Algebraic attacks and decomposition of Boolean functions

Willi Meier[1] and Enes Pasalic[2] and Claude Carlet[2]

[1] FH Aargau, Switzerland

[2] INRIA, Rocquencourt, France

Overview

- Algebraic attacks in general
- ... and on LFSR-based stream ciphers
- Scenarios
- New criterion: Immunity against algebraic attacks
- Problems solved on algebraic immunity
- Conclusions

Algebraic attacks known against

- Public key ciphers:

  Matsumoto-Imai (Patarin, 1995)
  HFE (Faugère-Joux, 2003)

- Block ciphers:

  AES, Serpent (Courtois-Pieprzyk, 2002)

- LFSR-based stream ciphers

Algebraic attack (Steps):

1. Set up system of equations:

   Multivariate algebraic equations of some degree

   System of equations, depends on cipher

   Involves plaintext, ciphertext and key

2. Solve system

   (Linearization, XL, Gröbner bases)

   Complexity depends on degree of equations

Solving systems of algebraic equations known to be hard in general

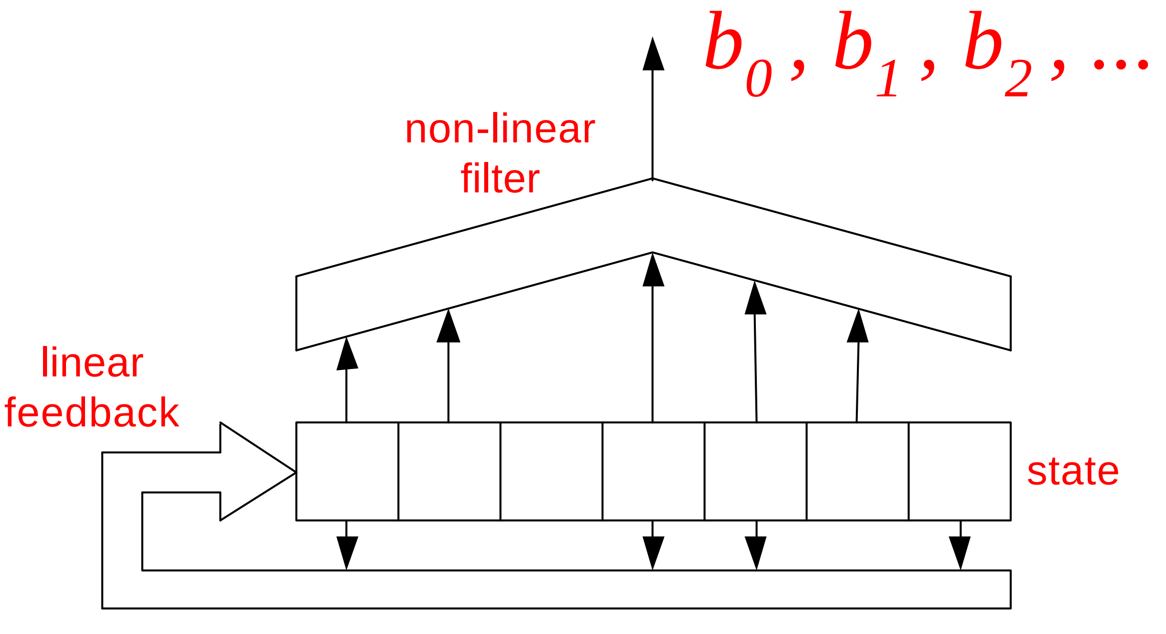
Search for:

- Equations of low degree
- Overdefined systems of equations

Under these conditions, solving is quite efficient

Algebraic attacks on LFSR-based stream ciphers

Example: Linear sequence generator plus combiner



System of Algebraic equations

$$\begin{cases} f(k_0,\ldots,k_{n-1}) & = b_0 \\ f(L(k_0,\ldots,k_{n-1})) & = b_1 \\ f(L^2(k_0,\ldots,k_{n-1})) & = b_2 \\ \ldots\ldots\ldots\ldots\ \ \ldots\ldots\ldots\ldots\ \ \ldots\ldots\ldots\ldots & \ldots\ldots \end{cases}$$

Is overdefined in known-plaintext attack.

However: Degree of equations too large.

Scenarios

Attempt: Lower degree of equations by multiplying combining function f with well chosen function g.

New result: Two scenarios suffice

S1: There exist functions g and h of low degree such that $f * g = h$

S2: There exists function g of low degree such that $f * g = 0$

Known result (Eurocrypt'03)

For any Boolean function f with n inputs there is a nonzero Boolean function g of degree at most $n/2$ such that $f * g$ is of degree at most $n/2$

Use of scenarios:

If output bit $b_i = 0$, use S1: $f * g = h$, i.e. get equation $h(x) = 0$

If output bit $b_i = 1$, use S2: $f * g = 0$, i.e. get equation $g(x) = 0$

Consequence: Class of stream ciphers is prone to algebraic attacks that were immune to all previous attacks.

Countermeasure: Choose combining function *f* with large number *n* of inputs, e. g., *n = 32*, to escape algebraic attacks.

But even then, no certainty whether no low degree multiples exist.

Contrast: Many stream ciphers proposed are provably secure against, e.g., Berlekamp-Massey shift register synthesis algorithm

New measure: Immunity against algebraic attacks

Recall S1: There exist g and h of low degree such that $f * g = h$

As $f^2 = f$ in GF(2),

$$f^2 * g = f * g = h,$$

and also

$$f^2 * g = f * h.$$

Hence $f * h = h$, or $(f+1) * h = 0$, i.e. we are in scenario S2, but for $f+1$ instead of f.

Notion:

Function g is called an annihilator of f if $f * g = 0$.

New measure:

Algebraic immunity, AI(f) of (combining) function f:

AI(f) is minimum value of d such that f or $f+1$ admits annihilator of degree d.

Problems on algebraic immunity

1. For given *f*, determine algebraic immunity of *f*
2. Probability that a random Boolean function has low algebraic immunity?
3. Classes of Boolean functions with low algebraic immunity?

Problem1

Known Algorithm for determining AI(f):

Assume f balanced. g of degree $d < n/2$.

Is g annihilator of f ?

Necessary and sufficient for $f * g=0$:
$g(x) = 0$ for all x for which $f(x) = 1$.

1. Substitute all these x in ANF of g

2. Obtain linear system of equations for coefficients of ANF of g.

3. If no solution: Print AI(f) $> d$

Large number of equations: 2^{n-1}

Complexity of solving: $2^{3(n-1)}$

Infeasible if number of inputs of f not small (e.g. if $n = 32$).

Idea: Equations are seen to have specific structure.

Substitute x with $f(x) = 1$ in $g(x) = 0$, but with increasing weight,

e.g. $x=(0,0,\ldots,0,1,0,\ldots 0)$, with 1 at i-th position.

Then for constant term a_0 and coefficients a_k of linear terms x_k, in ANF $(k=1,...,n)$, get linear equation

$$a_i + a_0 = 0$$

If x is of weight 2 and $f(x) = 1$, get equation

$$a_{ik} + a_i + a_k + a_0 = 0$$

More generally, for x of weight $w <= d$:
Only one coefficient of weight w does occur.

Use equation to express this coeff by coeff's of lower weight.

Assume f random:

Then for about half of arguments x, $f(x)=1$.

Roughly half of the a_{ik}'s can be expressed by coefficients of monomials of lower weight.

Reduces number of unknowns by factor 1/2.

Need additional equations: Choose random arguments x with $f(x) = 1$, until there are same number of equations as unknowns.

Solve system: Get reduction of complexity by factor 8.

Further improvements?

Use arguments x of weight $w = d+1, d+2, \ldots$

E.g., for x of weight $w=d+1$, $d+1$ weight d coeff‘s involved.

For some fraction of favorable arguments x, exactly d of these coeff‘s were already expressed by coeff‘s of lower weight.

Express remaining coeff by coeff‘s of lower weight as well.

Estimation of fraction of favorable arguments x for general degree d and number n of inputs of f shows:

This type of elimination of coeff's works well if $d < 6$, but will not work for $d >= 6$.

Case $d = 5$, $n = 32$: Can reduce complexity of solving linear equations from order 2^{53} to order 2^{45}.

For $d < 5$, reduction of complexity even larger.

Practical relevance of this result for realistic combiners (i.e., number *n* of inputs large):

1. If for combining function f (or $f+1$), an annihilator of degree $d <= 4$ is found by our algorithm, stream cipher is prone to algebraic attack.

2. If f and $f+1$ are shown to have no annihilators of degree $d < 6$, cipher has some immunity against algebraic attack:
   For $d = 6$, and for 128-bit key, computational complexity of basic attack is of order 2^{96}.

Problem 2:
Probability that a random Boolean function has low algebraic immunity

Exact determination of algebraic immunity still not feasible if $n \geq 32$ and $d \geq 6$.

Derive several bounds on probability that random balanced function has $\mathrm{AI}(f) \leq d$.

Estimates partly use results from coding theory.

Asymptotic bound for random Boolean functions with n inputs:

There is a constant, c, $c \approx 0.22$, such that for any sequence d_n of positive integers with $d_n <= c * n$,

$Pb\{AI(f) <= d_n\}$ goes to 0 as n goes to *infinity*

Bound gives good estimates already for moderate n

Result:

For random function *f* with large number *n* of inputs (e.g. *n >= 18*), low algebraic immunity is extremely unlikely.

	d = 5	d = 6	d = 7	d = 8
n Pb	18 10^{-1134}	22 10^{-6326}	26 10^{-23138}	31 10^{-10^7}

Pb: Probability that AI(*f*) <= *d*

Conclude: Low algebraic immunity of combining function in some stream ciphers not likely, but caused (presumably) by

- Requirement of implementation to be efficient
- Potential tradeoff between established design criteria and new criterion of algebraic immunity

Problem 3:
Boolean functions with relatively low algebraic immunity

Tradeoff between new criterium of high algebraic immunity and established criteria?

Known criteria:

- Large algebraic degree (to counter Berlekamp-Massey)

- Correlation immunity (to counter correlation attacks)

- Large distance to affine functions

Degree optimized Maiorana-McFarland functions:

Satisfy several desirable criteria. However:

Functions in this class can have relatively low algebraic immunity.

Result is consequence of useful representation of annihilators of given function:

Annihilator viewed as concatenation of annihilators from smaller variable space.

Conclusions

- Efficient algorithm for determining algebraic immmunity of Boolean functions:
  Significant step towards provable security against algebraic attacks.
- For random functions with many inputs:
  Low algebraic immunity is very unlikely.
- Functions exist, with desirable properties, but with relatively low alg. immunity:
  Suggests tradeoff between new and established criteria.