J. Math. Crypt. **3** (2009), 177–197

DOI 10.1515 / JMC.2009.009

Hybrid approach for solving multivariate systems over finite fields

Luk Bettale, Jean-Charles Faugère and Ludovic Perret

Communicated by Jaime Gutierrez

Abstract. In this paper, we present an improved approach to solve multivariate systems over finite fields. Our approach is a tradeoff between exhaustive search and Gröbner bases techniques. We give theoretical evidences that our method brings a significant improvement in a very large context and we clearly define its limitations. The efficiency depends on the choice of the tradeoff. Our analysis gives an explicit way to choose the best tradeoff as well as an approximation. From our analysis, we present a new general algorithm to solve multivariate polynomial systems. Our theoretical results are experimentally supported by successful cryptanalysis of several multivariate schemes (TRMS, UOV, ...). As a proof of concept, we were able to break the proposed parameters assumed to be secure until now. Parameters that resists to our method are also explicitly given. Our work permits to refine the parameters to be chosen for multivariate schemes.

Keywords. Gröbner bases, multivariate cryptography.

AMS classification. 13F20, 68W30, 94A60.

1 Overview

Multivariate Cryptography comprises all the cryptographic schemes that use multivariate polynomials. The use of polynomial systems in cryptography dates back to the mid eighties [25]. The most interesting one way function used in this context is the evaluation of multivariate polynomials. Namely, let $\mathbf{f} = (f_1(x_1, \dots, x_n), \dots, f_m(x_1, \dots, x_n)) \in \mathbb{K}[x_1, \dots, x_n]^m$, the one-way function is as follows:

$$F\colon \mathbf{x} = (x_1, \dots, x_n) \in \mathbb{K}^n \longmapsto (f_1(x_1, \dots, x_n), \dots, f_m(x_1, \dots, x_n)) .$$

Here, the computational hard problem is:

Polynomial System Solving (PoSSo)
Input: polynomials $f_1(x_1, \dots, x_n), \dots, f_m(x_1, \dots, x_n)$ in $\mathbb{K}[x_1, \dots, x_n]$.
Question: find a common zero $\mathbf{z} \in \mathbb{K}^n$ of the polynomials $f_1, \dots, f_m$.

It is well known that this problem is NP-HARD. Note that PoSSo remains NP-HARD even if we suppose that the input polynomials are quadratics. In this case, PoSSo is also called $\mathcal{MQ}$.

Luk Bettale is partially supported by DGA (french secretary of defense).

To introduce a trapdoor, we start from a carefully chosen algebraic system $\mathbf{g} = g_1(x_1, \ldots, x_n), \ldots, g_m(x_1, \ldots, x_n)$ which is easy to solve. In order to hide the specific structure of **g**, we choose two linear transformations – represented by invertible matrices – $(S, U) \in GL_n(\mathbb{K}) \times GL_m(\mathbb{K})$, and set

$$\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x})) = (g_1(\mathbf{x}S), \ldots, g_m(\mathbf{x}S))\, U = \mathbf{g}(\mathbf{x}S)U,$$

with $\mathbf{x} = (x_1, \ldots, x_n)$.

The public key of such systems will be the polynomials of $\mathbf{f}$ and the secret-key is the two matrices (S, U).

To encrypt, we evaluate a message $\mathbf{m} \in \mathbb{K}^n$ on $\mathbf{f}$, i.e. $\mathbf{c} = (f_1(\mathbf{m}), \ldots, f_m(\mathbf{m}))$. To recover the correct plaintext, the legitimate recipient uses the bijectivity of the linear transformations combined with the particular structure of the polynomials of $\mathbf{g}$. Namely, he computes $\mathbf{m}' \in \mathbb{K}^n$ such that $\mathbf{g}(\mathbf{m}') = \mathbf{c}U^{-1}$. This can be done efficiently due to the particular choice of $\mathbf{g}$. Finally, he recovers the message by computing $\mathbf{m} = \mathbf{m}'S^{-1}$. Note that this family of cryptosystems can also be used in signature. To verify a signature $\mathbf{s} \in \mathbb{K}^n$ of a message $\mathbf{m} \in \mathbb{K}^m$, we check whether the equality $\mathbf{f}(\mathbf{s}) = \mathbf{m}$ holds. To generate the signature of a message $\mathbf{m} \in \mathbb{K}^m$, we apply the decryption process to $\mathbf{m}$.

There are plenty of proposals [26, 28, 27, 23, 35] based on this principle which differ only in the way of constructing the polynomials of **g**. Such schemes are attractive because they offer the possibility to have short asymmetric signatures and require little RAM to be computed on a smart card. For instance, a European project (NESSIE[1]) has advised in 2003 to use such a scheme (namely, SFLASH [28]) in the smart-card context. Unfortunately, Dubois, Fouque, Shamir and Stern [15] discovered a sever flaw in the design of SFLASH, leading to an efficient cryptanalysis of this scheme. However, this area is still appealing since we have a great deals of schemes. For instance QUARTZ [27] allows to get 128-bit long signatures and has a public key of 71 kB. It is worth to mention that in [8] the authors claim that multivariate signature schemes can outperform some ECC implementations in terms of efficiency for comparable sizes, especially when the field $\mathbb{K}$ is big.

The arrival of these multivariate schemes in the cryptographic landscape motivates further study of the complexity of solving algebraic systems. The goal is to take advantage of the context (“big” finite field, quadratic equations) to have an improved semi-automatic method to solve this kind of systems. We present in this paper a hybrid approach which can improve the way of solving zero-dimensional multivariate systems over “big” finite fields with at least 2^2 elements. This approach uses Gröbner bases techniques and exhaustive search. For the parameters usually used in cryptography, our analysis shows that the hybrid approach brings a significant improvement over the classical methods. As proof of concept, we will present efficient attacks against various multivariate schemes. In this case, the security can be reduced to the difficulty of solving a “random” multivariate system with fairly low degree ($\ll$ size of the field). We present an algorithm to solve efficiently such systems. In [9], Braeken, Wolf

[1] https://www.cosic.esat.kuleuven.be/nessie/

and Preneel did not succeed to attack UOV with Gröbner bases. Indeed, the parameters were unreachable using a standard zero-dim solving approach. They applied the zero-dim solving strategy directly without taking advantage of the specificities of the context. With the hybrid approach, we were able to break these parameters. Using this algorithm we can show that the parameters for general multivariate schemes proposed in [8] are not secure. Our theoretic analysis allows to refine the security parameters. We give, at the end of the paper, suitable parameters to make multivariate schemes resistant to our approach (complexity of the attack above 2^{80}).

A similar idea has been proposed in [12] for the XL algorithm, the so-called FXL algorithm. The authors have remarked that guessing at few variables decreases the complexity of solving the system. In [36] the authors studied the asymptotic complexity of FXL. In [2], the authors showed that XL is a special case of Gröbner bases algorithms and that XL is less efficient than F_5. Still their results are not accurate for our approach. This is why we give an asymptotic analysis for our Hybrid approach. Moreover, the authors do not give a method (algorithm) to use efficiently FXL. This is done for the hybrid approach in this paper.

1.1 Organization of the paper

This paper is organized as follows. After this introduction, we recall (Section 2) the general strategy for solving zero-dimensional polynomial systems and the algebraic tools used (Gröbner bases). We give also some theoretical definitions and results necessary to understand our approach, namely semi-regular sequences, degree of regularity and the link between these two notions. The reader familiar with these notions may skip Section 2. In Section 3 we present our hybrid approach that mixes Gröbner bases computations with exhaustive search. We give an analysis of its complexity and the way to choose the best tradeoff. We discuss also the limitations of our approach. Finally, we present in Section 4 some experimental results obtained by analyzing several multivariate schemes. The first one is a signature scheme called TRMS. It is the first scheme cryptanalysed with our hybrid approach [5]. The study of TRMS encouraged us to try the approach to some other schemes. We present our results on UOV which is considered today one of the most resistant multivariate schemes. We were able to break some proposed parameters [20]. Finally, we applied our approach on multivariate hash functions [6]. We conclude in Section 5 by giving the parameters that we consider secure against our approach.

2 Polynomial system solving

In this section, we recall all the necessary material to understand our approach. We present the mathematical object used to solve polynomial systems, namely Gröbner basis [10, 1, 13], and briefly survey the algorithms to compute this object. We also present the notion of semi-regular sequences which will be useful to measure the efficiency of our approach. All this material has already been introduced (for example in [4]). This section may be skipped if the reader is familiar with theses notions.

2.1 Zero-dim solving strategy

The general problem of polynomial system solving is to find (if any) $(z_1, \dots, z_n) \in \mathbb{K}^n$ such that

$$\begin{cases} f_1(z_1, \dots, z_n) = 0 \\ \vdots \\ f_m(z_1, \dots, z_n) = 0 \end{cases}$$

with $f_i \in \mathbb{K}[x_1, \dots, x_n]$.

In our case we are only interested in the solutions whose components are lying in the coefficient field $\mathbb{K}$. To solve a polynomial system, the best known general method is to compute the Gröbner basis of the ideal generated by this system. We refer the reader to [10, 1, 13] for a more thorough introduction to ideals and Gröbner bases. We give here only the definition of a Gröbner basis as well as the property that interests us for solving a polynomial system.

Definition 2.1. A set $G \subset \mathbb{K}[x_1, \dots, x_n]$ is a Gröbner basis w.r.t. a monomial ordering $\prec$ of a polynomial ideal $\mathcal{I}$ if

$$\forall f \in \mathcal{I},\ \exists g \in G \text{ such that } \mathrm{LM}_{\prec}(g) \text{ divide } \mathrm{LM}_{\prec}(f),$$

where $\mathrm{LM}_{\prec}$ stands for leading monomial w.r.t. $\prec$.

As we can observe, the definition depends on the monomial ordering. This ordering has also a direct impact on the structure of a Gröbner basis. For instance, a Gröbner basis for a lexicographical order (Lex) of a zero-dimensional system (i.e. with a finite number of solutions) has the following shape:

$$\{g_1(x_1), \dots, g_2(x_1, x_2), \dots, g_{k_1}(x_1, x_2), g_{k_1+1}(x_1, x_2, x_3), \dots, g_{k_n}(x_1, \dots, x_n)\}.$$

With such structure, solutions can be easily computed by successively eliminating variables, namely computing solutions of univariate polynomials and back-substituting the results.

The historical method for computing Gröbner bases was introduced by Buchberger in [10, 11]. Many improvements have been done leading to more efficient algorithms such as F_4 and F_5 due to Faugère [16, 17]. The algorithm F_4 for example is the default algorithm for computing Gröbner bases in the computer algebra softwares MAGMA and MAPLE. The F_5 algorithm[2] is even more efficient. We have mainly used this algorithm in our experiments. For our purpose, it is not necessary to describe the algorithm, but we give its complexity.

Proposition 2.2. *The complexity of computing a Gröbner basis of a zero-dimensional system of m equations in n variables with* F_5 *is*

$$\mathcal{O}\Big(\Big(m \cdot \tbinom{n+d_{\mathrm{reg}}-1}{d_{\mathrm{reg}}}\Big)^{\omega}\Big)$$

where d_{reg} is the degree of regularity of the system and $2 \le \omega \le 3$ is the linear algebra constant.

[2] available through FGb

From a practical point of view, it is much faster to compute a Gröbner basis for a degree ordering such as the Degree Reverse Lexicographic (DRL) order than for a Lexicographic order (Lex). For zero-dimensional systems, it is usually less costly to first compute a DRL-Gröbner basis, and then to compute the Lex-Gröbner basis using a change ordering algorithm such as FGLM [18]. This strategy called zero-dim solving is performed blindly in modern computer algebra softwares (for instance in MAGMA, MAPLE). This is convenient for the user, but can be an issue for advanced users.

Proposition 2.3. *Given a Gröbner basis $G_1 \subset \mathbb{K}[x_1, \ldots, x_n]$ w.r.t. a monomial ordering $\prec_1$ of a zero-dimensional system, the complexity of computing a Gröbner basis $G_2 \subset \mathbb{K}[x_1, \ldots, x_n]$ w.r.t. a monomial ordering $\prec_2$ with* FGLM *is*

$$\mathcal{O}\left(n \cdot D^3\right),$$

where D is the degree of the ideal generated by G_1 (i.e. the number of solutions counted with multiplicity in the algebraic closure of $\mathbb{K}$).

We see easily that the cost of change ordering is negligible when the system has very few solutions.

For a finite field $\mathbb{K}$ with q elements, one can always add the field equations $x_1^q - x_1, \ldots, x_n^q - x_n$ to explicitly look for solutions over the ground field $\mathbb{K}$ and not in some extensions. By doing this, we will always obtain an over-defined system. This technique is widely used, and improves the computation of solutions if $q \ll n$. Otherwise, the addition of the field equations does not lead to a faster computation of a Gröbner basis. Even worse, this can slow down the computation due to the high degrees of the equations. In multivariate cryptography, some schemes use for example the field $\mathbb{F}_{2^8}$ whose elements can easily be represented with a byte. The hybrid method that we will present is especially suitable in such situation.

2.2 Semi-regular sequences

In order to study random systems, we need to formalize the definition of "random systems". To do so, the notion of regular sequences and semi-regular sequences (for over-defined systems) has been introduced in [3]. We give the definition here.

Definition 2.4. Let $\{p_1, \ldots, p_m\} \subset \mathbb{K}[x_1, \ldots, x_n]$ be homogeneous polynomials of degrees $d_1, \ldots, d_m$ respectively. This sequence is semi-regular if

- $\langle p_1, \ldots, p_m \rangle \neq \mathbb{K}[x_1, \ldots, x_n]$,
- for all $1 \leq i \leq m$ and $g \in \mathbb{K}[x_1, \ldots, x_n]$:

$$\deg(g \cdot p_i) < d_{\text{reg}} \text{ and } g \cdot p_i \in \langle p_1, \ldots, p_{i-1} \rangle \Rightarrow g \in \langle p_1, \ldots, p_{i-1} \rangle.$$

This notion can be extended to affine polynomials by considering their homogeneous components of highest degree. It has been proven in [3, 4] that for semi-regular sequences, the degree of regularity can be computed explicitly.

Property 2.5. The degree of regularity of a semi-regular sequence $p_1, \dots, p_m$ of respective degrees $d_1, \dots, d_m$ is given by the index of the first non-positive coefficient of

$$\sum_{k \geq 0} c_k \cdot z^k = \frac{\prod_{i=1}^{m}(1 - z^{d_i})}{(1-z)^n}.$$

This property allows us to have a very precise knowledge of the complexity of the computation of a Gröbner basis for semi-regular systems. For semi-regular systems it has been proven that the degree decreases as m goes larger. Thus, the more a system is over-defined, the faster its Gröbner basis can be computed.

3 Hybrid approach

We present in this section our hybrid approach mixing exhaustive search and Gröbner bases techniques. First we will discuss the complexity of this approach. Its efficiency depends on the choice of a proper tradeoff. We take advantage of the behavior of semi-regular systems to find the best tradeoff.

3.1 General case

When we want to solve a system which has coefficients over a finite field, we can always find all the solutions in the ground field by exhaustive search. The complete search should take $\mathcal{O}(\#\mathbb{K}^n)$ operations if n is the number of variables. The idea of the hybrid approach is to mix exhaustive search with Gröbner bases computations. Instead of computing one single Gröbner basis of the whole system, we compute the Gröbner bases of $\#\mathbb{K}^k$ subsystems obtained by fixing k variables.

The intuition is that the gain obtained by working on systems with less variables may overcome the loss due to the exhaustive search on the fixed variables. The problem is to choose the best tradeoff. That is to choose properly the value of k making the complexity of our hybrid approach minimal.

Proposition 3.1. *Let $\mathbb{K}$ be a finite field and $\{f_1, \dots, f_m\} \subset \mathbb{K}[x_1, \dots, x_n]$ be a polynomial system of equations of degree d. Let $d_{\text{reg}}(k)$ be the maximum degree of regularity of all the systems:*

$$\big\{\{f_1(x_1, \dots, x_{n-k}, v_1, \dots, v_k), \dots, f_m(x_1, \dots, x_{n-k}, v_1, \dots, v_k)\} : (v_1, \dots, v_k) \in \mathbb{K}^k\big\}.$$

If the system is zero-dimensional (which implies $m \geq n$), the complexity of the hybrid approach is bounded from above by

$$\mathcal{O}\left(\min_{0 \leq k \leq n}\left((\#\mathbb{K})^k \cdot \left(\left(m \cdot \binom{n-k+d_{\text{reg}}(k)-1}{d_{\text{reg}}(k)}\right)^{\omega} + (n-k)(d^{(n-k)})^{\omega}\right)\right)\right),$$

where $2 \leq \omega \leq 3$.

Proof. This bound can be easily derived from the complexity of F$_5$ plus FGLM. In the worst case, we can bound the number of solutions (counted with multiplicity) by $d^{(n-k)}$

in the algebraic closure of $\mathbb{K}$. We multiply it by the cost of the exhaustive search on the k fixed variables. Then, we find the best tradeoff by taking the minimum value. We can notice that the degree of regularity as well as the number of solutions will depend on the actual system. This will change according to the value of k, and possibly the chosen fixed variables $(v_1, \ldots, v_k)$. □

We can now write a simple algorithm describing the general hybrid approach.

Algorithm 3.2. GenHybridSolving

```
Require: $\mathbb{K}$ is finite, $\{f_1, \ldots, f_m\} \subset \mathbb{K}[x_1, \ldots, x_n]$ is zero-dimensional, $k \in \mathbb{N}$.
Ensure: $\mathcal{S} = \{(z_1, \ldots, z_n) \in \mathbb{K}^n : f_i(z_1, \ldots, z_n) = 0, 1 \leq i \leq m\}$.
  $\mathcal{S} := \emptyset$
  for all $(v_1, \ldots, v_k) \in \mathbb{K}^k$ do
    Find the set of solutions $\mathcal{S}' \subset \mathbb{K}^{(n-k)}$ of
    $f_1(x_1, \ldots, x_{n-k}, v_1, \ldots, v_k) = 0, \ldots, f_m(x_1, \ldots, x_{n-k}, v_1, \ldots, v_k) = 0$
    using the zero-dim solving strategy.
    $\mathcal{S} := \mathcal{S} \cup \{(z'_1, \ldots, z'_{n-k}, v_1, \ldots, v_k) : (z'_1, \ldots, z'_{n-k}) \in \mathcal{S}'\}$.
  end for
  return $\mathcal{S}$.
```

The overall complexity strongly depends on the degree of regularity of the systems generated by fixing the variables and the number of solutions. In the general case, we can not even say if the hybrid approach is relevant. The problem is that we can not predict the degree of regularity of a system, except if it is semi-regular.

3.2 Semi-regular systems

For semi-regular systems, the degree of regularity depends on the number of variables, equations, and their degrees. We know for example that when there is only one variable less than the number of equations, the degree of regularity will be divided by 2 [29] instead of the generic bound $n(d-1)+1$ for a square system [24, 21]. Moreover the number of solutions of an over-defined system will be generally 1 even in the algebraic closure. The cost of the change ordering algorithm can be neglected.

Let $\mathrm{d}_{\mathrm{reg}}(n, m, d)$ be the degree of regularity of a semi-regular system with m equations of degree d in n variables. As seen in Section 2.2, it corresponds to the index i of the first non-positive coefficient c_i of the series $\sum_{i \geq 0} c_i \cdot z^i = \frac{(1-z^d)^m}{(1-z)^n}$. As we have to deal with sub-systems of m equations with $n-k$ variables, we will make the following assumption.

Hypothesis 3.3. Let $\mathbb{K}$ be a finite field and $\{f_1, \ldots, f_m\} \subset \mathbb{K}[x_1, \ldots, x_n]$ a generic semi-regular system of equations of degree d. We will suppose that the systems

$$\left\{\{f_1(x_1, \ldots, x_{n-k}, v_1, \ldots, v_k), \ldots, f_m(x_1, \ldots, x_{n-k}, v_1, \ldots, v_k)\} : (v_1, \ldots, v_k) \in \mathbb{K}^k\right\}$$

are semi-regular for all $0 \leq k \leq n$.

This hypothesis is consistent with the intuition that when some variables of a random system are fixed, the system is still random. This hypothesis has been verified with a rather large amount of random systems as well as systems coming from the applications of Section 4. In practice, the constructed systems may even be easier to solve than a semi-regular system. We have observed that its degree of regularity is always lower than a random system. Thus, our hypothesis can be used as it provides an upper bound on the complexity of our approach.

We can now state the complexity of the hybrid approach for semi-regular systems.

Proposition 3.4. *Let $\mathbb{K}$ be a finite field and $\{f_1, \ldots, f_m\} \subset \mathbb{K}[x_1, \ldots, x_n]$ be a semi-regular system of equations of degree d. The complexity of solving the system with an hybrid approach, is bounded from above by*

$$\mathcal{O}\left(\min_{0 \leq k \leq n} \left((\#\mathbb{K})^k \cdot \left(m \cdot \binom{n-k-1+\mathrm{d}_{\mathrm{reg}}(n-k,m,d)}{\mathrm{d}_{\mathrm{reg}}(n-k,m,d)} \right)^{\omega} \right) \right),$$

where $2 \leq \omega \leq 3$.

Proposition 3.4 is always true but it does not give any clue on the value of k. We give in the next subsection a way to compute the best tradeoff.

3.3 Finding the best tradeoff

In the case of semi-regular systems, it is possible to know from a theoretical point of view how many variables we will have to fix to solve a given system, i.e. to choose the best tradeoff. For example, in the case of 20 quadratic equations in 20 variables over $\mathbb{F}_{2^8}$, it appears that the best tradeoff – theoretically – is obtained by fixing $k = 2$ variables (with $\omega = 3$). We have to compute 2^{16} Gröbner bases of systems with 20 equations and 18 variables to recover all the solutions of the initial system. We will study the best tradeoff for quadratic equations, which is the most important case in multivariate cryptography.

For a square quadratic system, an approximation of the degree of regularity has been given in [3] for a system with αn equations ($\alpha > 1$) in n variables:

$$d_{\mathrm{reg}} \sim \left(\alpha - \frac{1}{2} - \sqrt{\alpha(\alpha - 1)} \right) n + \mathcal{O}(n^{1/3})$$

when $n \to \infty$.

In our case, if we fix k of n variables of a square system, assuming $k > 0$, we will have

$$d_{\mathrm{reg}} \sim \frac{n+k}{2} - \sqrt{nk} + \mathcal{O}((n-k)^{1/3})$$

when $n \to \infty$.

Using the Stirling approximation $n! \sim \sqrt{2\pi n} \cdot (n/e)^n$, one can estimate the complexity of the hybrid approach C_{hyb}.

$$C_{\mathrm{hyb}} = q^k \left(\frac{n}{\sqrt{2\pi}} \right)^{\omega} \left(\frac{\left(\frac{3n-k}{2} - 1 - \sqrt{nk} \right)^{(3n-k-1)/2 - \sqrt{nk}}}{(n-k-1)^{(n-k-1/2)} \left(\frac{n+k}{2} - \sqrt{nk} \right)^{(n+k+1)/2 - \sqrt{nk}}} \right)^{\omega}.$$

For small values of k,

$$\log(C_{\text{hyb}}) \sim k\log(q) + \omega(0.955n + 0.5\log(n) + 0.144k - 1.099\sqrt{nk} - 0.919).$$

The value k of the best tradeoff corresponds to the value for which the logarithmic derivative of C_{hyb} is equal to 0. Thus, we have to solve:

$$\begin{aligned}\log(q) + \omega\Big(\log(n-k-1) + \frac{1}{2(n-k-1)}\Big) \\ -\frac{\omega}{2}(1+\sqrt{n/k})\left(\log\left(\frac{3n-k}{2} - 1 - \sqrt{nk}\right) + \frac{1}{2\left(\frac{3n-k}{2} - 1 - \sqrt{nk}\right)}\right) \\ -\frac{\omega}{2}(1-\sqrt{n/k})\left(\log\left(\frac{n+k}{2} - \sqrt{nk}\right) + \frac{1}{2\left(\frac{n+k}{2} - \sqrt{nk}\right)}\right) = 0.\end{aligned}$$

By specializing the parameters q, n and ω, a good approximation of the value k can be found. For example, when $n = 20$, $q = 2^8$, $\omega = 2$, we find $k = 0.859$. The theoretical value is $k = 1$. For these parameters, the complexity of a direct zero-dim solving is 2^{83} and it goes down to 2^{67} with the hybrid approach. This approximation is not very precise when $n - k$ is small, but it gives a good indication on the appropriate tradeoff.

3.4 Borderline case

The hybrid approach is a compromise between exhaustive search and Gröbner bases computation. If the size of the coefficient field is too big, the hybrid approach will not bring any improvements. We have seen in Section 2.2 that when one variable is fixed the degree of regularity goes from $n+1$ down to $\frac{n+1}{2}$ for a square system. If there is no gain in fixing one variable, we can be sure that the hybrid approach will not bring any gain for any amount of fixed variables. In this subsection, we are interested in the case when the hybrid approach is not efficient. We will assume that we want to solve a zero-dimensional semi-regular square system $\{f_1, \ldots, f_n\} \subset \mathbb{K}[x_1, \ldots, x_n]$, where $\mathbb{K}$ is a finite field.

The complexity of the hybrid approach with $k = 1$ will be

$$\mathcal{O}\left(q\left(n \cdot \binom{3(n-1)/2}{n-2}\right)^{\omega}\right).$$

We can compare this complexity with the complexity of F$_5$ ($k = 0$). We recall this complexity for quadratic equations:

$$\mathcal{O}\left(\left(n \cdot \binom{2n}{n-1}\right)^{\omega}\right).$$

The hybrid approach will bring a gain if

$$q \le \left(\frac{\binom{2n}{n-1}}{\binom{3(n-1)/2}{n-2}}\right)^{\omega}.$$

By using the Stirling approximation, asymptotically we find that

$$\log_2(q) \leq 0.6226 \cdot \omega \cdot n + \mathcal{O}(\log_2(n))$$

when $n \to \infty$.

For example, if $\omega = 2$, the complexity of computing a Gröbner basis of a semi-regular quadratic system with 20 variables and 20 equations is above 2^{80}, but with the hybrid approach, it will always be less if the field has a size below 2^{24}.

3.5 Summary

We have the necessary material to compute the theoretical tradeoff to be chosen. We give an algorithm for finding the exact theoretical value of k.

Algorithm 3.5. FindTradeoff

```
Require: $q$ the size of the field, $n$ the number of equations and variables of the system,
  $d$ the degree of the equations.
Ensure: $k$ the best theoretical tradeoff for hybrid solving.
  $\mathcal{A} := [\,]$.
  for $0 \leq k \leq n$ do
    Find the index $d_k$ of the first non-positive coefficient of the series $\frac{(1-z^d)^n}{(1-z)^{n-k}}$.
    Compute $\mathcal{A}[k] := q^k \left(n\binom{n-k-1+d_k}{d_k}\right)^\omega$.
  end for
  return $k$ such that $\mathcal{A}[k]$ is minimum.
```

In practice, even with the best tradeoff, the computation can fail because of memory issues. In some cases, it can be interesting to fix few more variables to save memory.

For the case of quadratic equations, we know from the previous subsection that if the field is too big ($\log_2(q) > 0.6226 \cdot \omega \cdot m$), fixing variables will not lead to any improvements. We can avoid searching for a tradeoff.

Algorithm 3.6. HybridSolving

```
Require: $\mathbb{K}$ is finite, $\{f_1, \ldots, f_n\} \subset \mathbb{K}[x_1, \ldots, x_n]$ is zero-dimensional, $\deg(f_i) = 2$,
  for $1 \leq i \leq n$.
Ensure: $\mathcal{S} = \{(z_1, \ldots, z_n) \in \mathbb{K}^n : f_i(z_1, \ldots, z_n) = 0, 1 \leq i \leq m\}$.
  if $\log_2(q) > 0.6226 \cdot \omega \cdot n$ then
    Use the zero-dim solving strategy to compute $\mathcal{S}$.
  else
    $k :=$ FindTradeoff$(q, n, 2)$.
    $\mathcal{S} :=$ GenHybridSolving$(\{f_1, \ldots, f_n\}, k)$.
  end if
  return $\mathcal{S}$.
```

One can replace the Algorithm 3.5 by our approximation to find the tradeoff. To have a better accuracy, we recommend to compute the tradeoff explicitly.

The overall complexity of the hybrid approach is still exponential. In terms of complexity class, it does not overpass the direct Gröbner bases techniques. Still, we have shown that the hybrid approach can decrease the complexity of solving a polynomial system, and this slight difference can bring a problem from computationally impossible to possible in practice. With our algorithm, we have a semi-automatic method to solve the systems with the best tradeoff between zero-dim solving and exhaustive search. As proof of concept we present in the next section some applications of our approach, in particular on UOV.

4 Applications

We have presented in the previous section an approach that permits in theory to improve the complexity of the resolution. We present in this section our experimental results using this approach on various multivariate schemes which illustrate the relevancy of our approach.

4.1 TRMS

TRMS (for Tractable Rational Map Signature) is a signature scheme proposed in [30]. The scheme is based on a Tractable Rational Map (TRM) which is a special construction of a mapping which can be efficiently inverted. The specific details are not important for our approach. We refer the reader to [30] for further information. We have started to study the hybrid approach with this scheme. This is why we present our results on TRMS.

Some multivariate cryptosystems like HFE have been broken by computing directly the Gröbner basis of the public system [19]. In a sense, the hidden structure has been uncovered by the Gröbner basis computation algorithm. In our case, we have an underdefined system over a rather big field. Thus, even if we were able to compute a DRL-Gröbner basis, we will not be able to compute the Lex-Gröbner basis because of the too large number of possible solutions. For example, with the given parameters of TRMS, there will be at least $\#\mathbb{K}^{(n-m)} = 2^{64}$ valid solutions. To address this problem, we can fix $n - m$ variables of the system randomly. We will still be able to find one valid solution. By fixing random variables, the system has been messed up, and has lost its internal structure. Experimentally, we noticed that the new systems are semi-regular systems. We fall right in the scope of our hybrid approach. We recall that the road-map of the attack is as follows:

(1) Fix $n - m$ variables of the system: we obtain a new system with m variables and m equations which will always have at least one valid solution.

(2) Solve the new system with the hybrid approach.

The authors gave explicitly the following parameters:

$$\mathbb{K} = \mathbb{F}_{2^8}, \quad n = 28, \quad m = 20.$$

The public system has only up to degree 2 polynomials. For these parameters, we give the theoretic complexity of the hybrid approach in Figure 1 and the memory in Figure 2.

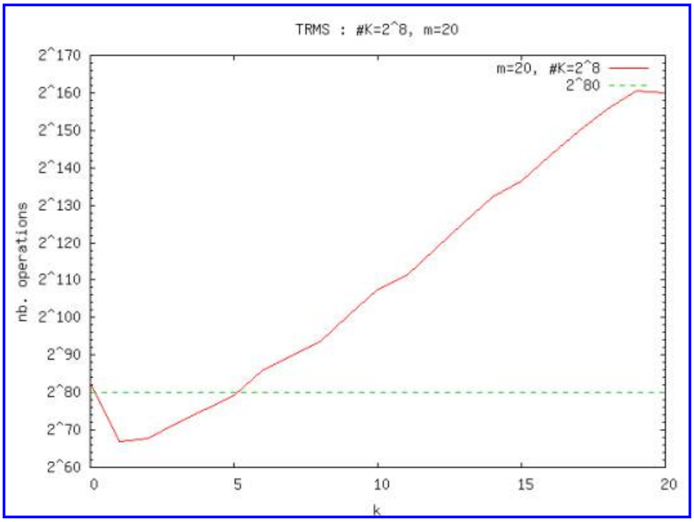

Figure 1. TRMS: Complexity of hybrid approach depending on k

According to Figure 1, fixing only one variable should take the least time in theory (about 2^{67} basic operations). The complexity is below the usual cryptographic security bound (2^{80}) also plotted in Figure 1. As Gröbner bases algorithms use a high amount of memory, it is also interesting to plot the minimum theoretic memory usage during the computations (Figure 2).

We will now present our experimental results. We have mounted this attack on a bi-pro Xeon 2.4 GHz with 64 GB of RAM. We give in Table 1 the experimental results for different tradeoffs. $\mathrm{T}_{\mathrm{F}_5}$, $\mathrm{Mem}_{\mathrm{F}_5}$ and $\mathrm{Nop}_{\mathrm{F}_5}$ are the time, memory and number of basic operations needed to compute one Gröbner basis with F_5. Nop is the total number of basic operations with the hybrid approach. It is obtained by computing $\#\mathbb{K}^k \cdot \mathrm{Nop}_{\mathrm{F}_5}$.

m	$m-k$	$\#\mathbb{K}^k$	$\mathrm{T}_{\mathrm{F}_5}$	$\mathrm{Mem}_{\mathrm{F}_5}$	$\mathrm{Nop}_{\mathrm{F}_5}$	Nop
20	18	2^{16}	51 h	41.940 GB	2^{41}	2^{57}
20	17	2^{24}	2 h 45 min	4.402 GB	2^{37}	2^{61}
20	16	2^{32}	626 s	912 MB	2^{34}	2^{66}
20	15	2^{40}	46 s	368 MB	2^{30}	2^{70}

Table 1. Experimental results on TRMS

When fixing only one variable, the computation was not feasible because of the too large amount of memory needed (more than 100 GB according to Figure 2). To

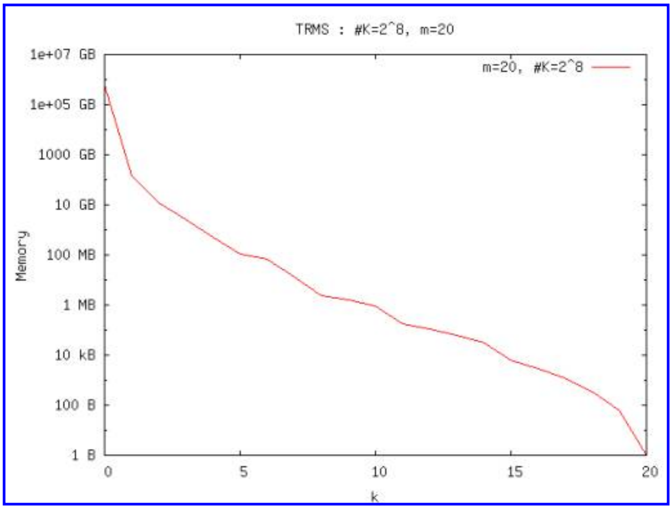

Figure 2. TRMS: Memory of hybrid approach depending on k

make the computation feasible, the best tradeoff in practice was to fix 2 variables. The total complexity is still acceptable (also from a theoretical point of view). It has to be noted that a signature forgery here would only take 51 hours assuming an access to $2^{16} = 65536$ processors (which is very reasonable).

4.2 UOV

UOV is a multivariate signature scheme proposed in [22]. It shares the same basic as TRMS, namely it uses a polynomial map easy to invert hidden with linear transformations. The set of variables $\{x_1, \ldots, x_n\}$ is partitioned in two sets $V = \{x_1, \ldots, x_{n-m}\}$ (vinegar variables) and $O = \{x_{n-m+1}, \ldots, x_n\}$ (oil variables). The quadratic part $f_k^{(2)}$ of each secret polynomials $f_1, \ldots, f_m$ has the special shape

$$f_k^{(2)} = \sum_{(x_i,x_j)\in V\times V} \alpha_{i,j}^{(k)} x_i x_j + \sum_{(x_i,x_j)\in V\times O} \beta_{i,j}^{(k)} x_i x_j .$$

For more details, the reader may refer to the initial paper [22]. It is easy to see that once the vinegar variables have been fixed, the system becomes a linear system with m equations in m variables, and will be easy to invert with a high probability. Again, the public system becomes an under-defined system of m quadratic equations in n variables. The oil and the vinegar variables are completely mixed and without the knowledge of the linear transformation, it should be impossible to solve the system.

The problem of forging a valid signature is equivalent to solving the under-defined quadratic public system. We can tackle the problem by using the same technique as for TRMS, namely first fixing $n - m$ variables to have a square system, and then solve it with the hybrid approach [20]. The authors recommended the following parameters:

$$\mathbb{K} = \mathbb{F}_{2^4}, \quad m = 16, \quad n = 32 \text{ (or 48)}.$$

It seems that the best theoretic tradeoff would be to fix 4 variables as it can be seen in Figure 3. But in practice, we obtain the best experimental tradeoff by fixing only 2 variables. We show these results in Table 2. We were able to forge signatures on a bi-pro Xeon 2.4 GHz with 64 GB of RAM. The labels used in Table 2 have the same meaning as in Table 1.

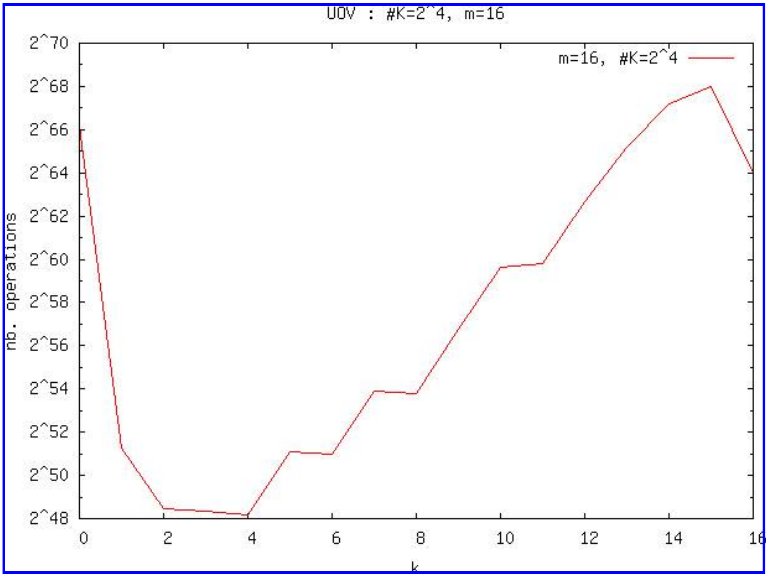

Figure 3. Theoretical complexity for UOV depending on k

m	$m-k$	$\#\mathbb{K}^k$	$\mathrm{T}_{\mathrm{F}_5}$	$\mathrm{Mem}_{\mathrm{F}_5}$	$\mathrm{Nop}_{\mathrm{F}_5}$	Nop
16	15	2^4	≈ 1 h	3.532 GB	$2^{36.9}$	$2^{40.9}$
16	14	2^8	126 s	270 MB	$2^{32.3}$	$2^{40.5}$
16	13	2^{12}	9.41 s	38 MB	$2^{28.7}$	$2^{40.7}$

Table 2. Experimental results on UOV

It is worth to remark that the efficiency of our attack does not rely on the number of the vinegar variables, but only on the number of equations. Even if there are many more variables than equations, one can always fix variables and still be able to forge one signature. Thus the results are valid for $n = 32$ as well as for $n = 48$.

4.3 Multivariate hash functions

The problem of solving a polynomial system can lead to the construction of hash functions. For example, in [14, 7], the authors proposed to build a hash function with an iterative Merkle–Damgård structure whose compression function is explicitly described by a multivariate polynomial system. It is known that the security of such hash functions strongly relies on the security of their compression function. We studied the robustness of this function with the given parameters for the dense and the sparse construction.

Let $\mathbb{K}$ be a finite field, $n, m \in \mathbb{N}$ and $\mathrm{F}\colon \mathbb{K}^{n+m} \longrightarrow \mathbb{K}^m$,

$$\mathrm{F}\colon (y_1,\ldots,y_m,x_1,\ldots,x_n) \longmapsto (f_1(y_1,\ldots,y_m,x_1,\ldots,x_n),\ldots,f_m(y_1,\ldots,y_m,x_1,\ldots,x_n)).$$

The evaluation of the compression function F in a chaining value $(y_1,\ldots,y_m)$ and a message block $(x_1,\ldots,x_n)$ give the next chaining value, and the last one will be the hash. From now on, we will only focus on the compression function F and will regardless call it the hash function.

Three kinds of generic attacks can be done on hash functions:

- Preimage attack: given a digest h, find a message x such that $F(x) = h$.
- Second preimage attack: given a message m, find a message m' such that $F(x) = F(x')$.
- Collision attack: find any pair (x, x') such that $F(x) = F(x')$.

The generic complexity of the preimage attacks is 2^n evaluations of the function where n is the length (bit-size) of the digest. The collision attack needs only $\mathcal{O}(2^{n/2})$ operations with the birthday paradox. A hash function is considered secure when there are no attacks with a better complexity.

With multivariate hash functions, given a digest $h = (z_1,\ldots,z_m)$, the preimage and second preimage attacks are equivalent to solving the following system:

$$\begin{cases} f_1(y_1,\ldots,y_m,x_1,\ldots,x_n) - z_1 & = & 0 \\ \vdots & & \\ f_m(y_1,\ldots,y_m,x_1,\ldots,x_n) - z_m & = & 0 \end{cases}$$

A collision attack is equivalent to find for a fixed non-zero difference $\delta = (\delta_1,\ldots,\delta_n)$ two messages such that $H(m) = H(m+\delta)$, in other words, solve the following system:

$$\begin{cases} f_1(y_1,\ldots,y_m,x_1,\ldots,x_n) - f_1(y_1,\ldots,y_m,x_1+\delta_1,\ldots,x_n+\delta_n) & = & 0 \\ \vdots & & \\ f_m(y_1,\ldots,y_m,x_1,\ldots,x_n) - f_m(y_1,\ldots,y_m,x_1+\delta_1,\ldots,x_n+\delta_n) & = & 0 \end{cases}$$

In fact, we compute a differential of the initial system at the point δ, thus the total degree of each polynomial will decrease by one.

4.3.1 Dense construction

In [14], the authors have proposed a construction using dense random cubic polynomials over an extension of $\mathbb{F}_2$. We are again in the scope of our hybrid approach. Trying to find preimages by solving directly the cubic system seems not to be possible for the given parameters. We focused on a collision attack which is less ambitious as we only have to deal with quadratic polynomials. Usually, for dedicated hash functions like MD5 or SHA1, finding a suitable difference δ is difficult. Many work has been done in this area, mainly by Wang [31, 32, 33, 34]. In our case, as the polynomials are randomly generated, any difference could lead to a collision with a good probability. The hard part is finally to solve the system generated with this difference. As we have seen previously, the system to solve will have its degree decreased by 1, in our case, we will only have polynomials of degree 2. We summarize the attack:

(1) Randomly choose a non-zero difference $(\delta_1, \ldots, \delta_n)$.

(2) Fix the values of $(y_1, \ldots, y_m)$ to the initial values $(v_1, \ldots, v_m)$ and build the system $f' = (f'_1, \ldots, f'_m)$ with

$$\begin{aligned} f'_1 &= f_1(v_1, \ldots, v_m, x_1, \ldots, x_n) - f_1(v_1, \ldots, v_m, x_1 + \delta_1, \ldots, x_n + \delta_n), \\ &\vdots \\ f'_m &= f_m(v_1, \ldots, v_m, x_1, \ldots, x_n) - f_m(v_1, \ldots, v_m, x_1 + \delta_1, \ldots, x_n + \delta_n). \end{aligned}$$

(3) Solve the system $f' = 0$.

(4) If we find a solution, then we have a collision, else come back to step 1.

The authors proposed several sets of parameters where $m = n$:

$$\begin{aligned} &\mathbb{K} = \mathbb{F}_{2^4},\ m = 40,\ n = 40, && \mathbb{K} = \mathbb{F}_{2^4},\ m = 64,\ n = 64, \\ &\mathbb{K} = \mathbb{F}_{2^8},\ m = 20,\ n = 20, && \mathbb{K} = \mathbb{F}_{2^8},\ m = 32,\ n = 32, \\ &\mathbb{K} = \mathbb{F}_{2^{16}},\ m = 16,\ n = 16. \end{aligned}$$

We have then to solve square systems, but we would emphasize that if the systems were under-defined, then we could fix more variables as in the previous applications.

We were able to break two of the proposed set of parameters:

- $\mathbb{K} = \mathbb{F}_{2^8},\ m = 20,$
- $\mathbb{K} = \mathbb{F}_{2^{16}},\ m = 16.$

As the parameters are the same as in our previous applications, we obtained about the same time and complexity. We also had to choose the same tradeoff. The only difference is that we needed more memory to actually solve the systems due to the fact that in the hash functions, the polynomials are more dense.

4.3.2 Sparse construction

The authors have also proposed a construction using sparse random polynomials. From a practical point of view, we have observed that the behavior of the systems is very different from semi-regular systems. It is not possible to predict their degree of regularity anymore. Still, with the given parameters, we were able to effectively forge signatures with our approach by adjusting it with a special strategy. The systems are now generated using a difference δ with low Hamming weight. The systems are even more sparse, and experimentally, very sparse systems are easier to solve. On the other hand, we decrease the probability of finding a collision. We have to determine an optimal Hamming weight for each set of parameters making the Gröbner bases computation possible (from a practical point of view) and leading with a reasonable probability to a collision. In our experiments, we have used the parameters given in Table 3. We give in the columns min (resp. max) the minimum (resp. maximum) time needed to compute a Gröbner basis of the systems obtained for different values of δ. The value ϵ is the ratio of non-zero monomials in each polynomial (density).

m	$\#\mathbb{K}$	ϵ	weight of δ	min	max	prob
20	2^8	0.2%	4	0.5 s	1289.5 s	$1/4$
16	2^{16}	0.2%	5	0.1 s	78.5 s	$1/3$
32	2^8	0.1%	2	0.5 s	690.3 s	$1/15$

Table 3. Experimental results on sparse multivariate hash functions

With the sparse construction, we did not need to fix any variables in order to make the computation possible, but as it can be seen in Table 3, the time for solving the systems are not the same depending on the choice of the difference δ. The time can vary between less than one second to 20 minutes. We see that we were able in this case to break one more set of parameters. Even if we can not state any theoretic conclusion on sparse systems, it happens that with the given parameters, sparse systems seem to be insecure.

4.4 Other constructions

The use of our approach is straightforward for any multivariate scheme. For underdefined systems ($m < n$ equations, n variables), we have to solve in fact a “generic” square system (m equations, m variables). In [8], the authors have proposed a comparison between their implementation of several multivariate signature schemes and implementations of ECC. They show that good implementations of multivariate schemes are much better than the best implementations of ECC in terms of time-area product. Unfortunately, most of the parameters proposed can be broken with our hybrid approach. We present in Table 4 the schemes and the parameters they have implemented as well as the theoretic complexity for forging a signature with our approach.

	m	#$\mathbb{K}$	expected security	direct Gröbner basis	fixed variables k	hybrid approach	mem.
UOV $(n =$ 30$)$	10	2^8	2^{80}	$2^{41.36}$	1	$2^{37.75}$	2 MB
UOV $(n =$ 60$)$ enTTS $(n =$ 28$)$	20	2^8	2^{160}	$2^{82.51}$	1	$2^{66.73}$	139 GB
					2	$2^{67.79}$	12 GB
Rainbow $(n =$ 48$)$ amTTS $(n =$ 34$)$	24	2^8	2^{192}	$2^{98.80}$	1	$2^{78.09}$	10 TB
					2	$2^{79.06}$	816 GB

Table 4. Analysis of several multivariate schemes

For each set of parameters, the cost of computing the Gröbner basis of the system (with F_5) is less than the expected security, but still above the 2^{80} (except for the short version of UOV). With the hybrid approach, by guessing only one variable, the cost of forging a signature goes below 2^{80} for each of the given parameters. Another variable can be fixed to decrease the memory needed, the complexities are still below 2^{80}. In practice, we were able to break systems with up to 20 variables by choosing $k = 2$. The schemes Rainbow and amTTS are still out of reach. In Table 4, our attack does not take into account that the systems are under-defined, some improvements could be done to further reduce the complexity of the hybrid attack.

5 Conclusion

We have presented in this paper a general method to solve polynomial systems over finite fields. We have computed explicitly the complexity of this approach, and we have given some applications where our approach allowed us to break some security challenges for concrete parameters of several multivariate schemes claimed to be secure. The method takes advantage of the predictable behavior of generic systems. All in all, from our contribution, we can explicitly give the parameters for which a random quadratic system can not be solved with our approach (i.e. complexity $> 2^{80}$) in Table 5. The column m is the minimum number of equations and variables that should be chosen. The column k is the best tradeoff for the hybrid approach and the column T is the corresponding complexity. To have a sketch of what could be the size of the public key and the signatures, we compute them for a system with $3/2$ times more variables than equations $(n = 3m/2)$. It is roughly the case of amTTS and TRMS. We emphasize that the complexities given in Table 5 are upper bounds which are reached for random dense systems. Note that the complexities have been computed with $\omega = 2$. This value matches our experiments. This can be explained because of the sparsity of the matrices computed in F_5. It has to be added that the values given in Table 5 are the minimal parameters that a $\mathcal{MQ}$-cryptosystem like UOV should have to resist our attack. The given parameters do not prevent from other kind of attacks.

$\#\mathbb{K}$	m	k	T	signature length	public key size
2^{32}	20	0	2^{82}	960 bits	39 kB
2^{16}	23	1	2^{81}	560 bits	29 kB
2^{8}	26	1	2^{83}	312 bits	21 kB
2^{4}	30	7	2^{83}	180 bits	16 kB
2^{2}	41	23	2^{82}	124 bits	20 kB

Table 5. Minimal recommended parameters

Finally, the best tradeoff between security/size of the public key/size of the signature is obtained by choosing a large amount of variables and a small field. We note that the key is smaller when $\#\mathbb{K} = 2^4$. This could be the more suitable field to build multivariate schemes. In view of these results, the question "$\mathcal{MQ}$-Cryptosystems as Replacement for Elliptic Curves?" [8] should be reevaluated.

References

[1] William W. Adams and Philippe Loustaunau, *An Introduction to Gröbner Bases*, Graduate Studies in Mahematics 3. AMS, 1994.

[2] Gwénolé Ars, Jean-Charles Faugère, Hideki Imai, Mitsuru Kawazoe, and Makoto Sugita, *Comparison between XL and Gröbner basis algorithms*. ASIACRYPT 2004 (Pil Joong Lee, ed.), LNCS 3329, pp. 338–353. Springer, December 2004.

[3] Magali Bardet, *Étude des systèmes algébriques surdéterminés. Applications aux codes correcteurs et à la cryptographie*, Ph.D. thesis, Université de Paris VI, 2004.

[4] Magali Bardet, Jean-Charles Faugère, and Bruno Salvy, *On the complexity of Gröbner basis computation of semi-regular overdetermined algebraic equations*. Proc. International Conference on Polynomial System Solving (ICPSS), pp. 71–75, 2004.

[5] Luk Bettale, Jean-Charles Faugère, and Ludovic Perret, *Cryptanalysis of the TRMS Signature Scheme of PKC'05*. Progress in Cryptology – AFRICACRYPT 2008, Lecture Notes in Computer Science 5023, pp. 143–155. Springer, 2008.

[6] ______, *Security analysis of multivariate polynomials for hashing*. Information Security and Cryptology – INSCRYPT 2008, Lecture Notes in Computer Science. Springer, 2008, to appear.

[7] Olivier Billet, Matthew J. B. Robshaw, and Thomas Peyrin, *On building hash functions from multivariate quadratic equations*. ACISP, LNCS 4586, pp. 82–95. Springer, 2007.

[8] Andrey Bogdanov, Thomas Eisenbarth, Andy Rupp, and Christopher Wolf, *Time-Area Optimized Public-Key Engines: $\mathcal{MQ}$-Cryptosystems as Replacement for Elliptic Curves?*. CHES '08: Proceedings of the 10th international workshop on Cryptographic Hardware and Embedded Systems, pp. 45–61. Springer-Verlag, 2008.

[9] An Braeken, Christopher Wolf, and Bart Preneel, *A Study of the Security of Unbalanced Oil and Vinegar Signature Schemes*. Topics in Cryptology – CT-RSA 2005, LNCS 3376, pp. 29–43. Springer, February 2005.

[10] Bruno Buchberger, *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal*, Ph.D. thesis, University of Innsbruck, 1965.

[11] Bruno Buchberger, Georges E. Collins, Rudiger G. K. Loos, and Rudolph Albrecht, *Computer algebra symbolic and algebraic computation*, SIGSAM Bull. 16 (1982), pp. 5–5.

[12] Nicolas Courtois, Alexander Klimov, Jacques Patarin, and Adi Shamir, *Efficient algorithms for solving overdefined systems of multivariate polynomial equations*. Advances in Cryptology – EUROCRYPT 2000, LNCS 1807, pp. 392–407. Springer, 2000.

[13] David A. Cox, John B. Little, and Don O'Shea, *Ideals, Varieties and Algorithms*. Springer, 2005.

[14] Jintai Ding and Bo-Yin Yang, *Multivariates Polynomials for Hashing*, Cryptology ePrint Archive, Report 2007/137, 2007.

[15] V Dubois, Pierre-Alain Fouque, Adi Shamir, and Jacques Stern, *Practical Cryptanalysis of SFLASH*. Advances in Cryptology – CRYPTO'07, 4622, pp. 1–12. Springer, 2007.

[16] Jean-Charles Faugère, *A new efficient algorithm for computing Gröbner bases (F4)*, Journal of Pure and Applied Algebra 139 (1999), pp. 61–88.

[17] ———, *A new efficient algorithm for computing Gröbner bases without reduction to zero (F5)*. Proceedings of the 2002 International Symposium on Symbolic and Algebraic Computation ISSAC (T. Mora, ed.), pp. 75–83. ACM Press, July 2002, ISBN: 1-58113-484-3.

[18] Jean-Charles Faugère, Patrizia M. Gianni, Daniel Lazard, and Teo Mora, *Efficient Computation of Zero-Dimensional Gröbner Bases by Change of Ordering*, Journal of Symbolic Computation 16 (1993), pp. 329–344.

[19] Jean-Charles Faugère and Antoine Joux, *Algebraic cryptanalysis of Hidden Field Equation (HFE) cryptosystems using Gröbner bases*. Advances in Cryptology – CRYPTO 2003 (Dan Boneh, ed.), LNCS 2729, pp. 44–60. Springer, 2003.

[20] Jean-Charles Faugère and Ludovic Perret, *On the security of UOV*. SCC 08, 2008.

[21] Marc Giusti, *Some Effectivity Problems in Polynomial Ideal Theory*. EUROSAM, Computation, pp. 159–171, 1984.

[22] Aviad Kipnis, Jacques Patarin, and Louis Goubin, *Unbalanced Oil and Vinegar Signature*. Advances in Cryptology – EUROCRYPT 1999, Lecture Notes in Computer Science 1592, pp. 206–222. Springer-Verlag, 1999.

[23] N. Koblitz, *Algebraic Aspects of Cryptography.*, Algorithms and Computation in Mathematics 3. Springer-Verlag, 1998.

[24] Daniel Lazard, *Gröbner-Bases, Gaussian elimination and resolution of systems of algebraic equations*. EUROCAL, pp. 146–156, 1983.

[25] Tsutomu Matsumoto and Hideki Imai, *Public Quadratic Polynomial-tuples for Efficient Signature-Verification and Message-Encryption*. Advances in Cryptology – EUROCRYPT 1988, LNCS 330, pp. 419–453. Springer-Verlag, 1988.

[26] Jacques Patarin, *The Oil and Vinegar Signature Scheme*, presented at the Dagstuhl Workshop on Cryptography, 1997.

[27] Jacques Patarin, Nicolas Courtois, and Louis Goubin, *QUARTZ, 128-Bit Long Digital Signatures*. CT-RSA'01, 2020, pp. 282–297. Springer, 2001.

[28] Jacques Patarin, Louis Goubin, and Nicolas Courtois, *$C^* -+$ and HM: Variations Around Two Schemes of T. Matsumoto and H. Imai*. Advances in Cryptology – Asiacrypt'98, 1514, pp. 35–49. Springer, 1998.

[29] Agnes Szanto, *Multivariate subresultants using Jouanolou's resultant matrices*, accepted to Journal of Pure and Applied Algebra (2001).

[30] Lih-Chung Wang, Feipei Lai Yuh-Hua Hu, Chun-Yen Chou, and Bo-Yin Yang, *Tractable Rational Map Signature*. PKC 05, 2005.

[31] Xiaoyun Wang, Dengguo Feng, Xuejia Lai, and Hongbo Yu, *Collisions for Hash Functions MD4, MD5, HAVAL-128 and RIPEMD*, Cryptology ePrint Archive, Report 2004/199, 2004.

[32] Xiaoyun Wang, Xuejia Lai, Dengguo Feng, Hui Chen, and Xiuyuan Yu, *Cryptanalysis of the Hash Functions MD4 and RIPEMD*. EUROCRYPT (Ronald Cramer, ed.), LNCS 3494, pp. 1–18. Springer, 2005.

[33] Xiaoyun Wang, Yiqun Lisa Yin, and Hongbo Yu, *Finding Collisions in the Full SHA-1*. CRYPTO (Victor Shoup, ed.), LNCS 3621, pp. 17–36. Springer, 2005.

[34] Xiaoyun Wang, Hongbo Yu, and Yiqun Lisa Yin, *Efficient Collision Search Attacks on SHA-0*. CRYPTO (Victor Shoup, ed.), LNCS 3621, pp. 1–16. Springer, 2005.

[35] C. Wolf and B. Preneel, *Taxonomy of Public Key Schemes based on the problem of Multivariate Quadratic equations*, Cryptology ePrint Archive, Report 2005/077, 2005, `http://eprint.iacr.org/`.

[36] Bo-Yin Yang, Jiun-Ming Chen, and Nicolas Courtois, *On Asymptotic Security Estimates in XL and Gröbner Bases-Related Algebraic Cryptanalysis*. ICICS 2004, pp. 401–413, 2004.

Received 30 December, 2008; revised 12 October, 2009

Author information

Luk Bettale, INRIA, Centre Paris-Rocquencourt, SALSA Project
UPMC, Univ Paris 06, LIP6
CNRS, UMR 7606, LIP6
104, avenue du Président Kennedy
75016 Paris, France.
Email: `luk.bettale@lip6.fr`

Jean-Charles Faugère, INRIA, Centre Paris-Rocquencourt, SALSA Project
UPMC, Univ Paris 06, LIP6
CNRS, UMR 7606, LIP6
104, avenue du Président Kennedy
75016 Paris, France.
Email: `jean-charles.faugere@inria.fr`

Ludovic Perret, INRIA, Centre Paris-Rocquencourt, SALSA Project
UPMC, Univ Paris 06, LIP6
CNRS, UMR 7606, LIP6
104, avenue du Président Kennedy
75016 Paris, France.
Email: `ludovic.perret@lip6.fr`