Chapter 3

HIEROCRYPT-L1

HIEROCRYPT-L1 was designed by Toshiba in 2000. The design paradigm was published at the Selected Areas on Cryptography (SAC) 2000 international workshop. HIEROCRYPT-L1 was also submitted to the NESSIE European development process.[1]

3.1 Design Properties

3.1.1 Normative Issues

First of all, we must notify that the reference part of the document delivered by IPA (Sec. 3 of the specifications) has more than 200 typos in total. Since it is impossible to report all of them we should on a more general level note that

1. HIEROCRYPT-L1 is a block cipher with 64-bit blocks and quantities like $X^{(t)}$ are 64-bit long whereas quantities like $Z^{(t)}$ and $K^{(t)}$ are 128-bit long;

2. the key lengths are 64, 96 or 128;

3. M_{5E} and M_{B3} are never defined;

4. at some places, ρ should read σ;

5. HIEROCRYPT-3 is out of the scope of HIEROCRYPT-L1.

We have been able to understand how HIEROCRYPT-L1 is defined by cross reading the document delivered by IPA, the document submitted to NESSIE,

[1] See `http://www.cryptonessie.org/`.

and the paper published in SAC2000. (The two later documents made similar mistakes though.) However, it was not possible to implement a full working version of the cipher in a reasonable amount of time using the given documentation.

3.1.2 Basic Properties

The secret key is expanded into a 128-bit subkeys sequence $K^{(1)}, \ldots, K^{(7)}$ by a key scheduling which can be preprocessed. Then, encryption is a cascade of

- XOR with a subkey half,
- eight parallel application of a substitution S defined by a table look-up,
- a linear transformation.

The linear transformation is alternatively a complicated one MDS_L defined with $\mathrm{GF}(2^8)$ elements, and a simple one MDS_H defined by bitwise XOR on bytes. Two iterations (XOR, substitution, MDS_L, XOR, substitution, MDS_H) are called a round. We have 6 rounds in total but the final MDS_H is replaced by a final XOR. This makes a cascade of 12 substitution and linear layers.

This simple design is inherited from Shannon in the mid-XXth Century. It is known to be a good design which should converge towards a perfect cipher. The minimal number of rounds for security is not quite clear though.

3.1.3 Substitution Boxes

The substitution box S is constructed as

$$S(x) = (\mathrm{Perm}(x))^{247} \oplus 7$$

where the power function defers to the $\mathrm{GF}(2^8)$ structure. This finite field has an interesting property: the Frobenius function which is linear and which commutes with the power function. We thus have

$$S(x)^2 = (\mathrm{Perm}(x)^2)^{247} \oplus 7^2.$$

Now we notice that Perm is a linear function. Thus if we let L be defined by

$$L(x) = \mathrm{Perm}^{-1}((\mathrm{Perm}(x)^2))$$

we have

$$S(x)^2 = S(L(x)) \oplus 7^2 \oplus 7.$$

Since $7^2 \oplus 7 = \texttt{0x12}$ we have

$$S(x)^2 = S(L(x)) \oplus \texttt{0x12}.$$

where L is given by

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

We can however outline that if S did not have the Perm and $\oplus 7$ functions, then the Frobenius operator would propagate through the whole circuit and we would have an interesting related key attack.

3.1.4 Implementation Viewpoint

From a programmer viewpoint, the HIEROCRYPT-L1 encryption is a circuit with bitwise XOR on bytes, tables lookups on bytes, and multiplications in $\mathrm{GF}(2^8)$. The multiplication is the only complicated part, but it can also be efficiently defined by table lookups. From the definition of MDS_L, we can see we only need multiplications by `C4`, `65`, `C8` or `8B` for the encryption process and `89`, `8E`, `A9` or `49` for the decryption process. We need hence nine 256-byte tables (one of them being the S-box), which is a total of 2.2KB table. Implementation is thus simply a circuit with table lookups and bitwise XORs.

3.1.5 Key Schedule

The subkey sequence is generated from a sequence

$$Z^0, Z^1, Z^2, Z^3, Z^4, Z^3, Z^2, Z^1.$$

Each subkey is generated from two consecutive terms:

- K^1 is generated from Z^0, Z^1,

- K^2 is generated from Z^1, Z^2,
- K^3 is generated from Z^2, Z^3,
- K^4 is generated from Z^3, Z^4,
- K^5 is generated from Z^4, Z^3,
- K^6 is generated from Z^3, Z^2,
- K^7 is generated from Z^2, Z^1.

Although we found it strange, we did not see any security problem due to this scheme.

3.2 Differential and Linear Cryptanalysis

Assuming we want to perform a regular differential or linear cryptanalysis, regular techniques leads to focusing on 5 rounds instead of 6. We can thus consider a cascade of 10 substitution and linear transformations. Since S has a maximal differential or linear probability of 2^{-6}, and the maximal usable probability is 2^{-64}, we must find a characteristic with at most 10.66 active S boxes. Considering the cascade of 10 substitution and linear transformations, this gives an average of less than 1.1 active S box per layer, which is not feasible. Cumulative effect of characteristics may however hide surprises.

Although we were not able to perform a deep security analysis, we think that six rounds (thus a cascade of twelve layers of substitution and linear transformations) are not enough.

3.3 Other Attacks

3.3.1 Truncated Differentials

As done in Section 1.3.1, we have investigated the truncated differentials for HIEROCRYPT-L1 using the same graph-oriented approach.

In the case of HIEROCRYPT-L1, such a computation gives as result a minimal non-zero weight of 21 for a path of length 6. Thus, we can conclude that there exists no useful truncated differential based on this structure.

3.3.2 Side Channel Cryptanalysis

Like for other block ciphers, assuming that we can trace the Hamming weight of CPU registers throughout the computation process, we can easily break HIEROCRYPT-L1 by power analysis. The design of HIEROCRYPT-L1 does not seem to offer much more potential weaknesses to side channel cryptanalysis, but with the table lookups. As for other block ciphers with table lookups, assuming that we can speed up the clock signal in order to make the tables invisible or that we can tamper the tables in memory, fault analysis may be able to break the cipher. We thus recommend that implementations care about power analysis and on the memory tempering attacks.

3.3.3 Decorrelation

This kind of design usually offers less decorrelation than others because of the limited size of internal random functions. When internal random function are over n-bit strings, we can expect a decorrelation to the order $2^{\frac{n}{2}}$. Here S is the only function which can be expected to be random, so we can expect at least a decorrelation to the order up to $2^4 = 16$.

Despite this limitation, the decorrelation of HIEROCRYPT-L1 may be limited by the structure of S: actually, S is distinguishable from a random function by only two chosen plaintext due to the property

$$S(x)^2 = S(L(x)) \oplus \texttt{0x12}.$$

Although the general design can provide pseudorandomness, this result cannot be applied here because of this property in S.

3.4 Available Literature

HIEROCRYPT-L1 specifications [8] and their publication at SAC'00 [9] are the only public documents we are aware of. We were not able to find any public review or cryptanalysis on a member of the HIEROCRYPT family.

3.5 Conclusion

HIEROCRYPT-L1 is a recent block cipher with conservative design. The original document is not mature enough: implementation is just impossible from this reference and security analysis was quite complicated in this situation. The general design was submitted to the academic world in an international scientific workshop, which may validate the original paradigm.

Besides, some strange internal properties were found, and it is note quite sure that a cascade of twelve substitutions and linear transformations is enough for security.

Here are our conclusion about HIEROCRYPT-L1.

1. **Discovery of unexpected internal properties: "−".** There is linear transformation which can go through S.
2. **Randomness provided by the key schedule: "+".** The key schedule seems to provide good randomness.
3. **Resistance against differential and linear cryptanalysis: "·".** Although traditional cryptanalysis seems to be impossible, we think that generalized version may still be feasible due to the low number of rounds.
4. **Resistance against side channel attacks: "+".** Resistance is quite standard: we only have bitwise operations and tables.
5. **Maturity of the algorithm: "−−".** The provided document makes implementation impossible. Extra analysis would be required.
6. **Overall security confidence: "−".** Simplicity and low number of rounds may hide important weaknesses.
7. **Beauty of the design: "·".** The design is quite conservative.