

Polynomial Rings and Efficient Public Key Authentication II

Jeffrey Hoffstein and Joseph H. Silverman

Abstract. In a recent paper [3] a highly efficient public key authentication scheme called PASS was introduced. In this paper we show how a small modification in the scheme cuts the size of the public key and the commitment in half while reducing an already minimal computational load.

Keywords. Authentication, Digital Signature, Public Key

Non-Technical Description of Work. A new public key authentication method was introduced in [3] featuring high speed, moderate key sizes, very low processing power required for both prover and verifier, and rapid generation of public-private key pairs. The efficiency and flexibility of the scheme is such that in addition to high security applications, it is also suitable for use on Smart Cards and in any other context, such as micropayments, where overhead considerations have made more traditional authentication schemes impractical. In this paper we show how the PASS scheme can be improved still further, reducing the already minimal computations of the prover substantially and decreasing communication requirements.

§1. Introduction

In a recent paper [3], a new highly efficient scheme for public key authentication and digital signatures called PASS was introduced. The ideas underlying PASS are related to the ideas originating in [1] and [2]. Each of these three papers used a combination of algebraic and analytic techniques in the context of a commutative ring

$$R = (\mathbf{Z}/q\mathbf{Z})[x]/(x^N - 1), \quad (1)$$

where q and N are moderately sized relatively prime integers.

In order to avoid excessive duplication of exposition, we will assume some familiarity with the previous paper [3]. We will, however, repeat some definitions and concepts when it appears that this would be useful. Thus this paper should be readable without reference to [3].

The general idea in the earlier paper [3] is as follows. Pearl, the prover, wishes to prove her identity to Vinnie, the verifier. Pearl has a secret key (f, f') consisting of a pair of “short” polynomials in R , i.e., having coefficients 1, -1 , and 0. Pearl’s public key is the collection of values $\{f(\alpha), f'(\alpha)\}_{\alpha \in S}$, where α varies over a set S consisting of half the numbers modulo q .

To identify herself, Pearl randomly picks a pair (g, g') of short polynomials in R . She keeps (g, g') secret, but as her commitment, Pearl reveals $\{g(\alpha), g'(\alpha)\}_{\alpha \in S}$, the collection of values of g and g' at the points in S . The verifier Vinnie sends Pearl a challenge c_0 that Pearl hashes with the commitment to produce a 4-tuple of extremely short polynomials (c_1, c_2, c_3, c_4) . Pearl computes and reveals the polynomial

$$h = c_1 * f * g + c_2 * f * g' + c_3 * f' * g + c_4 * f' * g'.$$

(Note all polynomial multiplications take place in the ring R .)

In order to verify Pearl's identity, Vinnie first checks that h is fairly short, and second he checks that the identity

$$h(\alpha) = c_1(\alpha)f(\alpha)g(\alpha) + c_2(\alpha)f(\alpha)g'(\alpha) + c_3(\alpha)f'(\alpha)g(\alpha) + c_4(\alpha)f'(\alpha)g'(\alpha)$$

is true for all $\alpha \in S$. If h passes both of these tests, then Vinnie accepts Peral's proof of identity, i.e., that she has knowledge of the secret short polynomial f .

In this paper we describe a modified version of the above scheme in which the public key and the commitment each consist of a single short polynomial, rather than a pair of short polynomials. This will improve the operating characteristics of the scheme. We call this variation on the PASS scheme PASS2.

The polynomial response h in PASS2 will take a somewhat different form. It is constructed using a pair of challenge polynomials (c_1, c_2) , and the check by Vinnie changes to a verification that h is short, followed by a verification that a certain combination of the values $f(\alpha), c(\alpha), g(\alpha), h(\alpha)$ are squares modulo q for all $\alpha \in S$.

In the following sections we give a precise description of PASS2, propose some specific parameters, and provide security analyses in these cases.

§1.1. An outline of the PASS2 authentication scheme

We first review some of the PASS notation. Let q be a prime and let $N = q - 1$. A typical element g of R has a representative of the form

$$g = a_0 + a_1x + a_2x^2 + \dots + a_{N-1}x^{N-1}$$

with coefficients $a_i \in \mathbf{Z}/q\mathbf{Z}$. It is useful to define two norms on R . Let g be a polynomial whose coefficients satisfy $|a_i| \leq q/2$ and $\sum_i a_i = 0$. We then define

$$|g|_2 = \sqrt{a_0^2 + \dots + a_{N-1}^2} \quad \text{and} \quad |g|_\infty = \text{Max } a_i - \text{Min } a_i.$$

We recall the notion of a short polynomial:

Definition. A polynomial f will be called "short" if its norm $|f|_2$ is smaller than a specified constant multiple of \sqrt{q} .

Very roughly, polynomials are called short if their coefficients are sufficiently small with respect to q that no reduction mod q occurs when two of them are multiplied together. We will occasionally find it useful to call a polynomial "moderately" short if its norm is less than a constant times q .

When polynomials are short, the two norms above are related by the rough inequality

$$|g * g'|_\infty \leq c_2 |g|_2 |g'|_2, \tag{2}$$

where c_2 is a constant that varies between 0.3 and 0.5 for parameters in the ranges discussed here. Also we have the approximate relation (for random choices of short polynomials)

$$|g * g'|_2 \approx |g|_2 |g'|_2. \tag{3}$$

The estimate (3) with appropriate choices of constants shows that the product of two short polynomials will have small $|\cdot|_2$ and $|\cdot|_\infty$ norms with respect to q .

For any integer d , we let $\mathcal{L}(d)$ denote the set of polynomials in R that have exactly d coefficients equal to each of 1 and -1 , with all other coefficients equal to 0. We fix a set S consisting of $t = N/2$ randomly chosen distinct non-zero elements $\alpha \in \mathbf{Z}/q\mathbf{Z}$. The set S is a system-wide parameter. For technical reasons, we assume that S is chosen so that if $\alpha \in S$, then $\alpha^{-1} \in S$, i.e., S is closed under taking inverses.

We further fix four system parameters d_f, d_g, d_c, γ . These are used to define four sets of polynomials:

$$\mathcal{L}_f = \mathcal{L}(d_f), \quad \mathcal{L}_g = \mathcal{L}(d_g), \quad \mathcal{L}_c = \mathcal{L}(d_c), \quad \mathcal{L}_h = \{h \in R : |h|_2 < \gamma q\}.$$

We now describe how an authentication session proceeds in PASS2. Pearl, the prover, has a private key f , known only to her. This private key is chosen by Pearl at random from \mathcal{L}_f . Her public key is the associated ordered collection of values mod q : $\{f(\alpha)\}_{\alpha \in S}$. We claim that the following scenario allows Pearl to prove to Vinnie, the verifier, that she possesses the secret key f associated to her public key, without revealing f or information that could help Vinnie, or a third party Irving observing the transaction, to discover f .

- Pearl randomly chooses a commitment $g_1 \in \mathcal{L}_g$ and sends the set of values $\{g_1(\alpha)\}_{\alpha \in S}$ to Vinnie.
- Vinnie chooses an 80 bit challenge c_0 at random and sends c_0 to Pearl. Pearl hashes c_0 with $\{g_1(\alpha)\}_{\alpha \in S}$ to obtain $c_1, c_2 \in \mathcal{L}_c$. Pearl checks that $c_1(\alpha) \not\equiv 0 \pmod{q}$ for all $2 \leq \alpha \leq q-2$ with $\alpha \notin S$. If this is not the case Pearl rechooses c_1 in a predefined way until c_1 has this property.
- Pearl chooses $g_2 \in \mathcal{L}_g$ and computes and reveals

$$h = (f + c_1 * g_1 + c_2 * g_2) * g_2.$$

- Vinnie verifies that:
 - (A) $h \in \mathcal{L}_h$.
 - (B) The quantity $(f(\alpha) + c_1(\alpha)g_1(\alpha))^2 + 4c_2(\alpha)h(\alpha)$ is a quadratic residue modulo q for every $\alpha \in S$.

If Pearl passes the two tests, then Vinnie accepts her claim of identity.

Remark 1. One can check that the probability that the c_1 chosen through a hashing process as above will have the desired non-vanishing property is greater than 50%. Thus it will not take long for Pearl to locate a satisfactory c_1 .

As with PASS, or any public key authentication scheme, one must verify, or at least make strong arguments in favor of, several things. First, it must be shown that if Pearl possesses the private key f , the probability that she will pass the test and be accepted by Vinnie as legitimate can be made arbitrarily high. Second, it must be shown that a potential impostor without knowledge of f or some other false key f' will have a very low probability of passing the test. Finally, it must be shown that even if an impostor knows the public key and has access to an arbitrarily long transcript of genuine authentication

transactions using f , he will have a close to zero chance of recovering either the original private key f or an equally useful false key f' .

In the following, we will generally suppress the $*$ in the notation when multiplying polynomials in R .

§1.2. Specific parameter choices

In this section we give concrete details for the PASS2 scheme described above. Let q be a small prime, for example $q = 769$ or $q = 929$, and let $N = q - 1$. We will establish below that the level of security for $q = 769$ is considerably greater than that of RSA 512, while that of $q = 929$ is greater than RSA 1024.

Let r be a primitive root modulo q , let $t = N/2$, and let J be a collection of t distinct indices j , chosen at random from the collection of integers less than N , with the condition that if $j \in J$, then $q - 1 - j \in J$. Define S by

$$S = \{r^j \bmod q : j \in J\}. \quad (4)$$

Then S consists of t distinct elements $\alpha \bmod q$. As they are non-zero, each has the property that $\alpha^N \equiv 1 \bmod q$. Also, by its definition, S is closed under the taking of multiplicative inverses mod q .

Fix t and a set S with $|S| = t$ as in (4) above. Set the parameters d_f, d_g, d_c, γ as follows:

$$d_f = [1/2 + q/3], \quad d_g = [1/2 + q/6], \quad d_c = 2, \quad \gamma = 1.8. \quad (5)$$

It is simple to check then that for any $q \geq 769$

$$|\mathcal{L}_f| > 2^{160}, \quad |\mathcal{L}_g| > 2^{160}, \quad |\mathcal{L}_c| > 2^{36}, \quad q^t > 2^{160}. \quad (6)$$

In fact these bounds are far exceeded for all spaces except for \mathcal{L}_c . Note that the space of challenges is the space of pairs of elements of \mathcal{L}_c and thus has size 2^{72} .

Let us first discuss completeness. We will show that Pearl, knowing the secret key f , can pass Vinnie's test with very high probability.

§1.3. On Completeness

Recall how the scenario works: Pearl chooses $g_1 \in \mathcal{L}_g$ and reveals the commitment $\{g_1(\alpha)\}_{\alpha \in S}$. A challenge is sent to Pearl, which she uses to create the pair $c_1, c_2 \in \mathcal{L}_c$. Pearl chooses $g_2 \in \mathcal{L}_g$, then uses her knowledge of f to compute and reveal

$$h = (f + c_1 * g_1 + c_2 * g_2) * g_2.$$

The test $h \in \mathcal{L}_h$ will be passed for the following reason. From (2) and (3), we see that the fact that $|f|_2, |g|_2, |c_1|_2$, and $|c_2|_2$ are small implies that $|h|_2$ and $|h|_\infty$ must be small. As with PASS, the probability that $|h|_2$ falls into a given range, or that individual coefficients of h fall into given ranges, can be computed theoretically, but it is far easier to do an

empirical computation. For example, in the case (5) above with $q = 769$, we found that in $5 \cdot 10^6$ tests of randomly chosen triples (f, g, c_1, c_2) from $\mathcal{L}_f, \mathcal{L}_g, \mathcal{L}_c$,

$$600250 \leq |h|_2^2 \leq 1916009$$

for all but one h , for which the value was 1972192. From this we conclude that the probability that $|h|_2 \leq 1.8q$ is roughly $2 \cdot 10^{-7}$ (and even for the exception this inequality held with 1.8 replaced by 1.83). Thus we claim that the probability of a false alarm, i.e., that Pearl will fail test (A) despite knowing the secret f , is less than 10^{-6} . If this occurs, the test can simply be repeated, and similarly with a digital signature.

Remark 2. If desired the test can be strengthened by lowering 1.8 to, say, 1.6. Then the chances of a false alarm are somewhat increased, but the security level at a given parameter setting increases dramatically.

Next consider the test

(B) $(f(\alpha) + c_1(\alpha)g_1(\alpha))^2 + 4c_2(\alpha)h(\alpha)$ is a quadratic residue mod q for every $\alpha \in S$.

This is will be true because $(f(\alpha) + c_1(\alpha)g_1(\alpha))^2 + 4c_2(\alpha)h(\alpha)$ will be a square if and only if the quadratic equation

$$c_2(\alpha)x^2 + (f(\alpha) + c_1(\alpha)g_1(\alpha))x - h(\alpha) \equiv 0 \pmod{q}$$

has a solution. But the construction of h guarantees the existence of a solution, namely $x = g_2(\alpha)$. Thus Pearl will pass this test also and her proof of identity will be accepted by Vinnie.

§1.4. Security discussion

We will now consider the chances that an imposter, Irving, can pretend to be Pearl without knowledge of the secret PASS2 key f . The first few arguments are identical to those in [3]. With the size of the spaces given in (6), the chances of Irving locating f , or an equivalently useful f' by an exhaustive search or meet-in-the-middle attack are, as in PASS, less than 2^{-80} . Since $|\mathcal{L}_c| = 2^{72}$, the chances that a repeat of a previously observed genuine session will help Irving are minimal.

In order to impersonate Pearl, Irving can either choose his h at random satisfying the quadratic constraints and hope that $|h|_2 < 1.8q$, or Irving can choose h with $|h|_2 < 1.8q$ and hope that h satisfies the quadratic constraints. In the first case, as in PASS, by using Sterling's formula to approximate the volume of an N -sphere one can check that

$$|\mathcal{L}_h| \approx (2\pi e)^{N/2} (1.8q)^N N^{-N/2}. \tag{7}$$

An h chosen to satisfy the quadratic constraints will be uniformly distributed inside a space of volume q^N . Thus by (7) we see that for our parameters, $|\mathcal{L}_h|q^{-N} < 2^{-160}$. This means that this approach will have a less than 2^{-80} chance of success, even including possible meet-in-the-middle off line attacks.

On the other hand, an h picked at random from \mathcal{L}_h will have a 50% chance of satisfying each quadratic constraint, and thus a $2^{-N/2}$ probability of satisfying all of them. For $N > 320$ this is also less than 2^{-160} .

Another potential attack for Irving is to cheat on his choice of g_1, g_2 and pick polynomials far shorter than they should be. In the most extreme case, Irving could choose g_1, g_2 to be simply x^k, x^l for some k, l . If Irving could find a false key f' with $|f'|_2 < 1.8q$ and $f'(\alpha) \equiv f(\alpha) \pmod q$ for all $\alpha \in S$, then this attack would succeed. The chances of Irving finding such an f' through a random search are covered by (7) above and are less than 2^{-80} . Keys f and f' can also be searched for via lattice reduction methods, which will be discussed below.

§1.5. Soundness

We will give a probabilistic argument here that for t a bit larger than $N/2$, if Irving can produce a sequence of responses to a single commitment $\{g_1(\alpha)\}_{\alpha \in S}$ and a sequence of challenge pairs c_1, c_2 then he must have knowledge of the secret key f . As in [3], our argument will not be airtight. But we hope it will be convincing.

Suppose that, given $\{g_1(\alpha)\}_{\alpha \in S}$, when confronted by a random challenge pair c_1, c_2 Irving can produce a moderately short polynomial h with the property that

$$(f(\alpha) + c_1(\alpha)g_1(\alpha))^2 + 4c_2(\alpha)h(\alpha)$$

is a quadratic residue mod q for every $\alpha \in S$. It may be the case that Irving does not really have short polynomials g_1, g_2 on hand but has simply selected the collection of values $\{g_1(\alpha)\}_{\alpha \in S}$ by some method. If so, the multiplication by the random c_1 and the inclusion of the random c_2 in the constraint seem to reduce Irving's situation to the general one of finding a moderately short polynomial satisfying a collection of t quadratic constraints. This problem is analyzed below in the section on lattice reduction attacks. With high probability there will exist a large number of potential responses h satisfying these constraints. However, the only method available for finding them seems to be lattice reduction methods, and the time estimates for Irving to find a response by this method are quite long.

Let us assume therefore that Irving's response actually has the form

$$h = (F + c_1g_1 + c_2g_2)g_2$$

for any challenges c_1, c_2 , with g_1, g_2 fixed, but not necessarily short. We also assume that F has the correct values at S but is not necessarily short. Then by taking the two responses H and H' to challenge pairs c_1, c_2 and $c_1, 1 + c_2$, Irving can obtain the difference $H' - H = g_2^2$. Unless Irving has solved the problem previously mentioned, of finding general short polynomials whose values are quadratic residues, it is highly probable that $H' - H = g_2^2$ is a square of a short polynomial, i.e. that g_2 really must be short. The square root can then be taken, as described in [3], recovering g_2 . The short polynomial $c_2g_2^2$ can then be subtracted from h , yielding a short polynomial $Fg_2 + c_1g_1g_2$. Performing this operation with c_1 and $1 + c_1$ (while still keeping g_1, g_2 fixed) Irving could obtain the short polynomial Fg_2 . This now has the same values at S as fg_2 , an actual product of short polynomials. If we knew that $Fg_2 = fg_2$ then Irving could divide by g_2 , recovering f . This however, can be seen to be true with high probability, by using the gaussian heuristic as follows.

The difference of the polynomials $H = Fg_2 - fg_2$ satisfies $H(\alpha) \equiv 0 \pmod{q}$ for all $\alpha \in S$. H is also moderately short, meaning that $|H|_2 < Kq$ for an absolute constant q . The probable existence of a non-zero H satisfying these constraints for large N can be calculated by approximating the volume of an N -sphere using Sterlings formula, as in [3], and applying the gaussian heuristic to a lattice of determinant q^t , described in the next section. One sees that for N large the expected number of such polynomials is on the order of

$$(2\pi e)^{N/2} N^{-N/2} K^N q^N q^{-t}.$$

If $t = N/2 + \epsilon N$ for some small $\epsilon > 0$, then this quantity approaches zero for large N , meaning that with high probability $H = 0$ and $Fg_2 = fg_2$.

§2. Lattice reduction techniques

Lattice reduction methods can be used by Irving to search for the private key f , or an equally useful false key f' . These methods can also be used in an off line attempt to construct a valid response h to a given challenge. Finally, they can be used in an attempt to recover g_1 from a given commitment and hence f from the corresponding response h . (In fact about 15 different g_1 recoveries would be necessary to recover f .) In this section we will discuss and quantify the difficulty of these questions. First we will discuss an attack on f using the public key $\{f(\alpha)\}_{\alpha \in S}$

§2.1. Formulation of a lattice attack on the public key.

This is approached exactly as in [3]. For convenience we will remind the reader of the outline. We begin by constructing a lattice as follows. For any polynomial $F \in R$, associate to F the vector of coefficients $(a_0, a_1, \dots, a_{N-1})$. Similarly for any such vector or point in \mathbf{Z}^N , one can take the polynomial built from these coefficients, reduce mod q , and obtain an $F \in R$.

Let L be the lattice of all points in \mathbf{Z}^N such the corresponding polynomial F satisfies

$$F(\alpha) \equiv 0 \pmod{q} \text{ for each } \alpha \in S.$$

It is easy to check that L is indeed a lattice, and that the determinant of L is equal to q^t .

It is not difficult to find a polynomial $F' \in R$ such that $F'(\alpha) \equiv f(\alpha) \pmod{q}$ for all $\alpha \in S$. However it is very unlikely that such an F' will have small coefficients. Suppose, instead, that we find an F' with non-small coefficients and then search for a point $F \in L$ close to F' . If such an F is found, set $f' = F' - F$. Then f' will still have the correct valuations at $\alpha \pmod{q}$, and if F is very close to F' , then $|f'|_2$ will be small.

The problem of finding an f' which will give a good impersonation of f is thus reduced to that of finding a point in a lattice which is as close as possible to a given point outside the lattice. This is a non-homogeneous version of the problem of finding a short vector in a lattice. It can also be translated into a homogeneous problem in a similar lattice of one higher dimension. Roughly speaking, an attacker's chance of success in a fixed amount of time improves as the distance of the given point to the lattice decreases. The attacker's chances also deteriorate as the dimension of the lattice increases.

§2.2. Some lattice reduction experiments

Consider a list of primes q and $N = q - 1$ with $d_f = \lceil 1/2 + q/3 \rceil$ as in (5). When $q = 769$, this gives $d_f = 256$. Our experiments used the lattice reduction package provided in version 3.1b of Victor Shoup's implementation of the Schnorr, Euchner and Hoerner improvements of the LLL algorithm. This is distributed in his NTL package, located at <http://www.cs.wisc.edu/~shoup/ntl/>. Our approach was to obtain results for an increasing sequence of primes q , and $N = q - 1$, and plot the log of the time it took to break a key or find an alternative key against N . We found in all cases that the log time increased linearly with N . We then extrapolated the line we obtained to obtain estimated breaking times for high N .

Table 1 gives the results of experiments to recover the private key f from $\{f(\alpha)\}_{\alpha \in S}$.

q	N	T_{avg}	Experimental Times T
101	100	171	149, 170, 193
109	108	226	205, 221, 253
113	112	366	317, 347, 433
149	148	3787	3445, 3912, 4005
157	156	7443	3522, 11363
173	172	32760	17437, 48082
181	180	167021	167021

Table 1. Time (secs) To Find Original Key f

The regression line for the average time (in seconds), as a function of N , is

$$\log(T) \approx 0.0803N - 3.1923.$$

The correlation coefficient is 0.9866. We have used the regression line to extrapolate the breaking time for larger values of N . The results are listed in Table 2. Note that the conversion factor from seconds to MIPS-years is 400/31557600, because our experiments were run on 400 MHz Celeron computers.

N	T (seconds)	T (MIPS-years)
640	$8.65 \cdot 10^{20}$	$1.10 \cdot 10^{16}$
768	$2.52 \cdot 10^{25}$	$3.20 \cdot 10^{20}$
928	$9.60 \cdot 10^{30}$	$1.22 \cdot 10^{26}$
1152	$6.24 \cdot 10^{38}$	$7.92 \cdot 10^{33}$

Table 2. Estimated Breaking Times For Original f Key

Now consider a list of primes q and $N = q - 1$ with $d_g = [1/2 + q/6]$ as in (5). When $q = 769$, this gives $d_g = 128$. Table 3 gives the results of experiments to recover g_1 from $\{g_1(\alpha)\}_{\alpha \in S}$. Note that an attempt could be made to recover g_2 from values given by the solution of the quadratic equation involving f, g_1, c_1, c_2 that g_2 satisfies. However, as the values of f and g_1 are only known in S and each $g_2(\alpha)$ has two possible solutions, this procedure is far more difficult than the problem of recovering g_1 .

q	N	T_{avg}	Experimental Times T
101	100	69	89, 66, 61, 61, 70
109	108	102	95, 103, 100, 118, 95
113	112	108	103, 136, 95, 114, 94
137	136	471	639, 419, 344, 500, 455
149	148	802	689, 773, 1034, 504, 1008
157	156	1712	1743, 2293, 1468, 1976, 1080
173	172	2983	2465, 2598, 1948, 2860, 5045
181	180	6836	4219, 11951, 7541, 3940, 6529
193	192	13227	22919, 4640, 8982, 22598, 6996
197	196	15169	15744, 28535, 20060, 1612, 9892

Table 3. Time (secs) To Recover g_1

The regression line for the average time (in seconds), as a function of N , is

$$\log(T) \approx 0.0574N - 1.6850.$$

The correlation coefficient is 0.9978. We have used the regression line to extrapolate the breaking time for larger values of N . The results are listed in Table 4.

N	T (seconds)	T (MIPS-years)
640	$1.64 \cdot 10^{15}$	$2.08 \cdot 10^{10}$
768	$2.54 \cdot 10^{18}$	$3.22 \cdot 10^{13}$
928	$2.47 \cdot 10^{22}$	$3.13 \cdot 10^{17}$
1152	$9.42 \cdot 10^{27}$	$1.19 \cdot 10^{23}$

Table 4. Estimated Breaking Times For g_1 Recovery

Table 5 gives the time required to produce an f' with the property that $f' \equiv f \pmod{q}$ for all $\alpha \in S$ and $|f'|_2 < 1.8q$. Such an f' would not equal the original f , but would be sufficient, once discovered, for Irving to have a reasonably good chance of impersonating

q	N	T_{avg}	Experimental Times T
193	192	281	268, 231, 446, 225, 234
197	196	395	284, 326, 287, 507, 573
229	228	956	671, 890, 1602, 989, 630
233	232	1423	1882, 1611, 1056, 1516, 1051
241	240	2369	3620, 1890, 1961, 2177, 2198
257	256	4021	4038, 6990, 2648, 3024, 3405
269	268	7297	10921, 4023, 8235, 8638, 4670
271	270	9949	11562, 7340, 5865, 18289, 6687
277	276	11415	12775, 10393, 20110, 5319, 8477
281	280	15578	24864, 18970, 11842, 9572, 12642
283	282	22022	15506, 14053, 22891, 41195, 16464
293	292	25239	17518, 39748, 18452
307	306	110973	107573, 114373

Table 5. Time (secs) To Find False Key f'

Pearl. To do this he would cheat on his commitment by choosing g_1, g_2 to be simple powers of x . We give the results of several experiments for each q between 193 and 307, together with the average time required for each q .

The regression line for the average time (in seconds), as a function of N , is

$$\log(T) \approx 0.0487N - 3.9606.$$

The correlation coefficient is 0.9876. We have used the regression line to extrapolate the breaking time for larger values of N . The results are listed in Table 6.

N	T (seconds)	T (MIPS-years)
640	$6.46 \cdot 10^{11}$	$8.19 \cdot 10^6$
768	$3.28 \cdot 10^{14}$	$4.16 \cdot 10^9$
928	$7.92 \cdot 10^{17}$	$1.00 \cdot 10^{13}$
1152	$4.31 \cdot 10^{22}$	$5.46 \cdot 10^{17}$

Table 6. Estimated Breaking Times for false PASS2 key f'

Remark 3. Table 6, the estimated time for recovery of a false key f' , gives the smallest breaking times, hence should be regarded as providing a lower bound for the security of the PASS2 scheme.

For comparison purposes, we note that the estimated time to break RSA 512 is $3 \cdot 10^4$ MIPS-years, and the estimated time to break RSA 1024 is $3 \cdot 10^{11}$ MIPS-years. So according to Table 6, the the PASS2 scheme with $N = 640$ should be considerably more secure than RSA 512, while for $N = 928$ security is greater than RSA 1024 and $N = 728$ lies in between.

§2.3. Zero-Forced Lattices

Alexander May [4] has given an improved method for searching for small vectors when the small vectors have a comparatively large number of coordinates equal to 0. These ideas lead to the notion of zero-forced lattices, in which one guesses that r particular coordinates of the target are 0, forces them to be zero, and thereby reduces the dimension of the lattice. Of course, if r is large, it may take many tries before one makes a correct guess. Full details of how zero-forced lattices work and how to estimate their effectiveness are given in [5]. However, since the polynomials have only 1/3 of their coefficients equal to 0, in the case of f , and 2/3 equal to 0, in the case of g_1 it is very difficult to correctly guess many zeros. As it would be necessary to guess considerably more than 100 zero locations correctly in order to reduce the key breaking time for g_1 or f down to even the time estimate for finding a false f' , one sees that the use of zero-forced lattices has a negligible effect on security estimates for PASS2.

§2.4. Lattice based creation of a response without the private key

Irving faces the following problem. Given a challenge c , he must find a polynomial h with $|h|_2 < 1.8q$ such that $(f(\alpha) + c_1(\alpha)g_1(\alpha))^2 + 4c_2(\alpha)h(\alpha)$ is a quadratic residue mod q for every $\alpha \in S$. There are several different approaches that Irving can take, but none seem to have any chance of success in time less than that estimated in Table 6 for recovery of a false key f' .

For example, Irving could choose his collection of commitment values $g_1(\alpha)$ at random. After receiving the challenges Irving could choose t values for $h(\alpha)$ at random that satisfy the quadratic constraints. He could then use LLL to search for h with $|h|_2 < 1.8q$ satisfying these constraints. For $t = N/2$ the expected size of a vector satisfying these constraints is (see [3]) about $q/\sqrt{2\pi e}$. As this is less than $1.8q$, there is a high probability that such an h exists. However the time required to find such an h by lattice reduction methods is greater than or equal to the time required to find a false key f' as given in Table 6. Thus the security estimate for PASS2 remains unchanged after considering this potential attack.

Another possibility for Irving is to pick a random collection of commitment values $g_1(\alpha)$. He can then choose G_1 very short, (even a power of x), and define $g_2(\alpha)$ by $G_1(\alpha) = g_2(\alpha)f(\alpha)$. Then he can search for a short G_2 such that $|G_2|_2 < 1.8q$ and such that $G_2(\alpha) = (c_1(\alpha)g_1(\alpha) + c_2(\alpha)g_2(\alpha))g_2(\alpha)$ for every $\alpha \in S$. This, however, reduces to the same search as just mentioned, and should be solvable in about the same time.

The last possibility we will consider is that Irving could find relatively short polynomials G_1, G_2, G_3 , i.e., polynomials satisfying $|G_1|_2, |G_2|_2, |G_3|_2 < 1.8q$, and collections of values $\{g_1(\alpha), g_2(\alpha)\}_{\alpha \in S}$ such that

$$G_1(\alpha) = f(\alpha)g_2(\alpha), \quad G_2(\alpha) = g_1(\alpha)g_2(\alpha), \quad G_3(\alpha) = g_2(\alpha)^2.$$

This problem seems to be just as hard as that mentioned in the first two possibilities, but the dimension is tripled, leading to considerably greater breaking times.

§2.5. Attacks on a transcript of authentication sessions.

Consider the information revealed in a large collection of distinct examples of

$$h = (f + c_1g_1 + c_2g_2)g_2$$

for fixed f and varying c_1, c_2 and g_1, g_2 . It is important to note that since f, g_1, g_2, c_1, c_2 are small, an attacker may assume that no reduction modulo q has occurred in the construction of h , and thus that the coefficients of h are given over \mathbf{Z} . Significant reduction, however, has occurred modulo $x^N - 1$.

First, fix some β not in S and let us consider the information revealed from a collection of responses h for which c_1 vanishes at β , i.e., $c_1(\beta) \equiv 0 \pmod{q}$. Let QR denote the set of quadratic residues mod q . Since $g_2(\beta)$ is the solution of a quadratic equation, it must be true that

$$(f(\beta) + c_1(\beta)g_1(\beta))^2 + 4c_2(\beta)h(\beta) \in QR.$$

Since we are assuming that $c_1(\beta)$ vanishes, it follows that $f(\beta)^2 + 4c_2(\beta)h(\beta) \in QR$. This constrains $f(\beta)^2$ to lie in the translated set

$$f(\beta)^2 \in QR - 4c_2(\beta)h(\beta) = \{u^2 - 4c_2(\beta)h(\beta) : u \pmod{q}\}.$$

Each response h for which $c_1(\beta) = 0$ will cut the possibilities for $f(\beta)^2$ by approximately 50%, so after little more than $\log_2 q$ such responses, an attacker can determine $f(\beta)^2$.

If this attack is carried out for every β not in S , then the polynomial $f(x)^2$ can be determined, since the values $f(\alpha)$ for $\alpha \in S$ are already public knowledge. It is then easy to extract the small square root and recover $f(x)$ itself, see [3] for details. The attack we have just described is the reason for the requirement in the protocol that $c_1(\beta) \neq 0$ for all $\beta \notin S$ (other than $\beta = 0, \pm 1$, which are not important). This requirement means that the above attack cannot even get started. We are indebted to Don Coppersmith for informing us of this potential attack.

One might ask if the attacker could apply the same approach using an irreducible quadratic factor of c_1 and thus a root of c_1 in a quadratic extension of $\mathbf{Z}/q\mathbf{Z}$. This will not work, because the polynomial h is only given modulo $x^N - 1$, and it is only elements of $\mathbf{Z}/q\mathbf{Z}$ that have the property that $\alpha^N = 1$; elements in extension fields do not have this property. In other words, the evaluation map at a element of an extension field is not a homomorphism from R to that extension field, so the attack using extension fields is not possible.

The collection of all h in a transcript will generate a lattice over \mathbf{Z} . However, because of the presence of the non-zero c_i , the full (and thus useless) lattice is generated by this collection.

One can consider the average of many different responses h . As in [3] this does not provide useful information because the expected values of the coefficients of g_1, g_2 are 0.

The expected value, incidentally, of the polynomial g_2^2 is $(x^N - 1)/(x^2 - 1)$. This is not quite zero, but highly non-invertible.

An attacker might also consider the product $h\sigma(h)$, the autocorrelation polynomial corresponding to h . This is potentially a very powerful attack, due to Burt Kaliski, as after averaging a long transcript, an attacker can hope to obtain the polynomial

$$a_f A_g^{(2)} + A_c^{(1)} A_g^{(1,2)} + A_c^{(2)} A_g^{(2,2)}.$$

Here for any polynomial F , a_F denotes the even autocorrelation polynomial $a_F = F * \sigma(F)$. Also $A_c^{(i)}$ denotes the expected value of $c_i * \sigma c_i$ for $i = 1, 2$ and $A_g^{(i,j)}$ denotes the expected value of $g_i g_j * \sigma(g_i g_j)$. The average of $h\sigma(h)$ will approach this limit as the cross terms of the product will have expected value zero.

If g_1, g_2, c_1, c_2 vary uniformly, the limiting autocorrelation polynomials are simple constants and hence a_f can be recovered. This means in effect that $f(\alpha)f(\alpha^{-1})$ can be assumed to be known, and thus that once $f(\alpha)$ is known mod q for any α , $f(\alpha^{-1})$ can be found. This is the reason for the original assumption that S is closed under multiplicative inverses, as an attacker gets no additional knowledge from a_f . We refer also to the analysis given in [3] for the conclusion that it is very difficult to factor a_f as a polynomial and obtain f .

We will close this section by remarking briefly on an important observation of Copersmith. By selecting any fixed 4-tuple of indices i, j, k, l and computing an average of the product of the i, j, k, l individual coefficients h_i, h_j, h_k, h_l , information can be obtained about a combination of second and fourth power moments of f . (In this terminology, a_f is the second power moment of f .) It is then possible to recover f by a process which, while computationally intensive, is still subexponential in N and feasible for the parameter choice $N = 768$. We have conducted a number of computer experiments to determine lower bounds that the length of a transcript must exceed before an attacker has a chance of determining the limiting value of the products h_i, h_j, h_k, h_l . Some experimental evidence is given in the Appendix 2. The experiments show that the convergence to the limiting value is extremely slow. Even after averaging 100 million responses, i.e., examining 100 million digital signatures produced by a single private key, the variation in each product is still wide enough to allow considerably greater than 2^{160} choices for a sufficiently large (greater than N) limiting collection of 4-tuple products. We thus feel that it is safe to use a single key for at least 100 million authentication sessions or digital signatures.

§2.6. Cheating Verifiers

A cheating verifier can pass specially constructed challenges with given expected values to Pearl and extract information from the responses as outlined above. (For example, choosing challenges equal to 0, or those where c_1 has roots consistently in specific places.) In this scheme, however, a challenge c_0 is hashed with the commitment. This seems to eliminate any chance of a cheating verifier obtaining an advantage.

§3. Key length and communication requirements

The key lengths and number of bits transmitted for $N = 768$ and $N = 928$ are given in Table 7. (For $N = 928, d_f = 62$.) It is worth noting that if desired, as in the PASS scheme, the private key can be stored as, or generated from, any random string of 80 bits, as long as a non-linear uniform mapping is provided into the space \mathcal{L}_f . The number of bits in the response is an upper bound, based on the fact that most coefficients of h will have a rather small absolute value and hence can be recorded using 5,6 or 7 bits. On average, one finds that with these parameter choices, about 34% of the coefficients can be recorded with 5 bits, 29.14% with 6 bits, 29.64% with 7 bits. Only about 0.03% will require 8 bits and one or two rare exceptions require 9. Note that the length of a digital signature attached to a message will be the total number of bits transmitted as recorded below, minus the 80 bits required for the challenge. This is because, as usual when constructing a digital signature, the message is hashed with the commitment to produce the challenge. The signature is then the commitment, followed by the response.

		N=768	N=928
Public key	$t \log_2 q$	3840	4640
Private key	$2d_f \log_2 N$	1020	1240
Commitment	$t \log_2 q$	3840	4640
Challenge	$\log_2 c_0$	80	80
Response	$\approx N \log_2 h _\infty$	4280	5170
Total Bits Transmitted		8,200	9,890

Table 7. Key Length and Communication Requirements in Bits

§4. Final Remarks

Recall that we established above that the security level of PASS2 with $q = 769$ is considerably greater than that of RSA 512, while the security level of PASS2 with $q = 929$ is greater to RSA 1024.

When Vinnie checks that the quadratic condition is fulfilled, he need only do this for a randomly chosen subset of 80 values in S . It will probably be most efficient for Vinnie to use a precomputed table of quadratic residues mod q , but if space is at a premium, then quadratic reciprocity could be used for this test.

Finally, we remark that the evaluation of polynomials by Pearl and Vinnie can be done most efficiently by means of the FFT. This is because the evaluation of a polynomial is simply the association between a vector over $\mathbf{Z}/q\mathbf{Z}$ and its discrete Fourier transform, where a polynomial is identified with the vector of its coefficients. Naive computation of discrete Fourier transforms of vectors of dimension N only takes N^2 steps, so is not an onerous task. However, the suggested parameter values were selected so that N is divisible by a reasonably large value of 2, which means that one can use Fast Fourier Transforms

(FFT) to speed the process. Note that one can do these FFT's in $\mathbf{Z}/q\mathbf{Z}$ working entirely with integers, because $\mathbf{Z}/q\mathbf{Z}$ contains a primitive N^{th} root of unity. There is no need to use real or complex numbers.

References

- [1] J. Hoffstein, B.S. Kaliski, D. Lieman, M.J.B. Robshaw, Y.L. Yin, "A New Identification Scheme Based on Polynomial Evaluation," *patent application*.
- [2] J. Hoffstein, J. Pipher, J. Silverman, "NTRU: A ring-based public key system," Proceedings of ANTS III, Portland (1998), Springer-Verlag.
- [3] J. Hoffstein, D. Lieman, J. Silverman, "Polynomial Rings and Efficient Public Key Authentication," *Proceeding of the International Workshop on Cryptographic Techniques and E-Commerce (CrypTEC '99)*, M. Blum and C.H. Lee, eds., City University of Hong Kong Press, to appear.
- [4] A. May, Cryptanalysis of NTRU, preprint, February 1999
- [5] J.H. Silverman, Dimension-Reduced Lattices, Zero-Forced Lattices, and the NTRU Public Key Cryptosystem, NTRU Technical Note 013, March 2, 1999, (www.ntru.com)

Appendix 1. Timing Comparisons

In this section we compare digital signature and verification times for various cryptosystems. We note that the PASS2 times are based on a preliminary non-optimized implementation by Tao Group, Inc. We also note that the extremely fast RSA verification times are due to the use of the very small value $k = 17$ as decryption exponent.

	Sign	Verify
PASS2 768	4.73	2.12
RSA 512	4.34	0.27
DSA 512	4.75	5.56
PASS2 928	10.69	4.25
RSA 1024	27.62	0.70
DSA 1024	15.38	18.16
ECC(p) 168	15.99	27.42
ECC(2^n) 155	20.12	24.51
PASS2 1152	8.31	3.73
RSA 2048	181.82	2.11

Table 8. Timing Estimates (Milliseconds Per Operation)

The timing data for the RSA, DSA, and ECC signature schemes in Table 8 are taken from the Crypto++ 3.1 Benchmarks page, which may be found at

<http://www.eskimo.com/~weidai/benchmarks.html>.

All were coded in C++ or ported to C++ from C implementations, compiled with Microsoft Visual C++ 6.0 SP2 (optimized for speed, Pentium Pro code generation), and run on

a Celeron 450MHz machine under Windows 2000 beta 3. No assembly language was used. The RSA computations were done using the small verification exponent 17. The DSA and ECC values can be improved somewhat (up to a factor of 2 in some cases) by storing precomputed values. The PASS2 times are for the preliminary implementation by Tao Group (run on a 300MHz machine and extrapolated to 450MHz). The reason that PASS2 1152 is faster than PASS2 928 is because 1152 is more highly divisible by 2 than is 928, which allows greater efficiency in the FFT routines.

Appendix 2. Transcript Experiments

We fixed PASS2 parameters

$$N = 768, \quad d_f = 256, \quad d_g = 128, \quad d_c = 2,$$

For each experiment we fixed a random polynomial f , and four random indices i, j, k, l . We randomly chose 100 million 4-tuples of polynomials g_1, g_2, c_1, c_2 according to these parameters. For each of these choices we computed

$$h = (f + g_1 * c_1 + g_2 * c_2) * g_2.$$

We then took the four random indices and computed the product

$$h_i h_j h_k h_l$$

of the corresponding four coefficients of h . We kept a running average of these quadruple products. Results from a typical experiment are given in Table 9. Other experiments gave similar behavior, so we have selected a few pieces of one run to give a feel for the rate of convergence. In this table, the four indices fixed were 55, 105, 537, and 551 and we have recorded the running average, denoted Avg_h , rounded to the nearest integer, for various numbers of trials. As is clear from the table, even after 10^8 trials, the value of the product has not fully settled down, so it would be difficult to guess the correct value. Note that even if the value of each quadruple product is known to within 2 or 3, say, the number of possible values for all of the products $h_i h_j h_k h_l$ would be far greater than 2^{768} , so it would not be possible to perform an exhaustive search.

# of Trials	Avg _h	# of Trials	Avg _h
29,990,000	-98	99,900,000	49
30,030,000	-108	99,905,000	49
30,070,000	-88	99,910,000	49
30,110,000	-78	99,915,000	49
30,150,000	-74	99,920,000	50
30,190,000	-63	99,925,000	50
30,230,000	-62	99,930,000	52
44,180,000	-166	99,935,000	52
44,220,000	-162	99,940,000	50
44,260,000	-164	99,945,000	51
44,300,000	-166	99,950,000	52
44,340,000	-156	99,955,000	50
44,370,000	-156	99,960,000	49
89,790,000	21	99,965,000	49
89,830,000	26	99,970,000	50
89,870,000	33	99,975,000	51
89,910,000	32	99,980,000	50
89,950,000	34	99,985,000	49
89,990,000	33	99,990,000	47
90,000,000	32	99,995,000	48
		100,000,000	48

Table 9. Average Values of Products $h_i h_j h_k h_l$