自己評価書:Hierocrypt–L1

株式会社 東芝

平成 13 年 10 月 1 日

目 次

1 はじめに **3**

2 安全性 **3**

2.1 差分解読法および線形解読法 3
2.1.1 差分確率と線形確率の定義 3
2.1.2 S-box の特性 3
2.1.3 活性 S-box 数 4
2.1.4 証明可能安全性に基づく評価 4
2.2 SQUARE 攻撃 5
2.2.1 Rijndael に対する基本的攻撃 6
2.2.2 Ferguson らによる改良 [7] 7
2.2.3 Hierocrypt–L1 に対する SQUARE 攻撃 9
2.3 truncated 差分 11
2.3.1 準備 11
2.3.2 構成要素の特性の検討 12
2.3.3 多段に対する評価 12
2.4 高階差分攻撃 13
2.5 補間攻撃 13
2.6 Impossible Differential 攻撃 14
2.7 Non-surjective 攻撃 14
2.8 Mod n 攻撃 14
2.9 χ^2 攻撃 14
2.10 Hierocrypt–L1 に対する攻撃論文 14
2.10.1 SQUARE 攻撃 14
2.10.2 Impossible Differential 攻撃 15
2.10.3 鍵スケジュール 15

3 ソフトウェアでの実装評価 **15**

3.1 実装環境 15
3.2 計測法について 16
3.3 計測結果 16

3.3.1 Client 環境 (Pentium III) 16
3.3.2 High End 環境 (Alpha 21264) 18
3.3.3 Server 環境 (Ultra SPARC IIi) 18
3.3.4 JAVA 環境 19
3.3.5 8-bit 環境 19
3.3.6 スマートカード 20

4 ハードウェアでの実装評価 21
4.1 ASIC による実装 21
4.1.1 速度優先設計 (ASIC-1) 21
4.1.2 速度優先設計 (ASIC-2) 21
4.1.3 面積優先設計 (ASIC-3) 22
4.2 FPGA による実装 22
4.2.1 速度優先設計 (FPGA-1) 22
4.3 ハードウェア実装性能のまとめ 22

5 おわりに 23

1 はじめに

当評価書では、提案者による安全性および性能に関する自己評価結果を報告する。

2 安全性

2.1 差分解読法および線形解読法

差分解読法は Biham と Shamir によって [3]、線形解読法は松井によって [13] 各々開発された共通鍵ブロック暗号に対する汎用の攻撃法である。差分 / 線形解読に対する安全性は、鍵の推定が十分大きな確率で成功するのに必要な平文と暗号文の組数で評価される。この組数は、鍵ビットの直接探索を行なう両端の数段 (通常、平文および暗号文側合わせて 2∼3 段) を除いた、中間段の最大差分確率 / 線形確率 (伴に正規化値) の逆数と同じオーダになることが知られている。実際に最大差分 / 線形確率を厳密に評価することは困難なので、差分 / 線形経路に関する総和を取らない最大差分特性確率および最大線形特性確率を利用して近似的な評価とすることが多い。本提案の Hierocrypt–L1 は入れ子型 SPN 構造と拡散層として MDS 行列を利用しているので、活性 S-box 数の下限が容易に求まり、それを利用して最大差分 / 線形特性確率の上限が評価できる。さらには、最近発見された SPN 暗号に対する証明可能性安全性の理論が Hierocrypt–L1 にも階層的に適用できることが分かっている [9, 11]。適用の結果、Hierocrypt–L1 の 2 段の最大差分 / 線形確率が厳密に求められる。以上の評価を評価することで、鍵長ごとの適切な段数が評価できるので、その結果を以下で報告する。

2.1.1 差分確率と線形確率の定義

関数 f に対する最大差分確率を次式で定義する。

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

同様に f に対する正規化された最大線形確率を次式で定義する。

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

最大線形確率の定義が通常と異なり正規化値としている。この定義を用いることで、鍵推定に必要な平文 / 暗号文組数が最大線形確率の逆数のオーダとなり、差分解読とほぼ並行した議論が可能となる。

2.1.2 S-box の特性

Hierocrypt–L1 の S-box 変換は、8-bit 入力に対し次の 3 段階の変換を順に行なうのと同値である。

(a) ビット置換

(b) GF(2^8) 上でのべき乗演算 x^{247}

(c) GF(2^8) 上での排他的論理和 $x \oplus 0\text{x}7$

変換 (a) と (c) の変換は差分 / 線形確率の分布を変えないので、S-box の最大差分 / 線形確率は (b) と一致し、ともに 2^{-6} となる。

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

2.1.3 活性 S-box 数

まず差分に対して考える。拡大 S-box XS が活性ならば、つまり、入力 32 ビットのうち差分が 0 でないものが 1 ビットでもあれば、中の 4 個の S-box のうち 3 個は活性 S-box である。また、連続する 2 段を考えたとき、入力ビットに 0 でないものが有れば、最低でも 3 個の拡大 S-box XS が活性となる。文献 [22, 15] の Proposition. 2 から連続する 2 段に含まれる活性 S-box の下限は、$5 \times 3 = 15$ となる。よって、連続する 2 段の最大差分特性確率 $DP^{2\mathrm{R}}$ は次式の不等式を満足する。

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

まったく同様に、入力差分ビット → 出力マスク・ビットの置換えで、線形解読に対し同様の評価が行なえる。すなわち、連続する 2 段に含まれる活性 S-box の下限は 15 個であり。連続する 2 段の最大線形特性確率は 2^{-90} を超えない。

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

実際の差分 / 線形解読法では両端の段は鍵推定を行なうので、十分な特性確率を満足する段数に 2 段または 3 段を足して段数とする。Hierocrypt–L1 の 1 段は通常の 2 段分に相当するので、 2 段 (通常の 4 段分) を足せば、差分 / 線形解読に対しては十分な security margin を持つと言える。

よって、鍵長 128 ビットに対しては、$2 + 2 = 4$ で、 4 段有れば十分である。

2.1.4 証明可能安全性に基づく評価

FSE 2000 において、高麗大学の Hong らが、SPN 型暗号で証明可能安全性を持つものが構成できることを示した [9]。

彼らが証明したのは次の定理である。

定理 1 n 個の並列 S-box 2 層を拡散層 (P) で直列に結合した SPS 構造を考え、次を仮定する。

- 拡大鍵に独立で偏りが無い
- 拡散層のブランチ数が $n + 1$
- S-box の最大差分確率が dp (または最大線形確率が lp)

このとき、SPS 構造の最大差分 (線形) 確率は、dp^n(または、lp^n) を超えない。¶

ここで、Hierocrypt–L1 の 2 段を考える。 2 段目の MDS_{H} は線形変換であり、差分 / 特性確率の分布を変えない。よって、Hierocrypt–L1 2 段の最大差分 / 線形確率は、上位レベルでの SPS の最大差分 / 線形確率と等しい。上位 SPS のブランチ数は 3 であるので、上位 S-box(XS) の最大差分 / 線形確率を dp^{XS} / lp^{XS} とすると、 2 段 Hierocrypt–L1 の各々は $\left(dp^{XS}\right)^2$ / $\left(lp^{XS}\right)^2$ を超えない。上位 S-boxXS は下位レベルの SPS 構造から成り、ブランチ数は 5 である。よって、XS の最大差分確率は次の値を超えない。

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

同様に最大線形確率に関し、次式が成立する。

$$\left(lp^{XS}\right) \leq 2^{-24}$$

2 段 Hierocrypt–L1(2 段目の上位拡散層 MDS_{H} を除いて) に対しても同様、XS が 4 個並列でブランチ数が 5 であり、2 段 Hierocrypt–L1 の最大差分 / 線形確率は、$\left(2^{-24}\right)^2 = 2^{-48}$ を超えない。まとめると、鍵スケジュールに欠陥が無ければ、2 段 Hierocrypt–L1 の差分 / 線形解読には、最低でも 2^{48} 個程度の平文 / 暗号分組が必要であることが数学的に保証できることを意味する。

しかし、2^{-48} は、全数探索より効率の良い解読の存在を否定しきれていない。そこで、数学的厳密さは落ちるものの、4 段 Hierocrypt–L1 の最大確率を評価してみる。

既に 2 段の最大確率が 2^{-48} であることは分かっている。1 段追加して 3 段としてとき、最悪でも追加した段の上位 S-box(XS)1 個は活性であるので、その分の最大確率は 2^{-12} と評価できる。よって、3 段の差分確率は、$2^{-48} \times 2^{-12} = 2^{-60}$ を超えない。つまり、3 段では差分 / 線形解読法の適用が可能か不可能化の境界付近であることが分かり、全数探索より効率的な解読の存在は否定できない。ここで、2 段と残りの 1 段の結合部における総和を取らず、確率の積が最大となる 1 個の値 (差分値またはマスク値) で置き換える近似を行なっている。

4 段での近似確率は、厳密に得られた 2 段の評価を自乗したものであり、$2^{-48} \times 2^{-48} = 2^{-96}$ を超えない。2^{-64} より小さいので、4 段では有効な確率を持つ差分 / 線形パターンは存在しないことがわかる。ここで、2 段組みの境界部分の総和を省略した点が、近似となっている。

つまり、3 段では差分 / 線形解読法の適用が可能か不可能化の境界付近であることが分かり、全数探索より効率的な解読の存在は否定できない。

以上の考察により、4 段 (中間) + 2 段 (両端)=6 段を段数と設定した。

表 1: 証明可能安全性に基づく最小段数の評価

鍵長	望ましい最小段数 R	$R-2$	最大確率の近似評価
–	4 段	2 段	2^{-48}
–	5 段	3 段	2^{-60}
128 ビット	6 段	4 段	2^{-96}

ここでの対差分 / 線形解読強度評価は、2 段での証明可能性安全性に基づくものであり近似精度は通常の最大特性確率よりも高い。2 段では厳密、3 段および 4 段でも近似は 1ヶ所の中間状態に関する総和を省略しただけであった。通常の活性 S-box 数による最大特性確率の評価は汎用性が高く実用的であるが、安全サイドに立つならば、ここで行なったような近似精度の高い評価に基づいて段数を設定することが望ましいであろう。

2.2 SQUARE 攻撃

SQUARE 攻撃は、SQUARE 暗号や Rijndael など類似の SPN 構造を持つ暗号に対する選択平文攻撃である。基本攻撃と鍵推定による適用段数の拡張が提案されている [5]。

Hierocrypt–L1 では、SUQARE 暗号とは異なる段数の数え方をしている。混乱を避けるため、段数に代えて、層数を定義する。層数とは、S-box 層の数とする。つまり、Hierocrypt–L1 では、2 層が 1 段に相当し、SQUARE 暗号では、1 層が 1 段に相当する。

SQUARE 攻撃は、鍵スケジュール部の性質を考慮しなければ、128 ビット鍵の SQUARE 暗号や Rijndael 暗号に対しては 7 層まで [5]、192 ビット鍵と 256 ビット鍵の Rijndael 暗号に対しては 8 層まで [7] 有効であることが分かっている。これに対し、ブロックサイズ 64 ビット, 鍵長 128 ビットの Hierocrypt–L1 では、7 層 (3.5 段) までしか有効ではないことを確認した。これに対し Hierocrypt–L1 の現在の解読可能段数は、7 層 (3.5 段) までである [21]。

昨年度の提案での自己評価では Hierocrypt–L1 の解読可能段数は 5 層までとしたが [23]、Ferguson らの改良を適用することによって解読可能段数を伸ばした。なお、我々と独立に Barreto らが全く同様の結果を導出している [16] (2.10.1 を参照)。

Hierocrypt–L1 は 12 層 (6 段) なので、現状では SQUARE 攻撃に対して十分に強いと考えられる。

2.2.1 Rijndael に対する基本的攻撃

本小節では Rijndael に対する SQUARE 攻撃の基本的攻撃について述べる。最初に基本事項を定義し、次に基本命題を示し、基本的な攻撃法を説明する。

層の番号付け　バイト代入 (S-box) からレイヤ鍵加算[1]に至る層の各々に、平文に近い方から 1,2,3... と順番を振る。

状態ブロックとレイヤ鍵　第 i 層への入力を状態ブロック $b^{(i-1)}$, その直前のレイヤ鍵を $k^{(i-1)}$ とする。つまり、$b^{(i-1)}$ は第 i 層のバイト代入層の入力になる。

状態バイト　Rijndael のバイト代入層は 16 個の 8 ビット S-box から構成され、その各々入力を状態バイトと呼ぶ。

Λ 集合　Daemen 等にならい、Λ 集合を次のように定義する。

1. Λ 集合は、256 個の状態ブロックから成る。
2. Λ 集合において、特定の状態バイトは 256 個の全状態を取るか、または、固定値である。

活性バイトと不活性バイト　Λ 集合の要素をある状態バイトの状態に制限するとき、Λ 集合で、全状態を廻るバイトを活性バイトと呼び、状態固定のバイトを不活性バイトと呼ぶ。

Λ 集合上でバランス　Λ 集合を入力したとき、(中間) 出力の排他的論理和の全ビットが 0 のとき、Λ 集合上でバランスすると呼ぶ。

Λ 和集合　複数の Λ 集合の和集合を Λ 和集合と呼ぶ。各 Λ 集合上でバランスしているならば、Λ 和集合上でもバランスする。

命題 1 バイトごとに全単射な変換は Λ 集合を Λ 集合に変換する。

命題 2 Λ 集合は線形変換 *(*ビット単位の排他的論理和*)* によってバランスした状態に変換される。

[1] 通常はラウンド鍵と呼ぶが、ここでは層数を使用するのでレイヤ鍵とした

命題 **3** $b^{(i)}$ が *1* バイトのみ活性の Λ 集合を廻るとき、$b^{(i+1)}$ は *1* 列 *4* バイトのみ活性な Λ 集合を成す。

命題 **4** $b^{(i)}$ が *1* バイトのみ活性の Λ 集合を廻るとき、$b^{(i+2)}$ は全 *16* バイトが活性な Λ 集合を成す。

基本攻撃　Rijndael に対する最も基本的な攻撃は、平文に 1 バイトのみ活性な Λ 集合を利用した 4 層縮小モデルに対する攻撃法である。平文の条件と性質 **2** と性質 **4** から、$b^{(3)}$ が Λ 上でバランスすることが分かる。最終層の層関数を $\rho^{(4)}$), 暗号文を c, 最終層鍵を $k^{(5)}$ とすると、$b^{(3)} = \rho^{(4)-1}(c, k^{(5)})$ によって 3 層目入力 $b^{(3)}$ は逆算でき、そのバランスの是非でレイヤ鍵 $k^{(4)}$ の判定が出来る。$b^{(3)}$ のバランスはバイトごとに独立に実行できるので、1 回に判定する鍵ビット長は 8 ビットで、逆算する S-box の個数は 1 個である。

ただし、このチェックでは正しくない擬似鍵も通過する可能性がある。擬似鍵のほとんどは、不活性部分が異なる複数の平文セットに対してチェックすることで排除できる。m 個の平文セットによるチェック全部に通過する擬似鍵の個数の期待値は、$2^8 * (1/2)^{8m} = 2^{8(1-m)}$ なので、基本攻撃では、$m = 2$ とすることで擬似鍵をほぼ排除できる表 2。

表 2 の複雑度は、鍵推定に必要な計算量を暗号化 1 回分の計算量で規格化した値の概算値である。Rijndael では暗号化 1 回分の計算量は 2^8 回のメモリ参照と同じと仮定して評価することが多く、本報でもそれに従う [6, 7]。

Type1 の拡張　暗号文側の鍵推定層を 1 層伸ばす拡張を Type1 の拡張と呼ぶ。推定する鍵ビット数は 1 列 4 バイト分の 16 ビット増え、攻撃できる層数は 1 層増える。

Type2 の拡張　平文側の鍵推定層を 1 層伸ばす拡張を Type2 の拡張と呼ぶ。その場合、$b^{(1)}$ が 1 バイトのみ活性の Λ 集合にするため、入力する平文集合にはその活性バイトに接続する 4 バイトが活性な Λ 集合 (要素数 2^{32}) を用意する。対応する 4 バイト (32 ビット) 鍵の推定ごとに、1 バイトのみ活性の条件を満たす 2^8 個の状態集合を構成する。推定する鍵ビット数は 4 バイト分の 32 ビット増え、攻撃できる層数は 1 層増える。

6 層攻撃 (Type1+Type2)　基本攻撃に Type1 および 2 の両方の拡張を適用したものが 6 層攻撃である。

2.2.2　Ferguson らによる改良 [7]

Type-2 拡張での初層鍵推定の省略 (Ferg1)　オリジナルの 6 層攻撃において、初層鍵 $b^{(0)}$ の推定を省略してみる。すると、第 2 層入力 $b^{(1)}$ では 4 個の活性バイトから成る 32 ビット長のコラムが、2^{32} 個の全状態を 1 回ずつ廻ることが分かる。別の見方をすると、$b^{(1)}$ が 1 バイトのみ活性で他の 3 バイトが固定の Λ 集合を 2^{24} 個足し合せた Λ 和集合を構成する。よって、$b^{(4)}$ が Λ 集合上でバランスすることになり、Type1 の拡張を利用すると 6 層攻撃が可能となる。この鍵推定の省略によって必要平文数は変わらないものの、4 バイト分の鍵推定が不要となるので、必要な計算量は 2^8 程度に削減できる。

さらに、以下に示す部分総和法を併用することで、計算量はさらに削減できる。

部分総和法　Ferguson らは SQUARE に対する 6 層攻撃での平文側鍵の推定を省略することに加え、カウンタを適当に利用して計算量を削減する部分総和法を提案している [7]。

6 層攻撃において、5 層目に対する入力 $b^{(4)}$ の注目する 1 バイトを $b_0^{(4)}$ とし、$b_0^{(4)}$ を入力とする S-box の出力に依存する 4 バイトの最終層 (第 6 層) 入力を各々$b_j^{(5)}$ $(j = 0, 1, 2, 3)$ とする。また、$b_j^{(5)}$ に対応する出力ビットを各々$c_j (= b_j^{(6)})$ $(j = 0, 1, 2, 3)$ とする。このとき、$b_0^{(4)}$ は出力 4 バイトとレイヤ鍵によって次のように表わせる。

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

ここで、w_j^{-1} は拡散行列の逆行列成分である。S-box の逆関数との合成関数 S_j を定義する。

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

また、$k_j^{(5)}$ の 4 バイトのうち実質的な情報量は 1 バイト分しかないので、$k_*^{(5)}$ で代表させる。

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

まとめると次式となり、5 バイトの鍵情報によって $b_0^{(4)}$ が計算できることが分かる。

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

部分総和法では、暗号文から一度にビット・バランスのチェック対象の中間値 (今の例では $b^{(4)}$) を復元するのではなく、段階的に鍵推定を行うことで計算量を節約する。表記を簡単にするため、次の x_l を定義する。

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

第 1 フェーズでは、$k_0^{(6)}$ と $k_1^{(6)}$ の 2 バイトを推定し、(x_2, c_2, c_3) を引数とするカウンタをカウントアップする。平文 2^{32} 個当りの計算回数は、$2^{16} \times 2^{32} = 2^{48}$。

第 2 フェーズでは、$k_2^{(6)}$ の 1 バイトを推定し、(x_3, c_3) を引数とするカウンタをカウントアップする。平文 2^{32} 個当りの計算回数は、$2^{16+8} \times 2^{24} = 2^{48}$。

第 3 フェーズでは、$k_3^{(6)}$ の 1 バイトを推定し、(x_4) を引数とするカウンタをカウントアップする。平文 2^{32} 個当りの計算回数は、$2^{16+8+8} \times 2^{16} = 2^{48}$。

第 4 フェーズでは、$k_*^{(5)}$ の 1 バイトを推定し、b_0^4 の総和を求め、ビット・バランスをチェックする。平文 2^{32} 個当りの計算回数は、$2^{16+8+8+8} \times 2^8 = 2^{48}$。

まとめると、平文 2^{32} 個当り、4×2^{48} 回の S-box 計算が必要である。擬似鍵を排除するためにこの計算を 6 回行なう必要があるので、S-box 計算回数は $6 \times 4 \times 2^{48} = 24 \times 2^{48} \sim 2^{52}$ と評価できる。

ここで、Ferguson らの評価に従い一回の暗号化の計算量を 2^8 と近似すると、規格化した計算量は $2^{52}/2^8 = 2^{44}$ と評価できる。

1 バイト不活性パターンの利用 (Ferg2)　Λ 和集合の考え方を利用すると、全平文を与えたとき、任意の層数でバランスが起こるので任意の層数に対する攻撃が可能であるように思える。しかし、出力も Λ 集合を形成するので鍵の推定は実際には出来ない。そこで、一部の状態バイトを固定する以下の攻撃法が提案された。

第 2 層入力 $b^{(1)}$ が 1 バイトのみ不活性で他の 15 バイトが活性となる入力を用意する。このとき、1 層後の $b^{(2)}$ の作る状態集合は、1 バイトのみ活性となる Λ 集合を $2^{8*(16-2)} = 2^{112}$ 個足しあわせた Λ 和集合である。よって、出力側に Type1 拡張を適用すると、7 層まで拡張できる。

なお、第 2 層入力 $b^{(1)}$ が 1 バイトのみ不活性とするには、2^{128} 個の平文を用意し、不活性バイトと結合する初層入力 $b^{(0)}$ の 4 バイトに対応する鍵 $k^{(0)}$ の 32 ビットを推定する必要がある。

表 2: Rijndael に対する SQUARE 攻撃

攻撃法	層数	平文数	複雑度	部分和
基本	4	2^9	2^9	無
Type1	5	2^{11}	2^{40}	無
Type2	5	2^{32}	2^{40}	無
Type1+Type2	6	2^{32}	2^{72}	無
Type1×2	6	2^{13}	2^{168}	無
Type2×2	6	2^{128}	2^{168}	無
Type1×2+Type2	7	2^{32}	2^{200}	無
Type1+Type2×2	7	2^{128}	2^{200}	無
Ferg1	6	2^{35}	$2^{44}/2^{64}$	有/無
Ferg1+Type1	7	2^{37}	2^{175}	有
Ferg2	7	2^{128}	2^{120}	有

2.2.3 Hierocrypt–L1 に対する SQUARE 攻撃

命題 5 *i が偶数のとき、$b^{(i)}$ が 1 バイトのみ活性の Λ 集合を廻るとき、$b^{(i+1)}$ は 1 列 4 バイトのみ活性な Λ 集合を成す。*

命題 6 *i が奇数のとき、$b^{(i)}$ が 1 バイトのみ活性の Λ 集合を廻るとき、$b^{(i+1)}$ は Λ 集合を成し、活性バイトは 4 個以上である。*

命題 7 *$b^{(i)}$ が 1 バイトのみ活性の Λ 集合を廻るとき、$b^{(i+2)}$ はバランスするが、一般には Λ 集合を成さない。*

命題 8 *i が偶数のとき、$b^{(i)}$ が同じ上位 S-box に属する 4 バイトのみが活性な 2^{32} 個の要素から成る Λ 集合を廻るとき、$b^{(i+1)}$ も全く同様の Λ 集合を成し、$b^{(i+2)}$ と $b^{(i+3)}$ は全バイトが活性な Λ 集合を成す。$b^{(i+4)}$ はバランスするが、一般には Λ 集合にならない。*

命題 9 *i が奇数のとき、$b^{(i)}$ が同じ上位 S-box に属する 4 バイトのみが活性な 2^{32} 個の要素から成る Λ 集合を廻るとき、$b^{(i+1)}$ と $b^{(i+2)}$ は全バイトが活性な Λ 集合を成す。$b^{(i+3)}$ はバランスするが、一般には Λ 集合にならない。*

基本攻撃　Rijndael と同様に、平文に 1 バイトのみ活性な Λ 集合を与えたとき、性質 **7** から $b^{(1)}$ は Λ 集合、$b^{(2)}$ はバランスするが、$b^{(3)}$ は一般にはバランスしない。よって、基本攻撃が適用可能な層数は一層減り、3 層となる。1 回に判定する鍵 $k^{(3)}$ のビット長は 8 ビットで、逆算する S-box の個数は 1 個である。

Type1 の拡張　基本攻撃を出力側に 1 段伸ばすには、推定する鍵ビットを 4 バイト=32 ビット増やせば良い。2 段伸ばすには、1+4+8 バイト (=104 ビット) の鍵推定が必要であるが、鍵の全数探索より効率が良い。

Type2 の拡張　基本攻撃を出力側に 1 段伸ばすには、平文セットを 4 バイトが活性な Λ 集合 (要素数 2^{32}) とし、推定する鍵ビットを 4 バイト=32 ビット増やす。推定する鍵ビット数は 4 バイト分の 32 ビット増え、攻撃できる層数は 1 層増える。

2 段伸ばすには、Type1 と同様 104 ビットの鍵推定が必要であり、鍵の全数探索より効率が良い。

5 層 (2.5 段) 攻撃 (Type1+Type2)　基本攻撃に Type1 および 2 の両方の拡張を適用したものが 5 層攻撃である。ただし、Type2 の拡張を行なうことで、活性 / 不活性のパターンが上位レベルでは 0.5 段ずれるので、Type1 拡張では最低 4 バイト (32 ビット) 分の鍵探索ビットの増加が必要となる。つまり探索鍵ビットは $4+1+4=9$ バイト=72 ビットとなる。

初段鍵推定の省略 (Ferguson らの改良 1)　4 バイト活性の平文セットを用意し、初期鍵の推定を行なわないのは Hierocrypt–L1 に対して極めて有効である。このとき、性質 **8** から $b^{(4)}$ が Λ 集合上でバランスするので、Type1 拡張と組み合わせることにより、7 層まで解読可能となる。

1 バイト不活性パターンの利用 (Ferguson らの改良 2)　第 2 層入力 $b^{(1)}$ で 1 バイトのみ不活性で他の 7 バイトが活性となる入力と鍵の組合せを利用する方式は Rijndael に対するほどには有効でない。$b^{(1)}$ の状態は Rijndael と同様に出来るが、Λ 和集合となるのは $b^{(3)}$ まで、バランスするのは $b^{(4)}$ までであり、Type1 の拡張を組み合わせても 6 層までが限界である。鍵推定ビット数は、$4+1+4$ バイト=72 ビットである。

表 3: Hierocrypt–L1 に対する SQUARE 攻撃

攻撃法	層数	平文数	複雑度	部分和
基本	3	2^9	2^9	無
Type1	4	2^{11}	2^{40}	無
Type2	4	2^{32}	2^{40}	無
Type1+Type2	5	2^{32}	2^{72}	無
Type1×2	5	2^{13}	2^{104}	無
Type2×2	5	2^{64}	2^{104}	無
Ferg1	6	2^{35}	$2^{44}/2^{64}$	有/無
Ferg1+Type1	7	2^{37}	$2^{110}/2^{132}$	有/無
Ferg2	6	2^{64}	$2^{86}/2^{123}$	有/無

2.3 truncated 差分

通常の差分解読法では、ビット単位の差分値に注目し、1 ビットでも差分値が異なれば別物として扱った。つまり、ブロック長が N ビットの場合、2^N 個の異なる差分値が存在した。

truncated 差分攻撃ではビット単位の違いではなく、S-box のサイズ (多くの場合、1 バイト=8 ビット) 単位で異なるビットの有無を調べ、その有無のパターンで差分値を分類する。つまり、8 ビット単位で異同を見た場合、差分パターンは $2^{N/8}$ 個に分類される。

truncated 差分は (全単射の)S-box や鍵加算によっては不変なので、拡散層の truncated 差分確率が計算でき、入出力分布に関する一様性の仮定を置けば、複数段での確率はそれらの積和として求められる。よって、truncated 差分攻撃は、特定のビット長単位の処理だけでアルゴリズムが構成されている場合、有効である可能性が高くなる。特に、入出力が 8 ビットの S-box と、バイト単位の演算による拡散層の組合せで構成される暗号に関して、truncated 差分攻撃による強度評価は不可欠である。

2.3.1 準備

Hierocrypt–L1 では全ての演算がバイト単位の処理で書けるので、バイト単位の truncated 差分を定義する。$8m$ ビット · データの差分 $\Delta X_{(8m)}$ に対する truncated 差分 $\chi\left(\Delta X_{(8m)}\right)$ を次式で定義する。

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

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

truncated 差分遷移 $\chi\left(\Delta X\right) \rightarrow \chi\left(\Delta Y\right)$ に対する truncated 差分確率を $\Pr\left(\chi\left(\Delta X\right) \rightarrow \chi\left(\Delta Y\right)\right)$ と書くことにする。

次に、32 ビット · データの truncated 差分 $\chi\left(\Delta X_{(32)}\right)$ のハミング重みを、truncated ハミング差分 $\eta\left(X_{(32)}\right)$ と定義する。

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

truncated ハミング差分確率は次式で定義する。

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

truncated ハミング差分と truncated ハミング差分確率は次式のように、$32m$ ビット · データに自然に一般化出来る。

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

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

後で述べるように、mds_{L} 関数および MDS_{L} 関数において、truncated 差分確率と truncated ハミング差分確率は一致する。

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

表 4: mds_{L} 関数の truncated ハミング差分確率 (確率の 2 べき近似を利用)

$\eta(\Delta X_{(32)})$ \ $\eta(\Delta Y_{(32)})$	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

$$\Pr\left(\chi\left(\Delta X_{(64)}\right) \to \chi\left(\Delta Y_{(64)}\right)\right) = \Pr\left(\eta\left(\Delta X_{(64)}\right) \to \eta\left(\Delta Y_{(64)}\right)\right) \ .$$

この性質を利用することによって、Hierocrypt–L1 の truncated 差分に関する強度解析は簡略化できる。

2.3.2 構成要素の特性の検討

S-box

truncated 差分攻撃では、S-box はランダムな全単射写像であることが仮定される。既に述べたように Hierocrypt–L1 の S-box は、理論的最大差分 / 線形確率が最小値を取り、代数次数が 7 次で多項式表現での項数が十分多いので、十分にランダムであると考えられる。

mds_{L} 関数

mds_{L} 関数は 4 並列の 8 ビット語に対する MDS(最大距離分離符号) になっている。この性質だけから、mds_{L} 関数の truncaed 差分確率は一意的に決まり、truncated 差分確率が truncated ハミング差分と一致することが分かる [12]。表 4 に、確率を 2 のべき乗で近似した結果を示す。

MDS_{H} 関数

MDS_{H} 関数はバイト単位の排他的論理和だけで構成されている。松井による truncated 差分確率値の近似計算アルゴリズムによって、truncated 差分確率の 2 べき乗で表わされる近似評価が得られる [14]。

2.3.3 多段に対する評価

Hierocrypt–L1 では、S-box と鍵加算を外すと、MDS_{H} 関数と MDS_{L} 関数が交互に並んでいる。既に述べたように、MDS_{L} 関数の truncated 差分確率は truncated ハミング差分確率に等しい。よって、MDS_{L} 関数が両端に来る組合せ (LHL$\cdots$HL) の最大 truncated 差分特性確率の解析には、MDS_{L} 関数と MDS_{H} 関数の truncated ハミング差分確率だけが必要である。

truncated ハミング差分を使用することの最大の利点は、MDS_{H} の遷移確率表のサイズにおいて顕著である。通常の truncated 差分確率の場合、表サイズは約 2^{16} であったが、truncated ハミング差分を利用すると約 $5^4 (\simeq 2^{9.29})$ と大幅な節約が可能になる。

以下に、多段に対する解析の手順を示す。

1. mds_{L} の truncated ハミング差分確率表を作成

2. MDS_{H} の truncated ハミング差分確率表を作成

3. LH(MDS_{L} 関数の後に MDS_{H} 関数を適用) の truncated ハミング差分確率表を作成

4. (LH) の t 乗の truncated ハミング差分確率表を順次作成。

5. 前項の結果に L を掛け合わせることで、$t+1$ 段の truncated ハミング差分確率表として求まる

6. ランダム行列に対する評価と比較し、差が無くなる段数を特定する。

上記の手順を実行することによって、Hierocrypt–L1 では 3 段 (LHLHL) で truncated 差分特性確率がランダム行列の場合と区別できなくなることが確認できた。よって、2 段攻撃を仮定した場合、5 段で十分安全と評価できる。

2.4 高階差分攻撃

高階差分攻撃法は、ブール関数の代数次数を d とするとき、次の性質を利用して連立方程式を立て、それを解くことで鍵を求める代数的攻撃法である [10]。

- $(d+1)$ 階差分値が 0
- d 階差分値が入力に依存しない定数

高階差分攻撃に対する安全性を保証するには、連立方程式で鍵を推定する段と鍵の直接探索を適用する段を取り除いた残りの部分に対し、可能ないかなる高階差分を取っても出力差分値の入力依存性が残ることを示せば良い。しかし、一般には解析的にこの条件が満たされるか否かを判断することは困難であり、暗号アルゴリズムの設計を工夫することで、危険性を抑えるのが通常である。対策の基本は次の通りである。

- 唯一の非線形変換である S-box の代数的性質の最適化
- 線形変換部分である拡散層の拡散性を最適化

S-box に関しては、代数次数を全単射関数として最大の 7 次にし、かつ、入力側でのビット置換という代数構造を複雑にする変換を入れてある。また、拡散層は、入出力の幅でのビット攪拌が十分に行なわれるように MDS 行列を用い、さらに、S-box との組合せたときの多項式表現の項数の最大化を行なっているので、効率的な高階差分が存在する可能性は極めて低いと期待できる。

2.5 補間攻撃

補間攻撃は、暗号化関数を多項式表現で表わし、各項の係数を推定することで暗号化関数自体を特定する攻撃である。暗号化関数を多項式表現で表わしたときの項の個数が評価指標となる。この攻撃が有効なのは、暗号化アルゴリズムが $\mathrm{GF}(2^8)$ 上の演算だけで構成されるといったように、代数構造を壊さない処理しか使われない場合である。Hierocrypt–L1 の場合、S-box には $\mathrm{GF}(2^8)$ 上のべき乗演算を使ってはいるものの、入力側でのビット置換を組み合わせているため代数構造が複雑化されているので、容易な適用は困難となっていると考えられる。

2.6 Impossible Differential 攻撃

Impossible Differential 攻撃は、端の鍵を仮定し平文側と暗号文側から中間段で出現可能なパターンの集合を割り出し、矛盾が生じたとき仮定した鍵を棄却することで、正しい鍵を絞り込む攻撃法である。

一般的な傾向として、拡散層の結合が密な場合ほど、段数の増加に対する出現不可能なパターンの減少が急であると考えられる。

Hierocrypt–L1 では 1 バイトの差分が 1 段後に全バイトに伝わるが、前述のように複数経路性が保証されるように MDS_{H} が設計されているので、どのバイトも零差分を取り得る。この性質によって、有効な不可能差分パスが存在する確率は極めて低いと考えられる。

以上の考察に加え、truncated 差分確率が近似評価ながら 3 段でランダム関数と区別できなくなったことも考え合わせると、Hierocrypt–L1 に対する Impossible Differential 攻撃は実用的でないと考えられる。

2.7 Non-surjective 攻撃

この攻撃は、全単射性を満たさない構成要素がある場合、実現できない中間状態が存在するので、その性質を利用して鍵を推定する攻撃法である。Hierocrypt–L1 では全部の構成要素が全単射なので、この攻撃は適用不可能である。

2.8 Mod n 攻撃

この攻撃も、Non-surjective 攻撃と同様、局所的な非全単射性から生じる出現可能パターンの分布の偏りを中間データの Mod n を取ることで調べ、利用する方式である。Hierocrypt–L1 では全単射の構成要素のみで構成されるので、このような攻撃は適用不可能である。

2.9 χ^2 攻撃

特定の入出力ビット集合の間の遷移確率の統計的偏りを理論および数値解析で割り出し、χ^2 検定で鍵推定の成否を判定する攻撃法。Hierocrypt–L1 では MARS で使用されている乗算のように、ビット相関の偏りが強い演算を用いていないので、χ^2 攻撃が有効である危険性は低いと考えられる。

2.10 Hierocrypt–L1 に対する攻撃論文

2.10.1 SQUARE 攻撃

Barreto らは、文献 [16] において Hierocrypt–L1 および Hierocrypt–L1 に対する SQUARE 攻撃の改良を提案した (2.2)。その結果、Hierocrypt–L1 が 7 層 (3.5 段) まで解読できること示した。この結果は、2001 年 4 月の FSE 2001 で発表されたが、我々は 2001 年 1 月に同じ結果を発表している [21]。

Hierocrypt–L1 の段数はそれぞれ、12 層 (6 段) と設定されているので、現状では SQUARE 攻撃に対して十分な安全と言える。

2.10.2 Impossible Differential 攻撃

Cheon らは、2 段の有効な Impossible Differential を発見し、3 段まで攻撃可能であることを示した [4](2.6)。彼らの評価によると、選択平文 2^{55} 個と暗号化 2^{71} 回分の計算で、鍵が推定できる。しかし、実際の Hierocrypt–L1 は 6 段以上であり、現状では Impossible Differential 攻撃に対して、十分安全と言える。

2.10.3 鍵スケジュール

Hierocrypt–L1 の鍵スケジュールは、128 ビット幅の折り返し型の中間鍵生成部と、中間鍵に線形変換を施してラウンド鍵を生成するラウンド鍵生成部の 2 つで構成されている。折り返し型であるため、折り返し点に対して対称な位置に有る 2 つの段は中間鍵を共有する。そのため、ラウンド鍵生成部は 2 つの段のラウンド鍵ビット間に簡単な関係が存在しないように設計する必要がある。

古屋他は、Hierocrypt–L1 の鍵スケジュールに対する解析を行ない、ラウンド鍵の間に存在する線形関係式を複数発見した [17]。このことは、設計方針が完全には満たされていないことを示している。しかし、この関係式を利用した、既存の攻撃法の改良または新規攻撃法は提案されていない。また、データ攪拌部が差分 / 線形解読法を始めとする既存の攻撃法に対する十分な強度があり、Hierocrypt–L1 の安全性に脅威を与えるものではない。

3 ソフトウェアでの実装評価

公募要領で要求されている，速度評価，メモリ使用量 (コード量・ワークエリア)・最適化の有無・記述言語・評価プラットフォーム等について記述する．

3.1 実装環境

性能評価計測を行なった実装環境を表 5, 6 に示す。表 5 の 3 種類の環境は、昨年度の CRYPTREC で実施された計測で利用されたものである [19]。Hierocrypt–L1 に適した実装の最適化については、文献 [8, 18, 20] を参照されたい。

表 5: プラットフォーム 1: Client, High performance, Sever 各環境

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	?	?

表 6: プラットフォーム 2: JAVA, 8-bit, Smart card 各環境

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

3.2 計測法について

ソフトウェア実装での性能評価の計測法について述べる。Client 環境 (Pentium III+MASM) で計測する場合を例に説明するが、他の環境についても同様である。

Pentium III に組み込まれた Time Stamp Counter (TSC) を利用して、鍵スケジュール・データ暗号化・データ復号の各プロセスに必要なサイクル数を計測する。計測の曖昧さを取り除くために、図 1 に示す速度評価用プログラム片を利用した。図で、"required CPU cycles/BENCH_COUNT" は function call を含む 1 ブロックの処理に必要なサイクル数を意味する。つまり、プロセッサの周波数に応じたスループットを表している。なお、Alpha 21264 プロセッサでは、RTSC の代わりに cl ock( ) を利用した。

以上の要領で、暗号化速度および復号速度、そして鍵セットアップ時間を含む暗号化速度および復号速度を計測した。

3.3 計測結果

3.3.1 Client 環境 (Pentium III)

表 5 の Client 環境は，昨年度の CRYPTREC が実施した実装性能評価で利用されたものであり、代表的 CPU である Pentium III を搭載した PC 上で計測を行なった [19] . 表 7 は、Hierocrypt–L1 の ECB モードで 100 万ブロック暗号化する試行を 10 回行なった際の、最小のサイクル数である。上段は各試行の最初のブロックだけ鍵セットアップを行ない、その後は鍵を固定して計測したもので，実質的には鍵セットアップなしの結果と等しい。下段は 1 ブロックごとに鍵セットアップを行なって計測したものである。

表 8 に使用メモリ量を示した。

表 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

```
#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;
```

図 1: Piece of speed evaluation program on Pentium III

表 8: Pentium III memory usage

Operation	Code size	Work area
Enc/Dec	52982	448

3.3.2 High End 環境 (Alpha 21264)

表 5 の High End 環境は，昨年度の CRYPTREC が実施した実装性能評価で利用されたものであり、高性能 CPU として代表的な DEC 社の Alpha 21264 を搭載したワークステーションで計測を行なった。

表 9 は、Hierocrypt–L1 の ECB モードで 100 万ブロック暗号化する試行を 10 回行なった際の、最小のサイクル数である。上段は各試行の最初のブロックだけ鍵セットアップを行ない、その後は鍵を固定して計測したもので，実質的には鍵セットアップなしの結果と等しい。下段は 1 ブロックごとに鍵セットアップを行なって計測したものである。

表 10 に使用メモリ量を示した。

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

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

表 10: Alpha 21264 memory usage

Operation	Code size	Work area
Enc/Dec	84328	448

3.3.3 Server 環境 (Ultra SPARC IIi)

表 5 の Server 環境は，昨年度の CRYPTREC が実施した実装性能評価で利用されたものであり、サーバ用ワークステーションの CPU としてポピュラーな Sun Microsystems 社製 Ultra SPARC IIi を搭載したワークステーションで計測を行なった。

表 11 は、Hierocrypt–L1 の ECB モードで 100 万ブロック暗号化する試行を 10 回行なった際の、最小のサイクル数である。また、表 12 に使用メモリ量を示した。上段は各試行の最初のブロックだけ鍵セットアップを行ない、その後は鍵を固定して計測したもので，実質的には鍵セットアップなしの結果と等しい。下段は 1 ブロックごとに鍵セットアップを行なって計測したものである。

表 12 に使用メモリ量を示した。

表 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

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

Operation	Code size	Work area
Enc/Dec	24496	448

3.3.4 JAVA 環境

JAVA 環境での性能評価は、Sun Microsystems 社が提供する JDK1.3.1 を利用した。また、セキュリティ・インターフェイスを付けたときのオーバヘッドの影響を見るために、IAIK 製の JCE(JAVA Cryptography Extension) である IAIK-JCE 2.61 を利用した。IAIK-JCE 2.61 は、Sun Microsystems 社が提供する仕様である JCE 1.2 に基づいて設計されている。

測定には東芝製ノート PC DynaBook SS3480 DS60P/1n2L を利用した[2]。

表 13 に開発環境ならびに実行環境を、表 14 に計測結果をまとめた。なお、速度は動作周波数を 200MHz に換算した値であり、実際の計測値の 1/3 である。

表 13: JAVA Environment

開発環境		Sun Microsystems 社製 JDK1.3.1
測定環境	測定機種	DynaBook SS 3480 DS60P/1N2L
	CPU	Intel SpeedStep テクノロジ対応低電圧版モバイル Pentium II (600MHz)
	メモリ	192 MByte
	OS	Microsoft Windows 2000 5.00.2195 Service Pack 2

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

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

3.3.5 8-bit 環境

8-bit 環境として Z80 での性能をシミュレータを利用して計測した。シミュレータは次のサイトから入手した z80pack に含まれる z80sim に、エンディアンを補正をするためのパッチを当てたものを利用した。

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

ECB モードでの暗号化および復号の 1 ブロック分の処理に必要な state 数を計測した。なお、暗号化および復号ともに鍵スケジュールを含んでいる。計測結果を表 15 に示した。

[2]仕様は、http://dynabook.com/pc/catalog/oldpc/ss/ss34t2/spec.htm

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

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

表 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	4196	–	–	–	–	–	–	–

3.3.6 スマートカード

本評価は，Z80[2] を CPU として搭載しているスマートカードにおいて，処理速度優先の方針で Z80 アセンブラにより記述した実装結果である．対象として東芝製スマートカード JT6N55 を使用した．

評価プラットフォームおよび実装環境　表 17 に記述言語および開発環境を含むソフトウェアでの実装評価に用いたプラットフォームの詳細を記述する．スマートカードではプログラムは ROM 領域に格納され改変は出来ない．また，暗号処理以外の処理にも ROM，RAM および EEPROM が必要とされるため，要求されている速度が得られる範囲内で小さなコード量が求められる．

表 17: 評価プラットフォーム仕様

機種	JT6N55	
プロセッサ	Z80 (5MHz)	
メモリ	ROM	48KB
	RAM	1KB
	EEPROM	8KB
記述言語	Z80 アセンブラ言語	

速度評価およびメモリ量評価　Z80 アセンブラ記述による JT6N55[1] 上での速度評価を行う．処理ステート数の見積もりには通常の Z80 アーキテクチャでの規定ステート数を用いており，Z80 の最小インストラクションの実行には 4 ステートが必要となる．実装においては，処理速度優先としたが，スマートカードの全領域を使用するコードは現実的でないため，3KB 以内のコード量と on-the-fly 鍵生成を実装条件とした．

平文格納エリア，暗号文格納エリア，鍵格納エリアのアドレスをパラメータ指定し，Hierocrypt-L1 サブルーチンを呼び出して暗号文が暗号文格納エリア (RAM) までに必要となるステート数および必要となるメモリ量を表 18 に示す．

本評価においては，処理速度優先の実装を行っており，最も効率の良い 1,280 バイトの参照テーブルを使用したためコード量はコンパクトな実装と比較して大きい．コンパクトな実装の場合には，参照テーブルは S-box の 256 バイトと若干の定数だけが必要であるため，コード量を削減できる．

表 18: スマートカードでの速度およびメモリ量評価

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

4 ハードウェアでの実装評価

ハードウェア実装は、ASIC で速度優先および面積優先の設計、および、FPGA での速度優先の設計を行なった。

4.1 ASIC による実装

4.1.1 速度優先設計 (ASIC-1)

1. 使用プロセス

   0.25 μm 3 層配線 CMOS

2. 設計環境

   SYNOPSYS 社製 Design Compier 1999.10-3

3. シミュレーション条件 (コマーシャル用ワーストケース)

   1.35V 70 ℃ (標準ケースでは, 1.5V 25 ℃)

4. スループット

   1081Mb/s(9.86ns, 6 clock)

5. 使用リソース

   81.2K ゲート

4.1.2 速度優先設計 (ASIC-2)

1. 使用プロセス

   0.15 μm 3 層配線 CMOS

2. 設計環境

   SYNOPSYS 社製 Design Compier 1999.10-3 ?

3. シミュレーション条件 (コマーシャル用ワーストケース)

   1.35V 70 ℃ (標準ケースでは, 1.5V 25 ℃)

4. スループット

   1568Mb/s(6.80ns, 6 clock)

5. 使用リソース

   54.9K ゲート

4.1.3 面積優先設計 (ASIC-3)

SBOX, MDSL を共有して面積縮小を目指した設計.

1. 使用プロセス

   0.25 μm 3 層配線 CMOS ?

2. 設計環境

   SYNOPSYS 社製 Design Compier 1999.10-3

3. シミュレーション条件 (コマーシャル用ワーストケース)

   1.35V 70 ℃ (標準ケースでは, 1.5V 25 ℃)

4. スループット

   135.0Mb/s(18.22ns, 26 clock)

5. 使用リソース

   9.9K ゲート

4.2 FPGA による実装

4.2.1 速度優先設計 (FPGA-1)

1. 設計環境

   ALTERA 社製 Max+plus II ver. 9.6

2. 動作速度

   51.0Mb/s(11.16MHz, 89.6ns, 14 clock)

3. 使用リソース

   11.0K ロジックセル, ALTERA 社製 Flex 10K ファミリー使用時

4.3 ハードウェア実装性能のまとめ

以上の結果を表にまとめる。

表 19: ASIC implementation

implementation	Rule	Throughput	Area		Critical path	Latency
	(μm)	(Mbps)	(Kgate)	(K logic cell)	(ns)	(clock)
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 おわりに

以上に述べたように、Hierocrypt–L1 は既存の各種解読法に対し十分に安全と考えられる。また、ミドルウェア、スマートカード、専用 LSI など幅広いプラットフォーム上で、高い性能が実現可能である。

参考文献

[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-H. Cheon, M-J. Kim, and K. Kim. Impossible differential cryptanalysis of Hierocrypt-3 reduced to 3 rounds. In *Proc. of 2nd NESSIE Workshop*, 2001. to appear in LNCS.

[5] J. Daemen, L. R. Knudsen, and V. Rijmen. The block cipher Square. In *FSE'97*, volume 1267 of *LNCS*, pages 149–165, 1997.

[6] J. Daemen and V. Rijmen. *AES Proposal: Rijndael*. http://www.esat.kuleuven.ac.be/∼rijmen/rijndael/rijndaeldocV2.zip.

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

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

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

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

[11] 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.

[12] 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.*

[13] M. Matsui. Linear cryptanalysis method for des cipher. In *Eurocrypt'93*, volume 765 of *LNCS*, pages 386–397. Springer Verlag, 1994.

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

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

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

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

[18] 佐野・村谷・大熊・川村・本山. Hierocrypt の実装について. *SCIS 2001*, 13A-2, 2001.

[19] 情報処理振興事業協会 セキュリティセンター. *CRYPTREC Report 2000*, 2001.

[20] 清水・佐野・本山・大熊・川村. Spn 型ブロック暗号の実装について. 電子情報通信学会 技術研究報告, ISEC 2001-55(2001-09):17–21, 2001.

[21] 大熊・佐野・村谷・本山・川村. ブロック暗号 Hierocrypt-3 および Hierocrypt-l1 の安全性について. *SCIS 2001*, 11A-4, 2001.

[22] 大熊・村谷・佐野・川村. ブロック暗号 Hierocrypt の仕様と評価. 電子情報通信学会 技術研究報告, ISEC2000-7(2000-05):77–104, 2000.

[23] 大熊・村谷・佐野・本山・川村. 'ブロック暗号 Hierocrypt-3 および Hierocrypt-l1 の 強度 / 性能評価. 電子情報通信学会 技術研究報告, ISEC 2000-71(2000-09):71–100, 2000.