Self Evaluation : Hierocrypt–L1

Toshiba Corporation

October 1, 2001

Contents

1 Introduction

Hierocrypt is a family of block ciphers whose data randomizing parts consist of the nested SPN structure, which is a hierarchical SPN structure where a higher-level S-box consists of the lower-level SP network [13, 19, 18, 7]. It is easy for the nested SPN structure to achieve a sufficient security level against the differential/linear cryptanalysis, as the number of active S-boxes in each level can be assured hierarchically[19].

The most recent versions of Hierocrypt are Hierocrypt–3 (128-bit block) and Hierocrypt–L1 (64-bit block) [18, 7]. This paper reports the result of our evaluation on security and performance for Hierocrypt–L1.

2 Security

2.1 Security against differential and linear cryptanalysis

The differential and linear cryptanalysis are effective against general symmetric block cryptosystems, the former of which was proposed by Biham and Shamir [3] and the latter proposed by Matsui [16]. The most important security measure is the number of plaintext-ciphertext pairs need for the cryptanalysis. The number of pairs is known to be the same order of the inverse of the maximum differential/linear probability of data randomizing part removing 2 or 3 rounds of both ends. As it is difficult to calculate their exact values, approximate values are often used, which are estimated on the basis of their characteristic probabilities where the summation for intermediate differences or mask patterns are not taken. For Hierocrypt–3 proposed in this paper, the maximum differential and linear characteristic probabilities can be easily evaluated (or bounded) by the minimum number of active S-boxes, as the cipher consists of the nested SPN structure. Furthermore, we found that the provable security for a two-round SPN structure proven by Hong et al. [8, 14] is applicable to two consecutive rounds of Hierocrypt–3. The provable security leads to the rigid upper bound of the maximum differential and linear probabilities. We will show the result of evaluation and proper numbers of rounds for respective key sizes.

2.1.1 Definition of differential and linear probabilities

The maximum differential probability for the function f is given as follows.

$$dp^f \equiv \max_{\Delta x \neq 0, \Delta y} \frac{\#\{x | f(x) \oplus f(x \oplus \Delta x) = \Delta y\}}{2^n}. \tag{1}$$

Similarly, the maximum linear probability for the function f is given as follows.

$$lp^f \equiv \max_{\Gamma x, \Gamma y \neq 0} \left| 2 \cdot \frac{\#\{x | x \cdot \Gamma x = f(x) \cdot \Gamma y\}}{2^n} - 1 \right|^2 . \tag{2}$$

The maximum linear probability is defined so that the optimal value is the same as that for the maximum differential probability.

2.1.2 S-box property

The S-box map of Hierocrypt–3 is equivalent to the combination of the following three transformations.

(a) bit permutation

(b) power operation x^{247} over GF(2^8)

(c) Affine transformation $ax + b$ over GF(2^8)

As the distributions of differential and linear probabilities are invariant for the transformations (a) and (c),

$$dp^S = lp^S = 2^{-6} . \tag{3}$$

2.1.3 Active S-box number

At first we consider about the S-box number for differential. When a higher-level S-box XS is active, that is, there is at least one bit whose differential value is not 0, no less than 5 (lower-level) S-boxes are active. And two consecutive rounds have no less than 3 active higher-level S-boxes (XS). Therefore, two consecutive rounds contain no less than 15 active S-boxes as shown in Proposition 2 of [13, 19]. Finally, the maximum differential characteristic probability of two consecutive rounds of Hierocrypt–3 $DP^{2\mathrm{R}}$ is bounded as follows.

$$DP^{2\mathrm{R}} \leq \left(dp^S\right)^{15} = \left(2^{-6}\right)^{15} = 2^{-90} \ . \tag{4}$$

A similar result is obtained for the maximum linear probability by the substitution : input differential bit $\rightarrow$ output mask bit. That is, the minimum active S-box number in two consecutive rounds is 15, and its maximum characteristic probability is bounded by 2^{-90}.

$$LP^{2\mathrm{R}} \leq \left(lp^S\right)^{15} = \left(2^{-6}\right)^{15} = 2^{-90} \ . \tag{5}$$

When the differential cryptanalysis is applied, 2-round or 3-round attack is used. Therefore, an appropriate round number is the round number which has a sufficiently small characteristic probability plus 2 or 3. As one round of Hierocrypt–3 corresponds to two rounds of the usual cipher, four rounds are regarded as sufficient for the additional rounds. Therefore, four (=2+2) rounds seems to be sufficient

However, the round number should be longer as the key size is twice as that of block size. We assume that the round number which can be cryptanalyzed increases by one by increasing the key search for one higher-level S-box. As the key bit number for one higher-level S-box is 64 ($= 32 \times 2$), we consider that one round should be increased for 128-bit key . Therefore, we consider that 5-round Hierocrypt–L1 is sufficiently secure, based on the minimum active S-box number.

2.1.4 Evaluation based on the provable security theorem

Hong et al. proves the following theorem about the security of 2-round SPN structure with an MDS diffusion layer [8].

Theorem 1 Consider a 2-round SPN structure (SPS) with n parallel S-boxes in one layer which satisfies the following conditions.

- the extended keys are independent and do not have any biases.
- the branch number of diffusion layer is $n+1$ (MDS)
- the maximum differential/linear probability is dp (lp).

Then, the maximum differential/linear probability for the SPS structure does not exceed dp^n / lp^n. ¶

Next, consider two rounds of Hierocrypt–3 (SPSP). As the second diffusion layer does not change the distribution of differential/linear probability, the maximum differential/linear probability for two rounds (SPSP) is the same as that for SPS in the higher-level. As the branch number of higher-level SPS, its maximum differential/linear probability is bounded by $\left(dp^{XS}\right)^4$ / $\left(lp^{XS}\right)^4$ when the maximum differential/linear probability for the higher-level S-box (XS) is dp^{XS} / lp^{XS}. Similarly, as XS consists of a lower-level SPS with branch number 3, its maximum differential/linear probability is bounded as follows.

$$\left(dp^{XS}\right) \leq \left(dp^S\right)^4 = \left(2^{-6}\right)^2 = 2^{-24}$$
$$\left(lp^{XS}\right) \leq 2^{-24}$$

Therefore, the maximum differential/linear probability for two rounds of Hierocrypt–3 does not exceeds $\left(2^{-24}\right)^2 = 2^{-48}$. In summary, if there is no deficiency in the key scheduling, the minimum plaintext-ciphertext number for differential/linear cryptanalysis against four-round Hierocrypt–L1 is about 2^{48}, when two-round attack is used.

But, probability 2^{-48} does not means that differential/linear cryptanalysis is not applicable. We give an approximate evaluation of the maximum probability for 3 and 4 rounds in the following. We already know that the maximum probability for two rounds is bounded by 2^{-48}. When the round number is three, at least one higher-level S-box XS of the additional round, whose maximum probability is bounded by

2^{-24}. Thus, an approximate evaluation of the maximum probability is bounded by $2^{-48} \times 2^{-24} = 2^{-72}$. This probability is not enough to deny the applicability of the differential/linear cryptanalysis. However, as key length is 64-bit larger than the key length, for additional one round should be added to achieve a sufficient security against differential/linear cryptanalysis. Therefore, we conclude that the round number of Hierocrypt–L1 should be 6 $(= 3 + 2 + 1)$ at least.

Table 1: Appropriate minimum round number based on provable security

key length	minimum round number	probability bound	
–	4	2	2^{-48}
–	5	3	2^{-72}
128 bits	6	4	2^{-96}

The evaluation of differential/linear probability is considered to be more precise than the evaluation of full characteristic evaluation.

2.2 SQUARE Attack

The SQUARE attack is a chosen-plaintext attack, which is applied to the SQUARE cipher and other SQUARE-like ciphers. The basic attack and its extensions of applicable round numbers by key estimation have been proposed[4].

The manner of counting rounds in Hierocrypt–L1 is different from that of the SQUARE cipher. To avoid confusion, we introduce a definition of a number of layers instead of the number of rounds. The number of layers is defined as a number of S-box layers. That is, in Hierocrypt–L1, two layers correspond to a round. On the other hand, in the SQUARE cipher, a layer corresponds to a round.

When the property of key scheduling part is not used, the SQUARE attack is effective up to 7 layers(7 rounds) for 128-bit key SQUARE cipher and Rijndael cipher[4], and up to 8 layers(8 rounds) for 192-bit key and 256-bit key Rijndael cipher[5].

On the other hand, the SQUAREattack is applicable up to 7 S-box layers(3.5 rounds) for Hierocrypt–L1of 64-bit block and 128-bit key [20].

By our self-evaluation in the last fiscal year, we estimated that the SQUARE attack is applicable against Hierocrypt–L1 up to 5-layer for 128-bit key[12]. Recently, we applyed the improvement by Ferguson et al. , and elongated the applicable number of layers to 7 S-box layers (3.5 rounds)[20]. Quite the same result was derived independently by Barreto et al.[21] (See 2.10.1).

As the number of layers for Hierocrypt–L1 is 12-layer(6-round) for 128-bit key, we consider that Hierocrypt–L1 is sufficiently secure against the SQUARE attack, despite of the improvement of the attack.

2.2.1 Fundamental Attacks against Rijndael

In this subsection, we describe fundamental SQUAREattacks against Rijndael. First, some fundamental ideas are defined. Nest, fundametal propositions are given, and finally, the fundamental attacks are explained.

Layer numbering The index for the i-th layer, which start from the byte substitution through layer-key addition, is given as i. The index for the first key addition is 0.

State block and layer key The input to i-th layer is defined as state block $b^{(i-1)}$, and the layer key just before the input is defined as $k^{(i-1)}$.

State byte SQUARE-like ciphers has a nonlinear layer composed of sixteen 8-bit input/output S-boxes. Each 8bit corresponding to a S-box is called a state byte. Each state byte takes 256 possible values.

Λ set According to the definition of Daemen et al., we introduce a definition of a Λ set as follows.

1. Elements of a Λ set are states of a system composed of 16 state bytes.
2. A Λ set is a set of 256 states.
3. By restricting elements of a Λ set to each state byte, a state byte takes either all 256 possible state or a fixed state.

Active byte and Passive byte By restricting elements of a Λ set to a state byte, if the state byte takes two or more states, we call it as an active byte. If the state byte takes only one state, we call it as a passive byte.

Balancedness over a Λ set Corresponding to all 256 states belonging to a Λ set in some layer, if a exclusive or of all states of some state byte in some layer vanishes, we call the state byte as balanced over the Λ set.

Proposition 1 *A transformation layer composed of bijective mappings for every state bytes maps a Λ set to a Λ set.*

For examples, a nonlinear transformation layer composed of a bijective S-boxes, this nonlinear transformation layer maps a Λ set of a Λ. A key addition layer by bitwise exclusive or maps a Λ set of a Λ set.

Proposition 2 *For a Λ set in some layer, the active bytes in the layer are balanced over the Λ set.*

Proposition 3 *A transformation layer composed of bijective mappings for every state bytes maps state bytes balanced over a Λ set to state bytes balanced over the Λ set.*

For example, a nonlinear transformation layer is composed of bijective S-boxes, this layer maps state bytes balanced over a Λ set to byte states balanced over a Λ set. A key addition layer by bitwise exclusive or maps state bytes balanced over a Λ set of state bytes balanced over the Λ set.

Proposition 4 *For a linear transformation composing a linear transformation layer, if all the active bytes input to the layer are balanced over a Λ set, state bytes output from the layer are balanced over the Λ set.*

Proposition 5 *If an input to some linear transformation layer is a Λ set, all the active bytes of the output from the layer balanced over the Λ set.*

Basic attack Consider a Λ set composed of a single active state byte and other 15 passive state bytes in input of the first layer. Assume that the Λ set transforms as a Λ set until some layer. By Proposition 2, a state byte-input into a linear transformation composing the linear transformation layer is balanced over the Λ set. By proposition 3, state bytes output from the linear transformation layer are also balanced. If ciphertexts are given at the layer just after passing through following linear transformation layer and key addition layer, by estimating the round key of the key addition layer, decrypting backward until the output of the linear transformation layer and checking whether the state byte is balanced over the Λ set or not.

Type 1 extension We call an extension which is applicable to a case which is extended by adding one extra layer at the end layer, by estimating a round key of the additional final layer as an extension of type 1.

Type 2 extension We call an extension which is applicable to a case which is extended by adding one extra layer at the beginning layer, by estimating a round key of the additional first layer as an extension of type 2.

In the following, generalizing these definitions, we call an extension by adding an extra final layer and estimating a round key of the layer as type 1 extension and an extension by adding an extra initial layer and estimating a round key of the layer as type 2 extension.

6-Layer attack(Type1+Type2) The 6-layer attack is made of the basic attack by applying both Type1 and Type2 exentions.

2.2.2 Improvements by Ferguson et al.[5]

Omitting the first layer key guessing in Type-2 extention(Ferg1) Consider the case where the first key guessing is omitted in the 6-round attack. Then, 32-bit column input $b^{(1)}$, consisting of 4-active bytes, takes all possible states once.

Therefore, $b^{(4)}$ is balance on the Λ set, and the modified attack is applicable to 6-round Rijndael. By omitting the first key guessing, the number of necessary plaintexts does not change, the complexity of attack is reduced by 2^{-8}.

Furthermore, the complexity of attack can be reduced by accompanying the following partial sum method.

Partial sum method In addtion to the idea of omitting the first key guess, Ferguson et al. proposed the partial sum method, which decreases the complexity of attack by efficiently using counters.

On the 6th layer, we pay attention to one byte of $b^{(4)}$, which is denoted as $b_0^{(4)}$. Then, we write 4 bytes of the last (6th) layer input depending on $b_0^{(4)}$ as $b_j^{(5)}$ $(j = 0, 1, 2, 3)$, respectively. And, we set $c_j (= b_j^{(6)})$ corresponds to $b_j^{(5)}$ $(j = 0, 1, 2, 3)$. Then, $b_0^{(4)}$ can be expressed by 4 bytes of output(ciphertext) and layer keys as follows.

$$b_0^{(4)} = S^{-1}\left(\sum_{j=0}^{3} w_j^{-1}\left[S^{-1}(c_j \oplus k_j^{(6)}) \oplus k_j^{(5)}\right]\right)$$

Here, w_j^{-1} is a component of the inverse matrix for the diffusion matrix. We define the combination function of this component and the inverse of S-box as S_j.

$$S_j(x) = w_j^{-1} S^{-1}(x)$$

As 4-byte value $k_j^{(5)}$ contains only 1 byte information, we express it as $k_*^{(5)}$.

$$k_*^{(5)} = \sum_{j=0}^{3} w_j^{-1} k_j^{(5)}$$

In summary, from the information of 5-byte key, we can computer $b_0^{(4)}$.

$$b_0^{(4)} = S^{-1}\left(\sum_{j=0}^{3} S_j(c_j \oplus k_j^{(6)}) \oplus k_*^{(5)}\right)$$

In the partial sum method, calculational complexty is decreased by guessing key step by step.

$$x_l = \sum_{j=0}^{l} S_j(c_j \oplus k_j^{(6)})$$

The key guess consists of 4 phases, and the key guess requires 6 times of 4×2^{48} S-box calculation, which is estimated as $6 \times 4 \times 2^{48} = 24 \times 2^{48} \sim 2^{52}$.

If the calculation of one block encryption as 2^8, the complexty of the key guess is normalized as $2^{52}/2^8 = 2^{44}$.

Using 1-byte-passive set (Ferg2) Suppose the case, where 1 byte of the 2nd layer input $b^{(1)}$ is passive, and where other 15 bytes are active. Then, $b^{(2)}$ consists of $2^{8*(16-2)} (= 2^{112})$ Λ sets, which have only 1-byte passive byte.

In order to let only 1 byte of $b^{(1)}$ be passive, we need to guess 4 bytes of $k^{(0)}$ which are connected to passive $b^{(0)}$.

Table 2: SQUARE attack against Rijndael

Attack	Layer	Plaintext	Complexity	Partial sum
Basic	4	2^9	2^9	No
Type1	5	2^{11}	2^{40}	No
Type2	5	2^{32}	2^{40}	No
Type1+Type2	6	2^{32}	2^{72}	No
Type1×2	6	2^{13}	2^{168}	No
Type2×2	6	2^{128}	2^{168}	No
Type1×2+Type2	7	2^{32}	2^{200}	No
Type1+Type2×2	7	2^{128}	2^{200}	No
Ferg1	6	2^{35}	$2^{44}/2^{64}$	Yes/No
Ferg1+Type1	7	2^{37}	2^{175}	Yes
Ferg2	7	2^{128}	2^{120}	Yes

2.2.3 SQUARE attack against Hierocrypt–L1

Layer numbering Similarly to the case of Rijndael, the index for the i-th layer, which start from the byte substitution through layer-key addition, is given as i for Hierocrypt–L1. The index for the first key addition is 0.

State byte The definitions for state byte, layer key, Λ set, active byte and passive byte are the same as those for Rijndael.

The following propositions are satisfied for Hierocrypt–L1.

Proposition 6 *If i is even, the diffusion layer maps the Λ set of $b^{(i)}$ with only one active byte to a Λ set of $b^{(i+1)}$ with 4 active bytes.*

Proposition 7 *In general, the Λ set of $b^{(i)}$ with only one active byte is mapped to a set of $b^{(i+2)}$ which is balanced but not a Λ set.*

Proposition 8 *If i is even and a set of $b^{(i)}$ makes a Λ set of 2^{32} elements with 4 active bytes belonging the same higher-level S-box, then, $b^{(i+1)}$ makes a Λ with the same property, and $b^{(i+2)}$ and $b^{(i+3)}$ make Λ sets where all bytes are active. $b^{(i+4)}$ makes a balanced set, but not a Λ set.*

Proposition 9 *If i is odd and a set of $b^{(i)}$ makes a Λ set of 2^{32} elements with 4 active bytes belonging the same higher-level S-box, $b^{(i+1)}$ and $b^{(i+2)}$ make Λ sets where all bytes are active. $b^{(i+3)}$ makes a balanced set, but not a Λ set.*

Basic attack Similarly to the case of Rijndael, if the plaintext makes a Λ set with one active byte, **Proposition 7** assures that $b^{(3)}$ is not balanced in general. There for basic attack is feasible up to 3 layers.

Type1 extension A type 1 extension of the basic attack is applied to a reduced version of 4 S-box layers. The total amount of the estimated keys is $(1+4)\times 8 = 40$ bits. In order to apply type 1 extension twice, the total amount of estimated keys is at least 1+4+8 bytes (=104 bits).

Type2 extension A type 2 extension for the basic attack is applied to a reduced version of 4 S-box layers. Because additional 8 bytes of keys in the first key addition must be estimated, the total amount of the estimated keys is $(1 + 4) \times 8 = 40$ bits.

A double of type 2 extensions for the basic attack is applied to a reduced version of 5 S-box layers. Because the total amount of the estimated keys is not less than $(1 + 4 + 8) \times 8 = 104$ bits.

5-layer (2.5-round) attack(Type1+Type2) A combination of a type 1 extension and a type 2 extension for the basic attack 1 is applied to a reduced version of 5 S-box layers. The total amount of the estimated keys is at least $(4 + 1 + 8) \times 8 = 104$ bits. The required number of Λ sets are constructed by chosing from 2^{128} known plaintexts.

Omitting the first layer key guess(Ferg1) The 1st improvement by Ferguson et al., using a set of 2^{32} plaintexts with only 4 active bytes and omitting the first layer key guess, is effective for Hierocrypt–L1. As $b^{(4)}$ is balanced over Λ set, **Property 8** assures that the improved attack is applicable up to 6 S-box layers.

By combining Type1 extenions, the attack is applicable up to 7 S-box layers.

Using the set with 1 passive byte(Ferg2) The improved attack, where using a set of $2^{8\times 7}$ plaintexts where only 1 byte of $b^{(1)}$ is passive, is not so efficient as for Rijndael. $b^{(i)}$ is balanced up to $i = 4$. Therefore, this improved attack is applicable up to 6 S-box layers. The amount of key estimation is $(4+1+4)\times 8 = 72$ bits.

Table 3: SQUARE attacks against Hierocrypt–L1

Attack	Layer	Plaintext	Complexity	Partial sum
Basic	3	2^9	2^9	No
Type1	4	2^{11}	2^{40}	No
Type2	4	2^{32}	2^{40}	No
Type1+Type2	5	2^{32}	2^{72}	No
Type1×2	5	2^{13}	2^{104}	No
Type2×2	5	2^{64}	2^{104}	No
Ferg1	6	2^{35}	$2^{44}/2^{64}$	Yes/No
Ferg1+Type1	7	2^{37}	$2^{110}/2^{132}$	Yes/No
Ferg2	6	2^{64}	$2^{86}/2^{123}$	Yes/No

2.3 truncated differential attack

Differentials with only one bit difference are regarded as different in the differential cryptanalysis. That is, there are 2^N distinct differentials when the block is N-bit length.

To the contrary, differentials are distinguished by word-wise zero-nonzero pattern in the truncated differential attack, where the word size is frequently that of the S-box. Therefore, when the S-box size is 8-bit, differentials are classified into $2^{N/8}$ patterns in the truncated differential.

As The truncated differential is invariant for the (bijective) S-box map and the bit-wise key addition, the encryption is characterized only by the truncated differential transition probabilities for the diffusion layers. And the transition probability for multiple rounds is given as the product sum of those for diffusion layers. Therefore, the truncated differential attack is considered to be effective if all the functions

in encryption are carried out by word-wise operations. As Hierocrypt–L1 consists only of byte-wise operations, its security against the truncated differential attack is indispensable.

2.3.1 Preparation

The byte-wise truncated differential is defined, as all functions of Hierocrypt–L1 encryption are done only by byte-wise operations. The truncated differential for $8m$-bit data differential, $\Delta X_{(8m)}$ is defined by $\chi\left(\Delta X_{(8m)}\right)$ as follows.

$$\chi\left(\Delta X_{(8m)}\right) = \delta(\Delta x_{1(8)}) \| \delta(\Delta x_{2(8)}) \| \cdots \| \delta(\Delta x_{m(8)}) \ ,$$

$$\delta\left(\Delta x_{(8)}\right) = \begin{cases} 1 \ , & \text{for } \Delta x_{(8)} \neq 0 \ , \\ 0 \ , & \text{for } \Delta x_{(8)} = 0 \ . \end{cases}$$

Let $\Pr\left(\chi\left(\Delta X\right) \rightarrow \chi\left(\Delta Y\right)\right)$ be the truncated differential probability for the transition $\chi\left(\Delta X\right) \rightarrow \chi\left(\Delta Y\right)$. Let $\eta\left(X_{(32)}\right)$ be a truncated Hamming differential for a 32-bit data $X_{(32)}$, which is regarded as the Hamming weight of truncated differential $\chi\left(\Delta X_{(32)}\right)$.

$$\eta\left(\Delta X_{(32)}\right) = \sum_{i=1}^{4} \delta\left(\Delta x_{i(8)}\right) \ .$$

The truncated Hamming differential probability is defined as follows.

$$\Pr\left(\eta\left(\Delta X_{(32)}\right) \rightarrow \eta\left(\Delta Y_{(32)}\right)\right) = \max_{\substack{\chi\left(\Delta X^{\emptyset}_{(32)}\right),\ \eta\left(\Delta X^{\emptyset}_{(32)}\right)=\eta\left(\Delta X_{(32)}\right), \\ \chi\left(\Delta Y^{\emptyset}_{(32)}\right),\ \eta\left(\Delta Y^{\emptyset}_{(32)}\right)=\eta\left(\Delta Y_{(32)}\right)}} \Pr\left(\chi\left(\Delta X'_{(32)}\right) \rightarrow \chi\left(\Delta Y'_{(32)}\right)\right) \ .$$

The truncated Hamming differential and the truncated Hamming differential probability can be naturally generalized for a $32m$-bit data as follows.

$$\eta\left(\Delta X_{(32m)}\right) = \sum_{i=1}^{m} \eta\left(\Delta X_{i(32)}\right) 5^{m-1-i} \ ,$$

$$\Pr\left(\eta\left(\Delta X_{(32m)}\right) \rightarrow \eta\left(\Delta Y_{(32m)}\right)\right) = \prod_{i=1}^{m} \Pr\left(\eta\left(\Delta X_{i(32)}\right) \rightarrow \eta\left(\Delta Y_{i(32)}\right)\right) \ .$$

The truncated Hamming differential probability is equal to the truncated differential probability for the mds_{L}-function and the MDS_{L}-function.

$$\Pr\left(\chi\left(\Delta X_{(32)}\right) \rightarrow \chi\left(\Delta Y_{(32)}\right)\right) = \Pr\left(\eta\left(\Delta X_{(32)}\right) \rightarrow \eta\left(\Delta Y_{(32)}\right)\right) \ ,$$

$$\Pr\left(\chi\left(\Delta X_{(128)}\right) \rightarrow \chi\left(\Delta Y_{(128)}\right)\right) = \Pr\left(\eta\left(\Delta X_{(128)}\right) \rightarrow \eta\left(\Delta Y_{(128)}\right)\right) \ .$$

These equations make the security evaluation of Hierocrypt–L1 against truncated differential much simpler.

2.3.2 Properties of the components

[**S-box**]
The S-box is required to be a random bijective function. The S-box of Hierocrypt–L1 has the theoretically minimum differential and linear probabilities, its algebraic order is seven, and the term number in polynomial expression is sufficiently large. Therefore, the S-box can be regarded as a random bijective function.
[mds_{L}**-function**]

Table 4: truncated Hamming differential probabilities for mds_{L}-function (power of 2 approximation)

$\eta(\Delta X_{(32)})$ \ $\eta(\Delta Y_{(32)})$	0	1	2	3	4
0	1	0	0	0	0
1	0	0	0	0	1
2	0	0	0	2^{-8}	1
3	0	0	2^{-16}	2^{-8}	1
4	0	2^{-24}	2^{-16}	2^{-8}	1

The mds_{L}-function is an MDS map for four parallel 8-bit words. This property uniquely determines the truncated differential probability of mds_{L}-function, and leads to the fact that truncated Hamming differential is equal to truncated differential [15]. Figure 4 shows approximate values (powers of 2) for the truncated differential probabilities.

[MDS_{H}**-function**]

The MDS_{H}-function consists only of byte-wise exclusive or's. Approximate values (powers of 2) of truncated differential are obtained by Matsui's algorithm [17].

2.3.3 Evaluation for multiple rounds

MDS_{H}-functions and the MDS_{L}-functions are put in alternate layers in Hierocrypt–L1 except for the S-boxes and the key additions. As previously stated, the truncated differential probability is equal to the truncated Hamming differential probability for MDS_{L}-function. Thus, when the both ends of sequence are MDS_{L}-functions (LHL$\cdots$HL) the maximum characteristic truncated differential probability can be derived only by the truncated Hamming differential probabilities of MDS_{L}-function and MDS_{H}-function.

The use of truncated Hamming differential probabilities reduces the size of MDS_{H} transition table. The size of transition probability table is about 2^{16} for truncated differential probability. On the contrary, the table size for truncated Hamming differential probabilities is much smaller and about $5^4 \simeq 2^{9.29}$.

The following is the process of analysis.

1. Make the truncated Hamming differential probability table for mds_{L}-function
2. Make the truncated Hamming differential probability table for MDS_{H}-function
3. Make the truncated Hamming differential probability table for LH (MDS_{H}-function after MDS_{L}-function)
4. Make truncated Hamming differential probability for t times of (LH)
5. Make $(t+1)$-round truncated Hamming differential probability table by multiplying the preceding result and L
6. Fix the round number where the truncated Hamming differential probability table is the same as that for the random function.

We confirm that the truncated differential characteristic probability table for three consecutive rounds (LHLHL) is the same as that for random function. Therefore, 5-round Hierocrypt–L1 is considered to be sufficiently secure against the two-round attack of truncated differential cryptanalysis.

2.4 higher-order differential attack

The higher-order differential attack is an algebraic attack, where some extended key bits are obtained by solving the equation, which is derived by the following properties for the Boule function whose algebraic order is d [11].

- All $(d+1)$-th order differentials are 0
- all d-th order differentials are constants.

The security against the higher-order differential attack is assured by showing that there is no efficient set of plaintexts. But, it is difficult to assure analytically that the condition is satisfied, heuristic conditions are applied in designing to reduce the applicability. The following are such fundamental conditions.

- the analytical property of the S-boxes is optimized, which are the only nonlinear components
- the property of diffusion layers is optimized, which are linear

As for Hierocrypt–L1, the algebraic order of S-box is seven, which is the highest value for 8-bit surjective functions, and bit permutation is inserted to increase the complexity of algebraic structure. Furthermore, the differential layer is an MDS map where the data is sufficiently mixed, and is determined such that the combined function with the S-box has the number of terms in the polynomial expression is maximum. Therefore, we consider that the applicability of the higher-order differential attack is sufficiently low.

2.5 Interpolation attack

The interpolation attack is an attack where encryption function is guessed by determining all coefficients of polynomial expression for the encryption function. The security against the interpolation attack is estimated by the number of terms in the polynomial expression for the encryption function. This attack is effective when the number of terms is sufficiently small for the polynomial expression over $\mathrm{GF}(2^8)$. As forHierocrypt–L1, a power operation over $\mathrm{GF}(2^8)$ is used to make the S-box. But, the bit permutation at the input side is also used, thus the number of terms in polynomial expression is sufficiently large. Thus, a simple application of the interpolation attack is considered to be ineffective for Hierocrypt–L1.

2.6 Impossible differential attack

Impossible differential attack is an attack where estimated extended key-patterns are narrowed down by discarding ones which lead to intermediate impossible differential patterns.

The number of impossible differential patterns is tends to decrease rapidly, when the connections of diffusion layers are dense.

The most important difference between Hierocrypt–L1 and Rijndael is the higher-level diffusion layer. For Rijndael, one byte differential spreads to all bytes after two diffusion layers, but through only one path. Therefore, when one byte is active and the others are not active, all bytes are active for the layer after two diffusion layers.

To the contrary, the diffusion layer MDS_{H} of Hierocrypt–L1 is designed such that one byte is connected with all bytes on the layer after two diffusion layers through more than one paths. As all bytes there can take zero differential, and there exist many possible differential patterns, Hierocrypt–L1 is considered to be much securer than Rijndael against the impossible differential attack. Therefore, Hierocrypt–L1 is considered to be secure against the impossible differential attack, as the impossible differential attack can not attack full-round Rijndael.

2.7 Non-surjective attack

The non-surjective attack uses patterns which can be realized because of the non-surjective property of components in encryption. As all components of Hierocrypt–L1 is bijective (i.e. surjective), the attack is not applicable to Hierocrypt–L1.

2.8 Mod n attack

This attack uses the bias of possible bit patterns arising from the local non-surjectivity. As all components of Hierocrypt–L1 are bijective (of course, surjective), the attack is not applicable to Hierocrypt–L1.

2.9 χ^2 attack

In the χ^2 attack, the transition probability distribution bias between certain input and output bit-sets is searched both theoretically and numerically at first. Then the feasibility of estimated key is determined by the χ^2-test for the bias. As Hierocrypt–L1 does not use operations with a high bit correlation bias such as multiplication used in MARS, it is considered to be secure against the χ^2 attack.

2.10 Attack papers against Hierocrypt–L1

2.10.1 SQUARE attack

Barreto et al. proposed an improved Square attack against Hierocrypt–3and Hierocrypt–L1in [21]. They showed that Hierocrypt–L1is vulnerable up to 7-layer(3.5-round).

Their result was published at FSE 2001 conferernce in April of 2001. But, we had already published quite the same result in January of the same year[20].

As the number of layers is 12(6-round), we consider that Hierocrypt–L1is sufficiently secure against the SQUARE attack.

2.10.2 Impossible Differential attack

Cheon et al. found a 2-round efficient impossible differential against Hierocrypt–L1, and showed that the attack is applicable up to 3-round. The complexity of the computation is estimated as; 2^{55} encryptions for 2^{71} known plaintexts.

But, Hierocrypt–L1is 6-round, therefore we consider that the attack is not efficient for the full spec Hierocrypt–L1.

2.10.3 Key Schedule

The key scheduling part of Hierocrypt–L1consists of the intermediate key generation part of 256-bit width, and the round key generation part which makes round keys from the intermediate keys. As the intermediate key generation part is round-trip type, a pair of intermediate keys which locate at the symmetric rounds for the turning point. Therefore, the round key generation part should be designed, so that there are no simple relation between the round key bits.

Furuya et al. anlyzed Hierocrypt–3's key scheduling part, and found many linear relations between the round key bits[23].

But, these relations have not be used for a new attack against Hierocrypt–L1. As the data randomizing part of Hierocrypt–L1is sufficiently secure, the relations do not seem to be a real threat to Hierocrypt–L1.

3 Software Implementation Evaluation

In this section, we discuss the software implementation evaluation of Hierocrypt–3. More specifically, we describe the following items: encryption speeds, memory requirement (e.g. code size, work area), optimization, language and platforms for evaluation.

3.1 Evaluation platform and implementation environment

Tables 5 and 6 show the implementation environments. The 3 kinds in Table 5 were used in the evaluation by CRYPTREC in the last fiscal year[10]. For efficient implementation techniques for Hierocrypt–L1, refer to [6, 22, 9].

3.2 Speed evaluation method

At first, we briefly describe how to evaluate the speed. By using the Time Stamp Counter (TSC) embedded in Pentium III, the number of cycles required for the following processes are measured: key scheduling,

Table 5: Platform 1: Client, High performance and Sever environments

Environment	Client	High performance	Server
CPU	Pentium III (650 MHz)	Alpha 21264(463 Hz)	Ultra SPARC(400 MHz)
OS	Windows 98 SE	Tru 64 UNIX V5.1	Solaris 7
RAM	64MB	512MB	256MB
Compiler	Visual C++ 6.0 SP3	DEC C	Forte C 6
Assembler	MASM 6.14	–	–

Table 6: Platform 2: JAVA, 8-bit, Smart card environments

Environment	JAVA	8-bit	Smart card
CPU	Pentium II (600 MHz)	Z80(5 MHz)	JT6N55(5 MHz)
OS	Windows 2000 SP2	–	?
RAM	192MB	512MB	73MB
Compiler	Sun JDK1.3.1	–	–
Simulator	–	z80pack+patch	z80pack+patch
Assembler	–	PROASM-II ver.3	PROASM-II ver.3

data encryption, and data decryption. In order to remove the ambiguity of measurement, a piece of speed evaluation program source is shown in Figure 1. The term "required CPU cycles/BENCH_COUNT" means the cycle number needed to 1 block operation including the function call, which gives a throughput corresponding to the CPU clock.

3.3 Speed evaluation

3.3.1 Pentium III

Table 5 shows the Client environment used in the implementation evaluation by CRYPTREC in the last fiscal year [10]. Table 7 shows the least number of cycles in ten trials to carry out 1,000,000 block encryptions of Hierocrypt–L1 in ECB mode for 128-bit key. The upper row shows the speed without key setup. On the other hand, the lowe row shows that with key setup.

Table 8 shows used memory requirements.

Table 7: Pentium III(650MHz) encryption/decryption performance

	Time(cycle)		Throughput(Mbps)	
Key setup	encryption	decryption	encryption	decryption
No	199	204	209.0	203.9
Yes	374	616	111.2	67.5

3.3.2 High End Environment(Alpha 21264)

Table 5 shows the Client environment used in the implementation evaluation by CRYPTREC in the last fiscal year [10]. As the High End environment, a workstation with high performance CPU, Alpha 21264 chip(DEC), was used.

Table 7 shows the least number of cycles in ten trials to carry out 1,000,000 block encryptions of Hierocrypt–3 in ECB mode for 128-bit key. The upper row shows the speed without key setup. On the other hand, the lowe row shows that with key setup.

Table 8 shows used memory requirements.

```
#define BENCH_COUNT 1000000
#define CPUID __asm __emit 0fh __asm __emit 0a2h
#define RDTSC __asm __emit 0fh __asm __emit 031h
  __asm {
    pushad
    CPUID
    RDTSC
    mov  cycles_high1, edx
    mov  cycles_low1, eax
    popad
  }
  for(i=0; i<BENCH_COUNT; i++)
     function_call(in, out, ekey); /* evaluation target */
  __asm {
    pushad
    CPUID
    RDTSC
    mov  cycles_high2, edx
    mov  cycles_low2, eax
    popad
  }
  temp_cycles1 = ((unsigned __int64)cycles_high1 << 32) | cycles_low1;
  temp_cycles2 = ((unsigned __int64)cycles_high2 << 32) | cycles_low2;
  split = temp_cycle2 - temp_cycle1;
```

Figure 1: Piece of speed evaluation program on Pentium III

Table 8: Pentium III memory usage

Operation	Code size	Work area
Enc/Dec	52982	448

Table 9: Alpha 21264(463MHz) encryption/decryption performance

Key setup	Time(cycle)		Throughput(Mbps)	
	encryption	decryption	encryption	decryption
No	210	210	141.1	141.1
Yes	390	625	76.0	47.4

Table 10: Alpha 21264 memory usage

Operation	Code size	Work area
Enc/Dec	84328	448

3.3.3 Server environment(Ultra SPARC IIi)

Table 5 shows the Client environment used in the implementation evaluation by CRYPTREC in the last fiscal year [10].

As the Server environment, a workstation with Ultra SPARC IIi(Sun Microsystems) was used.

Table 11 shows the least number of cycles in ten trials to carry out 1,000,000 block encryptions of Hierocrypt–3 in ECB mode for 128-bit key. The upper row shows the speed without key setup. On the other hand, the lowe row shows that with key setup.

Table 12 shows used memory requirements.

Table 11: Ultra SPARC IIi(400Mhz) encryption/decryption performance

	Time(cycle)		Throughput(Mbps)	
Key setup	encryption	decryption	encryption	decryption
No	378	500	67.7	51.2
Yes	718	1203	35.7	21.3

Table 12: Ultra SPARC IIi(400Mz) memory usage

Operation	Code size	Work area
Enc/Dec	24496	448

3.3.4 JAVA environment

The performance in JAVA environment was estimated with JDK 1.3.1. To observe the overhead of security interface, we used IAIK-JCE 2.61, which was designed on the specification of JCE 1.2(Sun Microsystems).

The evaluation was done on the note PC DynaBook SS3480 DS60P/1n2L(TOSHIBA) [1].

Table 13 shows the development and evaluation environments. Table 14 shows the performance. The throughput is a normalized for the frequency of 200MHz. Therefore, the estimated value are triple for the Pentium III(600MHz).

Table 13: JAVA Environment

Development		Sun Microsystems JDK1.3.1
Evaluation	Computer	DynaBook SS 3480 DS60P/1N2L
	CPU	Intel SpeedStep Technology Low-voltage Mobile Pentium II (600MHz)
	Memory	192 MByte
	OS	Microsoft Windows 2000 5.00.2195 Service Pack 2

3.3.5 8-bit environment

The performance on Z80, as an 8-bit environment, was estimated by using a simulator. The used simulator is z80sim included in z80pack obatained by the following site.

ftp://ftp.cs.uni-sb.de/pub/others/z80pack.tgz

State number was counted which was required for 1-block encryption/decryption. The implementation includes encryption, decryption and key scheduling. Table 15 shows the result.

[1] http://dynabook.com/pc/catalog/oldpc/ss/ss34t2/spec.htm

Table 14: JAVA(Pentium III, 600MHz) encryption/decryption performance

JCE	Class	Key generation		Throughput	
	size	(key/sec)		Mbps	
	(byte)	encryption	decryption	(byte)	(byte)
Yes	13315	224775	155638	30.69	29.94
No	11422	533219	–	32.05	30.75

Table 15: Z80(5MHz) encryption/decryption performance

Time(state)		ROM	RAM
encryption	decryption	(Byte)	(Byte)
18384	21588	4196	25

3.3.6 Smart card

This evaluation is for a smart card JT6N55[1], where Z80[2] is used as CPU and Z80 assembly language is used for coding.

Evaluation platform and implementation environment Table 17 shows the platform for evaluation of software implementation including the language and the developing environment. The program is stored in a ROM region, and it can not be modified for a smartcard. As ROM, RAM, and EEPROM are needed for the other processes except for encryption, the smallest code which achieves the required speed is needed.

Speed and Memory Evaluation Speed of implementation on JT6N55[1] coded by Z80 assembly language is estimated. As the prescribed state number of usual Z80 architecture is used for the estimating the process state number, 4 states are need for the minimum instruction of Z80. In this implementation, process speed is optimized under the constraint that source code is within 3KB and that on-the-fly key generation is used.

Table 18 shows the state number and memory, which are required to parametrize storage the addresses of areas for a plaintext, a ciphertext, and a key; to call Hierocrypt–L1 subroutine; and to store the ciphertext to the ciphertext storage area.

In this estimation, speed has a priority and a 1,280-byte reference table is used which is the most efficient. Thus, source code is rather large, and it can be decreased.

4 Hardware Implementations

In this section, we show some hardware implementations; high speed one and small area one by ASIC, and hight speed one by FPGA. For efficient implementation techniques, refer to [6, 22, 9].

Table 16: Z80(5MHz) memory usage

Operation	Code	Time	Stack	RAM(byte)				
	(Byte)	(state)	(Byte)	Plaintext	Key	Ciphertext	Work	Sum
Encryption	2,228	18,384	16	8	16	–	1	25
Decryption	3,200	21,588	16	8	16	–	1	25
Enc/Dec	4,196	–	–	–	–	–	–	–

Table 17: Evaluation platform specification

chip	JT6N55	
memory	ROM	48KB
	RAM	1KB
	EEPROM	8KB
language	Z80 assembly language	

Table 18: Speed and Memory on smartcard

	key length	ROM	RAM	encryption	
algorithm	(bits)	(bytes)	(bytes)	(states)	(ms @5MHz)
Hierocrypt-L1	128	26	2,447	19,399	3.88

4.1 ASIC implementation

4.1.1 High Speed Implementation (ASIC-1)

1. Semiconductor Technology
   0.25 μm 3 layer metal CMOS
2. Synthesis
   SYNOPSYS Design Compier 1999.10-3
3. Simulation Condition(Commercial Worst-case)
   1.35V, 70 degrees C(1.5V, 25 degrees C, typical-case)
4. Throughput
   1081Mb/s(9.86ns, 6 clock)
5. Gate Count
   81.2K gates

4.1.2 High Speed Implementation (ASIC-2)

1. Semiconductor Technology
   0.15 μm 3 layer metal CMOS
2. Synthesis
   SYNOPSYS Design Compier 1999.10-3 ?
3. Simulation Condition(Commercial Worst-case)
   1.35V, 70 degrees C(1.5V, 25 degrees C, typical-case)
4. Throughput
   1568Mb/s(6.80ns, 6 clock)
5. Gate Count
   54.9K gates

4.1.3 Small Areal Implementation(ASIC-3)

Small area implementation by sharing SBOX and MDSL.

1. Semiconductor Technology
   0.25 μm 3 layer metal CMOS
2. Synthesis
   SYNOPSYS Design Compier 1999.10-3
3. Simulation Condition(Commercial Worst-case)
   1.35V, 70 degrees C(1.5V, 25 degrees C, typical-case)
4. Throughput
   135.0Mb/s(18.22ns, 26 clock)
5. Gate Count
   9.9K gates

4.2 Implementation using FPGA

4.2.1 High Speed Implementation(FPGA-1)

1. Synthesis
   ALTERA Max+plus II ver. 9.6
2. Throughput
   51.0Mb/s(11.16MHz, 89.6ns, 14 clock)
3. Logic Cells
   11.0K Logic Cells, ALTERA Flex 10K family

4.3 Summary of hardware implementations

The hardware implementations for Hierocrypt–L1is summarized in Table 19.

Table 19: ASIC implementation

implementation	Rule (μm)	Throughput (Mbps)	Area		Critical path (ns)	Latency (clock)
			(Kgate)	(K logic cell)		
ASIC-1	0.25μm	1081	81.2	–	9.86	6
ASIC-2	0.13μm	1568	54.9	–	6.80	6
ASIC-3	0.25μm	135	9.9	–	18.22	26
FPGA-1	–	51.0	–	11.0	89.6	14

5 Conclusion

Hierocrypt–L1is considered to be sufficiently secure against well-known attacks. And Hierocrypt–L1is highly efficient in wide range of platforms, such as custom LSI and middleware.

References

[1] *JT6N55*. http://www.toshiba.co.jp/about/press/2000_02/pr_j1801.htm.

[2] *Z80 Microprocessor Products*. available on http://www.zilog.com/products/z80.html.

[3] E. Biham and A. Shamir. Differential cryptanalysis of des-like cryptosystems. *Journal of Cryptology*, 4(1):3–72, 1991.

[4] J. Daemen, L. R. Knudsen, and V. Rijmen. The block cipher Square. *FSE'97*, LNCS 1267:149–165, 1997.

[5] S.Lucks et al. *Attacking Seven Rounds of Rijndael under 192-bit and 256-bit Keys*, 2000. The third AES Conference.

[6] F.Sano, K.Ohkuma, H.Shimizu, M.Motoyama, and S.Kawamura. Efficient implementation of Hierocrypt. *Proc. of 2nd NESSIE Workshop*, 2001. to appear in LNCS.

[7] H.Muratani, K.Ohkuma, F.Sano, M.Motoyama, and S.Kawamura. Proposition of a 64-bit version of Hierocrypt. *IPSJ SIG Notes*, CSEC11(8):43–48, 2000.

[8] S. Hong, S. Lee, J. Lim, J. Sung, and D. Cheon. Provable security against differential and linear cryptanalysis for the SPN structure. *FSE 2000*, 2001.

[9] H.Shimizu, F.Sano, M.Motoyama, K.Ohkuma, and S.Kawamura. Implementation of SPN block cipher. *IEICE Technical Report*, ISEC 2001-55(2001-09):17–21, 2001.

[10] IT Security Center, Information-technology Promotion Agency. *CRYPTREC Report 2000*, 2001.

[11] L.R. Knudsen and T.A. Berson. Truncated differentials of safer. *FSE'96*, LNCS 1039:15–25, 1996.

[12] K.Ohkuma, H.Muratani, F.Sano, M.Motoyama, and S.Kawamura. Security and performance evaluations for the block ciphers Hierocrypt-3 and Hierocrypt-l1. *IEICE Technical Report*, ISEC 2000-71(2000-09):71–100, 2000.

[13] K.Ohkuma, H.Muratani, F.Sano, and S.Kawamura. Specification and assessment of the cipher Hierocrypt. *IEICE Technical Report*, ISEC2000-7(2000-05):77–104, 2000.

[14] K.Ohkuma, H.Shimizu, F.Sano, and S.Kawamura. Security of Hierocrypt and Rijndael against the differential and linear cryptanalysis. *Proc. of 2nd NESSIE Workshop*, 2001. to appear in LNCS.

[15] K. Uehara S. Kubota M. Sugita, K. Kobara and H. Imai. *Relationships among Differerential, Truncated Differential, Impossible Differential Cryptanalyses against Word-Oriented Block Ciphers like Rijndael, E2.*

[16] M. Matsui. Linear cryptanalysis method for des cipher. *Eurocrypt'93*, 765:386–397, 1994.

[17] M. Matsui. Cryptanalysis of a reduced version of the block cipher E2. *FSE'99*, LNCS 1636, 1999.

[18] K. Ohkuma, H. Muratani, F. Sano, and S. Kawamura. On the recursive SPN structure. *IEICE Technical Report*, ISEC 99-141(2000-03):99–104, 2000.

[19] K. Ohkuma, H. Muratani, F. Sano, and S. Kawamura. The block cipher Hierocrypt. *SAC 2000*, LNCS 2012:72–88, 2001.

[20] K. Ohkuma, F. Sano, H. Muratani, M. Motoyama, and S. Kawamura. On security of block ciphers Hierocrypt-3 and Hierocrypt-l1. *SCIS 2001*, 11A-4, 2001.

[21] P.Barreto, V.Rijmen, J.Nakahara Jr., B.Preneel, J.Vanderwalle, and H.Kim. Improved Square attacks against reduced-round Hierocrypt. *proc. of FSE 2001*, pages 173–182, 2001.

[22] F. Sano, H. Muratani, K. Ohkuma, S. Kawamura, and M. Motoyama. Implementation of Hierocrypt. *SCIS 2001*, 13A-2, 2001.

[23] S.Furuya and V.Rijmen. Observations on Hierocrypt-3/l1 key-scheduling algorithms. *Proc. of 2nd NESSIE Workshop*, 2001. to appear in LNCS.