Complexity Estimates for the F_4 Attack on the Perturbed Matsumoto-Imai Cryptosystem

J. Ding[1], J.E. Gower[1], D. Schmidt[2], C. Wolf [3], and Z. Yin[1]

[1] Department of Mathematical Sciences,
University of Cincinnati, Cincinnati,
OH 45211-0025, USA
{ding, gowerj, yinzhi}@math.uc.edu
[2] Department of Electrical & Computer Engineering and Computer Science,
University of Cincinnati, Cincinnati,
OH 45211-0030, USA
dieter.schmidt@uc.edu
[3] K.U. Leuven ESAT-COSIC,
Kasteelpark Arenberg 10,
B-3001 Leuven-Heverlee, Belgium
Christopher.Wolf@esat.kuleuven.ac.be or chris@Christopher-Wolf.de

Abstract. Though the Perturbed Matsumoto-Imai (PMI) cryptosystem is considered insecure due to the recent differential attack of Fouque, Granboulan, and Stern, even more recently Ding and Gower showed that PMI can be repaired with the Plus (+) method of externally adding as few as 10 randomly chosen quadratic polynomials. Since relatively few extra polynomials are added, the attack complexity of a Gröbner basis attack on PMI+ will be roughly equal to that of PMI. Using Magma's implementation of the F_4 Gröbner basis algorithm, we attack PMI with parameters $q = 2$, $0 \leq r \leq 10$, and $14 \leq n \leq 59$. Here, q is the number of field elements, n the number of equations/variables, and r the perturbation dimension. Based on our experimental results, we give estimates for the running time for such an attack. We use these estimates to judge the security of some proposed schemes, and we suggest more efficient schemes. In particular, we estimate that an attack using F_4 against the parameters $q = 2, r = 5, n = 96$ (suggested in [7]) has a time complexity of less than 2^{50} 3-DES computations, which would be considered insecure for practical applications.

Keywords: public-key, multivariate, quadratic polynomials, perturbation, Gröbner basis.

1 Introduction

1.1 $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic Cryptosystems and Perturbation

$\mathcal{M}$ultivariate $\mathcal{Q}$uadratic ($\mathcal{MQ}$) public key cryptosystems, first introduced in [6], have become a serious alternative to number theory based cryptosystems such as RSA, especially for small devices with limited computing resources. Since

N.P. Smart (Ed.): Cryptography and Coding 2005, LNCS 3796, pp. 262–277, 2005.

solving a set of multivariate polynomial equations over a finite field appears to be difficult (analogous to integer factorization, though it is unknown precisely how difficult either problem is), it seems reasonable to expect that we can build secure multivariate public key cryptosystems and signature schemes. In the last ten years, there has been significant effort put into realizing practical implementations of this idea, and many schemes have been proposed: Matsumoto-Imai, HFE, HFEv, Sflash, Oil & Vinegar, Quartz, TTM, and TTS, to name but a few.

At this stage, we seem to be more successful in building multivariate signature schemes than encryption schemes. For example, Sflash^{v2} [1] has been recommended by the New European Schemes for Signatures, Integrity, and Encryption (NESSIE, [17]) as a signature scheme for constrained environments. For encryption schemes, the best choice is probably HFE [19]. However, for a secure system, one must choose parameters which lead to a rather inefficient scheme.

Internal perturbation [7] was introduced as a general method to improve the security of multivariate public key cryptosystems. Roughly speaking, the idea is to "perturb" the system in a controlled way so that the resulting system is invertible, efficient, and much more difficult to break. The first application of this method was to the Matsumoto-Imai (MI) cryptosystem, a system that is otherwise vulnerable to the linearization attack [18]. The resulting system, called the perturbed Matsumoto-Imai cryptosystem (PMI), is slower as one needs to go through a search process on the perturbation space. However, we believe that for realistic choices of parameters, PMI is still much faster than HFE and provides superior security against all known attacks, except the recent differential attack of Fouque, Granboulan, and Stern [13]. Fortunately PMI is easily repaired with the Plus (+) [20] method of externally adding relatively few random quadratic polynomials. In fact, in the most general case of PMI, as few as 10 polynomials will be sufficient to protect PMI from the differential attack. Since so few extra polynomials are needed to create a secure Perturbed Matsumoto-Imai-Plus (PMI+) scheme, there is no significant difference between the two schemes regarding the Gröbner bases attack complexity [5,26]. Therefore, for simplicity we will consider Gröbner bases attacks on PMI.

1.2 Attacks Against Perturbed Multivariate Cryptosystems

In [2] it is shown that the XL algorithm will always need more time and space than either the F_4 or F_5 version of the Gröbner basis algorithm. Hence, it suffices to consider only Gröbner basis attacks. Both algorithms are quite similar in that they use the original Buchberger algorithm to compute a Gröbner basis for a given ideal of polynomials, and so for practical reasons we use only the F_4 version. Therefore, in this paper we analyse the security of PMI against Gröbner basis attacks as it depends on the parameter r, the perturbation dimension, and n, the message length. Specifically we give estimates for the time complexity of the F_4 Gröbner basis attack on PMI. Based on our experimental results, we give formulæ for these estimates that can be used to evaluate the security of proposed PMI systems against such attacks, and suggest parameters that may give better performance while providing sufficient security. These results can

then be used to infer similar statements regarding the security of PMI+. Since [8] shows that differential analysis cannot be effectively used against PMI+, it is sufficient to consider Gröbner attacks against PMI+ to determine its security. Hence, the most successful attack against PMI+ can be found in [12] while the most successful one against PMI is [13].

1.3 Outline

The remainder of this paper is organised as follows. After introducing the MI and PMI cryptosystems in Section 2, we describe our experimental evaluation of the security of PMI in Section 3. We then interpret the data and make some suggestions for improvement and give some predictions for the security of proposed instances of PMI in Section 4. We present our conclusions in Section 5.

2 The Perturbed Matsumoto-Imai Cryptosystem

2.1 The Original Matsumoto-Imai Cryptosystem

Let k be a finite field of size q and characteristic 2, and fix an irreducible polynomial of $g(x) \in k[x]$ of degree n. Then $K = k[x]/g(x)$ is an extension of degree n over k, and we have an isomorphism $\phi : K \longrightarrow k^n$ defined by $\phi(a_0 + \cdots + a_{n-1}x^{n-1}) = (a_0, \ldots, a_{n-1})$.

Fix θ so that $\gcd(1 + q^\theta, q^n - 1) = 1$ and define $F : K \longrightarrow K$ by $F(X) = X^{1+q^\theta}$. Then F is invertible and $F^{-1}(X) = X^t$, where $t(1 + q^\theta) \equiv 1 \bmod q^n - 1$. Define the map $\tilde{F} : k^n \longrightarrow k^n$ by $\tilde{F}(x_1, \ldots, x_n) = \phi \circ F \circ \phi^{-1}(x_1, \ldots, x_n) = (\tilde{F}_1, \ldots, \tilde{F}_n)$. In this case, the $\tilde{F}_i(x_1, \ldots, x_n)$ are quadratic polynomials in the variables $x_1, \ldots, x_n$. Finally, let L_1 and L_2 be two randomly chosen invertible affine linear maps over k^n and define $\overline{F} : k^n \longrightarrow k^n$ by $\overline{F}(x_1, \ldots, x_n) = L_1 \circ \tilde{F} \circ L_2\,(x_1, \ldots, x_n) = (\overline{F}_1, \ldots, \overline{F}_n)$. The public key of the Matsumoto-Imai cryptosystem (MI or C*) consists of the polynomials $\overline{F}_i(x_1, \ldots, x_n)$. See [16] for more details.

2.2 The Perturbed Matsumoto-Imai Cryptosystem

Fix a small integer r and randomly choose r invertible affine linear functions $z_1, \ldots, z_n$, written $z_j(x_1, \ldots, x_n) = \sum_{i=1}^{n} \alpha_{ij} x_i + \beta_j$, for $j = 1, \ldots, r$. This defines a map $Z : k^n \longrightarrow k^r$ by $Z(x_1, \ldots, x_n) = (z_1, \ldots, z_r)$. Now randomly choose n quadratic polynomials $f_1, \ldots, f_n$ in the variables $z_1, \ldots, z_r$. The f_i define a map $f : k^r \longrightarrow k^n$ by $f(z_1, \ldots, z_r) = (f_1, \ldots, f_n)$. Define $\tilde{f} : k^n \longrightarrow k^n$ by $\tilde{f} = f \circ Z$, and $\overline{\overline{F}} : k^n \longrightarrow k^n$ by $\overline{\overline{F}} = \tilde{F} + \tilde{f}$. The map $\overline{\overline{F}}$ is called the perturbation of $\tilde{F}$ by $\tilde{f}$, and as with MI, its components are quadratic polynomials in the variables $x_1, \ldots, x_n$. Finally, define the map $\hat{F} : k^n \longrightarrow k^n$ by $\hat{F}(x_1, \ldots, x_n) = L_1 \circ \overline{\overline{F}} \circ L_2(x_1, \ldots, x_n) = (y_1, \ldots, y_n)$. The public key of the perturbed Matsumoto-Imai (PMI) cryptosystem consists of the components y_i of $\hat{F}$. See Fig. 1 for an illustration of this idea, and [7] for more details.

Note that for MI there is a bijective correspondence between plaintext and ciphertext. However, PMI does not enjoy this property. Indeed, for a given ciphertext $c \in k^n$, $\hat{F}^{-1}(c)$ may have as many as q^r elements, though we may use the technique suggested for HFE to distinguish the plaintext from the other preimages. It has been proposed [7] that we can choose the parameters of PMI ($q = 2, r = 6, n = 136$) so that the resulting system is faster than HFE, and also claiming a very high level of security.

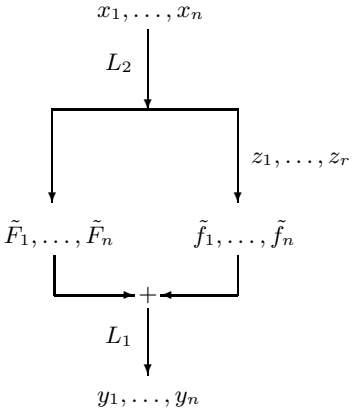

Fig. 1. Structure of PMI

2.3 Known Attacks Against MI and PMI

The most successful attack against MI is that of Patarin [18]. At present it is not clear whether this approach can be generalised to attack PMI, the main difficulty being that PMI mixes the operations in the extension field K from MI with the operations in the ground field k from the perturbation of MI. Another approach might involve ideas from the cryptanalysis of Sflash [14,15], though it is not immediately clear how this might work.

The differential attack [13] has rendered PMI insecure. However, it is easily repaired [8] using the Plus method of externally adding relatively very few Plus (+) polynomials [20]. The resulting scheme is call the Perturbed Matsumoto-Imai-Plus (PMI+) cryptosystem. Since the number of polynomials in PMI+ exceeds the number of unknowns by such a small amount, the attack complexity of a Gröbner basis attack very close to that of the same attack mounted against PMI. As a result, for simplicity we henceforth consider only PMI. These extra polynomials are added between the two linear transformations L_1, L_2. In particular, this means that we have $L_1 : k^{n+a} \to k^{n+a}$ now with $a \in \mathcal{N}$ added polynomials. These polynomials have the form

$$
\begin{aligned}
f_{n+1}(x_1,\ldots,x_n) := {} & \gamma'_{n+1,1,2}x_1x_2 + \ldots + \gamma'_{n+1,n-1,n}x_{n-1}x_n + \\
& \beta'_{n+1,1}x_1 + \ldots \beta'_{n+1,n}x_n + \alpha'_{n+1} \\
& \vdots \\
f_{n+a}(x_1,\ldots,x_n) := {} & \gamma'_{n+a,1,2}x_1x_2 + \ldots + \gamma'_{n+a,n-1,n}x_{n-1}x_n + \\
& \beta'_{n+a,1}x_1 + \ldots \beta'_{n+a,n}x_n + \alpha'_{n+a}
\end{aligned}
$$

for $\gamma', \beta', \alpha' \in_R k$ being random coefficients.

In [4] and [12], Gröbner bases have been used to break instances of HFE. By exploiting the underlying algebraic structure, they are able to break HFE in a far shorter time than it would take to solve a system of random equations [11,12]. For a fixed number of monomials in HFE, it can be shown that the running time will be polynomial. This result applies to MI as it uses only one monomial. The running time of this attack applied to PMI is not known.

3 Experiments with the F_4 Gröbner Basis Algorithm

3.1 Methodology

We attempted to experimentally determine the running time and memory requirements for a Gröbner basis attack on PMI. To this end we generated several instances of PMI. For each resulting set of polynomials $y_1, \ldots, y_n$ we chose several $(y'_1, \ldots, y'_n) \in k^n$ and timed how long it takes to find a Gröbner basis for the ideal $(y_1 - y'_1, \ldots, y_n - y'_n)$. Such a basis allows us to swiftly determine all $(x'_1, \ldots, x'_n) \in k^n$ such that $\hat{F}(x'_1, \ldots, x'_n) = (y'_1, \ldots, y'_n)$.

More specifically, we randomly generated 101 instances of PMI in Magma [3] for several values of n and r with $q = 2$, $14 \leq n \leq 59$, and $0 \leq r \leq 10$. In addition we randomly generated 101 elements in k^n and applied the F_4 version of the Gröbner basis implementation in Magma to each instance/element pair. In both cases, we used a uniform distribution on the private key/element from k^n. For all runs, we measured the memory and time needed until the algorithm found a solution. It did happen that some elements had no preimage under PMI, which is the same as with random systems of multivariate quadratic equations, hence we kept these timings in the sample. This decision was made as we were mainly interested in understanding the security of a signature scheme. For an encryption scheme, a more obvious choice would have been to encrypt random vectors $x \in k^n$ and then solving the corresponding equations, *i.e.*, $\hat{F}(x) = y$ for given $\hat{F}$ and y and "unknown" x.

We note that in theory it would be best to measure the maximal degree of the equations generated during a run of the F_4 algorithm. Unfortunately, Magma does not provide this feature, and so we had to use the indirect measurements of time and memory. It should also be noted that the F_5 algorithm [10] is said to be faster than the previous algorithm F_4 [9]. However, recent experiments by Steel show that the Magma implementation of F_4 is superior in the case of

HFE systems [23]. In particular, Steel was able to solve HFE Challenge 1 in less operations than Faugère with his own implementation of F_5. This is a rather surprising fact as F_5 should be faster than F_4 from a theoretical perspective in *all* cases. Still, Magma's implementation of F_4 achieves better timings than Faugère's implementation of F_5 when applied to HFE Challenge 1. For our experiments, we decided to use the F_4 implementation of Magma as it is the fastest, publicly available implementation of Gröbner base algorithms. We benchmarked its performance by solving random instances of PMI for various parameters n, r for the finite field $k = \mathrm{GF}(2)$. Although other ground fields with characteristic 2 are possible, we avoid them since solving PMI for a given private key takes an additional workload of $O(q^r)$. Also, the running time of Gröbner algorithms is very sensitive to the ground field k. Hence, it is difficult to obtain enough data for the cases $q = 4, 8$, and 16.

To ensure the accuracy and reliability of the data, we conducted the experiments with two independent teams, Team Q and Team Ω. Team Q used a cluster of identical AMD Athlon XP 2000+ with 900 MB of memory each, and Team Ω used an UltraSPARC-III+ 1.2 GHz dual processor with 8.0 GB of main memory. Because of these hardware and software differences, we expected to see differences in our measurements. However both data sets point to the same asymptotic behaviour. For brevity, we include only Team Q's data.

3.2 Empirical Data

It is clear that the case of $r = 0$ corresponds to MI, while the case of $r = n$ corresponds to a system of n randomly selected polynomials in n variables. Thus we expected the Gröbner basis attack on a system with $r = 0$ to be polynomial in time [12], while the same attack on a system with $r = n$ is expected to be exponential in time. Using our data, we wanted to answer two questions. First, for a fixed n we wanted to find the so-called "optimal perturbation dimension," *i.e.*, the minimal value of r for which a PMI system with parameters n and r is indistinguishable from a set of random polynomials. We also sought to obtain formulæ which would allow us to predict the running time behaviour of F_4 applied to PMI for any n and r.

The number of steps involved in attacking PMI with F_4 can be found in Table 5, while the memory requirements are shown in Table 6. Since no $\theta \in \mathbb{N}$ exists such that $\gcd(1 + 2^\theta, 2^n - 1) = 1$ for $n = 16, 32$, there is no corresponding instance of PMI and we hence have no data for these two cases. Each entry in these two tables is the median of 101 computations. This relatively small sample size was justified by additional experiments to determine the variation in larger (1001 computations) data sets. We found that the ratio of the maximum value to minimum value was always less than 2 in these larger sets. Data sets with median time below 0.05 seconds, or with memory requirements greater than 900 MB of memory were excluded from consideration in the final analysis on the grounds that they were either too noisy or suffered from the effects of extensive memory swapping. However, we actually performed many more experiments than are

listed in Tables 5 and 6. Moreover, all experiments that terminated prematurely were due to memory shortage and not time constraints.

4 Interpretation

From the point of view of cryptanalysis, most agree that it is the computational complexity that essentially determines the security of a cryptosystem. In our experiments we notice that the time and memory tables are closely correlated. The explanation for this can easily be seen from the structure of the F_4 algorithm. Therefore we believe it suffices to analyse the timing data, and hence we omit a detailed analysis of the memory usage. However, from our experiments we notice that the memory usage is on the same scale as that of the time complexity. Since memory is a much more critical constraint than time, in the end we believe it will be memory that will determine how far F_4 can go.

4.1 Polynomial and Exponential Models

It is known that the case $r = 0$ is precisely MI. Hence the attack from Faugère and Joux [12] using Gröbner bases should be polynomial. Thus we first consider the hypothesis that the data is well approximated by a polynomial model. Let $t(n, r)$ be the time to attack PMI with parameters n and r. We assume that our computer can perform $2 \cdot 10^9$ steps per second, and define the number of steps, $s(n, r) = 2 \cdot 10^9 t(n, r)$. We use $s(n, r)$ instead of $t(n, r)$ for all calculations. The polynomial model predicts the existence of constants $\alpha = \alpha(r)$ and $\beta = \beta(r)$ such that $s(n, r)$ is well approximated by αn^β. Table 1 shows the fitting obtained from applying the method of least squares for a fixed r on the data $\{\log_2 n, \log_2 s(n, r)\}$, where ε is the error sum of squares for this data set. We note that for $r = 0$, the exponent $\beta = 7.16$ is greater than that predicted in [12], though we speculate that the difference may be due to the fact that F_5 is used instead of F_4.

Table 1. Polynomial fittings

r	0	1	2	3	4	5	6	7	8	9	10
$\log_2 \alpha$	-4.81	-3.33	-12.64	-7.71	-13.74	-10.87	-29.38	-29.30	-29.84	-31.09	-30.27
β	7.16	7.12	9.50	9.22	10.81	10.18	14.90	15.02	15.17	15.48	15.28
ε	4.20	5.28	6.97	2.17	1.51	3.29	6.51	2.32	0.91	1.10	0.80

It should also be observed that there is some sort of "phase transition" that occurs in the fitting behaviour as r increase from 5 to 6. This is most obviously seen by looking at the values of $\log_2 \alpha$, which should not be either unusually small or unusually large, as this quantity represents the expected complexity for small n. Our data shows that as r increases from 5 to 6, α decreases from

Table 2. Exponential fittings

r	0	1	2	3	4	5	6	7	8	9	10
$\log_2 \alpha$	21.88	22.37	19.83	20.62	18.02	19.10	11.58	11.65	11.05	11.16	10.94
$\log_2 \beta$	0.27	0.30	0.45	0.55	0.72	0.68	1.15	1.18	1.22	1.21	1.23
ϵ	6.73	11.44	2.57	5.22	2.95	6.34	2.54	1.91	0.82	1.10	0.89

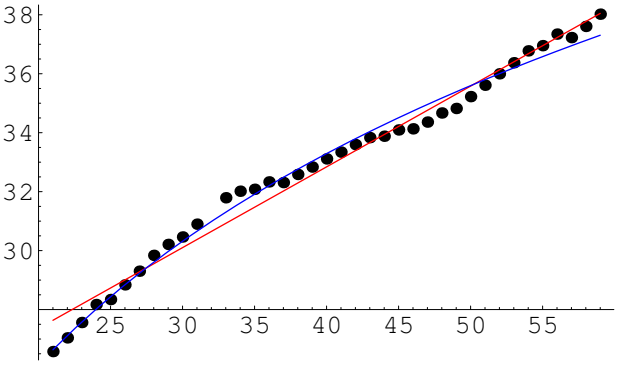

Fig. 2. Graph of $\log_2 s(n, 0)$

the reasonable scale of approximately 2^{-10} to 2^{-30}, which would seem to be unreasonable. Therefore, we suspect that it is at $r = 6$ where the transition from polynomial to exponential behaviour occurs.

To examine this possibility, we also consider the hypothesis that the data is well approximated by an exponential model. As before, let $t(n, r)$ be the time to attack PMI with parameters n and r. The exponential model predicts the existence of constants $\alpha = \alpha(r)$ and $\beta = \beta(r)$ such that $s(n, r)$ is well approximated by $\alpha\beta^n$. Table 2 shows the fitting obtained from applying the method of least squares for a fixed r on the data $\{n, \log_2 s(n, r)\}$, where again ε is the error sum of squares. To illustrate the fittings we present Table 2 and Fig. 2–5.

Observe that ε does not help to decide which fitting is more appropriate, so we must study the other parameters of the fitting. In particular, from Table 2 we note that again there is a transition happening with $\log_2 \alpha$ between $r = 5$ and $r = 6$. Once again, the important feature is the transition in $\log_2 \alpha$, which again happens between $r = 5$ and 6. Our reasoning is as before; *i.e.*, we do not believe α should either be too large or too small. In the case of the exponential model, α is too large for $r < 6$. Hence, we find the exponential model much more convincing for the case of $r \geq 6$, and the polynomial model a better fit for $r < 6$.

In summary, from this data we observe that the complexity makes a transition between two distinct regions, where the first region represents polynomial behaviour such as that of MI, and the second represents the exponential behaviour of a system defined by a random set of polynomials. We call the point at

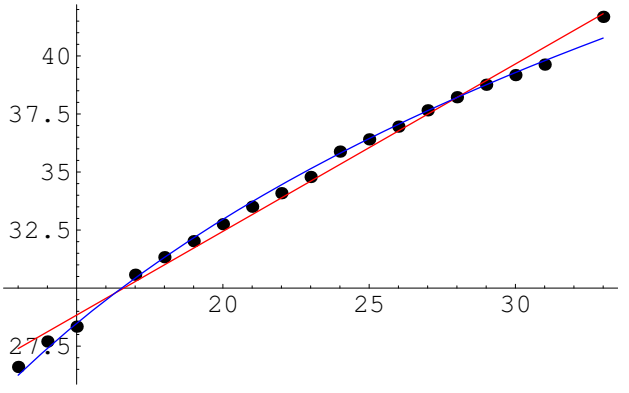

Fig. 3. Graph of $\log_2 s(n, 4)$

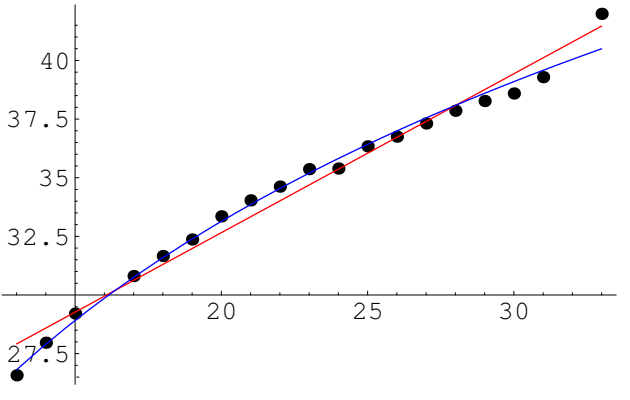

Fig. 4. Graph of $\log_2 s(n, 5)$

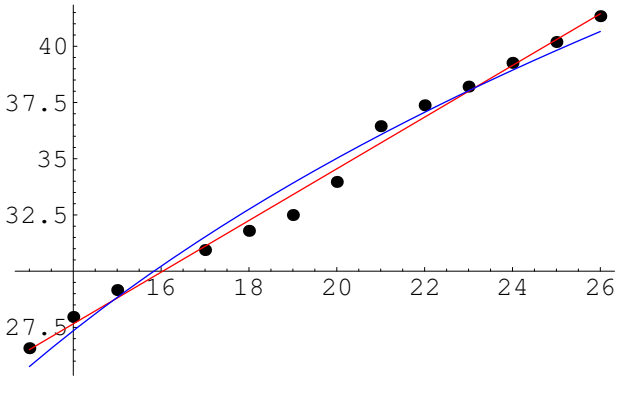

Fig. 5. Graph of $\log_2 s(n, 6)$

which this transition occurs the *phase transition point*, which we believe is $r = 6$. In our data, we did not find any other transition point. In particular, this behaviour fits well with the corresponding theory: as soon as the number of linearly independent monomials reaches a certain threshold, Gröbner base algorithms like F_4 or F_5 cannot make use of the structure of the private key anymore.

4.2 Optimal Perturbation Dimension

The analysis in the previous two sections assumed a fixed r and variable n. We now consider fixing n and varying r in order to study how the complexity achieves its maximum as r increases from zero (polynomial) to n (exponential). Fig. 6 illustrates the typical features of such a transition process.

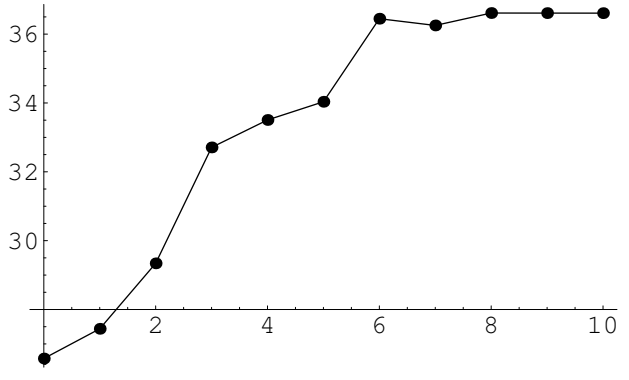

Fig. 6. Graph of $\log_2 s(21, r)$

It is important not to confuse the phase transition point with the minimal value of r (for a fixed n) for which the maximal complexity is achieved and no further advantage is gained by increasing r. We call such an r the *optimal perturbation dimension* for a given n. Based on the experiments above, we empirically determined this dimension and summarise our findings in Table 3.

Table 3. Optimal perturbation dimension

n	14...15	17...21	22...24
r	4	7	10

This data agrees very well with the theoretical explanation of the behaviour of Gröbner bases algorithms for solving HFE. There, the maximal degree of the polynomial equations derived during one computation is also discrete. It seems

that we get similar behaviour here. As was already pointed out, we could not make this observation directly as Magma does not provide the maximal degree. In order to confirm this behaviour and find formulæ that can be used to predict the location of the optimal perturbation dimension, we would need more data for larger values of n. However, this is not possible at present as the memory requirements for F_4 are quite severe, cf Table 6.

4.3 Practical Security

We now evaluate the security of some implementations of PMI. As was already pointed out, PMI by itself is insecure under the differential attack. Therefore the following analysis assumes that the Plus method has been applied. We first use the fittings to evaluate the security of the practical system proposed in [7]. We then use our fittings to propose some new optimised practical systems.

Evaluation of Practical Examples. In [7] the practical example suggested is the case of $q = 2, r = 6$, and $n = 136$. First we evaluate this example using the exponential model. The predicted security in this model is greater than 2^{160}. Assuming the validity of this model, the proposed system is very secure against Gröbner based attacks. In particular, according to the exponential model, since $\log_2 \beta > 1$ for $r \geq 6$, the running time of F_4 increases faster than exhaustive key search for $q = 2$. Therefore, these instances of PMI should be secure against these types of attacks assuming the validity of the exponential model. In fact, a practical and secure instance of PMI can use the parameters $q = 2, r = 6$, and $n = 83$ to meet the NESSIE requirements of 2^{80} 3-DES computations, if our model is valid. In particular, the security of the exponential model suggests a strength of 2^{100} 3-DES computations. However, for $q = 2, n < 80$, a brute-force search would take less than the required 2^{80} computations in 3-DES. Therefore, we decided to chose the first prime above 80, to rule out subfield-attacks as suggested in [22].

To be on the safe side, we also evaluate the system with $q = 2, r = 6$, and $n = 136$ using the polynomial model. The predicted security for this model is roughly 2^{70} 3-DES computations. According to this model, n must be greater than 227 to achieve the required NESSIE security level. However, if we consider the memory requirements needed to attack such a system, breaking these systems will be practically impossible with currently available resources. In particular, it is not clear at present how such an attack could be distributed over different machines, *e.g.*, using a distributed network of machines all agreeing to collaborate or possibly being captured by Trojan horses.

It is speculated in [7] that $q = 2$, $r = 5$, and $n = 96$ may also be secure. To evaluate this claim it is more appropriate to assume the polynomial model. According to this fitting, the security level is less than 2^{50} 3-DES computations, which is much less than the security level as requested in NESSIE. Therefore we conclude that this speculation may be overly optimistic. But again, we point out the severe memory requirements. Based on our experiments, we expect a number well above 2^{60} bytes.

The PMI scheme was originally proposed for use as an encryption scheme. One can easily modify the system for signature purposes; for example, one can use the Feistel-Patarin-Network, as in the signature scheme Quartz [21]. The scheme with the parameters $q = 2$, $r = 6$, $n = 83$ suggested above can be used in this way to build a secure signature scheme with only 83 bits.

Practical Perturbation Dimension. For PMI cryptosystems, one of the fundamental questions one should answer is how to choose r appropriately for a fixed n. From the behaviour of the complexity, a natural choice would be the optimal perturbation dimension, where the system becomes indistinguishable from a system defined by a random set of polynomials. However, we notice that this number may be too large for practical purposes, so we must choose some smaller value for r such that the system is practical and secure. Since the phase transition point is the minimal r that provides exponential behaviour we suggest this value for the perturbation dimension in practice.

4.4 Further Research

While the security of MI and also the behaviour of Gröbner base algorithm is well understood, this is not the case for PMI. Hence, it would be nice to have empirical data about the behaviour of other algorithms, *e.g.*, F_5 in the case of PMI. This proves difficult at present as F_5 is not available in a public implementation.

Using the maximal degree of the polynomials generated during a run on F_4 would have been more telling than our time or memory measurements. From [12], we expect this degree to be far more stable than the time or memory requirements. However, as Magma is also closed source, we could not modify the code to obtain this information. Hence, an open source implementation of F_4 would be preferable. In any case, we believe that this information would help us find the optimal perturbation dimension for a fixed n, the determination of which is important to completely understand PMI.

Apart from this, PMI seems secure against Gröbner attacks, and so we conclude that PMI+ is secure against both differential and Gröbner basis attacks.

5 Conclusions

In this paper we presented a security analysis of the PMI cryptosystem against Gröbner basis attacks. From this analysis we saw that for reasonable choices of parameters, PMI is secure against such attacks. Since PMI can be protected from the differential attack by externally adding as few as 10 random Plus polynomials, we conclude that this security analysis extends to that of PMI+.

Running various experiments with random instances of PMI, we established that PMI with small security parameter r can likely be solved in polynomial time. However, the rather large memory requirements will prevent such an attack from being practical with currently available resources. On the other hand, for

Table 4. Comparison between Quartz and PMI

	Quartz-7m	PMI
ground field k	GF(2)	
variables n	107	83
equations m	100	83
Signature Size [bits]	128	83
Public Key Size [kByte]	71	35

$r \geq 6$, we saw that the attacks using Gröbner bases become less efficient than exhaustive key search. Hence, we conclude that PMI is secure against these type of attacks. In particular, we suggest $q = 2, r = 6$, and $n = 83$ as a secure instance of PMI.

These results suggest that we can obtain a signature scheme from PMI that allows shorter signatures than other multivariate schemes, and in particular Quartz (128 bit, cf Table 4). For our comparison, we use the tweaked version Quartz-7m from [24–Sect. 4.3]. Moreover, this scheme would be the only known instance that gives a security level of 2^{80} 3-DES computations and still allows efficient decryption of messages in a multivariate public key scheme.

Acknowledgements

Christopher Wolf was supported in part by the Concerted Research Action (GOA) Mefisto-2000/06 and GOA Ambiorix 2005/11 of the Flemish Government and the European Commission through the IST Programme under Contract IST-2002-507932 ECRYPT.

References

1. M.-L. Akkar, N. T. Courtois, R. Duteuil and L. Goubin. A Fast and Secure Implementation of Sflash. In *PKC 2003*, LNCS 2567:267–278.
2. G. Ars, J.-C. Faugère, H. Imai, M. Kawazoe and M. Sugita. Comparison Between XL and Gröbner Basis Algorithms. In *Asiacrypt 2004*, LNCS 3329:338–353.
3. University of Sydney Computational Algebra Group. The MAGMA computational algebra system for algebra, number theory and geometry. `http://magma.maths.usyd.edu.au/magma`.
4. N. Courtois, M. Daum and P. Felke. On the Security of HFE, HFEv- and Quartz. In *PKC 2003*, LNCS 2567:337–350.
5. N. Courtois, A. Klimov, J. Patarin, and A. Shamir. Efficient Algorithms for Solving Overdefined Systems of Multivariate Polynomial Equations. In *Eurocrypt 2000*, LNCS 1807:392–407.
6. W. Diffie and M. Hellman. New Directions in Cryptography. In *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.
7. J. Ding. A New Variant of the Matsumoto-Imai Cryptosystem Through Perturbation. In *PKC 2004*, LNCS 2947:305–318.

8. J. Ding and Jason E. Gower. Inoculating Multivariate Schemes Against Differential Attacks. Pre-print, 12 pages. http://math.uc.edu/∼aac/pub/pmi+.pdf
9. J.-C. Faugère. A New Efficient Algorithm for Computing Gröbner Bases (F_4). In *Journal of Applied and Pure Algebra*, 139:61–88, June 1999.
10. J.-C. Faugère. A New Efficient Algorithm for Computing Gröbner Bases Without Reduction to Zero (F_5). In *ISSAC 2002*, pp. 75-83, ACM Press.
11. J.-C. Faugère. Algebraic Cryptanalysis of (HFE) Using Gröbner Bases. Technical report, Institut National de Recherche en Informatique et en Automatique, February 2003. http://www.inria.fr/rrrt/rr-4738.html, 19 pages.
12. J.-C. Faugère and A. Joux. Algebraic Cryptanalysis of Hidden Field Equation (HFE) Cryptosystems Using Gröbner Bases. In *Crypto 2003*, LNCS 2729:44-60.
13. P.-A. Fouque, L. Granboulan, and J. Stern. Differential Cryptanalysis for Multivariate Schemes. In *Eurocrypt 2005*, LNCS 3494:341–353.
14. W. Geiselmann, R. Steinwandt and T. Beth. Attacking the Affine Parts of SFlash. In *Cryptography and Coding – 8^{th} IMA International Conference*, LNCS 2260:355–359, 2001.
15. H. Gilbert and M. Minie. Cryptanalysis of SFLASH. In *Eurocrypt 2002*, LNCS 2332:288–298.
16. T. Matsumoto and H. Imai. Public Quadratic Polynomial-Tuples for Efficient Signature-Verification and Message-Encryption. In *Eurocrypt 1988*, LNCS 330:419–453.
17. NESSIE. European project IST-1999-12324 on New European Schemes for Signature, Integrity and Encryption. http://www.cryptonessie.org.
18. J. Patarin. Cryptanalysis of the Matsumoto and Imai Public Key Scheme of Eurocrypt'88. In *Crypto 1995*, LNCS 963:248–261.
19. J. Patarin. Hidden Fields Equations (HFE) and Isomorphisms of Polynomials (IP): Two New Families of Asymmetric Algorithms. In *Eurocrypt 1996*, LNCS 1070:33–48. Extended version: http://www.minrank.org/hfe.pdf.
20. J. Patarin, L. Goubin and N. Courtois. C^*_{-+} and HM: Variations Around Two Schemes of T. Matsumoto and H. Imai. In *Asiacrypt 1998*, LNCS 1514:35–50.
21. J. Patarin, L. Goubin and N. Courtois. QUARTZ, 128-Bit Long Digital Signatures. In *CT-RSA 2001*, LNCS 2020:298–307.
22. A. V. Sidorenko and E. M. Gabidulin. The Weak Keys for HFE. In *Proceedings of ISCTA 2003*, 6 pages.
23. A. Steel. Allan Steel's Gröbner Basis Timings Page. http://magma.maths.usyd.edu.au/users/allan/gb.
24. C. Wolf and B. Preneel. Asymmetric Cryptography: Hidden Field Equations. In *European Congress on Computational Methods in Applied Sciences and Engineering 2004*, 20 pages. Extended version: http://eprint.iacr.org/2004/072.
25. Christopher Wolf and Bart Preneel. Taxonomy of public key schemes based on the problem of multivariate quadratic equations. Cryptology ePrint Archive, Report 2005/077, 12th of May 2005. http://eprint.iacr.org/2005/077/, 64 pages.
26. B.-Y. Yang, J.-M. Chen, and N. Courtois. On Asymptotic Security Estimates in XL and Gröbner Bases-Related Algebraic Cryptanalysis. In *ICICS 2004*, LNCS 3269:410–423

Appendix

Table 5. Steps $s(n, r)$ in $\log_2$ for solving instances of PMI

n \ r	0	1	2	3	4	5	6	7	8	9	10
14	24.10	24.10	25.18	27.56	27.70	27.96	27.96	27.96	27.96	27.96	27.96
15	24.10	25.18	25.84	28.25	28.34	29.21	29.16	29.16	29.17	29.16	29.16
16											
17	25.18	25.79	26.84	29.78	30.58	30.81	30.94	32.35	32.35	32.35	32.35
18	25.32	25.84	27.83	31.13	31.33	31.66	31.79	33.37	33.37	33.37	33.37
19	25.84	26.55	28.34	31.71	32.02	32.37	32.50	34.33	34.32	34.31	34.32
20	26.22	27.58	28.83	32.16	32.75	33.35	33.97	35.32	35.35	35.34	35.34
21	26.58	27.44	29.34	32.71	33.51	34.03	36.45	36.25	36.61	36.61	36.61
22	27.04	28.51	30.14	33.12	34.09	34.62	37.38	37.16	37.46	38.28	38.29
23	27.56	28.42	30.16	33.68	34.79	35.30	38.21	37.97	38.98	39.14	38.97
24	28.17	29.57	30.66	33.84	35.88	35.39	39.26	40.16	40.28	39.82	40.02
25	28.34	29.21	31.03	34.84	36.41	36.34	40.19	41.31	41.40	41.28	
26	28.84	30.25	31.51	35.20	36.96	36.75	41.34				
27	29.30	30.73	31.97	35.84	37.66	37.31					
28	29.84	31.27	32.40	36.47	38.22	37.85					
29	30.21	31.53	32.69	36.75	38.76	38.27					
30	30.46	31.88	33.10	37.28	39.18	38.59					
31	30.90	32.30	33.57	38.08	39.63	39.29					
32											
33	31.79	33.01	35.08	39.28	41.68	41.99					
34	32.02	33.18	35.58	39.46							
35	32.08	33.30	36.01	39.75							
36	32.33	33.85	36.33	40.11							
37	32.31	33.80	37.09	40.55							
38	32.58	33.99	37.24	40.87							
39	32.84	34.18	37.46								
40	33.11	35.14	38.00								
41	33.34	34.48	38.63								
42	33.60	34.84	38.74								
43	33.83	34.94	39.06								
44	33.88	35.76	39.35								
45	34.09	35.41	40.07								
46	34.13	35.68	40.23								
47	34.36	35.67	40.36								
48	34.67	37.50	41.28								
49	34.82	36.01									
50	35.22	36.59									
51	35.61	36.75									
52	36.00	37.90									
53	36.37										
54	36.78										
55	36.95										
56	37.35										
57	37.23										
58	37.61										
59	38.02										

Table 6. Memory requirements in $\log_2$ for solving instances of PMI

n \ r	0	1	2	3	4	5	6	7	8	9	10
14	21.69	21.73	21.82	22.19	22.03	22.33	22.32	22.33	22.32	22.33	22.32
15	21.83	21.90	22.00	22.46	22.67	22.78	22.66	22.67	22.66	22.67	22.66
16											
17	22.34	22.33	22.44	23.12	23.52	23.59	23.53	24.74	24.73	24.74	24.73
18	23.01	23.01	22.97	23.55	24.06	24.07	24.47	25.41	25.41	25.41	25.41
19	23.81	23.81	23.47	24.08	24.60	24.63	24.71	26.08	26.09	26.09	26.09
20	24.46	24.46	24.22	24.72	25.17	25.20	25.77	26.71	26.71	26.71	26.71
21	24.46	24.46	24.22	24.72	25.17	25.20	28.00	27.10	27.10	27.10	27.10
22	24.46	24.46	24.22	24.85	25.40	25.64	28.35	27.61	27.61	28.56	28.57
23	24.46	24.46	23.47	25.25	25.75	26.36	28.94	28.12	29.18	29.29	29.05
24	24.46	24.46	24.22	25.50	26.24	26.83	29.61	29.90	29.99	29.72	29.78
25	24.46	24.46	24.22	26.11	26.44	27.35	29.90	30.41	30.55	30.43	
26	24.46	24.46	24.22	26.51	26.80	27.91	30.33				
27	24.46	23.36	24.22	26.87	27.85	28.30					
28	24.46	24.46	24.22	27.24	27.90	28.64					
29	24.46	24.46	24.35	27.55	28.37	29.05					
30	24.46	24.46	24.73	27.93	29.02	29.45					
31	24.46	24.46	24.79	28.16	29.40	29.79					
32											
33	24.46	24.64	25.70	28.82	30.39	30.69					
34	24.46	24.64	26.14	29.11							
35	24.46	24.64	26.57	29.40							
36	24.46	25.34	26.86	29.68							
37	24.46	25.34	27.01	29.89							
38	24.85	25.34	27.60	30.16							
39	24.85	26.05	27.87								
40	25.29	26.00	28.01								
41	25.30	26.06	28.21								
42	25.30	26.65	28.77								
43	25.30	26.77	28.71								
44	25.31	26.77	28.96								
45	25.51	27.30	29.51								
46	25.61	27.30	29.78								
47	25.72	27.30	29.68								
48	25.92	27.97	30.13								
49	25.98	27.93									
50	26.09	28.40									
51	26.28	28.41									
52	26.33	28.41									
53	26.43										
54	26.65										
55	26.67										
56	26.77										
57	27.03										
58	27.03										
59	27.11										