1$ has $q^{n-1} + \cdots + q + 1$ solutions in $\mathbf{F}_{q^n}$.

***Theorem* 10.** (1) The kernel $\{ 0 \neq x \in \mathbf{F}_{q^n} : N(x) = 1 \}$ of $N : \mathbf{F}_{q^n} \rightarrow \mathbf{F}_q$ is a cyclic group of $\mathbf{F}_{q^n}$ consisting of all $x/\sigma(x)$, $0 \neq x \in \mathbf{F}_{q^n}$, and its cardinality equals $(q^n - 1)/(q - 1)$.

(2) The kernel $\{ x \in \mathbf{F}_{q^n} : Tr(x) = 0 \}$ of $Tr : \mathbf{F}_{q^n} \rightarrow \mathbf{F}_q$ is an additive group of $\mathbf{F}_{q^n}$ consisting of all $x - \sigma(x)$, $x \in \mathbf{F}_{q^n}$, and cardinality q^{n-1}.

Proof (1): Let $U = \{ x \in \mathbf{F}_{q^n} : N(x) = 1 \}$ and $Y = \{ y = x/\sigma(x) : x \in \mathbf{F}_{q^n} \}$. From $N(\sigma(x)) = N(x)$, it is clear that $Y \subset U$. Now let $z \in U$ and suppose that $z \neq x/\sigma(x)$ for all $0 \neq x \in \mathbf{F}_{q^n}$. Then $1 = N(z) \neq N(x/\sigma(x)) = 1$. This shows that $U \subset Y$. Moreover, if $0 \neq a \in \mathbf{F}_q$, then $ax/\sigma(ax) = x/\sigma(x)$, and the inverse $\sigma(x)/x$ of $x/\sigma(x)$ is also in U. Thus the quotient $\mathbf{F}_{q^n} / \mathbf{F}_q \cong U$ is a group isomorphism, so $\#U = (q^n - 1)/(q - 1)$. The proof for (2) is the additive version of this. ■

Polynomials and Coefficients

A polynomial of degree $n \geq 0$ and coefficients from the finite field $\mathbf{F}_q$ is an expression of the form

$$f(x) = a_n x^n + a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \cdots + a_1 x + a_0, \; a_i \in \mathbf{F}_q.$$

The set of all polynomials of one variable is denoted by $\mathbf{F}_q[x]$.

The *minimal polynomial* of an element is defined by

$$f_\alpha(x) = (x-\alpha)(x-\alpha^q)(x-\alpha^{q^2})\cdots(x-\alpha^{q^{n-1}}), \tag{7}$$

and the *degree* $\deg(\alpha) = \deg(f_\alpha)$ is defined as the number of distinct conjugates or equivalently distinct roots of f_α. A single root of a polynomial $f(x)$ determines all its coefficients. The minimal polynomial also has the form

$$f_\alpha(x) = x^n + a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \cdots + a_1x + a_0 \tag{8}$$

where $a_{n-1} = -\mathrm{Tr}(\alpha)$ and $a_0 = (-1)^n\mathrm{N}(\alpha)$ are the trace and norm of the element α. The other coefficients a_i are also given in terms of the traces of powers of α. These are obtained by means of the Newton identities in finite fields for $i < p/2$, prime $p \mid q$, and other methods.

Let $x_{n-1}, \ldots, x_1, x_0$ be the roots of a polynomial $f(x) = (x - x_{n-1})(x - x_{n-2}) \cdots (x - x_0)$. The elementary symmetric functions of n variables are defined by

$$\sigma_1 = \sum_{0\le i<n} x_i, \quad \sigma_2 = \sum_{0\le i<j<n} x_ix_j, \quad \sigma_3 = \sum_{0\le i<j<k<n} x_ix_jx_k, \quad \ldots, \quad \sigma_n = (-1)^n x_{n-1}x_{n-2}\cdots x_0 \tag{9}$$

and the power functions are defined by

$$s_1 = \sum_{0\le i<n} x_i, \quad s_2 = \sum_{0\le i<n} x_i^2, \quad s_3 = \sum_{0\le i<n} x_i^3, \quad \ldots, \quad s_n = \sum_{0\le i<n} x_i^n . \tag{10}$$

The elementary symmetric functions form an algebraic basis of the ring of symmetric polynomials. An important special case of this fact is expressed by Waring's formula

$$s_k = \sum_{k_1+2k_2+3k_3+\cdots+nk_n=k} (-1)^{k_2+k_4+\cdots}\left(\frac{k_1+k_2+\cdots+k_n-1}{k_1!k_2!\cdots k_n!}\right)\sigma_1^{k_1}\cdots\sigma_n^{k_n}. \tag{11}$$

These sums are used to obtain a variety of information on the coefficients of polynomials. This is due to the relationships of these sums to the coefficients, for example,

$$f(x) = x^n - \sigma_1x^{n-1} + \sigma_2x^{n-2} + \cdots + (-1)^{n-1}\sigma_{n-1}x + (-1)^n\sigma_0. \tag{12}$$

Some of the most important formulae are *Newton's identities*. These are obtained by replacing a root $x_0 = \alpha$ in the power functions, and using the cyclic nature of the roots.

Newton Identities In Finite Fields

For every element $\alpha \in \mathbf{F}_{q^m}$ of degree $n \mid m$, the followings recurring formulas hold:

(i) $\mathrm{Tr}(\alpha^k) + a_1\mathrm{Tr}(\alpha^{k-1}) + \cdots + a_{k-1}\mathrm{Tr}(\alpha) + k!a_k = 0$, $1 \le k < n$,

(ii) $\mathrm{Tr}(\alpha^k) + a_1\mathrm{Tr}(\alpha^{k-1}) + \cdots + a_n\mathrm{Tr}(\alpha^{k-n}) = 0$, $n \le k$.

These follow from standard Newton identities $s_k - \sigma_1 s_{k-1} + \sigma_2 s_{k-2} + \cdots + (-1)^{k-1}\sigma_{k-1}s_1 + (-1)^k k!\sigma_k = 0$, $1 \le k < n$, the identity $a_i = (-1)^i\sigma_i$ in any characteristic, and the identity $s_k = \mathrm{Tr}(\alpha^k)$ in finite fields of characteristic $p > 0$.

Representations of Finite Fields

There are various methods of representing finite fields, and all these representations of finite fields are isomorphic or equivalent. These methods are of both theoretical and practical interests. The most common representations are vector spaces and polynomial quotient rings, cyclic representations, quotients of number fields, composite representations, matrix representations, and *l*–adic representations respectively. These are listed and described here in order.

(1) Vector Spaces

A basis $\{\alpha_0, \alpha_1, \ldots, \alpha_{n-1}\}$ is a subset of linearly independent elements. A vector spaces representation is the linear span of the basis $\mathbf{F}_{q^n} \cong \{ x_{n-1}\alpha_{n-1} + \cdots + x_1\alpha_1 + x_0\alpha_0 : x_i \in \mathbf{F}_q \}$. The most common vector space representations of finite fields are the polynomials quotient rings $\mathbf{F}_{q^n} \cong \mathbf{F}_q[x]/(f(x)) = \{ a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \cdots + a_1x + a_0 : a_i \in \mathbf{F}_q \}$, where $f(x) \in \mathbf{F}_q[x]$ is an irreducible polynomial of degree n.

Addition and Multiplication

(1) $a(x) + b(x) = (a_{n-1} + b_{n-1})x^{n-1} + (a_{n-2} + b_{n-2})x^{n-2} + \cdots + (a_1 + b_1)x + a_0 + b_0$.

(2) $a(x)b(x) \equiv c_{n-1}x^{n-1} + c_{n-2}x^{n-2} + \cdots + c_1x + c_0$, mod $f(x)$, where $f(x)$ is the field defining polynomial.

The fastest addition and subtraction algorithms are implemented in vector space representations.

***Example* 10.** The finite field $\mathbf{F}_{2^4} \cong \{ a_3x^3 + a_2x^2 + a_1x + a_0 : a_i \in \mathbf{F}_2 \}$ of 16 elements is represented as the set of all cubic polynomials with coefficients in $\mathbf{F}_2 = \{ 0, 1 \}$. The subset of elements $\{ x^3, x^2, x, 1 \}$ forms a basis because it is linear independent over $\mathbf{F}_2$. The operations are done modulo $f(x)$, with respect to any of irreducible polynomials $f(x) = x^4 + x + 1$, $x^4 + x^3 + 1$, or $x^4 + x^3 + x^2 + x + 1$.

(2) Cyclic Groups

A cyclic group representation is of the form $\mathbf{F}_{q^n} \cong \{ \alpha = \xi^k : k \in \mathbb{Z} \} \cup \{0\}$. The nonzero elements of the finite field are expressed as powers of a fixed ξ generator of the multiplicative group of $\mathbf{F}_{q^n}$.

Addition and Multiplication
(1) $\alpha + \beta = \xi^k + \xi^l = \xi^l(\xi^{k-l} + 1)$.
(2) $\alpha\beta = \xi^{k+l}$.

The fastest multiplications, divisions, discrete exponentiations, and certain root extractions algorithms are implemented with cyclic representations.

***Example* 11.** The finite field $\mathbf{F}_{2^4} \cong \{ \xi^k : 0 \le k < 16 \} \cup \{0\}$ of 16 elements is represented as the set of all powers of a root ξ of either $f(x) = x^4 + x + 1$ or $x^4 + x^3 + 1$. These are both primitive polynomials over $\mathbf{F}_2$.

(3) Quotient of Number Fields

The representations of finite fields in number fields are of the forms $\mathbf{F}_{q^n} \cong \boldsymbol{O}_K/(\mathcal{J})$, where $\mathcal{J}$ is a maximal ideal and $\boldsymbol{O}_K$ is the ring of integers in a numbers field $\boldsymbol{K}$. These representations are used in theory and applications. The fastest algorithms for computing discrete logarithms in finite fields appear to be those implemented in polynomial quotient rings and quotient of number fields, see [AM93], [EL85] etc.

***Example* 12.** (i) Characteristic $p = 4m + 1$. The prime p splits as a product of two irreducibles $p = \pi \cdot \bar{\pi}$, where $\pi = a + b\sqrt{-1}$. The quotient the ring of the quadratic integers $\mathbb{Z}[\pi]$ and the ideal (π) gives an isomorphism $\mathbf{F}_p \cong \mathbb{Z}[\pi]/(\pi)$. Otherwise $p = 4m + 3$ is inert and $\mathbf{F}_{p^2} \cong \mathbb{Z}[\pi]/(\pi)$.

(ii) Characteristic $p = 3m + 1$. The prime p splits as a product of two irreducibles $p = \pi \cdot \bar{\pi}$, where $\pi = a + b\sqrt{-3}$. The quotient the ring of the quadratic integers $\mathbb{Z}[\pi]$ and the ideal (π) gives an isomorphism $\mathbf{F}_p \cong \mathbb{Z}[\pi]/(\pi)$. Otherwise $p = 3m + 2$ is inert and $\mathbf{F}_{p^2} \cong \mathbb{Z}[\pi]/(\pi)$.

(4) **Composite Representations**

A *composite* representation of finite field $\mathbf{F}_{q^n}$ over $\mathbf{F}_q$ rewrites the field as a finite extension of some of its subfield: $\mathbf{F}_{q^n} \cong F_{q^d}[x]/(f(x))$, where $f(x) \in \mathbf{F}_{q^d}[x]$ is irreducible of degree $\deg(f) = n/d$. There is growing interest in this construction. A possible generalization of this notion would rewrite a finite field as a composition of two or more similar or distinct representations of finite fields. For example, the ground field $\mathbf{F}_{q^d}$ could be represented as a cyclic representation, and the last stage $\mathbf{F}_{q^n} \cong F_{q^d}[x]/(f(x))$ as polynomials quotient ring.

***Example* 13.** The finite field $\mathbf{F}_{2^8}$ of 256 elements has three types of composite representations (using only two extensions):
(i) $\mathbf{F}_{2^8} \cong F_2[x]/(f(x)) \cong \{ a_7x^7 + a_6x^6 + a_5x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0 : a_i \in \mathbf{F}_2 \}$, where $f(x)$ is irreducible over $\mathbf{F}_2$. The composition diagram is $\mathbf{F}_2 \rightarrow \mathbf{F}_{2^8}$.

(ii) $\mathbf{F}_{2^8} \cong F_{2^2}[x]/(f(x)) \cong \{ a_3x^3 + a_2x^2 + a_1x + a_0 : a_i \in \mathbf{F}_{2^2} \}$. Here $\mathbf{F}_{2^2} = \{ 0, 1, \alpha, \alpha^2 \}$, where α is a root of $x^2 + x + 1 \in \mathbf{F}_2[x]$, and $f(x) = x^4 + a_3x^3 + a_2x^2 + a_1x + a_0$ is irreducible over $\mathbf{F}_{2^2}$. The composition diagram is $\mathbf{F}_2 \rightarrow \mathbf{F}_{2^2} \rightarrow \mathbf{F}_{2^8}$.

(iii) $\mathbf{F}_{2^8} \cong F_{2^4}[x]/(f(x)) \cong \{ a_1x + a_0 : a_i \in \mathbf{F}_{2^4} \}$. Here $\mathbf{F}_{2^4} = \{ 0, 1, \alpha, \alpha^2, \ldots, \alpha^{15} \}$, where α is a generator of the nonzero elements, and $f(x) = x^2 + a_1x + a_0$ is irreducible over $\mathbf{F}_{2^4}$. The composition diagram is $\mathbf{F}_2 \rightarrow \mathbf{F}_{2^4} \rightarrow \mathbf{F}_{2^8}$. For more details on this representation, see *Irreducible Polynomials* in the Appendix.

(5) Matrix Groups

$\mathbf{F}_{q^n} \cong \{$ Subset of $n \times n$ matrices in the group of nonsingular matrices $GL_n(\mathbf{F}_q) \}$.

(6) Others Types

There are a few other representations of but these are not very common, for instance, *l–adic* representation of finite field $\mathbf{F}_{q^n} \cong \{$ *l–adic* Vectors $\}$, the vectors are defined by a function ϕ : $\mathbf{F}_{q^n} \rightarrow \mathbf{F}_l^n$. For example, for odd prime powers q, two instances of this map are the 2*–adic* representation (binary):

$$\phi(x) = ([(x+\alpha_{n-1})^{q^n-1} + (x+\alpha_{n-1})^{(q^n-1)/2}]/2, \ldots, [(x+\alpha_0)^{q^n-1} + (x+\alpha_0)^{(q^n-1)/2}]/2)$$

and the 3*–adic* representation (ternary):

$$\phi(x) = ((x+\alpha_{n-1})^{(q^n-1)/2}, \ldots, (x+\alpha_1)^{(q^n-1)/2}, (x+\alpha_0)^{(q^n-1)/2}),$$

where $\alpha_0, \alpha_1, \ldots, \alpha_{n-1} \in \mathbf{F}_{q^n}$ are fixed, and $n < 2\log(q)$. An application appears in [GZ97].

Some of these techniques of representing finite fields will be encountered in the analysis of block ciphers. A current topic of interest in cryptography is the implementation of highly efficient algorithms to compute multiplications and multiplicative inverses in vector space and polynomials quotient ring representations of the finite fields $\mathbf{F}_{q^n}$ over $\mathbf{F}_q$. The technique of constructing composite representations of finite fields has proven to be very effective in the design of fast multiplication algorithms, a few the fast implementation of AES uses this technique. But it appears that composite representations have weakness in some cryptosystems, see the literature for details.

B. Irreducible, Primitive, and Normal Polynomials

The polynomials over finite fields are classified in many different ways, and there are many different classes of polynomials. A large and important class of polynomials is the set of irreducible polynomials and its subclasses.

A polynomial $f(x) \in \mathbf{F}_q[x]$ of degree $n > 1$ is said to be *irreducible* over $\mathbf{F}_q$ if it has no proper factors other than constants and itself.

***Lemma* 1.** A polynomial $f(x) \in \mathbf{F}_q[x]$ of degree $n > 1$ is irreducible over $\mathbf{F}_q$ if and only if

(i) $x^{q^n} - x \equiv 0 \bmod f(x)$, and

(ii) $\gcd(f(x), x^{q^{n/r}} - x) = 1$ for every prime divisor r of n.

Proof: Suppose that $f(x)$ is reducible and both (i) and (ii) hold. Then by (i) there is a factor $g(x)$ of $f(x)$ of degree $d \mid n$ that divides $x^{q^n} - x$, which implies that $g(x)$ divides $x^{q^{n/r}} - x$, $n/r = d$. But by (ii) no factor of $f(x)$ divides $x^{q^{n/r}} - x$, $1 < r < n$. This is a contradiction, so $f(x)$ is irreducible over $\mathbf{F}_q$. The converse is similar. ■

This result is based on the fact that the polynomial $x^{q^n} - x$ is the product of all irreducible polynomials of degrees $d \mid n$, and the roots of a polynomial $f(x) \in \mathbf{F}_q[x]$ of degree $n > 1$ are contained in an extension $\mathbf{F}_{q^n}$ of $\mathbf{F}_q$ of degree n. Another approach to the irreducibility test of a polynomial would be to use the powers q^d for every integer $d \le n/2$, this eliminates the need to factor the integer n, but it can be very inefficient.

***Example* 2.** To test the irreducibility of the polynomial $f(x) = x^8 + x^4 + x^3 + x + 1 \in \mathbf{F}_2[x]$ of degree $\deg(f(x)) = n = 8$, the gcd is checked for every integer $d \le n/2 = 4$:

(1) $x^{2^8} - x \equiv 0 \bmod f(x)$, $f(x)$ is a factor of $x^{2^8} - x$.

(2) $\gcd(f(x), x^2 - x) = 1$, $f(x)$ has no linear polynomial factor.

(3) $\gcd(f(x), x^{2^2} - x) = 1$, $f(x)$ has neither linear nor quadratic polynomial factors.

(4) $\gcd(f(x), x^{2^3} - x) = 1$, $f(x)$ has neither linear nor cubic polynomial factors.

(5) $\gcd(f(x), x^{2^4} - x) = 1$, $f(x)$ has neither linear, quadratic nor quartic polynomial factors.

Steps 2-4 are shown for illustrative purpose only. By Lemma 1, it is clear that steps 1 and 5 are sufficient to test the irreducibility of this polynomial over $\mathbf{F}_2$.

***Lemma* 3.** (Gauss) The total number of irreducible polynomials $f(x) \in \mathbf{F}_q[x]$ of degree $n > 1$ is given by

$$I_n(q) = n^{-1}\sum_{d|n}\mu(d)q^{n/d}, \tag{1}$$

where μ is the Mobius function.

***Example* 4.** Determine the number of irreducible polynomials of degree $n = 8$ in $\mathbf{F}_2[x]$.

Solution: Using the formula and simplifying, there are

$$8^{-1}\sum_{d|8}\mu(d)2^{8/d} = 8^{-1}(\mu(1)2^8 + \mu(2)2^4 + \mu(4)2^2 + \mu(8)2) = 8^{-1}(2^8 - 2^4) = 30$$

irreducible polynomials of degree 8. Different choice of irreducible polynomial amount to a choice of polynomial basis.

The coefficients of the irreducible polynomial $f(z) = x^8 + a_7x^7 + a_6x^6 + a_5x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x + 1$ are cataloged in the table 1.

$a_7a_6a_5a_4a_3a_2a_1$	$a_7a_6a_5a_4a_3a_2a_1$	$a_7a_6a_5a_4a_3a_2a_1$	$a_7a_6a_5a_4a_3a_2a_1$	$a_7a_6a_5a_4a_3a_2a_1$
0001101	0100110	0111011	1010001	1101011
0001110	0101111	0111101	1010100	1101110
0010101	0110001	1000011	1011000	1110011
0010110	0110010	1000101	1011110	1111011
0011100	0110100	1000110	1100001	1111010
0011111	0111000	1001111	1100111	1111100

Table 1

The small table 2 provides a list of values of $I_n(q)$ for $q = 2$.

n	$I_n(2)$	n	$I_n(2)$	n	$I_n(2)$
1	1	11	186	21	99858
2	1	12	335	22	190557
3	2	13	630	23	364722
4	3	14	1161	24	698870
5	6	15	2182	25	1342176
6	9	16	4080	26	2580795
7	18	17	7710	27	4971008
8	30	18	14532	28	9586395
9	56	19	27594	29	18512790
10	99	20	52377	30	35790267

Table 2

The ideas of element of degree n and irreducible polynomial in $\mathbf{F}_q[x]$ of degree n are somewhat equivalent notions. The function $\gamma_n(q) = nI_n(q)$ of the variables n and q enumerates

the number of elements of degree n in a finite field $\mathbf{F}_{q^n}$ of elements q^n. This function is monotonically increasing and does not deviate very much from the center value q^n.

***Lemma* 5.** Let $1 < n \in \mathbb{N}$ and q be a prime power, then $q^n - q^{n/2} < \gamma_n(q) \le q^n - q$.

Proof: The upper estimate $\gamma_n(q) \le q^n - q$ is clear since every integer n has at least two divisors. And for the lower estimate one has

$$\begin{aligned} \gamma_n(q) &= \sum_{d \mid n} \mu(d) q^{n/d} \\ &= q^n - \sum_{1<d \mid n} \mu(d) q^{n/d} \\ &> q^n - \sum_{d=1}^{n/2} q^d > q^n - q^{n/2} \end{aligned} \quad . \tag{2}$$

This inequality gives a useful estimate of the integer $I_n(q) = \gamma_n(q)/n$. It is also common to state this in O-notation as $I_n(q) = O(q^n/n)$. The number $\gamma_n(q) = nI_n(q)$ can also be viewed as the number of elements in $\mathbf{F}_{q^n}$ not contained in any proper subfield. In other words,

$$\gamma_n(q) = \#\left(\mathbf{F}_{q^n} - \bigcup_{d \mid n, d<n} \mathbf{F}_{q^d} \right). \tag{3}$$

Probability of Irreducible Elements

The probability of a randomly selected element α in $\mathbf{F}_{q^n}$ of being an irreducible element is defined by

$$P(\deg(\alpha) = n) = \sum_{d \mid n} \mu(d) q^{n/d-n} = \prod_{p \mid n} \left(1 - 1/q^p\right), \tag{4}$$

where d runs through the divisors of n and p runs through the prime divisors of n respectively. The probability $P(\deg(\alpha) = n)$ of finding an elements α of degree n in $\mathbf{F}_{q^n}$ rapidly approaches 1 as n and or q increases.

***Lemma* 6.** Let q be a prime power and $n \ge 2$. Then

$$1 - 1/q^{n/2} < P(\deg(\alpha) = n) \le 1 - 1/q^n.$$

In particular, $P(\deg(\alpha) = n) = 1 - 1/q^n$ if n is prime.

Proof: Use the previous Lemma. ■

A refined version of Gauss's Lemma gives information about the trace of the polynomials.

***Lemma* 7.** ([RM01]) Let q be a prime power. The number of irreducible polynomial of degree $n > 0$ over $\mathbf{F}_q$ with a given nonzero trace t is

$$I_n(q,t) = (nq)^{-1} \sum_{d|n,\gcd(d,p)=1} \mu(d) q^{n/d} . \tag{5}$$

Time Complexity Of Constructing Irreducible Polynomials

The basic ideas of generating irreducible polynomials date back to the time of Galois and Gauss: The algorithm selects a random polynomial of degree n and tests its irreducibility over $\mathbf{F}_q$ using Lemma 1. By Lemma 2, the probability of selecting an irreducible polynomial of degree n is about $1/n$. Today there are several modern variants of the basic probabilistic polynomial irreducibility algorithm derived from Lemmas 1, and 2, see [SP93].

On the other hand, there is no algorithm of polynomial time complexity for constructing irreducible polynomials. The existence of such algorithm is an open problem. However, under the assumption of the extended Riemann hypothesis there is a polynomial time algorithm, see [AL86]. A deterministic, but exponential in running time is developed in [SP90], and by other authors. This algorithm runs in $O(q^{1/2}\log(q)^4 n^4)$ operations in $\mathbf{F}_q$.

Statistics of Composite Representations

The following result gives a method for counting the composite representations of finite fields. Some composite representations are useful in the implementation of fast algorithms. The performance of an algorithm can vary significantly from one representation to another. For example, the AES algorithm can have slightly different properties with respect to any composite representation of the finite field $\mathbf{F}_{2^8}$. One of the things that change is the multiplicative order of the maps (such as the s-box) since the coefficients of the maps have different order with respect to different irreducible polynomials. Probably these representations will be more important in the next generation of larger block ciphers. But even in the small finite field $\mathbf{F}_{2^8}$ this technique has proved to be effective.

***Lemma* 8.** For every pair (n, q) there are (using one or two extensions)

$$c(n,q) = \sum_{d|n,d<n} I_d(q) I_{n/d}(q^d) \tag{6}$$

composite representations of the finite field $\mathbf{F}_{q^n}$, where $I_1(q) = 1$.

***Example* 9.** Determine the number of composite representations of the finite field $\mathbf{F}_{2^8}$.

Solution: Here $n = 8$, $q = 2$. The total is (using one or two extensions)

$$c(n, q) = I_1(2)I_8(2) + I_2(2)I_4(2^2) + I_4(2)I_2(2^4) = 450.$$

Instances of the three possible types using one or two extensions are:
(i) $\mathbf{F}_2 \to \mathbf{F}_2[x]/(x^8 + x^4 + x^3 + x + 1) \cong \mathbf{F}_{2^8}$,
(ii) $\mathbf{F}_2 \to \mathbf{F}_2[x]/(x^2 + x + 1) \cong \mathbf{F}_{2^2} \to \mathbf{F}_{2^2}/(y^4 + y^2 + xy + x + 1) \cong \mathbf{F}_{2^8}$,
(iii) $\mathbf{F}_2 \to \mathbf{F}_2[x]/(x^4 + x + 1) \cong \mathbf{F}_{2^4} \to \mathbf{F}_{2^4}/(y^2 + y + x^3) \cong \mathbf{F}_{2^8}$
Accordingly, computations in $\mathbf{F}_{2^8}$ can be done as
(i) $v(x)$ mod $(x^8 + x^4 + x^3 + x + 1)$, (ii) $v(x,y)$ mod $(x^2 + x + 1, y^4 + y^2 + x + 1)$, (iii) $v(x,y)$ mod $(x^4 + x + 1, y^2 + y + x^3)$, etc.

Primitive Polynomials

An irreducible polynomial $f(x)$ is referred to as a *primitive* polynomial if every root of $f(x)$ has the maximal order in the multiplicative group of $\mathbf{F}_{q^n}$.

***Definition* 10.** An element $\alpha \in \mathbf{F}_{q^n}$ has multiplicative order $N = \text{ord}(\alpha)$ if and only if $N = \min\{ d \in \mathbb{N} : \alpha^d = 1 \}$. In particular, an element of order $\text{ord}(\alpha) = q^n - 1$ is called *primitive*.

***Lemma* 11.** (*Standard primitive test*) An element $\alpha \in \mathbf{F}_{q^n}$ is primitive (has maximal order) if and only if

$$\alpha^{(q^n-1)/p} \neq 1 \text{ for all prime divisors } p \mid q^n - 1. \tag{7}$$

In general this is not a polynomial time algorithm to determine or recognize a primitive elements in an arbitrary finite field. The existence of a polynomial time algorithm to determine or recognize a primitive elements is an open problem. However there are classes of finite fields for which the primitive elements are recognized or generated in probabilistic polynomial time, for instance, if the integer $q^n - 1$ is easy to factor. In some cases every elements in the field is primitive, exempli gratia, $q^n - 1 = 2^n - 1 =$ prime. And in other cases almost every elements in the field is not primitive, exempli gratia, $q^n - 1 = 2 \cdot 3 \cdot 5 \cdots (r - 1)$, a product of the first consecutive primes.

***Lemma* 12.** The total number of primitive polynomial $f(x) \in \mathbf{F}_q[x]$ of degree n is given by

$$P_n(q) = n^{-1}\varphi(q^n - 1), \tag{8}$$

where $\varphi(n)$ is the totient function over the integers $\mathbb{Z}$.

***Example* 13.** Determine the number of primitive of degree $n = 8$ in $\mathbf{F}_2[x]$. There are

$$8^{-1}\varphi(2^8-1) = 8^{-1}\varphi(3\cdot 5\cdot 17) = 8^{-1}\varphi(3)\varphi(5)\varphi(17) = 8^{-1}(2\cdot 4\cdot 16) = 16$$

primitive polynomials in $\mathbf{F}_2[x]$ of degree 8. These are, a subset of table 1, listed here as a reference:

$x^8+x^4+x^3+x^2+1$
$x^8+x^5+x^3+x+1$
$x^8+x^5+x^3+x^2+1$
$x^8+x^6+x^3+x^2+1$
$x^8+x^6+x^4+x^3+x^2+x+1$
$x^8+x^6+x^5+x+1$
$x^8+x^6+x^5+x^2+1$
$x^8+x^6+x^5+x^3+1$
$x^8+x^6+x^5+x^4+1$
$x^8+x^7+x^2+x+1$
$x^8+x^7+x^3+x^2+1$
$x^8+x^7+x^5+x^4+1$
$x^8+x^7+x^6+x+1$
$x^8+x^7+x^6+x^3+x^2+x+1$
$x^8+x^7+x^6+x^5+x^2+x+1$
$x^8+x^7+x^6+x^5+x^4+x^2+1$

Table 3 has a sample of these values for $q = 2$.

n	$\varphi(2^n-1)/n$	N	$\varphi(2^n-1)/n$	N	$\varphi(2^n-1)/n$
1	1	11	176	21	84672
2	1	12	144	22	120032
3	2	13	630	23	356960
4	2	14	756	24	276480
5	6	15	1800	25	1296000
6	6	16	2048	26	1719900
7	18	17	7710	27	4202496
8	16	18	7776	28	4741632
9	48	19	27594	29	18407808
10	60	20	24000	30	17820000

Table 3

The probability of a randomly selected element α of being a primitive element in $\mathbf{F}_{q^n}$ is defined by

$$P(ord(\alpha) = q^n - 1) = \frac{\varphi(q^n-1)}{q^n-1} = \prod_{p \mid q^n-1}\left(1-\frac{1}{p}\right). \tag{9}$$

The main obstacle to the calculation of this probability is the present of the product term. In most cases only estimates can be computed.

Normal Polynomials

A *normal* polynomial $f(x)$ is an irreducible polynomial with linearly independent roots over the ground field $\mathbf{F}_q$. The set of roots of a normal polynomial serves as a normal basis of the vector space representation of $\mathbf{F}_{q^n}$ over $\mathbf{F}_q$. A normal basis is applied in [RL03] to analyze the S-box of the AES algorithm.

***Theorem* 14.** *(Standard Normal Test)* An element $\eta \in \mathbf{F}_{q^n}$ is a normal element over $\mathbf{F}_q$ if and only if the system of inequalities

$$\frac{x^n - 1}{a(x)} \circ \eta \neq 0 \tag{10}$$

holds for all irreducible factors $a(x)$ of $x^n - 1 \in \mathbf{F}_q[x]$.

The techniques employed in the proof of this result are extensively used in the literature, see [GN90], [LS87], etc. As an illustration, consider a root η of the irreducible polynomial $f(x) = x^8 + x^7 + x^6 + x + 1 \in \mathbf{F}_2[x]$ of trace $\mathrm{Tr}(\eta) = 1$. Since the factorization of $x^8 - 1 = (x - 1)^8$, there is only one irreducible factor $x - 1$, and the normal test reduces to a single inequality

$$\frac{x^8 - 1}{x - 1} \circ \eta = (x^7 + x^6 \cdots x + 1) \circ \eta = \eta^{2^7} + \eta^{2^6} + \cdots + \eta^2 + \eta = Tr(\eta) \neq 0 . \tag{11}$$

The last sum is equal to the trace so it is a normal polynomial, and the set of roots form a basis of the finite field $\mathbf{F}_{2^8}$. Accordingly, any element $\alpha \in \mathbf{F}_{2^8}$ has an expansion of the form

$$\alpha = a_7 \eta^{2^7} + a_6 \eta^{2^6} + \cdots + a_1 \eta^2 + a_0 \eta , \tag{12}$$

where $a_i \in \mathbf{F}_2$.

***Lemma* 15.** There total number of normal polynomial $f(x) \in \mathbf{F}_q[x]$ of degree n is given by

$$N_n(q) = n^{-1} \Phi(x^n - 1) , \tag{13}$$

where the function Φ is the totient in the ring of polynomials $\mathbf{F}_q[x]$.

Methods of evaluating the totient function Φ are widely available in the literature, see [LN97], etc. The evaluation of the special case $\Phi(x^n - 1)$ can be done using the factorization of $x^n - 1$ over $\mathbf{F}_q$ or related methods. In this case there is the formula

$$n^{-1}\Phi(x^n-1) = n^{-1}q^n \prod_{f|x^n-1}(1-q^{-\deg(f)}), \tag{14}$$

where $f(x)$ runs over the irreducible factors of $x^n - 1$ can be used.

***Example* 16.** Determine the number of normal polynomials of degree $n = 8$ in $\mathbf{F}_2[x]$.

Solution: The factorization of $x^8 - 1 = (x - 1)^8$ in $\mathbf{F}_2[x]$, so there are

$$8^{-1}\Phi(x^8-1) = 8^{-1}\cdot 2^8(1-2^{-\deg(x-1)}) = 8^{-1}\cdot 2^8(1-1/2) = 16$$

normal polynomials in $\mathbf{F}_2[x]$ of degree 8. By the previous test every irreducible polynomial of nonzero trace $a_7 \neq 0$ in table 1 is normal, some of these are listed here as a reference:

$x^8 + x^7 + x^2 + x + 1$ $\quad$ $x^8 + x^7 + x^6 + x + 1$
$x^8 + x^7 + x^3 + x + 1$ $\quad$ $x^8 + x^7 + x^5 + x^3 + 1$
$x^8 + x^7 + x^3 + x^2 + 1$ $\quad$ $x^8 + x^7 + x^5 + x^4 + 1$
$x^8 + x^7 + x^4 + x^3 + x^2 + x + 1$ $\quad$ $x^8 + x^7 + x^5 + x^4 + x^3 + x^2 + 1$

The classification of primitive normal polynomials of degree n in almost any polynomial ring $\mathbf{F}_q[x]$ is essentially numerical since there are no general formula to count them. For the parameters $q = 2$, and $n = 8$, the complete list of primitive normal polynomials is

$x^8 + x^7 + x^2 + x + 1$ $\quad$ $x^8 + x^7 + x^6 + x^3 + x^2 + x + 1$
$x^8 + x^7 + x^3 + x^2 + 1$ $\quad$ $x^8 + x^7 + x^6 + x^5 + x^2 + x + 1$
$x^8 + x^7 + x^5 + x^4 + 1$ $\quad$ $x^8 + x^7 + x^6 + x^5 + x^4 + x^2 + 1$
$x^8 + x^7 + x^6 + x + 1$

C. Functions in Finite Fields

The set of functions on the finite field is denoted by $\mathcal{F} = \{ f : \mathbf{F}_q \rightarrow \mathbf{F}_q \}$. The cardinality of this set is $\#\mathcal{F} = q^q$.

***Theorem* 1.** Every function on $\mathbf{F}_q$ has a polynomial representation of degree $< q - 1$.

Proof: Given a set of pairs (x_0, y_0), (x_1, y_1), …, (x_n, y_n), the evaluation of an n degree polynomial $f(x) = a_n x^n + \cdots + a_1 x + a_0$ at the $n + 1$ points becomes a system of equations

$$\begin{aligned} x_0^n + a_{n-1} x_0^{n-1} + \cdots + a_1 x + a_0 &= y_0, \\ \vdots \qquad\qquad \vdots \qquad\qquad & \vdots \\ x_n^n + a_{n-1} x_n^{n-1} + \cdots + a_1 x + a_0 &= y_n. \end{aligned} \tag{1}$$

The matrix form of this system of equation involves a van der Monde matrix $V_n = (x_i^j)$ of determinant

$$\det(V_n) = \prod_{i \neq j} (x_i - x_j). \tag{2}$$

But for $n + 1$ distinct points x_n, …, x_0, the determinant $\det(V_n) \neq 0$, so there is a unique solution $a_{n-1}, a_{n-2}, \ldots, a_0$. In particular, this valid for $q = n + 1$. ■

This result does not hold in finite ring of characteristic N. The reason is that the determinant might be $\det(V_n) = 0$ even if there are $n + 1$ distinct points. However, if each difference $x_i - x_j$ is coprime to N, then $\det(V_n) \neq 0$, and it has an unique solution.

Interpolation Formulas

The polynomial representation of a function $f : \mathbf{F}_q \rightarrow \mathbf{F}_q$ is obtained via interpolation method or other methods. The formula

$$f(x) = \sum_{\alpha \in \mathbf{F}_q} f(\alpha)\left(1 - (x - \alpha)^{q-1}\right) \tag{3}$$

gives a compact polynomial representation of any function $f(x)$ on $\mathbf{F}_q$. It also obtainable via the finite field cardinal series

$$f(x) = f(0)(1 - x^{q-1}) + \sum_{0 \neq \alpha \in \mathbf{F}_q} f(\alpha) \sum_{i=0}^{q-2} \alpha^i x^{q-1-i}. \tag{4}$$

***Definition* 2.** The *weight* of a polynomial $f(x) = a_n x^n + \cdots + a_1 x + a_0$ is defined by $w(f) = \#\{ a_n \neq 0 \}$. The polynomial is called dense if $w(f) \approx q - 1$.

The known polynomial representations of some functions believed to be one-way functions on finite fields are dense. For example, the discrete logarithm on finite fields, used in many cryptographic protocols, has the maximal weight $w(f(x)) = q - 1$ possible. In other cases the weight $w(f(x))$ of a function depends on the equivalence class of the prime power q, for instance, the square root has a constant weight $w(f(x)) = k$ for any prime power $q = 2^k n + 1$.

The polynomial representations of functions in finite fields is a topic of much interest in the design of cryptographic systems.

Linear Maps and Linear Functionals

The polynomial representations of functions in finite fields are classified in accordance to various criteria. The simplest classification is extracted from the degrees of the polynomials.

***Definition* 3.** The A linear map $f: \mathbf{F}_q \rightarrow \mathbf{F}_q$ on a finite field $\mathbf{F}_q$ is a map of the form $f(x) = ax + b$, $a, b \in \mathbf{F}_q$. Otherwise a map on $\mathbf{F}_q$ is nonlinear.

***Example* 4.** The map

$$f(x) = (z^4 + z^3 + z^2 + z + 1)x + z^6 + z^5 + z + 1 \bmod (z^8 + 1)$$

is linear on the finite ring $\mathbf{F}_2[x]/(x^8 + 1)$. And since $\gcd(x^4 + x^3 + x^2 + x + 1, x^8 + 1) = 1$, is has an inverse

$$f^{-1}(x) = (z^6 + z^3 + z)x + z^2 + 1 \bmod (z^8 + 1).$$

***Definition* 5.** A linear functional L is a map from $\mathbf{F}_{q^n}$ to the ground field $\mathbf{F}_q$; in symbol this is $L: \mathbf{F}_{q^n} \rightarrow \mathbf{F}_q$.

The canonical linear functional $\mathbf{F}_q$ is the trace polynomial $Tr(x) = x^{q^{n-1}} + \cdots + x^q + x$. This functional is of central importance in finite field analysis.

***Lemma* 6.** Every linear functional has a linear form, and a trace representation:
(i) $L(x) = a_{n-1}x_{n-1} + \cdots + a_1 x_1 + a_0 x_0$, where $(x_{n-1},\ldots,x_1,x_0)$ is the coordinates vector of x with respect to some basis, and $(a_{n-1},\ldots,a_1,a_0)$, $a_i \in \mathbf{F}_q$ is a fixed vector.
(ii) $L(x) = Tr(\mu x) = \sum_{i=0}^{n-1} a_i x^{q^i}$ for all $x \in \mathbf{F}_{q^n}$, and some fixed $\mu \in \mathbf{F}_{q^n}$.

Permutation Polynomials

The value set of a polynomial $f(x)$ of degree $\deg(f) = n$ is defined by $V(f) = \{ f(x) : x \in \mathbf{F}_q \}$. Trivially this integer satisfies the inequalities $q/(1 + n) \leq V(f) \leq q$. The lower estimate follows from the fact that $f(x) = a$ has at most n solutions. If required, better lower estimates are also available in the literature. Functions that have maximal value sets $V(f) = \mathbf{F}_q$ are one-to-one functions, and often called permutation polynomials.

An invertible function on a finite set is a permutation of the set. Only a few classes of invertible functions on finite sets such as finite fields are easy to describe. For instance, all the monomial $f(x) = ax^n \in \mathbf{F}_q[x]$ polynomial functions such that $\gcd(n, q - 1) = 1$ are invertible. An encryption procedure is necessarily an invertible function.
The set of *injective* (or equivalently one-to-one) functions is denoted by $\mathcal{J} = \{ f : \mathbf{F}_q \to \mathbf{F}_q \}$. The cardinality of this set is $\#\mathcal{J} = q!$.

***Lemma* 7.** (1) The set of injective functions $\mathcal{J} = \{ f : \mathbf{F}_q \to \mathbf{F}_q \}$ is closed under composition of functions modulo $x^q - x$. This set is isomorphic to the set of polynomials $\mathbf{F}_q[x]/(x^q - x)$.
(2) The cardinality of this set is $\#\mathcal{J} = q!$, and there are $q! - q^2$ nonlinear injective functions.

Properties of Injective Functions
The followings are equivalent:
(i) The function f on $\mathbf{F}_q$ is one-to-one.
(ii) For each $\alpha \in \mathbf{F}_q$, the equation $f(x) = \alpha$ has only one solution.

Families of One-to-One Functions
(1) Linear functions $f(x) = ax + b$, where $a, b \in \mathbf{F}_q$.
(2) Linearized polynomials $f(x) = a_n x^{p^n} + a_{n-1} x^{p^{n-1}} + \cdots + a_1 x$ with a single root in $\mathbf{F}_q$, $q = p^v$.
(3) Power functions $f(x) = ax^n$, with $\gcd(n, q - 1) = 1$.
(4) Dickson polynomials $D_a(x) = \sum_{k=0}^{n/2} \frac{n}{n-k} \binom{n-k}{k} (-a)^k x^{n-2k}$, if and only if $\gcd(n, q^2 - 1) = 1$, and $a \neq 0$. The case $a = 0$ reduces to the power function $f(x) = x^n$.
(5) Fractional linear transformations $f(x) = (ax + b)/(cx + d)$, with $ad - bc \neq 0$.

Probability of One-to-One Functions
Injective functions on finite fields are rare and with the exception of specific families of these functions, these are difficult to identify.

***Lemma* 8.** Almost every function in a finite field is not one-to-one.

Proof: Apply the Sterling's inequality $n^{n+1/2}e^{-n} \leq n! \leq 3n^{n+1/2}e^{-n}$ of the factorial to compute the ratio of the number of injective functions to the total number of functions:

$$\frac{\sqrt{q}}{e^q} \le \frac{q!}{q^q} \le \frac{3\sqrt{q}}{e^q}.$$

This ratio quickly vanishes as the size q of the finite field $\mathbf{F}_q$ increases. ■

An encryption procedure is a parametized one-to-one function on a finite set, the key is the parameter. The parameter selects a random instance from the set of parametized one-to-one functions.

***Example* 9.** The statistic for the finite field $\mathbf{F}_{2^n}$, $n \le 10$, are tabulated below. For $n = 8$, the chance that a randomly selected function on the small finite field $\mathbf{F}_{2^8}$ of 256 elements is a one-to-one function is practically zero: $256! / 2^{2048} < 2.7 \times 10^{-110}$.

$q = 2^n$	$q! / q^q$	$q = 2^n$	$q! / q^q$
2	5.00×10^{-1}	2^6	3.220×10^{-27}
2^2	9.375×10^{-2}	2^7	7.299×10^{-55}
2^3	2.403×10^{-3}	2^8	2.654×10^{-110}
2^4	1.134×10^{-6}	2^9	2.483×10^{-221}
2^5	1.800×10^{-13}	2^{10}	1.537×10^{-443}

Testing Algorithms

A dense machine representation of a rational function $f(x) = r(x)/s(x)$, where $r(x)$, $s(x)$ are polynomials of degrees $\le n$, uses about $2n\log_2(q)$ bits. A rational function problem has polynomial time complexity if it is computable in $O(n^a \log(q)^b)$ operations in $\mathbf{F}_q$, where $a, b > 0$ are constants. Otherwise is either subexponential or exponential.

The decision problem of identifying an arbitrary rational function as either one-to-one or not is a topic of current research. There are algorithms of exponential time complexities, and probabilistic polynomial time complexities, but there is no algorithm of polynomial time complexity. Except for a few cases, for instance, modulo 2^n.

***Theorem* 10.** (*Hermite Test*) A function f is one-to-one on if and only if the following hold:
(1) $f(x)$ has exactly one root in $\mathbf{F}_q$.
(2) $f(x)^t \bmod (x^q - x)$ is of degree $\le q - 2$, for all $0 < t < q - 1$.

This test is of theoretical interest. But its practical significance is limited since it has an exponential time complexity of $O(q^2)$ operations in $\mathbf{F}_q$. This is effective only for small prime power $q = O(n^c)$, where $\deg(f) = n$ is the degree of the polynomial and c is a constant.

The identity $x^q - x = \prod_{t \in \mathbf{F}_q} (x - f(t))$ supplies another means of testing the 1-to-1 property of a function. However it also has exponential time running time complexity, that is $O(q(\log n)^3)$ operations in $\mathbf{F}_q$, see [GN91].

***Theorem* 11.** Let $f(x) = r(x)/s(x) \in \mathbf{F}_q(x)$ be a rational function of degree n, and $\varepsilon > 0$ a real number. Then
(1) The function $f(x)$ is recognizes as a permutation or not in probabilistic polynomial time in $O(n^3(\log q)(\log \varepsilon^{-1}))$ operations in $\mathbf{F}_q$ with probability at least $1 - \varepsilon$.
(2) A deterministic algorithm recognizes a rational function as a permutation or not in $O(q^{1/2}n^3 (\log q))$ operations in $\mathbf{F}_q$.

The actual algorithms are described in [GN94].

Combinatorics and Functions

There is a rich structural relationship between combinatorics and functions on finite fields. A few elementary details are introduced here.

***Theorem* 12.** The group of all one-to-one functions on a finite field $\mathbf{F}_q$ is isomorphic to the symmetric group S_{q-1}, and it is generated by $f(x) = x^{q-2}$ and $g(x) = ax + b$, $a, b \in \mathbf{F}_q$.

***Theorem* 13.** (*Wells* 1969) (1) Every 2-cycle (transposition) over $\mathbf{F}_q$ is represented by a unique polynomial of degree $q - 2$.
(2) If $q \equiv 2 \bmod 3$, then every 3-cycle over $\mathbf{F}_q$ is represented by a unique polynomial of degree $q - 2$.
(3) If $q \equiv 1 \bmod 3$, then every 3-cycle over $\mathbf{F}_q$, but $2q(q - 1)/3$ 3-cycles is represented by a unique polynomial of degree $q - 2$.

The polynomial representations of a 2-cycle and a 3-cycle are

$$f_2(x) = x + (a - b)(x - a)^{q-1} + (b - a)(x - b)^{q-1},$$

$$f_3(x) = x + (a - b)(x - a)^{q-1} + (b - c)(x - b)^{q-1} + (c - a)(x - c)^{q-1},$$

respectively. The polynomial $f_2(x)$ simply exchanges two elements of the finite field $\mathbf{F}_q$ and leaves everything else fixed: $f_2(a) = b, f_2(b) = a$, and $f_2(c) = c$, for all $c \neq a, b$. Nevertheless it has very high degree. This indicates that even simple one-to-one maps such as 2-cycles and 3-cycles can have polynomial representations of very high degrees. Information about the degrees of injective functions has been of interest for quite sometimes. The following was just recently proved.

***Theorem* 14.** ([KN01]) Almost every permutation of $\mathbf{F}_q$ is represented by a polynomial of degree $q - 2$.

Cycles And Orders

A permutation $\pi \in S_n$ is a product of disjointed cycles of length n_i such that $n_1 + \cdots + n_t = n$. The conjugate permutation $\tau = \rho^{-1}\pi\rho$ has the same cycle structure decomposition as π.

The lengths of the cycles of a permutation $\pi \in S_n$ on a finite set S of cardinality $\#S = n$ are the cardinalities of the orbits $\text{orb}_\pi(x) = \{ \pi(x), \pi^2(x), \pi^3(x), \ldots \}$, $x \in S$. The set is disjoint union $S = \bigcup_x orb_\pi(x)$ induced by the map π. The order of the map in the symmetric group S_n is given by $\text{lcm}(n_1, n_2, n_3, \ldots)$, with $n_i = \#\ \text{orb}_\pi(x_i)$.

D. Functions of *n* Variables In Finite Fields

The *algebraic normal form* of a function $f\colon \mathbf{F}_q^n \to \mathbf{F}_q$ of n variables is the sum of products

$$f(x_{n-1}\cdots x_0) = \sum_e a_e x_{n-1}^{e_{n-1}} x_{n-2}^{e_{n-2}} \cdots x_0^{e_0}, \tag{1}$$

where the index $\boldsymbol{e} = (e_{n-1}, e_{n-2}, \ldots, e_1, e_0)$, $0 \le e_i < q - 1$. The total degree of f is the integer $d = \max\{ e_{n-1} + \cdots + e_0, : a_\alpha \neq 0 \}$.

***Example* 1.** (1) Every $\mathbf{F}_2$-valued function of three variables in algebraic normal form is of the shape

$$f(x_2x_1x_0) = a_0 + a_1x_0 + a_2x_1 + a_3x_2 + a_4x_0x_1 + a_5x_0x_2 + a_6x_1x_2 + a_7x_0x_1x_2.$$

(2) Every $\mathbf{F}_4$-valued function of three variables in algebraic normal form is of the shape

$$f(x_2x_1x_0) = a_0 + a_1x_0 + a_2x_1 + a_3x_2 + a_4x_0x_1 + a_5x_0x_2 + a_6x_1x_2 + a_7x_0^2 + \cdots + \\ + \cdots + a_8x_1^2 + a_9x_2^2 + a_{10}x_0^2x_1 + \cdots a_ex_0^2x_1^2x_2^2.$$

The maximal degree in any single variable is 1 in the first case and 2 in the second case because $x^2 = 1$ in $\mathbf{F}_2$ and $x^3 = 1$ in $\mathbf{F}_4$ respectively

***Lemma* 2.** The number of functions $f(x_{n-1}\cdots x_0) \in \mathbf{F}_q[x_{n-1},\ldots,x_0]$ whose ANF have at most $k \le q^n$ monomials is given by

$$1 + \binom{q^n}{1} + \binom{q^n}{2} + \cdots + \binom{q^n}{k}. \tag{2}$$

A pair of functions f and g are equivalent if there exists a linear change of variables such that $f(x) = g(\gamma x)$, where $\gamma x = (a_{0,0}x_{n-1} + \cdots + a_{0,n-1}x_0, \ldots, a_{n-1,0}x_{n-1} + \cdots + a_{n-1,n-1}x_0)$.

The algebraic thickness of a function is defined by $T(f) = \min\{ \mathrm{wt}(f(\gamma x)) : \gamma \in \mathrm{GL}_n(\mathbf{F}_q) \}$. This is the minimal number of monomials $a_ex_{n-1}^{e_{n-1}} x_{n-2}^{e_{n-2}} \cdots x_0^{e_0}$ as γ varies over the set of all $n \times n$ invertible matrices.

***Lemma* 3.** The set $\mathbf{F}_q[x_{n-1}\cdots x_0]/(x_{n-1}^q - x_{n-1},\ldots,x_0^q - x_0)$ of functions of n variables with values on the finite field $\mathbf{F}_q$ is an algebra.

***Definition* 4.** A polynomial $f\colon \mathbf{F}_q^n \to \mathbf{F}_q$ of n variables on a finite field is a 1-to-1 function if for each $y \in \mathbf{F}_q$, the equation $f(x_{n-1}\cdots x_0) = y$ has q^{n-1} solutions $(x_{n-1}\cdots x_0)$.

The functions that satisfy the condition stated above are called balanced, usually only binary valued, because the frequencies of occurrences of all elements of $\mathbf{F}_q$ are the same.

***Lemma* 5.** Almost every functions $f(x_{n-1}\cdots x_0) \in \mathbf{F}_q[x_{n-1},\ldots,x_0]$ is balanced.

Proof: Assume the binary case $f(x_{n-1}\cdots x_0) \in \mathbf{F}_2[x_{n-1},\ldots,x_0]$ for simplicity. Then the number of ways of arranging 2^{n-1} zeros on the true table of f of 2^n rows is exactly $\binom{2^n}{2^{n-1}}$. Taking the ratio to the total number 2^{2^n} of all functions yields

$$\binom{2^n}{2^{n-1}} 2^{-2^n} \to 1. \tag{3}$$

as n increases. ■

Let $\beta_{n-1}\cdots\beta_0$ and $\delta_{n-1}\cdots\delta_0$ be dual bases of $\mathbf{F}_{q^n}$ over $\mathbf{F}_q$. Dual bases are characterized by the relation $\mathrm{Tr}(\beta_i\delta_j) = \delta_{i,j}$, where $\delta_{i,j}$ is the delta function. The *trace normal form* of a function f: $\mathbf{F}_{q^n} \to \mathbf{F}_q$ is the sum of traces

$$f(x_{n-1}\cdots x_0) = \sum_e a_e Tr(\delta_i x_{n-1}^{e_{n-1}} x_{n-2}^{e_{n-2}} \cdots x_0^{e_0}), \tag{4}$$

where $x = x_{n-1}\beta_{n-1} + \cdots + x_0\beta_0 \in \mathbf{F}_{q^n}$, $0 \le e_i < q^n - 1$. This is sort of a dual of the algebraic normal form of a function, it expresses a function of a single variable on $\mathbf{F}_{q^n}$ as a function of n variables on $\mathbf{F}_q$.

***Theorem* 6.** Let $f(x) = (f_{n-1}(x_{n-1}\cdots x_0), \ldots, f_0(x_{n-1}\cdots x_0))$ be a function of n variables on $\mathbf{F}_q^n$, and let $\beta_{n-1}\cdots\beta_0$ be a basis of $\mathbf{F}_{q^n}$ over $\mathbf{F}_q$. Then

$$F(x_{n-1}\ldots x_0) = \sum_{i=0}^{n-1} \beta_i f_i(x_{n-1}\ldots x_0) \tag{5}$$

is a permutation of $\mathbf{F}_{q^n}$ if and only if $f(x)$ is a permutation of $\mathbf{F}_q^n$.

Some of the essential details on the duality of permutation of $\mathbf{F}_q^n$ and of permutation of $\mathbf{F}_{q^n}$ appear in [CM97], [NR71] and similar references.

***Lemma* 7.** Let $F(x) = x^d$ be a power map on $\mathbf{F}_{q^n}$, and let $f_i(x_{n-1} \ldots x_0) = Tr(\delta_i x^d)$, with $\gcd(d, q^n - 1) = 1$. Then the function $f(x) = (f_{n-1}(x_{n-1} \ldots x_0), \ldots, f_0(x_{n-1} \ldots x_0))$ of n variables is one-to-one.

***Lemma* 8.** Let $0 \neq \alpha \in \mathbf{F}_{q^n}$, and $d \in \mathbb{N}$, $\gcd(d, q^n - 1) = 1$. Then the degree of the function $f(x) = Tr(\alpha x^d)$ is equals to $\deg(Tr(\alpha x^d) \bmod g(x))$ = binary weight of d, where $g(x)$ is the defining polynomial of the finite field $\mathbf{F}_{2^n}$.

A discussion of this result appears in [CK04]. This is important in determining the algebraic complexity of a function.

Discrete Transforms

The discrete Fourier transform transforms a function on t-domain to an unique image function on the s-domain. There are various form of the discrete Fourier transform depending on the t-domain and s-domain and other criteria. One of the necessary conditions for the existence of discrete Fourier transform of length N is the existence of Nth primitive roots of unity in the s-domain. An Nth primitive root ω is characterized by ω^d, $0 < d < N$.

***Example* 9.** (1) Let $f : \mathbb{Z}_N \to \mathbb{C}$ be a complex valued function. Since the set of complex numbers $\mathbb{C}$ contains Nth primitive roots of unity $e^{i2\pi st/N}$ for any $N \geq 1$, the DFT of any length N exists. More precisely a DFT pair is defined by

$$\hat{f}(s) = \sum_{t=0}^{N-1} f(t) e^{i2\pi sst/N} \text{ and } f(t) = \frac{1}{N} \sum_{s=0}^{N-1} \hat{f}(s) e^{-i2\pi sst/N}. \tag{6}$$

(2) Let $f : \mathbb{Z}_N \to \mathbf{F}_q$ be a finite field-valued function. Since the finite field $\mathbf{F}_q$ contains Nth primitive roots ω of unity whenever N is a divisor of $q - 1$, the DFT of any length $N \mid q - 1$ exists. More precisely a DFT pair is defined by

$$\hat{f}(s) = \sum_{t=0}^{N-1} f(t) \omega^{st} \text{ and } f(t) = \frac{1}{N} \sum_{s=0}^{N-1} \hat{f}(s) \omega^{-st}. \tag{7}$$

Scaling Properties of Discrete Fourier Transform

(1) $e^{i2\pi tx/N} f(t) \to \hat{f}(s + x)$, modulation in t-domain.

(2) $f(t + x) \to e^{-i2\pi tx/N} \hat{f}(s)$, translation in t-domain.

(3) $f(xt) \to \hat{f}(x^{-1}s)$, decimation in t-domain.

Modulation in the t-domain is transformed to a translation in the s-domain, and conversely. Decimation by $x > 0$ in decimation in t-domain is transformed to decimation by x^{-1} mod N in decimation in s-domain.

Summation Properties of the Discrete Fourier Transform

(1) Parseval Identity:

$$\sum_{s\in\mathbf{F}_q} \hat{f}(s)^2 = q^2 . \tag{8}$$

(2) Poisson Sum/Convolution Sum:

$$q\sum_{t\in\mathbf{F}_q} f(t)g(t-s) = \sum_{s\in\mathbf{F}_q} \hat{f}(s)\hat{g}(s) . \tag{9}$$

All these identities and properties are easy to derive from (4) or (5) or similar definition of the discrete Fourier transform.

***Definition* 10.** Let $q = p^e$ be a prime power. The complexification of a finite field $\mathbf{F}_q$-valued function $f(t)$ of n variables is specified by the complex valued function

$$f_\chi(t) = e^{i2\pi(Tr(f(t))/p} . \tag{10}$$

This is an embedding $\chi : \mathbf{F}_q[x_{n-1},\ldots,x_0] \rightarrow \mathbb{C}$, which transfer the analysis of functions on finite fields to the analysis of functions on the unit complex disk $D = \{ z \in \mathbb{C} : |z| = 1 \}$.

***Definition* 11.** The discrete Fourier transform of an $\mathbf{F}_q$-valued function $f_\chi(t)$ is defined by the pair of complex valued function

$$\hat{f}_\chi(s) = \sum_{t\in\mathbf{F}_q^n} e^{i2\pi(Tr(f(t)+s\cdot t)/p} , \qquad f_\chi(t) = \sum_{s\in\mathbf{F}_q^n} e^{-i2\pi(Tr(\hat{f}_\chi(s)+s\cdot t)/p} , \tag{11}$$

where $s\cdot t = s_{n-1}t_{n-1} + \cdots + s_0t_0$ is the inner product of the two vectors.

In characteristic 2 with $q = 2^e$, the complexification of a $\mathbf{F}_q$-valued function $f(t)$ is simply the polarization $f(t) \rightarrow f_\chi(t) = (-1)^{Tr(f(t))}$, it changes it from an $\mathbf{F}_q$-valued function to an $\{-1, 1\}$ valued function. And the discrete Fourier transform is called the *Walsh* transform.

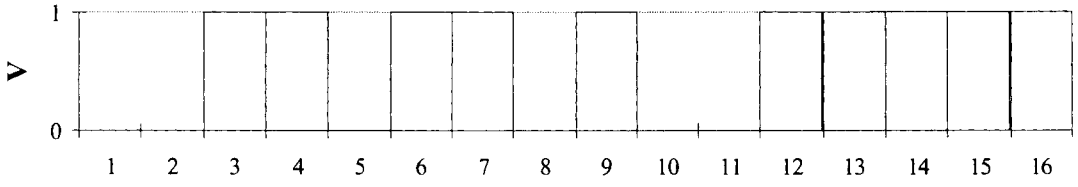
***Example* 12.** Compute the Walsh spectrum of the $\mathbf{F}_2$-valued function $f(t_3t_2t_1t_0) = t_3 + t_1 + t_0 + t_0t_2 + t_1t_2t_3$. This is a binary function of 4-variable over $\mathbf{F}_2$ and its discrete transform is

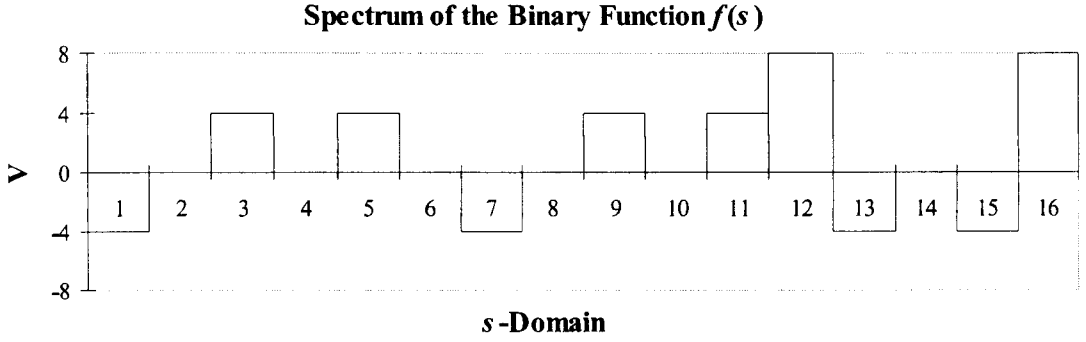
$$\hat{f}(s) = \sum_{t\in\mathbf{F}_2^4} (-1)^{f(t_3t_2t_1t_0)+s_3t_3+s_2t_2+s_1t_1+s_0t_0} . \tag{12}$$

These were computed and tabulated below.

	$t = t_3\ t_2\ t_1\ t_0$	$f(t_3t_2t_1t_0)$	$s = s_3\ s_2\ s_1\ s_0$	$\hat{f}_\chi(s_3s_2s_1s_0)$
0	0000	0	0000	–4
1	0001	0	0001	0
2	0010	1	0010	4
3	0011	1	0011	0
4	0100	0	0100	4
5	0101	1	0101	0
6	0110	1	0110	–4
7	0111	0	0111	0
8	1000	1	1000	4
9	1001	0	1001	0
10	1010	0	1010	4
11	1011	1	1011	8
12	1100	1	1100	–4
13	1101	1	1101	0
14	1110	1	1110	–4
15	1111	1	1111	8

Graph of the Binary Function $f(t)$





***Definition* 13.** A function $f \in \mathbf{F}_q[x_{n-1},\ldots,x_0]$ of n variables is called a *bent* function if the absolute value of the of the discrete Fourier transform $\hat{f}_\chi(s)$ of $f_\chi(t) = (-1)^{Tr(f(t))}$ satisfies

$$|\hat{f}_\chi(s)| = 1, \text{ for all } s \in \mathbf{F}_q^n. \tag{13}$$

This shows that the Walsh transform of a bent function has a flat spectrum. The standard references on bent functions are [DN74], [RS76], and its generalization in [KSW85].

***Lemma* 14.** The function $f(s)$ is a bent function if and only if its *autocorrelation* satisfies the relation

$$r_f(t) = \sum_{t \in \mathbf{F}_q^n} e^{i2\pi (Tr(f(s+t)-f(s)))/p} = \begin{cases} q^n & \text{if } t = 0, \\ 0 & \text{if } t \neq 0. \end{cases} \tag{14}$$

***Example* 15.** The map $f(x) = x^2 \in \mathbf{F}_q[x]$ is a bent function for prime power $q = p^e$, p odd. This follows from the properties of the quadratic Gaussian sum.

***Definition* 16.** A function $f \in \mathbf{F}_q[x_{n-1},\ldots,x_0]$ is called *perfect nonlinear* if its difference polynomial $g(x) = f(x+a) - f(x)$ is a 1-to-1 function for all $a \neq 0$.

***Theorem* 17.** A function $f \in \mathbf{F}_q[x_{n-1},\ldots,x_0]$ is perfect nonlinear if and only if it is a bent function.

Proof: It is about a page long, but it is not difficult, see [CM97]. ■

***Lemma* 18.** The sum of square errors of the spectrum of a function from any bent function is given by the sum of square autocorrelation

$$\sum_{0 \neq t \in \mathbf{F}_q^n}{}' r_f(t)^2 = q^{-n} \sum_{s \in \mathbf{F}_q^n}{}' (f(s)^2 - q^n)^2. \tag{15}$$

This is a generalization of the case $q = 2^e$, that appears in [KT03] to all prime powers.

Measures of Nonlinearity

Linear functions of n variables have measures of zero nonlinearity, and nonlinear functions of n variables over $\mathbf{F}_2$ have maximal measures of nonlinearity of $N_f \leq 2^{n-1} - 2^{n/2-1}$; the maximal measures of nonlinearity of nonlinear functions of n variables over $\mathbf{F}_q$ is not known, see [CT99] etc. for more details. A sketch of the methods used to derive these results is delineated here.

***Definition* 19.** The discrete measure μ on a finite set S is defined by the cardinality of the set $\mu(S) = \#S$.

***Definition* 20.** A *distance* function $d : S \to \mathbb{R}$, defined by $d(x,y) = d(x - y)$, is a metric on set if it satisfies the following properties
(i) $d(x,y) = 0 \Leftrightarrow x = y$,
(ii) $d(x,y) \geq 0$, positive definiteness.
(iii) $d(x,y) = d(y,x)$, symmetry,
(iv) $d(x,y) \leq d(x,z) + d(z,y)$, triangle inequality.

***Example* 21.** The discrete distance $\mu(x - y)$ (known as hamming weight) defines a topology on the vector space $\mathbf{F}_q^n = \{(x_{n-1},\ldots,x_0) : x_i \in \mathbf{F}_q\}$. The sphere and disk of radius r are given by

$$S_r(x) = \{x \in \mathbf{F}_q^n : \mu(y-x) = r\}, \quad D_r(x) = \{x \in \mathbf{F}_q^n : \mu(y-x) \leq r\}, \tag{16}$$

respectively. These are sets of (discrete area and volume) cardinalities

$$\#S_r(x) = \binom{n}{r}, \quad \#D_r(x) = \sum_{k=0}^{r} \binom{n}{k}, \tag{17}$$

respectively, where the bracket denotes the binomial coefficient.

The *support* of a function is defined by $\mu(f) = \#\{\, x \in \mathbf{F}_q : f(x) \neq 0 \,\}$.

***Lemma* 22.** Let $\delta(f, g) = \delta(f - g)$, with $f, g \in \mathcal{F}$. Then μ is a metric on the set of functions $\mathcal{F}$.

The binary distance between two functions is also (in term of the Walsh transform) written as

$$\delta(f,g) = 2^{n-1} - 2^{-1} \sum_{x \in \mathbf{F}_{2^n}} (-1)^{f(x)+g(x)}. \tag{18}$$

The nonlinearity measure of a function f quantifies the closeness of the function $f(x)$ to every linear function $g(x)$, in terms of discrete metric, this is defined by

$$N_f = 2^{n-1} - 2^{-1} \max_{u} \sum_{x \in F_{2^n}} (-1)^{f(x)+u \cdot x} . \tag{19}$$

The optimization is over all linear functions $g(x) = u \cdot x = a_{n-1}x_{n-1} + a_{n-2}x_{n-2} + \cdots + a_0x_0$.

***Theorem* 23.** Let c > 0 be an arbitrary constant. The density of the set of binary functions

$$\{ f : N_f \geq 2^{n-1} - c\sqrt{n}2^{(n-1)/2} \} \tag{20}$$

is $1 - c^2\log_2(e)$. In particular, if $n > 4$, then almost every binary function has nonlinearity $N_f \geq 2^{n-1} - n^{1/2}2^{(n-1)/2}$.

For more details, see [CT04].

Some Properties of Functions

Several characteristics of function of n variables over a finite field have simple and unified descriptions in the s-domain. Specifically the characteristics are given in terms of the Walsh spectrum of the functions. The same characteristics of functions can also be described in terms of probability in the t-domain, but are more difficult to understand, [NY93], [BT93], etc.

***Definition* 24.** A function $f \in \mathbf{F}_q[x_{n-1},\ldots,x_0]$ is called *balanced* if it has the spectrum value $\hat{f}_\chi(0) = 0$. This property indicates that the values $f(x)$ of a function as x ranges over all the n-tuples $(x_{n-1},\ldots,x_0)$ are essentially uniformly distributed and each value has the same frequencies q^{n-1}.

The discrete weight of a vector $\boldsymbol{x} = (x_{n-1},\ldots,x_0)$ is the integer $w(\boldsymbol{x}) = \#\ \{\ x_i \neq 0 : \boldsymbol{x} = (x_{n-1},\ldots, x_0)\}$. The discrete sphere and disk of radius r are the sets of vectors $S_r = \{\ \boldsymbol{x} = (x_{n-1},\ldots,x_0) : w(\boldsymbol{x}) = r\ \}$ and $D_r = \{\ \boldsymbol{x} = (x_{n-1},\ldots,x_0) : w(\boldsymbol{x}) \leq r\ \}$ respectively.

***Definition* 25.** A function $f \in \mathbf{F}_q[x_{n-1},\ldots,x_0]$ is called *correlation immune of order r* if the spectrum values $|\hat{f}_\chi(s)| = 0$ for every nonzero vector $\boldsymbol{s}$ in the discrete disk D_r of radius r. This property indicates that the function $f(x)$ is not properly approximated by any linear function $g(x) = a_{r-1}x_{r-1} + \cdots + a_0x_0$ of r variables.

***Definition* 26.** A function $f \in \mathbf{F}_q[x_{n-1},\ldots,x_0]$ is called *resilient of order r* if its spectrum $|\hat{f}_\chi(s)| = 0$ vanishes on the discrete disk D_r of radius r.

***Definition* 27.** A map $f : \mathbf{F}_q \rightarrow \mathbf{F}_q$ is called *almost perfect nonlinear* function if the difference equation $f(x+a) - f(x) = b$ has at most two solutions in $\mathbf{F}_q$.

The set of powers $S(d) = \{ e = dp^k \bmod (p^n - 1) : k \geq 0 \}$ classifies an equivalent class of almost perfect nonlinear power functions on $\mathbf{F}_{p^n}$.

Highly nonlinear encryption functions have high resistance to linear and differential cryptoanalysis.

Example of almost perfect nonlinear power function $f(x) = x^d$ on $\mathbf{F}_{2^n}$.

Case	Power	Constraint
I	$2^k + 1$	$\gcd(k, n) = 1$
II	$2^{2k} - 2^k + 1$	$\gcd(k, n) = 1$
III	$2^{4k} + 2^{3k} + 2^{2k} + 2^k - 1$	$n = 5k$
IV	$2^k + 3$	$n = 2k + 1$
V	$2^{2k} + 2^k - 1$	$2k + 1 \equiv 0 \bmod n$
VI	$2^n - 2$	$n = 2k + 1$

The proof of case III is given in [DN99].

E. Recurent Sequences and Pseudorandom Number Methods

A linear recurring sequence of order k is a sequence of elements generated by

$$s_n = a_{k-1}s_{n-1} + a_{k-2}s_{n-2} + \cdots + a_0 s_{n-k}, \qquad n \geq k, \tag{1}$$

where $a_{k-1}, a_{k-2}, \ldots, a_0$ are constants.

The first k terms $s_{k-1}, s_{k-2}, \ldots, s_0$, called the initial condition, specify an unique sequence. The characteristic polynomial of the sequence is a polynomial $s(x) = a_k x^k + \cdots + a_1 x + a_0$ of least degree that satisfies the previous recurrent relation.

Recurrent sequences are periodic of period ρ and satisfy the periodicity relation $s_{n+\rho} = s_n$ for all $n \geq n_0$, and $a_0 \neq 0$.

***Theorem* 1.** If the irreducible polynomial $s(x)$ is the characteristic polynomial of a sequence s_n and β is a root of $s(x)$, then there exists an element $\alpha \in \mathbf{F}_{q^n}$ such that $s_n = \mathrm{Tr}(\alpha\beta^n)$, $n \geq 0$.

***Theorem* 2.** The period ρ of a k order linear recurrent sequence s_n with characteristic polynomial $s(x)$ is equal to the order of $\mathrm{ord}(s(x))$, which is a divisor of $q^k - 1$.

***Lemma* 3.** ([LN97]) The set of maximal sequences with the same minimal polynomial f form a vector space of dimension $\deg(f)$.

***Definition* 4.** Linear complexity $L \geq 0$ of a sequence is equal to the degree $\deg(s(x))$ of characteristic polynomial $s(x)$ of the sequence.

Sequences of period N satisfy the relation $s_{n+N} - s_n = 0$, so the minimal polynomial of the sequences divide $x^N - 1$. This shows that the linear complexity $L \leq N$.

***Theorem* 5.** The linear complexity of a recurrent sequence s_n is equal to the Hamming weight of its inverse Fourier transform.

Proof: See [BT03, p.136], it also appeared in the IEEE Journal of Information Theory.

***Definition* 6.** Given a set of arbitrary points $x_{N-1}, x_{N-2}, \ldots, x_0 \in [0, 1]^s$, in the s-dimensional cube, the discrepancy

$$D_N(x_{N-1}\ldots x_0) = N^{-1} \sup_J | F_N(J) - V(J) |, \tag{2}$$

where $V(J)$ is the volume of the interval $J \subset [0, 1]^s$, and $F_N(J)$ is the number of points striking the interval J.

The discrepancy of N successive points of a sequence is used to measure the statistical independence of the N successive points vectors.

The law of iterated logarithm claims that the discrepancy of almost every true random set of points has an order of magnitude of

$$D_N(x_{N-1}...x_0) \approx N^{-1/2}(\log\log N)^{1/2}, \tag{3}$$

The recurrent sequence $y_n = af(n) + b$ over $\mathbf{F}_q$ is purely periodic of period q if the function f is one-to-one.

The discrepancy of a compound $af_i(n) + b_i$ (using s distinct recurrent sequences and over the full period of each sequence) is independent of the parameters a_i, b_i, f_i when the functions f_s, ..., f_0 are algebraic independent over $\mathbf{F}_q$. Its value satisfies the estimate

$$D_N(x_{N-1}...x_0) \le (d-1)q^{-1/2}(\log\log q)^s, \tag{4}$$

where the degrees $d_i = \deg(f_i)$, and $d = \max\{ d_i \}$.

F. Nonlinear Pseudorandom Number Methods

A nonlinear periodic sequence over a finite field $\mathbf{F}_q$ is defined by a nonlinear function $s_n = f(s_n)$.

The most common nonlinear periodic sequences are the followings:
(i) Inversive pseudo random number generators defined by the recursion

$$s_{n+1} = as_n^{-1} + b\,, \tag{1}$$

where s_0 is the initial condition (seed), $a, b \in \mathbf{F}_q$, and $n \geq 0$.
(ii) Quadratic pseudo random number generators defined by the recursion

$$s_{n+1} = as_n^2 + b\,, \tag{2}$$

In general $s_{n+1} = af(s_n) + b$ is a pseudo random number generators of maximal period q whenever f is a one-to-one function.

The theory of inversive pseudorandom number generators is a fairly advanced subject and a topic of current research. A short exposition of the theory of inversive pseudorandom number generator is included here.

***Lemma* 1.** ([NR94]) The inversive pseudorandom number generator $s_{n+1} = as_n^{-1} + b$ over the finite field $\mathbf{F}_q$ has period of length q if and only if the order of the product $\alpha\alpha^{-q}$ of the roots of the irreducible polynomial $x^2 - bx - a = (x - \alpha)(x - \alpha^q)$ is equal to $q + 1$.

The condition stated in the result can be verified using the fixed parameters a, b and q of the sequence. Assuming that $f(x) = x^2 - bx - a$ is irreducible over $\mathbf{F}_q$, it is sufficient to compute the power

$$(x - a)^{q-1} \bmod f(x) \not\equiv 1.$$

This follows from the relation

$$x \cdot x^{-q} = \frac{1}{x^{q-1}} = \left(\frac{x-a}{b}\right)^{q-1} = (x-a)^{q-1}\,. \tag{3}$$

***Definition* 2.** Let $\nu_n = (s_n, s_{n+1}, s_{n+2}, \ldots, s_{n+d-1})$ be a vector in $\mathbf{F}_q^d$, and let $V = \{ v = \sum_{i=1}^{d-2} a_i(\nu_i - \nu_0) : a_i \in \mathbf{F}_q \}$ be the linear span of the set of vectors $\{ \nu_i - \nu_0 \}$, $i \geq 1$. The

pseudorandom sequence $s_{n+1} = f(s_n)$, where f is a function on $\mathbf{F}_q$, passes the lattice test of fixed dimension $d \geq 2$ if the linear span V is identical to the vector space $\mathbf{F}_q^d$.

The lattice test is necessary but not sufficient to have a pseudorandom sequence. A method for constructing sequences that pass the lattice test but fail to be pseudorandom sequences is discussed in [ER88].

***Theorem* 2.** ([NR02]) A sequence $s_{n+1} = as_n^{-1} + b$ in finite field $\mathbf{F}_q$ of maximal period q has a linear complexity of at least $L \geq q/2$ and passes the lattice test for all dimension $d \leq (q-1)/2$.

A compound inversive pseudorandom sequence is a vector

$$\boldsymbol{v}_{n+1} = (a_t v_{t,n}^{-1} + b_t, a_{t-1} v_{t-1,n}^{-1} + b_{t-1}, ..., a_1 v_{1,n}^{-1} + b_1) . \tag{2}$$

of inversive pseudorandom sequences.

***Theorem* 4.** ([EH96]) A sequence of pseudorandom vectors $\boldsymbol{v}_0, \boldsymbol{v}_1, \boldsymbol{v}_2, \ldots$ generated by compound inversive method is always purely periodic of period length $q_1 \cdots q_t$ if and only if all the underlying sequences $v_{i,n+1} = a_i v_{i,n}^{-1} + b_i$ of pseudorandom numbers have period length q_i, $i = 1, \ldots, t$.

REFERENCES:

[AL86] LM Adleman, HW Lenstra, *Finding irreducible polynomials over finite fields*, ACM 18th Proceedings Symposium on Theory of Computing, 1986, pp-350-355.
[AM93] LM Adleman, J DeMarrais, *A subexponetial-time algorithm for computing discrete logarithm over finite fields,* Math. Comp. Vol. 61, 1993, pp. 1-15.
[CM97] RS Coulter, RW Matthews, *Bent Polynomials over Finite Fields*, Bull. Austral. Math. Soc. Vol. 56, No. 3, 1997, pp. 429-437.
[CK04] Cusick, Thomas W.; Ding, Cunsheng; Renvall, Ari Stream ciphers and number theory. Revised edition. North-Holland Mathematical Library, 66. Elsevier Science B.V., Amsterdam, 2004, ISBN: 0-444-51631
[CT99] C Carlet, On Cryptographic Propagation Criteria for Boolean Functions, Information Computation 151, pp. 32-56, 1999.
[CT04] _____, On the degree, nonlinearity, algebraic thickness etc, IEEE Trans. Inform. Theory Vol. 50, No. 9, 2004, pp. 2178 -2185.
[DN99] H Dubbertin, *Almost perfect nonlinear power functions on GF*(2^n), pp. 113-121, Finite Fields and Applications, Editors D Jungnikel et al., Springer, NY 2002
[DR01] J Daemen, V Rijmen, **The Design of Rijndael-AES**, Springer NY 2001.
[DT02], Y Desmedt et al, *Algebraic Structures and Cycling Test of Rijndael*, Fast Software Encryption Workshop, 2002.
[EH96] F Emmerich, Pseudorandom Vector Generation By The Compound Inversive Method, Math. Computation, Vol. 214, pp. 749-760, 1996.
[EL85] T ElGammal, *A subexponetial-time algorithm for computing discrete logarithm over GF*(p^2), IEEE Trans. Inform. Theory Vol. 31, 1985, pp. 473 -481.
[ER88] J Eichenaver, H Niederreiter, *On Marsaglia's lattice test for pseudorandom numbers*, Manuscripta Mathematika 62, 245-248 (1988).
[FN01] Ferguson, Niels; Schroeppel, Richard; Whiting, Doug A simple algebraic representation of Rijndael. Selected areas in cryptography, 103-111, Lecture Notes in Comput. Sci., 2259, Springer, Berlin, 2001.
[GN90] Joaquim von zur Gathen, Mark Giesbrecht, *Constructing normal bases in finite fields*, J. Symbolic Computation (1990) 10, 547-570.
[GN91] J zur van Gathen, *Tests for Permutation Polynomials*, SIAM J Computing, Vol. 20, No. 3, pp. 591-603, 1991.
[GN94] J von zur Gathen, The computational complexity of recognizing permutation functions, Proc. 26th ACM Sympos. Theory of Computing, pp. 392-401, 1994.
[GZ97] Ganz, Jürg Factoring polynomials using binary representations of finite fields. IEEE Trans. Inform. Theory 43 (1997), no. 1, 147-153.
[KN01] S Konyagin, F Pappalardi, *Enumerating Permutation Polynomials Over Finite Fields By Degree*, arXiv:math.NT/0106232v1, 2001.
[KSW85] PV Kumar, RA Scholtz, LR Welch, *Generalized Bent Functions And Their Properties*, J. Combin Ther. Ser. A, Vol. 40, 1985, pp. 90-107..
[KT03], S Kavut, M Yucel, Improved cost function in the design of Boolean functions, INDOACRYPT 2003, LNCS 2904, pp. 121-134, Springer-Verlag, NY.

[LA02] HW Lenstra, *Rijndael for algebraist*, Preprint 2002, www.math.berkeley.edu~hwl
[MS77] FJ Macwilliams, NA Sloan, **The Theory Of Error Correcting Codes**, North Holland, NY 1977.
[MS97] A.J. Menezes, P.C. van Oorschot, S.A. Vanstone et al., **Handbook of Cryptography**, CRC Press, Boca Raton,1997.
[LN97] Lidl, Rudolf; Niederreiter, Harald Finite fields. Second edition. Encyclopedia of Mathematics and its Applications, 20. Cambridge University Press, Cambridge, 1997. xiv+755 pp. ISBN: 0-521-39231-4
[LS87] H.W. Lenstra, R.J. Schoof, *Primitive Normal Bases for Finite Fields*; Math. of Computation, Vol. 48, Number 193, January 1987, pp. 217-231.
[NR71] H Niederreiter, *Orthogonal System Of Polynomials In Finite Fields*, Proc. Amer. Math. Soc. 28 (1971), 415-422.
[NE97] H Niederreiter, J Eichennauer, parallel Streams of Nonlinear Congruential Pseudo Numbers, FFA 3, 219-233.
[NR02] H Niederreiter, A Winterhof, *Lattice Structure And Linear Complexity Of Nonlinear Pseudonumbers*, AAECC 13, 319-326 (2002).
[RA01] A Rudra et al. Efficient *Rijndael Encryption Implementation with Composite Field Arithmetic*, CHES 2001, LNCS 2162, pp. 171-184, 2001.
[RL03] J Rosenthal, *A Polynomial Description Of The Rijndael Advanced Encryption Standard*, Preprint 2003, Available at www.
[RM01] F Rusky, C Miers, J Swada, *The number of irreducible polynomials with given trace*, SIAM J. Discrete Math., Vol. 14, No. 2, pp. 240-245, 2001.
[RS76], O.S. Rothaus, On bent function, J. Combin Ther. Ser. A, Vol. 20, 1976, pp. 300-305.
[SP93] V Shoup, *Fast construction of irreducible polynomials over finite fields,* ACM-SIAM 4th Proceedings Symposium on Discrete algorithms, pp. 484-492, 1993.
[SP90] _____, New Algorithm for finding irreducible polynomials over finite fields, Math. Comp. Vol. 54, No. , pp. 435-447, 1990.
[YT03] A Youssef, S Tavares, *On Some Algebraic Structures In The AES Round Function*, Preprint 2003, Available at www.
[WL69] C Wells, *The degrees of permutation polynomials over finite fields*, J. Combinatorial Theory 7, 1969, pp. 49-55.
[WR02] J Wolkerstorfer et al, *An ASIC implementation of the AES S Boxes*, Cryptographer's Track Conference 2002, LNCS 2271, pp. 67-78, 2002.

School of Computer Science and Information Systems
Pace University
Technical Report Series

EDITORIAL BOARD

The School of Computer Science and Information Systems, through the Technical Report Series, provides members of the community an opportunity to disseminate the results of their research by publishing monographs, working papers, and tutorials. *Technical Reports* is a place where scholarly striving is respected.

All preprints and recent reprints are requested and accepted. New manuscripts are read by two members of the editorial board; the editor decides upon publication. Authors, please note that production is Xerographic from the pages you have submitted. Statements of policy and mission may be found in issues #29 (April 1990) and #34 (September 1990).

Please direct submissions as well as requests for single copies to:

Allen Stix
School of CS & IS - Goldstein Academic Center
Pace University
861 Bedford Road
Pleasantville, NY 10570-2799