The 128bit blockcipher HyRAL
(Hybrid Randomization Algorithm)

Kouzou Hirata

Laurel Intelligent Systems Co., Ltd,
1-10 Toranomon 1-chome Minato-ku Tokyo, k-hirata2453@lbm.co.jp
Outline

- In the previous announcement of common key block cipher HyRAL \(^{[1]}\), the upper bound which defines the guarantee of safety on a difference probability and a linear probability was not successfully obtained.\(^{[2]}\)
- Therefore, HyRAL has been continuously improved to acquire the margin of safety and is now ready to be released again. Main improvements are made upon “s-box” and a fundamental function “f function”.
- Additionally, the design concepts formerly reported and reporting in this article are virtually the same. Generalized Feistel structure is adopted in order to support the data length of 128 bits and the key lengths of 128, 192, and 256 bits.

---

\(^{[1]}\) k-hirata 、Symmetric-key encryptionHy RAL16 、ISEC 2006-76 (2006-09) P29

\(^{[2]}\) y-igarashi・t-kaneko・n-hukubayashi 、Linear attack tolerance evaluation of Common key block cipher HyRAL 、ISEC 2006-157 2007-03) P99
Contents

1. Explanation of signs
2. Structural outline of HyRAL
3. Generation of “s-box”
4. Fundamental function ($f_i$)
5. Large function
6. Block Structure
7. Key Generation
8. Key Schedule
9. Security analysis
1. Explanation of signs [1]

• 1. 1 Data Structures
  \(x\) (1bit): “1” or “0”
  \(x\) (8bit): Slanting small letter
    \(x = ([\text{MSb}] x_7, x_6, x_5, x_4, x_3, x_2, x_1, x_0 [\text{LSb}])\)
  \(X\) (32bit): Slanting Bold \(X = ([\text{MSB}] x_0, x_1, x_2, x_3 [\text{LSB}])\)
  \(X\) (128bit): Bolding capital letter \(X = ([\text{MSW}] X_0, X_1, X_2, X_3 [\text{LSW}])\)

• 1. 2 Operations
  \(\oplus\): Xor
  \(+\): Arithmetic Addition
  \(\gg i\): i bit right shift
  The multiplication by the 8 bit data expresses the origin of \(\text{GF}(2^8)\) in the polynomial, and uses Irreducible polynomial \(x^8 + x^4 + x^3 + x + 1\).
  In the polynomial expression, the 0bit (LSb) is assumed to be a coefficient of “\(x^0\)”.

1. Explanation of signs [2]

• 1. 3 Function names
  \( G_1, G_2, F_1, F_2 \) (I/O 128bit)
  Fundamental function \( f_i \) (I/O 32bit)

• 1. 4 Display of keys
  \( R_{ki} \) (128bit) Round key \( (i = 1 \sim 9) \)
  \( I_{Ki} \) (128bit) Intermediate key \( (i = 1 \sim 6) \)
  \( K_{Mi} \) (128bit) Key Material \( (i = 1 \sim 4) \)
  \( O_{Ki} \) (128bit) Original Key \( (i = 1 \sim 2) \)
2. Structural outline of HyRAL [1]

• 2. 1 Globular conformation of HyRAL

In overall structure of HyRAL, key lengths of 128, 129 to 256 bits exist.

– $G_1$, $G_2$, $F_1$, and $F_2$ are connected with the functions of four different algorithms by generalized Feistel types of four affiliates.

– It is different from a Feistel type to decode by the same algorithm by the use of the difference expansion key to reverse order but combining it, each inverse function is used and decoded.

• 2. 2 Fundamental function (\(f_i\))

– There are 8 sorts by byte transposition of the input for input 32bit and output 32bit.

\(f_i\) \((i = 1, 2, 3, 4, 5, 6, 7, 8)\)
2. Structural outline of HyRAL [2]

• 2. 3  Large function
  – There are four kinds of G1, G2, F1, F2 Large function with a Fundamental function (f1 function)

• 2. 4  Key Generation parts
  – In the key processing mode, OKi and constant Li (i= 1,2,3) are entered to G1 and G2 function and the output is assumed to be KMi.
  – In allocations of the expansion key, two expansion keys RKi and IKi are generated from KMi.

- “s-box” is composed of the combinations of affine-conversion type 1 of a linear function, gray code conversion type 2, and inverse function $s = z^{-1}$ on GF($2^8$) of the nonlinear layer.
  (However, $z^{-1} = 0$)
- Maximum linear and maximum difference probability of s-box are $2^{-6}$.

- 3.1 Affine-conversion \( y = (x + 64) \text{MOD} 256 \)

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

- 3.2 Gray code \( z = y \oplus (y \gg 1) \)

\[
\begin{pmatrix}
Z_7 \\
Z_6 \\
Z_5 \\
Z_4 \\
Z_3 \\
Z_2 \\
Z_1 \\
Z_0
\end{pmatrix}
= \begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1
\end{pmatrix}
\begin{pmatrix}
y_7 \\
y_6 \\
y_5 \\
y_4 \\
y_3 \\
y_2 \\
y_1 \\
y_0
\end{pmatrix}
\]

- 3.3 s-box

<table>
<thead>
<tr>
<th>s-box</th>
<th>low8bit</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0 1 2 3 4 5 6 7 8 9 A B C D E F</td>
</tr>
<tr>
<td>0</td>
<td>16 5e d3 af 36 43 a6 49 33 93 3b 21 91 df 47 f4</td>
</tr>
<tr>
<td>1</td>
<td>b6 70 06 d0 81 82 fa a1 10 b5 3c ba 97 85 b7 79</td>
</tr>
<tr>
<td>2</td>
<td>ed 5c ca 05 87 bf 24 4c 51 ec 17 61 22 f0 3e 18</td>
</tr>
<tr>
<td>3</td>
<td>a7 64 13 ab e9 09 25 54 2d 31 69 f5 37 67 fe 1d</td>
</tr>
<tr>
<td>4</td>
<td>0b 28 a3 2f e4 0f d4 da 1b fc e6 ac 53 04 27 a9</td>
</tr>
<tr>
<td>5</td>
<td>94 8b d5 c4 90 6b f8 9d c5 db ea e2 ae 63 07 7a</td>
</tr>
<tr>
<td>6</td>
<td>5b 23 34 38 03 8c 46 68 cd 1a 1c 41 7d a0 9c dd</td>
</tr>
<tr>
<td>7</td>
<td>08 4e e3 d7 1e b3 50 5d c6 0e ad cf d6 eb 0d b1</td>
</tr>
<tr>
<td>8</td>
<td>fb 7c c3 2e 65 48 b8 8f ce e7 62 d2 12 4a c8 26</td>
</tr>
<tr>
<td>9</td>
<td>a5 8e 3d 76 86 57 bc bd 11 75 71 78 1f ef e0 0c</td>
</tr>
<tr>
<td>A</td>
<td>de 6a 6d 32 84 72 8a d8 f9 dc 9a 89 9f 88 14 2a</td>
</tr>
<tr>
<td>B</td>
<td>9b 9e d9 95 b9 a4 02 f7 96 73 56 be 7f 80 7e 83</td>
</tr>
<tr>
<td>C</td>
<td>00 01 f6 8d 7b d1 52 cb b0 e1 e7 e5 29 c0 4f e8</td>
</tr>
<tr>
<td>D</td>
<td>58 3f cc fd ee b2 40 ff 99 2b 5f 60 aa 4b b4 74</td>
</tr>
<tr>
<td>E</td>
<td>2c 45 6c 92 66 42 39 f3 77 bb 19 59 20 6f 35 f2</td>
</tr>
<tr>
<td>F</td>
<td>c1 0a 15 98 a2 c2 44 30 55 4d c9 a8 5a f1 6e 3a</td>
</tr>
</tbody>
</table>
4. Fundamental function \((f_i)\) [1]

- P layer is assumed to be MDS procession of 4byte I/O by the maximum, linear, and the difference probability's fourthly using eight bits “s-box” of \(2^{-6}\) for the \(f_i\) function in parallel by 32bit I/O function of the SP structure.
- In case the input of \(f_i\) function does the balance when S layer of the juxtaposition of same “s-box” is used with the MDS procession of the round type like AES, it has a characteristic that the output also does the balance.
- At 32bit variable \(X\) when the steps key supplied to the \(f_i\) function does the balance \(X = (x_0, x_1, x_2, x_3)\), It is called that \(X\) does the balance when \(x_0 = x_1 = x_2 = x_3\) consists. The balance of I/O is maintained in the \(f_i\) function length connection of two or more steps.
- Due to the probability of occurrence of the steps key that does the balance is very rare and ignorable, the nature of the probability might not be considered. However the rare possibility is assumed to be potentially relevant to unknown attack methods, HyRAL implements the MDS procession of non-round type. Each transposition kind is expressed as \(T_1 \sim T_8\) because there are eight kinds of methods of the transposition, which transpose immediately before entering to the function.
4. Fundamental function (\(f_i\)) [2]
4. Fundamental function \((f_i) \[3\]\)

- **Transposition**

<table>
<thead>
<tr>
<th>(T_1)</th>
<th>(T_5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(x_0') (x_1') (x_2') (x_3')</td>
<td>(x_0') (x_1') (x_2') (x_3')</td>
</tr>
<tr>
<td>(b_0) (b_1) (b_2) (b_3)</td>
<td>(b_0) (b_1) (b_2) (b_3)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(T_2)</th>
<th>(T_6)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(x_0') (x_1') (x_2') (x_3')</td>
<td>(x_0') (x_1') (x_2') (x_3')</td>
</tr>
<tr>
<td>(b_0) (b_1) (b_2) (b_3)</td>
<td>(b_0) (b_1) (b_2) (b_3)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(T_3)</th>
<th>(T_7)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(x_0') (x_1') (x_2') (x_3')</td>
<td>(x_0') (x_1') (x_2') (x_3')</td>
</tr>
<tr>
<td>(b_0) (b_1) (b_2) (b_3)</td>
<td>(b_0) (b_1) (b_2) (b_3)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(T_4)</th>
<th>(T_8)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(x_0') (x_1') (x_2') (x_3')</td>
<td>(x_0') (x_1') (x_2') (x_3')</td>
</tr>
<tr>
<td>(b_0) (b_1) (b_2) (b_3)</td>
<td>(b_0) (b_1) (b_2) (b_3)</td>
</tr>
</tbody>
</table>

- **P Layer (MDS)**

Calculation element “GF(2^8)”

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

Reference:

Reverse Transformation

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

- $G_1$ (encrypt)

- $G_1^{-1}$ (decrypt)
5. Large function [2]

- \( G_2 \) (encrypt)

- \( G_2^{-1} \) (decrypt)
5. Large function [2]

- \textbf{F}_1 (encrypt)

- \textbf{F}_{1^{-1}} (decrypt)
5. Large function [4]

- $F_2$ (encrypt)

- $F_2^{-1}$ (decrypt)
6. Block Structure [1]

- Key 128bit (encrypt)

- Key 128bit (decrypt)

- Key 192～256bit (encrypt)

Decoding traces back from the result as well as the 128bit code
7. Key Generation [1]

- **LABEL**
- \( L-1 (48\ 59\ 52\ 41\ 4C\ 40\ 40\ 40\ 00\ 00\ 00\ 00\ 00\ 00\ 12) \)
- \( L-2 (48\ 59\ 52\ 41\ 4C\ 40\ 40\ 40\ 00\ 00\ 00\ 00\ 00\ 00\ 22) \)
- \( L-3 (48\ 59\ 52\ 41\ 4C\ 40\ 40\ 40\ 00\ 00\ 00\ 00\ 00\ 00\ 32) \)
- **Key128bit**
7. Key Generation [2]

- Key 192,256bit

\[ \begin{align*}
G_1^{L-2} & \rightarrow Y_1 \\
G_2^{L-2} & \rightarrow Y_2 \\
G_1^{OK_1} & \rightarrow Y_3 \\
G_2^{OK_1} & \rightarrow Y_4 \\
G_1^{OK_1} & \rightarrow Y_5 \\
G_2^{OK_1} & \rightarrow Y_6 \\
G_1^{OK_1} & \rightarrow Y_7 \\
\end{align*} \]

\[ \begin{align*}
G_1^{L-3} & \rightarrow Z_1 \\
G_2^{L-3} & \rightarrow Z_2 \\
G_1^{OK_2} & \rightarrow Z_3 \\
G_2^{OK_2} & \rightarrow Z_4 \\
G_1^{OK_2} & \rightarrow Z_5 \\
G_2^{OK_2} & \rightarrow Z_6 \\
G_1^{OK_2} & \rightarrow Z_7 \\
\end{align*} \]

\[ \begin{align*}
K_{M1} & = Y_4 \oplus Z_4 \\
K_{M2} & = Y_6 \oplus Z_6 \\
K_{M3} & = Y_5 \oplus Z_5 \\
K_{M4} & = Y_7 \oplus Z_7 \\
\end{align*} \]
8. Key Scheduling [1]

- **Single Key Mode**

| KM1 | X₀ X₁ X₂ X₃ |
| KM4 | X₃ X₂ X₁ X₀ |
| RK₄ | X₀ X₁ X₂ X₃ |
| KM1 | X₁ X₂ X₃ X₀ |
| KM4 | X₀ X₃ X₂ X₁ |
| IK₁ | X₀ X₁ X₂ X₃ |
| KM1 | X₂ X₃ X₀ X₁ |
| KM4 | X₀ X₁ X₂ X₃ |
| IK₄ | X₀ X₁ X₂ X₃ |
| KM1 | X₃ X₀ X₁ X₂ |
| KM4 | X₃ X₀ X₁ X₂ |
| IK₂ | X₀ X₁ X₂ X₃ |
| KM1 | X₃ X₂ X₁ X₀ |
| KM2 | X₀ X₁ X₂ X₃ |
| RK₇ | X₀ X₁ X₂ X₃ |
| KM1 | X₀ X₃ X₂ X₁ |
| KM2 | X₁ X₂ X₃ X₀ |
| RK₂ | X₀ X₁ X₂ X₃ |
| KM1 | X₂ X₃ X₀ X₁ |
| KM2 | X₂ X₃ X₀ X₁ |
| RK₅ | X₀ X₁ X₂ X₃ |
| KM1 | X₃ X₀ X₁ X₂ |
| KM2 | X₃ X₀ X₁ X₂ |
| IK₃ | X₀ X₁ X₂ X₃ |

Unused
8. Key Scheduling [2]

- Double Key Mode(1)

<table>
<thead>
<tr>
<th>KM1</th>
<th>X₀ X₁ X₂ X₃</th>
<th>KM3</th>
<th>X₃ X₂ X₁ X₀</th>
<th>KM1</th>
<th>X₃ X₂ X₁ X₀</th>
</tr>
</thead>
<tbody>
<tr>
<td>KM4</td>
<td>X₃ X₂ X₁ X₀</td>
<td>KM4</td>
<td>X₀ X₁ X₂ X₃</td>
<td>KM2</td>
<td>X₀ X₁ X₂ X₃</td>
</tr>
<tr>
<td>RK5</td>
<td>X₀ X₁ X₂ X₃</td>
<td>RK1</td>
<td>X₀ X₁ X₂ X₃</td>
<td>RK9</td>
<td>X₀ X₁ X₂ X₃</td>
</tr>
<tr>
<td>KM1</td>
<td>X₁ X₂ X₃ X₀</td>
<td>KM3</td>
<td>X₀ X₃ X₂ X₁</td>
<td>KM1</td>
<td>X₀ X₃ X₂ X₁</td>
</tr>
<tr>
<td>KM4</td>
<td>X₀ X₃ X₂ X₁</td>
<td>KM4</td>
<td>X₁ X₂ X₃ X₀</td>
<td>KM2</td>
<td>X₁ X₂ X₃ X₀</td>
</tr>
<tr>
<td>IK1</td>
<td>X₀ X₁ X₂ X₃</td>
<td>RK8</td>
<td>X₀ X₁ X₂ X₃</td>
<td>RK2</td>
<td>X₀ X₁ X₂ X₃</td>
</tr>
<tr>
<td>KM1</td>
<td>X₂ X₃ X₀ X₁</td>
<td>KM3</td>
<td>X₀ X₁ X₂ X₃</td>
<td>KM1</td>
<td>X₀ X₁ X₂ X₃</td>
</tr>
<tr>
<td>KM4</td>
<td>X₀ X₁ X₂ X₃</td>
<td>KM4</td>
<td>X₂ X₃ X₀ X₁</td>
<td>KM2</td>
<td>X₂ X₃ X₀ X₁</td>
</tr>
<tr>
<td>IK6</td>
<td>X₀ X₁ X₂ X₃</td>
<td>RK3</td>
<td>X₀ X₁ X₂ X₃</td>
<td>RK7</td>
<td>X₀ X₁ X₂ X₃</td>
</tr>
<tr>
<td>KM1</td>
<td>X₃ X₀ X₁ X₂</td>
<td>KM3</td>
<td>X₃ X₀ X₁ X₂</td>
<td>KM1</td>
<td>X₃ X₀ X₁ X₂</td>
</tr>
<tr>
<td>KM4</td>
<td>X₃ X₀ X₁ X₂</td>
<td>KM4</td>
<td>X₃ X₀ X₁ X₂</td>
<td>KM2</td>
<td>X₃ X₀ X₁ X₂</td>
</tr>
<tr>
<td></td>
<td>Unused</td>
<td>RK4</td>
<td>X₀ X₁ X₂ X₃</td>
<td>RK6</td>
<td>X₀ X₁ X₂ X₃</td>
</tr>
</tbody>
</table>
8. Key Scheduling [3]

- Double Key Mode(2)

<table>
<thead>
<tr>
<th>KM2</th>
<th>X3 X2 X1 X0</th>
</tr>
</thead>
<tbody>
<tr>
<td>KM3</td>
<td>X0 X1 X2 X3</td>
</tr>
<tr>
<td>IK2</td>
<td>X0 X1 X2 X3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>KM2</th>
<th>X0 X3 X2 X1</th>
</tr>
</thead>
<tbody>
<tr>
<td>KM3</td>
<td>X1 X2 X3 X0</td>
</tr>
<tr>
<td>IK3</td>
<td>X0 X1 X2 X3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>KM2</th>
<th>X0 X1 X2 X3</th>
</tr>
</thead>
<tbody>
<tr>
<td>KM3</td>
<td>X2 X3 X0 X1</td>
</tr>
<tr>
<td>IK4</td>
<td>X0 X1 X2 X3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>KM2</th>
<th>X3 X0 X1 X2</th>
</tr>
</thead>
<tbody>
<tr>
<td>KM3</td>
<td>X3 X0 X1 X2</td>
</tr>
<tr>
<td>IK5</td>
<td>X0 X1 X2 X3</td>
</tr>
</tbody>
</table>
9. Security analysis

- As a result of 8-bit truncation analysis, upper bound of LCP$_{\text{MAX}}$ and DCP$_{\text{MAX}}$ is improved as follows, and these improvements proficiently proves margins of safety.
- Single Key Mode (private key 128bit) - $2^{-228}$, and $2^{-222}$ respectively. Double Key Mode (Private key 192bit and 256 bit) - $2^{-318}$, and $2^{-342}$ respectively.
- Maximum upper bound for linear and differential characteristic probability based on 8-bit truncation, shown above, were announced by the Tokyo University of Science, Kaneko lab at the SCIS2010 Symposium on Cryptography and Information Security held in January 2010.
- The tolerance evaluation tests to other potential attacks are scheduled.