UOV-Pepper: New Public Key Short Signature in Degree 3

Gilles Macario-Rat[1] and Jacques Patarin[2]

[1] Orange, Orange Gardens, 46 avenue de la République, F-92320 Châtillon, France
gilles.macariorat@orange.com
[2] Versailles Laboratory of Mathematics, UVSQ, CNRS, University of Paris-Saclay
jpatarin@club-internet.fr

Abstract. In this paper, we present a new perturbation for the design of multivariate schemes that we call "Pepper". From this idea, we present some efficient multivariate signature schemes with explicit parameters that resist all known attacks. In particular they resist the two main (and often very powerful) attacks in this area: the Gröbner attacks (to compute a solution of the system derived from the public key) and the MinRank attacks (to recover the secret key). Pepper can also be seen as a new perturbation that can be used to strengthen many other multivariate schemes. The "Pepper" perturbation works only for public key equations of degree (at least) 3. Despite this, the size of the public key may still be reasonable since we can use larger fields (and also maybe non dense equations). Furthermore, the size of the signatures can be very short.

Keywords: public-key cryptography, post-quantum multivariate cryptography, UOV, HFE.

1 Introduction

Many schemes in Multivariate cryptography have been broken. Among the most spectacular attacks we can mention that the C* scheme of Matsumoto and Imai [13] has been broken in [14], the SFlash scheme submitted to the NESSIE competition has been broken in [4, 7, 8] , the LUOV scheme [3] submitted to the Post-Quantum NIST competition has been broken in [6], and the GeMMS schemes [5] has been broken in [16]. At present the two main general attacks in multivariate cryptography are the use of Gröbner bases in "direct attacks" (in order to find a solution of the public equations involved without finding the secret key, cf [9]), and the MinRank attacks in order to find the secret key [12, 2]. In many schemes the Gröbner attack is dangerous because the degree of regularity of the public equations is smaller than for random quadratic equations. Recently the MinRank attacks have become much more powerful than before due to the introduction of the Minor equations [2].

Despite these dangerous and powerful attacks, multivariate cryptography remains an interesting area of research. This is mainly due to three facts. First,

the schemes, if they can resist non-quantum attacks, are also expected to resist quantum computers, i.e. multivariate cryptography is one of the family of "post-quantum" cryptography (with lattices, codes, hash-based cryptography, isogenies, combinatorial schemes). Second, the MQ problem (solving a set of Multivariate Quadratic equations on finite field) is NP-hard on any finite field, and seems to be very difficult to solve when the equations are random and the number of variables is about the same as the number of equations. Third, some properties can be obtained at present only with multivariate cryptography such as ultra-short public-key signatures, or encryption with ultra-short blocs [15].

In this paper, we present a new tool that can be useful to design new multivariate schemes, or to strengthen the security of existent multivariate schemes. We call it "Pepper".

2 Pepper: main ideas

2.1 Notations

As in all classical multivariate schemes, we use a finite field $\mathbb{F}_q$ with q elements and we deal with the ring of polynomials in n variables $(x_1, \ldots, x_n)$ (or simply x) over $\mathbb{F}_q$, denoted $\mathbb{F}_q[x]$. Therefore here $\mathbb{F}_q[x]^m$ will refer to the algebra of n-ary m-dimensional polynomials, that we call (n, m)-polynomials for short. The internal product of this algebra is implicitly defined as the extension of the product defined over $\mathbb{F}_q^m$, itself defined by the classical (field) product over $\mathbb{F}_{q^m}$ and transferred by a proper isomorphism between $\mathbb{F}_q^m$ and $\mathbb{F}_{q^m}$. For instance when $n = m$, the i-th iterate Frobenius mapping: $x \to x^{q^i}$ can be expressed a (n, n)-polynomial of degree 1. We note $\deg(f)$ the degree of a polynomial f. By extension, the degree of a (n, m)-polynomial is the maximum degree of its $(n, 1)$-components.

2.2 The tweak

We start from some trapdoor scheme, $y = f(x)$, where f is a degree-d (n, m)-polynomial that we can invert somehow. Typical examples of f are HFE and UOV among many others, but for the sake of our presentation, we can stick to a generic f. Our purpose is to add a perturbation to f, in order to get a new scheme, with stronger security, and in particular dismiss all possible "low rank" issues. We will discuss this point hereafter. Classical schemes have degree 2 such as HFE or UOV; we will consider here larger degrees, mainly $d = 3$, that provide more interest to our new perturbation.

We now introduce our new "tweak": $p(x)P(x)$, where p and P are respectively random $(n, 1)$- and (n, m)-polynomials. We also require that $\deg(p)+\deg(P) = d$. The new trapdoor can be expressed as $\tilde{f}(x) = f(x) + p(x)P(x)$. The advantages of this new trapdoor are the followings.

- Its degree is still at most d.
- Whatever the degree of p, $p(x)$ can only take q values (those of $\mathbb{F}_q$).

– It gives the opportunity to design "high rank" schemes, since the polynomials p and P may be chosen adequately, and therefore this tweak thwarts the MinRank attacks.

2.3 Pepper for signature

The Pepper perturbation is the conjunction of the tweak and a special mode of inversion. In order to invert the trapdoor $y = f(x) + p(x)P(x)$, we make an exhaustive search on all the q possible values of $p(x)$. For one possible value a, we simply invert $y = f(x) + aP(x)$, and keep only the solutions satisfying $p(x) = a$. Of course, we have made here the additional assumption that $y = f(x) + aP(x)$ is indeed efficiently invertible. This assumption is highly related to the choice of the initial scheme f but it is not difficult to fulfil. For instance for HFE, it suffices to assume that the polynomial P has "HFE" shape, or for UOV, that it is "UOV-compatible", etc. This mode of operation may be suitable for signature and encryption according to the original scheme f, but it also costs a factor q compared to the original scheme.

3 Security analysis in degree 2

We first give a word about these new perturbations when $d = 2$, and show that they have not much interest in this case. We analyse them particularly in the light of the HFE and UOV schemes, but the analysis can certainly be generalized to other schemes. The most restrictive condition we have to fulfil is $\deg(p) + \deg(P) = 2$, then we can imagine:

– $\deg(p) = 0$, $\deg(P) = 2$. We skip this case, since p must be constant and therefore does not add entropy.
– $\deg(p) = 1$, $\deg(P) = 1$. In this case p and P are linear. For HFE and regarding the MinRank attack, this perturbation simply increases the rank of the secret polynomial by an amount of 1, which is not competitive, since the perturbation already costs a factor q. For UOV, this perturbation amounts to transform 1 'oil' variable into 1 'vinegar' variable, which is useless compared to the original scheme.
– $\deg(p) = 2$, $\deg(P) = 0$. Here, p is a random quadratic form, P is a random element of $\mathbb{F}_q^m$. This case has already been described in literature as the "+" perturbation, i.e. adding a small amount of random quadratic polynomials to the original scheme. The effect of this perturbation can be cancelled by considering the adequate projection, and the natural isomorphism between $\mathbb{F}_q^m$ and $\mathbb{F}_{q^m}$:

$$\begin{aligned} y &= f(x) + p(x)P \\ y^q &= f(x)^q + p(x)P^q \\ yP^q - y^qP &= f(x)P^q - f(x)^qP \end{aligned}$$

In particular, regarding the Gröbner basis computation attack, this shows, that the degree of regularity does not increase compared to the one of $y = f(x)$

4 UOV-Pepper in degree 3: example of parameters and security results

We now propose a variant of UOV using the Pepper perturbation and degree 3, for signature mode.

4.1 Description of UOV-Pepper

In what follows, n_o and n_v are respectively the number of oil and vinegar variables. We note $n = n_o + n_v$, and the different vectors of variables $x_o = (x_1, \ldots, x_{n_o})$, $x_v = (x_{1+n_o}, \ldots, x_n)$, $x = (x_1, \ldots, x_n)$.

We set $f(x) = \sum_{i=1}^{n_o} x_i Q_i(x_v) + C(x_v)$, the "classical" UOV trapdoor extended in degree 3, where Q_i are random degree-2 (n_v, n_o)-polynomials, C is a random degree-3 (n_v, n_o)-polynomial. This is the same idea as in the original scheme: the trapdoor is linear in x_o, hence it can easily be inverted if x_v is set. The UOV perturbed scheme becomes: $\tilde{f}(x) = f(x) + p(x)P(x)$, where p is a degree-2 $(n, 1)$-polynomial, P is a degree-1 (n, n_o)-polynomial compatible with the UOV structure. More precisely $P(x) = \sum_{i=1}^{n_o} \alpha_i x_i + L(x_v)$ where the α_i are random elements of $\mathbb{F}_q^{n_o}$ and L is a random degree-1 (n_v, n_o)-polynomial. One can easily check that when x_v is set and $p(x)$ is guessed, the perturbed trapdoor can be inverted as claimed. The public key is a composition $\mathcal{P} = \tilde{f} \circ T$ where T is some secret linear bijective map over $\mathbb{F}_q^n$. The secret key is a description of Q_i, C, p, P and T. See below sec. 4.3 for parameter selection and further explanations.

4.2 Signature protocol

To sign a value y, the signer aims to find a value z such as $\mathcal{P}(z) = y$, or equivalently find x such as $\tilde{f}(x) = y$. In order to invert the trapdoor, the signer chooses at random x_v, makes all the q possible "guesses" on $p(x)$, invert each linear system on x_o if possible, and stops as soon as one guess on $p(x)$ is correct, otherwise makes a new choice for x_v. Then, once a value x is found, the signer outputs $z = T^{-1}(x)$. To verify a signature z of y, the verifier, simply checks that $\mathcal{P}(z) = y$.

4.3 Rationale, sizes and performance

The direct attack using Gröbner basis computation can be performed by fixing n_v variables out of n. The resulting system has n_o equations in n_o variables and we estimate with simulation that it behaves like a random system with the F4 algorithm as soon as $n_v \geq n_o$. So, the solving complexity of hybrid attacks can be used when $n_o = n_v$. We estimate that the Pepper perturbation thwarts the Kipnis-Shamir attack, and as a rule of thumb, that $q^{n_v} > 2^\lambda$ and $q^{n_o} > 2^{2\lambda}$ should be sufficient to thwart other obvious attacks.

There are many ways to choose the parameters (see Table 1). So, for a security level of $\lambda = 128$, (assuming $\omega = 2.37$) we propose the following parameters:

$q = 821$, $n_o = 27$, $n_v = 27$. The public key size is 936 Kilobytes, a signature size is $(27+27)*10 = 540$ bits (each element of $\mathbb{F}_{821}$ is coded on 10 bits), or 523 bits if it is coded as a binary number smaller than 821^{54}. On our Magma simulation, on average, a signature takes 330ms. and a verification 130ms.

Note that the signature is parallelizable and could take benefit from multi-core CPUs.

Table 1. Complexity of hybrid attacks for random systems of n equations in n variables of degree 3. k: number of variables to fix for the best trade-off, D_{reg}: degree of regularity for the best trade-off. $C = q^k \binom{n-k+D_{reg}}{D_{reg}}^{\omega}$, $\omega = 2.37$

q	n	$\log_2 q^n$	$\log_2 C$	D_{reg}	k	q	n	$\log_2 q^n$	$\log_2 C$	D_{reg}	k
5	56	130.03	130.03	1	56	23	34	153.80	128.42	10	16
7	47	131.95	129.72	5	36	29	33	160.31	128.74	16	9
8	45	135.00	130.90	6	32	37	32	166.70	129.24	18	7
9	42	133.14	128.60	9	24	49	31	174.06	128.19	16	8
11	40	138.38	128.63	9	22	73	30	185.69	128.06	18	6
13	39	144.32	131.30	8	23	121	29	200.65	128.31	19	5
16	37	148.00	130.41	11	17	277	28	227.18	128.04	22	3
19	36	152.93	130.38	11	16	821	27	261.39	128.02	24	2

4.4 Shorter signature thanks to UOV

In the previous section, we assumed that the input size of the signature function was twice the size of the security level, which enables to "plug" the output of a hash function of the same size, instead of the message itself, whenever the size of the message exceeds this size. This is the classical mode of signature using a hash function. In this section, we show how to reduce the size of the signature by exploiting the special property of UOV, that the inversion of UOV is based on the linearity of the oil variables. This idea was first exposed in [11].

Let's now suppose that y is the hash to sign, its size is at least 2λ bits as expected. Say it can be expressed as n' elements of $\mathbb{F}_q$. We define a UOV-Pepper scheme $\mathcal{P}$ as previously, with a parameter n_o satisfying $2^\lambda \leq q^{n_o} \leq 2^{2\lambda}$ (instead of $q^{n_o} \geq 2^{2\lambda}$ previously). We now introduce a public function $L(x, y)$: $\mathbb{F}_q^n \times \mathbb{F}_q^{n'} \mapsto \mathbb{F}_q^{n_o}$. We require this function to have the following properties: to be affine in x, to be easy to compute, and to be collision-free in y[3] with good probability. First we may write: $L(x, y) = g_0(y) + \sum_{i=1}^{n} x_i g_i(y)$ which is by design affine in x. Then for instance the public functions g_i may be random degree-1 (n', n_o)-polynomials. Or, in order to avoid to describe them, they can be also output by a pseudo random generator with y used as a seed. In these two cases obviously, we can assume that L is collision-free.

Now, to sign a value y, the signer must find a value z satisfying $\mathcal{P}(z) = L(z, y)$ and proceeds as follows. As y is given, the term $L(z, y)$ is evaluated in y and

[3] It means that it should be difficult to find y and y' such that $L(., y) = L(., y')$

the equation becomes $\mathcal{P}(z) = \alpha_0 + \sum_{i=1}^{n} \alpha_i z_i$, where the α_i belong to $\mathbb{F}_q^{n_o}$. Solutions in z can be found with the same method as before: make the change of variables $z = T^{-1}(x)$, draw vinegar variables at random, guess $p(x)$, solve the system in degree 1 obtained in the oil variables, check if $p(x)$ is correct, then go back in z variable. To verify a signature z of y, the verifier, simply checks that $\mathcal{P}(z) = L(z, y)$.

In table 2, S1 and S2 are two possible sizes for the signature, according to its coding : n words of $\lceil \log_2(q) \rceil$ bits, or 1 big word of $\lceil \log_2(q^n) \rceil$ bits. The public key amounts to the coefficients of n_o homogeneous degree-3 polynomials in $n_o + n_v$ variables, hence Pub= $n_o \binom{n_o+n_v+2}{3} \lceil \log_2 q \rceil$ in bits. Example, for a security level of $\lambda = 128$, we propose the following parameters: $q = 8$, $n_o = 45$, $n_v = 45$. The public key size is 2120 kilobytes, the signature is $(45 + 45) * 3 = 270$ bits, times for signature and verification are about 1.5s.

Table 2. Various sets of parameters, q, $n_o = n_v$, λ: security level, S1, S2: size of signature in bits, Pub: size of public key in kilobytes

q	n_o	λ	S1	S2	Pub.
5	35	80	210	163	783
7	29	80	174	163	373
8	27	80	162	162	281
11	25	80	200	173	277
13	24	80	192	178	236
79	18	80	252	227	133
223	17	80	272	266	122
5	39	90	234	182	1202
7	33	90	198	186	621
8	31	90	186	186	485
11	28	90	224	194	432
13	27	90	216	200	375
131	20	90	320	282	230
281	19	90	342	310	212

q	n_o	λ	S1	S2	Pub.
5	44	100	264	205	1939
7	36	100	216	203	876
8	34	100	204	204	698
11	31	100	248	215	646
13	30	100	240	223	568
157	22	100	352	321	334
457	21	100	378	372	313
5	56	128	336	261	5050
7	47	128	282	264	2519
8	45	128	270	270	2120
11	40	128	320	277	1772
121	29	128	406	402	869
277	28	128	504	455	972
821	27	128	540	523	936

q	n_o	λ	S1	S2	Pub.
5	83	192	498	386	24160
7	71	192	426	399	12976
8	68	192	408	408	10928
11	62	192	496	429	10091
13	59	192	472	437	8285
277	43	192	774	698	5309
433	42	192	756	736	4836
7	96	256	576	540	43134
8	92	256	552	552	36406
11	83	256	664	575	32213
13	80	256	640	593	27821
23	71	256	710	643	21626
277	58	256	1044	942	17417
421	57	256	1026	994	16254

4.5 Other attacks

Since we have chosen degree-3 UOV with perturbation, we have searched for a possible distinguisher of the public key or a statistical attack that may cancel the perturbation (see [10]). To our best knowledge (see [1]), the most appropriate tool that could give insights of the secret trapdoor is the differential at a point. We recall that the differential of f at k is defined as $\big(\Delta_k f\big)(x) = f(x+k) - f(x) - f(k) + f(0)$. When f is a degree-3 polynomial, its differential is a degree-2 polynomial from which one can extract two homogeneous parts of degree 2 and 1, that we note respectively H_2 and H_1. For example, since we have $\deg(p) = 2$ and $\deg(P) = 1$, then $H_2\big(\Delta_k pP\big)(x) = \big(\Delta_k p\big)(x)P(x) + p(x)P(k)$

and $H_1\big(\Delta_k pP\big)(x) = \big(\Delta_k p\big)(x)P(k) + p(k)P(x)$. More precisely in our case, p is a random quadratic form that can be chosen without loss of generality with full rank, and $P(x) = \sum_{i=1}^{n_o} \alpha_i x_i + L(x_v)$, where the family $\{\alpha_i\}$ can be chosen linearly independent, and L of full rank. Therefore the number of points k cancelling $p(k)$ and $P(k)$ or $\Delta_k p$ is negligible.

As explained in sec. 3, one can also get a new equation cancelling $p(x)$, using the fact that $p(x)^q = p(x)$, namely $\tilde{f}^q(x)P(x) - \tilde{f}(x)P^q(x) = f^q(x)P(x) - f(x)P^q(x)$. However, since P is a random linear polynomial in all variables, the latter expression loses most certainly the properties due to the UOV-shape of f.

4.6 Variants and further directions

An immediate generalization of our idea is to replace the tweak $p(x)P(x)$ by a sum with similar terms: $\sum_{i=1}^{s} p_i(x)P_i(x)$, where s is a small number. This variant seems to have significant effects on the signature scheme. First, the impact on the performance is clearly a slow down factor q^s on the search of a signature. However, it has also an impact on the security, since it "hides" more deeply the trapdoor inside the public key. Experiments show that it compensate for a decrease of the number of vinegar variables. For instance, for the same level $\lambda = 128$, we may propose the following parameters: $q = 8$, $s = 3$, $n_o = 45$, $n_v = 42$. The public key size is 1917 kilobytes, the signature is $(45+42)*3 = 261$ bits.

As for the secret trapdoor f, we also have tried cubic HFE and cubic triangular systems. This could have have been promising since these schemes may be used in signature and encryption mode as well, but unfortunately the differential attack still defeats the perturbation by revealing the rank defect.

5 Conclusion

"Pepper" is a new tool in Multivariate cryptography. From this idea and degree-3 UOV, we have obtained multivariate schemes for signature with very nice properties: they are fast, have a reasonable size of public key, and they resist all the known attacks in multivariate cryptography. We think that it is particularly important to notice that MinRank and Gröbner attacks are thwarted. However since "Pepper" is very new, and since many failures have occurred in multivariate cryptography, it is natural to be suspicious about new ideas and to wait for more analysis before using these schemes in real life applications. The fact in this paper we use public equations of degree 3 (instead of 2 usually in multivariate cryptography) might also open the door to new and powerful attacks...

References

1. John Baena, Daniel Cabarcas, Daniel E. Escudero, Karan Khathuria, and Javier Verbel. Rank analysis of cubic multivariate cryptosystems. In *International Conference on Post-Quantum Cryptography*, pages 355–374. Springer, 2018.

2. Magali Bardet, Maxime Bros, Daniel Cabarcas, Philippe Gaborit, Ray A. Perlner, Daniel Smith-Tone, Jean-Pierre Tillich, and Javier A. Verbel. Improvements of algebraic attacks for solving the rank decoding and minrank problems. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology - ASIACRYPT 2020 - 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7-11, 2020, Proceedings, Part I*, volume 12491 of *Lecture Notes in Computer Science*, pages 507–536. Springer, 2020.
3. Ward Beullens and Bart Preneel. Field lifting for smaller UOV public keys. In Arpita Patra and Nigel P. Smart, editors, *Progress in Cryptology - INDOCRYPT 2017 - 18th International Conference on Cryptology in India, Chennai, India, December 10-13, 2017, Proceedings*, volume 10698 of *Lecture Notes in Computer Science*, pages 227–246. Springer, 2017.
4. Charles Bouillaguet, Pierre-Alain Fouque, and Gilles Macario-Rat. Practical key-recovery for all possible parameters of SFLASH. In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology - ASIACRYPT 2011 - 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings*, volume 7073 of *Lecture Notes in Computer Science*, pages 667–685. Springer, 2011.
5. Antoine Casanova, Jean-Charles Faugère, Gilles Macario-Rat, Jacques Patarin, Ludovic Perret, and Jocelyn Ryckeghem. GeMSS: A Great Multivariate Short Signature. Research report, UPMC - Paris 6 Sorbonne Universités ; INRIA Paris Research Centre, MAMBA Team, F-75012, Paris, France ; LIP6 - Laboratoire d'Informatique de Paris 6, December 2017.
6. Jintai Ding, Joshua Deaton, Kurt Schmidt, Zheng Zhang, et al. Cryptanalysis of the lifted unbalanced oil vinegar signature scheme. In *Annual International Cryptology Conference*, pages 279–298. Springer, 2020.
7. Jintai Ding and Dieter Schmidt. Rainbow, a new multivariable polynomial signature scheme. In John Ioannidis, Angelos D. Keromytis, and Moti Yung, editors, *Applied Cryptography and Network Security, Third International Conference, ACNS 2005, New York, NY, USA, June 7-10, 2005, Proceedings*, volume 3531 of *Lecture Notes in Computer Science*, pages 164–175, 2005.
8. Vivien Dubois, Pierre-Alain Fouque, Adi Shamir, and Jacques Stern. Practical cryptanalysis of SFLASH. In *Annual International Cryptology Conference*, pages 1–12. Springer, 2007.
9. Jean-Charles Faugère and Antoine Joux. Algebraic cryptanalysis of hidden field equation (HFE) cryptosystems using gröbner bases. In Dan Boneh, editor, *Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings*, volume 2729 of *Lecture Notes in Computer Science*, pages 44–60. Springer, 2003.
10. Pierre-Alain Fouque, Louis Granboulan, and Jacques Stern. Differential cryptanalysis for multivariate schemes. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 341–353. Springer, 2005.
11. Aviad Kipnis, Jacques Patarin, and Louis Goubin. Unbalanced oil and vinegar signature schemes. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 206–222. Springer, 1999.
12. Aviad Kipnis and Adi Shamir. Cryptanalysis of the HFE public key cryptosystem by relinearization. In Michael J. Wiener, editor, *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California,*

USA, August 15-19, 1999, Proceedings, volume 1666 of *Lecture Notes in Computer Science*, pages 19–30. Springer, 1999.
13. Tsutomu Matsumoto and Hideki Imai. Public quadratic polynomial-tuples for efficient signature-verification and message-encryption. In *Workshop on the Theory and Application of of Cryptographic Techniques*, pages 419–453. Springer, 1988.
14. Jacques Patarin. Hidden fields equations (HFE) and isomorphisms of polynomials (IP): two new families of asymmetric algorithms. In Ueli M. Maurer, editor, *Advances in Cryptology - EUROCRYPT '96, International Conference on the Theory and Application of Cryptographic Techniques, Saragossa, Spain, May 12-16, 1996, Proceeding*, volume 1070 of *Lecture Notes in Computer Science*, pages 33–48. Springer, 1996.
15. Jacques Patarin, Gilles Macario-Rat, Maxime Bros, and Eliane Koussa. Ultrashort multivariate public key signatures. *IACR Cryptol. ePrint Arch.*, 2020:914, 2020.
16. Chengdong Tao, Albrecht Petzoldt, and Jintai Ding. Improved key recovery of the hfev- signature scheme. *IACR Cryptol. ePrint Arch.*, 2020:1424, 2020.

Appendices

A Magma code for the complexity of the hybrid attacks

```
HilbertSeries:= function(q,n,DEGS);
//Input
//  q size of the finite field
//  n number of variables
//  DEGS list of degree of equations
//Ouput
//  The index of the first non negative coefficient
//  of the Hilbert Series of the corresponding system
if #DEGS lt n then return "Under determined system"; end if;
if q eq 2 then return "Not implemented"; end if;
R<z>:= PowerSeriesRing(Rationals());
HS:= &*[ 1-z^d : d in DEGS] / (1-z)^n;
DREG := 0;
while Coefficient(HS,DREG) gt 0 do
    DREG +:= 1;
end while;
return DREG;
end function;

complexity:=function(q,n,Degs: omega := 2.37);
//Input
//  q size of the finite field
//  n number of variables
//  DEGS list of degree of equations
//  Optional: algebraic constant
//Output
```

```
//  Log2 of complexity
//  Estimated degree of regularity
//  Use of Field equations
dreg:=HilbertSeries(n,Degs);
dreg2:=HilbertSeries(n,Degs cat [q: i in [1..n]]);
res:=dreg2 lt dreg;
dreg:=Minimum(dreg,dreg2);
return Log(2,Binomial(n+dreg,n)^omega),dreg, res;
end function;

hybrid:=function(q,n,Degs);
//Input
//  q size of the finite field
//  n number of variables
//  DEGS list of degree of equations
//Output
//  Log2 of complexity
//  Estimated degree of regularity
//  Number of variables to fix : best trade-off
//  Use of Field equations
compmin:=1000;
dregmin:=0;
kmin:=0;
fieldmin:=false;
for k:=0 to n do;
    comp,dreg,field:=complexity(q,n-k,Degs);
    comp+:=k*Log(2,q);
    if comp lt compmin then
        compmin:=comp;
        kmin:=k;
        dregmin:=dreg;
        fieldmin:=field;
    end if;
end for;
return compmin, dregmin, kmin, fieldmin;
end function;
```