The Liğo Block Cipher

Isaiah Makwakwa*

March 31, 2011

Abstract

We present a modular iterated block cipher suitable for 32-bit parallel computing environments. The cipher implements a Substitution Permutation Network (SPN) on 128-bit blocks using 128-bit keys.

Keywords
composite primitive, involutionary matrices

1 Introduction

Liğo is an SPN based on a 32-bit composite primitive. It acts on 128-bit blocks using 128-bit keys. The cipher employs alternating layers of key addition, substitution and diffusion.

The substitution layer is based on affine equivalence of S-boxes while the diffusion layer is based on 4×4 *MDS matrix* multiplication.

*Email: imakwakwa@gmail.com

2 Preliminaries

2.1 Differential Cryptanalysis

Differential cryptanalysis of block ciphers was introduced by Biham and Shamir [11]. It exploits the existence of ordered pairs (α, β) of binary strings such that for any random plaintext x,

$$Prob(f(x) \oplus f(x \oplus \alpha) = \beta) > Prob(f(x) \oplus f(x \oplus \alpha) = \zeta) \tag{1}$$

for any random ζ and an n-bit block cipher $f(x)$ [2].

The *propagation ratio* of a difference sequence, for any iterated block cipher, is the total probability that a given difference sequence will result into a particular output XOR.

The attack is successful if

$$Prob(f(x) \oplus f(x \oplus \alpha) = \beta) > 2^{1-n} \tag{2}$$

for some ordered pair (α, β) [2, 8].

2.2 Linear Cryptanalysis

Linear cryptanalysis of block ciphers was introduced by Matsui in [12]. It exploits the existence of a pair of ordered binary strings (α, β) such that for any random plaintext x,

$$Prob(\alpha \cdot x \oplus f(x) \cdot \beta = 0) \neq \frac{1}{2} \tag{3}$$

where $f(x)$ is an n-bit block cipher and $\cdot$ is the inner product of binary vectors [2].

The *linear Correlation ratio* between a plaintext/ciphertext pair is the total probability that a linear function of the ciphertext is equal to a linear (or affine) function of the plaintext. A successful linear attack can be mounted if

$$Prob(\alpha \cdot x \oplus f(x) \cdot \beta = 0) > 2^{-\frac{n}{2}} \tag{4}$$

for some ordered pair (α, β) [8].

2.3 Wide Trail Strategy

In [5], it was shown that the propagation ratio (resp. linear correlation ratio) of a differential trail (resp. linear trail) can be approximated by simply multiplying the propagation (resp. correlation) ratios in the differential trail.

Therefore, the propagation ratio (resp. correlation ratio) of a differential trail (resp. linear trail) can be approximated using the number of active S-boxes in the trail [8].

2.4 MDS Matrix

Let $\mathbb{F}_q$ be a finite field, a matrix over $\mathbb{F}_q$ in which every square submatrix is non-singular is called an *MDS matrix* [6]. The linear transformation (over $\mathbb{F}_q$) defined by multiplying a vector over $\mathbb{F}_q$ with the MDS matrix is used in diffusion blocks of many ciphers (see [8, 10, 15]).

This transformation has the property that any input difference sequence with $\mathbb{F}_q$-weight t results in an output difference sequence with $\mathbb{F}_q$-weight at least $n - t + 1$ [7]. Thus the sum of the $\mathbb{F}_q$-weight of its input and output difference sequences is at least $n + 1$. This sum is called the *branch number* of the linear transformation [8].

3 The Liĝo Cipher

3.1 Representation of State in Liĝo

Let x be a 128-bit vector. The 16 byte state [8, 2] in Liĝo is respresented as

$$x = \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix} \in \left(\mathbb{F}_{2^8}^4\right)^4 \tag{5}$$

where the four byte component vectors, $x_i \in \mathbb{F}_{2^8}^4$, are defined as

$$x_i = (x_{4i}, x_{4i+1}, x_{4i+2}, x_{4i+3}).$$

and $\mathbb{F}_{2^8} = \mathbb{F}_2[x]/(x^8 + x^4 + x^4 + x^3 + x^2 + 1)$.

In particular, a binary vector $a = (a_7, a_6, a_5, a_4, a_3, a_2, a_1, a_0) \in \mathbb{F}_2^8$ [2] is considered as an element

$$a(x) = \sum_{i=0}^{7} a_i x^i \in \mathbb{F}_{2^8}.$$

3.2 The Substitution Layer

The substitution layer in Liğo is based on the inverse map over $\mathbb{F}_{2^8}$, affine equivalence of S-boxes [4, 3].

In particular, define $v(x), u(x) \in \mathbb{F}_2[x]/(x^8 + 1)$ by

$$v(x) = x^7 + x^6 + x^4 + x^3 + 1$$
$$u(x) = x^7 + x^4 + x^3 + x^2 + 1,$$

then we have $v(x)u(x) \equiv 1 \ mod \ x^8 + 1$.

Further, let $c(x) = x^7 + x^5 + x^4 + x \in \mathbb{F}_2[x]/(x^8 + 1)$ be a fixed constant, and $a(x)$ an element of $\mathbb{F}_{2^8}$, define an affine permutation ϑ on $\mathbb{F}_{2^8}$ by

$$\vartheta(a(x)) = a(x)v(x) + v(x)c(x) \ mod \ x^8 + 1. \tag{6}$$

Then the inverse permutation is given by

$$\vartheta^{-1}(a(x)) = a(x)u(x) + c(x) \ mod \ x^8 + 1 \tag{7}$$

Let $\pi : \mathbb{F}_{2^8} \rightarrow \mathbb{F}_{2^8}$ be the inverse map and set $\pi(0) = 0$, the Liğo S-box, φ, is defined by

$$\varphi(a(x)) = \pi \circ \vartheta(a(x)) \tag{8}$$

and the inverse S-box is given by

$$\varphi^{-1}(a(x)) = \vartheta^{-1} \circ \pi(a(x)) \tag{9}$$

The S-boxes, φ and φ^{-1} are given in **Table 1** and **Table 2** respectively.

Table 1: The φ S-box

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

Table 2: The φ^{-1} S-box

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

The choice of $v(x)$, $u(x)$ and $c(x)$ ensure that the resultant φ and φ^{-1} have no fixed points. Both S-boxes have maximum differential propagation ratio 2^{-6} and maximum linear correlation ratio 2^{-3} [4, 3].

3.3 Diffusion Layer

Let

$$M = \begin{pmatrix} 01 & 03 & 05 & 06 \\ 03 & 01 & 06 & 05 \\ 05 & 06 & 01 & 03 \\ 06 & 05 & 03 & 01 \end{pmatrix} \tag{10}$$

be an involutionary [9] 4×4 MDS matrix (in Hexadecimal) over $\mathbb{F}_{2^8}$, define a linear tranformation of $\mathbb{F}_{2^8}$-vectors, ϕ, by

$$\phi(x_k) = x_k M$$

where $x_k = (x_{4k}, x_{4k+1}, x_{4k+2}, x_{4k+3}) \in \mathbb{F}_{2^8}^4$. This transformation is used in the internal diffusion layer of the G composite primitive. It has branch number 5.

3.4 Key Addition Layer

Let κ denote the 11×16 subkey table from the key schedule algorithm, for any $x_k = (x_{4k}, x_{4k+1}, x_{4k+2}, x_{4k+3}) \in \mathbb{F}_{2^8}^4$, define the key addition function

$$\psi_{i,k}(x_k) = (x_{4k} \oplus \kappa[i]_{4k}, x_{4k+1} \oplus \kappa[i]_{4k+1}, x_{4k+2} \oplus \kappa[i]_{4k+2}, x_{4k+3} \oplus \kappa[i]_{4k+3})$$

where $\kappa[i]_j$ is the jth byte of $\kappa[i]$.

3.5 The G Composite Primitive

Let $x_k = (x_{4k}, x_{4k+1}, x_{4k+2}, x_{4k+3}) \in \mathbb{F}_{2^8}^4$. We define a composite primitive

$$G_{i,k}(x_k) = \phi \circ \varphi \circ \psi_{i,k} \circ \phi \circ \varphi^{-1}(x_k). \tag{11}$$

where φ and φ^{-1} are applied byte-wise on $\mathbb{F}_{2^8}$-vectors.

The inverse primitive is defined by

$$G_{i,k}^{-1}(x_k) = \varphi \circ \psi'_{i,k} \circ \phi \circ \varphi^{-1} \circ \phi(x_k) \tag{12}$$

where $\psi'_{i,k}$ is just $\psi_{i,k}$ using the subkeys $\phi(\kappa[i])$ (with ϕ applied on successive 4 bytes of $\kappa[i]$) instead of $\kappa[i]$ [9].

By the *branch number* of ϕ, any nonzero input difference sequence into $G_{i,k}$ (resp. $G_{i,k}^{-1}$) will result in an output difference sequence with at least 5 active S-boxes.

3.6 The σ Transformation

Let (b_{ij}) be a square matrix, denote by σ the matrix transpose function

$$(b_{ij})^{\sigma} = (b_{ji}) \tag{13}$$

Clearly, σ is an involution.

In fact, any permutation which ensures that the four inputs into any single G at any round (apart from the first) comes from *all* Gs at the previous round can be used to define σ.

3.7 The Number of Rounds

Liĝo is a 9 round iterative block cipher. Since complete diffusion in Liĝo takes place after two rounds, Liĝo has 5 complete diffusions.

3.8 Whitening Keys

Liĝo employs an initial and final whitening involving an exclusive-or of the plaintext and ciphertext with the $\kappa[9]$ and $\kappa[10]$ respectively.

Let $x = \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix} \in \left(\mathbb{F}_{2^8}^4\right)^4$ be a 16 byte vector, define the Liĝo initial

whitening key addition function, ω_0, by

$$\omega_0(x) = \begin{pmatrix} \psi_{9,0}(x_0) \\ \psi_{9,1}(x_1) \\ \psi_{9,2}(x_2) \\ \psi_{9,3}(x_3) \end{pmatrix} \tag{14}$$

and final whitening key addition function, ω_1, by

$$\omega_1(x) = \begin{pmatrix} \psi_{10,0}(x_0) \\ \psi_{10,1}(x_1) \\ \psi_{10,2}(x_2) \\ \psi_{10,3}(x_3) \end{pmatrix} \tag{15}$$

3.9 Encryption Function

Let $x = \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix} \in \left(\mathbb{F}_{2^8}^4\right)^4$ be any 16 byte vector, define the Liğo round function by

$$g_i(x) = \begin{pmatrix} G_{i,0}(x_0) \\ G_{i,1}(x_1) \\ G_{i,2}(x_2) \\ G_{i,3}(x_3) \end{pmatrix}^{\sigma} \tag{16}$$

Since the order of computation of the $G_{i,k}$ is immaterial, the round function in Liğo is optimised for parallel computing.

The Liğo encryption function is given by

$$f(x) = \omega_1 \circ \phi \circ \left(\bigcirc_{i=0}^{i=8} g_i\right) \circ \omega_0(x) \tag{17}$$

where the final ϕ is applied row-wise.

3.10 Decryption Function

Let $x = \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix} \in \left(\mathbb{F}_{2^8}^4\right)^4$ be any 16 byte vector, define a permutation on $\mathbb{F}_{2^8}^{16}$

$$h_i(x) = \begin{pmatrix} G_{i,0}^{-1}(x_0) \\ G_{i,1}^{-1}(x_1) \\ G_{i,2}^{-1}(x_2) \\ G_{i,3}^{-1}(x_3) \end{pmatrix} \tag{18}$$

The Liĝo decryption round function is defined by the permutation

$$g_i^{-1}(x) = h_i(x^\sigma) \tag{19}$$

Therefore, the Liĝo decryption function is defined by

$$f^{-1}(x) = \omega_0 \circ \left(\bigcirc_{i=8}^{i=0} g_i^{-1}\right) \circ \phi \circ \omega_1(x) \tag{20}$$

3.11 Round Constants

Define the Liĝo round constants, $C_i \in \mathbb{F}_{2^8}^{16}$, $(0 \leq i \leq 10)$, by

$$C_{i,j} = \varphi(7i + (16 - i)j), \quad 0 \leq j \leq 15 \tag{21}$$

where $C_{i,j}$ is the jth byte of C_i and φ is applied byte-wise.

4 The Liĝo Key Schedule Algorithm

The Liĝo key schedule algorithm is based on the Liĝo round function where the round constants are used as keys. Thus the key addition function, ψ, becomes

$$\psi_{i,k}(x_k) = (x_{4k} \oplus C_{i,4k}, x_{4k+1} \oplus C_{i,4k+1}, x_{4k+2} \oplus C_{i,4k+2}, x_{4k+3} \oplus C_{i,4k+3})$$

for any $x_k = (x_{4k}, x_{4k+1}, x_{4k+2}, x_{4k+3}) \in \mathbb{F}_{2^8}^4$ and $C_{i,j}$ the jth byte of the ith round constant defined in Section 3.11.

Let $x = \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix} \in \left(\mathbb{F}_{2^8}^4\right)^4$ be a 16 byte vector. Define a permutation on 16 bytes

$$\rho_i(x) = \phi \circ \begin{pmatrix} G_{i,0}(x_0) \\ G_{i,1}(x_1) \\ G_{i,2}(x_2) \\ G_{i,3}(x_3) \end{pmatrix}^{\sigma} \tag{22}$$

where ϕ is applied row-wise.

Let a_{-1} be the master key and set $i = 0$, then a round of the 5 round Liĝo key schedule is defined by

$$\begin{aligned} a_i &= \rho_i(a_{i-1}) \\ a_{i+1} &= \rho_{i+1}(a_{i-1} \oplus a_i) \\ i &= i + 2 \end{aligned}$$

Define the subkey table

$$\kappa[i] = a_i, \ 0 \leq i \leq 10$$

where the 128 bit values, a_i are represented as 16 byte-vectors.

5 Analysis of Liĝo

5.1 Efficient Implementation

The structure of the G composite primitive raises immediate questions about the efficiency of the Liĝo cipher. This is largely due to the double MDS matrix multiplication in G.

Since many devices perform both encryption and decryption (requiring storage for the inverse matrix), an efficient involutionary matrix presents a

significant improvement on storage. We note that the MDS matrix multiplication in Liĝo can be implemented using 12 XORs, 4 table lookups and 4 storage variables ($u[i], 0 \leq i \leq 3$) using the following pseudo-code:

```
u[0] = a3[v[0]^v[2]]; u[1] = a5[v[0]^v[1]];
u[2] = a3[v[1]^v[3]]; u[3] = a5[v[2]^v[3]];
v[3] = u[0]^u[1]^v[3]; v[2] = u[2]^u[1]^v[2];
v[1] = u[0]^u[3]^v[1]; v[0] = u[2]^u[3]^v[0];
```

where `a3[x]` and `a5[x]` are table lookups for multiplication by 03 and 05 respectively (in Hexadecimal) in $\mathbb{F}_{2^8}$.

5.2 Differential Cryptanalysis

The modular design of Liĝo allows us to use the wide trail strategy to determine the possibility of a linear or differential attack. Specifically, we only need count the number of active Gs in any differential (resp. linear) trail.

Consider a difference sequence with one active byte. In round one, exactly one G will be active. Thus, any one round differential trail will have at least 5 active S-boxes. By the diffusion properties of G, the first round output XOR will have at least one active byte.

This active byte will result in one active G in the second round under σ. In fact, in the worst case, each round will have exactly one active G. Therefore, after 5 rounds, any differential trail will have at least 5 active Gs.

Thus, any 5 round differential trail will have a propagation ratio of at most 2^{-150}. We conclude that a differential attack on a 5 round Liĝo is unlikely.

5.3 Linear Cryptanalysis

Using an analysis similar to the one in the case of differential cryptanalysis, any 5 round linear trail in Liĝo has linear correlation ratio 2^{-75}.

Therefore, a 5 round linear attack on Liĝo is unlikely.

5.4 Interpolation Attack

The Interpolation attack was introduced by Jackobsen and Knudsen in [13]. The attack seeks to construct polynomial equivalents to a cipher using plaintext/ciphertext pairs. The attack can also be modified to obtain the key.

The attack is feasible if the cost of constructing the polynomials (i.e. finding the coefficients) is less than the cost of encrypting plaintexts needed for the attack, that is, if the round functions have low algebraic degree and/or have a sparce polynomial representation over $\mathbb{F}_{2^8}$ [10, 8].

The polynomial representation of φ over $\mathbb{F}_{2^8}$ has degree 7 and 253 coefficients. The polynomial representation of φ^{-1} over $\mathbb{F}_{2^8}$ is given as (with 8 Hexadecimal coefficients)

$$\texttt{B2} + \texttt{C2}x^{127} + \texttt{22}x^{191} + \texttt{68}x^{223} + \texttt{A6}x^{247} + \texttt{28}x^{251} + \texttt{DC}x^{253} + \texttt{47}x^{254}$$

Clearly,φ and φ^{-1} are both of highest possible degree and compact which ensures that the interpolation attack against the full version Liğo is unlikely.

5.5 Related-Key Attacks

Related-key attacks [14] are successful in key schedules with low diffusion and non-linearity. The Liğo key schedule is based on the high non-linearity and complete diffusion in the ρ permutation.

These properties plus the chaining of successive outputs (subkeys) ensure that related-key attacks on Liğo is unlikely.

5.6 The Square Attack

The Square attack was introduced together with the cipher bearing the same name [15]. It is an attack on byte-oriented ciphers.

Consider 256 plaintexts all of which have one active byte on the same byte position. Then the output of the first ϕ in the active G will have 256 four byte

outputs in which every byte position has the 256 elements of $\mathbb{F}_{2^8}$ appearing exactly once. This is maintained through key addition and substitution by φ. Therefore by the linearity of ϕ, the 256 outputs XOR to zero before the action of σ at the end of the first round.

In the second round, φ^{-1} (resp. the action of σ) ensures that the output XOR is not zero at the output of the second round. Thus a two round square attack requires one to invert the following at the second round: the last ϕ, φ, the 32 bit key, the first ϕ, φ^{-1} and σ and then confirming that the result XORs to zero. Thus one needs to guess 32 (resp. at most 128 - due to having to reconstruct one G at the first round) key bits at round 2.

A three round extension will require reconstructing a single G at the second round (resp. all of the Gs at the second round). In both cases, this requires reversing the action of σ at round 2, which implies one has to invert all Gs at round 3. This in turn requires guessing 128 key bits in addition to the required 32 (resp. at most 128) bits at round 2. In total one requires a guess of 160 (resp. at most 256) key bits.

Therefore, a 4 round (including an extension at the begining) Square attack against Liğo is unlikely.

6 Conclusions

We have presented a new cryptographic primitive based on the Substitution Permutation Network. The cipher uses a single composite primitive and acts on 128 bit blocks using 128 bit keys.

Further work could be done in defining Liğo with variable length keys.

References

[1] F. J. MacWilliams N. Sloane: *The Theory of Error-Correcting Codes*, North-Holland Mathematical Library, Volume 16, pp 93-124, 2006

[2] D. Stinson: *Cryptography, Theory and Practice*, Chapman & Hall/CRC, Second Edition, pp 78, 2002

[3] C. Carlet et al: *Codes, Bent Functions and Permutations Suitable for DES-like Cryptosystems*, Design Codes and Cryptography, Kluwer Academic Publishers, 1998

[4] K. Nyberg: *Differential Uniform Mappings for Cryptology*, Advances in Cryptology, Proceedings Eurocrypt'93, LNCS 765, Springer Verlag, pp 55-64, 1994

[5] J. Daemen: *Cipher and Hash Function Design Strategies Based on Linear and Differential Cryptanalysis*, PHD Desertation, K.U. Leuven, March 1995

[6] P. Junod and S. Vaudenay: *Perfect Diffusion Primitives for Block Ciphers: Building Efficient MDS Matrices*, Selected Areas in Cryptography, LNCS 3357, pp 84-99, Springer-Verlag, 2004

[7] S. Vaudenay: *On the Need for Multipermutations: Cryptanalysis of MD4 and SAFER*, Fast Software Encryption, LNCS 1008, Springer-Verlag, pp. 286-297, 1995

[8] V. Rijmen and J. Daemen: *The Design of Rijndael*, Information Security and Cryptography, Springer-Verlag, Vol xvii No. 238, p. 48-, 2002

[9] P. Barreto and V. Rijmen: *The Khazad Legacy-Level Block Cipher*, Submission to the NESSIE Project

[10] B. Schneier et al: *Twofish: A 128-Bit Block Cipher*, Counterpane Systems, 1998

[11] E. Biham and Shamir: *Differential Cryptanalysis of DES-like Cryptosystems*, Journal of Cryptology, Vol. 4 No. 1, pp 3-72, 1991

[12] M. Matsui: *Linear Cryptanalysis Method for DES Cipher*, Advances in Cryptology, Proceedings Eurocrypt'93, LNCS 765, Springer Verlag, pp 386-397, 1994

[13] T. Jacobsen and L. R. Knudsen: *The Interpolation Attack on Block Ciphers*, Fast Software Encryption, LNCS 1267, Springer-Verlag, pp 28-40, 1997

[14] E. Biham: *New Types of Cryptanalytic Attacks Using Related Keys*, Journal of Cryptology, Vol. 7 No. 4, pp 229-246, 1994

[15] J. Daemen, L.R. Knudsen and V. Rijmen: *The Block Cipher Square*, Fast Software Encryption, LNCS 1267, Springer-Verlag, pp 149-165, 1997