J. Cryptology (2001) 14: 197–210
DOI: 10.1007/s00145-001-0003-x

Journal of
CRYPTOLOGY

Attacks on Block Ciphers of Low Algebraic Degree

Thomas Jakobsen
Technical University of Denmark,
DK-2800 Kgs. Lyngby, Denmark

Lars R. Knudsen
Department of Informatics, University of Bergen,
N-5020 Bergen, Norway

Communicated by Tom Berson

Received April 1999 and revised October 2000
Online publication 9 April 2001

Abstract. In this paper an attack on block ciphers is introduced, the interpolation attack. This method is useful for attacking ciphers that use simple algebraic functions (in particular quadratic functions) as S-boxes. Also, attacks based on higher-order differentials are introduced. They are special and important cases of the interpolation attacks. The attacks are applied to several block ciphers, the six-round prototype cipher by Nyberg and Knudsen, which is provably secure against ordinary differential cryptanalysis, a modified version of the block cipher SHARK, and a block cipher suggested by Kiefer.

Key words. Block cipher, Cryptanalysis, Interpolation, Higher-order differentials.

1. Introduction

In an r-round iterated cipher the ciphertext is computed by iteratively applying in r rounds a *round function* g to the plaintext, such that

$$x_i = g(k_i, x_{i-1}),$$

where x_0 is the plaintext, k_i is the ith round key, and $x_r = y$ is the ciphertext. A special kind of iterated ciphers are the **Feistel** ciphers. A Feistel cipher with block size $2m$ and r rounds is defined as follows. Let x_0^{L} and x_0^{R} be the left and right halves of the plaintext, respectively, each of m bits. The round function g operates as follows:

$$\begin{aligned} x_i^{\mathrm{L}} &= x_{i-1}^{\mathrm{R}}, \\ x_i^{\mathrm{R}} &= f(k_i, x_{i-1}^{\mathrm{R}}) + x_{i-1}^{\mathrm{L}}, \end{aligned}$$

and the ciphertext is the concatenation of x_r^{R} and x_r^{L}. Note that f can be any function taking as arguments an m-bit text and a round key k_i and producing m bits. "+" is a commutative group operation on the set of m-bit blocks. For the remainder of this paper it assumed that "$+$" is the exclusive-or operation ($\oplus$).

The attacks presented in this paper are classified according to the taxonomy of [5]. In a *key-recovery attack* an attacker finds the secret key. In a *global deduction* an attacker finds an algorithm, which encrypts any plaintext into a valid ciphertext without knowing the secret key. In an *instance deduction* an attacker finds an algorithm, which encrypts a subset of all plaintexts into valid ciphertexts without knowing the secret key.

The *reduced cipher* is the cipher that one gets by removing the final round of the original cipher. The output from this cipher is denoted $\tilde{y} = (\tilde{y}_L, \tilde{y}_R)$.

In a key-recovery attack one often tries to find the value of the last-round key. A guess of this value is used to decrypt the ciphertext by one round and in this way one hopes to obtain the output from the reduced cipher. If there exists a method to distinguish whether this is the actual output from the reduced cipher or not, then one can find the last-round key. Once this key has been found, attacks similar to the ones presented here can be mounted on a cipher one round shorter than the original. As the measurement of the time needed by an attack, the total number of encryptions of the attacked block cipher is used. Note that this general description of an attack can be extended to the case where the attacker looks for the first-round key instead of the last-round key or both at the same time.

This paper considers two types of attacks. First, attacks based on higher-order differentials are given. Generalizations of this attack are then introduced under the name of interpolation attacks.

Let the ciphertext bits y_j be expressed as multivariate polynomials $q_j(x) \in \mathrm{GF}(2)[x_1, \ldots, x_m]$, where the x_i's are the plaintext bits. The algebraic degree of the encryption function is then defined to be the maximum total degree of these polynomials, $\max_j \deg q_j$. The higher-order differential attack is applicable if the algebraic degree d of the ciphertext from the reduced cipher as a function of the plaintext is low. Since a dth-order differential over such a cipher is a constant, it is possible to predict certain values of the output of the reduced cipher, which can be used to recover the last-round key.

In the univariate version of the interpolation attack, the ciphertexts are expressed as polynomials $p(x) \in \mathrm{GF}(2^m)[x]$ of the plaintexts x. If such a polynomial has a sufficiently low degree or a low number of nonzero coefficients, it is possible to reconstruct it from a collection of plaintexts and their corresponding ciphertexts. This can be used to construct an algorithm which can encrypt and decrypt without knowledge of the secret key, and it can be used to recover the secret key in iterated ciphers. By viewing the plaintexts and ciphertexts as the concatenation of s blocks of m/s bits, where s divides m, the attack is generalized to multivariate polynomials. It follows that the higher-order differential attack is the special case where $s = m$.

This paper is organized as follows. Section 2 gives new attacks based on higher-order differentials and in Section 3 a new attack on block ciphers is presented, the interpolation attack. In Section 4 the attacks are applied to the cipher by Nyberg and Knudsen [10] and to a modification hereof, to a modified version of SHARK [11], and to a cipher by Kiefer [4]. Conclusions are in Section 5.

2. Attacks Using Higher-Order Differentials

In [7] Lai gave a definition of higher-order derivatives of discrete functions. Later Knudsen used higher-order differentials to cryptanalyze ciphers presumably secure against conventional differential attacks, that is, attacks based on first-order differentials [6]. An extension of Knudsen's attacks is given next. The reader is referred to [6] and [7] for the definitions of higher-order differentials.

Consider a Feistel cipher with block size $2m$. Suppose that x_{R} is kept constant and consider the right side $\tilde{y}_{\mathrm{R}}$ of the output from the reduced cipher. Since x_{R} is a constant, each bit of $\tilde{y}_{\mathrm{R}}$ can be expressed as a multivariate polynomial $\mathrm{GF}(2)[x_1, x_2, \ldots, x_m]$ in the bits of $x_{\mathrm{L}} = (x_1, x_2, \ldots, x_m)$. Assume that none of these polynomials have degree higher than d. Then according to Proposition 2 of [7] (see also [6]), we have

$$\sum_{x_{\mathrm{L}} \in \mathcal{L}_d} p(x_{\mathrm{L}}) = c, \tag{1}$$

where $\mathcal{L}_d$ denotes a d-dimensional subspace of $\mathrm{GF}(2)^m$, c is a constant for any space parallel to $\mathcal{L}_d$, and p is a function which computes the output from the reduced cipher. It follows that

$$\sigma(w) = \sum_{x_{\mathrm{L}} \in \mathcal{L}_{d+1}} p(x_{\mathrm{L}} + w) = 0, \qquad \text{for all} \qquad w \in \mathrm{GF}(2)^m, \tag{2}$$

if and only if $p(x)$ is a polynomial of degree d or lower. In the following algorithm the variables $x = (x_{\mathrm{L}}, x_{\mathrm{R}})$ and $y = (y_{\mathrm{L}}, y_{\mathrm{R}})$ hold the plaintext and the ciphertext, respectively, T is a full rank $(d+1) \times m$ matrix over GF(2), and f is the round function.

```
1. Let $x_R$ and $w$ be $m$-bit constants.
2. For all $a \in GF(2)^{d+1}$:
 (a) Let $x_L = aT + w$.
 (b) Obtain the ciphertext $y(a)$ of plaintext $(x_L, x_R)$.
3. For all values, $k$, of the last-round key:
 (a) Let $\sigma = 0$.
 (b) For all $a \in GF(2)^{d+1}$:
       (i) Let $y = y(a)$.
      (ii) Let $\tilde{y}_R = y_L \oplus f(k, y_R)$.
     (iii) Let $\sigma = \sigma \oplus \tilde{y}_R$.
```

The keys for which σ ends up being zero are candidates for the correct value of the last-round key. If it is assumed that for a wrongly guessed value of the last-round key the values of σ are random and uniformly distributed one can repeat the algorithm with another choice of w or x_{R} until only one value of the key remains suggested. This method is easily generalized to any iterated cipher, and we get the following result, extending that of Theorem 11 of [6].

Theorem 1. *Given an iterated block cipher, let d denote the maximum polynomial degree of $m' \geq 1$ ciphertext bits of the round next to the last expressed as a function of* *the plaintext bits. Furthermore, let b denote the number of last-round key bits. Assume that the maximum polynomial degree of the m' bits resulting from a wrongly guessed value of the last-round key is larger than d and that the values obtained when computing the m' bits in these cases are uniformly distributed. Then there exists a dth-order differential attack of expected time complexity 2^{b+d} requiring $2^{d+1} \cdot g$ chosen plaintexts with $g = \lceil b/m' \rceil$ which will successfully recover the last-round key.*

Proof. We give the proof in the case of a Feistel cipher. The proof in the general case is similar. Consider iteration 3(b). Let k denote the correct value of the last-round key, and let k' denote any wrong value. Then

$$\begin{aligned} \tilde{y}_R &= y_L \oplus f(k, y_R), \\ \tilde{y}'_R &= y_L \oplus f(k', y_R) \\ &= \tilde{y}_R \oplus f(k, y_R) \oplus f(k', y_R). \end{aligned}$$

The difference between $\tilde{y}_R$, obtained using the correct key, and $\tilde{y}'_R$, obtained with a wrong key, is two applications of the function f. Therefore it seems plausible to assume that the maximum polynomial degree of the m' bits of $\tilde{y}'_R$ is larger than that of $\tilde{y}_R$. By the assumption an incorrect value of the last-round key will be suggested with probability $2^{-m'}$. After one attack about $2^{b-m'}$ values will be candidates for the correct key. Repeating the attack $g = \lceil b/m' \rceil$ times discards most wrong values of the key. The maximum time complexity of the attack is $2^{b+d} + 2^{b+d-m'} + 2^{b+d-2m'} + \cdots + 1$ which is at most 2^{b+d+1} when $m' \geq 1$. □

The attack can be improved by a factor of two, if the constant of (1) can be predicted. In that case iterations 2 and 3(b) of the above algorithm are performed only for all $a \in \mathrm{GF}(2^d)$. The key for which $\sigma = c$ will be the correct key with some high probability. For most ciphers, depending on the f-function, there are possible extensions to the above attack. It may be possible to perform the attack for only a subset of the bits in the last-round key, and it may also be possible to search for (subsets of) the first-round key.

3. The Interpolation Attack

In this section a new attack is introduced on block ciphers. The attack is based on the following well-known formula.

Let F be a field and let $2n$ elements $x_1, \ldots, x_n, y_1, \ldots, y_n \in F$ be given, where the x_i's are distinct. Define

$$f(x) = \sum_{i=1}^{n} y_i \prod_{1 \leq j \leq n, j \neq i} \frac{x - x_j}{x_i - x_j}. \tag{3}$$

Then $f(x)$ is the only polynomial over F of degree at most $n - 1$ such that $f(x_i) = y_i$ for $i = 1, \ldots, n$. Equation (3) is known as the *Lagrange interpolation formula* (see, e.g., page 185 of [2]).

In the *interpolation attacks* presented in this paper one constructs polynomials using some plaintexts and the corresponding ciphertexts. For these attacks it is not always necessary to find all the coefficients of the polynomial, merely to compute the value of $f(x)$ for one or a few values of x. In the following it is shown how to evaluate f in linear time in some point x (in particular $x = 0$) using n (other) evaluations of f.

Assume that f has degree at most $n - 1$. Also assume that there are given n points $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ where $x_i = \alpha^i$ for some primitive element $\alpha \in F$ and $y_i = f(x_i)$. Then to evaluate f in a point $x \neq x_1, x_2, \ldots, x_n$, we use the Lagrange interpolation formula:

$$f(x) = \sum_{i=1}^{n} y_i \prod_{1 \le j \le n, j \ne i} \frac{x - x_j}{x_i - x_j}.$$

Evaluating this expression for $x = 0$ yields

$$f(0) = \sum_{i=1}^{n} y_i \prod_{1 \le j \le n, j \ne i} \frac{x_j}{x_j - x_i} = \sum_{i=1}^{n} y_i \cdot \frac{g_i}{h_i},$$

where (with $x_i = \alpha^i$)

$$g_i = \frac{\alpha^{(1+2+\cdots+n)}}{\alpha^i} = \alpha^{n(n+1)/2-i}$$

and

$$h_i = \prod_{1 \le j \le n, j \ne i} (\alpha^j - \alpha^i).$$

Rewriting one gets

$$h_1 = \prod_{j=2}^{n} (\alpha^j - \alpha)$$

and the recurrence relation

$$h_{i+1} = h_i \cdot \frac{\alpha^{n-1}(\alpha^i - 1)}{\alpha^n - \alpha^i}.$$

In a similar way, more general formulas for computing $f(x)$ for arbitrary values of x can be derived. Using these expressions, it is possible to compute $f(0)$ (or more generally $f(x)$) in linear time $O(n)$ and constant memory.

Note that the values of g_i and h_i can be precomputed since they are independent of the y_i values.

Next it is explained how to utilize the above in attacks on block ciphers. Consider an m-bit secret-key block cipher for a fixed (unknown) key. The ciphertext y can be described as a polynomial $p(x) \in \mathrm{GF}(2^m)[x]$ of the plaintext. If the number of coefficients of these polynomials is sufficiently low, one can reconstruct it with sufficiently many plaintext/ciphertext (p/c) pairs by solving a simple system of linear equations. Subsequently, one has an algorithm which can encrypt and decrypt plaintexts and ciphertexts but without knowledge of the secret key.

In a chosen plaintext variant of this attack it is possible for an attacker to establish polynomials with a reduced number of coefficients by fixing some of the bits in the chosen plaintexts. In that case, the result is an instance deduction, since the obtained algorithm can only encrypt plaintexts for which a number of bits are fixed to a certain value.

Consider a key-recovery variant of the attack. Instead of specifying the ciphertext as a function of the plaintext, the output from the reduced cipher $\tilde{y}$ is expressed as a polynomial $p(x) \in \mathrm{GF}(2^m)[x]$ of the plaintext. Assume that this polynomial has degree d (hence has $d+1$ unknown coefficients) and that $d+2$ known p/c pairs are available. Then for all values of the last-round key one decrypts the ciphertexts one round, constructs the polynomial with $d+1$ pairs, and checks whether the polynomial is correct for the remaining p/c pair. If this is the case, then the correct value of the last-round key has been found with some high probability, by reasoning similarly as in the proof of Theorem 1. In this variant of the attack, it suffices to be able to construct a function value of the polynomial, and the method in the beginning of this section can be applied. Moreover, even if the key guess is wrong, the value of the polynomial may still be accepted with some probability. However, if it is assumed that for a wrongly guessed value of the last-round key the resulting polynomial has a degree higher than d and that the values obtained in trying to construct a unique polynomial are uniformly distributed, then the probability of success of the attack can be increased by simply checking additional pairs with the polynomial until only one value of the key remains.

One may also consider a multivariate generalization of the interpolation attack. An m-bit plaintext can be viewed as the concatenation of s sub-blocks each of m/s bits corresponding to elements in $\mathrm{GF}(2^{m/s})$. Accordingly, the ciphertext may then also be viewed as s sub-blocks of m/s bits. Each of these ciphertext sub-blocks is then expressible as a multivariate polynomial evaluated in the plaintext sub-blocks.

Note that an unknown polynomial of degree d has exactly $d+1$ unknown coefficients. However, as we demonstrate later, in some cases (for some ciphers) the number of unknown nonzero coefficients will be less than that. Therefore, in the following we use the number of nonzero coefficients rather than the polynomial degree of the involved polynomials to estimate the success of the interpolation attack.

The following result sums up the attack.

Theorem 2. *Consider an m-bit iterated block cipher with s sub-blocks each of m/s bits. Express an output sub-block from the penultimate round as a (uni- or multivariate) polynomial in $GF(2^{m/s})$ of some plaintext blocks and let n denote the number of nonzero coefficients in the polynomial. Furthermore, let b denote the number of last-round key bits involved in the attack. Assume that the number of nonzero coefficients in the polynomials resulting from a wrongly guessed value of the last-round key is larger than n and that values obtained in trying to construct a unique polynomial are uniformly distributed. Then there exists an interpolation attack of expected time complexity $2^b(n+1)$ requiring $n+\lambda$ known (or chosen) plaintexts with $\lambda = \lceil bs/m \rceil$ which will successfully recover the last-round key.*

Proof. Let k denote the correct value of the last-round key, and let k' denote any wrong value. Let $\tilde{y}$ and $\tilde{y'}$ be the text obtained from the ciphertexts by decrypting one round with the correct key k, respectively a wrong key k'. Then

$$\tilde{y} = g^{-1}(k, y),$$

$$\begin{aligned}\tilde{y}' &= g^{-1}(k', y)\\ &= g^{-1}(k', g(\tilde{y}, k)).\end{aligned}$$

The difference between $\tilde{y}$, obtained using the correct key, and $\tilde{y}'$, obtained with a wrong key, is one application of the round function g and one of its inverse with a wrong key. Therefore it seems plausible that for a wrongly guessed value of the key one will not succeed in generating a correct polynomial. The probability of guessing correctly a random m/s-bit value is $2^{-m/s}$. Repeating the feat λ times has probability $2^{-\lambda m/s}$. Therefore the expected number of false positives is $t = 2^{b-\lambda m/s}$ for which the expected number of texts needed follows. The maximum time complexity is $(2^b(n+1)+2^{b-m/s}(n+2) + 2^{b-2m/s}(n+3) + \cdots)$ which is at most $2^{b+1}(n+1)$ for $m \geq s$. □

Similar to the attack of Theorem 1 it may be possible to perform the attack for only a subset of the bits of the last-round key, and it may also be possible to search for (a subset of) the first-round key, depending on the structure of the round function. Similarly for some round functions it may be advantageous to solve algebraically for the round key in the last round instead of trying all possible 2^b values, as illustrated in [12].

In the above interpretation, the higher-order differential attack is a special case with $s = m$, 1-bit sub-blocks, and polynomials in m variables. However, note that in the higher-order differential attacks typically one attacks many 1-bit sub-blocks simultaneously.

For some ciphers it is possible to mount attacks for several, different values of s. It depends on the specific block cipher which of these attacks is the most efficient. In Section 4 examples are given to illustrate this.

Meet-in-the-Middle Approach

The attacks described in this section are extensions of the attacks in the previous sections using a meet-in-the-middle technique. Only the extension of the key-recovery attack is described; the extensions of the global and instance deductions follow easily.

Once again, one tries to guess the correct last-round key and use this to obtain (hopefully) $\tilde{y}$, the output from the reduced cipher. In the following, only the verification of $\tilde{y}$ is described. Given an iterated cipher of r rounds, let z denote the output of round r', where $r' \leq (r-1)$. The value of z is expressible via the plaintext x as a polynomial $h_1(x) \in \mathrm{GF}(2^m)[x]$ where m is the block size. Similarly, z can be expressed as a polynomial $h_2(\tilde{y}) \in \mathrm{GF}(2^m)[\tilde{y}]$ of the output $\tilde{y}$ of the reduced cipher. Let the degree of $h_1(x)$ be d_1, let the degree of $h_2(\tilde{y})$ be d_2 and let $d = d_1 + d_2$. Thus, the following equation,

$$h_1(x) = h_2(\tilde{y}), \qquad (4)$$

has at most $d+2$ nonzero unknowns. One can show that the equation is uniquely solvable up to a multiplication and an addition of both h_1 and h_2 with a constant. To ensure that a nontrivial and unique solution is obtained, the coefficient corresponding to the highest exponent is set equal to 1 and the constant term equal to 0. After this, the equation is solved by using d known or chosen plaintexts. What is left is to check whether yet another p/c pair $(x, \tilde{y})$ satisfies $h_1(x) = h_2(\tilde{y})$. If it does, then it is assumed that the correct value of the last-round key has been found.

The following result sums up the attack.

Theorem 3. *Consider an m-bit iterated block cipher with s sub-blocks each of m/s bits and with r rounds. Express the output from round r', $r' \leq r - 1$, as a multivariate polynomial of (some of) the plaintext blocks and let n_1 denote the number of coefficients in the polynomial. Also, express the output from round r' as a polynomial of the output sub-blocks from round $(r - 1)$, and let n_2 denote the number of coefficients in the polynomial. Furthermore, set $n = n_1 + n_2$ and let b denote the number of last-round key bits. Assume that the number of nonzero coefficients in the polynomials resulting from a wrongly guessed value of the last-round key is larger than n and that values obtained in trying to construct such polynomials are uniformly distributed. Then there exists an interpolation attack of expected time complexity $2^b(n + 1)$ requiring $n + g$ known (or chosen) plaintexts with $g = \lceil bs/m \rceil$ which will successfully recover the last-round key.*

The best known methods for solving a system of n linear equations in n unknowns that the authors are aware of, require $O(n^2)$ words of memory and run in time (approximately) $O(n^3)$. However, it is not necessary to solve the systems of equations, merely to have an algorithm which can detect whether a solution exists. Furthermore, (4) has a very special form and it is expected that special methods will exist which require less memory and time.

4. Examples

In this section the attacks of the previous sections are applied to a range of proposed block ciphers.

4.1. *Nyberg and Knudsen's Cipher*

Based on the use of a quadratic function over a Galois field, Nyberg and Knudsen demonstrated in [10] how to construct a cipher which is provably secure against differential cryptanalysis [1]. The cipher is a Feistel cipher with the nonlinear function f given by $F : \mathrm{GF}(2^{32}) \to \mathrm{GF}(2^{32})$ with

$$f(k, x) = d(h(e(x) \oplus k)),$$

where $h : \mathrm{GF}(2^{33}) \to \mathrm{GF}(2^{33})$, $h(x) = x^3$, $k \in \mathrm{GF}(2^{33})$, $e : \mathrm{GF}(2^{32}) \to \mathrm{GF}(2^{33})$ is a function which extends its argument by concatenation with an affine combination of the input bits, and $d : \mathrm{GF}(2^{33}) \to \mathrm{GF}(2^{32})$ discards one bit from its argument. With at least six rounds one can show that the probability of any differential can be bounded sufficiently low and subsequently there is a proof that this yields a secure cipher (with respect to conventional differential cryptanalysis). Also, the cipher is secure against the linear attack [8], which follows from [9].

In the following the higher-order differential attack is applied. Choose plaintexts where the right halves are fixed. Since the output bits from the round function are only quadratic in the input bits, the polynomial degree of the bits in the reduced cipher as a function of the plaintext bits is not higher than eight.

Therefore from Theorem 1 it follows that there exists an attack, which requires only $2^{8+1} = 512$ chosen plaintexts and an expected running time of order 2^{41}. A variant of the attack searching for the keys in the last two rounds requires about 32 chosen

Table 1. Higher-order differential attacks on the Nyberg–Knudsen cipher.

Number of rounds	Number of chosen plaintexts	Running time
6	2^9	2^{41}
6	2^5	2^{70}
7	2^{17}	2^{49}
7	2^9	2^{74}
8	2^{17}	2^{82}

plaintexts and an expected running time of order 2^{70}. Similarly, there are attacks on versions with seven and eight rounds, the complexities are given in Table 1. The attack has been implemented on scaled-down versions, and it recovers the last-round key as predicted.

Also, in [12] it was shown that for these attacks the round key in the last round can be solved for algebraically as opposed to trying all possible 2^b values.

4.2. *A Dedicated Cipher*

Consider the Feistel cipher with the round function given by $f(k, x) = h(x \oplus k)$ where $h : \mathrm{GF}(2^{32}) \to \mathrm{GF}(2^{32})$, $h(x) = x^3$, that is, the input to the cubing function is not extended and the output not truncated as in the previous case. This cipher is similar to the cipher in the previous section, and is also secure against the (conventional) differential attacks [10] and against the linear attack [8]. The cipher is as vulnerable to the higher-order differential attack as the previous one, but much more vulnerable to the interpolation attack for $s > 1$, as shown in the following.

Express the ciphertext halves after r rounds of encryption as polynomials of the plaintext halves in $\mathrm{GF}(2^{32})[x]$. It follows by easy calculations that these polynomials have at most $3^{2r-1} + 3^r + 3^{r-1} + 1$ nonzero coefficients. Note, that degrees of x_R and x_L are at most 3^r and 3^{r-1}, respectively. Thus, this polynomial can be reconstructed by considering at most $3^{2r-1} + 3^r + 3^{r-1} + 1$ p/c pairs using, e.g., Lagrange interpolation. With $r = 6$ the attack needs at most 2^{18} known p/c pairs, which yields an algorithm for a global deduction. Note that the number of coefficients will be lower than specified, since not all elements $x_\mathrm{L}^i x_\mathrm{R}^j$ for $0 \le i \le 3^r$ and $0 \le j \le 3^{r-1}$ will appear in the polynomial.

For the key-recovery attack with $r = 6$ assume that the right half x_R of the plaintext is fixed (that is, consider a chosen plaintext attack), and consider the right side of the output $\tilde{y}_\mathrm{R} = p(x_\mathrm{L})$ from the reduced cipher expressed as a polynomial $p(x_\mathrm{L}) \in \mathrm{GF}(2^{32})[x_\mathrm{L}]$. This polynomial has degree at most $3^3 = 27$ since the degree does not increase in the first round and since $\tilde{y}_\mathrm{R}$ equals the left half of the output of the fourth round. Consequently, 28 pairs of corresponding values of x_L and $\tilde{y}$ are enough to determine it uniquely (using Lagrange interpolation). It is then tested whether $\tilde{y}$ is actually output from the reduced cipher or not. This is done by verifying whether a 29th p/c pair agrees with the obtained polynomial. If it does, then it is assumed that the correct key has been found. The expected time complexity is $29 \times 2^{32-1} \approx 2^{36}$.

The meet-in-the-middle variant can be applied as follows. Let $r = 6$ and assume again that the right half x_R of the plaintext is fixed. Let z_L denote the left half of the output from round four. The value of z_L is expressible via the plaintext as a polynomial

$g(x_L) \in GF(2^{32})[x_L]$. This polynomial has degree at most 3^2, that is, there are at most 10 nonzero coefficients in $g(x_L)$. Similarly, z_L can be expressed as a polynomial $h(\tilde{y}_L, \tilde{y}_R) \in GF(2^{32})[\tilde{y}_L, \tilde{y}_R]$ of the output from the reduced cipher. It follows that $h(\tilde{y}_L, \tilde{y}_R) = \tilde{y}_L^3 \oplus a\tilde{y}_L^2 \oplus b\tilde{y}_L \oplus c \oplus \tilde{y}_R$, where a, b, and c are some key-dependent constants. Thus, there are at most $10 + 3 = 13$ unknown coefficients of the equation

$$g(x_L) = h(\tilde{y}_L, \tilde{y}_R). \tag{5}$$

Setting the constant term of g to equal 0 (the coefficient corresponding to the highest exponent in h has already been found to equal 1), one proceeds to solve the resulting system of equations by using 12 p/c pairs from the reduced cipher. Thus, one obtains the polynomials g and h. It is then checked whether yet another p/c pair $(x, \tilde{y})$ satisfies $g(x_L) = h(\tilde{y}_L, \tilde{y}_R)$. If it does, then it is assumed that the correct key has been found.

Similar attacks can be applied to versions of the cipher with up to 32 rounds at least in theory. Consider the version with 32 rounds. Let $g(x_L) \in GF(2^{32})[x_L]$ be an expression of the left half z_L of the output from round 22. The degree of this polynomial is at most 3^{20}. Let $h(\tilde{y}_L, \tilde{y}_R) \in GF(2^{32})[\tilde{y}_L, \tilde{y}_R]$ be an expression of z_L from the output of the reduced cipher. In the bivariate polynomial $h(\tilde{y}_L, \tilde{y}_R)$, the number of exponents in $\tilde{y}_L$ and $\tilde{y}_R$ is at most $(3^9 + 1)$ and $(3^{10} + 1)$, respectively. Thus, the number of coefficients in $h(\tilde{y}_L, \tilde{y}_R)$ is at most $(3^9 + 1)(3^{10} + 1) \approx 3^{19}$. This means that the number of coefficients in (5) is at most $3^{20} + 3^{19} \approx 2^{32}$.

4.3. *Attacks on Modified SHARK*

The iterated cipher SHARK is proposed by Rijmen et al. in [11]. The cipher has a block size of nm bits and each round has a nonlinear layer and a diffusion layer. The nonlinear layer consists of n parallel m-bit S-boxes. The diffusion layer consists of an nm-bit linear mapping constructed from a Reed–Solomon code. There are two suggestions for introducing the keys into the cipher. The first is by a simple exclusive-or with the inputs to the S-boxes, the other uses a key-dependent affine mapping. Also, an output transformation is applied after the last round of SHARK. The transformation consists of a key addition and an inverse diffusion layer.

Denote by SHARK(n, m, r) the version with a block size of nm bits using n parallel m-bit S-boxes in r rounds. In [11] an implementation SHARK$(8, 8, r)$ (64-bit blocks) is given. The eight S-boxes are identical and constructed from the permutation h : $GF(2^m) \to GF(2^m)$ given by $h(x) = x^{-1}$, with an affine mapping in the output bits. The cipher is analyzed with respect to linear and differential attacks, and it is argued that eight rounds of SHARK$(8, 8, r)$ give a security level comparable with that of triple-DES, and from Table 1 of [11] it follows that four rounds of this version give a security level comparable with that of DES.

In the following it is shown that there are many instances of SHARK that can be broken significantly faster than expected.

First, the number of rounds of SHARK must be determined with respect to the algebraic degree of the S-boxes. Assume that the outputs of the S-box are of degree d in the input bits. Since the S-boxes represent the only nonlinear component in SHARK, the algebraic degree of the ciphertexts after r rounds of encryption will be at most d^r. To avoid attacks based on higher-order differentials it must be ensured that d^r is high, preferably that

$d^r \geq nm$. Thus, for a 64-bit block cipher, if $d = 2$, e.g., using the cubing function in a Galois field, the number of rounds must be at least six.

Consider versions of SHARK where the keys are mixed with the texts by the exclusive-or operation. As will be shown there are instances of SHARK(n, m, r) for which the interpolation attacks are applicable. Consider 64-bit versions using as the S-box $h(x) = x^{-1}$ in GF(2^m). The inverse permutation in a Galois field has a high algebraic degree, note that $h(x) = x^{-1} = x^{2^m-2}$ in GF(2^m). However, as will be shown, the interpolation attack is applicable with a complexity which depends only on the number of S-boxes and on the number of rounds in the cipher. Note that the attacks presented are not applicable to the specific instance of SHARK from [11].

Consider first a version with $n = 1$. It follows by easy calculations that the ciphertext y *after any number of rounds* can be expressed as a fraction of polynomials of the plaintext x (or, similarly, x can be expressed as a polynomial of y) as follows:

$$y = \frac{x \oplus a}{bx \oplus c}, \tag{6}$$

where a, b, c are key-dependent constants. These three constants can be found using the interpolation attack with only four known p/c pairs by considering and solving $y \cdot (bx \oplus c) = (x \oplus a)$. The result is a global deduction, that is, an algorithm that encrypts (decrypts) any plaintext (ciphertext).

For $n > 1$ the number of coefficients in the polynomials used in the attacks increases with the number of diffusion layers in the cipher. Note that because of the inverse diffusion layer in the output transformation there are only $r - 1$ diffusion layers in an r-round version of SHARK. In the following consider a version of SHARK(n, m, r). Let the plaintext words each of m bits be denoted $x_1, \ldots, x_n$, and let the ciphertext words be denoted $y_1, \ldots, y_n$. Express the ciphertext words as polynomials in GF(2^m) in terms of the plaintext words. For $r = 1$, one gets expressions of the form $y_i = ax_i/(bx_i \oplus c)$. For $r = 2$ each ciphertext word can be written as a fraction of polynomials where in the denominator one gets an expression of degree at most n. Also, the degree of the polynomial in the enumerator is at most the degree of the polynomial in the denominator. Thus, the number of coefficients in the fraction of polynomials is at most 2×2^n. By doing similar calculations for $r = 3$ and so on, it follows that the number of coefficients in the polynomials for r rounds is at most

$$2 \cdot (n^{r-2} + 1)^n.$$

This is also the number of known plaintexts for the interpolation attack on an r-round version yielding a global deduction. It follows that the attack is independent of the sizes of the S-boxes, and depends only on the number of S-boxes and the number of rounds.

The interpolation attack with the meet-in-the-middle technique can also be applied for these ciphers. Consider the interpolation attack with known plaintexts. One first establishes

$$\frac{q_{j,1}(y_1, \ldots, y_n)}{q_{j,2}(y_1, \ldots, y_n)} = \frac{p_{i,1}(x_1, \ldots, x_n)}{p_{i,2}(x_1, \ldots, x_n)}, \tag{7}$$

that is, expressions of the ciphertexts in one middle round, where $i + j = r - 1$, using polynomials of both the plaintext and the ciphertext. Subsequently, one can solve the

Table 2. Complexities of the interpolation attack on variants of SHARK using as S-box $h(x) = x^{-1}$.

Number of rounds	Number of S-boxes	Known plaintexts	Memory	Time
Any	1	3		
6	2	2^9	2^{18}	2^{27}
6	4	2^{27}	2^{54}	2^{81}
3	8	2^{17}	2^{34}	2^{51}
4	8	2^{35}	2^{70}	2^{105}
5	8	2^{52}	2^{104}	2^{156}
6	8	2^{75}	2^{150}	2^{225}

following systems of equations:

$$q_{j,1}(y_1, \ldots, y_n) \cdot p_{i,2}(x_1, \ldots, x_n) = p_{i,1}(x_1, \ldots, x_n) \cdot q_{j,2}(y_1, \ldots, y_n). \tag{8}$$

The number of known plaintexts required to solve (8) is

$$2 \cdot (n^{r_1 - 1} + 1)^n \cdot (n^{r_2 - 1} + 1)^n,$$

where $r_1 + r_2 = r - 1$ and $r_1, r_2 \geq 1$, which follows by calculations similar to the above.

The round keys for SHARK are typically quite big, so the general key-recovery attack described earlier in this paper may be impractical. However, it is possible to perform the attack for only a subset of the first-round and/or last-round keys. As an example, one can repeat the attack for all values of the first s words of the first-round key and express the ciphertext (of a middle round) as a polynomial $p_{i,1}(S(x_1 \oplus k_1), \ldots, S(x_s \oplus k_s), x_{s+1}, \ldots x_n)$, where $S(\cdot)$ are the S-boxes and x_i are the plaintext words. The values of the key words for which the interpolation succeeds are candidates for the secret key, and the attack is repeated sufficiently many times until one value of the secret key is found.

Table 2 gives the complexities of the interpolation attack on variants of SHARK using as the S-box $h(x) = x^{-1}$ in GF(2^m). It follows that using eight S-boxes, a 64-bit block variant with up to five rounds and a 128-bit block variant with up to eight rounds are, at least in theory, vulnerable to our attacks. The required amount of memory and time for the versions with five and six rounds are of course unrealistic today. However, as discussed earlier, the linear equations obtained in the attacks are of a very special form. Therefore there might exist methods solving such systems faster than for systems of arbitrary linear equations. Furthermore, the complexities were computed assuming that the number of coefficients in the polynomials are maximum. A closer analysis might reveal that this number is smaller. Also, the attacks can be faster than an exhaustive search for the keys depending on the chosen key length. In a chosen plaintext attack the number of coefficients in the polynomials used in the attack can be reduced by fixing some plaintext bits. As examples, there exist interpolation attacks on the variant with eight S-boxes and four rounds using about 2^{21} chosen plaintexts and on the variant with eight S-boxes and seven rounds using about 2^{61} chosen plaintexts.

It has been demonstrated that certain instantiations of SHARK are insecure. The results also demonstrate a case where the use of bigger and fewer S-boxes does not result in more secure ciphers. Finally, it is noted that the designers of SHARK expressed their concern with the use of the inverse in a Galois field as S-boxes [11].

4.4. *Kiefer's Scheme*

In this section the scheme by Kiefer [4] is attacked in a higher-order differential attack. The cipher is probabilistic and uses the following encryption rule:

$$x_i \mapsto (F(k) \oplus r_i,\ f_k(r_i) \oplus x_i), \tag{9}$$

where $F : \mathrm{GF}(2^m) \to \mathrm{GF}(2^m)$ is a one-way function, $f_k : \mathrm{GF}(2^m) \to \mathrm{GF}(2^m)$ is a function depending on the key $k \in \mathrm{GF}(2^m)$ in some complex way, $r_i \in \mathrm{GF}(2^m)$ is a random value, and $x_i \in \mathrm{GF}(2^m)$ is a message block. The function f_k has the form $f_k = \pi_k \circ g$ where $\pi_k : \mathrm{GF}(2^m) \to \mathrm{GF}(2^m)$ is a bitwise linear transform depending on k, and $g : \mathrm{GF}(2^m) \to \mathrm{GF}(2^m)$ is a public, almost perfectly nonlinear, function of the form $g(x) = x^{2^s+1}$ for some s.

Assume that enough plaintext is available to have four pairs of ciphertexts of the form

$$(a_i, b_i) = (F(k) \oplus r_i,\ f_k(r_i)), \qquad i = 1, \ldots, 4, \tag{10}$$

such that $a_1 \oplus a_2 = a_3 \oplus a_4$. Define $\beta = \bigoplus_{i=1}^4 b_i$ and $\gamma = \bigoplus_{i=1}^4 g(r_i)$. Then

$$\beta = \bigoplus_{i=1}^{4} b_i = \pi_k\left(\bigoplus_{i=1}^{4} g(r_i)\right) = \pi_k(\gamma). \tag{11}$$

Since $\{a_1, \ldots, a_4\}$ is a two-dimensional subspace of $\mathrm{GF}(2^n)$, the elements in $\{r_1, \ldots, r_4\}$ also constitute a two-dimensional subspace which, with a fixed key, is parallel to the first one. Note also that the Hamming weight of the exponent in the definition of g expressed as a binary number is only two, implying that the output bits are only quadratic in the input bits. By (1), this implies that one can compute the value of $\gamma = \bigoplus_{i=1}^4 g(a_i)$.

If repeated m times, one obtains m corresponding pairs of β and γ. This makes it possible to solve (11) with respect to the unknown function π_k (it is a linear transform). After having found π_k, invert f_k and thus obtain a value of r_i. Subsequently, compute $F(k)$ and the system is broken.

It remains to compute the minimum number t of known plaintexts needed to obtain m times four pairs (a_i, b_i) with the required property; recall that the cipher is probabilistic and thus the attacker has no control over the values of r_i. By using a birthday paradox type argument it can be shown that $t \approx (m \cdot 2^{m+2})^{1/4}$. For a typical block size of $m = 64$ this gives $t \approx 2^{18}$.

5. Concluding Remarks

A new attack on block ciphers, the interpolation attack, was introduced. The interpolation attack is a natural extension of the higher-order differential attack, but in many cases much more efficient than the latter. It was demonstrated that the attack works on several proposed block ciphers. In particular, it was shown that a cipher provably secure against differential and linear cryptanalysis is very vulnerable to the interpolation attack. Also, variants of the attack were used to cryptanalyze the (unmodified) cipher by Nyberg and Knudsen, variants of the cipher SHARK, and a cipher by Kiefer.

Recently, a probabilistic version of the interpolation attack was introduced [3]. This version of the attack finds a polynomial relation between plaintexts and ciphertexts which hold only for a fraction of all cases. It was demonstrated that this attack can also be used to break the cipher by Nyberg and Knudsen [10].

Acknowledgments

The authors thank Vincent Rijmen and the anonymous referees for helpful comments.

References

[1] E. Biham and A. Shamir. *Differential Cryptanalysis of the Data Encryption Standard*. Springer-Verlag, New York, 1993.

[2] P.M. Cohn. *Algebra*, Volume 1. Wiley, New York, 1982.

[3] T. Jakobsen. Cryptanalysis of block ciphers with probabilistic non-linear relations of low degree. In H. Krawczyk, editor, *Advances in Cryptology*: *CRYPTO* '98, LNCS 1462, pages 212–222. Springer-Verlag, Berlin, 1998.

[4] K. Kiefer. A new design concept for building secure block ciphers. In J. Pribyl, editor, *Proceedings of the* 1*st International Conference on the Theory and Applications of Cryptology*, *PRAGOCRYPT* '96, *Prague*, *Czech Republic*, pages 30–41. CTU, Prague, 1996.

[5] L.R. Knudsen. Block Ciphers – Analysis, Design and Applications. Ph.D. thesis, Aarhus University, Denmark, 1994.

[6] L.R. Knudsen. Truncated and higher order differentials. In B. Preneel, editor, *Fast Software Encryption - Second International Workshop*, *Leuven*, *Belgium*, LNCS 1008, pages 196–211. Springer-Verlag, Berlin, 1995.

[7] X. Lai. Higher order derivatives and differential cryptanalysis. In R. Blahut, editor, *Communication and Cryptography*, *Two Sides of One Tapestry*. Kluwer, Dordrecht, 1994. ISBN 0-7923-9469-0.

[8] M. Matsui. Linear cryptanalysis method for DES cipher. In T. Helleseth, editor, *Advances in Cryptology - EUROCRYPT* '93, LNCS 765, pages 386–397. Springer-Verlag, Berlin, 1993.

[9] K. Nyberg. Linear approximations of block ciphers. In A. De Santis, editor, *Advances in Cryptology - EUROCRYPT* '94, LNCS 950, pages 439–444. Springer-Verlag, Berlin, 1995.

[10] K. Nyberg and L.R. Knudsen. Provable security against a differential attack. *Journal of Cryptology*, 8(1):27–38, 1995.

[11] V. Rijmen, J. Daemen, B. Preneel, A. Bosselaers, and E. De Win. The cipher SHARK. In D. Gollmann, editor, *Fast Software Encryption*, *Third International Workshop*, *Cambridge*, *UK*, *February* 1996, LNCS 1039, pages 99–112. Springer-Verlag, Berlin, 1996.

[12] T. Shimoyama, S. Moriai, and T. Kaneko. Improving the higher order differential attack and cryptanalysis of the KN cipher. Presented at Information Security Workshop '97, ISW 97, Kanazawa, Sept. 1997.