An Exposition of Pollard's Algorithm for Quadratic Congruences

Jeffrey Shallit
Department of Computer Science
University of Chicago
Chicago, IL 60637

I. Introduction.

Recently, Pollard broke a signature scheme suggested by Ong, Schnorr, and Shamir [OSS]. His interesting method involved finding a solution (x, y) to the congruence

$$x^2 - Dy^2 \equiv k \pmod{N}, \tag{1}$$

where N is a composite number with unknown factorization.

I first heard of Pollard's algorithm during an informal talk by Adi Shamir in late April, 1984. At the time, the method seemed very mysterious to me, and I didn't see where the motivation came from. Also, there seemed to be several points whose justification was skimpy (due, no doubt, to the informality of the presentation.) I have spent some time filling in the missing details.

Pollard's algorithm for solving (1) has yet to appear in print. Since it contains several clever ideas, at least one of which can be used to solve other problems [Sha2], it seemed worthwhile to given a reasonably detailed exposition, and an analysis of the expected running time. I do this below.

I have made some small modifications to the algorithm as presented by Shamir, so if there are any mistakes, they are my fault.

II. Sketch of the algorithm.

Equation (1) above, though it doesn't look it, really is the most general second degree equation in two variables. This is well-known to people who play with quadratic forms; here's how to see it:

Take the general second degree equation

$$ax^2 + bxy + cy^2 + dx + ey + f = 0; \tag{2}$$

we want to rewrite it in the form

$$u^2 - Dt^2 = B. \tag{3}$$

First, consider (2) as an equation in x and complete the square. We find

$$(2ax + by + d)^2 = (by + d)^2 - 4a(cy^2 + ey + f)$$

Now expand the right side and define $D = b^2 - 4ac$ (the ubiquitous discriminant!); $g = bd - 2ae$, and $h = d^2 - 4af$. If we now write

$$t = 2ax + by + d$$

we have

$$t^2 = Dy^2 + 2gy + h.$$

Now multiply this last equation through by D and complete the square to get

$$(Dy + g)^2 = Dt^2 + g^2 - Dh$$

Now if we put $u = Dy + g$ and $B = g^2 - Dh$, we are done.

Note that solutions to (3) provide solutions to (2), since

$$y = \frac{u - g}{D}; \quad x = \frac{t - by - d}{2a}.$$

For example, see [Leg].

Eric Bach notes that this result is a special case of the well-known theorem from algebraic geometry, which states that every rational curve of genus 0 is isomorphic; i. e., there exists a 1-1 polynomial map taking you from one to the other.

The next thing to note is that

$$x^2 - Dy^2 \equiv k \pmod{p}, \tag{4}$$

$k \neq 0$, can be solved in random polynomial time if p is a prime. The basic idea is that the values of $x^2 - Dy^2$ are in fact *equally distributed* among the $p - 1$ nonzero residue classes (mod p). Thus for any k there must be roughly p pairs (x, y) that give a solution to (4). On the other hand, there are no more than two solutions x to (4) for any particular value

of y; hence (4) must be solvable for about $p/2$ different values of y. Thus we can choose y at random and then try to solve

$$x^2 \equiv k + Dy^2 \pmod{p}$$

using any random polynomial time algorithm for square roots (mod p), for example, the one in [AMM]. We succeed about half the time, so this algorithm runs in random polynomial time. All the details are given in [Sha].

Once we have a solution to (4), we can "lift" it to a solution (mod p^e) using Hensel's method, or a quadratic lifting procedure.

(It would be nice if we could generalize this method to work in the case where $p = N$ is composite. Unfortunately, it requires extraction of a square root (mod N), and this is as hard as factoring N.)

Thus we see that it is easy to solve

$$x^2 - Dy^2 \equiv k \pmod{N} \tag{5}$$

if we know the factorization of N, since we can solve the congruence modulo each prime power and then combine the results using the Chinese remainder theorem (CRT).

But how do we solve (5) if we don't know the factorization of N? There are four basic ideas (proofs will be given in section III):

Lemma 1.

The product of two numbers, each of the form $x^2 - Dy^2$, can also be written in that form...and the same thing holds for the quotient.

Lemma 2.

The substitution $u = xy^{-1}; v = y^{-1}$ transforms an equation of the form

$$x^2 - Dy^2 \equiv k \pmod{N}$$

to one of the form

$$u^2 - kv^2 \equiv D \pmod{N}.$$

Phrased in another way, the roles of k and D are interchangeable.

Lemma 3.

Given a solution r to the congruence

$$x^2 \equiv D \pmod{m},$$

we can find (in polynomial time) integers u and v such that

$$u^2 - Dv^2 = \lambda m$$

where $\lambda \leq \frac{1}{4} + \sqrt{\frac{4|D|}{3}}$.

Lemma 4.

It is easy to find solutions to

$$x^2 \pm y^2 \equiv k \pmod{N}.$$

Given these basic ideas, Pollard's algorithm may be stated as follows:

P1. Find, by probabilistic prime tests, a small prime $p \equiv k \pmod{N}$ such that D is a quadratic residue of p.

P2. Solve $x^2 - D \equiv 0 \pmod{p}$ using any random polynomial time square-root algorithm.

P3. Use Lemma 3 to find u, v such that $u^2 - Dv^2 = \lambda p$. Suppose we could solve

$$w^2 - Dz^2 \equiv \lambda \pmod{N}.$$

Then we would have

$$(u^2 - Dv^2)(w^2 - Dz^2)^{-1} \equiv p \equiv k \pmod{N}$$

and by Lemma 1, the left side is of the proper form. Then we'd be done!

P4. But how do we solve

$$w^2 - Dz^2 \equiv \lambda \pmod{N} \ ?$$

That's easy: we use Lemma 2 to interchange the roles of D and λ. Now we have a congruence with much smaller coefficients; we solve it (recursively) and interchange D and λ again. Our only worry is that we might not be able to solve the congruence at the "bottom" of the recursion– but Lemma 4 says we can!

Pollard's terrific idea is step 1–the step that gets us started. And in fact, this is the step that's the hardest to analyze.

In the next section, we sketch the proofs of these four lemmas.

III. Proofs of the lemmas.

Proof of Lemma 1.

To prove the product rule, we have

$$(x^2 - Dy^2)(w^2 - Dz^2) = (xw + Dyz)^2 - D(yw + xz)^2. \tag{6}$$

Of course, this can be verified just by multiplying out both sides, but it would be nice to see where the motivation comes from. In fact, it comes from considering expressions of the form $a = x + y\sqrt{D}$. Define the *conjugate* of a, $\bar{a}$, by $\bar{a} = x - y\sqrt{D}$. Then put

$$N(a) = a\bar{a} = x^2 - Dy^2.$$

Then equation (6) is just the statement that the norm is multiplicative, i. e.

$$N(x + y\sqrt{D})N(w + z\sqrt{D}) = N((x + y\sqrt{D})(w + z\sqrt{D})).$$

To prove the quotient result, it suffices to show that the inverse of $x^2 - Dy^2$ can be written in the same form. In stating Lemma 1, we deliberately glossed over the question of what sort of structure we are working with (i.e. ring, field, or what?). In any event, if $x^2 - Dy^2$ is invertible, and $a = (x^2 - Dy^2)^{-1}$, then by taking norms of

$$(x + y\sqrt{D})^{-1} = xa - ya\sqrt{D}$$

we find

$$(x^2 - Dy^2)^{-1} = (xa)^2 - D(ya)^2. \blacksquare$$

Proof of Lemma 2.

Left to the reader. ■

Proof of Lemma 3.

Lemma 3 is very interesting; it relates congruences (mod m) to the solution of equations in *integers*. Uspensky and Heaslet, in their 1939 book [UH], attribute it to Lagrange. Indeed, most of the ideas in Lemma 3 can be found in the 1769 paper of Lagrange [Lag].

The basic idea is sort of a successive reduction of the modulus m. Let us put $x_0 = r$ and $m_0 = m$; then we have

$$x_0^2 - D \equiv 0 \pmod{m_0};$$

hence

$$x_0^2 - D = m_1 m_0$$

for some integer m_1. Now reduce $x_0 \pmod{m_1}$ and call the result x_1. (In fact, let x_1 be the *absolutely least residue*; thus x_1 can be negative, and we have $|x_1| < \frac{1}{2}|m_1|$.) It is easily verified that

$$x_1^2 - D \equiv 0 \pmod{m_1};$$

so again we have

$$x_1^2 - D = m_2 m_1$$

for some integer m_2.

Continuing in this fashion, we get an initially decreasing sequence $|m_i|$. But notice that if $|m_i|$ is small compared to $|D|$, then $|m_{i+1}| > |m_i|$. Hence there is a limit to this decreasing behavior. In fact, since

$$m_{i+1} = \frac{x_i^2 - D}{m_i}$$

we see that

$$|m_{i+1}| \leq \frac{|m_i|}{4} + \frac{|D|}{|m_i|}. \tag{7}$$

From this, it easily follows that the $|m_i|$ decrease as long as $|m_i| > \sqrt{\frac{4}{3}|D|}$.

Thus eventually we find

$$x_n^2 - D = m_{n+1} m_n$$

with $m_{n+1} \leq \sqrt{\frac{4}{3}|D|}$. Now put $\lambda = m_{n+1}$ and define a sequence A_i such that

$$(A_i x_i - A_{i+1} m_i)^2 - D A_i^2 = \lambda m_i. \tag{8}$$

Clearly $A_n = 1$ and $A_{n+1} = 0$; it can be verified that the recurrence

$$A_{i-1} = A_{i+1} + \frac{x_{i-1} - x_i}{m_i} A_i$$

(continued fractions!) solves (8). For the grubby details, see [UH].

(We will see in section VI that it is possible to dispense with the sequence A_i.)

It remains to see that this reduction procedure runs quickly. From equation (7) we see that as long as $|m_i| > \sqrt{\frac{4}{3}|D|}$ we have

$$|m_{i+1}| \leq \frac{|m_i|}{4} + \sqrt{\frac{3}{4}|D|}.$$

It is easy to prove by induction that

$$|m_{i+n}| \leq \frac{|m_i|}{4^n} + \frac{4}{3}\sqrt{\frac{3}{4}|D|} = \frac{|m_i|}{4^n} + \sqrt{\frac{4}{3}|D|}.$$

Now set $i = 0$ and choose $n = 1 + \log_4 m_0$ and we see that $|m_n| < \frac{1}{4} + \sqrt{\frac{4}{3}|D|}$. ■

Proof of Lemma 4.

First, consider the congruence

$$x^2 - y^2 \equiv k \pmod{N}.$$

We may assume N is odd, for otherwise we can solve the equation for 2^c and $N/2^c$ and put the results together using the Chinese remainder theorem, as discussed above in section II.

Let r equal k or $k + N$, whichever is odd. Then it is easy to see

$$\left(\frac{r+1}{2}\right)^2 - \left(\frac{r-1}{2}\right)^2 = r \equiv k \pmod{N}.$$

The congruence

$$x^2 + y^2 \equiv k \pmod{N} \tag{9}$$

is just a little harder. One way is to examine k, $k + N$, $k + 2N \ldots$ successively until you find a prime $p = k + iN$ of the form $4j + 1$. Then

$$x^2 + y^2 = p$$

can be solved using the random polynomial time algorithm in [Sha]. The resulting pair (x, y) is clearly also a solution to (9). Unfortunately, it seems difficult to prove that this method runs in polynomial time. A different method, which can be given a rigorous proof of running time, is described in [Sha]. ■

IV. Formal Statement of the Algorithm & Running Time.

function Pollard(D, k, N) **returns**(x, y);

reduce D and k to their absolutely least residues (mod N);

if $D = 0$ or $k = 0$ **then** output "Cannot solve!" and stop;

else if N is even **then** write $N = 2^a \cdot c$ and solve the congruence
for 2^a and c; then combine the results using the CRT;

else if gcd$(k,\ N) \neq 1$ or gcd$(D,\ N) \neq 1$ **then**
write N as the product of two non-trivial factors, and solve for each
factor; then combine the results using the CRT;

else if $\pm D$ is an integer square, $D = \pm b^2$ **then**
begin
Use Lemma 4 to solve $c^2 \pm d^2 \equiv k (\text{mod } N)$;
return$(c,\ db^{-1})$
end;

else if $|k| < |D|$ **then**
begin { interchange roles of k and d}
$(c, d) :=$ Pollard(k, D, N);
return$(cd^{-1},\ d^{-1})$
end;

else begin
$p := k$;
$notfound :=$ true;
while $(notfound)$ **do**
if $p > 0$ and p is prime and D is a quadratic residue (mod p)
then $notfound :=$ false
else $p := p + N$;
use Lemma 3 to find λ, u, and v such that $u^2 - Dv^2 = \lambda p$;
$(w, z) :=$ Pollard(D, λ, N); {here's the recursive call}
use Lemma 1 to write $x^2 - Dy^2 \equiv (u^2 - Dv^2)(w^2 - Dz^2)^{-1}$;
return(x, y);
end;

end;

Verification is left to the reader.

Let us attempt to estimate the running time of the function Pollard. The hardest thing to estimate is how long it will take to find the suitable prime numbers congruent to $k \pmod N$. There is a heuristic argument, due to Wagstaff, that if p is the smallest prime congruent to $k \pmod N$, then "usually"

$$p \approx \varphi(N) \log N \log \varphi(N).$$

He collected statistics to support this conjecture. (See [Wag] for the heuristic argument and related conjectures by other authors.) Since D is a quadratic residue of approximately half of all primes, it seems reasonable to assume the following

Conjecture.

The smallest prime p that is congruent to $k \pmod N$ for which D is a quadratic residue $\pmod p$ is $O(N(\log N)^2)$.

Thus, assuming the truth of this conjecture, we need to look at $O((\log N)^2)$ numbers before we find a prime p of the requisite form.

We have shown that each application of Lemma 3 takes fewer than $1 + \log_4 |D|$ steps.

What's left to do is estimate the number of recursive calls in the function Pollard. Since one application of Lemma 3 reduces k to $\frac{1}{4} + \sqrt{\frac{4}{3}|D|}$, we only need $O(\log\log|D|)$ iterations to get to the "bottom" of the recursion.

Thus, assuming the truth of the conjecture, we see that the running time is approximately

$$O((\log N)(\log\log N)M(N)) + O((\log N)^2(\log\log N)P(N))$$

where $M(N)$ is the time required to multiply or divide integers $\pmod N$ and $P(N)$ is the time required to test a number of magnitude N for primality. If we use probabilistic prime tests, and check the results of our probabilistic square-root algorithm, then the algorithm runs in random polynomial time. Note that we can check the result to make sure it is correct; hence the algorithm *never* returns an incorrect result.

V. An Example.

Consider solving the congruence

$$x^2 - 2345y^2 \equiv 5521 \pmod{8023}.$$

The first step is to find a prime congruent to 5521 (mod 8023). We're lucky here, since 5521 is prime. And we're still luckier, because we find that $2345^{2760} \equiv 1 \pmod{5521}$, so 2345 is a quadratic residue (mod 5521). Now we run the algorithm of Lemma 3. We need to solve $x_0^2 \equiv 2345 \pmod{5521}$. We easily find $x_0 = 812$. Following Lemma 3, we find successively

$x_0^2 - 2345 = m_1 m_0 = 119 \cdot 5521$

$x_1 \equiv x_0 \pmod{m_1} \equiv -21$

$x_1^2 - 2345 = m_2 m_1 = -16 \cdot 119.$

We stop here, since $|m_2|$ is sufficiently small.

Now let's find the A's: we get $A_2 = 0$, $A_1 = 1$, and $A_0 = -7$. From equation (8), with $i = 0$, we find

$$163^2 - 2345 \cdot 7^2 = -16 \cdot 5521. \tag{10}$$

Now it remains to solve

$$w^2 - 2345 z^2 \equiv -16 \pmod{8023}. \tag{11}$$

Let's switch the roles of 2345 and -16 using Lemma 2. We get the new equation $c^2 + 16d^2 \equiv 2345 \pmod{8023}$. But now we can pull out the square factor 16 to get

$$c^2 + (4d)^2 \equiv 2345.$$

This can be solved by Lemma 4, as follows:

We are looking for a prime p of the form $4t + 1$ with $p \equiv 2345 \pmod{8023}$. Eventually we find $p = 66529$. Running the algorithm in [Sha1], we find

$$252^2 + 55^2 = 66529.$$

Thus $c = 252$ and $4d = 55$. Thus $d = 6031$. Now we have the solution to (11): $w = cd^{-1} = 4978$, $z = d^{-1} = 4668$. Use Lemma 1 to write

$$(4978^2 - 2345 \cdot 4668^2)^{-1} = 6709^2 - 2345 \cdot 1714^2,$$

and multiply by (10) to get

$$1088^2 - 2345 \cdot 5425^2 \equiv 5521 \pmod{8023},$$

as desired.

VI. More observations.

It is actually possible to dispense with the computation of the A_i in Lemma 3. All we really need to do is continue the algorithm until m_k is sufficiently small. At this point we have the equations

$$x_0^2 - D = m_1 m_0;$$
$$x_1^2 - D = m_2 m_1;$$
$$x_2^2 - D = m_3 m_2; \ldots$$
$$x_n^2 - D = m_{n+1} m_n.$$

Now, writing (e, f) for $e^2 - Df^2$, we multiply the above equations together to get

$$(x_0, 1)(x_1, 1) \cdots (x_n, 1) = m_0 m_1^2 m_2^2 \cdots m_n^2 m_{n+1}.$$

If $a = m_1 m_2 \cdots m_n$, then this can be rewritten as

$$(a^{-1}, 0)(x_0, 1)(x_1, 1) \cdots (x_n, 1) \equiv k\lambda \pmod{N},$$

since $m_0 = p \equiv k \pmod{N}$ and $m_{n+1} = \lambda$. Using Lemma 1, we can combine all the forms on the left side of the congruence. Now it suffices to solve $u^2 - Dv^2 \equiv \lambda$, as before.

(In fact, this is the way Shamir presented the algorithm.)

Another point: in the course of this algorithm, we frequently compute inverses (mod N). In practice, we may occasionally encounter the case where we must calculate a^{-1} but a has a non-trivial gcd with N. This is not a problem, since if it occurs, we can split N as the product of two non-trivial factors and run the algorithm on both pieces; then put the results together using the Chinese remainder theorem.

VII. How to view the reduction of Lemma 3 as a lattice problem.

It's possible to view the reduction of Lemma 3 as a lattice problem! Suppose we want to write

$$x^2 + Dy^2 = \lambda p$$

where $D > 0$ and λ is as small as possible. Bring the "D" inside the square; then the equation becomes

$$x^2 + \left(\sqrt{D}y\right)^2 = \lambda p.$$

Now the left side looks like the norm of some vector!

A simple example should make this clear. Suppose we want to solve

$$x^2 + 264y^2 = \lambda \cdot 997$$

for x and y, where λ is as small as possible. Compute a square root a of -264 in $\mathbb{Z}_p$; we find $a \equiv \pm 428$. Now consider the lattice in $\mathbf{R}^2$ generated by the rows of

$$\begin{pmatrix} 428 & \sqrt{264} \\ 997 & 0 \end{pmatrix}.$$

Find a reduced basis: we get

$$\begin{pmatrix} 5 & 7\sqrt{264} \\ 141 & -2\sqrt{264} \end{pmatrix}.$$

This shows that

$$5^2 + 264 \cdot 7^2 = \lambda \cdot 997$$

and we have $\lambda = 13$.

We can even use Minkowski's inequality to improve the constant 4/3 from Lemma 3; we find that we can write

$$x^2 + Dy^2 = \lambda \cdot p$$

with $\lambda < \frac{4}{\pi}\sqrt{D}$.

Now let's consider the case where $D < 0$. We cannot use the method of the above paragraph directly since $\sqrt{D}$ is imaginary. Instead, we use the same trick that was employed in [Sha], and convert a lattice problem over $\mathbb{Z}[i]$ to one over $\mathbb{Z}$.

A simple example should make this clear. Suppose we want to solve

$$x^2 - 667480y^2 = \lambda \cdot 738121$$

for x and y, where λ is as small as possible. Compute a square root a of 667480 in $\mathbb{Z}_p$; we find $a = \pm 63657$. Now consider the lattice in $\mathbf{R}[i]$ generated by the rows of

$$\begin{pmatrix} 63657 & i\sqrt{667480} \\ 738121 & 0 \end{pmatrix}.$$

Convert this to a lattice over $\mathbf{R}$ using the standard isomorphism

$$a + bi \Leftrightarrow \begin{pmatrix} a & b \\ -b & a \end{pmatrix};$$

we get

$$\begin{pmatrix} 63657 & 0 & 0 & \sqrt{667480} \\ 0 & 63657 & -\sqrt{667480} & 0 \\ 738121 & 0 & 0 & 0 \\ 0 & 738121 & 0 & 0 \end{pmatrix}.$$

At this point, we could use a basis reduction algorithm to find a shortest vector in this lattice; we would obtain the new matrix

$$\begin{pmatrix} 12131 & 0 & 0 & -23\sqrt{667480} \\ 0 & 12131 & 23\sqrt{667480} & 0 \\ -25763 & 0 & 0 & -12\sqrt{667480} \\ 0 & 25763 & 12\sqrt{667480} & 0 \end{pmatrix}.$$

This implies that a solution is $x = 12131$, $y = -23$ and indeed we find

$$12131^2 - 667480 \cdot 23^2 = -279 \cdot 738121.$$

An easier solution is obtained if we note that the matrix above can be decomposed into two 2-dimensional pieces. If we apply basis reduction to

$$\begin{pmatrix} 63657 & \sqrt{667480} \\ 738121 & 0 \end{pmatrix}$$

we get

$$\begin{pmatrix} 12131 & -23\sqrt{667480} \\ -25763 & -12\sqrt{667480} \end{pmatrix}$$

which, of course, gives us the same solution found above.

VIII. Postscript.

After this report was written, I received the preprint [Pol] in which he discusses his algorithm. A paper is being written by Schnorr et al. on new versions of the signature scheme which seem harder than the original OSS scheme.

References

[AMM] L. Adleman, K. Manders, and G. Miller, On taking roots in finite fields, 18th FOCS, 1977.

[Lag] Joseph Louis Lagrange, Sur la solution des problèmes indéterminés du second degré, Mémoires de l'Académie royale des Sciences et Belles-Lettres de Berlin, 23 (1769); reprinted in Oeuvres de Lagrange, V. II (Gauthier-Villars, Paris) 1868, pp. 377-535.

[Leg] Adrien-Marie Legendre, Zahlentheorie, Leipzig (1893) 32-33.

[OSS] H. Ong, C. P. Schnorr, and A. Shamir, An efficient signature scheme based on quadratic equations, Proc. 16th ACM Symp. Theor. Comput. (1984) 208-216.

[Pol] J. M. Pollard, Solution of $x^2 + ky^2 \equiv m \pmod{n}$, with application to digital signatures, typed ms., July, 1984.

[Sha] J. Shallit, Random polynomial time algorithms for sums of squares, University of Chicago, Department of Computer Science, Technical Report 85-001, January, 1985.

[UH] J. V. Uspensky and M. A. Heaslet, Elementary Number Theory, McGraw-Hill, 1939.

[Wag] Samuel S. Wagstaff, Jr., Greatest of the least primes in arithmetic progressions having a given modulus, Math. Comp. 33 (1979) 1073-1080.

October 22, 1984

1st Revision: October 28, 1984

2nd Revision: October 31, 1984

3rd Revision: November 12, 1984