Introducing a New Variant of Fast Algebraic Attacks and Minimizing Their Successive Data Complexity

Frederik Armknecht[1] and Gwénolé Ars[2]

[1] Theoretische Informatik, Universität Mannheim, 68131 Mannheim, Germany
armknecht@th.informatik.uni-mannheim.de
[2] IRMAR, University of Rennes, Campus de Beaulieu 35042 Rennes, France
gwenole.ars@math.univ-rennes1.fr

Abstract. Algebraic attacks have established themselves as a powerful method for the cryptanalysis of LFSR-based keystream generators (e.g., E_0 used in Bluetooth). The attack is based on solving an overdetermined system of low-degree equations $R_t = 0$, where R_t is an expression in the state of the LFSRs at clock t and one or several successive keystream bits $z_t, \ldots, z_{t+\delta}$.

In fast algebraic attacks, new equations of a lower degree are constructed in a precomputation step. This is done by computing appropriate linear combinations of T successive initial equations $R_t = 0$. The successive data complexity of the attack is the number T of successive equations.

We propose a new variant of fast algebraic attacks where the same approach is employed to eliminate some unknowns, making a divide-and-conquer attack possible. In some cases, our variant is applicable whereas the first one is not.

Both variants can have a high successive data complexity (e.g., $T \geq 8.822.188$ for E_0). We describe how to keep it to a minimum and introduce suitable efficient algorithms for the precomputation step.

Keywords: fast algebraic attacks, stream ciphers, linear feedback shift registers, Bluetooth

1 Introduction

Keystream generators are designed for online encryption of secret plaintext bitstreams M passing an insecure channel. Depending on a secret key K, they produce a regularly clocked bitstream called the keystream $Z = (z_1, z_2, \ldots)$, $z_i \in \mathbb{F}_2$. M is encrypted by XORing both streams termwise. A legal receiver decrypts by applying the same procedure.

Many keystream generators consist of combining several linear feedback shift registers (LFSRs) and possibly some additional memory. One example is the E_0 keystream generator which is part of the Bluetooth standard [6]. An LFSR is a finite automaton which produces a bitstream of arbitrary length depending on its

E. Dawson and S. Vaudenay (Eds.): Mycrypt 2005, LNCS 3715, pp. 16–32, 2005.

initial state. LFSRs are very efficient in hardware and can be designed such that the produced bitstream has maximum period and good statistical properties. Many different approaches to the cryptanalysis of LFSR-based stream ciphers were discussed in literature (e.g., time-memory-tradeoff [5], fast correlation attacks [17] or BDD-based attacks [13]). For some keystream generators, algebraic attacks outmatched all previously known attacks [8,2,7]. They consist of finding and solving a system of (low-degree) equations in the key bits and the known keystream bits. More precisely, the equations are of the form $R_t = 0$ where R_t is an expression in the state of the LFSRs at clock t and one or several successive keystream bits $z_t, \ldots, z_{t+\delta}$.

If the system is overdetermined, it can be solved by linearization: each occurring monomial is replaced by a new variable, giving a system of linear equations. If all equations are of degree $\leq d$ and $n := |K|$ denotes the key size, then the number of monomials is $\approx \binom{n}{d}$. Thus, the time effort is $\approx \binom{n}{d}^3$ (using Gaussian elimination), the amount of space is $\approx \binom{n}{d}^2$ and the data complexity is $\geq \binom{n}{d}$.

The idea of fast algebraic attack is to find linear combinations $\bigoplus_{i=0}^{T} \lambda_i R_{t+i}$ of the initial equations to obtain a new equation of a lower degree $e < d$ in a precomputation step. As this reduces the number of monomials, the time for computing the solution drops. In this paper, we focus on this precomputation step. We propose a new variant of this approach. The difference is that the number of unknowns in reduced instead of the degree. We present an example where our attack is faster than all other algebraic attacks proposed so far.

Both variants work only if the attacker knows the value of the keystream bits involved in $R_t, \ldots, R_{t+T}$. We introduce the term *successive data complexity* for $T = T(R)$. We present a theory which allows to specify the exact minimum successive data complexity for both attacks. The amount of data is only slightly decreased indeed, but it might help to make fast algebraic attacks more practically. In particular, we give efficient algorithms to achieve the miminum data complexity in precomputation step.

The paper is organized as follows: In Section 2, we provide some definitions and basic results needed for the rest of the paper, and we describe fast algebraic attacks in Section 3. We introduce a variant of these attacks in Section 4. In Section 5, theory and methods are developed for efficient precomputation steps having the minimum successive data complexity. Finally, we give a short conclusion in Section 6.

2 Definitions and Basics Results

In this Section, we provide some definitions and facts, used in the paper.

For an integer $\alpha = \sum_i \alpha_i \cdot 2^i$, $\alpha_i \in \{0, 1\}$, we define its weight $wt(\alpha) := \sum_i \alpha_i$. For a vector $(\alpha^{(1)}, \ldots, \alpha^{(m)})$ of integers, we extend this definition to $wt(\alpha^{(1)}, \ldots, \alpha^{(n)}) = \sum wt(\alpha^{(i)})$.

For positive integers $n_1, \ldots, n_m$, let $X_i := (x_{i,1}, \ldots, x_{i,n_m})$. For $\alpha_i = \sum_{j=0}^{n_i-1} \alpha_{i,j} 2^j$, we define $X_i^{\alpha_i} := \prod_{j=1}^{n_i} x_{i,j}^{\alpha_{i,j}}$ and for $\alpha = (\alpha_1, \ldots, \alpha_m)$ the expression