Designing Against the ‘Overdefined System of Equations’ Attack

Carlisle Adams
University of Ottawa
Ottawa, Ontario, Canada K1N 6N5
cadams@site.uottawa.ca

Abstract. Recently, Courtois and Pieprzyk proposed an attack on symmetric ciphers that takes advantage of a previously-unexploited property of substitution boxes, or s-boxes, in the round function. This paper gives a brief overview of this “overdefined system of equations” attack and shows how the attack may be avoided through the use of round functions that contain a variety of protection mechanisms, including combinations of operators from different algebraic groups, a circular rotation step, and substitution boxes (s-boxes) of large dimension.

Keywords: symmetric cipher design, cryptanalysis, substitution box, round function

1 Introduction

The recent attack by Courtois and Pieprzyk on a class of symmetric ciphers [6] exploits s-boxes whose polynomial representation creates a system of equations in a set of unknowns that is “overdefined” in the sense that it contains more than the minimum amount of information required to determine the unique solution. If the system of equations is not only overdefined, but also sparse (i.e., contains a relatively small number of monomial terms), then an algorithm referred to as XSL [6] (a modification of the XL algorithm [11]) may be used to solve the equations for the unknown key bits and break the cipher with lower complexity than exhaustive search over the key space. Although an explicit demonstration has not yet been published, Courtois and Pieprzyk claim that their Overdefined System of Equations (OSE) attack has shown some success against both Rijndael (AES) [7] and Serpent [4].

Since the publication of the OSE attack, some controversy has arisen over the actual effectiveness of the XSL algorithm (see, for example, [5, 8, 9]). This paper does not attempt to take a position on that debate. Rather, the formulation of the attack itself is described with respect to a generic block cipher employing s-boxes for the “confusion” component [12]. The attack is then examined with regard to its effectiveness against a design procedure used in the CAST family of ciphers [1–3]. It is shown that ciphers constructed according to this design procedure are not susceptible to the OSE attack. This lends support to the conjecture that mixing operations from different algebraic groups is an important design criterion for strengthening ciphers against a variety of attacks.

The remainder of the paper is organized as follows. Section 2 describes the OSE attack at a high level, without going into detail on the workings of the XSL algorithm. Section 3 shows how the CAST design procedure, and the CAST-128 cipher in particular, are immune to this attack. Section 4 concludes the paper and discusses the significance of its main results.

2 The OSE Attack

Let $S()$ be an s-box with m input bits and n output bits so that $y_1 y_2 \dots y_n = S(x_1, x_2, \dots, x_m)$, or $y = S(x)$. When analyzing cryptographic properties of an s-box such as nonlinearity or strict avalanche criterion [13], designers and cryptanalysts are typically interested in the mapping from x to y as a multivariate function ($y = f(x)$) or as a collection of Boolean functions ($y_i = f(x), 1 < i < n$). In addition to such analysis, Courtois and Pieprzyk propose examining the polynomial formed from all the input and output variables: $p(x_1, x_2, \dots, x_m, y_1, y_2, \dots, y_n)$. For a given degree, the polynomial has the obvious generic form. For example, the quadratic polynomial for a 3×3 s-box has the form

$$\begin{aligned} P = p(x,y) = {} & x_1 \oplus x_2 \oplus x_3 \oplus y_1 \oplus y_2 \oplus y_3 \oplus \\ & x_1x_2 \oplus x_1x_3 \oplus x_1y_1 \oplus x_1y_2 \oplus x_1y_3 \oplus x_2x_3 \oplus \\ & x_2y_1 \oplus x_2y_2 \oplus x_2y_3 \oplus x_3y_1 \oplus x_3y_2 \oplus x_3y_3 \oplus \\ & y_1y_2 \oplus y_1y_3 \oplus y_2y_3 \oplus 1 \end{aligned}$$

The number of terms in this generic quadratic form is $T = \binom{m+n}{2} + m + n + 1$, which is 22 for a 3×3 s-box.

For a specific s-box, S_1, a $2^m \times T$ binary matrix M can be constructed where each row, m_i, of the matrix is the polynomial P with the component terms evaluated according to the definition of S_1. An example will help to illustrate the construction of M. Let S_1 be the 3×3 s-box defined as follows:

Input	Output
0	6
1	3
2	1
3	5
4	0
5	7
6	4
7	2

which, in binary, is equivalent to

x_1	x_2	x_3	y_1	y_2	y_3
0	0	0	1	1	0
0	0	1	0	1	1
0	1	0	0	0	1
0	1	1	1	0	1
1	0	0	0	0	0
1	0	1	1	1	1
1	1	0	1	0	0
1	1	1	0	1	0

Then m_1 is the component terms of P evaluated with $x_1 = 0, x_2 = 0, x_3 = 0, y_1 = 1, y_2 = 1$, and $y_3 = 0$ (i.e., the first row of S_1):

$$m_1 = 0001100000000000001001$$

The row m_2 is similarly evaluated with $x_1 = 0, x_2 = 0, x_3 = 1, y_1 = 0, y_2 = 1$, and $y_3 = 1$. The full 8×22 matrix M that corresponds to S_1 is thus

$$M = \begin{bmatrix} 0\,0\,0\,1\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,1 \\ 0\,0\,1\,0\,1\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,0\,0\,1\,1 \\ 0\,1\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,1 \\ 0\,1\,1\,1\,0\,1\,0\,0\,0\,0\,0\,1\,1\,0\,1\,1\,0\,1\,0\,1\,0\,1 \\ 1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1 \\ 1\,0\,1\,1\,1\,1\,0\,1\,1\,1\,1\,0\,0\,0\,0\,1\,1\,1\,1\,1\,1\,1 \\ 1\,1\,0\,1\,0\,0\,1\,0\,1\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,1 \\ 1\,1\,1\,0\,1\,0\,1\,1\,0\,1\,0\,1\,0\,1\,0\,0\,1\,0\,0\,0\,0\,1 \end{bmatrix}$$

This matrix has rank, r, at most 8 (the smaller of the row and column dimensions) and so there are at most 8 linearly independent columns c_I and at least $22 - 8 = 14$ dependent columns c_D, each of which is equal to some linear combination of the c_I. Thus,

$$c_{D_i} = a_{i_1} c_{I_1} \oplus a_{i_2} c_{I_2} \oplus \ldots \oplus a_{i_8} c_{I_8}, \quad a_{i_j} \in \{0, 1\}, \quad 1 \leq i \leq (T - r)$$

(Note that M may be converted to Row Reduced Echelon [10] form, or any other convenient form, using standard binary matrix reductions so that the c_{D_i} have a simple desired representation.) Given the rank, r, of M, the c_{D_i} define a set of $T - r \geq 14$ simultaneous equations in six unknowns $(x_1, x_2, x_3, y_1, y_2, y_3)$ which is called "overdefined" because there are more equations than are theoretically required to determine an exact solution. This set of equations is also said to be "sparse" (and is claimed to be significantly easier to solve) if a reasonable number of the a_{i_j} are equal to 0.

Depending upon how the s-boxes are incorporated into the cipher round function, it may be possible to string the c_{D_i} together from one round to the next. In many designs, the i^{th} s-box input at round ℓ is equal to the XOR sum of a plaintext bit and a round key bit, $x_{i_\ell} = b_{i_\ell} \oplus k_{i_\ell}$. Thus, in a known or chosen plaintext attack, the unknown variables in the first-round c_{D_i} equations are round-one key bits. The s-box outputs may be expanded and/or permuted prior to being input to the next round. The c_{D_i} solutions at round ℓ, therefore, become the appropriate input values to the c_{D_i} equations at round $\ell + 1$. For example, imagine a matrix M for an s-box S_1 whose independent columns are $x_1, x_2, x_3, x_1x_2, x_1x_3, x_2x_3, 1$, and y_1y_3. Furthermore, for the sake of simple illustration, assume a 2-round cipher whose round function contains only S_1 and whose permutation layer connects the outputs y_1, y_2, and y_3 from round 1 to the inputs x_3, x_2, and x_1, respectively, in round 2. Let three of the c_{D_i} equations be as follows:

$$\begin{aligned} y_1 &= x_2 \oplus x_1x_3 \oplus 1 \\ y_2 &= y_1y_3 \oplus x_1 \oplus x_1x_3 \\ y_3 &= x_2x_3 \oplus x_1x_2 \end{aligned}$$

Then, at the output of round 2,

$$\begin{aligned}
y_2 &= y_1y_3 \oplus x_1 \oplus x_1x_3 \\
&= y_1y_3 \oplus [[x_2x_3 \oplus x_1x_2] \oplus k_{1,2}] \oplus [[x_2x_3 \oplus x_1x_2] \oplus k_{1,2}] \cdot [x_2 \oplus x_1x_3 \oplus 1] \oplus k_{3,2}] \\
&= y_1y_3 \oplus [[(b_2 \oplus k_{2,1})(b_3 \oplus k_{3,1}) \oplus (b_1 \oplus k_{1,1})(b_2 \oplus k_{2,1})] \oplus k_{1,2}] \oplus \\
&\quad [[(b_2 \oplus k_{2,1})(b_3 \oplus k_{3,1}) \oplus (b_1 \oplus k_{1,1})(b_2 \oplus k_{2,1})] \oplus k_{1,2}] \cdot [[(b_2 \oplus k_{2,1}) \oplus (b_1 \oplus k_{1,1})(b_3 \oplus k_{3,1}) \oplus 1] \oplus k_{3,2}]
\end{aligned}$$

(The second line comes from replacing x_1 and x_3 in round 2 with y_3 and y_1 from the output of round 1, and adding in the relevant round 2 key bits $k_{i,2}$. The third line comes from replacing x_1, x_2, x_3 in round 1 with the plaintext bits XORed with the relevant round 1 key bits $k_{i,1}$.) Finally, replacing the b_i and the y_i with the appropriate values from the known or chosen plaintext/ciphertext pair allows the attacker to determine information about some key bits. Using many different c_{D_i} equations over many plaintext/ciphertext pairs may result in significant cost savings over exhaustive search in breaking the cipher.

The primary and overwhelming benefit of the OSE attack compared with many other attacks such as linear and differential cryptanalysis and their variants is that the relationships computed between input and output bits hold with equality rather than with some smaller probability. Probabilistic relationships lead to attack complexities that grow exponentially with the number of rounds (because the probabilities multiply), whereas equality relationships lead to attack complexities that grow linearly or polynomially with the number of rounds. This means that a cipher broken by the OSE attack cannot easily be repaired by the addition of rounds.

Thus, the OSE attack is, in a sense, more devastating to a cipher than previous attacks and should be considered when a new cipher is being designed. To that end, it is important to define some design criteria that are guaranteed to render the OSE attack ineffective as a cryptanalytic tool. This is the goal of the following section.

3 OSE and the CAST Round Function

The CAST design procedure [1] uses large $m \times n$ s-boxes with $m \ll n$ and uses operations from different algebraic groups both to combine plaintext bits with key bits, and to combine the outputs from the round function s-boxes. As a concrete instantiation of this design procedure, the CAST-128 encryption algorithm [1, 2] specifies three different round functions that are used on a rotating basis throughout the cipher as follows (see [1, 2] for details):

$$
\begin{aligned}
f_1 : I &= ((k_m + b) \hookleftarrow k_r) \\
f &= (((S_1[I_a] \oplus S_2[I_b]) - S_3[I_c]) + S_4[I_d]) \\
\\
f_2 : I &= ((k_m \oplus b) \hookleftarrow k_r) \\
f &= (((S_1[I_a] - S_2[I_b]) + S_3[I_c]) \oplus S_4[I_d]) \\
\\
f_3 : I &= ((k_m - b) \hookleftarrow k_r) \\
f &= (((S_1[I_a] + S_2[I_b]) \oplus S_3[I_c]) - S_4[I_d])
\end{aligned}
$$

Here, b, k_m, I, and the output f are 32-bit words; I_a, I_b, I_c, and I_d are the first, second, third, and fourth bytes of I, respectively; S_1 - S_4 are 8×32 s-boxes; "$\oplus$" is addition modulo 2 (XOR); "+" and "-" are addition and subtraction modulo 2^{32}; and $\hookleftarrow$ is circular left rotation by a 5-bit value k_r.

3.1 Complexity due to S-Box Dimension

With respect to the OSE attack by Courtois and Pieprzyk, there are two things to notice about the CAST-128 s-boxes. First, at size 8 × 32, each s-box can certainly be described by an overdefined system of equations. In particular, the matrix M describing each s-box will have $2^8 = 256$ rows and $\binom{8+32}{2} + 8 + 32 + 1 = 821$ columns. Thus, the rank of M will be at most 256 and the number of dependent columns c_D will be at least $821 - 256 = 565$. However, the second observation is that according to Courtois and Pieprzyk, the complexity of their OSE attack has a constant factor that is doubly exponential in the size of the s-box [6]. Thus, unless S has exploitable degenerate properties, dimensions greater than 4 or 5 render the attack computationally infeasible. This leads to the conclusion that the 8 × 32 s-boxes in CAST-128 are immune to the OSE attack.

3.2 Uncertainty in Estimating S-Box Inputs

It is interesting to note that s-box size is not CAST's only, or even primary, defense against the OSE attack. As noted above, in many symmetric ciphers the round key is XORed with the plaintext bits immediately prior to input to the s-boxes in each round. This means that in the j^{th} round, the i^{th} input bit $x_{i,j}$ can be replaced with $(b_{i,j} \oplus k_{i,j})$ in a c_D equation. The quadratic equation c_D can still be evaluated modulo 2 and the relationship that it specifies between s-box bits still holds with equality. In the CAST round function, the plaintext and key may instead be combined using addition or subtraction modulo 2^{32}. For the case of addition (the analysis for subtraction is similar), $x_{i,j}$ can no longer be replaced with $(b_{i,j} \oplus k_{i,j})$, but must instead be replaced with $(b_{i,j} \oplus k_{i,j} \oplus \pi_{i,j})$, where $\pi_{i,j}$ is the carry value resulting from the sum $(b_{(i-1),j} + k_{(i-1),j} + \pi_{(i-1),j})$. In general,

$$\pi_{p,j} = \left\{ \begin{array}{lll} 1 & : & (b_{(p-1),j} + k_{(p-1),j} + \pi_{(p-1),j}) \geq 2 \\ 0 & : & \text{otherwise} \end{array} \right\}, \qquad 1 \leq p \leq i$$

$$\pi_{p,j} = \quad 0, \qquad p = 0$$

Thus, $prob(x_{i,j} = (b_{i,j} \oplus k_{i,j})) = prob(\pi_{i,j} = 0) = (1 - prob(\pi_{i,j} = 1))$. Now $prob(\pi_{0,j} = 1) = 0$ by definition. It is easy to see that $prob(\pi_{1,j} = 1) = prob(b_{0,j} = 1 \quad \text{AND} \quad k_{0,j} = 1) = \left(\frac{1}{2} \cdot \frac{1}{2}\right) = \frac{1}{4}$, for random vectors b and k. Continuing, we see that

$$\begin{aligned} prob(\pi_{2,j} = 1) = prob((&b_{1,j} = 1 \quad \text{AND} \quad k_{1,j} = 1 \quad \text{AND} \quad \pi_{1,j} = 0) \qquad \text{OR} \\ (&b_{1,j} = 1 \quad \text{AND} \quad k_{1,j} = 0 \quad \text{AND} \quad \pi_{1,j} = 1) \qquad \text{OR} \\ (&b_{1,j} = 0 \quad \text{AND} \quad k_{1,j} = 1 \quad \text{AND} \quad \pi_{1,j} = 1) \qquad \text{OR} \\ (&b_{1,j} = 1 \quad \text{AND} \quad k_{1,j} = 1 \quad \text{AND} \quad \pi_{1,j} = 1)) \\ &= \left(\left(\frac{1}{2} \cdot \frac{1}{2} \cdot \frac{3}{4}\right) + \left(\frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{4}\right) + \left(\frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{4}\right) + \left(\frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{4}\right) \right) \\ &= \left(\frac{3}{16} + \frac{1}{16} + \frac{1}{16} + \frac{1}{16} \right) \\ &= \frac{3}{8} \end{aligned}$$

In general, we have $prob(\pi_{i,j} = 1) = \left(\frac{2^i - 1}{2^{i+1}}\right)$, $1 \leq i \leq 31$, which approaches $\frac{1}{2}$ rapidly as i grows.

The presence of the carry values in the c_D equations changes the nature of these equations dramatically since the relationships described by those equations now hold with some probability, rather than with equality. That is, an equation such as $y_1 = x_{1,j} \oplus x_{2,j} x_{3,j} \oplus 1$ becomes $y_1 = (b_{1,j} \oplus k_{1,j}) \oplus (b_{2,j} \oplus k_{2,j}) \cdot (b_{3,j} \oplus k_{3,j}) \oplus 1$ with a probability governed by the probability that $\pi_{1,j}, \pi_{2,j}$, and $\pi_{3,j}$ are simultaneously equal to zero. This probability is approximately $\frac{1}{8}$ (the exact value depends on the specific (i, j) involved in the equation), rather than 1 (which is the case when no carry values are possible). If v variables are present in a c_D equation, the probability that the carry values are all simultaneously zero is 2^{-v}, which means that the carries cannot be ignored in the analysis.

In addition, from the specification of the CAST-128 round functions above it is clear that another key, k_r, is also involved in the modification of the round function input data. That is, $x_{i,j}$ is not simply the sum of $b_{i,j}, k_{i,j}$, and $\pi_{i,j}$ (where $k_{i,j}$ is the i^{th} bit in the j^{th}-round masking key), but $x_{i,j}$ is a circular rotation of that sum by the value of k_r. Taking the carry $\pi_{i,j}$ into account means that $x_{i,j}$ depends on $b_{i,j}, b_{(i-1),j}, \ldots, b_{0,j}$ and $k_{i,j}, k_{(i-1),j}, \ldots, k_{0,j}$. However, because the rotation amount could be any value, it follows that $x_{i,j}$ could depend on all $b_{i,j}$ and all $k_{i,j}$. Therefore, it is nessary to know all of b_i, all of k_i, and all of k_r in order to know any bit of x_i with certainty. Without such full knowledge, then, nothing exact can be known about any s-box inputs.

3.3 Uncertainty in Estimating Round Function Outputs

Turning now to the s-box outputs, the f_i above show that these are combined using addition modulo 2, and addition and subtraction modulo 2^{32}. Again, addition and subtraction modulo 2^{32} involve carry (or borrow) propagation, and so if the four s-box output vectors are not known fully then round function output bits cannot be known with certainty. To illustrate, let the four s-box output vectors be $y1, y2, y3$, and $y4$. For round function f_3 as described above (f_1 and f_2 are similar), $f = (((y1 + y2) \oplus y3) - y4) = (((y1 + y2) \oplus y3) + (-y4))$, since subtraction is accomplished as a negation of $y4$ (using 2s complement) followed by an addition. The 2s complement representation is defined to be 1s complement (a simple complement of every bit) plus 1. This is equivalent to leaving all least significant bits (in the original vector) up to and including the first "1" alone, and complementing all the higher-order bits. Thus, in the original vector, bit i stays the same if and only if every less significant bit is a 0 bit, so that *prob*(bit i unchanged) $= 2^{-i}$, for $1 \leq i \leq 31$. Therefore, with probability 2^{-i}, bit i will remain unchanged, and with probability $(1 - 2^{-i}) = \left(\frac{2^i-1}{2^i}\right)$ it will be complemented.

Looking at the i^{th} bit of the round function output, we have

$$\begin{aligned} f &= (((y1_i \oplus y2_i \oplus \pi_a) \oplus y3_i) \oplus (y4_i \oplus \pi_b) \oplus \pi_c) \\ &= y1_i \oplus y2_i \oplus y3_i \oplus y4_i \oplus (\pi_a \oplus \pi_b \oplus \pi_c) \end{aligned}$$

where π_a is the carry that pertains to bit i from the addition of $y1$ and $y2$, π_b is the carry that pertains to bit i from the 2s complement representation of $y4$, and π_c is the carry that pertains to bit i from the addition of the complemented $y4$ with the result of $((y1 + y2) \oplus y3)$. If an attacker has only partial information about the s-box outputs, such as $y1_i, y2_i, y3_i$, and $y4_i$, then the attacker's approximation of the i^{th} output bit of the round function by $y1_i \oplus y2_i \oplus y3_i \oplus y4_i$ will only be accurate if $\pi = (\pi_a \oplus \pi_b \oplus \pi_c)$ is zero. Now

$$\begin{aligned} prob(\pi = 0) &= prob((\pi_a = 0, \pi_b = 0, \pi_c = 0) \quad \text{OR} \quad (\pi_a = 1, \pi_b = 1, \pi_c = 0) \quad \text{OR} \\ &\qquad (\pi_a = 1, \pi_b = 0, \pi_c = 1) \quad \text{OR} \quad (\pi_a = 0, \pi_b = 1, \pi_c = 1)) \\ &= \left[\frac{2^i+1}{2^{i+1}} \cdot \frac{1}{2^i} \cdot \frac{2^i+1}{2^{i+1}}\right] + \left[\frac{2^i-1}{2^{i+1}} \cdot \frac{2^i-1}{2^i} \cdot \frac{2^i+1}{2^{i+1}}\right] + \\ &\quad \left[\frac{2^i-1}{2^{i+1}} \cdot \frac{1}{2^i} \cdot \frac{2^i-1}{2^{i+1}}\right] + \left[\frac{2^i+1}{2^{i+1}} \cdot \frac{2^i-1}{2^i} \cdot \frac{2^i-1}{2^{i+1}}\right] + \\ &\approx \left[\frac{1}{2} \cdot \frac{1}{2^i} \cdot \frac{1}{2}\right] + \left[\frac{1}{2} \cdot 1 \cdot \frac{1}{2}\right] + \\ &\quad \left[\frac{1}{2} \cdot \frac{1}{2^i} \cdot \frac{1}{2}\right] + \left[\frac{1}{2} \cdot 1 \cdot \frac{1}{2}\right] \\ &= \left(\frac{1}{2^{i+2}} + \frac{1}{4} + \frac{1}{2^{i+2}} + \frac{1}{4}\right) \\ &= \left(\frac{1}{2} + \frac{1}{2^{i+1}}\right) \end{aligned}$$

Thus, the attacker can determine the i^{th} bit of the round function output from his partial knowledge of the s-box outputs with an advantage of approximately $\frac{1}{2^{i+1}}$ over a random guess. Clearly this advantage is negligible for most bits of the output vector. Although it is the case that for low order bits this advantage may be significant (in particular, for the least significant bit, $\pi = 0$ with probability 1), it is infeasible for an attacker to exploit this over multiple rounds because at the input stage of the next round, this vector will be rotated by an unknown amount (determined by the rotation key k_r) and so is highly unlikely to remain in a low order bit position.

3.4 Summary of Protection Against the OSE Attack

As outlined above in this section, the CAST design procedure incorporates several criteria that contribute in a significant way to its immunity against the OSE attack. These may be summarized as follows.

- The use of addition and subtraction modulo 2^{32} in the data and key combining stage means that anything less than full knowledge of all the inputs to the round function leads to only probabilistic knowledge of each of the s-box inputs.
- Probabilistic knowledge of the s-box inputs leads to a bad estimate of the s-box outputs because the CAST s-boxes use bent functions, which guarantee that each output bit depends in a complex, highly-nonlinear way on *all* input bits (not just a proper subset); see [1] for details.
- Less than full knowledge of s-box outputs leads to probabilistic knowledge of the round function outputs because of the use of addition and subtraction modulo 2^{32} in the s-box output combining stage.
- The use of multiple different round functions means that adjacent round functions behave differently (this makes it difficult to build *characteristics*, which are specific round function attacks chained together over multiple consecutive rounds).
- The presence of the circular rotation operation also ensures that building *characteristics* is difficult because the output bits from one round are shifted by an unknown (key-determined) value prior to being input to s-boxes in the following round.
- The use of s-boxes of large dimension ensures that the calculations required for the OSE attack are computationally infeasible for the foreseeable future.

Given the fact that all these protective features occur simultaneously in the constructed cipher, it is clear that CAST-designed encryption algorithms are immune to the OSE attack as described in [6].

4 Conclusion

The Overdefined System of Equations attack by Courtois and Pieprzyk is a relatively new type of attack against symmetric ciphers. It takes advantage of a previously-unexploited property of the s-boxes contained in certain classes of these ciphers: an overdefined and sparse system of equations may be derived from an s-box, resulting in expressions relating specific input and output bits that hold with equality, rather than with some small probability. Equality in these expressions, at least theoretically, means that the work factor for the attack grows linearly with the number of rounds (rather than exponentially, as is the case with many similar attack methods).

This paper has given a brief overview of the OSE attack and shown how it can be applied to a toy cipher. The paper then concentrated on design principles that can be employed to guarantee that the OSE attack will not be applicable to a cipher constructed according to those principles. It was shown that the CAST design procedure includes a number of principles in the round function and in the overall algorithm that provide immunity to this attack. For example, the large s-box dimension makes the OSE attack computationally infeasible, and the use of circular rotation makes certain that any advantage in some bit positions in one round will be nullified in subsequent rounds. Of particular significance, however, it was shown that the mixing of operations from different algebraic groups ensures that the attack complexity will no longer grow linearly with the number of rounds. This lends support to the conjecture that mixing operations is an important design criterion for strengthening ciphers against a variety of attacks.

From this work, we conclude that it is possible to design ciphers with immunity to the OSE attack. Furthermore, the CAST-128 cipher is a concrete example showing that this is not only possible, but that such ciphers can be readily implemented and efficient in practice.

References

1. Adams, C., "Constructing Symmetric Ciphers Using the CAST Design Procedure", *Designs, Codes and Cryptography*, vol. 12, no. 3, November, 1997, pp. 71-104.
2. Adams, C., "The CAST-128 Encryption Algorithm", *Internet Request for Comments RFC 2144*, May 1997.
3. Adams, C., and J. Gilchrist, "The CAST-256 Encryption Algorithm", *Internet Request for Comments RFC 2612*, June 1999.
4. Anderson, R., E. Biham, and L. Knudsen, "Serpent: A Proposal for the Advanced Encryption Standard". Available from `http://www.cl.cam.ac.uk/ rja14/serpent.html`
5. Coppersmith, D., "Impact of Courtois and Pieprzyk Results", *NIST AES Discussion Forum*, September 19, 2002. Available from `http://www.nist.gov/aes` (or see `http://www.makeashorterlink.com/?K27C515E1`)
6. Courtois, N., and J. Pieprzyk, "Cryptanalysis of Block Ciphers with Overdefined Systems of Equations", 2002. Available from `http://eprint.iacr.org/2002/044/` (See also *Proceedings of AsiaCrypt 2002*, LNCS 2501, Springer, pp. 267-287, and some further discussion at `http://www.cryptosystem.net/aes/`)
7. Daemen, J., and V. Rijmen, "AES proposal: Rijndael". Available from `http://csrc.nist.gov/CryptoToolkit/aes/rijndael/Rijndael.pdf`
8. Moh, T., "On the Courtois-Pieprzyk's Attack on Rijndael", September 18, 2002. Available from `http://www.usdsi.com/aes.html`
9. Murphy, S., and M. Robshaw, "Comments on the Security of the AES and the XSL Technique", September 26, 2002. Available from `http://www.cosic.esat.kuleuven.ac.be/nessie/reports/phase2/Xslbes8_Ness.pdf`
10. Row Reduced Echelon form for solving a system of linear equations. See, for example, *Module for Row Reduced Echelon Form*, available from `http://mathews.ecs.fullerton.edu/n2003/EchelonFormMod.html`
11. Shamir, A., J. Patarin, N. Courtois, and A. Klimov, "Efficient Algorithms for solving Overdefined Systems of Multivariate Polynomial Equations", *Advances in Cryptology: Proceedings of Eurocrypt 2000*, LNCS 1807, Springer, pp. 392-407.
12. Shannon, C., "Communication Theory of Secrecy Systems", *Bell System Technical Journal 28*, 1949, pp. 656-715.
13. Webster, A., and S. Tavares, "On the Design of S-Boxes", *Advances in Cryptology: Proceedings of CRYPTO '85*, Springer-Verlag, 1986, pp. 523-534.