Improved preprocessing for the Crossbred algorithm and application to the MQ problem

Damien Vidal*†[1], Claire Delaplace[1], and Sorina Ionica[2]

[1]Modélisation, Information et Systèmes - UR UPJV 4290 – Université de Picardie Jules Verne, Université de Picardie Jules Verne : UR4290 – France
[2]Laboratoire de Mathématiques de Versailles – Université de Versailles Saint-Quentin-en-Yvelines, Université Paris-Saclay, Centre National de la Recherche Scientifique – France

Résumé

Given a polynomial system of m polynomials and n variables over a finite field Fp, finding its solutions is a NP-complete problem. Commonly used methods to solve these systems are algorithms computing Gröbner basis (F4, F5) or based on linear algebra (XL). In this work, we focus on the M Q (Multivariate Quadratic) problem, which means that we consider polynomials of degree 2. In particular, we are interested in the case where the polynomial system is defined over F2. In this case, exhaustive search becomes a viable way to solve a polynomial system (FES). Another approach consists in specifying some of the variable and try solving the resulting systems via algebraic approach. In particular, this is the idea behind the Crossbred algorithm (JV17).

Crossbred is one of the most efficient algorithm in practice, with implementations breaking records on the Fukuoka MQ challenge1. Previous work on this algorithm suggests there is room for improvement on its running time. In a first step, called pre-processing, the algorithm generates polynomials with certain properties. These polynomials are added to the initial system, which is eventually solved after specialisation of certain variables. However, it was shown that

*Intervenant
†Auteur correspondant: damien.vidal@u-picardie.fr

the number of polynomials generated most of the case is far greater that the mininimal number of necessary polynomials. With that in mind, we propose an analysis on the minimal number of polynomials needed and, as a consequence, an improvement of the pre-processing of the Crossbred algorithm. Moreover, we propose our own complexity of the pre-processing as we noted a common mistake in previous works in the complexity of the algorithm.

I will also present our complexity for the pre-proccesing and we use this result to analyse the security of MQOM.