ALGEBRAIC CRYPTANALYSIS OF SIMPLIFIED AES

SEAN SIMMONS

Abstract. Simplified AES was developed in 2003 as a teaching tool to help students understand AES. It was designed so that the two primary attacks on symmetric-key block ciphers of that time, differential cryptanalysis and linear cryptanalysis, are not trivial on simplified AES. Algebraic cryptanalysis is a technique that uses modern equation solvers to attack cryptographic algorithms. There have been some claims that AES is threatened by algebraic cryptanalysis. We will use algebraic cryptanalysis to attack simplified AES.

In his 1949 paper, "Communication Theory of Secrecy Systems [8]," Claude Shannon asked the question "How can we ever be sure that a [crypto]system which is not ideal ... will require a large amount of work to break with *every* method of analysis?" [[8], 704] Shannon suggested two approaches to that problem:

> (1) We can study the possible methods of solution available to the cryptanalyst and attempt to describe them in sufficiently general terms to cover any of the methods he might use. We then construct our system to resist this "general" method of solution. (2) We may construct our cipher in such a way that breaking it is equivalent to (or requires at some point in the process) the solution of some problem known to be laborious. Thus, if we could show that solving a certain system requires at least as much work as solving a system of simultaneous equations in a large number of unknowns, of a complex type, then we would have a lower bound of sorts for the work characteristic. [[8], 704]

It is the second approach – in reverse – that is the basis of algebraic cryptanalysis. To do algebraic cryptanalysis, the cryptanalyst models the cryptosystem as a system of polynomial equations and then attempts to solve that system. The cryptanalyst's success is determined by whether or not it is possible to solve the resulting system. The technique of linear cryptanalysis, which has been known since the mid-1990s, attempts to find "approximately" linear relationships and solve the resulting system of *linear* equations, which is easy to do. Algebraic cryptanalysis determines exact (probably nonlinear) polynomial models of the cryptosystem, but then it depends on using powerful software and "tricks of the trade" to solve the

Key words and phrases. Algebraic cryptanalysis, AES, F4, symmetric-key block ciphers.

This work is a result of the author's participation in the 2008 National Science Foundation-sponsored Research Experience for Undergraduates (REU) in Mathematical Cryptology jointly offered by Northern Kentucky University and the University of Cincinnati. Thanks to the United States Air Force Office of Scientific Research for funding the REU and to the Computational Algebra Group within the School of Mathematics and Statistics of the University of Sydney for supplying Magma for the REU and for this project.

systems. In 2006, Nicolas Courtois attracted attention to algebraic cryptanalysis by claiming to have a "fast algebraic attack against block ciphers" that would threaten the security of AES. An excellent introduction to algebraic cryptanalysis, some discussion of Courtois' claims, and an attack on the Courtois Toy Cipher may be found in [1].

In November 2001 the National Institute of Standards and Technology (NIST) selected Rijndael, a cryptographic algorithm submitted by Joan Daemen and Vincent Rijmen of Belgium, to be the Advanced Encryption Standard (AES) to replace the well-used but aged Data Encryption Standard (DES), which had served since 1977. AES is a 128-bit symmetric-key block cipher. AES is designed to be more efficient than 3DES, have a larger block size than 64-bit DES, and be resistant to two common attacks on block ciphers, differential and linear cryptanalysis, which have been known since the late 1980s (See [2], [3], and [4].) and 1994 [6], respectively. For a discussion of the AES algorithm see [9].

In 2003, Musa, Schaefer, and Wedig published a simplified version of AES [7] for instructional purposes.

> Though AES is not inordinately complicated, it would be best understood if one could work through an example by hand. However, this is not feasible. So we have designed a simplified version of AES for which it is possible to work through an example by hand. In addition, we believe that we have shrunk the parameters as much as possible without losing the essence of the algorithm. The parameters were chosen so that linear and differential cryptanalyses are not trivial. [[7], 149]

Simplified AES is a two-round algorithm. (AES is a ten-round algorithm for 128-bit keys, a twelve-round algorithm for 192-bit keys, and a fourteen-round algorithm for 256-bit keys.) Musa, Schaefer, and Wedig do differential cryptanalysis of one-round and two-round simplified AES and linear cryptanalysis of one-round simplified AES. We will refer to simplified AES as S-AES.

In what follows, we will describe a simple algebraic cryptanalysis of S-AES.

1. Description of Simplified AES

The following description of S-AES is an abbreviation of the description of S-AES given in [7].

S-AES uses a 16-bit key to encrypt 16-bit blocks. Because there are only 2^{16} keys for S-AES, S-AES can be successfully attacked by brute force. A keyspace of size 2^{80} is commonly thought to be necessary to protect against brute force attacks today.

1.1. **The S-box.**

The substitution box, the S-box, is the heart of S-AES. Because solving systems of linear equations is easy, cryptosystems need to have non-linear components. The S-box is a nonlinear component of S-AES.

The S-box takes a 4-bit input and produces a 4-bit output. A block of four bits is often called a nibble (half of a byte).

The simplest way to describe the S-box of S-AES is as a lookup table:

Input	*Output*
0000	1001
0001	0100
0010	1010
0011	1011
0100	1101
0101	0001
0110	1000
0111	0101
1000	0110
1001	0010
1010	0000
1011	0011
1100	1100
1101	1110
1110	1111
1111	0111

The S-Box can be described in another manner. Define $GF(16) = GF(2)[x]/(x^4+x+1)$ (Because x^4+x+1 is irreducible over $GF(2)[x]$, $GF(2)[x]/(x^4+x+1)$ is a field of $2^4 = 16$ elements.) Let the nibble $b_0b_1b_2b_3$ be the input of the S-box where b_0, b_1, b_2, and b_3 are the four bits. Consider $p = b_0x^3 + b_1x^2 + b_2x + b_3 \in GF(16) \cong GF(2)[x]/(x^4+x+1)$, where the bits $b_0, ..., b_3$ are considered as elements of $GF(2)$. If $p \neq 0$, let $q = p^{-1}$ where q is the inverse of p considered as an element of $GF(2)[x]/(x^4+x+1)$. If $p = 0$, let $q = 0$. Write $N(y) = a_0y^3 + a_1y^2 + a_2y + a_3$ in $GF(2)[y]/(y^4+1)$ where $q = a_0x^3+a_1x^2+a_2x+a_3 \in GF(16) \cong GF(2)[x]/(x^4+x+1)$. Let $a(y) = y^3 + y^2 + 1$ and $b(y) = y^3 + 1$. The coefficients of the resulting polynomial $a(y)N(y) + b(y)$ form the 4-bit output of the S-box. Notice that $GF(2)[y]/(y^4+1)$ is not a field; however, $a(y) = y^3 + y^2 + 1$ does have an inverse in $GF(2)[y]/(y^4+1)$.

The inversion of p is what makes this process nonlinear. For $p \neq 0$, $q = p^{-1}$; so $qp = 1$, which is a quadratic relation.

1.2. **Key scheduling.** Let $k_1, ..., k_{16}$ be the 16 given key bits. The two rounds of S-AES require 48 key bits. The 48 key bits are constructed from the 16-bit key in a very complicated way. For $i = 1, 2$, let $RC[i] = x^{2+i} \in GF(16)$; i.e., a nibble. $RCON[i] = RC[i]0000$ represents the concatenation of $RC[i]$ with 0000. (RC and $RCON$ are abbreviations for *round constant*.) Then given a byte N_0N_1, where N_0 is the first nibble of the byte, and N_1 the second nibble of the byte; define (*rotate nibble*) $RotNib(N_0N_1) = N_1N_0$ and (*substitute nibble*) $SubNib(N_0N_1) = SBox(N_0)SBox(N_1)$ (where $SBox(N_i)$ is the output obtained when the nibble N_i is acted upon by the S-box). Then, letting $\oplus$ represent the XOR operation:

$W[0] = k_1...k_8$
$W[1] = k_9...k_{16}$
If $i = 0(mod 2)$, $i > 0$ $W[i] = W[i-2] \oplus RCON(i/2) \oplus SubNib(RotNib(W[i-1]))$
Else $W[i] = W[i-2] \oplus W[i-1]$

$W[0]W[1] = K_0$ is the first string of 16 key bits (which is used prior to round 1), $W[2]W[3] = K_1$ is the second string of 16 key bits (which is used at the end of

round 1), and $W[4]W[5] = K_2$ is the third string of 16 key bits (which is used at the end of round 2).

1.3. **The S-AES algorithm.** Let $M = m_1m_2...m_{16}$ be a 16-bit block of plaintext message. K_0, K_1, K_2 represents the 48 key bits that were determined from the 16 bits of key as described above. The S-AES algorithm has three more components:

Nibble substitution (which acts on 16-bit blocks) NS: Given a 16-bit input $N_0N_1N_2N_3$, where N_0 is the first nibble, N_1 the second nibble, etc., then $NS(N_0N_1N_2N_3) = SBox(N_0)SBox(N_1)SBox(N_2)SBox(N_3)$.

Shift row (which acts on 16-bit blocks) SR: Given $N_0N_1N_2N_3$, $SR(N_0N_1N_2N_3) = N_0N_3N_2N_1$.

Mix column (which acts on 16-bit blocks) MC: There are several ways to look at MC. (See [7] [1].) For our purposes, probably the best way to view MC is in the following way:

Given two nibbles, $b_0b_1b_2b_3|b_4b_5b_6b_7$ (where each b_i is a bit), define

$$f(b_0b_1b_2b_3|b_4b_5b_6b_7) =$$
$$(b_0 \oplus b_6)|(b_1 \oplus b_4 \oplus b_7)|(b_2 \oplus b_4 \oplus b_5)|(b_3 \oplus b_5)|(b_2 \oplus b_4)|(b_0 \oplus b_3 + \oplus b_5)|(b_0 \oplus b_1 \oplus b_6)|(b_1 \oplus b_7)$$

where $\oplus$ is XOR, and $|$ is the symbol for concatenation.

The S-AES algorithm is:

$$C = SR(NS(MC(SR(NS(M \oplus K_0))) \oplus K_1)) \oplus K_2,$$

where M is a 16-bit block of plaintext and C is a 16-bit block of ciphertext.

2. Algebraic Cryptanalysis of S-AES

Our algebraic cryptanalysis of S-AES is very simple to construct. First, we will construct polynomials over $GF(2)$ that represent the cipher. Variables in our polynomials will be plaintext bits and ciphertext bits and unknowns will be keybits. Our attack is a known plaintext attack; so, next, we substitute known plaintext-ciphertext pairs into the polynomials that we have constructed. Finally, we will solve the system of polynomials for the key bits using the computer algebra software Magma [5].

2.1. **Generating the S-box polynomials.** The first step in our algebraic attack on S-AES is to model the S-box as a system of polynomials. We have chosen a very straightforward procedure: we have constructed an ordered list of four polynomials each of which accepts 4-bit input and returns one bit of output. The four ordered bits of output are the four bits of the the S-box output.

To model the S-box, we used the lookup table representation of the S-box. For each of the output bits, we generated the corresponding polynomial in the following way. We first listed all possible monomials in terms of the input bits x_1, x_2, x_3, and x_4. To calculate the coefficients of the monomials, we substituted the known plaintext input/output pairs into the polynomials, and created a system of linear equations in terms of the coefficients of the polynomial. Solving this system gave us the coefficients, and thus the S-box polynomials. Here are the four S-box polynomials, where $sbox_i(x_1, ..., x_4)$ is the i-th output bit of the S-box.

[1] In [7] the authors also explain the reasons for the terms *shift row* and *mix column*.

$sbox_1(x_1,..,x_4) = x_1x_2 \oplus x_1x_3x_4 \oplus x_1x_4 \oplus x_1 \oplus x_2x_3x_4 \oplus x_3x_4 \oplus x_4 \oplus 1,$

$sbox_2(x_1,..,x_4) = x_1x_2x_4 \oplus x_1x_2 \oplus x_1x_3 \oplus x_1 \oplus x_2x_3x_4 \oplus x_2x_3 \oplus x_2 \oplus x_3x_4 \oplus x_4,$

$sbox_3(x_1,..,x_4) = x_1x_2x_3 \oplus x_1x_2x_4 \oplus x_1x_2 \oplus x_1x_3x_4 \oplus x_1 \oplus x_2x_3 \oplus x_3,$

$sbox_4(x_1,..,x_4) = x_1x_2x_3 \oplus x_1x_2x_4 \oplus x_1x_3x_4 \oplus x_1x_3 \oplus x_1x_4 \oplus x_1 \oplus x_2x_3x_4 \oplus x_2x_4 \oplus x_3 \oplus x_4 \oplus 1$

2.2. **Modeling the key schedule.** Let $q_1, ..., q_{16}$ be unknowns that correspond to the bits of the 16-bit key. Then let $M = m_1...m_{16}$ represent the 16 bits of a known plaintext message and $C = c_1...c_{16}$ represent the corresponding 16 bits of ciphertext. The first step involves writing $W[0], ..., W[5]$ as ordered lists of polynomials over $GF(2)$ in terms of the unknowns $q_1, ..., q_{16}$. ($W[0], ..., W[5]$ are defined in section 1.2.) To do this, we first define $W[0] = q_1...q_8$ and $W[1] = q_9...q_{16}$. The next step is to calculate $W[2]$, which results from XORing $W[0]$ and $RCON[1]$. We then take $W[1] = q_9...q_{16}$ and apply RotNib (which is defined in section 1.2) to get $RotNib(W[1]) = q_{13}q_{14}q_{15}q_{16}q_9q_{10}q_{11}q_{12}$. We then apply the SubNib operation.

$$SubNib(RotNib(W[1])) = SBox(q_{13}q_{14}q_{15}q_{16})SBox(q_9q_{10}q_{11}q_{12}) = p_1p_2...p_8,$$

where $p_1, ..., p_4$ are polynomials in terms of the unknowns $q_{13}, q_{14}, q_{15}, q_{16}$, and $p_5, ..., p_8$ are polynomials in terms of the unknowns $q_9, q_{10}, q_{11}, q_{12}$. Finally, we XOR $p_1...p_8$ with $W[0]$ and $RCON[1]$ to get $W[1]$ as set of polynomials in terms of the unknown key variables – the qs. Similarly we can determine polynomials from $W[3]$,$W[4]$, and $W[5]$.

The polynomials for the $W[4]$ and $W[5]$ are too large to display, but the polynomials for $W[0]$,$W[1]$,$W[2]$, and $W[3]$ are:

$W[0] = (q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8)$

$W[1] = (q_9, q_{10}, q_{11}, q_{12}, q_{13}, q_{14}, q_{15}, q_{16})$

$W[2]$=($q_1 \oplus q_9 \oplus q_{13}q_{14} \oplus q_{13}q_{15}q_{16} \oplus q_{13}q_{16} \oplus q_{13} \oplus q_{14}q_{15}q_{16} \oplus q_{15}q_{16} \oplus q_{16}$, $q_2 \oplus q_{10} \oplus q_{13}q_{14}q_{16} \oplus q_{13}q_{14} \oplus q_{13}q_{15} \oplus q_{13} \oplus q_{14}q_{15}q_{16} \oplus q_{14}q_{15} \oplus q_{14} \oplus q_{15}q_{16} \oplus q_{16}$, $q_3 \oplus q_{11} \oplus q_{13}q_{14}q_{15} \oplus q_{13}q_{14}q_{16} \oplus q_{13}q_{14} \oplus q_{13}q_{15}q_{16} \oplus q_{13} \oplus q_{14}q_{15} \oplus q_{15}$,$q_4 \oplus q_{12} \oplus q_{13}q_{14}q_{15} \oplus q_{13}q_{14}q_{16} \oplus q_{13}q_{15}q_{16} \oplus q_{13}q_{15} \oplus q_{13}q_{16} \oplus q_{13} \oplus q_{14}q_{15}q_{16} \oplus q_{14}q_{16} \oplus q_{15} \oplus q_{16} \oplus 1$,$q_5 \oplus q_9q_{10} \oplus q_9q_{11}q_{12} \oplus q_9q_{12} \oplus q_9 \oplus q_{10}q_{11}q_{12} \oplus q_{11}q_{12} \oplus q_{12} \oplus q_{13} \oplus 1$, $q_6 \oplus q_9q_{10}q_{12} \oplus q_9q_{10} \oplus q_9q_{11} \oplus q_9 \oplus q_{10}q_{11}q_{12} \oplus q_{10}q_{11} \oplus q_{10} \oplus q_{11}q_{12} \oplus q_{12} \oplus q_{14}$, $q_7 \oplus q_9q_{10}q_{11} \oplus q_9q_{10}q_{12} \oplus q_9q_{10} \oplus q_9q_{11}q_{12} \oplus q_9 \oplus q_{10}q_{11} \oplus q_{11} \oplus q_{15}$, $q_8 \oplus q_9q_{10}q_{11} \oplus q_9q_{10}q_{12} \oplus q_9q_{11}q_{12} \oplus q_9q_{11} \oplus q_9q_{12} \oplus q_9 \oplus q_{10}q_{11}q_{12} \oplus q_{10}q_{12} \oplus q_{11} \oplus q_{12} \oplus q_{16} \oplus 1$)

$W[3]$= ($q_1 \oplus q_9 \oplus q_{13}q_{14} \oplus q_{13}q_{15}q_{16} \oplus q_{13}q_{16} \oplus q_{13} \oplus q_{14}q_{15}q_{16} \oplus q_{15}q_{16} \oplus q_{16}$, $q_2 \oplus q_{10} \oplus q_{13}q_{14}q_{16} \oplus q_{13}q_{14} \oplus q_{13}q_{15} \oplus q_{13} \oplus q_{14}q_{15}q_{16} \oplus q_{14}q_{15} \oplus q_{14} \oplus q_{15}q_{16} \oplus q_{16}$, $q_3 \oplus q_{11} \oplus q_{13}q_{14}q_{15} \oplus q_{13}q_{14}q_{16} \oplus q_{13}q_{14} \oplus q_{13}q_{15}q_{16} \oplus q_{13} \oplus q_{14}q_{15} \oplus q_{15}$, $q_4 \oplus q_{12} \oplus q_{13}q_{14}q_{15} \oplus q_{13}q_{14}q_{16} \oplus q_{13}q_{15}q_{16} \oplus q_{13}q_{15} \oplus q_{13}q_{16} \oplus q_{13} \oplus q_{14}q_{15}q_{16} \oplus q_{14}q_{16} \oplus q_{15} \oplus q_{16} \oplus 1$, $q_5 \oplus q_9q_{10} \oplus q_9q_{11}q_{12} \oplus q_9q_{12} \oplus q_9 \oplus q_{10}q_{11}q_{12} \oplus q_{11}q_{12} \oplus q_{12} \oplus q_{13} \oplus 1$, $q_6 \oplus q_9q_{10}q_{12} \oplus q_9q_{10} \oplus q_9q_{11} \oplus q_9 \oplus q_{10}q_{11}q_{12} \oplus q_{10}q_{11} \oplus q_{10} \oplus q_{11}q_{12} \oplus q_{12} \oplus q_{14}$, $q_7 \oplus q_9q_{10}q_{11} \oplus q_9q_{10}q_{12} \oplus q_9q_{10} \oplus q_9q_{11}q_{12} \oplus q_9 \oplus q_{10}q_{11} \oplus q_{11} \oplus q_{15}$, $q_8 \oplus q_9q_{10}q_{11} \oplus q_9q_{10}q_{12} \oplus q_9q_{11}q_{12} \oplus q_9q_{11} \oplus q_9q_{12} \oplus q_9 \oplus q_{10}q_{11}q_{12} \oplus q_{10}q_{12} \oplus q_{11} \oplus q_{12} \oplus q_{16} \oplus 1$)

2.3. **Generating the system of polynomials.** The next step is to construct the system of equations that models the rest of S-AES encryption. We will be using as an example throughout this section the case we have a plaintext message of all 1's. We start with the ordered list of polynomials corresponding to K_0. Then add K_0 to the ordered list $(m_1, ..., m_{16})$ coordinate wise (where $m_1, ..., m_{16}$ are the bits of the known plaintext); this results in an ordered list of 16 polynomials, $f_1, ..., f_{16}$. For example, if our plaintext message were the "all 1's" message, the polynomials would be:

$$f_1 = q_1 \oplus 1,\ f_2 = q_2 \oplus 1,\ f_3 = q_3 \oplus 1,\ f_4 = q_4 \oplus 1,\ f_5 = q_5 \oplus 1,\ f_6 = q_6 \oplus 1,$$
$$f_7 = q_7 \oplus 1,\ f_8 = q_8 \oplus 1,\ f_9 = q_9 \oplus 1,\ f_{10} = q_{10} \oplus 1,\ f_{11} = q_{11} \oplus 1,\ f_{12} = q_{12} \oplus 1,$$
$$f_{13} = q_{13} \oplus 1,\ f_{14} = q_{14} \oplus 1,\ f_{15} = q_{15} \oplus 1,\ f_{16} = q_{16} \oplus 1$$

The next step is to apply NS, which results from substituting the first four of the polynomials into the S-box polynomials to get 4 new polynomials, and repeating the process with the next four and the next four and the next four. This yields an ordered list of 16 more polynomials,

$$(sbox_1(f_1, ..., f_4), sbox_2(f_1, ..., f_4), ..., sbox_4(f_{13}, ..., f_{16})) = (g_1, ..., g_{16}).$$

In our example, this results in the list of polynomials:

$g_1 = q_1q_2 \oplus q_1q_3q_4 \oplus q_1q_4 \oplus q_1 \oplus q_2q_3q_4 \oplus q_3q_4 \oplus q_4 \oplus 1,$

$g_2 = q_1q_2q_4 \oplus q_1q_2 \oplus q_1q_3 \oplus q_1 \oplus q_2q_3q_4 \oplus q_2q_3 \oplus q_2 \oplus q_3q_4 \oplus q_4,$

$g_3 = q_1q_2q_3 \oplus q_1q_2q_4 \oplus q_1q_2 \oplus q_1q_3q_4 \oplus q_1 \oplus q_2q_3 \oplus q_3,$

$g_4 = q_1q_2q_3 \oplus q_1q_2q_4 \oplus q_1q_3q_4 \oplus q_1q_3 \oplus q_1q_4 \oplus q_1 \oplus q_2q_3q_4 \oplus q_2q_4 \oplus q_3 \oplus q_4 \oplus 1,$

$g_5 = q_5q_6 \oplus q_5q_7q_8 \oplus q_5 \oplus q_6q_7q_8 \oplus q_6q_8 \oplus q_6,$

$g_6 = q_5q_6q_8 \oplus q_5q_6 \oplus q_5q_7 \oplus q_6q_7q_8 \oplus q_6q_7 \oplus q_6 \oplus q_7q_8 \oplus q_7,$

$g_7 = q_5q_6q_7 \oplus q_5q_6q_8 \oplus q_5q_7q_8 \oplus q_5q_8 \oplus q_5 \oplus_6 q_8 \oplus q_6 \oplus q_7q_8 \oplus q_7 \oplus q_8,$

$g_8 = q_5q_6q_7 \oplus q_5q_6q_8 \oplus q_5q_6 \oplus q_5q_7q_8 \oplus q_5q_7 \oplus q_6q_7q_8 \oplus q_6q_7 \oplus q_6q_8 \oplus q_6 \oplus q_7q_8 \oplus q_8,$

$g_9 = q_9q_{10} \oplus q_9q_{11}q_{12} \oplus q_9q_{12} \oplus q_9 \oplus q_{10}q_{11}q_{12} \oplus q_{11}q_{12} \oplus q_{12} \oplus 1,$

$g_{10} = q_9q_{10}q_{12} \oplus q_9q_{10} \oplus q_9q_{11} \oplus q_9 \oplus q_{10}q_{11}q_{12} \oplus q_{10}q_{11} \oplus q_{10} \oplus q_{11}q_{12} \oplus q_{12},$

$g_{11} = q_9q_{10}q_{11} \oplus q_9q_{10}q_{12} \oplus q_9q_{10} \oplus q_9q_{11}q_{12} \oplus q_9 \oplus q_{10}q_{11} \oplus q_{11},$

$g_{12} = q_9q_{10}q_{11} \oplus q_9q_{10}q_{12} \oplus q_9q_{11}q_{12} \oplus q_9q_{11} \oplus q_9q_{12} \oplus q_9 \oplus q_{10}q_{11}q_{12} \oplus q_{10}q_{12} \oplus q_{11} \oplus q_{12} \oplus 1,$

$g_{13} = q_{13}q_{14} \oplus q_{13}q_{15}q_{16} \oplus q_{13}q_{15} \oplus q_{13}q_{16} \oplus q_{13} \oplus q_{14}q_{15}q_{16} \oplus q_{14}q_{15} \oplus q_{14} \oplus q_{15}q_{16} \oplus q_{15} \oplus 1,$

$g_{14} = q_{13}q_{14}q_{16} \oplus q_{13}q_{15} \oplus q_{13}q_{16} \oplus q_{13} \oplus q_{14}q_{15}q_{16} \oplus q_{14}q_{16} \oplus q_{14} \oplus 1,$

$g_{15} = q_{13}q_{14}q_{15} \oplus q_{13}q_{14}q_{16} \oplus q_{13}q_{15}q_{16} \oplus q_{13}q_{16} \oplus q_{13} \oplus q_{14}q_{16} \oplus q_{15}q_{16} \oplus q_{16} \oplus 1,$

$g_{16} = q_{13}q_{14}q_{15} \oplus q_{13}q_{14}q_{16} \oplus q_{13}q_{14} \oplus q_{13}q_{15}q_{16} \oplus q_{13}q_{15} \oplus q_{13} \oplus q_{14}q_{15}q_{16} \oplus q_{15}.$

The next step is to apply SR, which simply changes the order of some of the polynomials – transforming $g_1, ..., g_{16}$ into $g_1, ..., g_4, g_{13}, ..., g_{16}, g_9, ..., g_{12}, g_5, ..., g_8$.

Then MC is applied; this can be modeled by a simple linear transformation on the vector of polynomials that gives us a new list of polynomials, $(h_1, ..., h_{16})$.

In our example, this gives us the set of polynomials:

$h_1 = q_1q_2 \oplus q_1q_3q_4 \oplus q_2q_3q_4 \oplus q_2q_4 \oplus q_4 \oplus q_{13}q_{14}q_{15} \oplus q_{13}q_{14}q_{16} \oplus q_{13}q_{14} \oplus q_{13}q_{15}q_{16} \oplus q_{13}q_{15} \oplus q_{13}q_{16} \oplus q_{14}q_{15} \oplus 1$,

$h_2 = q_1q_2q_4 \oplus q_1q_2 \oplus q_1q_3 \oplus q_1q_4 \oplus q_1 \oplus q_2q_3q_4 \oplus q_2q_3 \oplus q_2q_4 \oplus q_3 \oplus q_4 \oplus q_{13}q_{14}q_{15} \oplus q_{13}q_{14}q_{16} \oplus q_{13}q_{14} \oplus q_{13}q_{16} \oplus q_{13} \oplus q_{14}q_{16} \oplus q_{15}q_{16} \oplus q_{15} \oplus q_{16}$,

$h_3 = q_1q_2q_3 \oplus q_1q_2q_4 \oplus q_1q_3q_4 \oplus q_1q_3 \oplus q_1 \oplus q_2q_3 \oplus q_2 \oplus q_{13}q_{14}q_{16} \oplus q_{13}q_{15}q_{16} \oplus q_{13}q_{15} \oplus q_{14}q_{15} \oplus q_{14} \oplus q_{15}$,

$h_4 = q_1q_2q_3 \oplus q_1q_2q_4 \oplus q_1q_2 \oplus q_1q_3q_4 \oplus q_1q_4 \oplus q_1 \oplus q_2q_3q_4 \oplus q_3q_4 \oplus q_3 \oplus q_4 \oplus q_{13}q_{14}q_{16} \oplus q_{13}q_{14} \oplus q_{13}q_{15} \oplus q_{13}q_{16} \oplus q_{14}q_{15}q_{16} \oplus q_{14}q_{15} \oplus q_{14} \oplus q_{15} \oplus q_{16} \oplus 1$,

$h_5 = q_1q_2q_3 \oplus q_1q_2q_4 \oplus q_1q_3q_4 \oplus q_1q_3 \oplus q_1 \oplus q_2q_3 \oplus q_2 \oplus q_{13}q_{14} \oplus q_{13}q_{15}q_{16} \oplus q_{13}q_{16} \oplus q_{14}q_{15}q_{16} \oplus q_{16} \oplus 1$,

$h_6 = q_1q_2q_3 \oplus q_1q_2q_4 \oplus q_1q_4 \oplus q_1 \oplus q_2q_4 \oplus q_3q_4 \oplus q_3 \oplus q_{13}q_{14}q_{16} \oplus q_{13}q_{14} \oplus q_{13}q_{15} \oplus q_{13}q_{16} \oplus q_{14}q_{15}q_{16} \oplus q_{14}q_{15} \oplus q_{14} \oplus q_{15} \oplus q_{16}$,

$h_7 = q_1q_2q_4 \oplus q_1q_3q_4 \oplus q_1q_3 \oplus q_1q_4 \oplus q_1 \oplus q_2q_3 \oplus q_3 \oplus q_{13}q_{14}q_{15} \oplus q_{13}q_{14}q_{16} \oplus q_{13}q_{14} \oplus q_{13}q_{15}q_{16} \oplus q_{13}q_{15} \oplus q_{13}q_{16} \oplus q_{14}q_{15} \oplus 1$,

$h_8 = q_1q_2q_4 \oplus q_1q_2 \oplus q_1q_3 \oplus q_1q_4 \oplus q_1 \oplus q_2q_3q_4 \oplus q_2q_3 \oplus q_2q_4 \oplus q_3 \oplus q_4 \oplus q_{13}q_{14}q_{15} \oplus q_{13}q_{14}q_{16} \oplus q_{13}q_{15}q_{16} \oplus q_{13} \oplus q_{14}q_{15}q_{16} \oplus q_{14}q_{16} \oplus q_{15}q_{16} \oplus q_{15} \oplus 1$,

$h_9 = q_5q_6q_7 \oplus q_5q_6q_8 \oplus q_5q_7q_8 \oplus q_5q_8 \oplus q_5 \oplus q_6q_8 \oplus q_6 \oplus q_7q_8 \oplus q_7 \oplus q_8 \oplus q_9q_{10} \oplus q_9q_{11}q_{12} \oplus q_9q_{11} \oplus q_{10}q_{11}q_{12} \oplus q_{10}q_{11} \oplus q_{10}q_{12} \oplus q_{10} \oplus q_{12}$,

$h_{10} = q_5q_6q_7 \oplus q_5q_6q_8 \oplus q_5q_7 \oplus q_5 \oplus q_6q_7 \oplus q_7q_8 \oplus_8 \oplus q_9q_{10}q_{12} \oplus q_9q_{11} \oplus q_9q_{12} \oplus q_{10}q_{11}q_{12} \oplus q_{10}q_{12} \oplus q_{10} \oplus q_{11} \oplus q_{12} \oplus 1$,

$h_{11} = q_5q_6q_8 \oplus q_5q_7q_8 \oplus q_5q_7 \oplus q_5 \oplus q_6q_7 \oplus q_6q_8 \oplus q_7q_8 \oplus q_7 \oplus q_9q_{10}q_{11} \oplus q_9q_{10}q_{12} \oplus q_9q_{10} \oplus q_9q_{11}q_{12} \oplus q_9 \oplus q_{10}q_{11} \oplus q_{10}$,

$h_{12} = q_5q_6q_8 \oplus q_5q_6 \oplus q_5q_7 \oplus q_6q_7q_8 \oplus q_6q_7 \oplus q_6 \oplus q_7q_8 \oplus q_7 \oplus q_9q_{10}q_{11} \oplus q_9q_{10}q_{12} \oplus q_9q_{11}q_{12} \oplus q_9q_{11} \oplus q_9q_{12} \oplus q_{10}q_{11}q_{12} \oplus q_{10}q_{11} \oplus q_{11}q_{12} \oplus q_{12} \oplus 1$,

$h_{13} = q_5q_6 \oplus q_5q_7q_8 \oplus q_5 \oplus q_6q_7q_8 \oplus q_6q_8 \oplus q_6 \oplus q_9q_{10}q_{11} \oplus q_9q_{10}q_{12} \oplus q_9q_{10} \oplus q_9q_{11}q_{12} \oplus q_9 \oplus q_{10}q_{11} \oplus q_{10}$,

$h_{14} = q_5q_6q_8 \oplus q_5q_6 \oplus q_5q_7 \oplus q_6q_7q_8 \oplus q_6q_7 \oplus q_6 \oplus q_7q_8 \oplus q_7 \oplus q_9q_{10}q_{11} \oplus q_9q_{10}q_{12} \oplus q_9q_{10} \oplus q_9q_{12} \oplus q_{10}q_{12} \oplus q_{10} \oplus q_{11}q_{12} \oplus 1$,

$h_{15} = q_5q_6q_7 \oplus q_5q_6q_8 \oplus q_5q_7q_8 \oplus q_5q_8 \oplus q_5 \oplus q_6q_8 \oplus q_6 \oplus q_7q_8 \oplus q_7 \oplus q_8 \oplus q_9q_{10}q_{12} \oplus q_9q_{10} \oplus q_9q_{11}q_{12} \oplus q_9q_{12} \oplus q_{10}q_{11} \oplus q_{11} \oplus 1$,

$h_{16} = q_5q_6q_7 \oplus q_5q_6q_8 \oplus q_5q_6 \oplus q_5q_7q_8 \oplus q_5q_7 \oplus q_6q_7q_8 \oplus q_6q_7 \oplus q_6q_8 \oplus q_6 \oplus q_7q_8 \oplus q_8 \oplus q_9q_{10}q_{12} \oplus q_9q_{11} \oplus q_9q_{12} \oplus q_{10}q_{11}q_{12} \oplus q_{10}q_{12} \oplus q_{10} \oplus q_{11} \oplus q_{12} \oplus 1$.

We continue this process until all the components of S-AES encryption have been applied. (At this point we must abandon the example because the resulting polynomials become too large to display.) The result is an ordered list of 16 polynomials in terms of the key variables, $(p_1(q_1, ..., q_{16}), ..., p_{16}(q_1, ..., q_{16}))$. By its construction, it is easy to see that this system of polynomials models the S-AES encryption: For example, if one were to take the key 16 bit key 1111111111111111 and substitute it into the polynomials, one would get $(p_1(1, ..., 1), ..., p_{16}(1, ..., 1))$, which is exactly the cipher text that results by applying S-AES to the plaintext $(m_1, ..., m_{16})$ using the all 1's key. We do not know what the key is, but we do know the plaintext $M = m_1, ..., m_{16}$ and the corresponding ciphertext, $C = c_1, .., c_{16}$. Therefore, we can set up a system of equations, namely $(p_1(q_1, ..., q_{16}), ..., p_{16}(q_1, ..., q_{16})) = (c_1, ..., c_{16})$,

or, equivalently, $(p_1(q_1, ..., q_{16}) - c_1, ..., p_{16}(q_1, ..., q_{16}) - c_{16}) = (0, ..., 0)$ (We write it in the second form so we can solve it using Gröbner basis, as described below.)

It is often helpful to repeat the above process of generating polynomials with multiple known plaintext/ciphertext pairs; doing that would result in a larger number of polynomials to use when solving for the key. In the next section, we will display results based upon the number of known plaintext input and output pairs n.

3. Gröbner Basis

The key bits, which are the solution of the system of polynomial equations, were obtained by using the *variety* command of the powerful F4 Gröbner Basis algorithm of the computer algebra software Magma[2]. As long as we used at least 8 plaintext/ciphertext pairs, we were able to solve the equations, and we always obtained an unique solution. The following data describes how long it took to solve for the key bits given n random plaintext/ciphertext pairs and the key consisting of all 0's, run with Magma on a desktop computer[3]. Two trials were conducted. Times are in seconds.

n	$Trial1$	$Trial2$
8	329.562	106.063
9	108.234	148.172
10	72.719	71.562
11	40.563	41.203
12	34.797	35.422
13	36.343	33.391
14	30.844	26.734
15	27.719	46.812
16	34.375	38.875

4. Improvements

Our method of creating the polynomials for S-AES is quite simple, and can be easily improved using methods demonstrated in [1]. For example, instead of creating a set of polynomials of the form $f(plaintext) - ciphertext = 0$, as done in our cryptanalysis, intermediate variables could be used or polynomials of the form $f(plaintext, ciphertext) = 0$ could be created.

An example of what might speed up this process occurs in modeling the S-Box. As we noted earlier, the S-Box can be thought of in different ways. The method we use here to implement the S-Box involves treating it as a look up table, which is probably the simplest method to implement. Another possibility would be to explicitly model the inversion step $q = p^{-1}$. This would require some extra variables (so called, intermediate variables), and would insert simple quadratic relations into the system of equations: relations of the form $X = Y^{-1}$ or, equivalently, $XY = 1$. Inversion would then be accomplished by solving the polynomial $X^2Y = X$ in $GF(4)$ as part of the solution of the system of equations. This might possibly speed up the process of solving the system of polynomials.

[2]See [1] for a discussion of some equation solving algorithms including F4.

[3]The specifications for the computer that was used are: Dell Optiplex 745, Intel Q965 chipset, 2.4 GHz Intel Pentium-D 925 CPU, 2 GB PC2-5300 (333 MHz) DDR2 RAM.

5. Algebraic Cryptanalysis

Algebraic cryptanalysis is a potentially powerful attack on symmetric key block ciphers. Algebraic cryptanalysis begins by constructing a (probably quite large) system of (mostly nonlinear) polynomial equations in terms of key bits, plaintext bits, and ciphertext bits and then attempts to solve that system by using powerful equation solving software. The number of variables, the number of polynomials, and the degrees of the polynomials; the power of the equation solver; and the speed and amount of memory of the computer being used determines whether the system can be solved for the key bits. Whether AES is really threatened by algebraic cryptanalysis remains a question, but S-AES is quickly broken by the powerful F4 Gröbner Basis algorithm of Magma, even when naively used.

References

1. Albrecht, M. 2008. "Algebraic Attacks on the Courtois Toy Cipher," *Cryptologia* 32(3), 220 - 276.
2. Biham, E. and Shamir, A. 1990. "Differential Cryptanalysis of DES-like Cryptosystems," *Advances in Cryptology – CRYPTO '90*, Springer-Verlag, 2 - 21.
3. Biham, E. and Shamir, A. 1991. "Differential Cryptanalysis of the Full 16-Round DES," *Proceedings of CRYPTO '92*, Lecture Notes in Computer Science 740, Springer-Verlag, 2 - 21.
4. Biham, E. and Shamir, A. 1993. *Differential Cryptanalysis of the Data Encryption Standard*, Springer-Verlag, New York.
5. Bosma, W., Cannon, J., and Playoust, C. 1997. "The Magma algebra system I. The user language," *Journal of Symbolic Computation* 34(3 - 4), 235 - 265.
6. Matsui, M. 1994. "Linear cryptanalysis method for DES cipher," *Advances in Cryptology – EUROCRYPT '93*, Lecture Notes in Computer Science 765, Springer-Verlag, 386 - 397.
7. Musa, M., Schaefer, E. F., and Wedig, S. 2003. "A Simplified AES Algorithm and its Linear and Differential Cryptanalysis," *Cryptologia* 27(2), 148 - 177.
8. Shannon, C. E. 1949. "Communication Theory of Secrecy Systems," *Bell System Technical Journal* 28, 656 - 715.
9. Stallings, W. 2002. "The Advanced Encryption Standard," *Cryptologia* 26(3), 165 - 188.

University of Texas
E-mail address: seankenneths@mail.utexas.edu