AN IDENTIFICATION SCHEME BASED ON SPARSE POLYNOMIALS

William D. Banks[1], Daniel Lieman[2] and Igor E. Shparlinski[3]

[1] Department of Mathematics, University of Missouri
Columbia, MO 65211, USA
bbanks@math.missouri.edu

[2] Department of Mathematics, University of Missouri
Columbia, MO 65211, USA
lieman@math.missouri.edu

[3] Department of Computing, Macquarie University
Sydney, NSW 2109, Australia
igor@mpce.mq.edu.au

Abstract. This paper gives a new example of exploiting the idea of using polynomials with restricted coefficients over finite fields and rings to construct reliable cryptosystems and identification schemes.

1 Overview

The recently discovered idea of using polynomials with restricted coefficients in cryptography has already found several cryptographic applications such as the NTRU cryptosystem [7], the ENROOT cryptosystem [4] and the PASS identification scheme [6]; see also [5].

In contrast to the constructions of NTRU and PASS, which consider classes of polynomials of low degree with many “small” non-zero coefficients, ENROOT introduced a public key cryptosystem where the polynomials are of high degree, but extremely sparse. In this paper, we give a new application of this idea to the design of a fast and reliable identification scheme.

Let q be a prime power and let $\mathbb{F}_q$ be the finite field of q elements.

Given a set $S \subseteq \mathbb{F}_q$, we say that a polynomial $G(X) \in \mathbb{F}_q[X]$ is an S-*polynomial* if every coefficient of G belongs to S, and we say that it is an *essentially S-polynomial* if $G(X)-G(0)$ is an S-polynomial. (This notation is somewhat reminiscent of the idea of S-units in number theory, and is not related to constructions in algebraic geometry.)

Finally, we say that a polynomial $G(X) \in \mathbb{F}_q[X]$ is τ-*sparse* if it has at most τ non-zero coefficients.

Throughout this paper $\log z$ denotes the binary logarithm of z.

The "hard" problem underlying our one-way functions can be stated as follows:

> *Given $2m$ arbitrary elements $\alpha_1, \dots \alpha_m, \gamma_1, \dots, \gamma_m \in \mathbb{F}_q$ and a set $S \subseteq \mathbb{F}_q$ of small cardinality, it is unfeasible to find a τ-sparse S-polynomial $G(X) \in \mathbb{F}_q[X]$ of degree $\deg G \le q-1$ such that $G(\alpha_j) = \gamma_j$ for $j = 1, \dots, m$, provided that q is of "medium" size relative to the choice of $m \ge 1$ and $\tau \ge 3$.*

More precisely, we expect that if one fixes the number of points m, the cardinality $|S|$ and the sparsity $\tau \ge 3$, then the problem requires exponential time as $q \to \infty$. Of course, we mean exponential time in the bit length of q, that is, in $\log q$.

Indeed, consider the special case $q = p$, where p is a prime number. Let $a_{ij} \equiv \alpha_j^i \pmod p$ and $c_j \equiv \gamma_j \pmod p$ be chosen so that $0 \le a_{ij}, c_j \le p-1$ for $i = 0, \dots, p-1$ and $j = 1, \dots, m$. In this (simplified!) case, the hard problem above is still equivalent to the hard problem of finding a feasible solution to the *integer programming problem*

$$\sum_{i=0}^{p-1} x_i \varepsilon_i a_{ij} + y_j p = c_j, \qquad j = 1, \dots, m, \qquad \sum_{i=0}^{p-1} \varepsilon_i \le \tau,$$

where $y_j \in \mathbb{Z}$, $x_i \in S$, and $\varepsilon_i \in \{0, 1\}$ for all i and j.

2 Basic Idea

Let us fix the finite field $\mathbb{F}_q$ and some integer parameters $k \ge 1$ and $r, s, t \ge 2$.

To create the signature *Alice* uses the following algorithm – which we denote SPIFI, for Secure Polynomial IdentiFIcation.

Initial Set-up

Step 1
Select at random k distinct elements $a_0, \ldots a_{k-1} \in \mathbb{F}_q$ and a random t-sparse $\{0,1\}$-polynomial $\varphi(X) \in \mathbb{F}_q[X]$ of degree at most $q-1$ and with $\varphi(0) = 0$.

Step 2
Compute $A = -\varphi(a_0)$, and put $f(X) = \varphi(X) + A$. Thus f is a t-sparse essentially $\{0,1\}$-polynomial with $f(a_0) = 0$ and $f(0) = A$.

Step 3
Compute $C_j = f(a_j)$, $j = 1, \ldots, k-1$.

Step 4
Make the values of A, $a_0, \ldots a_{k-1}$ and $C_1, \ldots, C_{k-1}$ public.

To verify *Alice*'s identity, *Alice* and *Bob* use the following procedure:

Verification Protocol

Step 1
Bob selects at random an s-sparse essentially $\{0,1\}$-polynomial $h(X) \in \mathbb{F}_q[X]$ with $h(0) = B$ and sends it to *Alice*.

Step 2
Alice selects at random an r-sparse $\{0,1\}$-polynomial $g(X) \in \mathbb{F}_q[X]$ of degree at most $q-1$ with $g(0) = 1$.

Step 3
Alice computes

$$F(X) \equiv f(X)g(X)h(X) \pmod{X^q - X}$$

and

$$D_j = g(a_j), \qquad j = 1, \ldots, k-1,$$

and sends the polynomial F and $D_1, \ldots, D_{k-1}$ to *Bob*.

Step 4
Bob computes

$$E_j = h(a_j), \qquad j = 1, \ldots, k-1,$$

and verifies that $F(X)$ is an rst-sparse $\{0,1,A,B,AB\}$-polynomial with $F(0) = AB$, and

$$F(a_j) = C_j D_j E_j, \qquad j = 0, \ldots, k-1,$$

where $D_0 = E_0 = 1$, $C_0 = 0$.

Of course, there is a negligible chance that the constructed polynomial $F(X)$ is not a $\{0, 1, A, B, AB\}$-polynomial, however if rst is substantially smaller than q this chance is extremely small (and in this case *Alice* and *Bob* can always repeat the procedure).

The sparsity of the polynomials involved guarantees the computational efficiency of this scheme.

In particular, using repeated squaring one can compute any power a^e with $a \in \mathbb{F}_q$ and an integer e, $0 \le e \le q-1$, in about $2 \log q$ arithmetic operations in $\mathbb{F}_q$ in the worst case and about $1.5 \log q$ arithmetic operations on average; see Section 1.3 of [1], Section 4.3 of [2], or Section 2.1 of [3]. Thus any τ-sparse $G(X) \in \mathbb{F}_q[X]$ can be evaluated at any point in about $O(\tau \log q)$ arithmetic operations in $\mathbb{F}_q$.

It is also well known that any element of $\mathbb{F}_q$ can be encoded by using about $\log q$ bits.

Finally, we remark that if $0 \in S \subseteq \mathbb{F}_q$ then any τ-sparse S-polynomial $G(X) \in \mathbb{F}_q[X]$ of degree at most $q-1$ can be encoded with about $\tau \log(q|S| - q)$ bits. Indeed, we have to identify at most τ positions at which G has a non-zero coefficient. Encoding of each position requires about $\log q$ bits. For each such position, about $\log(|S| - 1)$ bits are then required to determine the corresponding element of S.

For example, the signature must encode rst positions of the polynomial F (corresponding to its non-zero coefficients), which takes about $rst \log q$ bits. Each position requires two additional bits to distinguish between the possible coefficients 1, A, B and AB. The encoding of $D_1, ..., D_{k-1}$ requires about $(k-1)\log q$ bits. Thus, the total signature size is $(rst + k - 1)\log q + 2rst$ bits.

Putting everything together, after simple calculations we derive that, using the naive repeated squaring exponentiation,

- the *initial set-up* takes $O(kt \log q)$ arithmetic operations in $\mathbb{F}_q$;
- the *private key size* is about $(t+1)\log q$ bits;
- the *public key size* is about $2k \log q$ bits;
- *signature generation*, that is, computation of the polynomial F and elements D_j, $j = 0, \ldots, k-1$, takes $O(rst)$ arithmetic oper-

ations with integers in the range $[0, 2q-2]$ and $O\left((k-1)r\log q\right)$ arithmetic operations in $\mathbb{F}_q$;

- ○ the *signature size* is about $(rst + k - 1)\log q + 2rst$ bits;
- ○ *signature verification*, that is, computation $F(a_j)$ and the products $C_jD_jE_j$, $j = 0, \ldots, k-1$, takes about $O\left(ksrt\log q\right)$ arithmetic operations in $\mathbb{F}_q$.

We remark that the practical and asymptotic performance of this scheme can be improved if one uses more sophisticated algorithms to evaluate powers and sparse polynomials; see [1–3, 9, 11]. In particular, one can use precomputation of certain powers of the a_j's and several other clever tricks which we do not consider in this paper.

3 Possible Attacks

It is clear that recovering or faking the private key (that is, finding a t-sparse essentially $\{0,1\}$-polynomial polynomial $\widetilde{f}(X) \in \mathbb{F}_q[X]$ with $\widetilde{f}(0) = A$, and $\widetilde{f}(a_j) = C_j$, $j = 0, \ldots, k-1$, $C_0 = 0$) or faking the signature (that is, finding a rst-sparse $\{0, 1, A, B, AB\}$-polynomial $\widetilde{F}(X) \in \mathbb{F}_q[X]$ with $\widetilde{F}(0) = AB$, and $\widetilde{F}(a_j) = C_jD_jE_j$, $j = 0, \ldots, k-1$) represent the same problem (with slightly different parameters).

We also remark that that without the reduction

$$f(X)g(X)h(X) \pmod{X^q - X},$$

one of the one possible attacks would be via polynomial factorization. In particular, in a practical implementation of this scheme, one should make sure that both f and g have terms of degree greater than $q/2$ (so there are some reductions). Moreover, even without the reduction modulo $X^q - X$, the factorization attack does not seem to be feasible because of the large degrees of the polynomials involved; all known factorization algorithms (as well as their important components such as irreducibility testing and the greatest common divisor algorithms) do not take advantage of sparsity or any special structure of the coefficients; see [2, 10]. In fact the first factor any of this algorithms will find will be the trivial one, that is, $X - a_0$.

However the quotient $F(X)/(X - a_0)$ is most likely neither sparse nor a $\{0,1\}$-polynomial.

It is also possible that by using some "clever" choice of polynomials h, after several rounds of identification, *Bob* will be able to gain some information about f. Although the polynomials g are supposed to prevent him from doing this, in the next section we present another idea, which can be applied to other situations and which should make this attack completely unfeasible.

One might also consider lattice attacks. In the abstract, they could succeed, but since the dimension of the lattice would be equal to the (very large) degree of the polynomials involved, any such attack would be completely unfeasable. In particular, with current technology one can reduce lattices of degrees in the hundreds, while our lattices will have dimension roughly 2^{31}.

Finally, the probability of success in a brute force attack to discover or fake the signature, when the attacker selects a random rst-sparse $\{0, 1, A, B, AB\}$-polynomial $\widetilde{F}(X) \in \mathbb{F}_q[X]$ with $\widetilde{F}(0) = AB$ that verifies only if $\widetilde{F}(a_j) = C_j D_j E_j$, $j = 0, \ldots, k-1$, is about

$$\min\left\{ 4^{-rst+1} \binom{q-1}{rst-1}^{-1}, q^{-k} \right\}.$$

Similarly, the probability of randomly guessing or faking the private key f is

$$\min\left\{ \binom{q-1}{t-1}^{-1}, q^{-k} \right\}.$$

In particular, we do not see any security flaws in this scheme even if $k = 1$. While the choice $k = 1$ has the obvious advantage of minimizing the signature size (in this case, *Alice* sends only the polynomial F), in order to provide the required level of security, quite large values of q must be used. From a practical standpoint, it is very convenient to work in the field with $p = 2^{31} - 1$ elements. Thus, to guarantee the 2^{90} level of security, which is currently accepted as standard, it is better to take $k \geq 3$. We believe that the choices $q = p = 2^{31} - 1$, $r = s = t = 5$, and $k = 3$ provide a fast, short (about 4200 bits long), and reliable signature.

4 Modification of the Basic Scheme

To prevent *Bob* from gaining any useful information about f by selecting some special polynomials h, *Alice* can select $a \in \mathbb{F}_q$ and two t-sparse essentially $\{0,1\}$-polynomials $f_1(X), f_2(X) \in \mathbb{F}_q[X]$ of degree at most $q-1$ which for some $A, C_1, \ldots, C_{k-1} \in \mathbb{F}_q$ and distinct $a_0, \ldots, a_{k-1} \in \mathbb{F}_q$ satisfy the conditions

$$f_1(0) = f_2(0) = A \tag{1}$$

and

$$f_1(a_j) = f_2(a_j) = C_j, \qquad j = 0, \ldots, k-1, \tag{2}$$

where $C_0 = 0$.

To do so, *Alice* selects a certain parameter $n < t$, considers random $\{-1,1\}$-polynomials $\psi(X)$ and tries to find a root of this polynomial over $\mathbb{F}_q$. For values of n of reasonable size this can be done quite efficiently, at least in probabilistic polynomial time; see [2, 10].

It follows from Theorem 3 of [8] that for sufficiently large q the probability of a monic polynomial of degree n over $\mathbb{F}_q$ having k distinct roots in $\mathbb{F}_q$ is

$$P_k(n,q) = \sum_{m=k}^{\infty} \binom{q}{m} q^{-m} \sum_{l=0}^{n-m} (-1)^l \binom{q-m}{l} q^{-l}.$$

In particular,

$$\lim_{n\to\infty} \lim_{q\to\infty} P_k(n,q) = \frac{1}{k\,!\,e^k}.$$

Therefore, after $O(k!e^k)$ *Alice* will find with high probability an n-sparse $\{-1,1\}$-polynomial $\psi(X) \in \mathbb{F}_q[X]$, having k distinct roots $a_0, \ldots, a_{k-1} \in \mathbb{F}_q$. Then she can write $X\psi(X) = \varphi_1(X) - \varphi_2(X)$ where φ_1, φ_2 are $\{0,1\}$-polynomials. Obviously,

$$\varphi_1(a_j) = \varphi_2(a_j), \qquad j = 0, \ldots, k-1,$$

and

$$\varphi_1(0) = \varphi_2(0) = 0$$

Then *Alice* selects a random $(t-n)$-sparse $\{0,1\}$-polynomial $\varphi(X) \in \mathbb{F}_q[X]$ of degree at most $q-1$ and with $\varphi(0) = 0$ Now *Alice* puts

$$f_i(X) = \varphi(X) + \varphi_i(X) + A, \qquad i = 1, 2,$$

where

$$A = -\varphi(a_0) - \varphi_1(a_0) = -\varphi(a_0) - \varphi_2(a_0).$$

Thus f_1 and f_2 are t-sparse essentially $\{0,1\}$-polynomials which satisfy (1) and (2). Therefore, now *Alice* can alternate f_1 and f_2 in a random order.

Instead of the sum $\varphi(X)+\varphi_i(X)$, $i = 1, 2$, one can also consider more complicated linear combinations with $\{0,1\}$-polynomial coefficients. For example, one can put

$$f_i(X) = \varphi(X) + \psi(X)\varphi_i(X) + A, \qquad i = 1, 2,$$

for a random $\{0,1\}$-polynomial $\phi(X) \in \mathbb{F}_q[X]$ and

$$A = -\varphi(a_0) - \psi(a_0)\varphi_1(a_0) = -\varphi(a_0) - \psi(a_0)\varphi_2(a_0).$$

5 Concluding Remarks

It is natural to try to construct and use more than two t-sparse essentially $\{0,1\}$-polynomials which take the same value at k distinct points. However our approach of Section 4 does not seem to extend to this case.

Acknowledgments. A part of this work was done during visits by W. B. and D.L. to Macquarie University, whose hospitality and support are gratefully acknowledged.

Work supported in part, for D. L. by the National Science Foundation and a Big 12 Faculty Fellowship from the University of Missouri and for I. S. by the Australian Research Council.

References

1. H. Cohen *A course in computational algebraic number theory*, Springer-Verlag, Berlin, 1997.
2. J. von zur Gathen and J. Gerhard, *Modern computer algebra*, Cambridge Univ. Press, Cambridge, 1999.
3. D. M. Gordon, ‘A survey of fast exponentiation methods’, *J. Algorithms*, **27** (1998), 129–146.

4. D. Grant, K. Krastev, D. Lieman and I. E. Shparlinski, ‘A public key cryptosystem based on sparse polynomials’, *In: Proc. International Conference on Coding Theory, Cryptography and Related Areas, Guanajuato, 1998, Lect. Notes in Comp. Sci.*, Springer-Verlag, Berlin, 1999 (to appear).
5. J. Hoffstein, B. S. Kaliski, D. Lieman, M. J .B. Robshaw and Y. L. Yin, ‘A new identification scheme based on polynomial evaluation’, *Preprint*, 1997, 1–9.
6. J. Hoffstein, D. Lieman and J. H. Silverman, ‘Polynomial Rings and Efficient Public Key Authentication’, *In: Proc. the International Workshop on Cryptographic Techniques and E-Commerce*, City University of Hong Kong Press, to appear.
7. J. Hoffstein, J. Pipher and J. H. Silverman, ‘NTRU: A ring based public key cryptosystem’, *Lect. Notes in Comp. Sci.*, Springer-Verlag, Berlin, **1433** (1998), 267–288.
8. A. Knopfmacher and J. Knopfmacher, ‘ounting polynomials with a given number of zeros in a finite field’, *Linear and Multilinear Algebra*, **26** (1990), 287–292.
9. N. Pippenger, ‘On the evaluation of powers and monomials’, *SIAM J. Comp.*, **9** (1980), 230–250.
10. I. E. Shparlinski, *Finite fields: Theory and computation*, Kluwer Acad. Publ., Dordrecht, 1999.
11. A. C.-C. Yao, ‘On the evaluation of powers’, *SIAM J. Comp.*, **5** (1976), 100–103.