共通鍵128ビットブロック暗号　HyRAL
The 128bit blockcipher HyRAL

平田　耕蔵
Kouzou Hirata

株式会社ローレルインテリジェントシステムズ
東京都港区虎ノ門1-1-10
Laurel Intelligent Systems Co.,Ltd,
1-10 Toranomon 1-chome Minato-ku Tokyo, k-hirata2453@lbm.co.jp

J-Ver2

あらまし

- 先に発表した共通鍵ブロック暗号HyRAL[1]では、差分確率および線形確率において安全性を保証する上界値が得られなかった 。[2]
- その為、安全性を確保すべく改良を行い再度発表するものである。今回の改良の主たるものは、ｓ－ｂoxと基本関数となる*f*関数である。
- その他は、先に発表したものと設計概念は異なっていない。データ長１２８ビット、鍵長１２８ビット、１９２～２５６ビットをサポートしたもので、一般化Ｆeistel型構造を採ったものである。

[1]平田耕蔵、共通鍵暗号HyRAL16、信学技報　ISEC　2006-76（2006-09）P29
[2]五十嵐保隆・金子敏信・福林直人、共通鍵ブロック暗号HyRALの線形攻撃耐性評価、信学技報　ISEC　2006-157 2007-03）P99

目次

1. 記号の説明
2. HyRALの構造概要
3. s-boxの生成
4. 基本関数(*fi*関数)
5. 拡大関数
6. 全体構造
7. Key Generation
8. Key Scheduli
9. 安全性について

１．記号の説明[１]

- １．１　データ構造

  x（1bit）：1または0

  x（8bit）：斜体小文字 x =（[上位] x7、x6、x5、x4、x3、x2、x1、x0 [下位]）

  $\boldsymbol{X}$（32bit）：斜体太字　$\boldsymbol{X}$ =（[上位] $x0$、$x1$、$x2$、$x3$ [下位]）

  $\mathbf{X}$（128bit）：大文字太字 $\mathbf{X}$=（[上位] $\boldsymbol{X}0$、$\boldsymbol{X}1$、$\boldsymbol{X}2$、$\boldsymbol{X}3$ [下位]）

- １．２　演算

  ⊕：Xor

  ＋：算術加算

  ≫i：iビット右シフト

  8 bit データの乗算は、GF(2^8)の元を多項式表現し、法多項式
  　x^8+x^4+x^3+x+1を使用。

  多項式表現においては、0ビット目（最下位）をx^0の係数とする。

１．記号の説明[２]

- １．３　関数名

  G1、**G2**、**F1**、**F2**（128bit入力）

  基本関数 *fi*（32bit入力）

- １．４　鍵の表示

  Rki（128bit）　ラウンド鍵（i = 1～9）

  IKi（128bit）　*fi* 関数入力鍵（i = 1～6）

  KMi（128bit）　中間鍵（i = 1～4）

  OKi（128bit）　秘密鍵（i = 1～2）

２．HyRALの構造概要[１]

- ２．１　HyRALの全体構造

  HyRALの全体構造には鍵長128bitの場合と、129～256bitが存在する。

  – ４系列の一般化Feistel型で**G1**、**G2**、**F1**、**F2**と4つの異なるアルゴリズムの関数を接続している。
  – これまでのFeistel型とは異なり拡大鍵を逆順に使用し同一のアルゴリズムで復号する方法ではなく、複合時には各々の逆関数を使用して復号する。

- ２．２　基本関数（ fi関数）

  – 入力32bit出力32bitのものであり入力のバイト転置により８種類ある。
  fi（ i= 1、2、3、4、5、6、7、8）

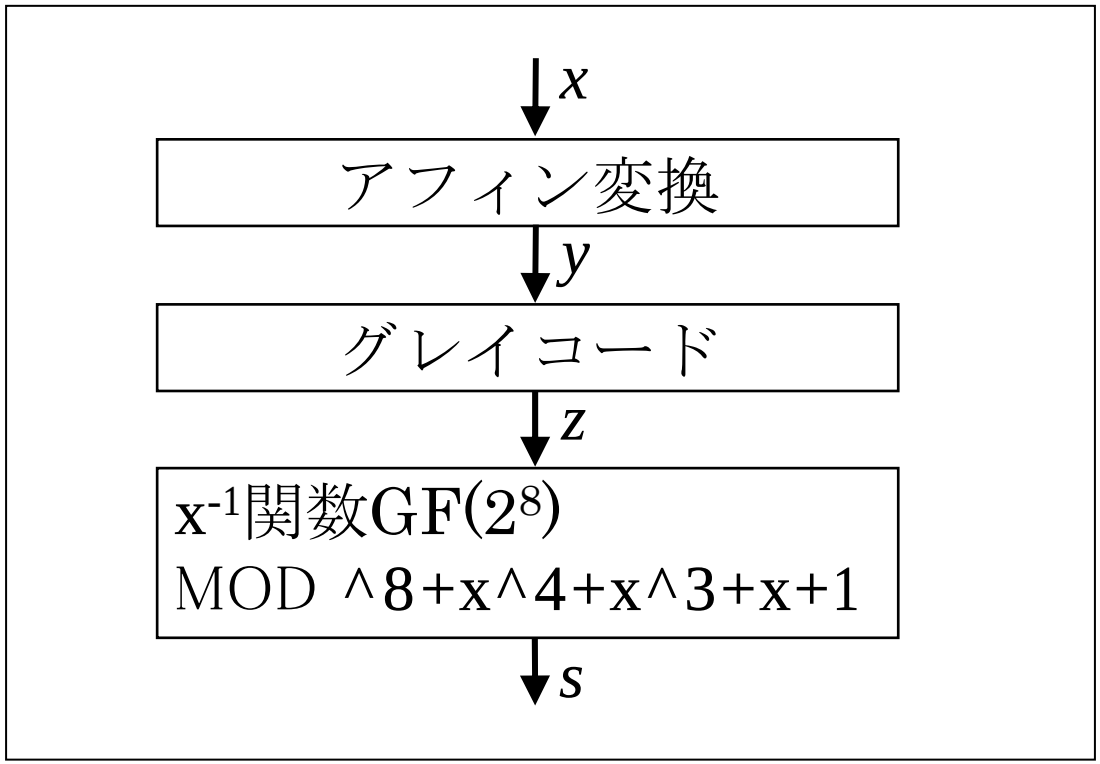
２．HyRALの構造概要[２]

- ２．３　拡大関数
  - 基本関数（ *fi*関数）を用いた**G1**、**G2**、**F1**、**F2**の４種類の拡大関数がある。

- ２．４　鍵生成部
  - 鍵処理モードでは **OKi**と定数**Li**（*i*= 1、2、3）を**G1**、**G2**関数に入力しその出力を**KMi**とする。
  - 拡大鍵の割り当てでは**KMi**から拡大キーである**RKi**と**IKi**を生成する。

3. s-boxの生成［1］

- s-boxは非線形層であるGF(2^8)上の逆関数$s = z^{-1}$と線形関数であるアフィン変換式1とグレイコード変換式2の組み合わせで構成される。(ただし$z^{-1}=0$)
- s-boxの最大線形及び最大差分確率は2^{-6}である。



3. s-boxの生成[2]

- 3．1　アフィン変換 $y = (x + 64)\mathrm{MOD}256$

$$\begin{pmatrix} y_7 \\ y_6 \\ y_5 \\ y_4 \\ y_3 \\ y_2 \\ y_1 \\ y_0 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_7 \\ x_6 \\ x_5 \\ x_4 \\ x_3 \\ x_2 \\ x_1 \\ x_0 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

３．s-boxの生成[３]

- ３．２　グレイコード$z = y \oplus (y \gg 1)$

$$\begin{pmatrix} z_7 \\ z_6 \\ z_5 \\ z_4 \\ z_3 \\ z_2 \\ z_1 \\ z_0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} y_7 \\ y_6 \\ y_5 \\ y_4 \\ y_3 \\ y_2 \\ y_1 \\ y_0 \end{pmatrix}$$

3. s-boxの生成[4]

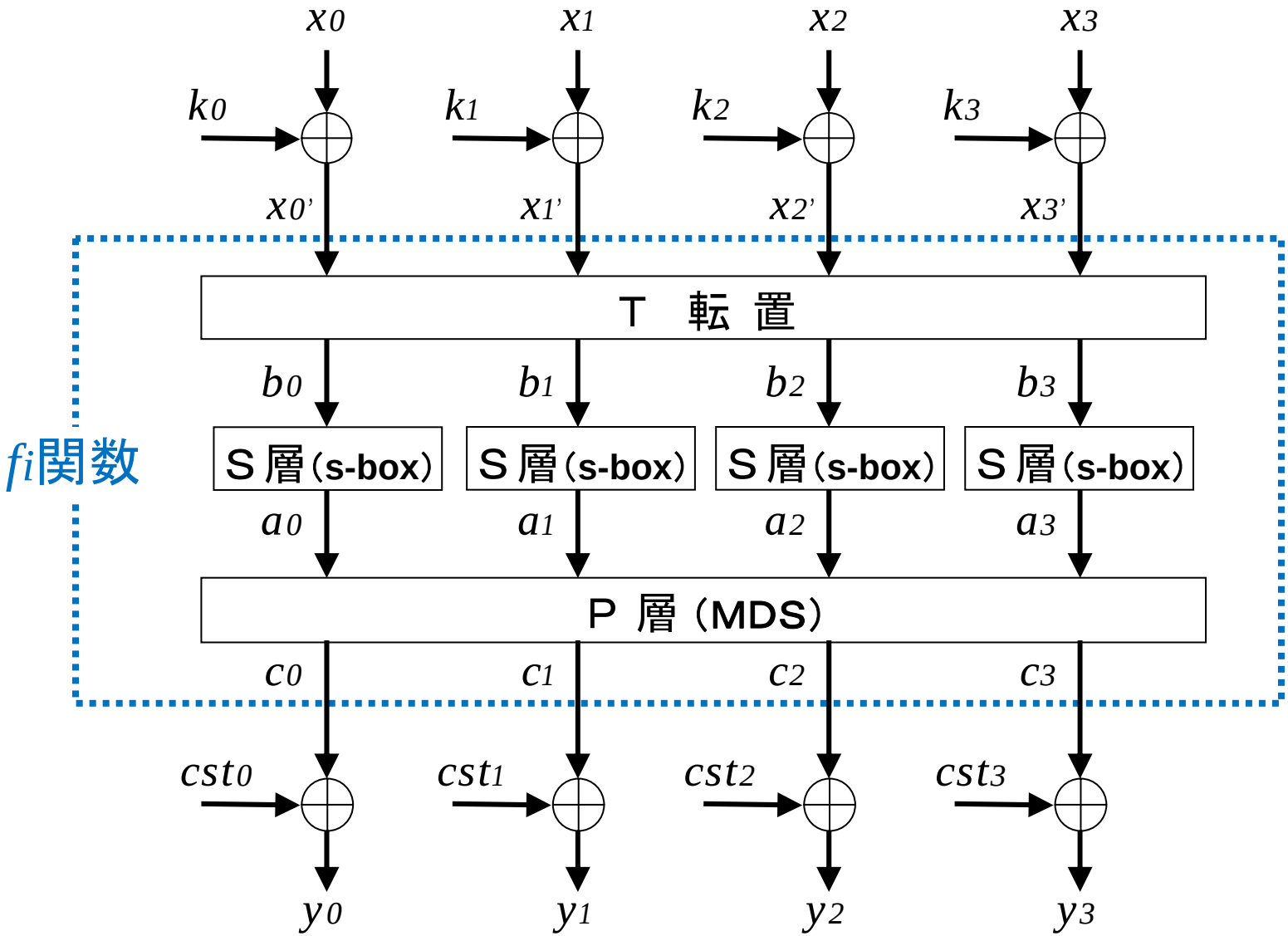
- 3. 3　s-box

上位＼下位	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	16	5e	d3	af	36	43	a6	49	33	93	3b	21	91	df	47	f4
1	b6	70	06	d0	81	82	fa	a1	10	b5	3c	ba	97	85	b7	79
2	ed	5c	ca	05	87	bf	24	4c	51	ec	17	61	22	f0	3e	18
3	a7	64	13	ab	e9	09	25	54	2d	31	69	f5	37	67	fe	1d
4	0b	28	a3	2f	e4	0f	d4	da	1b	fc	e6	ac	53	04	27	a9
5	94	8b	d5	c4	90	6b	f8	9d	c5	db	ea	e2	ae	63	07	7a
6	5b	23	34	38	03	8c	46	68	cd	1a	1c	41	7d	a0	9c	dd
7	08	4e	e3	d7	1e	b3	50	5d	c6	0e	ad	cf	d6	eb	0d	b1
8	fb	7c	c3	2e	65	48	b8	8f	ce	e7	62	d2	12	4a	c8	26
9	a5	8e	3d	76	86	57	bc	bd	11	75	71	78	1f	ef	e0	0c
A	de	6a	6d	32	84	72	8a	d8	f9	dc	9a	89	9f	88	14	2a
B	9b	9e	d9	95	b9	a4	02	f7	96	73	56	be	7f	80	7e	83
C	00	01	f6	8d	7b	d1	52	cb	b0	e1	c7	e5	29	c0	4f	e8
D	58	3f	cc	fd	ee	b2	40	ff	99	2b	5f	60	aa	4b	b4	74
E	2c	45	6c	92	66	42	39	f3	77	bb	19	59	20	6f	35	f2
F	c1	0a	15	98	a2	c2	44	30	55	4d	c9	a8	5a	f1	6e	3a

4. 基本関数(*fi*関数) [1]

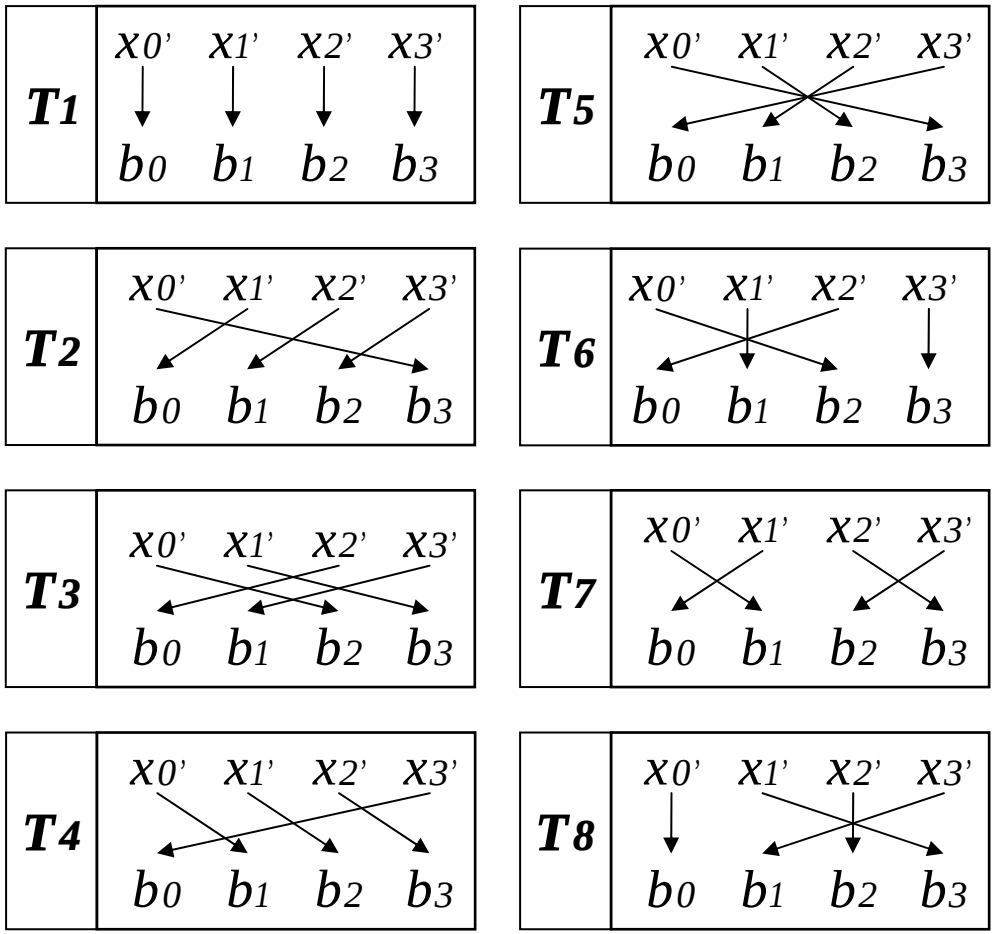
- *fi*関数は、SP構造の32bit入出力関数で、最大線形・差分確率が2^{-6}の8ビットS-boxを4並列で用い、P層は4バイト入出力のMDS行列とする。
- AESの様に、巡回型のMDS行列と同一s-boxの並置のS層を用いた場合、*fi*関数の入力がバランスしている時、出力もバランスしている性質を持つ。
- *fi*関数に供給される段鍵がバランスしている場合、32bit変数 $\boldsymbol{X}$ = ($x0$、$x1$、$x2$、$x3$) において、$x0 = x1 = x2 = x3$が成り立つ時、$\boldsymbol{X}$はバランスしていると呼ぶ。複数段の*fi*関数縦接続においても入出力のバランス性が維持される。
- バランスしている段鍵の発生確率は、無視できるという理由で、この性質を考慮する必要は無いのかもしれないが、未知の攻撃法に結びつく可能性も考え、HyRALでは、非巡回型のMDS行列を用いることにする。但し、関数に入力される直前に転置を行い、転置の仕方が8通りあるので、転置種類毎に$T1$～$T8$と表記しておく。

4. 基本関数(*fi*関数) [2]


x0
x1
x2
x3
k0
k1
k2
k3
x0'
x1'
x2'
x3'
T 転 置
b0
b1
b2
b3
fi関数
S 層(s-box)
S 層(s-box)
S 層(s-box)
S 層(s-box)
a0
a1
a2
a3
P 層 (MDS)
c0
c1
c2
c3
cst0
cst1
cst2
cst3
y0
y1
y2
y3

4. 基本関数(*fi*関数) [3]

- T 転置



- P層(MDS)

要素の計算はGF(2^8)

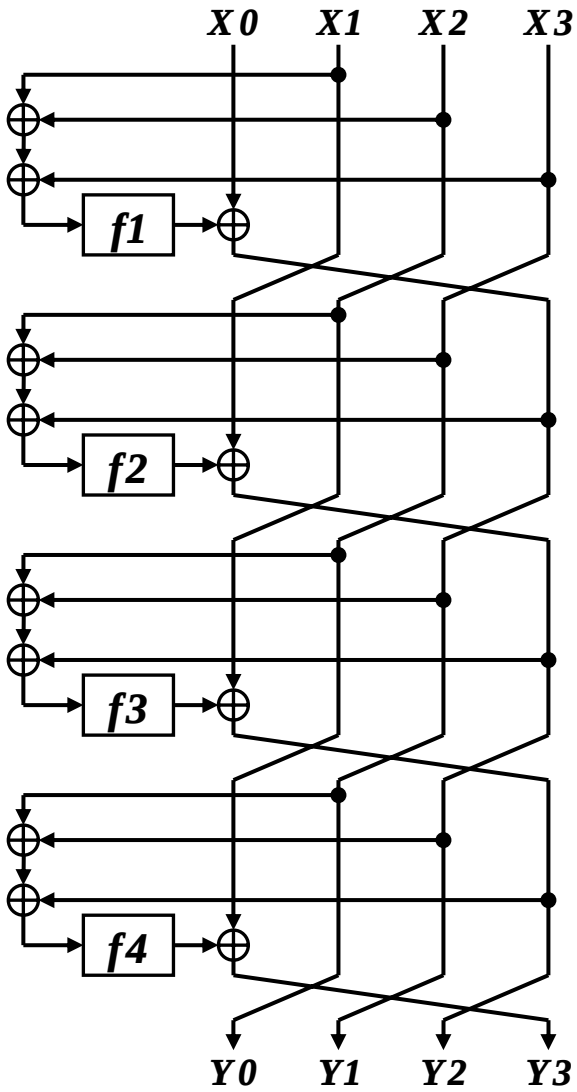
$$\begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} 3 & 3 & 2 & 1 \\ 1 & 2 & 2 & 2 \\ 7 & 3 & 1 & 2 \\ 7 & 4 & 5 & 3 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix}$$

参考：逆行列

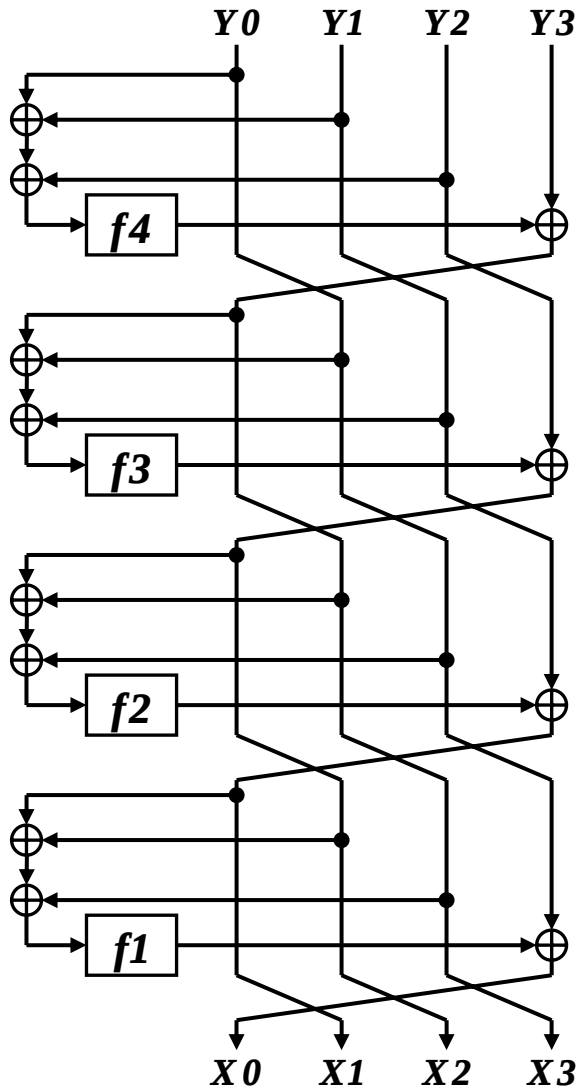
$$\begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} 46 & cb & cb & cb \\ d4 & 93 & e8 & 1e \\ c0 & 0a & 23 & 87 \\ 37 & fc & 23 & 71 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix}$$

5. 拡大関数[1]

- **G1暗号**

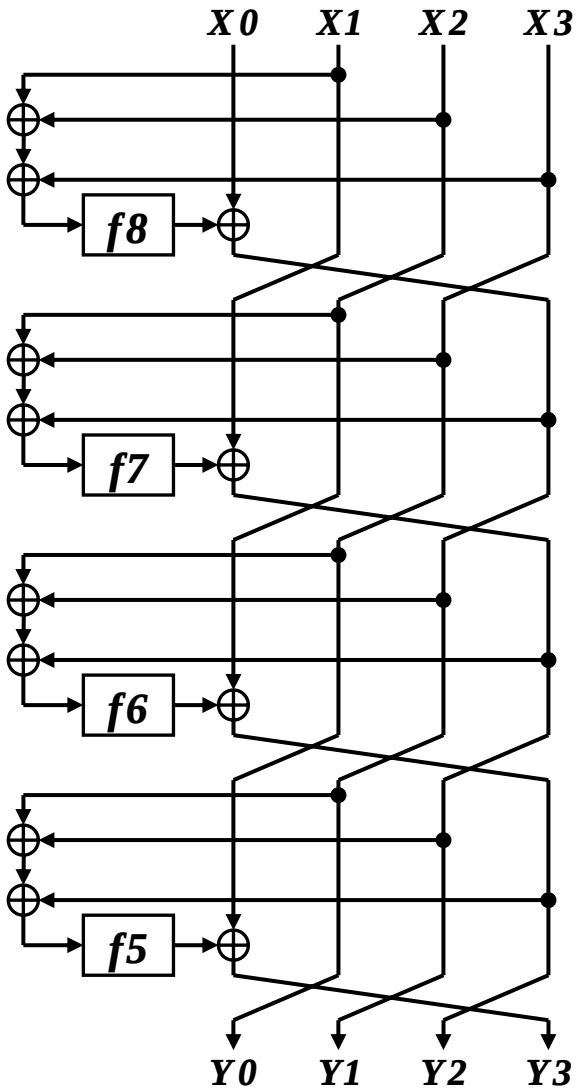


- **$G1^{-1}$復号**

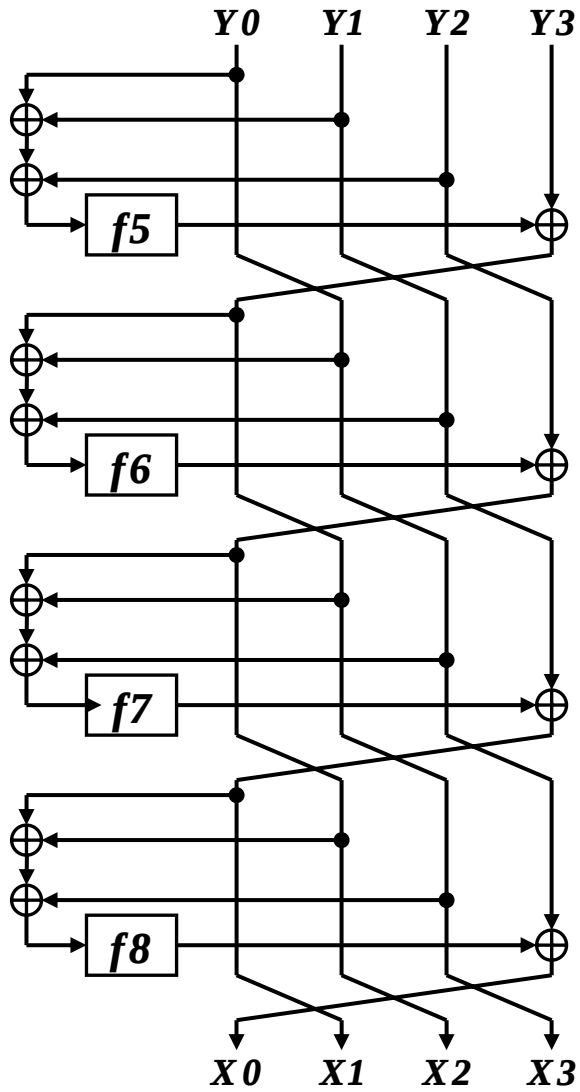


5. 拡大関数[2]

- **G2暗号**

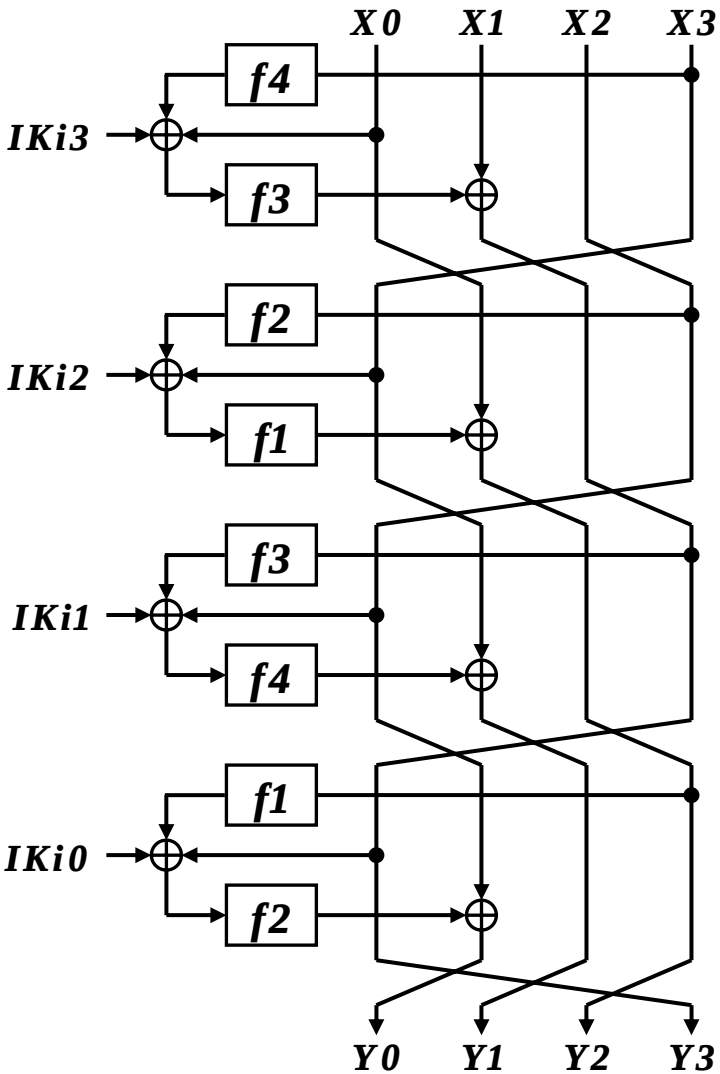


- **$G2^{-1}$復号**

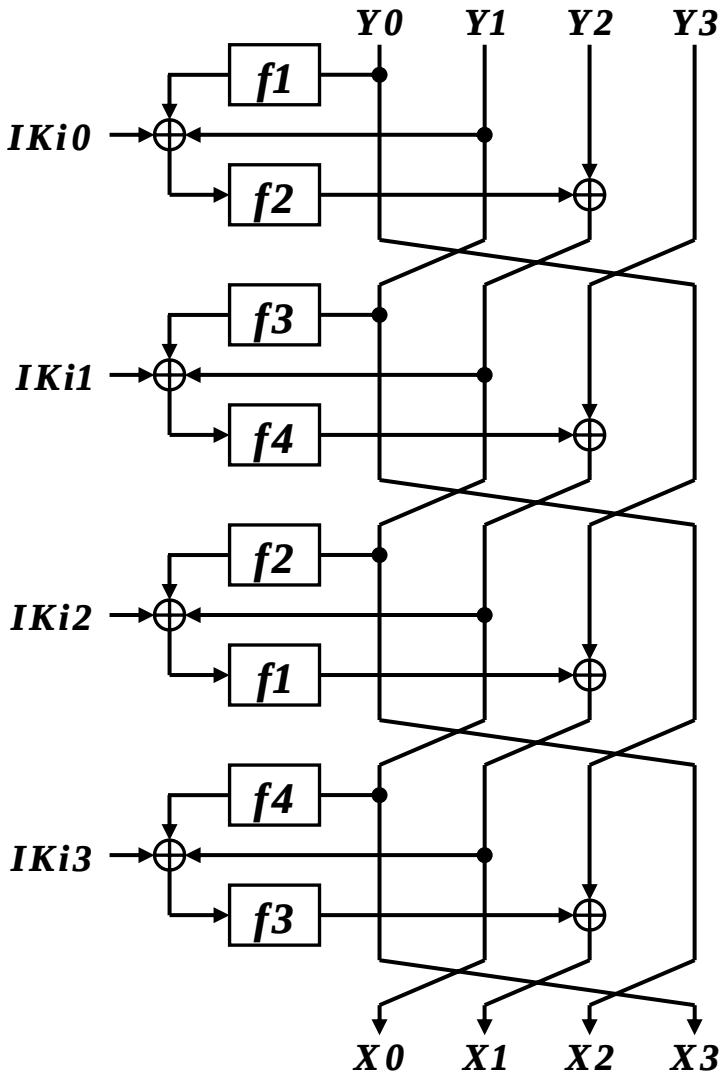


5. 拡大関数[3]

- **F1暗号**

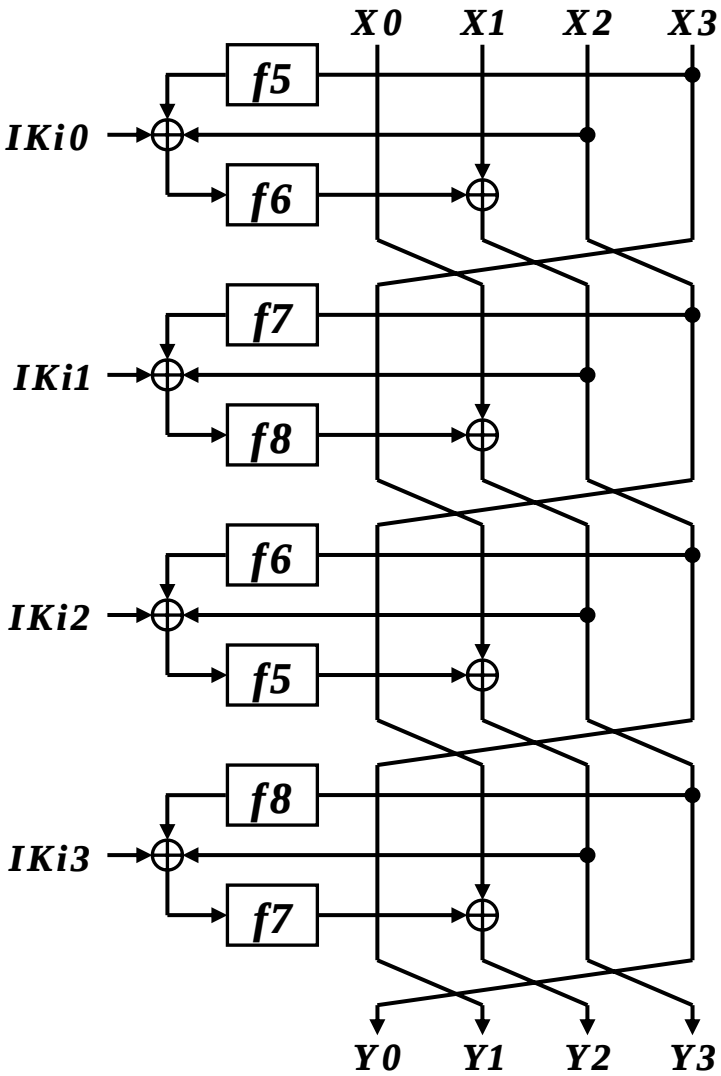


- **F1^{-1}復号**

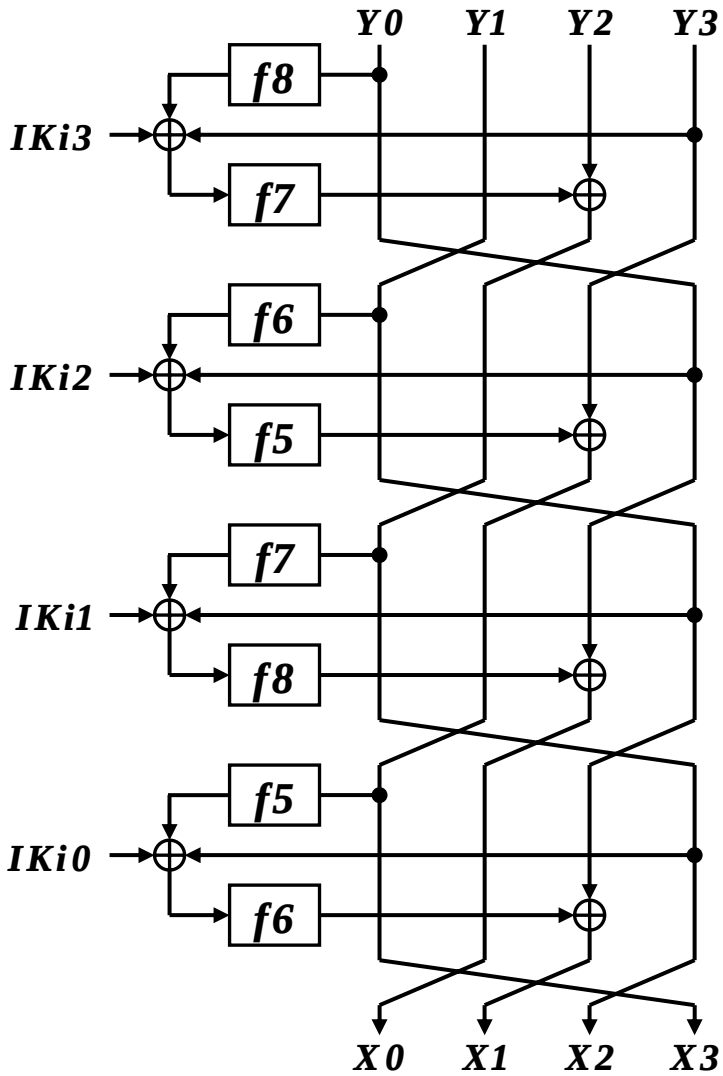


5. 拡大関数[4]

- **F2暗号**

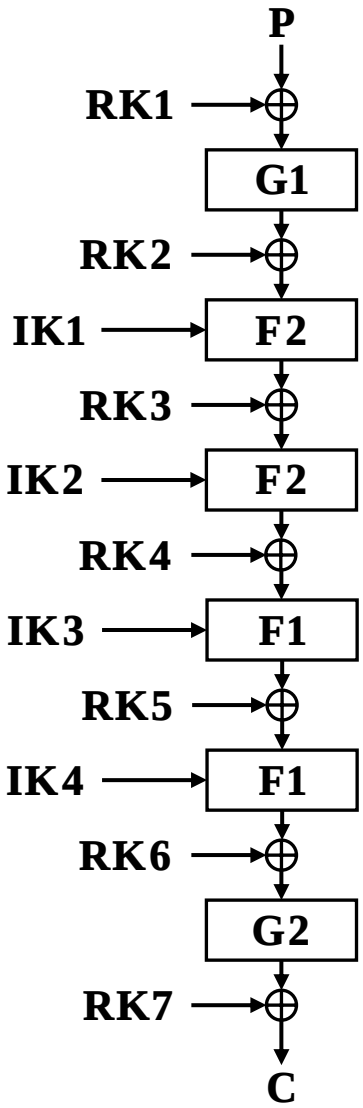


- **F2^{-1}復号**

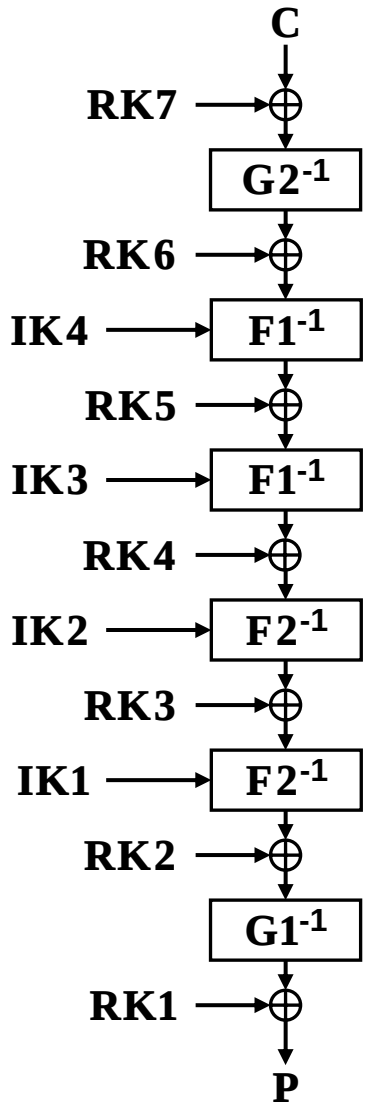


6. 全体構造[1]

- Key128ﾋﾞｯﾄ暗号

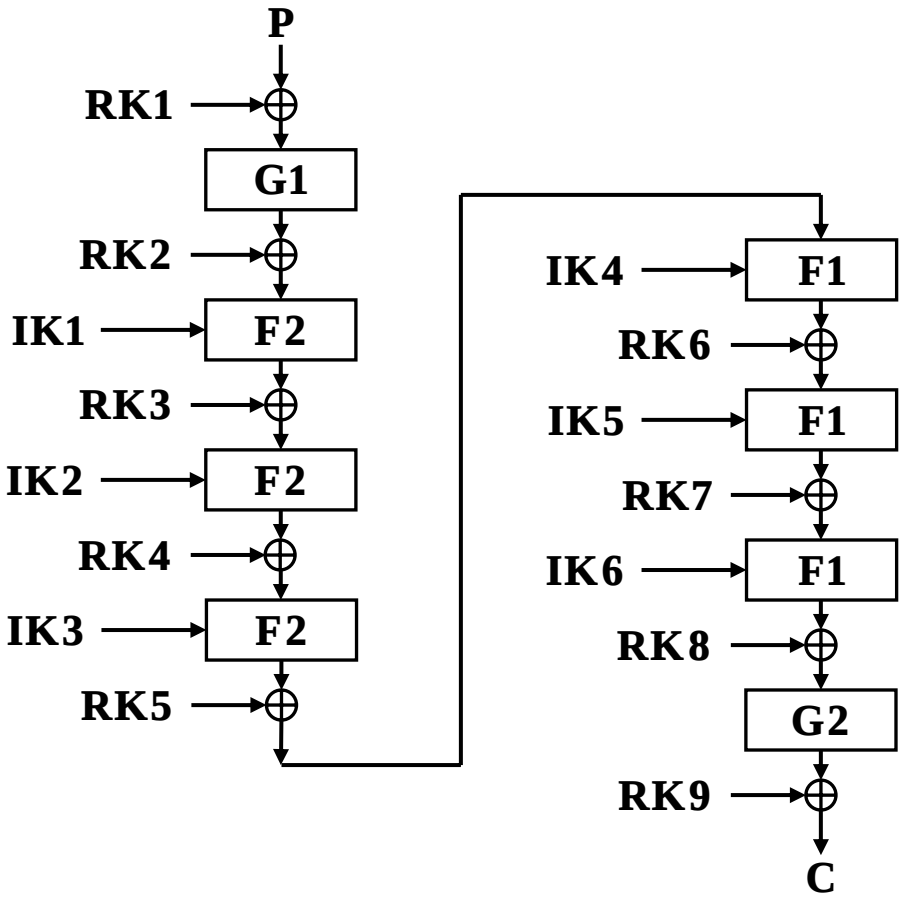


- Key128ﾋﾞｯﾄ復号



6. 全体構造[2]

- Key 192～256ﾋﾞｯﾄ暗号

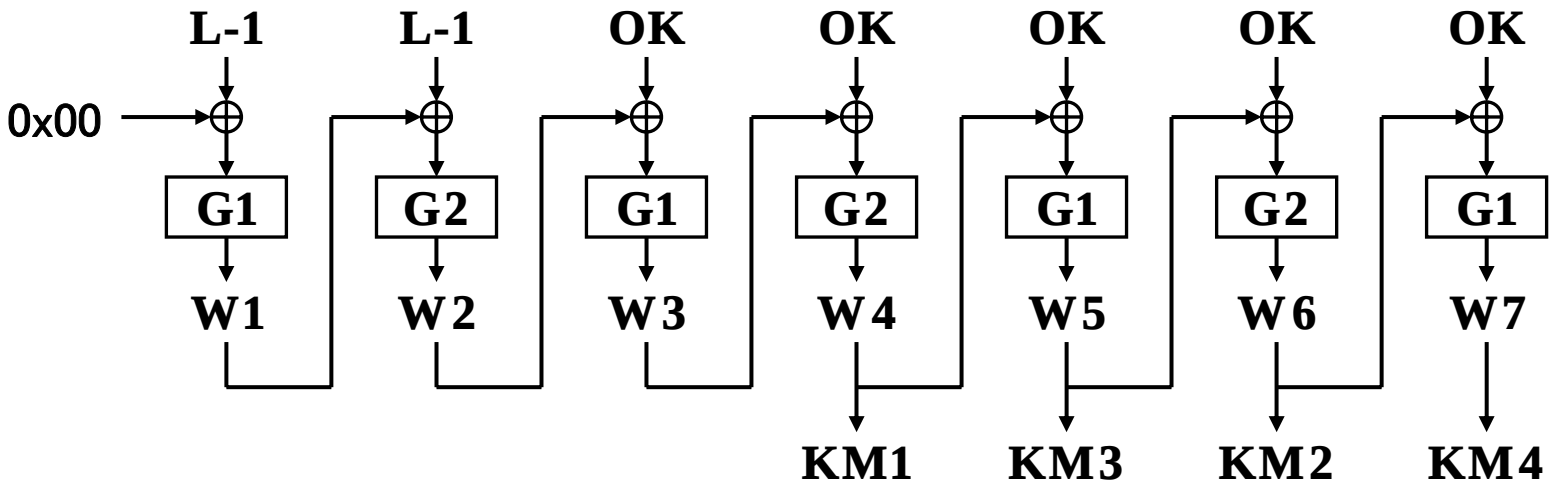


復号はKey128ﾋﾞｯﾄと同様に結果から遡る

7. Key Generation［1］

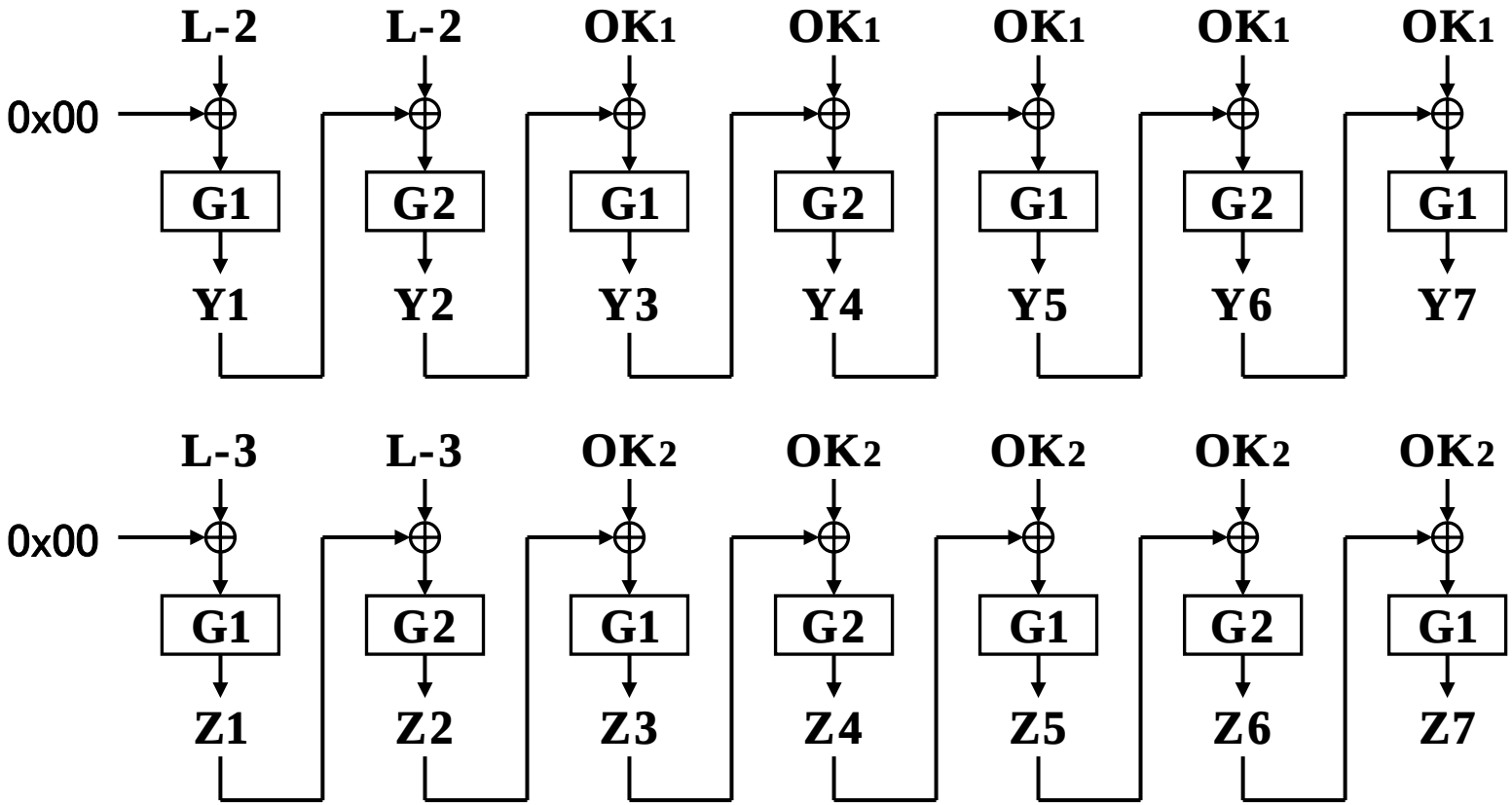
- LABEL
- L－1（48 59 52 41 4C 40 40 40 00 00 00 00 00 00 00 12）
- L－2（48 59 52 41 4C 40 40 40 00 00 00 00 00 00 00 22）
- L－3（48 59 52 41 4C 40 40 40 00 00 00 00 00 00 00 32）

- Key128ﾋﾞｯﾄ



7. Key Generation［2］

- Key 192～256ﾋﾞｯﾄ



KM1=Y4 ⊕Z4　　KM2=Y6 ⊕Z6
KM3=Y5 ⊕Z5　　KM4=Y7 ⊕Z7

8. Key Scheduling(拡大キーの割り当て)［1］

- Single Key Mode

	KM1	X0 X1 X2 X3
⊕	KM4	X3 X2 X1 X0
	RK4	X0 X1 X2 X3

	KM3	X3 X2 X1 X0
⊕	KM4	X0 X1 X2 X3
	RK1	X0 X1 X2 X3

	KM1	X3 X2 X1 X0
⊕	KM2	X0 X1 X2 X3
	RK7	X0 X1 X2 X3

	KM1	X1 X2 X3 X0
⊕	KM4	X0 X3 X2 X1
	IK1	X0 X1 X2 X3

	KM3	X0 X3 X2 X1
⊕	KM4	X1 X2 X3 X0
	RK6	X0 X1 X2 X3

	KM1	X0 X3 X2 X1
⊕	KM2	X1 X2 X3 X0
	RK2	X0 X1 X2 X3

	KM1	X2 X3 X0 X1
⊕	KM4	X0 X1 X2 X3
	IK4	X0 X1 X2 X3

	KM3	X0 X1 X2 X3
⊕	KM4	X2 X3 X0 X1
	RK3	X0 X1 X2 X3

	KM1	X0 X1 X2 X3
⊕	KM2	X2 X3 X0 X1
	RK5	X0 X1 X2 X3

	KM1	X3 X0 X1 X2
⊕	KM4	X3 X0 X1 X2
		未使用

	KM3	X3 X0 X1 X2
⊕	KM4	X3 X0 X1 X2
	IK2	X0 X1 X2 X3

	KM1	X3 X0 X1 X2
⊕	KM2	X3 X0 X1 X2
	IK3	X0 X1 X2 X3

8. Key Scheduling(拡大キーの割り当て)［2］

- Double Key Mode(1)

KM1	X0 X1 X2 X3	
⊕ KM4	X3 X2 X1 X0	
RK5	X0 X1 X2 X3	

KM3	X3 X2 X1 X0	
⊕ KM4	X0 X1 X2 X3	
RK1	X0 X1 X2 X3	

KM1	X3 X2 X1 X0	
⊕ KM2	X0 X1 X2 X3	
RK9	X0 X1 X2 X3	

KM1	X1 X2 X3 X0	
⊕ KM4	X0 X3 X2 X1	
IK1	X0 X1 X2 X3	

KM3	X0 X3 X2 X1	
⊕ KM4	X1 X2 X3 X0	
RK8	X0 X1 X2 X3	

KM1	X0 X3 X2 X1	
⊕ KM2	X1 X2 X3 X0	
RK2	X0 X1 X2 X3	

KM1	X2 X3 X0 X1	
⊕ KM4	X0 X1 X2 X3	
IK6	X0 X1 X2 X3	

KM3	X0 X1 X2 X3	
⊕ KM4	X2 X3 X0 X1	
RK3	X0 X1 X2 X3	

KM1	X0 X1 X2 X3	
⊕ KM2	X2 X3 X0 X1	
RK7	X0 X1 X2 X3	

KM1	X3 X0 X1 X2	
⊕ KM4	X3 X0 X1 X2	
	未使用	

KM3	X3 X0 X1 X2	
⊕ KM4	X3 X0 X1 X2	
RK4	X0 X1 X2 X3	

KM1	X3 X0 X1 X2	
⊕ KM2	X3 X0 X1 X2	
RK6	X0 X1 X2 X3	

8. Key Scheduling(拡大キーの割り当て)［3］

- Double Key Mode(2)

	KM2	X3 X2 X1 X0
⊕	KM3	X0 X1 X2 X3
	IK2	X0 X1 X2 X3

	KM2	X0 X3 X2 X1
⊕	KM3	X1 X2 X3 X0
	IK3	X0 X1 X2 X3

	KM2	X0 X1 X2 X3
⊕	KM3	X2 X3 X0 X1
	IK4	X0 X1 X2 X3

	KM2	X3 X0 X1 X2
⊕	KM3	X3 X0 X1 X2
	IK5	X0 X1 X2 X3

９．安全性について

- 今回の改良により、8bitトランケーションによる線形および差分の最大特性確率の上界は、次のように向上し、安全性を示すことが出来た。
- Single Key Mode (秘密鍵128bit)でそれぞれ2^{-228}、2^{-222}。Double Key Mode (秘密鍵192bitおよび256 bit) でそれぞれ2^{-318}、2^{-342}である。
- SCIS２０１０暗号と情報セキュリティシンポジウム（2010年1月）において東京理科大学金子教室より、8bitトランケーションによる線形および差分の最大特性確率の上界は、前述の様に発表された。
- 今後他の攻撃に対する耐性についても評価していく予定である。