Cryptanalysis of an Authentication Scheme Using Truncated Polynomials

Markus Grassl[1] and Rainer Steinwandt[2]

[1] Institut für Quantenoptik und Quanteninformatik, Österreichische Akademie der Wissenschaften, Technikerstraße 21a, A-6020 Innsbruck, Austria, Markus.Grassl@uibk.ac.at

[2] Department of Mathematical Sciences, Florida Atlantic University, 777 Glades Road, Boca Raton, FL 33431, USA, rsteinwa@fau.edu

Abstract. An attack on a recently proposed authentication scheme of Shpilrain and Ushakov is presented. The public information allows the derivation of a system of polynomial equations for the secret key bits. Our attack uses simple elimination techniques to distill linear equations. For the proposed parameter choice, the attack often finds secret keys or alternative secret keys within minutes with moderate resources.

Keywords: cryptanalysis, authentication scheme, multivariate polynomials

1 Introduction

In [2] Shpilrain and Ushakov propose an authentication scheme, using 2×2 matrices over the algebra of "truncated polynomials" $\mathbb{F}_2[x]/(x^N)$. Below we describe a simple heuristic approach to derive linear equations from the public key, that often allows the secret key or an alternative secret key to be found. For the proposed value $N = 300$, typical observed running times are a few minutes using a computer algebra system on a Linux PC.

2 The proposed authentication scheme

For our purposes it is enough to recall the key generation of the proposal in [2]. Details of the actual authentication scheme are not relevant for our attack, and we refer to the original paper [2] for protocol details. By $R := \mathbb{F}_2[x]/(x^N)$ we denote the quotient of the univariate polynomial ring $\mathbb{F}_2[x]$ by the ideal (x^N), and we write $R^* := \{f(x) \in R : f(0) \neq 0\}$ for the elements in R with constant coefficient 1. Moreover, for $f \in R$ and $g = g(x) \in R \setminus R^*$, we write $f \circ g := f(g(x)) \in R$ for the functional composition of f with g. The key generation of the scheme proposed by Shpilrain and Ushakov can be summarized as follows:

Secret key: Choose $s_1, s_2, s_3, s_4 \in R^*$ uniformly at random.
Public key: Choose $w_1, w_2, w_3, w_4 \in R^*$ and $p_1, p_2 \in R \setminus R^*$ uniformly at random.[1] The public key is $(p_1, p_2, w_1, \ldots, w_4, t_1, \ldots, t_4)$ where

$$\begin{pmatrix} t_1 & t_2 \\ t_3 & t_4 \end{pmatrix} := \begin{pmatrix} s_1 \circ p_1 & s_3 \circ p_1 \\ s_2 \circ p_1 & s_4 \circ p_1 \end{pmatrix} \cdot \begin{pmatrix} w_1 & w_2 \\ w_3 & w_4 \end{pmatrix} \cdot \begin{pmatrix} s_1 \circ p_2 & s_2 \circ p_2 \\ s_3 \circ p_2 & s_4 \circ p_2 \end{pmatrix} .$$

Given such a public key $(p_1, p_2, w_1, \ldots, w_4, t_1, \ldots, t_4)$ for the proposed choice $N = 300$, the goal of our attack is to find $s'_1, s'_2, s'_3, s'_4 \in R^*$ with

$$\begin{pmatrix} t_1 & t_2 \\ t_3 & t_4 \end{pmatrix} = \begin{pmatrix} s'_1 \circ p_1 & s'_3 \circ p_1 \\ s'_2 \circ p_1 & s'_4 \circ p_1 \end{pmatrix} \cdot \begin{pmatrix} w_1 & w_2 \\ w_3 & w_4 \end{pmatrix} \cdot \begin{pmatrix} s'_1 \circ p_2 & s'_2 \circ p_2 \\ s'_3 \circ p_2 & s'_4 \circ p_2 \end{pmatrix} . \quad (1)$$

In other words $s'_1, \ldots, s'_4$ serve as alternative secret key. As noted in [2], this is sufficient to impersonate the owner of the actual secret $s_1, \ldots, s_4$.

3 Attacking the secret key

Our starting point is to replace each unkown bit η_{ij} of the secret key

$$s_i = 1 + \sum_{j=1}^{N-1} \eta_{ij} x^j + (x^N) \quad (1 \leq i \leq 4, 1 \leq j < N),$$

by an indeterminate y_{ij}. This yields a system of $4 \cdot (N-1)$ polynomial equations in $4 \cdot (N-1)$ unknowns (cf. [2, Section 4]):

1. Evaluate the matrix product on the right hand side of Equation (1) with each s'_i being replaced by a generic sum $1 + \sum_{j=1}^{N-1} y_{ij} x^j$ and computations being performed modulo (x^N).
2. For each matrix entry, equate the coefficients of all x^j $(1 \leq j < N)$ on both sides of Equation (1).

By construction, we know these polynomials to have a common root over $\mathbb{F}_2$. Hence we could add the relations

$$\{y_{ij}^2 - y_{ij} : 1 \leq i \leq 4, 1 \leq j < N\} \subset \mathbb{F}_2[y_{ij} : 1 \leq i \leq 4, 1 \leq j < N]$$

and then try to compute a lexicographic Gröbner basis to find a zero of the resulting (zero-dimensional) system of equations. Our attack tries to avoid this, potentially expensive, computation of a (lexicographic) Gröbner basis. Instead, we resort to a heuristic approach, which experimentally turned out to perform well.

[1] The specification in [2, Section 3] does not exclude the choice of w_is with absolute coefficient 0, but the analysis in [2, Section 4] uses $w_1, \ldots, w_4 \in R^*$. The proposed attack is not restricted to the case $w_1, \ldots, w_4 \in R^*$.

3.1 Deriving linear equations

We process the coefficients one by one and try to derive linear equations by means of a truncated Gröbner basis computation. More specifically, starting with degree $d = 1$, the proposed attack proceeds as follows:

0. Initialize $\mathcal{B} := \{y_{ij}^2 - y_{ij} : 1 \leq i \leq 4, 1 \leq j < N\}$.
1. Equate the coefficients of x^d on both sides of Equation (1). This yields a set of (four) polynomials $\mathcal{B}_d$. Let $\mathcal{B} := \mathcal{B} \cup \mathcal{B}_d$.
2. Using a graded reverse lexicographic term order, compute a truncated degree-2 Gröbner basis $\mathcal{G}$ of $\mathcal{B}$. In other words, when computing $\mathcal{G}$, ignore all S(yzygy)-polynomial pairs of degree greater than 2.
3. Extract all linear polynomials—i. e., polynomials of total degree 1—from $\mathcal{G}$, and compute a reduced echelon form. If one or several polynomials of the form $y_{i_0 j_0}$ or $y_{i_0 j_0} - 1$ are found, a uniquely determined part of the secret key has been recovered.
4. After echelonization, each linear polynomial has the form $y_{i_0 j_0} - \sum_{(i,j) \neq (i_0, j_0)} \gamma_{ij} y_{ij}$ with $y_{i_0 j_0}$ not occurring in other polynomials. Substituting each occurrence of $y_{i_0 j_0}$ in $\mathcal{B}$ with the respective $\sum_{(i,j) \neq (i_0, j_0)} \gamma_{ij} y_{ij}$, we can reduce the number of indeterminates.
5. If $d < N - 1$ and the complete secret key is not found yet, we can increase d by 1 and go back to Step 1.

Fig. 1. A heuristic attack on the private key.

In general, the approach in Figure 1 does not yield a unique solution for $s'_1, \ldots, s'_4$. In our experiments we simply set all variables in the remaining system of polynomial equations to 0, and tested if the resulting candidate key satisfies Equation (1). If not, we counted the attack as failed. This heuristic is based on the observation that in many cases the only constraints on the remaining variables are of the form $y_{ij}^2 = y_{ij}$, i. e., we can choose any value from $\mathbb{F}_2$ for y_{ij}.

3.2 Experimental results

Using the computer algebra system Magma [1], we experimented with different values for N. Our main interest was in the proposed choice $N = 300$, but the attack worked well with larger instances ($N = 1000$), too. Truncated degree-2 Gröbner bases were computed with Magma's command `GroebnerBasis(·,2)`.—Note that when computing a reduced (partial) Gröbner basis, the linear equations are already in echelon form.

Table 1 summarizes experimental results with Magma V2.14 on an Opteron 252 with 2.6 GHz. In case of a successful attack, the memory requirement was typically modest. We aborted computations where the memory consumption was large (several Gigabyte), and counted these cases as failed attacks as well.

N	#experiments	#success	$\frac{\#\text{success}}{\#\text{experiments}}$	time/success	memory/success
100	1000	763	76.3%	minimum: 3.7 s maximum: 12.3 s average: 6.2 s	minimum: 10 MB maximum: 62 MB average: 12.3 MB
200	100	85	85%	minimum: 36.8 s maximum: 123.3 s average: 67.1 s	minimum: 23 MB maximum: 46 MB average: 26.2 MB
300	100	88	88%	minimum: 135.6 s maximum: 629.6 s average: 295.4 s	minimum: 33 MB maximum: 68 MB average: 45.4 MB

Table 1. Running times and success rates, where *#success* counts those experiments where the secret key or an alternative secret key has been found; *time/success* and *memory/success* give the approximate running time and memory usage observed for a successful attack.

Certainly, there is room for improving our simple approach, but already in the present form, it seems fair to consider the attack as practical.

4 Conclusion

The above discussion gives ample evidence that in the proposed form the authentication scheme put forward in [2] does not provide strong cryptographic security guarantees: In experiments with moderate computational resources, secret keys or alternative secret keys could often be found within minutes.

Acknowledgment. We thank the Institut für Algorithmen und Kognitive Systeme at Universität Karlsruhe (Germany) for kindly permitting us to use their computer resources for our experiments.

References

1. Wieb Bosma, John J. Cannon, and Catherine Playoust. The Magma Algebra System I: The User Language. *Journal of Symbolic Computation*, 24:235–265, 1997.
2. Vladimir Shpilrain and Alexander Ushakov. An authentication scheme based on the twisted conjugacy problem. In *Applied Cryptography and Network Security – ACNS 2008*, volume 5037 of *Lecture Notes in Computer Science*, pages 366–372. Springer, 2008. Preprint available at `http://arXiv.org/abs/0805.2701v1`.