Boolean Functions
Algebraic attacks

Christina Boura

École de printemps en codage et cryptographie
May 18, 2016

Outline

Boolean functions

Inspired by Anne Canteaut's Lecture Notes

A Boolean function f of n variables is a function

$$\begin{array}{rcl} f : \mathbf{F}_2^n & \to & \mathbf{F}_2 \\ x = (x_1, \ldots, x_n) & \mapsto & f(x) \end{array}$$

Value vector: Binary vector v_f of length 2^n composed of all values $f(x)$, for $x \in \mathbf{F}_2^n$.

Example: $f : \mathbf{F}_2^3 \to \mathbf{F}_2$

$$v_f = (f(1,1,1), f(1,1,0), f(1,0,1), f(1,0,0), f(0,1,1), f(0,1,0), f(0,0,1), f(0,0,0))$$

$$v_f = (1,0,0,1,1,0,1,0)$$

Truth table

x_1	1	0	1	0	1	0	1	0
x_2	1	1	0	0	1	1	0	0
x_3	1	1	1	1	0	0	0	0
$f(x_1, x_2, x_3)$	1	0	0	1	1	0	1	0

Question

Question: How many different Boolean functions of n variables exist?

Question

Question: How many different Boolean functions of n variables exist?

$$2^{2^n}$$

Hamming weight of a Boolean function

Let $f : \mathbf{F}_2^n \to \mathbf{F}_2$.

The Hamming weight of f is defined as the number of 1's in v_f.

$$wt(f) = wt(v_f) = \#\{x \in \mathbf{F}_2^n : f(x) \neq 0\}$$

- For many cryptographic applications, we need Boolean functions that have a behaviour close to random functions.
  $\Rightarrow$ Use balanced functions.

$$f \text{ is balanced} \Leftrightarrow wt(f) = 2^{n-1}$$

Balancedness and bias

Let $f : \mathbf{F}_2^n \to \mathbf{F}_2$. The bias of f is

$$\begin{aligned}
\mathcal{E}(f) &= \sum_{x \in \mathbf{F}_2^n} (-1)^{f(x)} \\
&= \#\{x \in \mathbf{F}_2^n : f(x) = 0\} - \#\{x \in \mathbf{F}_2^n : f(x) = 1\} \\
&= 2^n - \#\{x \in \mathbf{F}_2^n : f(x) = 1\} - \#\{x \in \mathbf{F}_2^n : f(x) = 1\} \\
&= 2^n - 2wt(f)
\end{aligned}$$

f is balanced $\Leftrightarrow \mathcal{E}(f) = 0$

Alternative representation of a Boolean function

Representation of a Boolean function, where the function is seen as a multivariate polynomial.

In $\mathbf{F}_2$:

- $+$: XOR
- $\times$: AND
- $x_i^2 = x_i$ (as $0^2 = 0$ and $1^2 = 1$)

Monomial in $\mathbf{F}_2[x_1, \ldots, x_n]/(x_1^2 + x_1, \ldots, x_n^2 + x_n)$: product of distinct variables

Examples: x_1, x_3x_4, $x_2x_4x_5$, $x_1x_2 \ldots x_n$

Monomials

Notation : Monomial in $\mathbf{F}_2[x_1, \ldots, x_n]/(x_1^2 + x_1, \ldots, x_n^2 + x_n)$:

$$x^u = \prod_{i=1}^{n} x_i^{u_i},$$

where $u = (u_1, \ldots, u_n) \in \mathbf{F}_2^n$.

Example: $x \in \mathbf{F}_2^4$: $x^{1010} = x_1^1 x_2^0 x_3^1 x_4^0 = x_1 x_3$

Algebraic normal form (ANF)

Proposition: Any $f : \mathbf{F}_2^n \to \mathbf{F}_2$ can be uniquely written as a multivariate polynomial in $\mathbf{F}_2[x_1, \ldots, x_n]/(x_1^2 + x_1, \ldots, x_n^2 + x_n)$:

$$f(x_1, \ldots, x_n) = \sum_{u \in \mathbf{F}_2^n} a_u x^u, \text{ where } a_u \in \mathbf{F}_2.$$

This polynomial is called the Algebraic Normal Form (ANF) of f.

The coefficients a_u can be computed as follows:

$$a_u = \sum_{x \preceq u} f(x),$$

where $x \preceq u \Leftrightarrow x_i \leq u_i$, pour $1 \leq i \leq n$

Example

(x_1, x_2, x_3)	$(1,1,1)$	$(0,1,1)$	$(1,0,1)$	$(0,0,1)$	$(1,1,0)$	$(0,1,0)$	$(1,0,0)$	$(0,0,0)$
$f(x_1, x_2, x_3)$	1	0	0	1	1	0	1	0

- $a_{000} = f(0,0,0) = 0$
- $a_{100} = f(1,0,0) + f(0,0,0) = 1 + 0 = 1$
- $a_{010} = f(0,1,0) + f(0,0,0) = 0 + 0 = 0$
- $a_{110} = f(1,1,0) + f(0,1,0) + f(1,0,0) + f(0,0,0) = 1 + 0 + 1 + 0 = 0$
- $a_{001} = f(0,0,1) + f(0,0,0) = 1 + 0 = 1$
- $a_{101} = f(1,0,1) + f(1,0,0) + f(0,0,1) + f(0,0,0) = 0 + 1 + 1 + 0 = 0$
- $a_{011} = f(0,1,1) + f(0,1,0) + f(0,0,1) + f(0,0,0) = 0 + 0 + 1 + 0 = 1$
- $a_{111} = \sum_{x \in \mathbf{F}_2^3} f(x) = wt(f) \mod 2 = 0$

Example

(x_1, x_2, x_3)	$(1,1,1)$	$(0,1,1)$	$(1,0,1)$	$(0,0,1)$	$(1,1,0)$	$(0,1,0)$	$(1,0,0)$	$(0,0,0)$
$f(x_1, x_2, x_3)$	1	0	0	1	1	0	1	0

- $a_{000} = f(0,0,0) = 0$
- $a_{100} = f(1,0,0) + f(0,0,0) = 1 + 0 = 1$
- $a_{010} = f(0,1,0) + f(0,0,0) = 0 + 0 = 0$
- $a_{110} = f(1,1,0) + f(0,1,0) + f(1,0,0) + f(0,0,0) = 1 + 0 + 1 + 0 = 0$
- $a_{001} = f(0,0,1) + f(0,0,0) = 1 + 0 = 1$
- $a_{101} = f(1,0,1) + f(1,0,0) + f(0,0,1) + f(0,0,0) = 0 + 1 + 1 + 0 = 0$
- $a_{011} = f(0,1,1) + f(0,1,0) + f(0,0,1) + f(0,0,0) = 0 + 0 + 1 + 0 = 1$
- $a_{111} = \sum_{x \in \mathbf{F}_2^3} f(x) = wt(f) \mod 2 = 0$

$$f(x_1, x_2, x_3) = x_1 + x_3 + x_2 x_3$$

Degree of a Boolean function

The algebraic degree of a Boolean function f is defined as

$$\deg(f) = \max_{u \in \mathbf{F}_2^n} \{wt(u) : a_u \neq 0\}$$

Example: $f(x_1, x_2, x_3) = x_1x_2x_3 + x_1x_3 + x_1 + 1.$

$$\deg(f) = 3$$

Functions of degree n

Let $f : \mathbf{F}_2^n \to \mathbf{F}_2$.

$$a_{1\ldots1} = \bigoplus_{x \in \mathbf{F}_2^n} f(x) = wt(f) \mod 2$$

$\deg(f) = n$ **iff** $wt(f)$ is odd.

Functions of maximum degree are not balanced.

- Maximal degree functions are not used in cryptographic applications.

Affine functions

Let $f : \mathbf{F}_2^n \to \mathbf{F}_2$ of degree 1. Then,

$$wt(f) = 2^{n-1}.$$

- Affine functions are balanced.

Let $f = b \cdot x + \varepsilon$, with $b \in \mathbf{F}_2^n \setminus \{0\}$ and $\varepsilon \in \mathbf{F}_2$.

- If $\varepsilon = 1$, $f(x) = 1$ iff $b \cdot x = 0$ iff $x \in \langle b \rangle^\perp$ (hyperplane)
- If $\varepsilon = 0$, $f(x) = 1$ iff $b \cdot x = 1$ iff $x \in \mathbf{F}_2^n \setminus \langle b \rangle^\perp$

Cryptographic Sboxes

An Sbox S from $\mathbf{F}_2^n$ into $\mathbf{F}_2^m$ is a collection of m Boolean functions of n variables.

Example (PRESENT Sbox $S : \mathbf{F}_2^4 \to \mathbf{F}_2^4$)

x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
$S(x)$	c	5	6	b	9	0	a	d	3	e	f	8	4	7	1	2
$S_1(x)$	0	1	0	1	1	0	0	1	1	0	1	0	0	1	1	0
$S_2(x)$	0	0	1	1	0	0	1	0	1	1	1	0	0	1	0	1
$S_3(x)$	1	1	1	0	0	0	0	1	0	1	1	0	1	1	0	0
$S_4(x)$	1	0	0	1	1	0	1	1	0	1	1	1	0	0	0	0

ANF of the Sbox

$$
\begin{aligned}
S_1 &= x_1 + x_3 + x_4 + x_2x_3 \\
S_2 &= x_2 + x_4 + x_2x_4 + x_3x_4 + x_1x_2x_3 + x_1x_2x_4 + x_1x_3x_4 \\
S_3 &= 1 + x_3 + x_4 + x_1x_2 + x_1x_4 + x_2x_4 + x_1x_2x_4 + x_1x_3x_4 \\
S_4 &= 1 + x_1 + x_2 + x_4 + x_2x_3 + x_1x_2x_3 + x_1x_2x_4 + x_1x_3x_4
\end{aligned}
$$

- The functions $S_1, \ldots, S_m$ are called the coordinates of the Sbox.

Components of the Sbox

Let $S : \mathbf{F}_2^n \to \mathbf{F}_2^m$.
The components of the Sbox are the n-variable Boolean functions

$$S_\lambda : x \mapsto \lambda \cdot S(x)$$

for all $\lambda \in \mathbf{F}_2^m$.
Examples:

- $S_3 = S_1 + S_2$
- $S_{15} = S_1 + S_2 + S_3 + S_4$

- The components of an Sbox offer a useful characterisation.

When an Sbox is a permutation

Let $S : \mathbf{F}_2^n \to \mathbf{F}_2^n$.

S is a permutation **iff** all its non-trivial components are balanced.

Proof. (S permutation $\Rightarrow$ S_λ are balanced)

Suppose S is a permutation and let $\lambda \neq 0$. Then,

$$\mathcal{E}(S_\lambda) = \sum_{x \in \mathbf{F}_2^n} (-1)^{\lambda \cdot S(x)} = \sum_{y \in \mathbf{F}_2^n} (-1)^{\lambda \cdot y} = 0.$$

Algebraic degree of an Sbox

Let $S : \mathbf{F}_2^n \to \mathbf{F}_2^m$.

The degree of S is the maximal degree of the ANF of its components.

Example: $S = (S_1, S_2, S_3, S_4)$

$$
\begin{aligned}
S_1 &= x_1 + x_3 + x_4 + x_2x_3 \\
S_2 &= x_2 + x_4 + x_2x_4 + x_3x_4 + x_1x_2x_3 + x_1x_2x_4 + x_1x_3x_4 \\
S_3 &= 1 + x_3 + x_4 + x_1x_2 + x_1x_4 + x_2x_4 + x_1x_2x_4 + x_1x_3x_4 \\
S_4 &= 1 + x_1 + x_2 + x_4 + x_2x_3 + x_1x_2x_3 + x_1x_2x_4 + x_1x_3x_4
\end{aligned}
$$

- $\deg(S) = 3$

Algebraic degree of a permutation

- Boolean functions of maximal degree are not balanced.
- An Sbox is a permutation iff all its non-trivial components are balanced.
- The degree of an Sbox is the maximal degree of its components.

Let $S : \mathbf{F}_2^n \to \mathbf{F}_2^n$.

$$S \text{ is a permutation} \Leftrightarrow \deg(S) \leq n - 1$$

Univariate representation

Identify the vector space $\mathbf{F}_2^n$ with the finite field $\mathbf{F}_{2^n}$.

$$S(X) = \sum_{i=0}^{2^n-1} b_i X^i,\ b_i \in \mathbf{F}_{2^n}.$$

Degree in the univariate representation

Let S be an n-bit Sbox and let

$$F(x) = \sum_{i=0}^{2^n-1} b_i x_i$$

be its univariate representation in $\mathbf{F}_{2^n}[x]$.

The degree of F is given by

$$\deg(F) = \max\{wt(i) : 0 \leq i < 2^n \text{ and } b_i \neq 0\}.$$

Outline

Basic algebraic attack

- Principle introduced by Claude Shannon in 1949.
- Express the whole cipher as a large system of multivariate algebraic equations.
- Known-plaintext attack
  - **Known** coefficients : plaintext and ciphertext bits
  - **Uknowns**: key bits

Solve the algebraic system and recover the secret key.

Linearization (I)

The **complexity** of the attack depends on the degree of the system.

A (naive) method for solving such a system: linearization.

Idea: Identify the system with a linear system of $\sum_{i=1}^{d} \binom{n}{i}$ variables,

where n is the block size. Each product of i initial variables, $1 \leq i \leq d$ is seen as a new variable.

Linearization (II)

Solve the linear system by linear algebra.

Complexity:

$$\left(\sum_{i=1}^{d}\binom{n}{i}\right)^{\omega} \approx n^{\omega},$$

where ω depends on the method used for the resolution ($\omega \approx 2.37$).

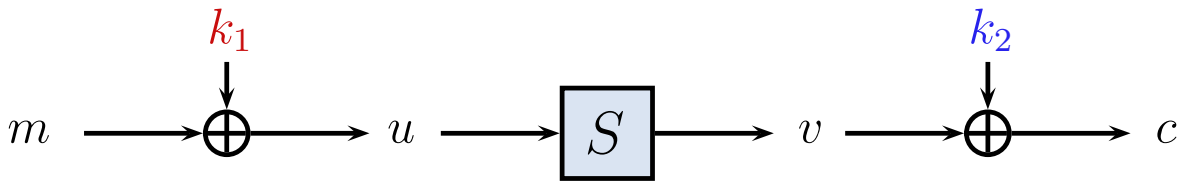
Other methods for solving the system:

- Gröbner basis algorithms
- ad-hoc techniques: XL, XSL.

Example on a toy cipher

Anne Canteaut's Lecture Notes

- Block size: $n = 4$ bits
- Key size: 8 bits



- $c = k_2 \oplus S(m \oplus k_1)$
- $c \oplus k_2 = S(m \oplus k_1)$

One plaintext-ciphertext pair gives 4 equations in 8 variables.

ANF of the Sbox

x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
S(x)	f	e	b	c	6	d	7	8	0	3	9	a	4	2	1	5

$$
\begin{aligned}
S_1 &= 1 + x_1 + x_3 + x_2x_3 + x_4 + x_2x_4 + x_3x_4 + x_1x_3x_4 + x_2x_3x_4 \\
S_2 &= 1 + x_1x_2 + x_1x_3 + x_1x_2x_3 + x_4 + x_1x_4 + x_1x_2x_4 + x_1x_3x_4 \\
S_3 &= 1 + x_2 + x_1x_2 + x_2x_3 + x_4 + x_2x_4 + x_1x_2x_4 + x_3x_4 + x_1x_3x_4 \\
S_4 &= 1 + x_3 + x_1x_3 + x_4 + x_2x_4 + x_3x_4 + x_1x_3x_4 + x_2x_3x_4
\end{aligned}
$$

Write down the equations

Express each ciphertext bit c_i, $1 \leq i \leq 4$, as a multivariate polynomial in the plaintext bits $m_1, \ldots, m_4$ and in the key bits $k_1, \ldots, k_8$.

$$
\begin{aligned}
c_1 + k_5 &= 1 + (m_1 + k_1) + (m_3 + k_3) + (m_2 + k_2)(m_3 + k_3) + (m_4 + k_4) \\
&+ (m_2 + k_2)(m_4 + k_4) + (m_3 + k_3)(m_4 + k_4) + (m_1 + k_1)(m_3 + k_3)(m_4 + k_4) \\
&+ (m_2 + k_2)(m_3 + k_3)(m_4 + k_4) \\
c_2 + k_6 &= 1 + (m_1 + k_1)(m_2 + k_2) + (m_1 + k_1)(m_3 + k_3) + (m_1 + k_1)(m_2 + k_2)(m_3 + k_3) \\
&+ (m_4 + k_4) + (m_1 + k_1)(m_4 + k_4) + (m_1 + k_1)(m_2 + k_2)(m_4 + k_4) \\
&+ (m_1 + k_1)(m_3 + k_3)(m_4 + k_4) \\
c_3 + k_7 &= 1 + (m_2 + k_2) + (m_1 + k_1)(m_2 + k_2) + (m_2 + k_2)(m_3 + k_3) + (m_4 + k_4) \\
&+ (m_2 + k_2)(m_4 + k_4) + (m_1 + k_1)(m_2 + k_2)(m_4 + k_4) + (m_3 + k_3)(m_4 + k_4) \\
&+ (m_1 + k_1)(m_3 + k_3)(m_4 + k_4) \\
c_4 + k_8 &= 1 + (m_3 + k_3) + (m_1 + k_1)(m_4 + k_3) + (m_4 + k_4) + (m_3 + k_2)(m_4 + k_4) \\
&+ (m_3 + k_3)(m_4 + k_4) + (m_2 + k_1)(m_3 + k_3)(m_4 + k_5) \\
&+ (m_2 + k_2)(m_3 + k_4)(m_4 + k_4)
\end{aligned}
$$

Re-write the equations

$$
\begin{aligned}
c_1 + k_5 &= S_1(m) + (1 + m_3m_4)k_1 + (m_3 + m_4 + m_3m_4)k_2 \\
&+ (1 + m_2 + m_4 + m_1m_4 + m_2m_4)k_3 \\
&+ (1 + m_2 + m_3 + m_1m_3 + m_2m_3)k_4 + m_4k_1k_3 + m_3k_1k_4 + (1 + m_4)k_2k_3 \\
&+ (1 + m_3)k_2k_4 + (1 + m_1 + m_2)k_3k_4 + k_1k_3k_4 + k_2k_3k_4 \\
c_2 + k_6 &= S_2(m) + (m_2 + m_3 + m_2m_3 + m_4 + m_2m_4 + m_3m_4)k_1 \\
&+ (m_1 + m_1m_3 + m_1m_4)k_2 + (m_1 + m_1m_2 + m_1m_4)k_3 \\
&+ (1 + m_1 + m_1m_2 + m_1m_3)k_4 + (1 + m_3 + m_4)k_1k_2 + (1 + m_2 + m_4)k_1k_3 \\
&+ (1 + m_2 + m_3)k_1k_4 + m_1k_2k_3 + m_1k_2k_4 + m_1k_3k_4 + k_1k_2k_3 + k_1k_2k_4 + k_1k_3k_4 \\
c_3 + k_7 &= S_3(m) + (m_2 + m_2m_4 + m_3m_4)k_1 + (1 + m_1 + m_3 + m_4 + m_1m_4)k_2 \\
&+ (m_2 + m_4 + m_1m_4)k_3 + (1 + m_2 + m_3 + m_1m_2 + m_1m_3)k_4 + (1 + m_4)k_1k_2 \\
&+ m_4k_1k_3 + (m_2 + m_3)k_1k_4 + k_2k_3 + m_1k_3k_4 + (1 + m_1)k_2k_4 + k_3k_4 + k_1k_2k_4 \\
&+ k_1k_3k_4 \\
c_4 + k_8 &= S_4(m) + (m_3 + m_3m_4)k_1 + (m_4 + m_3m_4)k_2 \\
&+ (1 + m_1 + m_4 + m_1m_4 + m_2m_4)k_3 + (1 + m_2 + m_3 + m_1m_3 + m_2m_3)k_4 \\
&+ (1 + m_4)k_1k_3 + (m_3)k_1k_4 + m_4k_2k_3 + (1 + m_3)k_2k_4 + (1 + m_1 + m_2)k_3k_4 \\
&+ k_1k_3k_4 + k_2k_3k_4
\end{aligned}
$$

Replace the known values

From the plaintext-ciphertext couple $(m, c) = (\texttt{0x0}, \texttt{0x4})$ we get

$$
\begin{aligned}
c_1 + k_5 &= 1 + k_1 + k_3 + k_4 + k_2k_3 + k_2k_4 + k_3k_4 + k_1k_3k_4 + k_2k_3k_4 \\
c_2 + k_6 &= 1 + k_4 + k_1k_2 + k_1k_3 + k_1k_4 + k_1k_2k_3 + k_1k_2k_4 + k_1k_3k_4 \\
c_3 + k_7 &= 1 + k_2 + k_4 + k_1k_2 + k_2k_3 + k_2k_4 + k_3k_4 + k_1k_2k_4 + k_1k_3k_4 \\
c_4 + k_8 &= 1 + k_3 + k_4 + k_1k_3 + k_2k_4 + k_3k_4 + k_1k_3k_4 + k_2k_3k_4
\end{aligned}
$$

- Polynomial system of degree $d = 3$ with 8 unknowns.

Linearize the system

Replace each monomial in the key bits of degree 2 or 3 with a **new** unknown:

$$k_9 = k_1k_2, k_{10} = k_1k_3, \ldots, k_{14} = k_3k_4, k_{15} = k_1k_2k_3, \ldots, k_{18} = k_2k_3k_4$$

$$\begin{aligned}
c_1 + k_5 &= 1 + k_1 + k_3 + k_4 + k_{12} + k_{13} + k_{14} + k_{16} + k_{18} \\
c_2 + k_6 &= 1 + k_4 + k_9 + k_{10} + k_{11} + k_{15} + k_{17} + k_{16} \\
c_3 + k_7 &= 1 + k_2 + k_4 + k_9 + k_{12} + k_{13} + k_{14} + k_{17} + k_{16} \\
c_4 + k_8 &= 1 + k_3 + k_4 + k_{10} + k_{13} + k_{14} + k_{16} + k_{18}
\end{aligned}$$

- Linear system with $8 + \binom{4}{2} + \binom{4}{3} = 18$ unknowns.

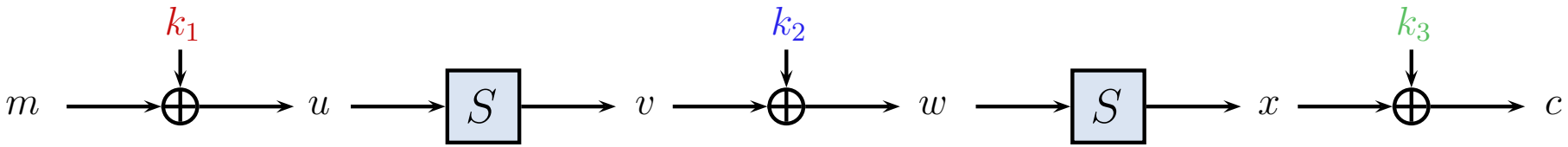
Solve the system

- Here, 5 (m, c) couples are enough to solve the system ($4 \times 5 = 20$ equations).
- In practice, block ciphers have a much larger block size and are composed of many rounds.
- The degree of the polynomial system increases with the number of rounds.

Solving such systems: infeasible even for a few rounds.

Alternative solution: use intermediate variables

- Use intermediate variables to control the degree of the system.



- Consider the 4 bits of v as additional unknowns.
- One known P-C pair gives 8 equations of degree 3 with 16 unknowns (12 key bits + 4 bits of v).
- For any additional P-C pair : $+4$ equations **but** $+4$ unknowns

N P-C pairs $\rightarrow$ $8N$ equations and $12 + 4N$ unknowns.

Advanced algebraic attack

Decrease the degree of the polynomial system even if the round function has a high degree.

Idea introduced by Courtois and Pieprzyk in 2002.

Example: Relations of degree 2 between inputs and outputs:

$$x_2x_4 + x_2S_1(x_1, \ldots, x_4) + x_2S_2(x_1, ..., x_4) = 0$$

We get then the following quadratic equation:

$$(m_4+c_1+c_2)k_2+m_2k_4+m_2k_5+m_2k_6+k_2k_4+k_2k_5+k_2k_6 = m_2m_4+m_2c_1+m_2c_2.$$

Relations of degree 2

- 21 linearly independent relations of degree 2 between the input and the output bits can be exhibited.
- System easier to solve than the original equations.

Question: What is the least number of linearly independent relations of degree at most d?

$$\sum_{i=0}^{d} \binom{2n}{i} - 2^n$$

Example

Any function from $\mathbf{F}_2^4$ into $\mathbf{F}_2^4$ has at least

$$\sum_{i=0}^{2} \binom{8}{i} - 2^4 = 37 - 16 = 21$$

quadratic relations between its inputs and outputs.

The case of AES (I)

The AES Sbox can be seen as the composition of the inversion over $\mathbf{F}_{2^8}$ with an affine function.

For the inverse operation, the input a and output b satisfy the relation

$$ab = 1$$

over $\mathbf{F}_{2^8}$.

$$\begin{aligned} & (a_7X^7 + a_6X^6 + a_5X^5 + a_4X^4 + a_3X^3 + a_2X^2 + a_1X + a_0) \\ \times \quad & (b_7X^7 + b_6X^6 + b_5X^5 + b_4X^4 + b_3X^3 + b_2X^2 + b_1X + b_0) \\ = \quad & 1 \end{aligned}$$

The case of AES (II)

Derive 8 multivariate quadratic equations over $\mathbf{F}_2$ (one for each coefficient of the previous equation).

Example

$$\begin{aligned} & a_0b_0 + a_7b_1 + a_6b_2 + a_5b_3 + a_4b_4 + a_3b_5 + a_2b_6 \\ + \quad & a_1b_7 + a_7b_6 + a_6b_7 + a_7b_5 + a_6b_6 + a_5b_7 \\ = \quad & 1. \end{aligned}$$

- Derive other equations by exploiting for example relations of the form $a^2b = a$ and $ab^2 = b$ over $\mathbf{F}_{2^8}$.

Quadratic system for AES

- There are in total 39 quadratic relations for the AES Sbox (much more than for a randomly chosen mapping over $\mathbf{F}_2^8$.
- Use these relations of degree 2 to form a quadratic system by introducing new variables for the outputs of successive rounds.
- 8000 quadratic equations of 1600 variables.

Solving the system

How to solve the resulting system?

- XSL (eXtended Sparse Linearisation): based on linearization, but attempting to exploit the sparsity and specific structure of the equation system.
- Gröbner Basis algorithms, SAT-solvers, etc.

Courtois and Pieprzyk claimed that by using XSL it was possible to mount an (at least theoretical) successful attack against AES-128.

However, it was shown by Cid and Leurent (Asiacrypt 05) that the algorithm did not work as expected, so one could not claim that AES was broken.

The limitations of algebraic attacks

- No well-known block cipher has been broken using pure algebraic techniques faster than with other techniques.
- Algebraic cryptanalysis works better in the case of stream ciphers and resistance against such attacks is a design criteria goal.

The applicability of an algebraic attack mainly depends on the algebraic degree of the block cipher.

Other attacks depending on the algebraic degree:

- Higher-order differential attacks, their derivatives and extensions.