An Analysis of XSL Applied to BES

Chu-Wee Lim, Khoongming Khoo

DSO National Laboratories, 20 Science Park Drive, S118230, Singapore.
lchuwee@dso.org.sg, kkhoongm@dso.org.sg

Abstract. Currently, the only plausible attack on the Advanced Encryption System (AES) is the XSL attack over F_{256} through the Big Encryption System (BES) embedding. In this paper, we give an analysis of the XSL attack when applied to BES and conclude that the complexity estimate is too optimistic. For example, the complexity of XSL on BES-128 should be at least 2^{401} instead of the value of 2^{87} from current literature. Our analysis applies to the eprint version of the XSL attack, which is different from the compact XSL attack studied by Cid and Leurent at Asiacrypt 2005. Moreover, we study the attack on the BES embedding of AES, while Cid and Leurent studies the attack on AES itself. Thus our analysis can be considered as a parallel work, which together with Cid and Leurent's study, disproves the effectiveness of both versions of the XSL attack against AES.

Keywords. XSL algorithm, AES, BES, linearisation.

1 Introduction

In 2001, after a standardisation period of 5 years, NIST finally adopted the Rijndael block cipher as the AES standard. A year later, Courtois and Pieprzyk surprised the cryptographic community by proposing an algebraic attack [3, 4] on Rijndael and Serpent, which could obtain the key faster than an exhaustive search. This attack was named XSL for eXtended Sparse Linearisation [3, 4], which is a modification of the earlier XL (eXtended Linearisation) attack proposed by Courtois et al [2]. In XSL, the authors exploited the algebraic simplicity of Rijndael's S-box and obtained a number of quadratic equations. Then in order to apply linearisation, equations are multiplied by S-box monomials (as compared to arbitrary monomials in the case of the XL attack). This enables the number of occurring monomials to be kept within a manageable range, and so upon linearisation, the system of linear equations can be solved faster than an exhaustive key search.

At around the same time, Murphy and Robshaw [6] proposed a method of reinterpreting the AES system by writing its equations over the field F_{256} instead of the smaller field F_2. Over F_{256}, the input x and output y of the Rijndael S-box satisfies the simple equation $xy = 1$. In this way, the number of equations of each S-box remains the same while the number of monomials involved is reduced by half. The authors later noted [7] that the BES embedding can lead to a dramatic decrease in the complexity of XSL attack. For example, as Courtois observed in §7 of [4], AES-128 can be broken with complexity 2^{87}.

However, Murphy and Robshaw also expressed skepticism at the practicality of XSL in [7], and concluded that it was unlikely to work. T. T. Moh and D. Coppersmith were similarly skeptical and remarked that XSL was unlikely to work.

In Asiacrypt 2005, Cid and Leurent [1] gave an analysis of the compact XSL attack [4] on AES-128 and proved that it is equivalent to a substitution-then-XL (sXL) method. They concluded that XSL attack is essentially an XL attack on a system of equations larger than that of the original AES. Thus compact XSL is not an effective attack against the AES cipher. This partly answers some uncertainties of the compact XSL attack and suggests that it may not be an effective method against block ciphers.

However, it does not give us the full answer on whether XSL is effective against AES. This is because:

1. In [1], only the compact XSL attack [4] is analysed. But the eprint version of the XSL attack (eprint XSL) [3] is different from the compact XSL attack as it uses a larger set of monomials. Moreover, the bounds derived from the eprint XSL are better than that of the compact XSL.
2. In [1], only the complexity of the compact XSL attack on AES over F_2 is analyzed. However the best attack on AES in the current literature is the eprint XSL attack on AES over the bigger field F_{256}.

In this paper, we provide a more complete answer to the effectiveness of the XSL attacks on AES by analyzing the above two points. We will focus on the embedding of the AES in the Big Encryption System (BES) over F_{256}. Because of the nice algebraic structure of the equations of the Rijndael S-box over F_{256}, we can provide an exact analysis of the linear dependencies that exist between the equations of BES. This allows us to give a more accurate estimate of the number of linearly independent equations in the eprint XSL attack. Based on our estimate, we deduce that the complexity of the eprint XSL attack on BES-128 is *at least* 2^{401} instead of 2^{87} from the known literature. Similar complexity estimates for BES-192 and BES-256 proves that the eprint XSL attack is ineffective against them.

2 The XSL Attack on BES

There are currently two versions of XSL. In this paper, we shall consider the version in [3]. Furthermore, in that paper, the authors described two forms of XSL - the first and the second. We shall look at the second version, which only requires at most two plaintext-ciphertext pairs. This section is essentially a summary of [3], [6] and [7], but we need it to establish notations which will be used in subsequent sections.

Note: we often mention linear equations from the AES cipher, when a technically more accurate name would be *affine*. This is in accordance with the literature.

2.1 A Summary of the XSL Attack

First, recall the S-box in the AES cipher. This comprises of the *inverse* map[1] on the finite field F_{256}, followed by an F_2-affine map on the output. Thus, the only nonlinear component of the cipher is the map $x \mapsto x^{-1}$ on the finite field F_{256}. The authors of [3] believe that this algebraic definition of the S-box presents an opportunity for algebraic attacks on the cipher. They noted that if $y = x^{-1}$ in F_{256}, then we get $xy = 1$, $x^2y = x$ and $xy^2 = y$. The Frobenius map $x \mapsto x^2$ is F_2-linear on F_{256}, so by taking the components, we get $3 \times 8 = 24$ equations in F_2. There is a slight caveat though: since $xy = 1$ is only true most of the time, it only produces 7 equations instead of 8. The eighth equation only holds $\frac{255}{256}$ of the time. Nevertheless, for the sake of our argument, we shall assume that we get 24 equations. It can be easily verified that up to linear dependence, these are the only equations we can get.

Thus, let $x_0, \ldots, x_7$ and $y_0, \ldots, y_7$ denote the input and output bits of an S-box respectively, We get 24 equations which are linear combinations of the monomials $1, x_i, y_j, x_i y_j$ $(0 \le i, j \le 7)$; i.e. , we have 24 equations involving 81 monomials. Such a monomial is said to *belong* to the S-box. Furthermore, if S and S' are distinct S-boxes, then the set of monomials belonging to S and S' are *disjoint*.

On the other hand, there are also F_2-linear equations for the `Shiftrows` and `MixColumns` algorithms. Hence, we can write down all the S-box and linear equations and solve them in order to extract the key. We shall use the following notation subsequently:

$$\begin{aligned}
S &= \text{number of S-boxes in the } \textit{cipher and key schedule}, \\
L &= \text{number of linear equations in the } \textit{cipher and key schedule}, \\
r &= \text{number of equations in each S-box} = 24, \\
t &= \text{number of terms used in the equations for each S-box} = 81, \\
P &= \text{XSL parameter (to be described later).}
\end{aligned}$$

The values for S and L for AES-128, -192 and -256 respectively are given in Table 1 below:

	AES-128	AES-192	AES-256
S	201	417	501
L	1664	3520	4128

Table 1. Parameters for Various AES Ciphers

Note that the values of S and L in Table 1 are based on one plaintext-ciphertext pair for the second XSL attack on AES-128 and two plaintext-ciphertext

[1] Strictly speaking, this is not true since the map $x \mapsto x^{-1}$ is not defined at 0. It would be correct mathematically to write this map as $x \mapsto x^{254}$.

pairs for the second XSL attack on AES-192, AES-256. They are derived from Section 7 and Appendix B of [3].

The XSL method can be summarised in the following steps, for a parameter P:

1. Pick P distinct S-boxes, and select one of them to be *active*; the others are declared *passive*. From the active S-box, select one equation out of the r quadratic ones; from each of the $P-1$ passive S-boxes, select a monomial t_i ($i = 2, \ldots, P$). We then multiply the active S-box equation with $t_2 t_3 \ldots t_P$ to obtain a new equation. Let us call such an equation an **extended S-box equation**. Take the collection Σ_S of all extended S-box equations. Also, we call the monomials which occur in the equations of Σ_S the **extended S-box monomials**.
2. Pick a linear equation and select $P-1$ distinct S-boxes. From each of the selected S-boxes, choose a monomial t_i ($i = 2, 3, \ldots, P$). We then multiply the linear equation by $t_2 t_3 \ldots t_P$ to obtain a new equation. Let us call such an equation an **extended linear equation**. Take the collection Σ_L of all extended linear equations.
3. Take the equations in $\Sigma := \Sigma_S \cup \Sigma_L$ and solve them via linearisation. In other words, replace each occurring monomial by a new variable and solve the set of linear equations by (say) Gaussian elimination. Furthermore, these linear equations are sparse so we can apply advanced techniques like the block Lanczos algorithm [5] to solve them.
4. If there are not enough equations for a complete solution, we can apply the T'-method to produce more equations. This comprises of (i) fixing a variable, say x_i, and (ii) attempting to find equations whose degrees remain the same upon multiplication by x_i.

In [3], it was noted that the equations in Σ_S have a lot of linear dependencies after linearisation. For example, if we pick two active S-boxes and equations eqn_1 and eqn_2 from them, as well as $P-2$ passive S-boxes with monomials $t_3, t_4, \ldots, t_P$, then by expanding the equation $eqn_1 \cdot eqn_2 \cdot (t_3 \ldots t_P)$ we obtain linear dependencies between extended S-box equations of the form $eqn_1 \cdot (t_2 t_3 \ldots t_P)$ and $t_1 \cdot eqn_2 \cdot (t_3 \ldots t_P)$. After removing some obvious dependencies like these, we get

$$R = \binom{S}{P}(t^P - (t-r)^P) \tag{1}$$

equations. Likewise, by fixing $t-r$ linearly independent terms in each S-box and expressing the remaining terms as linear combinations of them, the number of extended linear equations[2] is (after removing obvious linear dependencies):

$$R' = L \times (t-r)^{P-1} \binom{S}{P-1}. \tag{2}$$

[2] Note that in [3], the author used R' and R'' to denote the extended linear equations from the cipher and the key schedule respectively. Here we denote all extended linear equations by R'.

Hence, the total number of equations obtained is $R + R'$.

On the other hand, it was assumed that the monomials in Σ_L are also extended S-box monomials. Hence, the total number of terms occurring in Σ is the number of extended S-box monomials:

$$T = t^P \binom{S}{P}. \tag{3}$$

Finally, for the T'-method, the authors of [3] remarked that to apply T', we need at least $99.4\% \times T$ equations in the first place. In other words, T'-method can only be applied if the number of equations we have is very very close to being sufficient; in which case, the method helps to increase the number of equations slightly. Thus we shall leave it out in our discussion.

The objective of XSL is to select an appropriate P so that we get enough equations. The following computations apply for AES-128, AES-192 and AES-256.

1. **AES-128:** The smallest P where $R + R' > T$ is $P = 7$. The parameters are $R = 4.95 \times 10^{25}$, $R' = 4.85 \times 10^{24}$, $T = 5.41 \times 10^{25}$. We have $(R + R')/T = 1.004$ and the complexity of XSL attack is $T^{2.376} \approx 2^{203}$.

2. **AES-192:** The smallest P where $R + R' > T$ is $P = 7$. The parameters are $R = 8.65 \times 10^{27}$, $R' = 8.50 \times 10^{26}$, $T = 9.46 \times 10^{27}$. We have $(R + R')/T = 1.004$ and the complexity of XSL attack is $T^{2.376} \approx 2^{221}$.

3. **AES-256:** The smallest P where $R + R' > T$ is $P = 7$. The parameters are $R = 3.15 \times 10^{28}$, $R' = 3.02 \times 10^{27}$, $T = 3.45 \times 10^{28}$. We have $(R + R')/T = 1.002$ and the complexity of XSL attack is $T^{2.376} \approx 2^{225}$.

2.2 A Summary of the BES Cipher

At Crypto 2002, Robshaw and Murphy introduced the Big Encryption System (BES) embedding for the AES cipher [6]. In this embedding, the quadratic equations describing the S-box input-output become much simpler. This results in a substantial reduction in the number of monomials occurring in the system of equations, which can lead to an exponential reduction in the complexity of the XSL attack [7].

While AES performs its operations in the field F_2, BES achieves the same purpose by performing operations in the field F_{256}. The advantage of this rewriting lies in the simplicity of the S-box equation: we have $xy = 1$ immediately instead of 8 quadratic equations in the input and output bits. This, of course, conveniently ignores the case when $x = y = 0$. However this can be countered by other means.

The problem occurs with some of the linear equations, which are F_2-linear but not F_{256}-linear. This is overcome by introducing the conjugates of each variable $x \in F_{256}$, i.e. we have to consider $x_i := x^{2^i}$ for $0 \le i \le 7$. Upon introducing these variables, all F_2-linear equations can be expressed as F_{256}-linear equations, by virtue of the following result.

Lemma 1. *Consider the finite field $K = F_{2^n}$. Then any F_2-affine map $K \to K$ can be written in the form:*

$$f(x) = c + a_0 x + a_1 x^2 + \cdots + a_{n-1} x^{2^{n-1}},$$

for some constants $c, a_0, a_1, \ldots, a_{n-1} \in K$.

We will skip the proof, though it suffices to say that the result follows easily from dimension counting (over F_2).

In introducing the conjugates to the S-boxes, we have to express their relationship as $x_{i+1} = x_i^2$, where the subscript is taken from $\mathbb{Z}/8\mathbb{Z}$ (the integers modulo 8). This gives 24 equations for each S-box: indeed, if we denote the input and output variables by $x_0, x_1, \ldots, x_7 \in F_{256}$ and $y_0, y_1, \ldots, y_7 \in F_{256}$ respectively, then

$$x_0 y_0 = 1,\ x_1 y_1 = 1,\ x_2 y_2 = 1,\ \ldots\ , x_7 y_7 = 1,$$
$$x_0^2 = x_1,\ x_1^2 = x_2,\ x_2^2 = x_3,\ \ldots\ , x_7^2 = x_0,$$
$$y_0^2 = y_1,\ y_1^2 = y_2,\ y_2^2 = y_3,\ \ldots\ , y_7^2 = y_0.$$

For convenience, we make the following definition.

Definition 1. *Let x_i be an input variable of an S-box and y_i be the corresponding output variable such that $x_i y_i = 1$. We shall say x_i and y_i are* **dual** *to each other.*

Although the number of equations per S-box remains $r = 24$, we now only have 41 monomials! These are: $1, x_i, y_i, x_i^2, y_i^2, x_i y_i$ for $0 \le i \le 7$. Hence, we can apply the technique of XSL to this cipher, in which case formulae (1), (2) and (3) in the previous section still hold, with $t = 81$ replaced with $t = 41$. With this new value of t, it turns out that we can pick a smaller P and dramatically reduce the complexity:

1. **BES-128:** The smallest P where $R + R' > T$ is $P = 3$. The parameters are
$$R = 85341866400,\ R' = 9666009600,\ T = 91892369300.$$
So $(R + R')/T = 1.03$ and the complexity of XSL attack is $T^{2.376} \approx 2^{87}$.
2. **BES-192:** The smallest P where $R + R' > T$ is $P = 3$. The parameters are
$$R = 767998707840,\ R' = 88234798080,\ T = 826947240080.$$
So $(R + R')/T = 1.04$ and the complexity of XSL attack is $T^{2.376} \approx 2^{94}$.
3. **BES-256:** The smallest P where $R + R' > T$ is $P = 3$. The parameters are
$$R = 1333494666000,\ R' = 149422248000,\ T = 1435848423250.$$
So $(R + R')/T = 1.03$ and the complexity of XSL attack is $T^{2.376} \approx 2^{96}$.

Finally, the T'-method for BES was not mentioned in [7]. Although the premise of the method should remain the same - find equations whose degree remain the same upon multiplication by some variable - it's not entirely clear if this is effective. This is because the set of monomials involved in the S-box equations forms a very small subset of the set of monomials, thus multiplying an equation by a variable would almost certainly introduce new monomials even if the degree of the equation does not increase.

3 An Analysis of This Attack

In this section, we shall provide an in-depth analysis of the XSL attack when applied to the BES cipher [6]. Due to the nice structure of the BES S-box equations, we can obtain accurate numbers in many cases. Throughout this section, let us fix the XSL parameter P.

3.1 Analysing the Extended S-box Equations

First, let us consider the S-box equations, each of which is an equality of two monomials. Hence, each extended S-box equation is also of the form ($monomial_1$) = ($monomial_2$). Solving them linearly is rather easy: we get a collection of equivalence classes of monomials, where two monomials are considered equivalent if and only if we can obtain one from the other by a finite number of extended S-box equations.

Example 1. Consider three S-boxes, given by S_1, S_2 and S_3. The input and output pairs of these S-boxes are given by (a_i, b_i), (c_i, d_i) and (e_i, f_i) respectively, for $0 \le i \le 7$. To clarify the notation further, $a_i b_i = 1$ and $e_i^2 = e_{i+1}$ are examples of their equations. Then the monomials $a_3 b_3 c_2^2 e_5$ and $c_3 e_5$ are considered equivalent:

$$(a_3 b_3) c_2^2 e_5 = (1) c_2^2 e_5 = (1) c_3 e_5,$$

since the first equality follows from an extended S_1-equation, while the second follows from an extended S_2-equation.

Inspired by this example, we make the following definition.

Definition 2. *Let $\alpha = \alpha_1 \alpha_2 \dots \alpha_Q$ be an extended S-box monomial, where each α_i is a variable belonging to some S-box. Then α is said to be reduced if no two variables belong to the same S-box. The set of reduced S-box monomials of degree Q is denoted by Φ_Q.*

It follows that a reduced monomial of degree Q is a product of monomials from Q distinct S-boxes, and so $Q \le P$. Furthermore, we have the following theorem:

Theorem 1. *Every extended S-box monomial α is equivalent to a unique reduced monomial β. Furthermore, it is possible to obtain the equivalence via:*

$$\alpha = \gamma_1 = \gamma_2 = \gamma_3 = \dots = \gamma_r = \beta,$$

where each equality is an extended S-box equation, and the degree of each term is strictly less than the previous one.

Proof. Let $\alpha = \alpha_1 \alpha_2 \dots \alpha_Q$ be an extended S-box monomial, where each α_i is an S-box variable. If α were reduced, there is nothing to do. Otherwise, if α_i and α_j belong to the same S-box then they are either identical or *dual.*

In the first case, we have $\alpha_i = \alpha_j = x_k$ for some input/output variable x_k; hence upon removing α_i and α_j and adding an x_{k+1}, we get an equivalent monomial of degree $Q-1$. In the second case, we have $\alpha_i\alpha_j = 1$ and we get an equivalent monomial of degree $Q-2$. This gives us an extended S-box equation

$$\alpha = \gamma_1,$$

where $\deg\gamma_1 < \deg\alpha$. Repeating the above process with α replaced by γ_1, we get the desired result.

Finally, we need to prove uniqueness, i.e. two distinct reduced monomials are not equivalent. This is quite easy: write $\alpha_1\alpha_2\ldots\alpha_Q = \beta_1\beta_2\ldots\beta_{Q'}$, where the α_i and β_i are S-box variables. Now, the α_i all belong to distinct S-boxes, as do the β_i. For equality to hold, some α_i and β_j must belong to the same S-box. It immediately follows that $\alpha_i = \beta_j$, so we may cancel them from the equation and repeat. □

Note that each Φ_i has cardinality $\binom{S}{i}16^i$. Thus, upon solving the extended S-box equations, the number of linearly independent terms is exactly

$$D_0 = \sum_{i=0}^{P} |\Phi_i| = \sum_{i=0}^{P} \binom{S}{i} 16^i.$$

According to theoretical bound in [3], that should have been:

$$T - R = \binom{S}{P}(t-r)^P = \binom{S}{P}17^P.$$

Hence, we see that the estimate in [3] was rather close.

3.2 Adding the Extended Linear Equations

The second step is to multiply each linear equation by an extended S-box monomial. However, by Theorem 1 it is equivalent to multiply each linear equation by a reduced S-box monomial of degree at most $P-1$ (note: we cannot multiply by a reduced monomial of degree P since we only multiply $P-1$ monomials from passive S-boxes). In a nutshell, the XSL method is equivalent to the following:

1. Obtain the set Σ_S of extended S-box equations.
2. For each linear equation, we multiply it by a reduced monomial from $\Phi_0 \cup \Phi_1 \cup \cdots \cup \Phi_{P-1}$ and obtain the set Σ'_L of extended linear equations.
3. Solve $\Sigma_S \cup \Sigma'_L$ together via linearisation.

Consider hypothetically the case where we just do steps 2 and 3 (i.e. step 3 is performed with $\Sigma_S = \emptyset$). In short, we took a bunch of linear equations, multiply them by some monomials and attempt to solve them by linearisation. The question is: how many linearly independent terms will we get from this attempt?

It is not difficult to give a lower bound for this number. First, consider the set of non-reduced S-box equations and linear equations. If we fix the 8 input variables of an S-box, then its output is known. Thus the removal of each S-box contribute 8 free variables. Then the removal of the S S-boxes results in the introduction of $8S$ totally free variables, as the diagram in figure 1 shows.

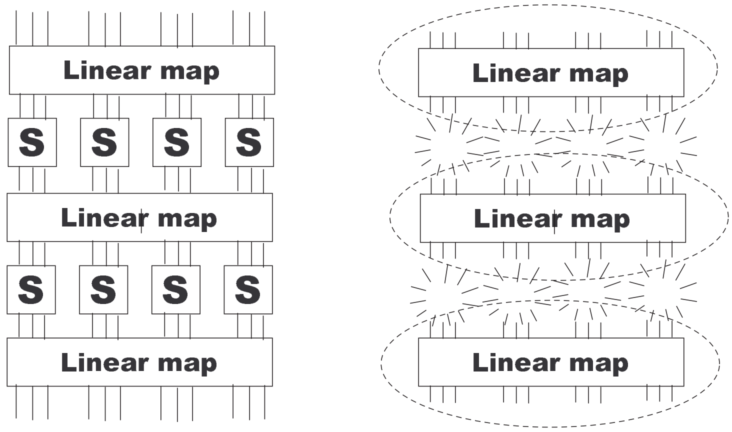

Fig. 1. Removal of the S-boxes gives $8S$ free variables

Without loss of generality, we take the $8S$ input variables of the S-boxes to be the free variables. In other words, the equations in Σ'_L can be satisfied regardless of the values these $8S$ variables take. Hence, the number of linearly independent terms is at least the number of reduced monomials formed by these $8S$ free variables:

$$D_1 = \sum_{i=0}^{P} \binom{S}{i} 8^i.$$

Next, we ask ourselves: *does step 1 provide sufficiently many equations to remove this number of linearly independent terms?*

By theorem 1, we can replace each monomial in Σ'_L by a corresponding reduced one. Furthermore, after such replacements, the extended S-box equations are no longer of any use. This brings us to the next theorem.

Theorem 2. *When solving Σ'_L with the extended S-box equations, we only need to include those extended S-box equations of the form:*

$$(v)(m_1) = (m_2),$$

where:

1. $m_1, m_2 \in \Phi_0 \cup \Phi_1 \cup \cdots \cup \Phi_{P-1}$, *i.e.* m_1 *and* m_2 *are reduced monomials of degree at most* $P - 1$*;*

2. v is an S-box variable such that it or its dual occurs in m_1;
3. the remaining variables in m_1 are among the $8S$ free (input) variables.

Proof. Recall that each equation of Σ'_L is of the form: $l \cdot m = 0$, where l is a linear equation and m is a *reduced* monomial of degree at most $P-1$. Hence, after linearisation, the only occurring monomials in Σ'_L are those of the form $v \cdot m$, where v is an S-box variable and m is a reduced monomial of degree $\leq P-1$. If neither v nor its dual occur in m, then this monomial itself is already reduced. Otherwise, we can use an extended S-box equation satisfying conditions 1 and 2 to reduce the monomial further. By Theorem 1, these are the only extended S-box equations we need.

Finally, note that any variable which is not one of the $8S$ free variables, can be expressed as a linear combination of these $8S$ variables. Hence, if $(v)(m_1) = (m_2)$ is an equation satisfying conditions 1 and 2, we can replace each of the remaining variables in m_1 with a linear combination of the $8S$ variables and expand the resulting monomial. Thus, the set of extended S-box equations satisfying conditions 1-3 suffices. □

We say that an extended S-box equations is "**relevant**" if it satisfies conditions 1-3 in theorem 2.

Now let us compute the number of such equations. Take $(v)(m_1) = (m_2)$, and v' occurs in m_1 where $v' = v$ or the dual of v.

For the case $v = v'$, there are $16S$ choices for the pair $(v, v') = (v, v)$, where v can be one of 16 input or output variables of an S-box. For the case where v' is the dual of v, there are $8S$ choices for the unordered pair (v, v'), which is one of 8 dual pairs of an S-box satisfying $vv' = 1$. Thus there are $16S + 8S = 24S$ choices for the unordered pair (v, v') in the equation $(v)(m_1) = (m_2)$.

For the remaining $P-2$ variables in m_1, there are $\sum_{i=0}^{P-2} \binom{S-1}{i} 8^i$ choices. Thus, the number of "relevant" equations is

$$D_2 = 24S \times \sum_{i=0}^{P-2} \binom{S-1}{i} 8^i.$$

For the equations in $\Sigma_S \cup \Sigma'_L$ to be solved via linearisation, we must have $D_2 \geq D_1$. However, the values in Table 2 indicate that the stipulated values of P which were supposed to work (see §2.2), actually don't.

To be able to solve for the secret key, we need $D_2 > D_1$. From Table 2, the smallest P where this condition is satisfied is when $P = 23, 33, 36$ for BES-128, BES-192 and BES-256 respectively. In that case, we have the following complexity for the XSL attack:

1. **BES-128:** $S = 201$, $P = 23$.

$$\text{Complexity } = D_1^{2.376} = (5.9 \times 10^{50})^{2.376} \approx 2^{401}.$$

2. **BES-192:** $S = 417$, $P = 33$.

$$\text{Complexity } = D_1^{2.376} = (5.857 \times 10^{78})^{2.376} \approx 2^{622}.$$

	BES-128	BES-192	BES-256
$P=2$	$D_1 = 1.288 \times 10^6$ $D_2 = 4.824 \times 10^3$	$D_1 = 5.554 \times 10^6$ $D_2 = 1.001 \times 10^4$	$D_1 = 8.02 \times 10^6$ $D_2 = 1.202 \times 10^4$
$P=3$	$D_1 = 6.839 \times 10^8$ $D_2 = 7.723 \times 10^6$	$D_1 = 6.149 \times 10^9$ $D_2 = 3.332 \times 10^7$	$D_1 = 5.093 \times 10^{10}$ $D_2 = 4.811 \times 10^7$
$P=4$	$D_1 = 2.71 \times 10^{11}$ $D_2 = 6.152 \times 10^9$	$D_1 = 5.093 \times 10^{12}$ $D_2 = 5.532 \times 10^{10}$	$D_1 = 1.063 \times 10^{13}$ $D_2 = 9.605 \times 10^{10}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$P=23$	$D_1 = 5.9 \times 10^{50}$ $D_2 = 6.245 \times 10^{50}$	$\vdots$	$\vdots$
$P=33$	$\vdots$	$D_1 = 5.857 \times 10^{78}$ $D_2 = 6.02 \times 10^{78}$	$\vdots$
$P=36$	$\vdots$	$\vdots$	$D_1 = 3.798 \times 10^{87}$ $D_2 = 3.849 \times 10^{87}$

Table 2. Number of linearly independent monomials D_1 from the extended linear equations and number of relevant extended S-box equations D_2 for various P

3. **BES-256:** $S = 501$, $P = 36$.

$$\text{Complexity } = D_1^{2.376} = (3.798 \times 10^{87})^{2.376} \approx 2^{691}.$$

In comparison, the XL attack against the AES-128 cipher has complexity 2^{330} [3, Section 5.2]. *Thus we see that the XSL attack is not effective against the BES cipher. In fact, it gives worse complexity than the XL attack against AES-128.*

3.3 Further Analysis

The above results show that there are many hidden linear dependencies which were not accounted for in the computations in §2. Here, we attempt to account for some of this discrepancy.

(a) *It is not true that all the monomials which occur are extended S-box monomials.* The problem occurs with the extended linear equations. It can happen that l is a linear equation which involves (say) x_2 from a certain S-box, while the chosen passive S-boxes also include the term (say) y_5 from the same S-box; in which case, the term x_2y_5 is part of an occurring monomial. This monomial is then not an extended S-box monomial. Heuristically, the number of monomials unaccounted for is not very significant.

(b) *The second and worse problem is the presence of inherent linear dependencies among the extended linear equations.* This is essentially identical to the linear dependencies among the extended S-box equations mentioned in §2.1. For example, suppose $linear_1 = 0$ and $linear_2 = 0$ are two different linear equations. By taking terms from $P-2$ distinct passive S-boxes, we get the monomial $t_3 \ldots t_P$; expanding the equation $(linear_1)(linear_2)(t_3 \ldots t_P) = 0$ results in a linear relation between the equations extended from $linear_1 = 0$

and those extended from $linear_2 = 0$. Notice that this linear dependency is inherent among the extended linear equations. It is completely unrelated to the S-box equations or their extended counterparts. Unfortunately, this has been neglected in the original estimates.

While (a) appears to be a minor oversight, (b) has a much larger effect on the estimates. As mentioned in [3], the removal of obvious linear dependencies on the S-box equations causes the number of such equations to reduce from $R_{old} = rSt^{P-1}\binom{S-1}{P-1}$ to $R = \binom{S}{P}(t^P - (t-r)^P)$, which is quite a significant difference. It is likely that similar considerations on the extended linear equations would also result in a significant reduction of useful equations.

4 Conclusion

The purpose of our paper is to analyse XSL when applied to BES, and determine if it would work in practice. Due to the nice S-box equations of BES, we can explicitly deduce the linear dependencies between the equations. *Our conclusion is that if XSL works on BES, then it is worse than brute foce.* However, it leaves open the question of whether XSL works for some P at all.

Furthermore, our computations do not carry over to the original Rijndael (with equations over F_2 instead of F_{256}) or to the Serpent cipher. Due to the complexity of the S-box equations, an explicit list of linearly independent terms of the extended S-box equations cannot be easily described. Naturally, we ask if XSL works in those cases.

First, it should be noted that the linear dependencies among the extended linear equations - as mentioned in §3.3 - hold in general, so it is likely that the number of linearly independent equations obtained after linearisation is actually smaller than expected. Second, in the XSL method, the final ratio (number of equations)/(number of terms) after linearisation is about 1.004, which is precariously close to 1. Hence, the presence of a large number of linear dependencies can have an adverse effect on the solvability of the system. Finally, note that we had left the T'-method out of the discussion totally. However we had noted that the T'-method is only effective if the number of linearly independent equations is already extremely close to being sufficient (about 99.4% of what is needed). Thus, even the application of the T'-method is unlikely to help.

References

1. C. Cid and G. Leurent, "An Analysis of the XSL Algorithm", LNCS 3788, *Asiacrypt 2005*, pp. 333-352, Springer-Verlag, 2005.
2. N. Courtois, A. Klimov, J. Patarin and A. Shamir, "Efficient Algorithms for Solving Systems of Multivariate Polynomial Equations", LNCS 1807, *Eurocrypt 2000*, pp. 392-407, Springer-Verlag, 2000.
3. N. Courtois and J. Pieprzyk, "Cryptanalysis of Block Ciphers with Overdefined Systems of Equations", IACR eprint server, `http://www.iacr.org`, 2002/044, March 2002.

4. N. Courtois and J. Pieprzyk, “Cryptanalysis of Block Ciphers with Overdefined Systems of Equations”, LNCS 2501, *Asiacrypt 2002*, pp. 267-287, Springer-Verlag, 2002.
5. P. L. Montgomery, “A Block Lanczos Algorithm for Finding Linear Dependencies over GF(2)”, LNCS 921, *Eurocrypt'95*, pp. 106-120, Springer-Verlag, 1995.
6. S. Murphy and M. Robshaw, “Essential Algebraic Structure Within the AES”, LNCS 2442, *Crypto 2002*, pp. 1-16, Springer-Verlag, 2002.
7. S. Murphy and M. Robshaw, “Comments on the Security of the AES and the XSL Technique”, *Electronic Letters*, 39:26-38, 2003.