Improved SQUARE Attacks against Reduced-Round HIEROCRYPT

Paulo S.L.M. Barreto[1], Vincent Rijmen[2⋆], Jorge Nakahara Jr.[2⋆⋆], Bart Preneel[2], Joos Vandewalle[2], and Hae Y. Kim[3]

[1] Scopus Tecnologia S. A.
Av. Mutinga, 4105 - Pirituba
BR–05110–000 São Paulo (SP), Brazil
pbarreto@scopus.com.br
[2] Katholieke Universiteit Leuven, Dept. ESAT,
Kasteelpark Arenberg 10,
B–3001 Leuven-Heverlee, Belgium
{vincent.rijmen,jorge.nakahara,bart.preneel,
joos.vandewalle}@esat.kuleuven.ac.be
[3] Universidade de São Paulo, Departamento de Engenharia Eletrônica.
Av. Prof. Luciano Gualberto, tr. 3, 158,
BR–05508-900, São Paulo (SP), Brazil.
hae@lps.usp.br

Abstract. We present improved SQUARE attacks against the NESSIE and ECTP candidate block ciphers HIEROCRYPT-3 and HIEROCRYPT-L1, designed by Toshiba. We improve over the previous best known attack on five S-box layers of HIEROCRYPT-3 by a factor of 2^{128} computational steps with an attack on six layers for 128-bit keys, and extend it to seven S-box layers for longer keys. For HIEROCRYPT-L1 we are able to improve previous attacks up to seven S-box layers (out of twelve).

1 Introduction

The HIEROCRYPT ciphers are candidate block ciphers for the NESSIE Project [8] and (ECTP) *Evaluation of Cryptographic Techniques Project* [3]. NESSIE is a project within the Information Societies Technology (IST) Programme of the European Commission (Key Action II, Action Line II.4.1). ECTP is part of the *MITI Action Plan for a Secure E-Government* announced by the Ministry of International Trade and Industry (MITI) of Japan in April 2000.

SQUARE attacks have first been described in [1]. Variations have been developed for related ciphers [10,2], and also for ciphers of quite distinct nature (as in [6], where the technique is called a 'saturation attack').

The designers of HIEROCRYPT-3 and HIEROCRYPT-L1 describe SQUARE-like attacks on reduced-round variants of the ciphers on the submission document.

⋆ F.W.O. Postdoctoral Researcher, sponsored by the Fund for Scientific Research – Flanders (Belgium)
⋆⋆ sponsored in part by GOA project Mefisto 2000/06

M. Matsui (Ed.): FSE 2001, LNCS 2355, pp. 165–173, 2002.

The best attacks against HIEROCRYPT-3 [11, p. 8, Table 5] breaks four S-box layers (2 rounds) at the cost of 2^{40} key guesses using 2^{11} chosen plaintexts, and five S-box layers (2.5 rounds) at the cost of 2^{168} key guesses using 2^{13} chosen plaintexts. The best attack against HIEROCRYPT-L1 [12, p. 8, Table 4] breaks five S-box layers (2.5 rounds) at the cost of 2^{72} key guesses using 2^{32} chosen plaintexts. In this paper, we present more effective attacks on reduced-round variants of the HIEROCRYPT ciphers.

This paper is organised as follows. In Sect. 2 we review the structure of the HIEROCRYPT ciphers. Sect. 3 describes the principles behind the SQUARE attack. We describe our SQUARE-like attacks against HIEROCRYPT-3 in Sect. 4, and show how to apply them to HIEROCRYPT-L1 in Sect. 5. Sect. 6 provides some observations on an attack due to Gilbert and Minier against RIJNDAEL and its implications for HIEROCRYPT. Finally, Sect. 7 summarises the results.

2 Description of the HIEROCRYPT Ciphers

HIEROCRYPT-3 (HC-3) is an iterated 128-bit block cipher [13] with a variable number of rounds which depends on the cipher key size. For 128-bit keys, HC-3 has 6 rounds, for 192-bit keys there are 7 rounds, and for 256-bit keys, 8 rounds. The last round consists of an output transformation slightly different from the other rounds.

HC-3 has a hierarchical structure. At the highest level, an HC-3 round (see Fig. 1) consists of the following operations, in order:

- a layer of four simultaneous applications of 32×32-bit keyed substitution boxes (the XS-boxes).
- a diffusion layer consisting of a bytewise linear transform defined by the so-called MDS_H matrix.

Within each round, a similar structure exists. A 32×32-bit XS-box contains:

- a subkey mixing layer consisting of exoring the 32-bit input data with four subkey bytes. This layer is called the *upper* subkey layer in our attacks.
- a key-independent nonlinear layer composed of the parallel application of four 8×8-bit S-boxes. This layer is called *upper* S-box layer in our attacks.
- a diffusion layer consisting of a bytewise linear transform defined by the so-called MDS_L matrix, which is a Maximum Distance Separable (MDS) matrix [9,7].
- another subkey mixing layer consisting of exor with four subkey bytes. This layer is called *lower* subkey layer to contrast with the upper subkey layer.
- another key-independent nonlinear layer composed of the parallel application of four 8×8-bit S-boxes. Therefore, each *round* consists of *two* S-box layers. This layer is mentioned later as the *lower* S-box layer, to contrast with the previous similar layer.

The last round is an output transformation composed of an XS-box layer followed by an exor with the last round subkey (or alternatively, it has a round

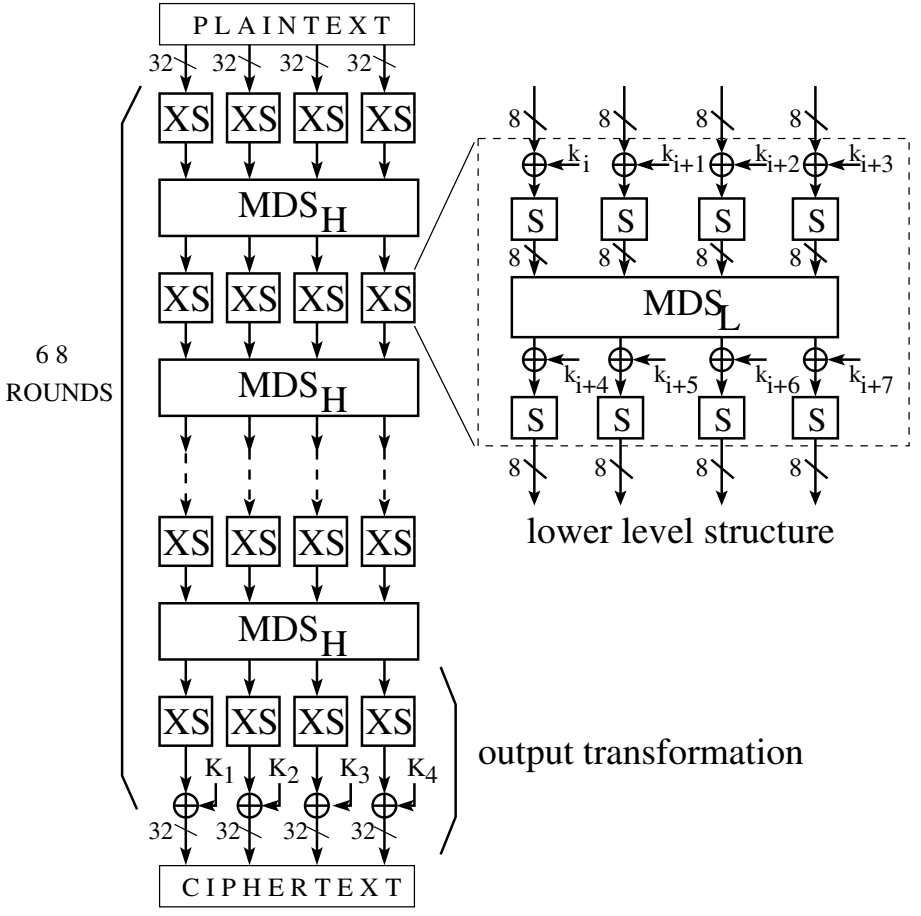

Fig. 1. Graphical representation of the HIEROCRYPT-3 encryption structure

structure with the MDS_H transform substituted by an exor layer with a subkey). The result is the 128-bit ciphertext block.

HIEROCRYPT-L1 (HC-L1) is a 64-bit block iterated cipher [14] using a 128-bit cipher key (see Fig. 2). It consists of six rounds plus an output transformation. Like HC-3, HC-L1 has a hierarchical structure. A high-level round consists of:

- a layer of two simultaneous applications of 32×32-bit keyed XS-boxes.
- a diffusion layer composed of a bytewise linear transform defined by the MDS'_H matrix.

Within each high-level round a similar structure exists. A 32-bit HC-L1 XS-box is identical to an HC-3 XS-box. The output transformation consists of an XS-box layer, followed by an exor mixing layer with the final 64-bit subkey. The result is the 64-bit ciphertext block.

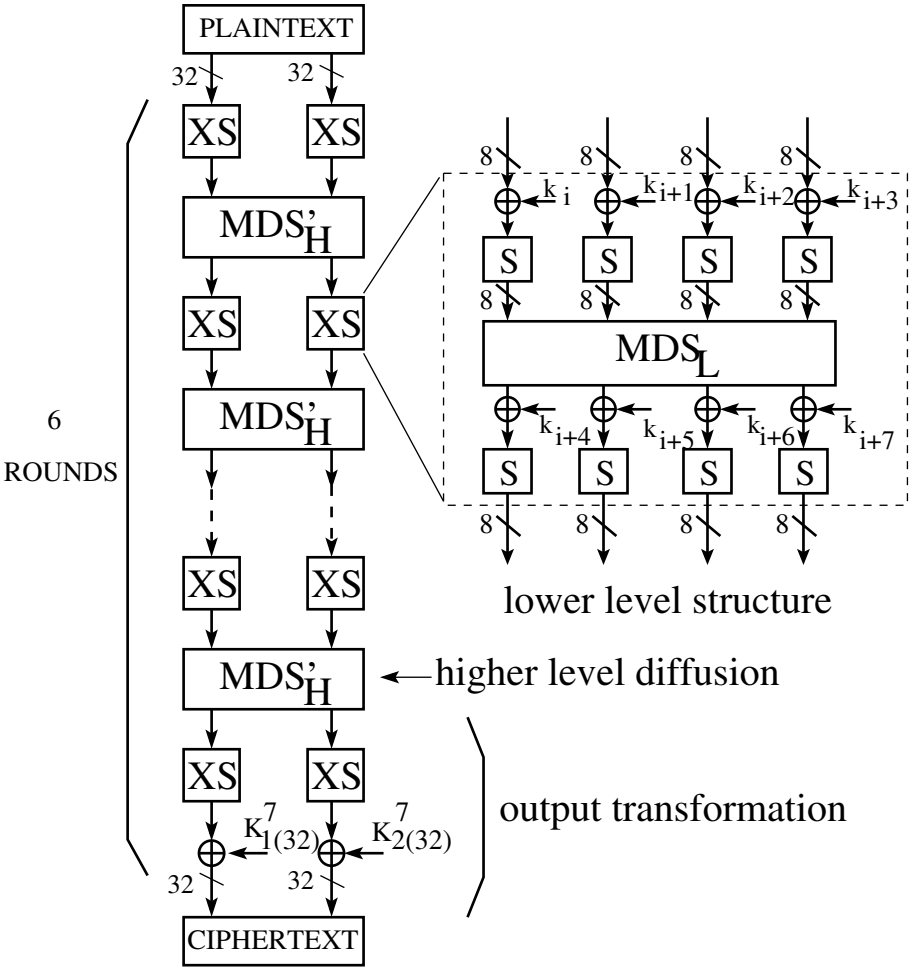

Fig. 2. Graphical representation of the HIEROCRYPT-L1 encryption structure

3 The SQUARE Attack

The SQUARE attack is a chosen-plaintext attack effective against reduced-round versions of block ciphers of the SQUARE family (see [1]). This attack explores the structure of SQUARE, where the *text* (the data being encrypted) is neatly partitioned into smaller component *words*. Let us provide some definitions.

Definition 1. *A method to distinguish the output of a certain number of rounds of an iterated cipher from the output of a truly random function is called a* distinguisher.

Definition 2. *([1]) A Λ-set is a set of* texts *that are all different in some of the component words (called* active words*) and are all equal in the other words (the* passive words*). Let $x = (x_1, \ldots, x_i, \ldots)$ and $y = (y_1, \ldots, y_j, \ldots)$ be two texts in the Λ-set, and λ be the set of indices of the active words. Then,*

$$\forall x, y \in \Lambda : \begin{cases} x_i \neq y_i \text{ for } i \in \lambda \\ x_i = y_i \text{ for } i \notin \lambda \end{cases}$$

The number of texts in a Λ-set depends on the word size. For SQUARE, where the word size is eight bits, a Λ-set consisted of 2^8 texts; in our present attacks a Λ-set consists of 2^{32} texts because we work with permutations of 32-bit words.

Definition 3. *(Balanced positions within a Λ-set) If the exor of all values at some word position in a Λ-set vanishes then that word position is called* balanced *over the Λ-set.*

The SQUARE attack starts by carefully choosing a Λ-set. By tracking the propagation of the active words through a round, it is possible to identify a pattern of active and passive words at the output of several rounds. That pattern contains a set of balanced words. These balanced words are used as a distinguishing test (further called the *Λ-distinguisher*) to identify the correct keys in the outer rounds of the cipher, either the first or the last round subkeys. Like SQUARE, HC-L1 and HC-3 possess a wordwise structure and similar variant attacks are possible.

The *partial sum technique* [4] is a dynamic programming technique that reduces the computational complexity of SQUARE attacks. It trades computational effort for storage by reorganising the intermediate computations. We refer to the original description of the technique [4] for a detailed explanation.

4 Attacking Reduced-Round HIEROCRYPT-3

4.1 An Observation on the Structure of the XS Transform

An equivalent representation of an XS-box, which is useful to describe our attack, can be obtained by pushing up the lower subkey layer to the input of the MDS_L matrix. This transformation creates an equivalent subkey layer.

Let the input to MDS_L be denoted by $X = (x_1, x_2, x_3, x_4) \in \mathrm{GF}(2^8)^4$ and its output by $Y = (y_1, y_2, y_3, y_4) \in \mathrm{GF}(2^8)^4$. Let $K = (k_1, k_2, k_3, k_4) \in \mathrm{GF}(2^8)^4$ be the subkey used in the lower subkey layer. The equivalent lower subkey layer is calculated by multiplying the original subkey by the MDS_L^{-1} transform:

$$MDS_L^{-1} = \begin{pmatrix} \mathsf{82}_x & \mathsf{C4}_x & \mathsf{34}_x & \mathsf{F6}_x \\ \mathsf{F6}_x & \mathsf{82}_x & \mathsf{C4}_x & \mathsf{34}_x \\ \mathsf{34}_x & \mathsf{F6}_x & \mathsf{82}_x & \mathsf{C4}_x \\ \mathsf{C4}_x & \mathsf{34}_x & \mathsf{F6}_x & \mathsf{82}_x \end{pmatrix}$$

For example, the equivalent most significant input to MDS_L is given by: $x_1 = \mathsf{82}_x \cdot (y_1 \oplus k_1) \oplus \mathsf{C4}_x \cdot (y_2 \oplus k_2) \oplus \mathsf{34}_x \cdot (y_3 \oplus k_3) \oplus \mathsf{F6}_x \cdot (y_4 \oplus k_4)$. It means, for instance, that the most significant subkey byte of the equivalent lower subkey layer is $ek_1 = \mathsf{82}_x \cdot k_1 \oplus \mathsf{C4}_x \cdot k_2 \oplus \mathsf{34}_x \cdot k_3 \oplus \mathsf{F6}_x \cdot k_4$.

4.2 Attacking Six S-Box Layers

We consider here a reduced-round version of HIEROCRYPT-3 that consists of two rounds, followed by the output transformation. In this version, there are three XS-box layers, which means six layers of the 8×8 S-box.

The main idea is to initially consider the nested structure as a black box and view the XS transform as the application of four keyed XS-boxes mapping $\mathrm{GF}(2^8)^4$ onto itself. The attack starts by choosing a set of 2^{32} plaintexts that have a fixed value for three of the four 32-bit blocks (the passive), while the fourth block (the active) takes all 2^{32} possible values. The position of the active block is not important. After the first round, all four 32-bit words are active (see [15], p. 5, Sect. 2.2.3).

After the second round the exor of all values from the Λ-set at each word position is zero (each word position is balanced). Since the round subkeys are constants (i.e. the same round subkeys are used to encrypt all the texts in the set), the words remain balanced after the addition with the upper round subkey of the third round (which is actually part of the output transformation). The attack proceeds by considering the ciphertexts. By guessing some bits of the round subkeys, we will partially decrypt up to the output of the second round and see whether the Λ-distinguisher holds. For correct guesses of the subkey bits, the Λ-distinguisher will always hold, while for wrong guesses the Λ-distinguisher will be destroyed with high probability. The following observation leads to a significant reduction of the attack complexity. Since the exor operation does not mix bits from different positions in the words, a 32-bit word being balanced implies all b-bit sub-blocks, $1 \leqslant b < 32$, will also be balanced. As a consequence, after the second round, every *byte* is balanced, and we only have to guess subkey bits to determine a single byte.

Now starting at the output, we have to guess 32 subkey bits that are added to one word in the output transformation, thus recovering the output from an XS-box. Looking at the inner structure of an XS-box, we see that we can invert the lower S-boxes as stated in Sect. 4.1, then move up the lower subkey layer, guess one byte of the equivalent subkey, undo the key addition and the upper S-box for one byte and verify the Λ-distinguisher.

Overall we are guessing 40 subkey bits. Since we are testing the Λ-distinguisher for one byte, we expect that one out of about 2^8 wrong keys will pass the test. If we repeat the attack with six Λ-sets, a wrong subkey will pass the test with probability $(2^{-8})^6 = 2^{-48}$. Since we try only 2^{40} subkeys, it is likely that only the correct one is able to pass through.

At first sight, it would seem that the 2^{40} key guessing steps should be repeated for each of the 2^{32} plaintexts in a Λ-set, with a resulting complexity of 2^{72} S-box lookups per Λ-set. However, the partial sum technique [4, section 2.3] provides a more efficient way to organise the guessing. At the modest cost of about 2^{24} extra bits, it reduces the total attack complexity to only 5×2^{48} S-box lookups per Λ-set.

Therefore, the complexity of the attack on six S-box layers is 6×2^{32} chosen plaintexts (six Λ-sets), and $6 \times 5 \times 2^{48} \approx 2^{53}$ S-box lookups.

4.3 Attacking Seven S-Box Layers

Attacking seven S-box layers is effective only for 192-bit and 256-bit keys. The strategy consists of guessing the first subkey (128 key bits) layer in an XS-box, and just use the previous attack on six S-box layers.

The computational effort becomes $2^{128+40} = 2^{168}$ subkey guesses. As 2^8 wrong subkeys will be filtered out per guess, and we guess 168 subkey bits at a time, the number of chosen plaintexts required to assure that only the correct subkey survives the filtering is $(\frac{168}{8}+1) \times 2^{32} = 22 \times 2^{32}$. By applying the partial sum technique, the overall complexity becomes $22 \times 5 \times 2^{48+128} \approx 2^{183}$ S-box lookups.

5 Attacking Reduced-Round HIEROCRYPT-L1

5.1 Attacking Six S-Box Layers

The best published attack against HIEROCRYPT-L1 is due to the cipher designers (see [12], p. 8, Table 4) and breaks five S-box layers at the cost of 2^{72} subkey guesses using 2^{32} chosen plaintexts. The same attack described in Sect. 4.2 for HIEROCRYPT-3 works against six S-box layers of HIEROCRYPT-L1, with the same requirements of $\approx 2^{53}$ S-box lookups and 6×2^{32} chosen plaintexts.

5.2 Attacking Seven S-Box Layers

The strategy for attacking seven S-box layers consists of guessing the complete last subkey (64 key bits) and using the previous attack on six S-box layers.

The computational effort becomes $2^{64+40} = 2^{104}$ subkey guesses. As 2^8 wrong subkeys will be filtered out per guess, and we guess 104 subkey bits at a time, the number of chosen plaintexts required to ensure that only the correct subkey survives the filtering is $(\frac{104}{8}+1) \times 2^{32} = 14 \times 2^{32}$. By applying the partial sum technique, the overall complexity becomes $14 \times 5 \times 2^{48+64} \approx 2^{118}$ S-box lookups.

6 The Gilbert-Minier Attack

The Gilbert-Minier attack [5], like the SQUARE attack, is based on distinguisher. In the terminology of [5], the SQUARE attack is based on a 3-round distinguisher. By guessing some round key bytes, it is possible to break six rounds of SQUARE. Gilbert and Minier developed a 4-round distinguisher for SQUARE (or RIJNDAEL), which makes it possible to break seven rounds.

In the case of the HIEROCRYPT ciphers, our extension of the SQUARE attack is already based on a 4-layer distinguisher. An improvement would thus require the construction of a 5-layer distinguisher. According to our analysis, the alternation of MDS_L and MDS_H (or MDS'_H) effectively prohibits the construction of a 5-round distinguisher. However, we feel that this issue deserves further investigation.

7 Conclusion

We presented improved SQUARE-like attacks against reduce-round versions of HIEROCRYPT-3, for six and seven S-box layers (3 and 3.5 rounds), compared to previous figures reported by its designers. The new attack requirements are summarised in Table 1. The columns labeled "subkey guesses," "S-box lookups," and "encryptions" actually refer to the same attack effort measured in different units. For the rightmost column (attack effort measured in number of encryptions) we use the approximate equivalence 1 encryption $\approx 2^7$ S-box lookups.

Table 1. Attack requirements for HIEROCRYPT-3

Attack	S-box layers	chosen plaintexts	subkey guesses	S-box lookups	encryptions
HC-3 paper[11]	4	2^{11}	2^{40}	$\leqslant 2^{51}$	$\leqslant 2^{44}$
HC-3 paper[11]	5	2^{13}	2^{168}	$\leqslant 2^{181}$	$\leqslant 2^{174}$
ours	6	6×2^{32}	2^{40}	2^{53}	2^{46}
ours	7	22×2^{32}	2^{168}	2^{183}	2^{176}

Table 2. Attack requirements for HIEROCRYPT-L1

Attack	S-box layers	chosen plaintexts	subkey guesses	S-box lookups	encryptions
HC-L1 paper[12]	5	2^{32}	2^{72}	$\leqslant 2^{104}$	$\leqslant 2^{97}$
ours	6	6×2^{32}	2^{40}	2^{53}	2^{46}
ours	7	14×2^{32}	2^{104}	2^{118}	2^{111}

The improved attacks work also for HIEROCRYPT-L1, with the appropriate changes due to the block size. The new attack requirements are summarised in Table 2. Again, the columns labeled "subkey guesses," "S-box lookups," and "encryptions" actually refer to the same attack effort measured in different units, and for the rightmost column we use the approximate equivalence 1 encryption $\approx 2^7$ S-box lookups. Our attacks, nonetheless, do not represent a threat to the security of either cipher.

References

1. J. Daemen, L.R. Knudsen, V. Rijmen, "The Block Cipher SQUARE," *Fast Software Encryption, LNCS 1267*, E. Biham, Ed., Springer-Verlag, 1997, pp. 149–165.
2. C. D'Halluin, G. Bijnens, V. Rijmen, B. Preneel, "Attack on Six Rounds of CRYPTON," *Fast Software Encryption, LNCS 1636*, L. Knudsen, Ed., Springer-Verlag, 1999, pp. 46–59.

3. "Evaluation of Cryptographic Techniques Project," http://www.ipa.go.jp/security/enc/CRYPTREC/index-e.html.
4. N. Ferguson, J. Kelsey, S. Lucks, B. Schneier, M. Stay, D. Wagner, and D. Whiting, "Improved Cryptanalysis of RIJNDAEL," to appear in *Fast Software Encryption'00*, Springer-Verlag.
5. H. Gilbert, M. Minier, "A Collision Attack on Seven Rounds of RIJNDAEL," *Third Advanced Encryption Standard Candidate Conference*, NIST, April 2000, pp. 230–241.
6. S. Lucks, *"The Saturation Attack – A Bait for Twofish,"* these Proceedings.
7. F.J. MacWilliams, N.J.A. Sloane, "The Theory of Error-Correcting Codes," *North-Holland Mathematical Library*, vol. 16, 1977.
8. NESSIE Project – New European Schemes for Signatures, Integrity and Encryption – http://cryptonessie.org.
9. V. Rijmen, "Cryptanalysis and Design of Iterated Block Ciphers," *Doctoral Dissertation*, October 1997, K.U.Leuven.
10. V. Rijmen, J. Daemen, B. Preneel, A. Bosselaers, E. De Win, "The Cipher SHARK," *Fast Software Encryption, LNCS 1039*, D. Gollmann, Ed., Springer-Verlag, 1996, pp. 99-112
11. Toshiba Corporation, "Security Evaluation: HIEROCRYPT-3," September 25, 2000 – available at http://cryptonessie.org.
12. Toshiba Corporation, "Security Evaluation: HIEROCRYPT-L1," September 25, 2000 – available at http://cryptonessie.org.
13. Toshiba Corporation, "Specification of HIEROCRYPT-3," submitted to the First Open NESSIE Workshop, 13-14 November 2000, Leuven, Belgium – available at http://cryptonessie.org.
14. Toshiba Corporation, "Specification of HIEROCRYPT-L1," submitted to the First Open NESSIE Workshop, 13-14 November 2000, Leuven, Belgium – available at http://cryptonessie.org.
15. Toshiba Corporation, "Specification on a Block Cipher: HIEROCRYPT-3," Toshiba Corporation, Sep. 15, 2000 – submitted to the First Open NESSIE Workshop, 13-14 November 2000, Leuven, Belgium – available at http://cryptonessie.org.