Security Analysis of Multivariate Polynomials for Hashing

Luk Bettale, Jean-Charles Faugère, 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
luk.bettale@lip6.fr, jean-charles.faugere@inria.fr,
ludovic.perret@lip6.fr

Abstract. In this paper, we investigate the security of a hash function based on the evaluation of multivariate polynomials [17]. The security of such hash function is related to the difficulty of solving (under-defined) systems of algebraic equations. To solve these systems, we have used a general hybrid approach [8] mixing exhaustive search and Gröbner bases solving. This shows that this approach is general and can be used in several contexts. For the sparse construction, we have refined this strategy. From a practical point of view, we have been able to break several challenges proposed by Ding and Yang [17] in real time.

1 Introduction

Multivariate Cryptography is the set of all the cryptographic primitives using multivariate polynomials. The use of algebraic systems in cryptography dates back to the mid eighties [15, 26], and was initially motivated by the need for alternatives to number theoretic-based schemes. Indeed, although quite a few problems have been proposed to construct public-key primitives, those effectively used are essentially factorization (e.g. in RSA [27]) and discrete logarithm (e.g. in Diffie-Hellman key-exchange [16]). It has to be noted that multivariate systems enjoy low computational requirements. Moreover, such schemes are not concerned with the quantum computer threat, whereas it is well known that number theoretic-based schemes like RSA, DH, or ECDH are [28].

Multivariate cryptography has become a dynamic research area, as reflected by the ever growing number of papers in the most famous cryptographic conferences. This is mainly due to the fact that an European project (NESSIE[1]) has advised in 2003 to use such a signature scheme (namely, SFLASH [13]) in the smart-card context. Unfortunately, Dubois, Fouque, Shamir and Stern [18] discovered a sever flaw in the design of SFLASH, leading to an efficient cryptanalysis of this scheme.

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

A new trend of multivariate cryptography is to design symmetric primitives. In this context, it is often possible to relate the security of the primitive to the difficulty of solving a random system of algebraic equations. A very interesting example of such construction is the stream cipher QUAD [7].

In this paper, we will study the security of a hash function proposed by Ding and Yang [17] based on the evaluation of multivariate polynomials. It has to be noted that Billet, Peyrin and Robshaw proposed in about the same time [11] a similar construction.

1.1 Previous works

A previous analysis of multivariate hash functions has been done by Aumasson and Meier [2]. Let ϵ be the ratio of monomials in each polynomial of the system describing the multivariate hash function. Aumasson and Meier pointed out that – when generating random sparse polynomials – there is a probability of $(1-\epsilon)^n$ that a given variable x_i appears in none of the n polynomials of the system. In this case, we can find trivial collisions by taking two messages with a difference only at the ith position. In addition, they proposed an interesting technique inspired from coding theory to solve random system of sparse equations. Finally, they demonstrated that families of low-degree functions over $\mathbb{F}_2$ are neither pseudo-random nor unpredictable.

In [24], Luo and Lai proposed a generic attack against multivariate hash functions slightly better than exhaustive search. They gave an explicit method to compute 2^d digests using $2^d - 1$ queries to the hash function (assuming the knowledge of the d^{th} derivative of the function, where d is the maximum degree of the polynomials describing the function).

In this paper, we will apply a general technique already presented in [8] allowing to find collisions on several challenges of the multivariate hash function [17] in real time.

1.2 Organization of the paper

After this introduction, the paper is organized as follows. In Sect. 2, we briefly introduce the principle of multivariate hash function [17] as well as the general framework of our attack. The security of such hash functions is related the difficulty of solving algebraic equations. We do not present in this paper the mathematical tools (ideals, varieties and Gröbner bases), or the algorithmic tools (F_4/F_5) for solving algebraic systems. These tools have already been defined in [1, 20, 21, 4, 5, 8, 22]. In this last section, we analyze the security of the hash function proposed by Ding and Yang [17]. We have focused our attention on the constructions based on cubic equations. By using a technique introduced in [8], we will show that we can find collisions on some of the parameters proposed in [17]. Our experiments suggest that the sparse construction is weaker than the dense construction.

2 Multivariate Hash Functions

In this part, we recall the principle of multivariate hash functions [17]. After that, we will describe the algebraic tools which will be used for mounting our attack.

2.1 Multivariate Hash Function

We shall call "*Multivariate Hash Function*" a hash function explicitly described by a set of multivariate polynomials. As usual, we will focus our attention to the compression function which will be plugged into a Merkle-Damgård construction. For a multivariate hash function, the compression function is defined by a mapping $F : (y_1, \ldots, y_m, x_1, \ldots, x_n) \in \mathbb{K}^{m+n} \rightarrow$

$$\big(f_1(y_1, \ldots, y_m, x_1, \ldots, x_n), \ldots, f_m(y_1, \ldots, y_m, x_1, \ldots, x_n)\big) \in \mathbb{K}^m,$$

where $f_1, \ldots, f_m \in \mathbb{K}[y_1, \ldots, y_m, x_1, \ldots, x_n]$ are algebraic polynomials.

Let $a_0 \in \mathbb{K}^m$ be the Initial Value (IV); the digest is computed using the following procedure :

1: Let $(v_1, \ldots, v_k) \in (\mathbb{K}^n)^k$ be a padded message
2: **for** $i = 0$ **to** $(k-1)$ **do**
3: $\quad a_{i+1} = F(a_i, v_i)$
4: **end for**
5: **return** a_k

It is well known that the security of this procedure relies on the properties of $F : \mathbb{K}^{m+n} \rightarrow \mathbb{K}^m$. To construct this map, Ding and Yang [17] proposed to use cubic polynomials and stacked (composed) quadratics. Note that the stacked composed quadratics construction was also described – in about the same time – by Billet, Peyrin and Robshaw [11]. In this paper, we will only consider the cubic construction of [17].

There are two variants of the cubic construction. First, it is suggested to use random dense cubic polynomials with the following set of parameters :

160-bit hash	256-bit hash
$\#\mathbb{K} = 2^4, n = 40, m = n$	$\#\mathbb{K} = 2^4, n = 64, m = n$
$\#\mathbb{K} = 2^8, n = 20, m = n$	$\#\mathbb{K} = 2^8, n = 32, m = n$
	$\#\mathbb{K} = 2^{16}, n = 16, m = n$

The second variant consists in considering sparse cubic polynomials. Namely, they proposed to generate cubic equations having a proportion of ϵ non-zero coefficients. This construction permits to drastically improve the efficiency of a multivariate hash function. The authors [17] claimed that the security of this construction is as secure as the dense construction. The parameters proposed are :

160-bit hash	256-bit hash
$\#\mathbb{K} = 2^4, n = 40, m = n, \epsilon = 0.1\%$	$\#\mathbb{K} = 2^4, n = 64, m = n, \epsilon = 0.1\%$
$\#\mathbb{K} = 2^8, n = 20, m = n, \epsilon = 0.2\%$	$\#\mathbb{K} = 2^8, n = 32, m = n, \epsilon = 0.1\%$
	$\#\mathbb{K} = 2^{16}, n = 16, m = n, \epsilon = 0.2\%$

2.2 Algebraic Attacks on Multivariate Hash Functions

The security of a multivariate hash function is obviously related to the difficulty of solving algebraic systems of equations. For instance, let $f_1, \ldots, f_m$ be the polynomials describing the compression function F. Let also $(z_1, \ldots, z_m) \in \mathbb{K}^m$ be a valid digest. The problem of finding preimages (resp. second preimages) is equivalent to solving :

$$f_1(a_1, \ldots, a_m, x_1, \ldots, x_n) = z_1, \ldots, f_m(a_1, \ldots, a_m, x_1, \ldots, x_n) = z_m,$$

with $(a_1, \ldots, a_m) \in \mathbb{K}^m$ be a chaining constant.

In this paper, we will consider a less ambitious attack, namely finding collisions. The goal is to find a pair of messages $(M, M') \in \mathbb{K}^n \times \mathbb{K}^n$ such that $F(M) = F(M')$. To do so, we can fix a difference $\delta \in \mathbb{K}^n$ between the two messages M and M' and try to solve the system :

$$\begin{gathered} f_1(a_1, \ldots, a_m, x_1 + \delta_1, \ldots, x_n + \delta_n) - f_1(a_1, \ldots, a_m, x_1, \ldots, x_n) = 0 \\ \vdots \\ f_m(a_1, \ldots, a_m, x_1 + \delta_1, \ldots, x_n + \delta_n) - f_m(a_1, \ldots, a_m, x_1, \ldots, x_n) = 0 \end{gathered}$$

One can remark that this is (almost) equal to the discrete differential of $F' = F(a_1, \ldots, a_m, x_1, \ldots, x_n)$ at δ. Formally, this differential is $DF'_\delta(x_1, \ldots, x_n) =$

$$F(y_1, \ldots, y_m, x_1 + \delta_1, \ldots, x_n + \delta_n) - F(y_1, \ldots, y_m, x_1, \ldots, x_n) - F(y_1, \ldots, y_m, 0).$$

The monomials of highest degree will cancel. Thus, we have to solve a multivariate polynomial system of degree $d-1$, where d=max(degree(f_i),$i \in \{1, \ldots, m\}$). This explains why you have to consider cubic polynomials. For quadratic polynomials, the problem of finding a collision is equivalent to solve a linear system of equations.

To find a collision or a preimage, we have then to solve an algebraic systems of equations. To date, Gröbner bases [9, 10] provide the most efficient algorithmic solution for this problem.

3 Security Analysis of Multivariate Hash Functions

In this part, we will analysis the actual constructions proposed by Ding and Yang [17]. The compression function $F : \mathbb{K}^{2n} \to \mathbb{K}^n$ ($m = n$) is given by a random or

sparse cubic polynomial system over a finite field $\mathbb{K} = \mathbb{F}_q$, q being a power of 2. Precisely, F is given by the polynomials :

$$(f_1(y_1, \ldots, y_n, x_1, \ldots, x_n), \ldots, f_n(y_1, \ldots, y_n, x_1, \ldots, x_n)).$$

From now on, we will assume that the variables $y_1, \ldots, y_n$ (chaining variables) are fixed.

We have considered the following attack scenario. We randomly fix a difference $(\delta_1, \ldots, \delta_n) \in \mathbb{K}^n$. Our goal is to find a message $M = (M_1, \ldots, M_n) \in \mathbb{K}^n$ such that M and $M + (\delta_1, \ldots, \delta_n)$ collide. As already explained, this is equivalent to find the solutions of the system of quadratic equations :

$$\begin{aligned} f'_1(x_1, \ldots, x_n) &= f_1(a_1, \ldots, a_n, x_1 + \delta_1, \ldots, x_n + \delta_n) - f_1(a_1, \ldots, a_n, x_1, \ldots, x_n) = 0 \\ &\vdots \\ f'_n(x_1, \ldots, x_n) &= f_n(a_1, \ldots, a_n, x_1 + \delta_1, \ldots, x_n + \delta_n) - f_n(a_1, \ldots, a_n, x_1, \ldots, x_n) = 0 \end{aligned}$$

3.1 Random System of Cubic Equations

In order to solve such systems, we will use the hybrid approach proposed in [8]. This strategy is relevant for systems over fields whose cardinality $\geq 2^4$. Instead of computing directly the variety (i.e. the set of solutions), we will specify k variables of the system. This permits to decrease the complexity of the Gröbner basis computation. On the other hand, we don't know if our guess is correct. Thus, the cost of an exhaustive search on the k variables must be added to the global cost of the attack. That is, we have to perform $(\#\mathbb{K})^k$ Gröbner bases computations (but of easier systems). This approach has been already successfully applied against TRMS [8] and UOV [22]. In our context, the systems will not have necessarily a solution. This due to the fact that no collision exists for a given difference. In such case, we simply repeat the process with a new difference. To summarize :

1. Choose a random non-zero difference $\delta = (\delta_1, \ldots, \delta_n) \in \mathbb{K}^n$
2. Generate the system $\left(f'_1(x_1, \ldots, x_n) = 0, \ldots, f'_n(x_1, \ldots, x_n) = 0\right)$ as explained previously.
3. Compute the variety V associated to this system using the hybrid approach described above.
4. If the variety V is not empty, we have found $\#V$ collisions, and we have finished. Otherwise, we repeat from step 1

For the dense construction, we have observed that a considerable number of differences lead to collisions.

To illustrate this approach, we will consider the construction using random system of cubic equations. We will present experimental results for the parameters ($\#\mathbb{K} = 2^{16}, n = 16, m = n$) and ($\#\mathbb{K} = 2^8, n = 20, m = n$). The others parameters proposed seems to be out of reach of our approach and can be considered secure.

The important observation here is that the system that we obtained behave like a semi-regular systems [3, 6, 4]. We will present experimental results supporting this claim. These results agree with the conjecture [5]:

"Almost all overdetermined polynomial system is a semi-regular system."

We can precisely estimate the degree of regularity of the systems, and then the complexity of the F_5 algorithm [21]. Let $k \geq 0$ be the numbers of variables fixed, the degree of regularity is given by the index of the first non-positive coefficient of the series [3, 6, 4] :

$$\frac{(1-z^2)^n}{(1-z)^{n-k}}.$$

In the next tables, we have quoted the degree of regularity observed in our experiments. Namely, the maximum degree reached during F_5 on systems obtained by fixing k variables on collision-finding systems. We have also quoted the theoretical degree of regularity of a semi-regular system of n equations in $n-k$ variables.

n	$n-k$	k	d_{reg} (theoretical)	d_{reg} (observed)
16	16	0	17	
16	15	1	9	9
16	14	2	7	7
16	13	3	6	6
16	12	4	5	5
16	11	5	5	5

Fig. 1. Comparaison with theoretical results ($\#\mathbb{K} = 2^{16}, n = 16, m = n$).

n	$n-k$	r	d_{reg} (theoretical)	d_{reg} (observed)
20	20	0	21	
20	18	2	9	9
20	17	3	8	8
20	16	4	7	7
20	15	5	6	6

Fig. 2. Comparaison with theoretical results ($\#\mathbb{K} = 2^{8}, n = 20, m = n$).

By fixing variables, we can obtain a significant gain on the complexity of F_5. On the other hand, as soon as $k > 0$, each specification of the r variables

n	$n-k$	k	T_{F_5}	$\mathrm{Nop}_{\mathrm{F}_5}$	N
16	15	1	≈ 1 h.	$2^{36.9}$	$2^{52.9}$
16	14	2	126 s.	$2^{32.3}$	$2^{64.3}$
16	13	3	9.41 s.	$2^{28.7}$	$2^{84.9}$

Fig. 3. Experiments for $\#\mathbb{K} = 2^{16}, n = 16, m = n$ (256-bit hash).

n	$n-k$	k	$\mathrm{T}_{\mathrm{F}_5}$	$\mathrm{Nop}_{\mathrm{F}_5}$	T
20	18	2	51h	2^{41}	2^{57}
20	17	3	2h45min.	2^{37}	2^{61}
20	16	4	643.1 sec.	2^{34}	2^{66}
20	15	5	48.7 sec.	2^{30}	2^{70}

Fig. 4. Experiments for $\#\mathbb{K} = 2^8, n = 20, m = n$ (160-bit hash).

will not necessarily lead to an algebraic system whose set of solutions is not empty. Thus, we have to perform an exhaustive search on k variables. In other words, instead of computing one Gröbner basis of a system of n equations and variables, we compute $(\#\mathbb{K})^k$ Gröbner bases of "easier" systems (n equations with $n-k$ variables). We have then to find an optimal tradeoff between the cost of F_5 and the number of Gröbner basis that we have to compute. With this technique, we were able to mount a theoretical collision attack with a complexity bounded from above by :

$$\mathcal{O}\left((\#\mathbb{K})^k \left(\mathrm{C}_{n-k+d_{\mathrm{reg}}-1}^{d_{\mathrm{reg}}}\right)^{\omega}\right),$$

with $\omega, 2 \leq \omega \leq 3$ being the linear algebra constant.
This complexity comes directly from the complexity of F_5 in the semi-regular case [4, 5].

In the next tables, we have quoted the practical results that we have obtained. T_{F_5} is the time of computing one Gröbner basis with F_5. We have also included the corresponding number of operations (field multiplications) $\mathrm{Nop}_{\mathrm{F}_5}$ performed by F_5, and the total number N of operations of our attack (i.e. the cost of computing $(\#\mathbb{K})^k$ Gröbner bases). The experimental results have been obtained using a bi-pro Xeon 2.4 Ghz with 64 Gb. of Ram.
The most interesting tradeoff is obtained with $k = 1$. In this case, we obtain a complexity of $2^{52.9}$. In this case, the maximum memory used during the Gröbner bases computations was bounded from above by 4Gb.
We observe that the optimal choice is for $k = 2$, for which you obtain a complexity bounded from above by 2^{57} to actually find a collision. We emphasize that this approach is fully parallelizable (each computation of the $(\#\mathbb{K})^k$ Gröbner basis are totally independent). For instance, assuming an access to 2^{16} processors (which is very reasonable), the computation can be done in two days. For $k = 1$, we can extrapolate that one could find a collision in 2^{53} (fields operations).

3.2 Sparse Cubic Construction

In this part, we investigate the security of the sparse construction. From a practical point of view, we have observed that the behavior of the systems is very different from the dense construction. The systems no longer behave like semi-regular systems. It is very difficult to predict the degree of regularity of such systems, and then the complexity of a Gröbner basis computation.

In this context, we used a special strategy for solving the systems. First, we have generated collision-finding systems using differences $\delta \in \mathbb{K}^n$ with a low Hamming weight. This permits to have systems even more sparse, and in general easier to solve. On the other hand, this constraint restrict our chance of finding a collision. For each set of parameters, we have to determine an optimal Hamming weight making the Gröbner bases computation possible, and leading with a reasonable probability to a collision. In our experiments, we have used the following parameters which seems well suited in practice. Note that the values have been determined experimentally.

	parameters	weight of δ
A	$\#\mathbb{K} = 2^8, n = 20, \epsilon = 0.2\%$	4
B	$\#\mathbb{K} = 2^{16}, n = 16, \epsilon = 0.2\%$	5
C	$\#\mathbb{K} = 2^8, n = 32, \epsilon = 0.1\%$	2

Fig. 5. Weight of δ w.r.t. the parameters.

Once the δ is fixed, we directly try to compute the Gröbner basis; we no longer use a hybrid strategy here. This is not necessary since the systems are sparse and most of them are easy to solve. To summarize, our strategy is :

1. Choose a non-zero difference $\delta = (\delta_1, \ldots, \delta_n) \in \mathbb{K}^n$ with low Hamming weight
2. Generate the system

$$\begin{aligned} f'_1(x_1, \ldots, x_n) &= f_1(a_1, \ldots, a_n, x_1 + \delta_1, \ldots, x_n + \delta_n) - f_1(a_1, \ldots, a_n, x_1, \ldots, x_n) = 0 \\ &\vdots \\ f'_n(x_1, \ldots, x_n) &= f_n(a_1, \ldots, a_n, x_1 + \delta_1, \ldots, x_n + \delta_n) - f_n(a_1, \ldots, a_n, x_1, \ldots, x_n) = 0 \end{aligned}$$

1. Compute the variety V associated to this system using a Gröbner basis computation.
2. If the variety V is not empty, we have found $\#V$ collisions, and we have finished. Otherwise, we repeat from step 1

In this case, we have to try several δ before finding a non empty variety, and then a collision.

The results that we have obtained are given below. We would like to emphasize that these results are not uniform at all. For the same set of parameters, the time for computing the Gröbner basis can be very different depending of the δ chosen. To illustrate this fact, we have quoted :

- $\min_0$/$\max_0$: the minimum/maximum time for computing the variety (assuming that there is no solution to the system).
- $\min_1$/$\max_1$: the minimum/maximum time for computing a non-empty variety. We have then found a collision.
- prob : a very rough estimation of the proportion of the δ (with a fixed Hamming weight) leading to a collision

The results are given below :

	parameters	$\min_0$	$\max_0$	$\min_1$	$\max_1$	prob
A	$q = 2^8, n = 20, \epsilon = 0.2\%$	0. s.	1088.9 s.	0.5 s.	1289.5 s.	$1/4$
B	$q = 2^{16}, n = 16, \epsilon = 0.2\%$	0. s.	1301.1 s.	0.1 s.	78.5 s.	$1/3$
C	$q = 2^8, n = 32, \epsilon = 0.1\%$	0. s.	7.3 s.	0.4 s.	690.3 s.	$1/15$

All in all, we can mount our attack, and find a collision, on the set of parameters A, B, and C in few minutes.

4 Conclusion

In this paper, we have investigated the security of a multivariate hash function proposed in [17]. We first studied the cubic construction. For such construction, the problem of finding collisions is equivalent to the problem of solving a system of quadratic equations. To tackle this problem, we used a general technique previously used to analyze the security of TRMS and UOV. This method has already shown its efficiency [8, 22] and our results can be used to better calibrate the parameters of future multivariate schemes. For instance, we have been able to break two challenges proposed [17]. But, as soon as $n \geq 32$, this construction can be considered as secure.

The conclusion concerning the sparse construction is different. Our experiments tend to prove that this construction as not the same level of security than the dense construction (and seems to be much weaker). For this reason, we believe that the sparse construction should be avoided.

Interestingly enough, we have observed that the behavior F_5 is different for sparse systems. To our point of view, it could be interesting to further investigate the theoretical and practical complexity of solving random sparse systems of equations with Gröbner bases.

References

1. W.W. Adams and P. Loustaunau. *An Introduction to Gröbner Bases.* Graduate Studies in Mathematics, Vol. 3, AMS, 1994.
2. J.-P. Aumasson and W. Meier. *Analysis of Multivariate Hash Functions.* Information Security and Cryptology - ICISC 2007, Lecture Notes in Computer Science, vol. 4817, pp. 309–323, 2007.
3. M. Bardet. *Etude des systèmes algébriques surdéterminés. Applications aux codes correcteurs et à la cryptographie.* Thèse de doctorat, Université de Paris VI, 2004.
4. M. Bardet, J.-C. Faugère, and B. Salvy. *On the complexity of Gröbner basis computation of semi-regular overdetermined algebraic equations.* In Proc. International Conference on Polynomial System Solving (ICPSS), pp. 71–75, 2004. Available at *http://www-calfor.lip6.fr/ICPSS/papers/43BF/43BF.htm.*
5. M. Bardet, J.-C. Faugère, and B. Salvy. *Complexity Study of Gröbner Basis Computation.* Technical report, INRIA, 2002. Available at *http://www.inria.fr/rrrt/rr-5049.html.*
6. M. Bardet, J.-C. Faugère, B. Salvy and B-Y. Yang. *Asymptotic Behaviour of the Degree of Regularity of Semi-Regular Polynomial Systems.* In Proc. of MEGA 2005, Eighth International Symposium on Effective Methods in Algebraic Geometry, 2005.
7. C. Berbain, H. Gilbert, J. Patarin. *QUAD: A Practical Stream Cipher with Provable Security.* Advances in Cryptology – EUROCRYPT 2006, Lecture Notes in Computer Science, vol. 4004, Springer–Verlag, pp. 109–128, 2006.
8. L. Bettale, J.-C. Faugère, and L. Perret. *Cryptanalysis of the TRMS Signature Scheme of PKC'05*. Progress in Cryptology – AFRICACRYPT 2008, Lecture Notes in Computer Science, vol. 5023, pp. 143–155, 2008.
9. B. Buchberger, G.-E. Collins, and R. Loos. *Computer Algebra Symbolic and Algebraic Computation.* Springer-Verlag, second edition, 1982.
10. B. Buchberger. *Gröbner Bases : an Algorithmic Method in Polynomial Ideal Theory.* Recent trends in multidimensional systems theory. Reider ed. Bose, 1985.
11. O. Billet, M. Robshaw, T. Peyrin. *On Building Hash Functions from Multivariate Quadratic Equations.* Information Security and Privacy – ACISP 2007, Lecture Notes in Computer Science, vol. 4586, pp. 82–95, 2007.
12. N. Courtois, A. Klimov, J. Patarin, and A. Shamir. *Efficient Algorithms for Solving Overdefined Systems of Multivariate Polynomial Equations.* Advances in Cryptology – EUROCRYPT 2000, Lecture Notes in Computer Science, vol. 1807, Springer–Verlag, pp. 392-407, 2000.
13. N. Courtois, L. Goubin, and J. Patarin. *SFLASH, a Fast Symmetric Signature Scheme for low-cost Smartcards – Primitive Specification and Supporting documentation.* Available at `www.minrank.org/sflash-b-v2.pdf`.
14. D. A. Cox, J.B. Little and D. O'Shea. *Ideals, Varieties, and algorithms: an Introduction to Computational Algebraic Geometry and Commutative algebra.* Undergraduate Texts in Mathematics. Springer-Verlag. New York, 1992.
15. W. Diffie, and H. J. Fell. *Analysis of a Public Key Approach Based on Polynomial Substitution.* Advances in Cryptology – CRYPTO 1985, Lecture Notes in Computer Science, vol. 218, pp. 340–349, 1986.
16. W. Diffie, and M.E. Hellman. *New Directions in Cryptography.* IEEE Transactions on Information Theory, IT–22(6), pp. 644–654, 1976.
17. J. Ding, and B.-Y. Yang. *Multivariates Polynomials for Hashing* Information Security and Cryptology (Inscrypt 2007), Lecture Notes in Computer Science, vol. 4990, pp. 358-371, 2007.

18. V. Dubois, P.-A. Fouque, A. Shamir, and J. Stern. *Practical Cryptanalysis of SFLASH.* Advances in Cryptology – CRYPTO 2007, Lecture Notes in Computer Science, vol. 4622, pp. 1-12, 2007.
19. J. C. Faugère, P. Gianni, D. Lazard, and T. Mora. *Efficient Computation of Zero-Dimensional Gröbner Bases by Change of Ordering.* Journal of Symbolic Computation, 16(4), pp. 329–344, 1993.
20. J.-C. Faugère. *A New Efficient Algorithm for Computing Gröbner Basis: F_4.* Journal of Pure and Applied Algebra, vol. 139, pp. 61–68, 1999.
21. J.-C. Faugère. *A New Efficient Algorithm for Computing Gröbner Basis without Reduction to Zero: F_5.* Proceedings of ISSAC, pp. 75–83. ACM press, July 2002.
22. J.-C. Faugère, and L. Perret. *On the Security of UOV*. In Proc. First International Conference on Symbolic Computation and Cryptography (SCC 08) , pp. 103–110, 2008.
23. M. R. Garey, and D. B. Johnson. *Computers and Intractability. A Guide to the Theory of NP-Completeness.* W. H. Freeman, 1979.
24. Y. Luo and X. Lai *Higher Order Differential Cryptanalysis of Multivariate Hash Functions* Cryptology ePrint archive, Report 2008/350, available at http://eprint.iacr.org.
25. F. S. Macaulay. *The Algebraic Theory of Modular Systems.* Cambrige University Press, Cambrige, 1916.
26. T. Matsumoto, and H. Imai. *Public Quadratic Polynomial-tuples for Efficient Signature-Verification and Message-Encryption.* Advances in Cryptology – EUROCRYPT 1988, Lecture Notes in Computer Science, vol. 330, Springer–Verlag, pp. 419–453, 1988.
27. R. Rivest, A. Shamir and L. Adleman. *A Method for Obtaining Digital Signatures and Public-Key Cryptosystems.* Communications of the ACM, 21(2), pp. 120–126, 1978.
28. P. W. Shor. *Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer*. SIAM J. Computing 26, pp. 1484-1509 (1997).
29. A. Szanto. *Multivariate subresultants using jouanolouõs resultant matrices.* Journal of Pure and Applied Algebra, to appear.