3 Algebraic Cryptanalysis

Attacks with Known Plaintext

Consider a bitblock cipher, given by the map

$$F\colon \mathbb{F}_2^n \times \mathbb{F}_2^l \longrightarrow \mathbb{F}_2^n$$

Then F is an n-tuple $F = (F_1, \ldots, F_n)$ of polynomial functions in $n + l$ variables all of whose partial degrees are ≤ 1.

An attack with known plaintext $a \in \mathbb{F}_2^n$ and corresponding ciphertext $c \in \mathbb{F}_2^n$ leads to a system

$$F(a, x) = c$$

of n polynomial equations for the unknown key $x \in \mathbb{F}_2^l$.

Systems of polynomial equations (over arbitrary fields) are one of the subjects of algebraic geometry. A rule of thumb says

The solution set for x has dimension 0 "in general" for $n \geq l$.

(I. e. it consists of a few isolated solutions. Otherwise, if the solution set allows for free parameters—or has dimension ≥ 1—, the attacker needs some more blocks of known plaintext.)

The general theory of polynomial equations is quite deep, in particular if we search for concrete solution procedures. But maybe the observation that only partial degrees ≤ 1 occur makes a difference?

Examples

Example 1: Let $n = l = 2$,

$$F(T_1, T_2, X_1, X_2) = (T_1 + T_2X_1, T_2 + T_1X_2 + X_1X_2),$$

$a = (0, 1)$, $c = (1, 1) \in \mathbb{F}_2^2$. Then the system of equations for the key $(x_1, x_2) \in \mathbb{F}_2^2$ is

$$\begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 + x_1 \\ 1 + 0 + x_1x_2 \end{pmatrix}.$$

The obvious solution is $x_1 = 1$, $x_2 = 0$.

Example 2, linear maps: If F is a *linear* map, then the system of equations has an efficient solution by the methods of linear algebra (n linear equations in l unknowns). For this method to work F needs to be linear only in x.

Example 3, substitution: The complexity (or simplicity) of a polynomial equation is not always clear at first sight. Here is an example (over $\mathbb{F}_2$):

$$x_1x_2x_3 + x_1x_2 + x_1x_3 + x_2x_3 + x_2 + x_3 = 0.$$

The substitutions $x_i = u_i + 1$ transform it to

$$u_1u_2u_3 + u_1 = 0$$

(for an easy proof look in the reverse direction). The solutions are

$$u_1 = 0,\ u_2, u_3 \text{ arbitrary } \textit{or } u_1 = u_2 = u_3 = 1.$$

Thus the complete solution of the original equation is

$$x_1 = 1,\ x_2, x_3 \text{ arbitrary } \textit{or } x_1 = x_2 = x_3 = 0.$$

The Complexity of the Algebraic Attack

In the examples the solutions were easily found. But in general this task is too complex.

There are two powerful general approaches for solving systems of (polynomial) equations over $\mathbb{F}_2$:

- SAT solvers. SAT denotes the satisfiability problem of propositional logic. Consider a logical expression in Boolean variables $x_1, \ldots, x_n$ and ask if there exist values of the variables that make the expression "True". In other words consider a Boolean function f and ask if it assumes the value 1. A SAT solver is an algorithm that takes a logical expression and decides the satisfiability by finding a solution x, or showing there's no solution. The naive algorithm uses the truth table and exhausts the 2^n possible arguments. However there are much faster algorithms, the most popular being the DPLL algorithm (after Davis, Putnam, Logemann, and Loveland) and BDD based algorithms (Binary Decision Diagram).
- Elimination using Groebner bases.

Both methods work well for a small number of unknowns. With a growing number of unknowns their complexity becomes unmanageable. Of course we always find a solution by searching through the complete value table. But this naive method is inefficient (exponential in the number of unknowns, hopeless for 80 or more unknowns). But also the costs of SAT solvers and Groebner-basis methods grow exponentially with the number of unknowns. Not even the fact that all partial degrees are ≤ 1 is of vital help. The basic resault is:

Theorem 2 (Garey/Johnson) *The problem of finding a common zero of a system of polynoms $f_1, \ldots, f_r \in \mathbb{F}_2[T_1, \ldots T_n]$ is* **NP***-complete.*

Proof. See [1]. ◇

What "**NP**-complete" means will be answered later in this lecture (see Part III, Chapter 6). In fact SAT was the first problem in history shown to be **NP**-complete.

Interpretation

A common interpretation of this theorem is: For an appropriately chosen block cipher $F\colon \mathbb{F}_2^n \times \mathbb{F}_2^l \longrightarrow \mathbb{F}_2^n$ the attack with known plaintext (against the key $k \in \mathbb{F}_2^l$) is not efficient. However from a strict mathematical point of view the theorem *doesn't prove anything* of practical relevance:

1. It relates to an algorithm for *arbitrary* polynomial equations (over $\mathbb{F}_2$). It doesn't contain any assertion for special classes of polynomials, or for a concrete system of equations.

2. It gives a pure proof of (non-) existence, and provides no hint as how to construct a concrete example of a "difficult" system of equations. Note that we know that some concrete systems admit easy solutions.

3. Even if we could find concrete examples of "difficult" systems the theorem would not make any assertion whether only some rare instances (the "worst cases") are difficult, or almost all (the "generic cases")—and this is what the cryptologist wants to know. Maybe there is an algorithm that solves polynomial systems for almost all tuples of unknowns in an efficient way, and only fails for a few exceptional tuples.

Despite these critical comments the theorem raises hope that there are "secure" bitblock ciphers, and the designers of bitblock ciphers follow the

Rule of thumb *Systems of linear equations for bits admit very efficient solutions. Systems of nonlinear equations for bits in almost all cases admit no efficient solution.*

A recent article on the difficulty of systems of polynom equations is

- D. Castro, M. Giusti, J. Heintz, G. Matera, L. M. Pardo: The hardness of polynomial equation solving. Found. Comput. Math. 3 (2003), 347–420.

Interpolation Attack

A variant of algebraic cryptanalysis with known plaintext is the interpolation attack, developed in

- Thomas Jakobsen, Lars R. Knudsen: The interpolation attack on block ciphers, FSE 1997.

The idea is simple: Equip the vector space $\mathbb{F}_2^n$ with a suitable multiplication and interpret it as the finite field $K = \mathbb{F}_{2^n}$ of characteristic 2. An encryption function with a fixed key $k \in \mathbb{F}_2^l$ then is a function $F_k : K \longrightarrow K$, hence a polynomial in one indeterminate over K. Let d be its degree. Then using

interpolation this polynomial is determined by $d+1$ known plaintext blocks. The same is true for the inverse function. Using this the attacker can encrypt and decrypt without knowing the key explicitly.

The cipher designer who wants to prevent this attack should take care that encryption and decryption functions for every fixed key have large degrees as polynomials over K. This is realistic since polynomials over K may have (effective) degrees up to $2^n - 1$.

But beware that this attack may also work for some polynomials of high degree, for example for "sparse" polynomials having only a few coefficients $\neq 0$.

Linearisation of Overdetermined Systems of Equations

Systems of equations of higher order are sometimes solvable, if they are overdetermined, consisting of much more equations than unknowns. Then one simply treats some monomials as additional independent unknowns. Let's illustrate this by a simple example of three equations with two unknowns x and y:

$$\begin{aligned} x^3 + xy + y^5 &= 1, \\ 2x^3 - xy &= 0, \\ xy + 3y^5 &= 3. \end{aligned}$$

We substitute all occuring monomials: $u := x^3$, $v := xy$, $w := y^5$ and get the linear system

$$\begin{aligned} u + v + w &= 1 \\ 2u - v &= 0 \\ v + 3w &= 3 \end{aligned}$$

consisting of three equations involving three unknowns. The solution (in this case even manually derived) is $u = 0$, $v = 0$, $w = 1$. It is unique over a field K of characteristic $\neq 7$. From here we get the complete solution of the original system: $x = 0$, $y = 1$ or any 5th root of unity of K.

This attack gained some popularity in 2002 when there was a rumor that the new AES be vulnerable under this attack. However this rumor didn't survive a closer examination. As it turned out there were much too many dependencies between the linear equations.