A Family of Trapdoor Ciphers

Vincent Rijmen* Bart Preneel**

Katholieke Universiteit Leuven,
Department Electrical Engineering-ESAT/COSIC
K. Mercierlaan 94, B-3001 Heverlee, Belgium
{vincent.rijmen,bart.preneel}@kuleuven.ac.be

Abstract. This paper presents several methods to construct trapdoor block ciphers. A trapdoor cipher contains some hidden structure; knowledge of this structure allows an attacker to obtain information on the key or to decrypt certain ciphertexts. Without this trapdoor information the block cipher seems to be secure. It is demonstrated that for certain block ciphers, trapdoors can be built-in that make the cipher susceptible to linear cryptanalysis; however, finding these trapdoors can be made very hard, even if one knows the general form of the trapdoor. In principle such a trapdoor can be used to design a public key encryption scheme based on a conventional block cipher.

1 Introduction

Researchers have been wary of trapdoors in encryption algorithms, ever since the DES [9] was proposed in the seventies [15]. In spite of this, no one has been able to show how to construct a practical block cipher with a trapdoor. For most current block ciphers it is relatively easy to give strong evidence that there exist no full trapdoors. We define a *full* trapdoor as some secret information which allows an attacker to obtain knowledge of the key by using a very small number of known plaintexts, no matter what these plaintexts are or what the key is.

In this paper we consider *partial* trapdoors, i.e., trapdoors that not necessarily work for all keys, or that give an attacker only partial information on the key. We show that it is possible to construct block ciphers for which there exists a linear relation with a high probability; knowledge of such a relation allows for a linear attack which requires only a very small number of known plaintexts [13, 14]. A trapdoor is said to be *detectable* (*undetectable*) if it is computationally feasible (infeasible) to find it even if one knows the general form of the trapdoor.

The rest of this paper is organized as follows. In §2 we explain how both detectable and undetectable trapdoors can be built into S-boxes. §3 deals with trapdoors in round functions and complete block ciphers. Extensions are discussed in §4, and the conclusions are presented in §6.

* F.W.O research assistant, sponsored by the Fund for Scientific Research - Flanders (Belgium).

** F.W.O. postdoctoral researcher, sponsored by the Fund for Scientific Research - Flanders (Belgium).

The inner product of two Boolean vectors x and y with components $x[0]$ through $x[m]$ and $y[0]$ through $y[m]$ will be denoted with

$$x \bullet y = \bigoplus_{i=1}^{m} x[i] \cdot y[i] \,.$$

2 Trapdoor $m \times n$ S-boxes

In this section we discuss the construction and hiding of trapdoors in S-boxes.

2.1 Construction

An $m \times n$ substitution box (or S-box) can be defined by its component functions: a collection of n Boolean functions $f_i(x)$, $i = 1, \ldots, n$, that take as input Boolean vectors x of dimension m. We start with an $m \times (n-1)$ S-box $S(x)$ consisting of $n-1$ functions f_i, $i = 1, \ldots, n$, $i \neq q$ selected randomly according to a uniform distribution (or following an arbitrary design criterion). The trapdoor $m \times n$ S-box $T(x)$ is derived from $S(x)$ by adding an extra function in the following way. We choose an n-bit Boolean vector β with $\beta_q = 1$ for some q with $1 \leq q \leq n$ and ensure that

$$f_q(x) = \bigoplus_{i=1, i \neq q}^{n} \beta[i] \cdot f_i(x) \tag{1}$$

with probability p_T. Then

$$\beta \bullet T(x) = 0 \tag{2}$$

holds with probability $p_T(\beta)$. This is equivalent to a correlation

$$c_T(\beta) = 2 \cdot p_T(\beta) - 1$$

between the constant zero function and $\beta \bullet T(x)$. The trapdoor information is the vector β.

2.2 Hiding the Trapdoor

If the S-box is claimed to be selected randomly according to a uniform distribution from all $m \times n$ S-boxes, it is rather easy to hide a trapdoor in it. Indeed, for large values of n and m, the function $f_q(x)$ is computationally indistinguishable from a randomly selected one. We first prove that this construction in fact introduces only one β-vector with a high correlation value, not accompanied by a range of β-vectors with 'smaller' correlation values. Then we discuss the difficulty of finding this trapdoor vector.

Introducing no more than one β with high correlation: Suppose $S(x)$ is an $m \times (n-1)$ S-box selected such that for all n-bit vectors γ,

$$c_S(\gamma) \leq c_{\max}\,.$$

Consider now the $m \times n$ S-box $T(x)$ that results from adding $f_q(x)$ to $S(x)$. It still holds that for all γ with $\gamma_q = 0$,

$$c_T(\gamma) = c_S(\gamma) \leq c_{\max}\,,$$

so we are left with the cases where $\gamma_q = 1$. If $p_T = 1$, then $\beta \bullet T(x) = 0$ and:

$$\begin{aligned} \gamma \bullet T(x) &= (\gamma \bullet T(x)) \oplus (\beta \bullet T(x)) \\ &= (\gamma \oplus \beta) \bullet T(x) \\ &= (\gamma \oplus \beta) \bullet S(x)\,. \end{aligned} \tag{3}$$

Since $(\gamma \oplus \beta)_q = 0$, for all $\gamma \neq \beta$,

$$c_T(\gamma) = c_S(\gamma \oplus \beta) \leq c_{\max}\,. \tag{4}$$

Equation (3) holds with probability p_T. If $p_T < 1$ it is possible that (4) does not hold. Consider in this case the S-box $T'(x)$ that results from (1) if $p_T = 1$. All correlations of $T'(x)$ are below $c_{\max}$. Thus $T(x)$ can be considered as being constructed by applying $(1-p_T) \cdot 2^m$ random changes to one component of $T'(x)$. The probability that these random changes to the random S-box will result in a significant change of $c_{\max}$ is very small.

Recovering β If a cryptanalyst suspects a relation of the form (2), he can decide to examine the $2^n - 1$ non-zero values of β exhaustively. For each value of β, verifying p_T requires the computation of a Walsh-Hadamard transform on an m-bit Boolean function [2], which requires $O(m \cdot 2^m)$ operations. If $(m, n) = (8, 32)$ this is feasible and the trapdoor is detectable, but for $(m, n) = (8, 64)$ this requires about 2^{64} Walsh-Hadamard transformations on 8-bit functions, which is currently quite hard (256 times more difficult than a DES key search). For $(m, n) = (10, 80)$ an exhaustive search is at present not feasible. The search can possibly be sped up by lattice methods (such as LLL [12]) or coding theory techniques, but the applicability of these techniques is still an open problem.

The search for the β-vector that has high correlation is equivalent to the problem of learning a parity function in the presence of noise. The Parity Assumption [4] tells that this problem is probably NP-hard. This classification only deals with the general problem; specific instances might be easier to solve. For instance, if p_T is very close to one, it is possible to use Gaussian elimination to solve the problem.

Define the m Boolean vectors $a^{(j)}$, $j = 1, \ldots, n$ as $a^{(j)}[i] = f_j(i)$, $i = 0, \ldots 2^m - 1$. Equation (1) can then be translated into

$$\bigoplus_{i=1}^{n} \beta[i] \cdot a^{(i)} = \delta\,. \tag{5}$$

If (1) holds with probability one, or $p_T(\beta) = 1$, then $\delta = 0$. In this case the $a^{(i)}$'s are linearly dependent and the linear relation between the vectors can be recovered in a very efficient way with Gaussian elimination on (5). If the probability of (1) is smaller than one, the vectors $a^{(i)}$ are independent; δ is different from zero and unknown to the cryptanalyst, and the Hamming weight of δ is given by

$$\mathrm{w}_h(\delta) = 2^m(1 - p_T)\,.$$

The cryptanalyst can still try to recover β by guessing a (low-weight) value for δ and solving the set of equations (5). Equation (5) will only have a solution when the guess for δ is correct. A more complex strategy for the cryptanalyst is to use the following equations:

$$\bigoplus_{i=1}^{n} \beta[i] \cdot a^{(i)} = \bigoplus_{i=1}^{d} \gamma[i] \cdot \delta^{(i)}\,. \tag{6}$$

The d vectors $\delta^{(i)}$ are guessed by the cryptanalyst. If the unknown δ can be expressed as a linear combination of the vectors $\delta^{(i)}$, the cryptanalyst can hope to find the trapdoor by solving (6) for $\beta[i]$ and $\gamma[i]$. The probability that δ is a linear combination of the d vectors $\delta^{(i)}$ increases with d.

If the $\delta^{(i)}$ vectors are linearly independent, they generate a vector space of size 2^d. Note that we are only interested in the vectors with a low Hamming weight, say all vectors with Hamming weight $\leq D$. For simplicity we assume that all the $\delta^{(i)}$ vectors have Hamming weight one. The number of vectors in a d-dimensional space with Hamming weight $\leq D$ is given by

$$\sum_{k=1}^{D} \binom{d}{k}\,.$$

Table 1 shows the numerical values for several choices of D and d.

D	$d = 64$	$d = 96$	$d = 256$	$d = 1024$
1	2^6	2^7	2^8	2^{10}
10	2^{37}	2^{44}	2^{58}	2^{78}
20	2^{55}	2^{68}	2^{98}	2^{139}
32	2^{63}	2^{86}	2^{136}	2^{202}
40	2^{64}	2^{92}	2^{156}	2^{240}

Table 1. The number of vectors in a d-dimensional space with Hamming weight $\leq D$.

For example, consider a 10×40 S-box. There are 2^{10} inputs, and for each input the equations may hold or not hold, resulting in a number of $2^{2^{10}}$ possible δ vectors; 2^{202} of them have Hamming weight ≤ 32. If we take $d = 64$, the probability p_{lc} that δ is a linear combination of d randomly chosen $\delta^{(i)}$ vectors is equal to $2^{63}/2^{202}$. The work factor of this algorithm is determined by p_{lc} and by the work to solve (6), which is $\mathcal{O}((2^m + n + d)^3)$ (note that the best asymptotic algorithms reduce the exponent from 3 to 2.376 [6]).

By increasing d we increase p_{lc}. However, if d becomes larger than a certain threshold value, spurious solutions for δ will start to appear that have a large Hamming weight. These unwanted solutions correspond to β vectors with low correlation values. This effect limits the use of Gaussian elimination. This algorithm will be be more useful than exhaustive search for β if D and n are small, and m is large.

2.3 Bent Functions

The construction of §2.1 can be extended to deal with additional constraints imposed on the functions $f_i(x)$. For example, in some block ciphers (such as the CAST family [1]), it is necessary that the component functions $f_i(x)$ are bent functions. The Maiorana construction for bent functions [7] can then be used to obtain an S-box satisfying property (2): an m-bit bent function $f(x)$ (m is even) is obtained from an $m/2$-bit permutation $\pi(y)$ and an arbitrary $m/2$-bit function $g(z)$ as follows:

$$f(x) = f(y, z) = \pi(y) \cdot z \oplus g(z) .$$

Here '$\cdot$' denotes multiplication in $GF(2^{m/2})$. If two component functions $f_i(x)$ and $f_j(x)$ are derived from the same permutation $\pi(y)$, we obtain

$$f_i(y, z) \oplus f_j(y, z) = g_i(z) \oplus g_j(z) ,$$

which can be chosen arbitrarily close to a constant function. To hide (2) in a bent function based S-box we proceed as follows. we choose a β with even Hamming weight. We divide the set of indices where $\beta_i = 1$ arbitrarily into pairs. For each pair of indices we select a different mapping $x \mapsto (y, z)$ and a different permutation π. We define $m/2$-bit functions $g_i(z)$, and extend them to full m-bit functions by adding zero values. Then

$$\beta \bullet T(x) = \bigoplus_{i=1}^{m} g_i(x) = 0$$

with probability p_T.

This construction shows that is possible to find a set of bent functions that sum to an almost constant function. We believe that is also possible to use other bent functions in a similar construction.

3 Trapdoor Ciphers

In this section we propose several constructions for trapdoors in block ciphers starting from the building blocks, i.e., the round functions.

3.1 Trapdoor Round Functions

We now show that the trapdoors in S-boxes can be extended to trapdoors in the round function of a Feistel cipher [8]. A $2p$-bit Feistel cipher with r rounds operates as follows: plaintext and ciphertext consist of two p–bit halves denoted with L_0, R_0 and L_r, R_r respectively. The key is denoted with K. Each round takes a $2p$–bit message input block L_{i-1}, R_{i-1} and a k–bit key input K_i which is derived from K using the key scheduling algorithm. The output of the ith round is computed as follows:

$$\begin{aligned} R_i &= L_{i-1} \oplus F(K_i \oplus R_{i-1}) \\ L_i &= R_{i-1} \qquad\qquad\qquad i = 1, \ldots, r\,. \end{aligned}$$

Here F is the round function of the Feistel cipher. Note that after the last round, the swapping of the halves is undone to make encryption and decryption similar.

In this section we consider the round functions of variants on CAST [10] and LOKI91 [5].

tCAST: The ciphers of the CAST family are 64-bit Feistel ciphers, or $p = 32$. The round function F is based on four 8×32 S-boxes, which have components that are either randomly selected functions or are bent functions [1]. The 32-bit input of the round function is divided into four bytes, each going to one of the four S-boxes; the 32-bit output is obtained as the sum modulo 2 of the outputs of the four S-boxes. Using four S-boxes with the same trapdoor β (but with a different value of c_T, denoted with $c_{T^{(i)}}$), we obtain

$$\beta \bullet F(x^{(1)}, x^{(2)}, x^{(3)}, x^{(4)}) = \bigoplus_{i=1}^{4} \beta \bullet T^{(i)}(x^{(i)})\,.$$

Hence the round function correlates with the constant zero function with a correlation equal to

$$c_F = c_{T^{(1)}} c_{T^{(2)}} c_{T^{(3)}} c_{T^{(4)}}\,.$$

As mentioned before, 8×32 S-boxes can be checked for this type of trapdoors. However, should CAST be extended in a natural way to an 128-bit block cipher by using 8×64-bit S-boxes, finding this trapdoor becomes very difficult. The technique can be extended to CAST variants where the exor operation is replaced by a modular addition or multiplication.

tLOKI: The expansion in the round function of LOKI91 [5] allows for a subtle trapdoor, not visible in the individual S-boxes, but only in the round function.

We denote concatenation with '||'. The round function of LOKI91 uses four times the same 12×8 S-box, and is defined as:

$$\begin{aligned} F(x[1], \ldots, x[32]) = P(S(x[29], x[30], x[31], x[32], x[1], \ldots, x[8]) \,||\, \\ S(x[5], x[6], \ldots, x[16]) \,||\, S(x[13], x[14], \ldots, x[24]) \,||\, S(x[21], x[22], \ldots, x[32])) \end{aligned}$$

In this analysis the bit permutation P is not relevant and will be ignored. We can build a trapdoor into this round function as follows. Let $a^{(1)}(x)$, $a^{(2)}(x)$, $a^{(3)}(x)$, and $a^{(4)}(x)$ be four 8-bit Boolean functions and $\beta = \beta^{(1)}||\beta^{(2)}||\beta^{(3)}||\beta^{(4)}$ a 32-bit Boolean vector. Suppose the following nonlinear relations hold with probabilities p_1, p_2, p_3, p_4 respectively.

$$\begin{aligned}
\beta^{(1)} \bullet S(x[1], \ldots x[12]) &= a^{(1)}(x[1], x[2], x[3], x[4]) \oplus a^{(2)}(x[9], x[10], x[11], x[12]) \\
\beta^{(2)} \bullet S(x[1], \ldots x[12]) &= a^{(2)}(x[1], x[2], x[3], x[4]) \oplus a^{(3)}(x[9], x[10], x[11], x[12]) \\
\beta^{(3)} \bullet S(x[1], \ldots x[12]) &= a^{(3)}(x[1], x[2], x[3], x[4]) \oplus a^{(4)}(x[9], x[10], x[11], x[12]) \\
\beta^{(4)} \bullet S(x_1, \ldots x_{12}) &= a^{(4)}(x[1], x[2], x[3], x[4]) \oplus a^{(1)}(x[9], x[10], x[11], x[12])
\end{aligned}$$

The use of nonlinear relations in a linear approximation was already studied by Knudsen and Robshaw [11]. The correlation between

$$\begin{aligned}
\beta \bullet F(x[1], \ldots, x[32]) = \beta^{(1)} \bullet S(x[29], \ldots, x[8]) \;\oplus\; \beta^{(2)} \bullet S(x[5], \ldots, x[16]) \\
\oplus\; \beta^{(3)} \bullet S(x[13], \ldots, x[24]) \;\oplus\; \beta^{(4)} \bullet S(x[21], \ldots, x[32])
\end{aligned}$$

and the constant zero function is now given by $(2p_1-1)(2p_2-1)(2p_3-1)(2p_4-1)$. For the parameters of LOKI91, this is probably a detectable trapdoor, at least for someone who knows what he is looking for. Again, larger block sizes and S-boxes would make such trapdoors harder to detect.

3.2 Trapdoor Ciphers

The trapdoor round functions defined above can be used to construct a trapdoor cipher. The resulting cipher will have iterative linear relations that approximate the output of every other round. For a cipher with r rounds, one needs $\lfloor r/2 \rfloor - 1$ round approximations.

For example, consider a version of tCAST with 16 rounds, block size 80 bits, and using four 10×40 S-boxes. If $p_T = 1-2^{-5}$ we can recover the round key of the first and the last round with Matsui's algorithm 2 [13] using approximately 875 known plaintexts. Since the Hamming weight of δ is 32, the Gaussian elimination technique to find the trapdoor will not work faster than exhaustive search.

4 Extensions

The trapdoors we considered make all use of "type II" linear relations as defined in [14]: correlations that exist between the output bits of the round function. It is also possible to hide "type I" linear relations: correlations between input and output bits of the round function. For example, we can construct S-boxes such that

$$\beta \bullet S[x] = \alpha \bullet x \qquad (A)$$

$$\alpha \bullet S[x] = \beta \bullet x \qquad (B)$$

with high probability. It is easy to see that these relations can be concatenated in the following way: $AB - BA - AB - \ldots$ The main advantage of this type of relations is that there are more of them: 2^{n+m} instead of 2^n. If $(m, n) = (8, 32)$, as in CAST, there are already 2^{40} cases to verify.

When building the trapdoor in the round function of tLOKI, we make use of the fact that in LOKI91 the key is added before the expansion (the input to the round function consists of 32 bits, but some of these are duplicated such that 48 bits are input to the S-boxes). In the DES the key is added after the expansion; in this case one can introduce trapdoors as well. A first approach consists of choosing linear functions $a^{(i)}(x)$. In this way the absolute value of the correlation between bits is independent of the key. However this imposes a severe restriction on the number of possible trapdoors, which makes them easy to detect. (We checked the DES for these trapdoors and have not found any.) Another option is to hide several key dependent trapdoors. The key schedule could be carefully adapted such that only a small number of key bits have an actual influence.

In a similar way one can hide differentials into block ciphers, in order to make them vulnerable to differential cryptanalysis [3]. However, exploitation of such trapdoors requires chosen rather than known plaintexts, which is much less practical.

5 Public Key Encryption

Besides the obvious use by government agencies to catch dangerous terrorists and drug dealers, trapdoor block ciphers can also be used for public key cryptography. For this application on selects a block cipher with variable S-boxes and makes it widely available (it is a system-wide public parameter). Bob generates a set of S-boxes with a secret trapdoor. These S-boxes form his public key. If Alice wants to send a confidential message to Bob, she generates a random session key, encrypts her message and a fixed set of plaintexts and sends the ciphertexts to Bob. The set of plaintexts can be fixed, or can be generated from a short seed using a pseudo-random bit generator. Bob uses the trapdoor and the known plaintexts to recover the session key and decrypts the message.

There seems to be no obvious way to extend this construction to digital signatures.

6 Conclusion

We have shown that is rather easy to hide trapdoors in expanding S-boxes like the 8 × 32-boxes that are currently used in some ciphers. Extending the S-boxes to 10 × 80 bits makes the trapdoors undetectable.

The expansion function that is used in LOKI and the DES can be used to combine 'innocent' S-boxes into a trapdoor round function. The fact that key addition in the DES is done after the expansion creates the possibility for key dependent trapdoors.

We conclude that the danger of trapdoors in block ciphers is real. Defending against built-in trapdoors can be done in several ways. For some ciphers it is feasible check for several classes of trapdoors. A pro-active approach is to nourish a healthy distrust for other people's pseudo-random generators. A design that uses random elements should clearly explain the process of the pseudo-random bit generation, and, if applicable, the screening process. For algorithms which are kept secret, such as Skipjack, this is an even more worrying problem.

Acknowledgments

We would like to thank the anonymous reviewers and several attendants of the workshop for the interesting suggestions. Special thanks to Ron Rivest for drawing our attention to some relevant papers on learning theory.

References

1. C.M. Adams, S.E. Tavares, Designing S-boxes for ciphers resistant to differential cryptanalysis, *Proceedings of the 3rd Symposium on State and Progress of Research in Cryptography,* W. Wolfowicz, Ed., Fondazione Ugo Bordoni, 1993, pp. 181–190.
2. K.G. Beauchamp, *Walsh Functions and Their Applications,* Academic Press, New York, 1975.
3. E. Biham, A. Shamir, *Differential Cryptanalysis of the Data Encryption Standard,* Springer-Verlag, 1993.
4. A. Blum, M. Furst, M. Kearns, R.J. Lipton, "Cryptographic primitives based on hard learning problems," *Advances in Cryptology, Proceedings Crypto'93, LNCS 773,* D. Stinson, Ed., Springer-Verlag, 1994, pp. 278–291.
5. L. Brown, M. Kwan, J. Pieprzyk, J. Seberry, "Improving resistance against differential cryptanalysis and the redesign of LOKI," *Advances in Cryptology, Proceedings Asiacrypt'91, LNCS 739,* H. Imai, R.L. Rivest, and T. Matsumoto, Eds., Springer-Verlag, 1993, pp. 36–50.
6. D. Coppersmith, S. Winograd, "Matrix multiplication via arithmetic progressions," *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing,* 1987, pp. 1–6.
7. J.F. Dillon, "Elementary Hadamard difference sets," *Proceedings of the Sixth Southeastern Conference on Combinatorics, Graph Theory and Computing, Boca Raton, Florida, Congressum Numerantium No. XIV, Utilitas Math., Winnipeg, Manitoba,* 1975, pp. 237–249.
8. H. Feistel, W.A. Notz, J.L. Smith, "Some cryptographic techniques for machine-to-machine data communications," *Proceedings IEEE,* Vol. 63, No. 11, November 1975, pp. 1545–1554.
9. FIPS 46, *Data Encryption Standard,* NBS, U.S. Department of Commerce, Washington D.C., Jan. 1977.
10. H.M. Heys, S.E. Tavares, On the security of the CAST encryption algorithm, *Canadian Conference on Electrical and Computer Engineering,* pp. 332–335, Sept. 1994, Halifax, Canada.
11. L.R. Knudsen, M.J.B. Robshaw, "Non-linear approximations in linear cryptanalysis," *Advances in Cryptology, Proceedings Eurocrypt'96, LNCS 1070,* U. Maurer, Ed., Springer-Verlag, 1996, pp. 224–236.

12. A.K. Lenstra, H.W. Lenstra, Jr., L. Lovász, "Factoring polynomials with rational coefficients," *Math. Annalen,* No. 261, pp. 513–534, 1982.
13. M. Matsui, "On correlation between the order of S-boxes and the strength of DES," *Advances in Cryptology, Proceedings Eurocrypt'94, LNCS 950,* A. De Santis, Ed., Springer-Verlag, 1995, pp. 366–375.
14. M. Matsui, "The first experimental cryptanalysis of the Data Encryption Standard," *Advances in Cryptology, Proceedings Crypto'94, LNCS 839,* Y. Desmedt, Ed., Springer-Verlag, 1994, pp. 1–11.
15. M.E. Smid, D.K. Branstad, "The Data Encryption Standard. Past and future," in *"Contemporary Cryptology: The Science of Information Integrity,"* G.J. Simmons, Ed., IEEE Press, 1991, pp. 43–64.