コンピュータセキュリティ 11－8
（2000．9．8）

64ビット版Hierocryptの提案

村谷 博文† 大熊 建司† 佐野 文彦‡ 本山 雅彦† 川村 信一†
†(株) 東芝 研究開発センター 〒212-8582 川崎市幸区小向東芝町1
{hirofumi.muratani,kenji.ohkuma,masahiko.motoyama,shinichi2.kawamura}@toshiba.co.jp
‡(株) 東芝 SI技術開発センター 〒183-8512 東京都府中市片町3-22
fumihiko.sano@toshiba.co.jp

あらまし 著者らは階層的に差分／線形解読法に対する安全性を保証する、2階層入れ子型(再帰的)SPN構造を持った128ビット・ブロック暗号 "Hierocrypt" を提案した。その提案では、上位拡散層の設計に利用するガロア体の違いによって、Type-I ($GF(2^{32})$) と Type-II ($GF(2^4)$) の2種類があった。また、鍵スケジュールには、256ビット幅の変形 Feistel 型構造を採用した。本報告では、両方式の実装検討結果を参考に、入れ子型 SPN 構造の柔軟性を利用して開発した、64ビット版 Hierocrypt-L1 を提案する。

Proposition of a 64-bit version of Hierocrypt

Hirofumi Muratani† Kenji Ohkuma† Fumihiko Sano‡ Masahiko Motoyama† Shinichi Kawamura†
† Computer & Network Systems Laboratory, Corporate Research & Development Center
1, Komukai Toshiba-cho, Saiwai-ku, Kawasaki, 212-8582, Japan
‡ System Integration Technology Center
3-22, Katamachi, Fuchu-shi, Tokyo 183-8512, Japan

Abstract The authors proposed two 128-bit block cipher algorithms Hierocrypt Type-I and Type-II based on a nested SPN strucuture where the upper-level S-box consists of the lower-level SP network hierarchically. The key scheduling parts are designed on a 256-bit modified Feistel structure. This paper proposes a 64-bit version "Hierocrypt-L1," based on the Type-II algorithm.

1 はじめに

近年、暗号アルゴリズム標準化の動きが活発である。次世代共通鍵ブロック暗号の主流はブロック長128ビットであるが、64ビット暗号も少なくとも今後10年は利用されると見られている。

筆者らは、入れ子型SPN構造暗号の構成要素がある程度独立に設計できることによる柔軟性に着目し、128ビット改良版 Hierocrypt-3[2] をベースに 64 ビット版 Hierocrypt-L1(鍵長128ビット) を開発した。両者には、2階層入れ子型構造, 下位 S-box(s), 上位 S-box(XS) など共通点が多く、64ビット化の作業は短期間であった。主な相違点は、上位拡散層の MDS 変換 MDS_{H} と鍵スケジュール部であり、基本設計自体はほぼ同じである。

以下では、Hierocrypt-L1 の設計方針, 設計基準, 安全性評価, アルゴリズムの仕様について、Hierocrypt-3 との違いを中心に述べる。

2 設計方針

128ビット暗号 Hierocrypt-3 を 64 ビット化して Hierocrypt-L1 を設計するに当たり、Hierocrypt-3 の特長はそのまま保つことを基本とした。そこで、データ攪拌部は上位攪拌層 MDS_{H} のみの変更に留めた。Hierocrypt-3 の鍵スケジュール部は 64 ビット幅の処理が中心だったが、これを 32 ビット幅の処理に置き換えることでサイズを半分にした。その際、拡散効果を持つ $P^{(n)}$ 関数のスケーラビリティが有効だった。

本節では、これらの構造と選択の理由について述べる。次節で設計基準について述べる。

2.1 データ攪拌部

2.1.1 入れ子型 SPN 構造

データ攪拌部の基本構造として、Hierocrypt-3 と同様、2 階層入れ子型 SPN 構造を採用した。

我々は、一般の入れ子型 SPN 構造暗号設計に当たり、満たすべき条件を文献 [3, 4] で提案した。それを具体的に 64 ビット暗号に適用した結果、以下の条件が得られた。

(a) (下位)S-box のサイズは 8 ビット
(b) 2 階層の入れ子構造 (上位レベルと下位レベル)
(c) 下位レベルは 2 段 SPN 構造
(d) 上位／下位レベルとも、拡散層幅は S-box 2 個分

ここで、64 ビット化に伴ない変更した条件は (d) のみである。ここで、下位レベルを S-box 2 個分 (16 ビット幅)、上位レベルを S-box 4 個分にすることも考えられた。しかし、差分／線形解読法に対する強度は差が無いので、Hierocrypt-3 と構成要素を共通化できる (d) の方を選んだ。

Hierocrypt-L1 のデータ攪拌部でも、簡潔で透明性が高いという入れ子型 SPN 構造の特長は成立している。また、Hierocrypt-L1 と Hierocrypt-3 は共通部分が多く、ブロック・サイズの変更に容易に対処できることを実証している。

2.2 鍵スケジュール部

Hierocrypt-L1 の鍵スケジュール部は、Hierocrypt-3 同様、繰り返し段構造から成る中間鍵生成部と、各段の中間鍵から拡大鍵を生成する拡大鍵生成部から成り、安全性および実装面での長所がそのまま保たれるように設計されている。

主要な違いの一つは、中間鍵の幅を半分の 128 ビットにした点であり、鍵の段関数を全単射としたのは同じ。拡大鍵生成部はデータ攪拌部 1 段当たり 128 ビットの拡大鍵を供給する。構成は以下の通り。

128 ビットを 64 ビット 2 組に分け、一方を繰り返し線形変換型、他方を Feistel 型とし、前者が後者の鍵に相当するデータを供給する。この構造で線形変換を全単射にすれば、128 ビット幅の全単射の繰り返し段構造が構成できる。また、Feistel 構造の F 関数の拡散層は 32 ビット幅になり、十分高速な動作が実現できる。

中間鍵生成部も Hierocrypt–3 と同様に、折り返し型を採用した。なお、段依存定数の入れ方や中間鍵段関数 σ の設計は、データ幅を考慮した適切な変更を行なった。

中間鍵から拡大鍵の生成は、中間鍵とそれが作られる途中の中間状態のデータを 32 ビット単位に分け、それらの排他的論理和で構成した (Hierocrypt–3 では 64 ビット幅)。弱鍵の出現を抑えるよう、拡大鍵生成方法や段依存定数を適切で透明な設計を行なった。

3 設計基準

Hierocrypt の設計に当たり、次の点を重視した。

- 主要な攻撃法に対する十分な安全性
- スマートカードやミドルウェアでの暗号化の高速性
- 実装の効率性
- 設計の透明性

安全性・高速性・実装効率に関する基準も Hierocrypt–3 と同様である。

3.1 構成要素の設計

3.1.1 S-box

(下位)S-box は、Hierocrypt–3 と共通である。

3.1.2 mds_{L}

下位拡散層 mds_{L} は、Hierocrypt–3 と共通である。

3.1.3 MDS_{H}

MDS_{H} の設計は基本的に Hierocrypt–3 と同じである。主な相違点は、行列サイズが半分になることと、行列の巡回性を外した点である。外した理由は、行列候補の数が限られるので巡回性を課さなくても全探索が十分可能だからである。

設計基準は以下の通りである。

i) 最大距離分離 (MDS) 行列である
ii) バイト単位で見た複数経路性
iii) バイト間結合数が出来るだけ少ない

設計手順は以下の通り。

1. 定数倍の候補である GF(2^4) の元で行列を作る
2. MDS であれば次項へ。否なら前項に戻る
3. 逆行列を求め、行列要素が全部 GF(2^4) の候補に含まれているなら次項へ。否なら先頭に戻る。
4. 最終的に残った巡回行列を候補とする

上記の手順により、行内要素の回転と反転の自由度を除き、4 種類の候補が求まった。この中から、ソフト実装が最も効率良く出来るものを選んだ結果、下記の行列が得られた。

$$\begin{pmatrix} 5 & 7 \\ \mathrm{A} & \mathrm{B} \end{pmatrix}$$

ここで、行列要素は GF(2^4) の元を 16 進表現したものであり、原始多項式は z^4+z+1 である。

3.1.4 $P^{(n)}$

拡散層 $P^{(n)}$ は Hierocrypt–3 と共通である。(n) が拡散幅の 1/4 というスケーラビリティがあるので、対応部分の拡散幅が半分になってもそのまま利用できる。

4 安全性評価

4.1 差分解読法および線形解読法

関数 f の最大差分確率を dp^f、最大線形確率を lp^f とする。

4.1.1 S-box の特性

(下位)S-box は、Hierocrypt–3 と共通である。

4.1.2 活性 S-box 数

上位 S-box XS は Hierocrypt–3 と共通である。よって、XS が活性であれば、中に含まれる S-box のうち少なくとも 5 個は活性である。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 個の上位 S-box XS が活性となる。文献 [4, 3] 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 (線形) 特性確率 $DP^{2\mathrm{R}}(LP^{2\mathrm{R}})$ は次式の不等式を満足する。

$$DP^{2\mathrm{R}} \le (dp^S)^{15} = (2^{-6})^{15} = 2^{-90} .$$

$$LP^{2\mathrm{R}} \le (dp^S)^{15} = (2^{-6})^{15} = 2^{-90} .$$

これらの値は 2^{-64} より小さく、有効な特性確率の経路は存在しない。2 段攻撃を仮定すれば、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。

4.1.3 証明可能安全性に基づく評価

上位 S-box XS は Hierocrypt–3 と共通なので、その最大差分 (線形) 確率 $dp^{XS}(lp^{XS})$ の上限値は以下のように評価される。

$$dp^{XS} \le (dp^f)^4 = (2^{-6})^4 = 2^{-24} ,$$

$$lp^{XS} \le (lp^f)^4 = (2^{-6})^4 = 2^{-24} .$$

同様に、連続する 2 段に対し、次の不等式が成立する。

$$dp^{2\mathrm{R}} \le (dp^{XS})^2 = (2^{-24})^2 = 2^{-48} ,$$

$$lp^{2\mathrm{R}} \le (lp^{XS})^2 = (2^{-24})^2 = 2^{-48} .$$

2 段での最大確率 2^{-48} は、全数探索より効率の良い解読の存在を否定しない。そこで、2 段および XS に対する最大確率の上界値を用いて、3 段以上の近似確率評価を行なう。2 段攻撃を仮定し、偶数段に対しては 2 段の確率の積で、奇数段に対しては 2 段の確率の積と残り 1 段分の XS の確率の積で評価した結果を表 1 に示す。この表からは一見 5 段で十分に見えるが、しかし、鍵長は 128 ビットなので 64 ビット分余計に鍵探索ビットが確保できる。活性 S-box 数に基づく評価と同じ仮定と合わせて表 1 の結果を得た。

Table 1: 証明可能安全性に基づく最小段数の評価

鍵長 (bit)	最小段数 R	$R-2$	特性確率
–	4 段	2 段	2^{-48}
–	5 段	3 段	2^{-72}
128 ビット	6 段	4 段	2^{-96}

ここでの対差分／線形解読強度評価の近似精度は通常の最大特性確率よりも高い。

4.2 SQUARE 攻撃

SQUARE 攻撃は、Hierocrypt–L1 に対して 5 層までしか有効ではないことを確認した。
Hierocrypt–L1 は、鍵長 128 ビット、12 層に設計されているので、SQUARE 攻撃に対して十分な安全性を有する。

基本攻撃 Hierocrypt–3 と同様、層数と基本攻撃 1、基本攻撃 2 を定義する。

タイプ 1 およびタイプ 2 の拡張 Hierocrypt–3 と同様、初層側に 1 層伸ばすのをタイプ 1 拡張、終層側に 1 層伸ばすのをタイプ 2 拡張とする。
拡張の各種組合せに対する考察結果を表 2 に示す。

4.3 truncated 差分

Hierocrypt–L1 でも Hierocrypt–3 と同様、truncated 差分攻撃に対する耐性を評価する必要がある。また、解析においては、全く同じ定義の truncated 差分およびに truncated ハミング差分が重要な役割を果たす。

Table 2: Hierocrypt-L1 への SQUARE 攻撃

攻撃のタイプ	層数	平文数	演算数	メモリ量
基本 1	3 層	2^9	2^9	small
基本 1+ タイプ 1	4 層	2^{11}	2^{40}	small
基本 1+ タイプ 1×2 回	5 層	2^{12}	2^{104}	2^{12}
基本 1+ タイプ 2	4 層	2^{64}	2^{72}	2^{64}
基本 1+ タイプ 1 + タイプ 2	5 層	2^{64}	2^{104}	2^{64}
基本 2	3 層	2^9	2^9	small
基本 2+ タイプ 1	4 層	$\geq 2^{11}$	$\geq 2^{40}$	$\geq 2^{11}$
基本 2+ タイプ 1×2 回	5 層	$\geq 2^{12}$	$\geq 2^{104}$	$\geq 2^{12}$
基本 2+ タイプ 2	4 層	2^{32}	2^{40}	2^{32}
基本 2+ タイプ 2×2 回	5 層	2^{64}	2^{104}	2^{64}
基本 2+ タイプ 1 + タイプ 2	5 層	2^{32}	2^{72}	2^{32}

4.3.1 構成要素の特性の検討

S-box Hierocrypt-3 の解析で示したように、十分にランダムであると考えられる。

mds_{L} 関数 Hierocrypt-3 と同じものであり、同じ truncated ハミング差分確率表 (確率の 2 べき近似を利用) が解析に使える。

MDS_{H} 関数 MDS_{H} 関数は Hierocrypt-3 のときと同様、バイト単位の排他的論理和だけで構成される。松井によるアルゴリズムによって、truncated 差分確率の 2 べき乗近似評価が得られる [1]。

4.3.2 多段に対する評価

Hierocrypt-3 と同様、truncated ハミング差分を使用することで MDS_{H} の遷移確率表サイズは大幅に縮小できる。通常の truncated 差分確率を使った場合のサイズが約 2^{16} なのに対し、truncated ハミング差分を利用すると約 $5^4 (\simeq 2^{9.29})$ となる。
多段に対する解析の手順は、MDS_{H} が異なる以外は全く同じである。解析の結果、Hierocrypt-L1 では 3 段 (LHLH) 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。

4.4 その他の攻撃

高階差分攻撃 Hierocrypt-3 のときと同様の考察により、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。

補間攻撃 Hierocrypt-3 と比較すると、上位 S-box xs が共通、上位拡散層 MDS_{H} も複数経路性は共通なので、適用は困難と期待できる。

Impossible Differential 攻撃 Hierocrypt-3 と同様、MDS_{H} に複数経路性が有るので、適用は困難と考えられる。

Non-surjective 攻撃 Hierocrypt-3 と同様、全構成要素に全単射性があるので適用不可能である。

Mod n 攻撃 Hierocrypt-3 と同様、全構成要素に全単射性があるので適用不可能である。

χ^2 攻撃 Hierocrypt-3 と同様、ビット相関の偏りが強い演算を用いておらず、適用の可能性は低いと考えられる。

References

[1] M. Matsui. ブロック暗号 e2 の差分経路探索. *Technical Report of IEICE(Japan) ISEC99-19*, 1999.

[2] K. Ohkuma, H. Muratani, F. Sano, M. Motoyama, and S. Kawamura. A revised nested spn cipher. *Technical Report of IPSJ(Japan) CSEC11-7*, Vol. 11(in this volume), No. 7, 2000.

[3] K. Ohkuma, H. Muratani, F. Sano, and S. Kawamura. The block cipher hierocrypt. In *Selected Areas in Cryptography 2000*, 2000.

[4] K. Ohkuma, H. Muratani, F. Sano, and S. Kawamura. Specification and assessment of the cipher hierocrypt. *Technical Report of IEICE(Japan) ISEC2000-7*, 2000.

Appendix

A Algorithm of Hierocrypt–L1

The encryption algorithm of Hierocrypt–L1 is presented in this appendix, which consists of "Notations"(A.1), "Structure"(A.2), and "Fundamental Operations"(A.2). In A.2, operations for both Hierocrypt–L1 and Hierocrypt–3 are presented.

A.1 Notations

We follow the notation of Hierocrypt–3 to express an n-bit value. The following shows an example of concatenation expression of a (64)-bit value $X_{(64)}$.

$$X_{(64)} = X_{1(32)} \| X_{2(32)} \ ,$$
$$X_{i(32)} = x_{4i-4+1(8)} \| x_{4i-4+2(8)} \| \cdots \| x_{4i-4+4(8)} \ , \quad i = 1, 2,$$
$$X_{j(8)} = x_{8j-8+1(1)} \| x_{8j-8+2(1)} \| \cdots \| x_{8j-8+8(1)} \ , \quad j = 1, 2, \ldots, 8.$$

Note that the LSB of the value $X_{i(n)}$ (i=1,2,...,m) is $x_{in(1)}$, which is the in-th MSB of $X_{(mn)}$.

A.2 Structure

The structures of data randomization part and the key scheduling part are described in this section. Fundamental operations used there are described in the next section.

A.2.1 Encryption

The 6-round encryption of Hierocrypt–L1 consists of 5 operations of round function ρ, an operation of XS-function, and the final key addition (AK) (See Figure 1).

$$P_{(64)} \equiv X^{(0)}_{(64)} \overset{\rho}{\longmapsto} X^{(1)}_{(64)} \overset{\rho}{\longmapsto} \cdots$$
$$\cdots \overset{\rho}{\longmapsto} X^{(5)}_{(64)} \overset{XS}{\longmapsto} X^{(6)}_{(64)} \overset{AK}{\longmapsto} C_{(64)}$$

Figure 1: Encryption of Hierocrypt–L1

The number of rounds is 6.
The 128-bit value $X^{(i)}_{(64)}$ is the output of the i-th operation of round function ρ ($i = 1, 2, \ldots, 5$). The plaintext $P_{(64)}$ is assigned to the 0-th value $X^{(0)}_{(64)}$.
The value $X^{(t)}_{(64)}$ is the output of the t-th operation of ρ-function for the input $X^{(t-1)}_{(64)}$ and the round key $K^{(t)}_{(128)}$.

$$X^{(t)}_{(64)} = \rho(X^{(t-1)}_{(64)}, K^{(t)}_{(128)}), \quad t = 1, 2, \cdots, 5.$$

Similarly, $X^{(6)}_{(64)}$ is the output of XS-function for the input $X^{(5)}_{(64)}$ and the final key $K^{(6)}_{(128)}$.

$$X^{(6)}_{(64)} = XS(X^{(5)}_{(64)}, K^{(6)}_{(128)}) \ .$$

The ciphertext $C_{(64)}$ is given as the addition (XOR, exclusive or) between the 6th round output $X^{(6)}_{(64)}$ and the first half of the final key $K^{(7)}_{1(64)}$

$$C_{(64)} = X^{(6)}_{(64)} \oplus (K^{(7)}_{1(32)} \| K^{(7)}_{2(32)}) \ .$$

A.2.2 decryption

The decryption of Hierocrypt–L1 is the inverse of encryption, and consists of the final key addition, the inverse of XS-function (XS^{-1}), and 5 inverse operations of round function (ρ^{-1}).

$$X^{(6)}_{(64)} = C_{(64)} \oplus (K^{(7)}_{1(32)} \| K^{(7)}_{2(32)}) \ ,$$
$$X^{(5)}_{(64)} = XS^{-1}(X^{(6)}_{(64)}, K^{(6)}_{(128)}) \ ,$$
$$X^{(t-1)}_{(64)} = \rho^{-1}(X^{(t)}_{(64)}, K^{(t)}_{(128)}) \ , \ t = 5, \ldots, 2, 1.$$

The plaintext $P_{(64)}$ is given as the final output $X^{(0)}_{(64)}$.

$$P_{(64)} = X^{(0)}_{(64)} \ .$$

$$K_{(128)} \overset{\text{padding}}{\longmapsto} Z^{(-1)}_{(128)} \overset{\sigma_0}{\longmapsto} Z^{(0)}_{(128)} \overset{\sigma}{\longmapsto} Z^{(1)}_{(128)}$$
$$\overset{\sigma}{\longmapsto} Z^{(2)}_{(128)} \overset{\sigma}{\longmapsto} \cdots \overset{\sigma}{\longmapsto} Z^{(4)}_{(128)}$$
$$\overset{\sigma^{-1}}{\longmapsto} Z^{(5)}_{(128)} \overset{\sigma^{-1}}{\longmapsto} \cdots \overset{\sigma^{-1}}{\longmapsto} Z^{(7)}_{(128)}$$

Figure 2: Intermediate key scheduling

A.2.3 Key scheduling

The main part of key scheduling consists of the intermediate key generation part and the round key generation part, preceded by the intermediate key initialization. The intermediate key part recursively generates intermediate key outputs $Z^{(t)}_{(128)}$ ($t = 1, 2, \ldots, 7$), and the round key generation part gerates round keys $K^{(t)}_{(128)}$ ($t = 1, 2, \ldots, 7$) from the corresponding intermediate keys. The intermediate key $Z^{(t)}_{(128)}$ and the round key $K^{(t)}_{(128)}$ are divided into 4 pieces.

$$Z^{(t)}_{(128)} = Z^{(t)}_{1(32)} \| Z^{(t)}_{2(32)} \| Z^{(t)}_{3(32)} \| Z^{(t)}_{4(32)} \ ,$$

$$K^{(t)}_{(128)} = K^{(t)}_{1(32)} \| K^{(t)}_{2(32)} \| K^{(t)}_{3(32)} \| K^{(t)}_{4(32)} \ .$$

To generate the intermediate keys, the σ-function is used for $5 \le t \le 7$, and the σ^{-1}-function is used for $5 \le t \le 7$. Under the recursion rule, the intermediate key values are symmetric with regard to the point $t = 4$.

$$Z^{(t)}_{(128)} = Z^{(8-t)}_{(128)} \ , \quad 5 \le t \le 7 \ .$$

round-dependent constants To prevent periodic patterns from appearing in the intermediate key generation, and to improve resistance against the related key attack, we introduce round-dependent key additions to the intermediate key generation part. The round-dependent keys have been made by combining two from the four 32-bit values which are given as binary expansions of irrational numbers.

$$\begin{aligned} H_0 &= \text{0x5A827999} = trunc(\sqrt{2}/4), \\ H_1 &= \text{0x6ED9EBA1} = trunc(\sqrt{3}/4), \\ H_2 &= \text{0x8F1BBCDC} = trunc(\sqrt{5}/4), \\ H_3 &= \text{0xCA62C1D6} = trunc(\sqrt{10}/4), \\ H_4 &= \text{0xF7DEF58A} = trunc(\sqrt{15}/4), \end{aligned}$$

$$trunc(x) = \left\lfloor 2^{32} x \right\rfloor \ ,$$

Preprocessing The intermediate key $Z^{(-1)}_{(128)}$ is made of the encryption key $K_{(\text{length})}$ (length = 128, 192, 256) with an initial operation. The intermediate key $Z^{(0)}_{(128)}$ is deriven from $Z^{(-1)}_{(128)}$ through the pre-whitening operation σ_0. The padding operation is done when length = 192, 256 where padded values are concatenations of the above mentioned 32-bit constants H_i. The padding operations are described as follows.

[128-bit key]

$$K_{1(32)} \| K_{2(32)} = K_{(64)} \ ,$$
$$Z^{(-1)}_{1(32)} = K_{1(32)} \ , \quad Z^{(-1)}_{2(32)} = K_{2(32)} \ ,$$
$$Z^{(-1)}_{3(32)} = K_{1(32)} \ , \quad Z^{(-1)}_{4(32)} = H_3 \| H_2 \ .$$

[192-bit key]

$$K_{1(32)} \| K_{2(32)} \| K_{3(32)} = K_{(192)} \ ,$$
$$Z^{(-1)}_{1(32)} = K_{1(32)} \ , \quad Z^{(-1)}_{2(32)} = K_{2(32)} \ ,$$
$$Z^{(-1)}_{3(32)} = K_{3(32)} \ , \quad Z^{(-1)}_{4(32)} = H_2 \| H_3 \ .$$

[256-bit key]

$$K_{1(32)} \| K_{2(32)} \| K_{3(32)} \| K_{4(32)} = K_{(256)} \ ,$$
$$Z^{(-1)}_{1(32)} = K_{1(32)} \ , \quad Z^{(-1)}_{2(32)} = K_{2(32)} \ ,$$
$$Z^{(-1)}_{3(32)} = K_{3(32)} \ , \quad Z^{(-1)}_{4(32)} = K_{4(32)} \ .$$

[pre-whitening](ρ_0)

The pre-whitening is done, before iterative operation by the σ-function. The pre-whitening operation ρ_0 is made from ρ by removing $P^{(16)}$.

$$
\begin{aligned}
Z^{(0)}_{(128)} &= \sigma_0(Z^{(-1)}_{(128)}, G^{(0)}_{(32)}),\\
Z^{(0)}_{3(32)} &= M_{5E}(Z^{(-1)}_{3(32)}) \oplus G^{(0)}_{(32)} ,\\
Z^{(0)}_{4(32)} &= M_{5E}(Z^{(-1)}_{4(32)}) ,\\
Z^{(0)}_{1(32)} &= Z^{(-1)}_{2(32)} ,\\
Z^{(0)}_{2(32)} &= Z^{(-1)}_{1(32)} \oplus F_\sigma(Z^{(-1)}_{2(32)} \oplus Z^{(0)}_{3(32)}) .
\end{aligned}
$$

As the round-dependent constant $G^{(0)}_{(32)}$, the following 64-bit concatenated value is used.

$$G^{(0)}_{(32)} = G_0(5) = H_1 \| H_0 .$$

Round function for intermediate key (σ-function) The intermediate key $Z^{(t)}_{(128)}$ is generated by the operation σ up to $t = 4$, and afterwards by the inverse operation σ^{-1}. The sequence of intermediate keys is symmetric with respect to the point $t = 4$ for this round-trip-type scheduling.

$$Z^{(t)}_{(128)} = Z^{(8-t)}_{(128)} , \quad 4 \le t \le 7 .$$

We call the region: $(1 \le t \le 4)$ as the plaintext side, and the other region: $(5 \le t \le 7)$ as the ciphertext side, corresponding to the position in the data randomizing part.

[Iteration of intermediate key(plaintext side)] $(1 \le t \le 4)$

$$
\begin{aligned}
Z^{(t)}_{(128)} &= \sigma(Z^{(t-1)}_{(128)} , \ G^{(t)}_{(32)}) ,\\
W^{(t-1)}_{1(32)} \| W^{(t-1)}_{2(32)} &= P^{(16)}(Z^{(t-1)}_{3(32)} \| Z^{(t-1)}_{4(32)}) ,\\
Z^{(t)}_{3(32)} &= M_{5E}(W^{(t-1)}_{1(32)}) \oplus G^{(t)}_{(32)} ,\\
Z^{(t)}_{4(32)} &= M_{5E}(W^{(t-1)}_{2(32)}) ,\\
Z^{(t)}_{1(32)} &= Z^{(t-1)}_{2(32)} ,\\
Z^{(t)}_{2(32)} &= Z^{(t-1)}_{1(32)} \oplus F_\sigma(Z^{(t-1)}_{2(32)} \oplus Z^{(t)}_{3(32)}) .
\end{aligned}
$$

[Iteration of intermediate key(ciphertext side)] $(5 \le t \le 7)$

$$
\begin{aligned}
Z^{(t)}_{(128)} &= \sigma^{-1}(Z^{(t-1)}_{(128)}, G^{(t-1)}_{(32)}) ,\\
Z^{(t)}_{1(32)} &= Z^{(t-1)}_{2(32)} \oplus F_\sigma(Z^{(t-1)}_{1(32)} \oplus Z^{(t-1)}_{3(32)}) ,\\
Z^{(t)}_{2(32)} &= Z^{(t-1)}_{1(32)} ,\\
W^{(t)}_{1(32)} &= M_{B3}(Z^{(t-1)}_{3(32)} \oplus G^{(t-1)}_{(32)}) ,\\
W^{(t)}_{2(32)} &= M_{B3}(Z^{(t-1)}_{4(32)}) ,\\
Z^{(t)}_{3(32)} \| Z^{(t)}_{4(32)} &= P^{(32)^{-1}}(W^{(t)}_{1(32)} \| W^{(t)}_{2(32)}) .
\end{aligned}
$$

Round key generation The different rules are applied to generate a round key from the corresponding intermediate key for the plaintext side and the ciphertext side.

[Round key generation(plaintext side)] $(1 \le t \le 4)$

$$
\begin{aligned}
V^{(t)}_{(32)} &= F_\sigma(Z^{(t-1)}_{2(32)} \oplus Z^{(t)}_{3(32)}) ,\\
K^{(t)}_{1(32)} &= Z^{(t-1)}_{1(32)} \oplus V^{(t)}_{(32)} ,\\
K^{(t)}_{2(32)} &= Z^{(t)}_{3(32)} \oplus V^{(t)}_{(32)} ,\\
K^{(t)}_{3(32)} &= Z^{(t)}_{4(32)} \oplus V^{(t)}_{(32)} ,\\
K^{(t)}_{4(32)} &= Z^{(t-1)}_{2(32)} \oplus Z^{(t)}_{4(32)} .
\end{aligned}
$$

[Round key generation(ciphertext side)] $(5 \le t \le 7)$

$$
\begin{aligned}
V^{(t)}_{(32)} &= F_\sigma(Z^{(t-1)}_{1(32)} \oplus Z^{(t-1)}_{3(32)}) ,\\
K^{(t)}_{1(32)} &= Z^{(t)}_{1(32)} \oplus Z^{(t-1)}_{3(32)} ,\\
K^{(t)}_{2(32)} &= W^{(t)}_{1(32)} \oplus V^{(t)}_{(32)} ,\\
K^{(t)}_{3(32)} &= W^{(t)}_{2(32)} \oplus V^{(t)}_{(32)} ,\\
K^{(t)}_{4(32)} &= Z^{(t-1)}_{1(32)} \oplus W^{(t)}_{2(32)} .
\end{aligned}
$$

Table 3: Key schedule of Hierocrypt–L1

round key	t	operation	$G^{(t)}_{(32)}$
–	0 (PW)	σ_0	H_0
$K^{(1)}_{(128)}$	1	σ	H_1
$K^{(2)}_{(128)}$	2	σ	H_2
$K^{(3)}_{(128)}$	3	σ	H_3
$K^{(4)}_{(128)}$	4	σ	H_4
$K^{(5)}_{(128)}$	5	σ^{-1}	H_4
$K^{(6)}_{(128)}$	6	σ^{-1}	H_3
$K^{(7)}_{(128)}$	7	σ^{-1}	H_2

A.3 Fundamental Operations

Fundamental operations for both Hierocrypt–L1(HC-L1, for short) and Hierocrypt–3(HC-3, for short) are presented.

Round functions for HC-L1 and HC-3 (ρ) The ρ-function, which is the round function of the data randomization part, is a composite function of the XS-function and the $MDS_{\rm L}$-function.

[HC-L1]: The input data are the $X_{(64)}$ and the $K_{(128)}$.

$$\rho\,(X_{(64)}, K_{(128)}) = MDS_{\rm H}\,(XS\,(X_{(64)}, K_{(128)}))\ .$$

[HC-3]: The input data are the 128-bit value $X_{(128)}$ and the 256-bit value $K_{(256)}$.

$$\rho\,(X_{(128)}, K_{(256)}) = MDS_{\rm H}\,(XS\,(X_{(128)}, K_{(256)}))\ .$$

$MDS_{\rm H}$-functions for HC-L1 and HC-3 The $MDS_{\rm H}$-function is a linear transformation consisting of exclusive or's between 8-bit subdata $x_{i(8)}$ $(\in {\rm GF}(2)^8)$, where $i = 1, 2, \ldots, 16$ for HC-L1; and $i = 1, 2, \ldots, 8$ for HC-3. $MDS_{\rm H}$ is represented by the following matrix form.

[HC-L1]

$$Y_{(64)} = MDS_{\rm H}(X_{(64)}) ,$$

$$
\begin{pmatrix} y_{1(8)}\\ y_{2(8)}\\ y_{3(8)}\\ y_{4(8)}\\ y_{5(8)}\\ y_{6(8)}\\ y_{7(8)}\\ y_{8(8)} \end{pmatrix}
=
\begin{pmatrix} 10101110\\ 11011111\\ 11100111\\ 01011101\\ 11010101\\ 11101010\\ 11111101\\ 10101011 \end{pmatrix}
\begin{pmatrix} x_{1(8)}\\ x_{2(8)}\\ x_{3(8)}\\ x_{4(8)}\\ x_{5(8)}\\ x_{6(8)}\\ x_{7(8)}\\ x_{8(8)} \end{pmatrix} .
$$

[HC-3]

$$Y_{(128)} = MDS_{\rm H}(X_{(128)}) ,$$

$$
\begin{pmatrix} y_{1(8)}\\ y_{2(8)}\\ y_{3(8)}\\ y_{4(8)}\\ y_{5(8)}\\ y_{6(8)}\\ y_{7(8)}\\ y_{8(8)}\\ y_{9(8)}\\ y_{10(8)}\\ y_{11(8)}\\ y_{12(8)}\\ y_{13(8)}\\ y_{14(8)}\\ y_{15(8)}\\ y_{16(8)} \end{pmatrix}
=
\begin{pmatrix}
1010101011011111\\ 1101110111001111\\ 1110111011111011\\ 0101010110101110\\ 1111101010101101\\ 0111111011011110\\ 0011111011101111\\ 1110010101011010\\ 1101111110101010\\ 1110011111011101\\ 1111001111101110\\ 1010111001010101\\ 1010111011111010\\ 1101111001111101\\ 1110111100111110\\ 0101101011100101
\end{pmatrix}
\begin{pmatrix} x_{1(8)}\\ x_{2(8)}\\ x_{3(8)}\\ x_{4(8)}\\ x_{5(8)}\\ x_{6(8)}\\ x_{7(8)}\\ x_{8(8)}\\ x_{9(8)}\\ x_{10(8)}\\ x_{11(8)}\\ x_{12(8)}\\ x_{13(8)}\\ x_{14(8)}\\ x_{15(8)}\\ x_{16(8)} \end{pmatrix} .
$$

XS-functions for HC-L1 and HC-3 XS-function is a composite function of the S-function, key addition, and $MDS_{\rm L}$-function.

$$XS(X_{(bl)}, K_{(bld)}) = S(MDS_{\rm L}(S(X_{(bl)} \oplus K_{1(bl)})) \oplus K_{2(bl)}) ,$$

$$\begin{cases} bl = 64, \ bld = 128, & \text{for HC-L1,}\\ bl = 128, \ bld = 256, & \text{for HC-3.} \end{cases}$$

S-functions for HC-L1 and HC-3 The S-functions consists of parallel operations of s-function.

$$Y_{(bl)} = S\,(X_{(bl)}) \ , \quad y_{i(8)} = s\,(x_{i(8)}) \ ,$$

$$\begin{cases} bl = 64, & i = 1, 2, \ldots, 8 \text{ for HC-L1,}\\ bl = 128, & i = 1, 2, \ldots, 16 \text{ for HC-3.} \end{cases}$$

MDS_{L}-functions for HC-L1 and HC-3 The MDS_{L}-function consists of parallel operations of mds_{L}-function for 32-bit subdata.

$$Y_{(bl)} = MDS_{\mathrm{L}}\left(X_{(bl)}\right) \ , \quad y_{i(32)} = mds_{\mathrm{L}}\left(x_{i(32)}\right) \ ,$$

$$\begin{cases} bl = 64, & i = 1, 2, \quad \text{for HC-L1,} \\ bl = 128, & i = 1, 2, 3, 4, \quad \text{for HC-3.} \end{cases}$$

s-function for both HC-L1 and HC-3 The s-function is a nonlinear transformation for 8-bit input/output value, which is given as the following table where all numbers are represented in hexadecimal.

$s(256) = ($
$(s(0)\ s(1)\ s(2)\ \ldots\ s(\mathrm{F})\ s(10)\ s(11)\ \ldots\ s(\mathrm{FF})) =$
(07 FC 55 70 98 8E 84 4E BC 75 CE 18 02 E9 5D 80
1C 60 78 42 9D 2E F5 E8 C6 7A 2F A4 B2 5F 19 87
0B 9B 9C D3 C3 77 3D 6F B9 2D 4D F7 8C A7 AC 17
3C 5A 41 C9 29 ED DE 27 69 30 72 A8 95 3E F9 D8
21 8B 44 D7 11 0D 48 FD 6A 01 57 E5 BD 85 EC 1E
37 9F B5 9A 7C 09 F1 B1 94 81 82 08 FB C0 51 0F
61 7F 1A 56 96 13 C1 67 99 03 5E B6 CA FA 9E DF
D6 83 CC A2 12 23 B7 65 D0 39 7D 3B D5 B0 AF 1F
06 C8 34 C5 1B 79 4B 66 BF 88 4A C4 EF 58 3F 0A
2C 73 D1 F8 6B E6 20 B8 22 43 B3 33 E7 F0 71 7E
52 89 47 63 0E 6D E3 BE 59 64 EE F6 38 5C F4 5B
49 D4 E0 F3 BB 54 26 2B 00 86 90 FF FE A6 7B 05
AD 68 A1 10 EB C7 E2 F2 46 8A 6C 14 6E CF 35 45
50 D2 92 74 93 E1 DA AE A9 53 E4 40 CD BA 97 A3
91 31 25 76 36 32 28 3A 24 4C DB D9 8D DC 62 2A
EA 15 DD C2 A5 0C 04 1D 8F CB B4 4F 16 AB AA A0) .

mds_{L}-function for HC-L1 and HC-3 The mds_{L}-function is a linear transformation which is represented by 4×4 matrix multiplication where all matrix and vector elements are regarded as elements of $\mathrm{GF}(2^8)$.

$$Y_{i(32)} = mds_{\mathrm{L}}(X_{i(32)}) \ ,$$

$$\begin{pmatrix} y_{4i-4+1(8)} \\ y_{4i-4+2(8)} \\ y_{4i-4+3(8)} \\ y_{4i-4+4(8)} \end{pmatrix} = \begin{pmatrix} \mathrm{C4} & 65 & \mathrm{C8} & 8\mathrm{B} \\ 8\mathrm{B} & \mathrm{C4} & 65 & \mathrm{C8} \\ \mathrm{C8} & 8\mathrm{B} & \mathrm{C4} & 65 \\ 65 & \mathrm{C8} & 8\mathrm{B} & \mathrm{C4} \end{pmatrix} \begin{pmatrix} x_{4i-4+1(8)} \\ x_{4i-4+2(8)} \\ x_{4i-4+3(8)} \\ x_{4i-4+4(8)} \end{pmatrix} .$$

Here, 8-bit data $x_{(8)}$ and the matrix element a (in hexadecimal) are regarded as elements of $\mathrm{GF}(2^8)$ related as follows.

$$x_{(8)} \Leftrightarrow \sum_{i=1}^{8} x_{i(1)} z^{8-i} \ , \quad a = \sum_{i=0}^{7} a_i 2^i \Leftrightarrow \sum_{i=0}^{7} a_i z^i \ .$$

The polynomial $p(z) = z^8 + z^6 + z^5 + z + 1$ is used as the primitive polynomial for the Galois field $\mathrm{GF}(2^8)$.

$P^{(n)}$-function for HC-L1 and HC-3 The $P^{(n)}$ function consists of the a linear transformation for the input $X_{(4n)}$ which is a concatenation of four n-bit values $x_{i(n)}$ $(i = 1, 2, 3, 4)$ where each element is regarded as an element of $\mathrm{GF}(2)^n$.

$$Y_{(4n)} = P^{(n)}\left(X_{(4n)}\right) \ ,$$

$$X_{(4n)} = x_{1(n)} \| x_{2(n)} \| x_{3(n)} \| x_{4(n)} \ ,$$

$$Y_{(4n)} = y_{1(n)} \| y_{2(n)} \| y_{3(n)} \| y_{4(n)} \ ,$$

$$\begin{pmatrix} y_{1(n)} \\ y_{2(n)} \\ y_{3(n)} \\ y_{4(n)} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_{1(n)} \\ x_{2(n)} \\ x_{3(n)} \\ x_{4(n)} \end{pmatrix} .$$

The inverse function $P^{(n)-1}$, is given by the following equation.

$$X_{(4n)} = P^{(n)-1}\left(Y_{(4n)}\right) \ ,$$

$$\begin{pmatrix} x_{1(n)} \\ x_{2(n)} \\ x_{3(n)} \\ x_{4(n)} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} y_{1(n)} \\ y_{2(n)} \\ y_{3(n)} \\ y_{4(n)} \end{pmatrix} .$$

M_5-function for HC-L1 The M_5-function consists of a 32-bit linear transformations, where each 8-bit subdata is regarded as an element of $\mathrm{GF}(2)^8$.

$$Y_{(32)} = M_5\left(X_{(32)}\right) \ ,$$

$$\begin{pmatrix} y_{1(8)} \\ y_{2(8)} \\ y_{3(8)} \\ y_{4(8)} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{1(8)} \\ x_{2(8)} \\ x_{3(8)} \\ x_{4(8)} \end{pmatrix} .$$

M_{B}-function for HC-L1 The M_{B}-function consists of a 32-bit linear transformations, where each 8-bit subdata is regarded as an element of $\mathrm{GF}(2)^8$.

$$Y_{(32)} = M_{\mathrm{B}}\left(X_{(32)}\right) \ ,$$

$$\begin{pmatrix} y_{1(8)} \\ y_{2(8)} \\ y_{3(8)} \\ y_{4(8)} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_{1(8)} \\ x_{2(8)} \\ x_{3(8)} \\ x_{4(8)} \end{pmatrix} .$$

$M_{5\mathrm{E}}$-function for HC-3 The $M_{5\mathrm{E}}$-function consists of a concatenation of two 32-bit linear transformations, where each 8-bit subdata is regarded as an element of $\mathrm{GF}(2)^8$.

$$Y_{(32)} = M_{5\mathrm{E}}\left(X_{(32)}\right) \ ,$$

$$\begin{pmatrix} y_{1(8)} \\ y_{2(8)} \\ y_{3(8)} \\ y_{4(8)} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{1(8)} \\ x_{2(8)} \\ x_{3(8)} \\ x_{4(8)} \end{pmatrix} ,$$

$$\begin{pmatrix} y_{5(8)} \\ y_{6(8)} \\ y_{7(8)} \\ y_{8(8)} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} x_{5(8)} \\ x_{6(8)} \\ x_{7(8)} \\ x_{8(8)} \end{pmatrix} .$$

M_{B3}-function for HC-3 The M_{BE}-function consists of a concatenation of two 32-bit linear transformations, where each 8-bit subdata is regarded as an element of $\mathrm{GF}(2)^8$.

$$Y_{(32)} = M_{\mathrm{B3}}\left(X_{(32)}\right) \ ,$$

$$\begin{pmatrix} y_{1(8)} \\ y_{2(8)} \\ y_{3(8)} \\ y_{4(8)} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_{1(8)} \\ x_{2(8)} \\ x_{3(8)} \\ x_{4(8)} \end{pmatrix} ,$$

$$\begin{pmatrix} y_{5(8)} \\ y_{6(8)} \\ y_{7(8)} \\ y_{8(8)} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{5(8)} \\ x_{6(8)} \\ x_{7(8)} \\ x_{8(8)} \end{pmatrix} .$$

F_σ-functions for HC-L1 and HC-3 The F_σ-function is a nonlinear function which consists of the s-functions and the $P^{(n)}$-functions.

$$Y_{(blh)} = F_\sigma\left(X_{(blh)}\right) \ ,$$

$$u_{i(8)} = s\left(x_{i(8)}\right) \ , \quad Y_{(blh)} = P^{(blo)}\left(U_{(blh)}\right) \ ,$$

$$\begin{cases} blh = 64, & blo = 16, \quad \text{for HC-L1,} \\ blh = 128, & blo = 32, \quad \text{for HC-3.} \end{cases}$$