RC 20089 (88845) 6/6/95
Computer Science/Mathematics 24 pages

Research Report

Quadratic Integer Programming, and Factoring with a Clue

Don Coppersmith

IBM Research Division
T. J. Watson Research Center
Yorktown Heights, NY 10598

IBM Research Division
Almaden • T.J. Watson • Tokyo • Zurich

Quadratic Integer Programming, and Factoring with a Clue

Don Coppersmith

IBM Research
T. J. Watson Research Center
Yorktown Heights, NY 10598

Abstract

We present a computational method for solving a single quadratic equation in two unknowns which are constrained to be integers from bounded intervals. We give heuristic estimates of the running time in terms of the parameters.

We then apply this method to the problem of finding the factors P,Q of an integer $N = PQ$ when given many high order bits of P and Q. If each of P,Q has m bits and we know the high order $((0.6m) - \ell)$ bits of P, we heuristically expect to find the factorization after a computation of length $O(2^{\ell})$. We also treat the case where we know the high order bits of a *multiple* of Q.

Quadratic integer programming, and factoring with a clue

Don Coppersmith
IBM Research Division
T. J. Watson Research Center
Yorktown Heights, NY 10598
May 30, 1995

Abstract

We present a computational method for solving a single quadratic equation in two unknowns which are constrained to be integers from bounded intervals. We give heuristic estimates of the running time in terms of the parameters.

We then apply this method to the problem of finding the factors P,Q of an integer $N = PQ$ when given many high order bits of P and Q. If each of P,Q has m bits and we know the high order $((0.6m) - \ell)$ bits of P, we heuristically expect to find the factorization after a computation of length $O(2^{\ell})$. We also treat the case where we know the high order bits of a *multiple* of Q.

Quadratic Integer Programming

Let

$$f(x, y) = C_1 + C_x x + C_y y + C_{xx} x^2 + C_{xy} xy + C_{yy} y^2$$

be a quadratic polynomial in two integer variables (x, y) with integer coefficients C_*. Let $F \subseteq \mathbb{R}^2$ be the curve

$$F = \{(x, y) \in \mathbb{R}^2 \mid f(x, y) = 0\}$$

The unknowns x, y are bounded:

$$|x| \le X, \qquad |y| \le Y \le X.$$

We define the bound

$$C = \max \{|C_1|,\ |C_x X|,\ |C_y Y|,\ |C_{xx} X^2|,\ |C_{xy} XY|,\ |C_{yy} Y^2|\}$$

Our task is to find integers (x, y) from the indicated ranges satisfying $f(x, y) = 0$.

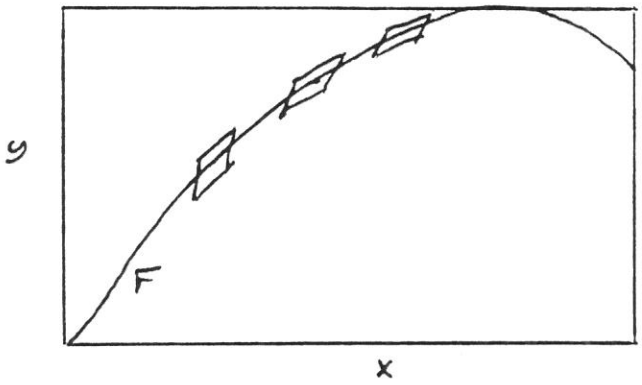
We will use as an example the following parameters, motivated by the application in section 3:

$$X = 2^{120}, \quad Y = 2^{112}, \quad C = 2^{496}.$$

Caveat: I will be dropping small constant factors throughout the paper.

One method of attack would be to iterate through possible values of y, solve the resulting quadratic equation in x, and check whether x is an integer in the desired range. But this requires $O(Y)$ computations, which in our example is infeasible (2^{112}).

Instead, we will cover the curve F by a collection of long skinny diamonds and search in each diamond for a solution.



For each rational slope s/t with s,t sufficiently small, we find a point (x_0, y_0) on F where the tangent to F has slope s/t. (In out example, "sufficiently small" will mean $|s| \le S = 2^{18}$, $|t| \le T = 2^{26}$.) Here x_0 and y_0 need not be rational. We then parameterize the integer lattice points (x, y) which are near (x_0, y_0) and quite near the curve F, in terms of two integer parameters u,v, where u runs along the tangent to F, and v runs mostly along the tangent but with a small component along the normal.

The parameters (u,v) depend on (x, y) through an affine unimodular transformation depending on (s,t). Specifically, from the continued fraction expansion of s/t we find integers q,r with $qt - rs = 1$, and with $|q|$ as large as possible subject to $|q| < |s|$. Find real numbers u_1, v_1, each bounded by 1/2 in absolute value, causing x_1, y_1 as defined below to be integers:

$$x_1 = x_0 + tu_1 + rv_1$$
$$y_1 = y_0 + su_1 + qv_1$$

The lattice point (x_1, y_1) will be fairly close to (x_0, y_0) and quite close to F:

$$\text{Distance}((x_1, y_1),(x_0, y_0)) \le \sqrt{t^2 + s^2}$$
$$\text{Distance}((x_1, y_1), F) \le \frac{1}{\sqrt{t^2 + s^2}}$$

The integer lattice points (x, y) near (x_0, y_0) and quite near F are parameterized as

$$x = x_1 + tu + rv$$
$$y = y_1 + su + qv$$

where u,v are now integers.

We convert the quadratic equation to these new parameters:

$$\hat{f}(u,v) = c_1 + c_u u + c_v v + c_{uu} u^2 + c_{uv} uv + c_{vv} v^2 = 0$$

Both c_1 and c_u will be small, because of the proximity of (x_1, y_1) to F and to (x_0, y_0), respectively.

We will impose bounds on the parameters u and v:

$$|u| < U, \qquad |v| < V$$

Defining

$$c = \max\{|c_1|,\ |c_u U|,\ |c_v V|,\ |c_{uu} U^2|,\ |c_{uv} UV|,\ |c_{vv} V^2|\},$$

we will require the two conditions

$$c = |c_v V|$$
$$c \ge U^4 V^4$$

The former condition ensures that we have chosen V large enough to capture deviations of F from its tangent due to quadratic terms. The latter condition is required for the heuristic estimates in the lattice basis reduction.

A word about sizes

We have mentioned that the coefficients c_1, c_u are both small. We can bound c_v as follows:

$$tc_v - rc_u = (qt - rs)[C_y + C_{xy}x + 2C_{yy}y] = (1)[C_y + C_{xy}x + 2C_{yy}y]$$
$$tc_v = rc_u + C_y + C_{xy}x + 2C_{yy}y$$
$$|tc_v| \le |rc_u| + 4C/Y.$$

Bounds on c_{uu}, c_{uv}, c_{vv} follow from:

$$c_{uu} = C_{xx}t^2 + C_{xy}ts + C_{yy}s^2$$
$$c_{uv} = 2C_{xx}tr + C_{xy}(qt + rs) + 2C_{yy}sq$$
$$c_{vv} = C_{xx}r^2 + C_{xy}rq + C_{yy}q^2$$
$$|c_{uu}|,\ |c_{uv}|,\ |c_{vv}| \le 6C \max\{(t/X)^2, (s/Y)^2\}$$

Recalling our example

$$C \simeq 2^{496},\ X \simeq 2^{120},\ Y \simeq 2^{112},\ T \simeq 2^{26},\ S \simeq 2^{18},$$

we would obtain

$$c_v \simeq C/(YT) \simeq 2^{358}, \quad c_{uu} \simeq C(T/X)^2 \simeq 2^{308}.$$

Appropriate choices of U, V would appear to be

$$U \simeq (c_v)^{0.4}(c_{uu})^{-0.3} \simeq 2^{51}$$
$$V \simeq (c_{uu})^{0.4}(c_v)^{-0.2} \simeq 2^{52}$$
$$c \simeq c_v V \simeq 2^{410}$$

Lattice basis reduction

We would like to solve $\hat{f}(u,v) = 0$ by lattice basis reduction methods [LLL]. These methods find bounded integer solutions to linear equations and inequalities. But we have a quadratic equation to solve.

We use the following technique, which seems to be due to Odlyzko [Odl]: replace the quadratic terms u^2, uv, v^2 in $\hat{f}$ by new variables x_{uu}, x_{uv}, x_{vv}, which we treat as independent unknowns. We then have a single linear equation in five bounded unknowns:

$$\tilde{f}(u, v, x_{uu}, x_{uv}, x_{vv}) \equiv c_1 + c_u u + c_v v + c_{uu}x_{uu} + c_{uv}x_{uv} + c_{vv}x_{vv} = 0$$
$$|u| < U, \quad |v| < V$$
$$|x_{uu}| < U^2, \quad |x_{uv}| < UV, \quad |x_{vv}| < V^2$$

We use lattice basis reduction techniques to find a solution to this linear equation. Use an LLL-type reduction on the column basis of the following matrix:

$$M = \begin{bmatrix} 10c_1 & 10c_u & 10c_v & 10c_{uu} & 10c_{uv} & 10c_{vv} \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/U & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/V & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/U^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/UV & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/V^2 \end{bmatrix}$$

If (u,v) is a solution to $\hat{f}(u,v) = 0$, then the column vector

$$M \times [1, u, v, u^2, uv, v^2]^T = [0, 1, u/U, v/V, u^2/U^2, uv/UV, v^2/V^2]^T$$

is guaranteed to have small Euclidean norm (less than $\sqrt{6}$), and heuristically we do not expect to have many other vectors of smaller norm among integer combinations of columns of M (see below). If the correct column vector is indeed among the few smallest, lattice basis reduction can be expected to find it quickly. Conversely, given a small vector of the form

$$M \times [1, u, v, x_{uu}, x_{uv}, x_{vv}]^T = [0, 1, u/U, v/V, x_{uu}/U^2, x_{uv}/UV, x_{vv}/V^2]^T$$

it will represent a solution to the original problem if and only if $x_{uu} = u^2$, $x_{uv} = uv$, and $x_{vv} = v^2$ all hold.

Here is the *heuristic argument*: The number of lattice points inside the five-dimensional box is

$$(2U - 1)(2V - 1)(2U^2 - 1)(2UV - 1)(2V^2 - 1) < 32U^4V^4$$

If everything were random, the probability that a given lattice point yielded the solution $\tilde{f}(u, v, x_{uu}, x_{uv}, x_{vv}) = 0$ would be about $1/c$, since the terms we are adding are each bounded by about c. Since $c > U^4V^4$, we *expect* only a small bounded number of lattice points yielding $\tilde{f}(u, v, x_{uu}, x_{uv}, x_{vv}) = 0$. Points yielding $\tilde{f} \neq 0$ will have large norm (at least 10) and need not be considered.

But there may be a larger family of lattice points yielding small norm. The intersection of the 5-dimensional box with the hyperplane $\tilde{f} = 0$ may be a piece of a lattice of dimension anywhere between 0 and 4.

If the dimension of this lattice is 1, 2 or 3, we still have a hope of success. Taking for example a 3-dimensional intersection, there are three bounded independent integer parameters r_1, r_2, r_3, in terms of which the five variables are expressed:

$$\begin{aligned} u &= d_{11}r_1 + d_{12}r_2 + d_{13}r_3 \\ v &= d_{21}r_1 + d_{22}r_2 + d_{23}r_3 \\ x_{uu} &= d_{31}r_1 + d_{32}r_2 + d_{33}r_3 \\ x_{uv} &= d_{41}r_1 + d_{42}r_2 + d_{43}r_3 \\ x_{vv} &= d_{51}r_1 + d_{52}r_2 + d_{53}r_3 \\ d_{ij} &\in \mathbb{Z} \text{ known} \end{aligned}$$

Replace the independent variables x_{uu}, x_{uv}, x_{vv}, by their meanings u^2, uv, v^2, respectively, to obtain five equations

$$\begin{aligned} u &= d_{11}r_1 + d_{12}r_2 + d_{13}r_3 \\ v &= d_{21}r_1 + d_{22}r_2 + d_{23}r_3 \\ u^2 &= d_{31}r_1 + d_{32}r_2 + d_{33}r_3 \\ uv &= d_{41}r_1 + d_{42}r_2 + d_{43}r_3 \\ v^2 &= d_{51}r_1 + d_{52}r_2 + d_{53}r_3 \end{aligned}$$

in five unknowns u, v, r_1, r_2, r_3. Although these unknowns happen to be bounded integers, we can solve the equations over the reals, and if an integer solution arises, it yields the solution to our original problem. The cases of 1- and 2-dimensional intersections are handled more easily.

A 5-dimensional intersection will not occur except in highly degenerate cases, which can be handled by other means. So the only problematic case is a 4-dimensional intersection: four parameters r_1, r_2, r_3, r_4 in terms of which the five variables are expressed. This gives five equations in six variables, which we do not know how to solve efficiently. (This is very similar to the original problem: more variables than equations.) The extent to which this 4-dimensional intersection is encountered in practice will determine how likely it is that our procedure will fail.

Coverage

Each choice of slope (s/t) leads to a lattice problem which should (heuristically) find any solutions to the original problem within the appropriate diamond-shaped region. It is of interest to know how much of the original parameter space has been covered by each such region.

Recalling $x = x_1 + tu + rv$ and the bounds $|u| < U \simeq 2^{51}$, $|v| < V \simeq 2^{52}$, and $|t| \simeq T \simeq 2^{26}$, we see that the variable x covers a range of about 2^{77}. Since the total range in x is $X = 2^{120}$, we will need about 2^{43} such regions to cover the entire range.

Our choice of slopes (s/t) must satisfy $|s| < S \simeq 2^{18}$, $0 \le t < T \simeq 2^{26}$, and $g.c.d.(s,t) = 1$. There are about 2^{43} such choices. So, by a counting argument, we more or less cover the entire range of possibilities with these diamond-shaped regions. ("More or less" because the regions might not mesh nicely.)

This also implies that the computational load is about 2^{43} separate LLL calculations.

Sizes revisited

Now we are in a position to see how the various parameters fit together. We assume that C, X, Y are given to us by the problem statement. The following additional constraints present themselves:

$$\frac{S}{T} = \frac{Y}{X}$$

The characteristic slope should reflect the proportions of the bounding rectangle.

$$U \ge V$$

Variations in u go straight along the tangent line, while variations in v are skewed. In a diamond where $V > U$, although two points of the curve F may be included in the diamond, the intervening curve may lie outside.

$$ST \ge \frac{X}{TU}$$

We have about ST choices of slope, each covering TU values of x, so this requirement is needed to cover the entire range of x. The number of LLL problems is given by ST.

$$c_v V \ \ge \ \max(\ c_{uu} U^2 \ , \ U^4 V^4)$$

$$i.e. \quad \frac{CV}{YT} \ \ge \ \max(\frac{CT^2 U^2}{X^2} \ , \ U^4 V^4)$$

These inequalities summarize restrictions developed in earlier sections.

The solution to these restrictions, and specialization to the example problem, is given below:

$$T = X^{0.7} Y^{-0.3} C^{-0.05} \simeq 2^{25.6}$$
$$S = X^{-0.3} Y^{0.7} C^{-0.05} \simeq 2^{17.6}$$
$$U = V = X^{-0.1} Y^{-0.1} C^{0.15} \simeq 2^{51.2}$$
$$\text{work} = TS = X^{0.4} Y^{0.4} C^{-0.1} \simeq 2^{43.2}$$

Factoring with a clue.

An important application is to integer factorization. Suppose we are given an integer $N = PQ$, with P,Q of roughly equal sizes, $P \simeq Q \simeq M = 2^m$. Rivest and Shamir [RS] have shown how to factor N if given the high order $2m/3$ bits of P. With the techniques of this paper, if we are given the high order $(0.6m) - \ell$ bits of P we can sometimes factor N using $O(2^\ell)$ computational steps, where again each "computational step" involves an LLL reduction.

We have approximations P_0, Q_0 to P, Q:

$$P = P_0 + a$$
$$Q = Q_0 + b$$
$$|a|, |b| < A = 2^{0.4m + \ell}$$

We assume that P_0, Q_0 have been chosen to satisfy

$$|P_0 Q_0 - N| < A^2;$$

this is not hard to accomplish.

Applying continued fraction expansion to the fraction P_0/Q_0, we find integers g_1, h_1, g_2, h_2 satisfying

$$\frac{g_1}{h_1} < \frac{P_0}{Q_0} < \frac{g_2}{h_2}$$
$$g_2 h_1 - g_1 h_2 = 1$$
$$h_i \simeq H \equiv 2^{0.3m - 0.5\ell}$$

Caveat: This depends on the ratio P_0/Q_0 having approximants whose denominators have the right magnitude, which may or may not be applicable in a given situation.

We set

$$a = x g_1 + y g_2$$
$$b = -x h_1 - y h_2$$

so that

$$0 = PQ - N = (P_0 + xg_1 + yg_2)(Q_0 - xh_1 - yh_2) - N$$
$$0 = (P_0Q_0 - N) + (g_1Q_0 - h_1P_0)x + (g_2Q_0 - h_2P_0)y - g_1h_1x^2 - (g_1h_2 + g_2h_1)xy - g_2h_2y^2$$
$$= C_1 + C_x x + C_y y + C_{xx}x^2 + C_{xy}xy + C_{yy}y^2$$

A straightforward calculation gives:

$$|C_1| < A^2 = 2^{0.8m + 2\ell}$$
$$|C_x|,\ |C_y| < \frac{HQ}{H^2} = 2^{0.7m + 0.5\ell}$$
$$|C_{xx}|,\ |C_{xy}|,\ |C_{yy}| \simeq H^2 = 2^{0.6m - \ell}$$
$$X \simeq Y \simeq 2^{0.1m + 1.5\ell}$$
$$C \simeq 2^{0.8m + 2\ell}$$

We can apply the techniques of the previous section directly, obtaining a work factor of

$$\text{work} = X^{0.4}Y^{0.4}C^{-0.1} = 2^{\ell}$$

applications of LLL.

Another interpretation is that we are using 2^{ℓ} invocations of a routine that uses the high order $0.6m$ bits of P to find P,Q, and cycling over the 2^{ℓ} possible choices of the ℓ bits that are not among the $(0.6m) - \ell$ actually given to us. I prefer the interpretation of the previous section, where we are partitioning a hyperbola into smaller sections and blowing them up.

High order versus low order bits

Instead of iterating on the 2^{ℓ} possible values of the high order ℓ bits among the unknown $(0.4m) + \ell$ low bits of P, suppose we were to iterate on the 2^k possible values of the lower order k bits of P, for some value $k \leq \ell$. What would be the resulting efficiency?

Once we know the low order k bits of P, we easily obtain the low order k bits of Q from $N \equiv PQ \pmod{2^k}$. Then since the variables a,b are related to x, y by a unimodular transformation, we easily compute the low order k bits of x and of y.

Letting x', y' be the (unknown) high order bits of x,y, we see that x', y' are subject to the tighter bounds:

$$|x'|,\ |y'| \leq X' = \frac{X}{2^k}$$

$$0 = PQ - N = (P_0 + xg_1 + yg_2)(Q_0 - xh_1 - yh_2) - N$$
$$0 = (P_0Q_0 - N) + (g_1Q_0 - h_1P_0)x + (g_2Q_0 - h_2P_0)y - g_1h_1x^2 - (g_1h_2 + g_2h_1)xy - g_2h_2y^2$$
$$= C_1 + C_x x + C_y y + C_{xx}x^2 + C_{xy}xy + C_{yy}y^2$$

A straightforward calculation gives:

$$|C_1| < A^2 = 2^{0.8m + 2\ell}$$
$$|C_x|,\ |C_y| < \frac{HQ}{H^2} = 2^{0.7m + 0.5\ell}$$
$$|C_{xx}|,\ |C_{xy}|,\ |C_{yy}| \simeq H^2 = 2^{0.6m - \ell}$$
$$X \simeq Y \simeq 2^{0.1m + 1.5\ell}$$
$$C \simeq 2^{0.8m + 2\ell}$$

We can apply the techniques of the previous section directly, obtaining a work factor of

$$\text{work} = X^{0.4}Y^{0.4}C^{-0.1} = 2^{\ell}$$

applications of LLL.

Another interpretation is that we are using 2^{ℓ} invocations of a routine that uses the high order $0.6m$ bits of P to find P,Q, and cycling over the 2^{ℓ} possible choices of the ℓ bits that are not among the $(0.6m) - \ell$ actually given to us. I prefer the interpretation of the previous section, where we are partitioning a hyperbola into smaller sections and blowing them up.

High order versus low order bits

Instead of iterating on the 2^{ℓ} possible values of the high order ℓ bits among the unknown $(0.4m) + \ell$ low bits of P, suppose we were to iterate on the 2^k possible values of the lower order k bits of P, for some value $k \le \ell$. What would be the resulting efficiency?

Once we know the low order k bits of P, we easily obtain the low order k bits of Q from $N \equiv PQ \pmod{2^k}$. Then since the variables a,b are related to x,y by a unimodular transformation, we easily compute the low order k bits of x and of y.

Letting x', y' be the (unknown) high order bits of x,y, we see that x', y' are subject to the tighter bounds:

$$|x'|,\ |y'| \le X' = \frac{X}{2^k}$$

When we rephrase the quadratic equation $f(x,y) = 0$ in terms of these new parameters $f'(x', y') = 0$, we see that the low order k bits are automatically satisfied. The effect is the same as dividing the equation (and the coefficients C_*) by 2^k, and we get

$$C'_1 = C_1/2^k = 2^{0.8m + 2\ell - k}$$
$$C'_x = C_x = 2^{0.7m + 0.5\ell}$$
$$C'_{xx} = C_{xx}2^k = 2^{0.6m - \ell + k}$$
$$C' = C/2^k = 2^{0.8m + 2\ell - k}$$
$$\text{work}' = X'^{0.4}Y'^{0.4}C'^{-0.1} = 2^{\ell - 0.7k}$$

Recalling that we have to perform 2^k instances (corresponding to the 2^k possible values of the low order bits), our total work has become

$$\text{Total work} = \text{work}' \times 2^k = 2^{\ell + 0.3k}.$$

In general this increases our work factor. It does lend a certain amount of flexibility to the process, which might be useful in some circumstances. Also, it allows us to take advantage of other information about P,Q. For instance, instead of iterating on possible values of ($P \bmod 2^k$), we can iterate on values of ($P \bmod K$) where K need not be a power of 2. Since P is prime, we know $g.c.d.(P,K) = 1$, so that we need only examine $\phi(K)$ possible values (Euler's ϕ-function). For $K = 6$ this saves us a little, since $\phi(6) = 2 < 3.505 = 6^{0.7}$.

Application to the Vanstone-Zuccherato scheme

In a recent proposal of Vanstone and Zuccherato [VZ], a composite number $N = PQ$ is known, and the secret prime factors P,Q are constructed in such a way that their high order bits are known. We show that for one choice of the parameters in [VZ], the opponent can use this information to find P,Q efficiently. We also show that for a second choice in the paper, the scheme is less secure than the authors believe, though our proposed attack on this second choice is on the fringes of feasibility.

The first choice of parameters

In Section 2 of [VZ], the following scheme is proposed.

$$N = PQ$$
$$P = 2^{364} g_1 + a_1$$
$$Q = 2^{364} h_1 + a_2$$
$$2^{147} \le g_1, h_1 < 2^{148}$$
$$0 \le a_i < 2^{213}$$

$$N = 2^{728}(g_1 h_1) + 2^{364}(g_1 a_2 + h_1 a_1) + (a_1 a_2) \tag{1}$$

The integers g_1, h_1 are known to the opponent; the a_i are unknown. Each of P and Q has a long string of binary 0s in the middle of its binary representation. In the binary representation of N, the quantity $2^{728}(g_1 h_1)$ occupies the field between bits 728 and 1023; the quantity $2^{364}(g_1 a_2 + h_1 a_1)$ occupies the field between bits 364 and 726; and the quantity $(a_1 a_2)$ occupies the field between bits 0 and 425. Notice that the latter two fields overlap in 62 bits.

The catch here is that g_1/h_1 is an unusually good rational approximation to P/Q. This raises a warning flag concerning existence of approximations g_i/h_i with denominators of the right size $h_i \simeq H$. We will work directly with the given approximation g_1/h_1.

A possible attack, costing 2^{62} iterations, involves guessing the low-order 62 bits of the quantity

$$g_1 a_2 + h_1 a_1$$

(we already know its high order bits), calculating the value

$$a_1 a_2 = N - 2^{728} g_1 h_1 - 2^{364}(g_1 a_2 + h_1 a_1),$$

solving a quadratic equation to find (a_1, a_2), and checking for a_i being integers. The cost per iteration is quite small: considerations modulo each small prime allow us to discard about half the possible values, so that sieving techniques will apply. This roughly corresponds to iterating over possible values of y in the previous section. This attack is related to one given in section 11 of [VZ].

Instead we mimic our work in the previous section. By the extended Euclidean algorithm, assuming $g.c.d.(g_1, h_1) = 1$, compute integers g_2, h_2 with $0 \le g_2 \le g_1$, $0 \le h_2 < h_1$, such that

$$g_2 h_1 - g_1 h_2 = 1. \tag{2}$$

As usual, this implies that $g.c.d.(h_1, h_2) = 1$.

By integer programming in two dimensions, compute integers e_1, e_2 in the range $0 \le e_i < 2^{213}$ satisfying

$$g_1 e_2 + h_1 e_1 = \left\lfloor (N - 2^{728}(g_1 h_1) - 2^{425})/2^{364} \right\rfloor,$$

which implies

$$\left| N - \left(2^{728}(g_1 h_1) + 2^{364}(g_1 e_2 + h_1 e_1) + 2^{425}\right) \right| < 2^{364}. \tag{3}$$

(The term 2^{425} is an estimate for the missing quantity $a_1 a_2$.) These e_i will be a starting point for computing a_i.

For some unknown coefficients x, y, w, z, we can express

$$\begin{aligned} a_1 &= e_1 + x g_1 + y g_2 \\ |y| &\le g_1/2 \\ a_2 &= e_2 + (w - x)h_1 + (z - y)h_2 \\ |z| &\le h_1/2 \end{aligned}$$

The coefficients x, y, w, z are integers because of (2). From $0 \le a_1 a_2 < 2^{426}$ we deduce

$$|a_1 a_2 - 2^{425}| \le 2^{425}. \tag{4}$$

Combining equations (1, 3, 4) we find

$$\left| 2^{364}\{(g_1 a_2 + h_1 a_1) - (g_1 e_2 + h_1 e_1)\} \right| < 2^{425} + 2^{364}$$

or, dividing by 2^{364},

$$|(g_1 a_2 + h_1 a_1) - (g_1 e_2 + h_1 e_1)| < 2^{61} + 1.$$

All the quantities are integers, so we have

$$\begin{aligned} 2^{61} &\ge |g_1(a_2 - e_2) + h_1(a_1 - e_1)| \\ &= \left|g_1[(w - x)h_1 + (z - y)h_2] + h_1[x g_1 + y g_2]\right| \\ &= |g_1(w h_1 + z h_2) - y(g_1 h_2 - g_2 h_1)| \\ &= |g_1(w h_1 + z h_2) \pm y \times 1| \end{aligned}$$

We know $|y| \le g_1/\sqrt{2}$ and $g_1 > 2^{147}$. So the coefficient of g_1 must be zero, and we get

$$w h_1 + z h_2 = 0, \qquad |y| \le 2^{61}.$$

From $|z| \le h_1/2$ and $g.c.d.(h_1, h_2) = 1$, we deduce that $w = z = 0$. So we have expressed

$$a_1 = e_1 + xg_1 + yg_2$$
$$a_2 = e_2 - xh_1 - yh_2$$
$$|y| \le 2^{61}$$

By size bounds on a_1, e_1, g_1, y, we deduce

$$|x| \le 2^{66} + 2^{61}.$$

In fact, if we had chosen e_i to be close to the centers of their appropriate intervals (selecting τ so that $e'_1 = e_1 + \tau g_1$, $e'_2 = e_2 - \tau h_1$, $|(e'_1/g_1) - (e'_2/h_1)| \le 1$, and replacing e_i by e'_i), we could achieve a little better:

$$|x| \le 2^{65} + 2^{61}.$$

We will use the latter bound on $|x|$.

The relevant equation becomes

$$(2^{364}g_1 + e_1 + xg_1 + yg_2)(2^{364}h_1 + e_2 - xh_1 - yh_2) - N = 0$$
$$[(2^{364}g_1 + e_1)(2^{364}h_1 + e_2) - N] + [-(2^{364}g_1 + e_1)h_1 + (2^{364}h_1 + e_2)g_1]x +$$
$$+ [-(2^{364}g_1 + e_1)h_2 + (2^{364}h_1 + e_2)g_2]y - g_1h_1x^2 - (g_1h_2 + g_2h_1)xy - g_2h_2y^2 = 0$$

Now we can apply the general formulation of the problem, with

$$X = 2^{65}, \quad Y = 2^{61}, \quad C = 2^{426},$$

to obtain

$$\text{work} = X^{0.4}Y^{0.4}C^{-0.1} = 2^{26.0 + 24.4 - 42.6} = 2^{7.8} < 230.$$

(Recall that "work" is the number of LLL invocations.)

Larger choice of parameters

In Section 3 of [VZ], an alternate scheme is proposed, apparently differing only slightly in the parameters, but this slight variance makes a large difference in our ability to solve the equations.

$$N = PQ$$
$$P = 2^{384} g_1 + a_1$$
$$Q = 2^{384} h_1 + a_2$$
$$2^{127} \le g_1, h_1 < 2^{128}$$
$$0 \le a_i < 2^{248}$$

$$N = 2^{768}(g_1 h_1) + 2^{384}(g_1 a_2 + h_1 a_1) + (a_1 a_2)$$

This time the two quantities $2^{384}(g_1 a_2 + h_1 a_1)$ and $(a_1 a_2)$ occupy fields which overlap in 112 bits. Again the opponent knows N, g_1, h_1, and needs to discover a_i.

Again we compute $g_2 < g_1, h_2 < h_1$, and e_i, $0 \le e_i < 2^{248}$. There are unknown integers x, y, with $|x| \le 2^{120}$, $|y| \le 2^{112}$, such that

$$a_1 = e_1 + x g_1 + y g_2$$
$$a_2 = e_1 - x h_1 - y h_2$$

This time our trial and error is more expensive. From

$$X = 2^{120}, \quad Y = 2^{112}, \quad C = 2^{496},$$

we compute

$$\text{work} = X^{0.4} Y^{0.4} C^{-0.1} = 2^{43.2}.$$

Since this represents the number of LLL invocations, the work factor is on the fringe of feasibility. It is far below the work factors quoted in [VZ].

A different hint: a near multiple of P

Some proposed cryptographic systems based on the difficulty of integer factorization (but not RSA itself) use extra information, such as an integer close to a multiple of one of the unknown prime factors. We show in this section that this extra information can lead to an efficient method of factoring the composite modulus, thus introducing a weakness into the cryptographic system. (This section has appeared previously as an internal report [Cop].)

Our setup will be as follows. Again we know a composite number N but not its factorization $N = PQ$, where P and Q are prime. This time we have a *hint*, an integer $S = AQ + B$ with B relatively small, so that S is close to a multiple of Q; the closer to a multiple of Q, the better the hint.

How good a hint do we require to enable us to factor N efficiently? If we have multiple hints, how good need they be?

Single hint

We consider the case of a single hint.

$$N = PQ$$
$$S = AQ + B$$
$$|B| < K$$

We know N, S, and we assume given an order-of-magnitude estimate of K. (In the important special case where P and Q are primes of about equal size, we can use $K \simeq (1/8)N^{1/6}$.) In the earlier section we essentially had $A = 1$, but here A is secret. This complicates matters considerably.

We will introduce many variables in our description. Those whose values we know are: $N, S, K, c, d, e, f, g, h, r, s$. Those whose values are unknown are: $P, Q, A, B, u, v, w, x, y, z$.

Start by doing a continued fraction expansion of the fraction S/N. Stop when the denominators of the successive approximations are comparable to a bound $\sqrt{N/K}$. That is,

$$\frac{c}{d} < \frac{S}{N} < \frac{e}{f}$$
$$de - cf = 1$$
$$df < N/K$$
$$d + f > \sqrt{N/K}$$

The reason we can assume $d + f > \sqrt{N/K}$ is that the interval $[c/d, e/f]$ is broken into two sub-intervals

$$[c/d, e/f] = [c/d, (c+e)/(d+f)] \cup [(c+e)/(d+f), e/f],$$

one of which contains S/N, and the product of the denominators for that sub-interval, either $(d)(d+f)$ or $(f)(d+f)$, exceeds N/K, so that $d + f > \sqrt{N/K}$.

Then we have

$$\frac{e}{f} - \frac{c}{d} = \frac{1}{df} > \frac{K}{N},$$

and since S/N lies in the interval $[c/d, e/f]$, we see that A/P lies close to the interval:

$$\frac{c}{d} - \frac{K}{N} \leq \frac{A}{P} \leq \frac{e}{f} + \frac{K}{N}$$

Thus c/d and e/f are reasonable approximations to A/P as well as to S/N.

Requirements on K. Having defined d and f, we remark that the condition we require of K is that

$$K^2 < \frac{Ndf}{256P^3} \tag{5}$$

In the favorable case $N/2K < df < N/K$, this will follow if we have

$$K^3 < \frac{N^2}{512P^3}$$
$$K < \frac{Q^{2/3}}{8P^{1/3}}$$

Notice that again we are requiring a rational approximation with a denominator of an appropriate size. When P and Q are of about equal size, this means

$$K \lesssim \frac{1}{8} N^{1/6}.$$

We will assume from here out that

$$\frac{c}{d} < \frac{c+e}{d+f} < \frac{S}{N} < \frac{e}{f}$$

and remark that a similar analysis yields the same results in the other case $S/N < (c+e)/(d+f)$.

Define integers x, y (not necessarily positive) by

$$x = Pe - Af$$
$$y = Ad - Pc$$

so that

$$xd + yf = P(ed - cf) + A(-fd + df) = P(1) + A(0)$$
$$xd + yf = P$$

We find bounds on their magnitudes:

$$\left|\frac{A}{P}-\frac{e}{f}\right| \le \left|\frac{A}{P}-\frac{S}{N}\right|+\left|\frac{S}{N}-\frac{e}{f}\right| \le \frac{K}{N}+\frac{1}{(f)(d+f)} \le \frac{2K}{N}$$

$$|x| = |Pe - Af| \le \frac{2PfK}{N}$$

$$\left|\frac{A}{P}-\frac{c}{d}\right| \le \left|\frac{A}{P}-\frac{S}{N}\right|+\left|\frac{S}{N}-\frac{c}{d}\right| \le \frac{K}{N}+\frac{1}{(f)(d)} \le \frac{2}{df}$$

$$|y| = |Ad - Pc| \le \frac{2P}{f}$$

Besides magnitude, we also use modular arithmetic to exploit the fact that N is an exact multiple of P. From

$$N = PQ = (xd + yf)Q,$$

we know

$$xQ \equiv Nd^{-1} \bmod f$$
$$yQ \equiv Nf^{-1} \bmod d$$

Defining $g = (Nd^{-1}) \bmod f$ and $h = (Nf^{-1}) \bmod d$, with $0 \le g < f$, $0 \le h < d$, we have that

$$xQ = g + uf$$
$$yQ = h + vd$$
$$xyQ = yg + (yu)f = xh + (xv)d$$

for some integers u,v. Now let us define

$$w = xv, \qquad z = yu.$$

We have the approximate bounds

$$|u| \simeq \left|\frac{xQ}{f}\right| \le \frac{2PKQ}{N} = 2K$$

$$|v| \simeq \left|\frac{yQ}{d}\right| \le \frac{2N}{df}$$

$$|z| = |yu| \le \frac{4PK}{f}$$

$$|w| = |xv| \le \frac{4PK}{d}$$

Now we try to solve the equation

$$-xh + yg + zf - wd = 0$$

with x, y, z, w bounded as above, and treating x, y, z, w as *independent* integers. Specifically, we do not use the fact that z is a multiple of y. As in the previous section, we have turned a quadratic equation into a linear equation by introducing new variables to replace the quadratic terms.

We go through the lattice basis reduction step in more detail than before, although the procedure is entirely analogous.

Create the matrix

$$M = \begin{bmatrix} -h & g & f & -d \\ 1/X & 0 & 0 & 0 \\ 0 & 1/Y & 0 & 0 \\ 0 & 0 & 1/Z & 0 \\ 0 & 0 & 0 & 1/W \end{bmatrix}$$

where $X \simeq 2Pf K/N$ is the approximate bound on $|x|$, and similarly $Y \simeq 2P/f$, $Z \simeq 4PK/f$, and $W \simeq 4PK/d$ are approximate bounds on $|y|$, $|z|$, $|w|$. (We do not know P, of course, but we use an order-of-magnitude estimate.) Use lattice basis reduction [LLL] on the column basis of M to find a column vector with small Euclidean norm. In fact the vector $M \times [x, y, z, w]^{\mathsf{T}}$ has small Euclidean norm, its entries being bounded by about (0,1,1,1,1), respectively, so that its norm is at most about 2.

In order to pick out this vector efficiently, we will require that not many vectors in the lattice have much smaller norm. (In fact we will restrict our consideration to those columns having 0 as first element. This can be done by increasing the weight of the first row relative to the other rows.) We ask how many *linearly independent* vectors (with first element 0) have norm smaller than 1; we break into cases, depending on the number of such linearly independent lattice elements. (The difference between norm 1 and norm 2 is a nuisance which will be handled case-by-case below.)

Case 1: Three or more elements. Suppose there are three independent vectors in the lattice, each with norm less than 1, and each having 0 as its first element. Represent each by

$$t_i = M \times [x_i, y_i, z_i, w_i]^{\mathsf{T}} = [0, x_i/X, y_i/Y, z_i/Z, w_i/W]^{\mathsf{T}}.$$

Then the vector

$$(x_2 y_3 - x_3 y_2)t_1 + (x_3 y_1 - x_1 y_3)t_2 + (x_1 y_2 - x_2 y_1)t_3$$

also lies in the lattice and has its first three components 0. Calling this vector

$$\hat{t} = M \times [0, 0, \hat{z}, \hat{w}]^{\mathrm{T}} = [0,0,0, \hat{z}/Z, \hat{w}/W]^{\mathrm{T}},$$

we see that

$$\hat{z}f - \hat{w}d = 0$$

while

$$|\hat{z}| < 6XYZ = 6\frac{2PfK}{N}\frac{2P}{f}\frac{4PK}{f} = \frac{96P^3K^2}{Nf}$$
$$< \frac{96}{256}d$$

from (5). Since d and f are relatively prime, this implies $\hat{z} = \hat{w} = 0$, and the three vectors were really not independent.

Put another way, if three linearly independent vectors are in the lattice, then the product of their norms exceeds 1. Either at least one of the vectors has norm much larger than 1 (in which case we only have two elements, as in Case 3 below), or all three have norm around 1, in which case there are a small number of lattice elements with norm less than 2, and these can be treated by exhaustion.

Case 2: One element. If only one basis element has norm smaller than 1, then that element (or its negative) must represent P:

$$P = xd + yf.$$

Any multiple of that basis element would yield a multiple of P.

Case 3: Two elements. The interesting case is when two basis elements have norm smaller than 1. As before, let the two basis elements be

$$t_i = M \times [x_i, y_i, z_i, w_i]^{\mathrm{T}} = [0, x_i/X, y_i/Y, z_i/Z, w_i/W]^{\mathrm{T}}.$$

Break into subcases based on the determinant

$$\Delta = x_1 y_2 - x_2 y_1.$$

Case 3a: Two elements, large determinant. If $|\Delta| > 1$, then we are restricted to a sublattice of the original lattice. We can set

$$\tilde{d} \stackrel{\text{def}}{=} x_1 d + y_1 f$$
$$\tilde{f} \stackrel{\text{def}}{=} x_2 d + y_2 f$$

and start over with smaller ranges $\tilde{X}$, $\tilde{Y}$ for the unknowns $\tilde{x}, \tilde{y}$, where

$$P = \tilde{x}\tilde{d} + \tilde{y}\tilde{f}.$$

This process cannot happen more than a few times (certainly no more than $\log N$, and probably only once) before reducing to a different case.

Case 3b: Two elements, determinant = 1. If the determinant is $\Delta = \pm 1$, we employ some new trickery. Assume for concreteness

$$\Delta = x_1 y_2 - x_2 y_1 = +1.$$

The solution

$$P = xd + yf$$

is some linear combination of the two basis elements, say

$$x = \tau_1 x_1 + \tau_2 x_2$$
$$y = \tau_1 y_1 + \tau_2 y_2$$
$$z = \tau_1 z_1 + \tau_2 z_2$$
$$w = \tau_1 w_1 + \tau_2 w_2$$

We know that $g.c.d.(\tau_1, \tau_2) = 1$, because this $g.c.d.$ is a divisor of P. Define

$$r = x_1 w_2 - x_2 w_1$$
$$s = z_1 y_2 - z_2 y_1$$

For the correct solution we know that x is a divisor of w. From

$$x_1 w - w_1 x = \tau_2 r$$
$$x_2 w - w_2 x = -\tau_1 r$$
$$g.c.d.(\tau_1, \tau_2) = 1$$

we find that x is a divisor of r.

Similarly y is a divisor of z and hence y is a divisor of s.

Since

$$-x_1h + y_1g + z_1f - w_1d = 0$$
$$-x_2h + y_2g + z_2f - w_2d = 0$$

we have

$$\begin{aligned} 0 &= x_1(-x_2h + y_2g + z_2f - w_2d) - x_2(-x_1h + y_1g + z_1f - w_1d) \\ &= (x_1y_2 - x_2y_1)g + (x_1z_2 - x_2z_1)f - (x_1w_2 - x_2w_1)d \\ &\equiv g - rd \pmod{f} \end{aligned}$$

Then

$$rd^2 \equiv (rd)d \equiv gd \equiv N \pmod{f}.$$

Similarly we have

$$sf^2 \equiv (sf)f \equiv hf \equiv N \pmod{d}.$$

So if we set

$$Q_{\text{trial}} = \frac{r}{x}d + \frac{s}{y}f,$$

recalling that r/x and s/y are integers, we find that

$$\begin{aligned} PQ_{\text{trial}} &= (xd + yf)\left(\frac{r}{x}d + \frac{s}{y}f\right) \\ &= rd^2 + sf^2 + \left(\frac{sx}{y} + \frac{ry}{x}\right)df \end{aligned}$$

is equivalent to N both mod f and mod d, and hence it is equivalent to $N \bmod (df)$. So the correct value of Q differs from Q_{trial} by a multiple of df.

We wish to show that Q_{trial} is much smaller than df, so that adding any nonzero multiple of df would make Q too large, implying that in fact $Q = Q_{\text{trial}}$. To this end, recall that $g.c.d.(\tau_1, \tau_2) = 1$, implying that either τ_1 or τ_2 is nonzero; assume for concreteness that $\tau_2 \neq 0$.

$$|Q_{\text{trial}}| = \left|\frac{r}{x}d\right| + \left|\frac{s}{y}f\right| \le \left|\tau_2{}^r\frac{d}{x} + \tau_2{}^s\frac{f}{y}\right|$$

$$\tau_2{}^r = x_1 w - w_1 x = x_1(x\,v) - w_1 x = x(x_1\,v - w_1)$$

$$|\tau_2{}^r| \le |x|\left(X\frac{2N}{df} + W\right) \le |x|\left(\frac{2PfK}{N}\frac{2N}{df} + \frac{4PK}{d}\right)$$

$$= \frac{8|x|PK}{d}$$

$$\tau_2{}^s = z_1 y - y_1 z = z_1 y - y_1(y\,u) = y(z_1 - y_1\,u)$$

$$|\tau_2{}^s| \le |y|(Z + Y(2K)) \le |y|\left(\frac{4PK}{f} + \frac{2P}{f}(2K)\right)$$

$$= \frac{8|y|PK}{f}$$

$$|Q_{\text{trial}}| \le \left|\tau_2{}^r\frac{d}{x}\right| + \left|\tau_2{}^s\frac{f}{y}\right| \le \frac{8|x|PK}{d}\frac{d}{|x|} + \frac{8|y|PK}{f}\frac{f}{|y|} = 8PK + 8PK$$

$$|Q_{\text{trial}}| \le 16PK$$

As long as the following condition is met:

$$df > 16PK + Q,$$

we will have achieved our goal: for Q to have the right size, it must be true that $Q = Q_{\text{trial}}$, without any added multiple of df.

Assume this to be the case: $Q = Q_{\text{trial}}$. Then we have

$$N = PQ = PQ_{\text{trial}} = rd^2 + sf^2 + \left(\frac{sx}{y} + \frac{ry}{x}\right)df$$

This is equivalent to a quadratic equation in (y/x):

$$(r)\left(\frac{y}{x}\right)^2 + \left(\frac{rd^2 + sf^2 - N}{df}\right)\left(\frac{y}{x}\right) + (s) = 0,$$

which can be solved over the rationals (that is, not mod N) and will yield a rational value for y/x. Reducing to lowest terms (because $g.c.d.(x, y) = 1$) we find the correct values for x, y, and hence P.

We note here that this case will occur if the hint was in fact a double hint:

$$S = A_1 Q + B_1 = A_2 P + B_2$$
$$|B_i| < K = N^{1/6}$$

The two roots of the quadratic equation will correspond to P and Q.

Remark. This special case could also be handled by noticing that

$$(S - B_1)(S - B_2) = S^2 - (B_1 + B_2)S + (B_1 B_2)$$

is a multiple of N, with unknown coefficients $(B_1 + B_2)$ and $(B_1 B_2)$ which are both small. Even if the B_i are allowed to range up to about $N^{1/3}$, this equation can be solved by minimizing the basis of the lattice formed by the columns of

$$\tilde{M} = \begin{vmatrix} N & S^2 & S & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/N^{1/3} & 0 \\ 0 & 0 & 0 & 1/N^{2/3} \end{vmatrix}$$

Multiple hints

If we are given several hints,

$$\begin{gathered} S_i = A_i Q + B_i \\ |B_i| < K \\ i = 1,2,\ldots, i_{\max} \end{gathered}$$

then we can use simultaneous diophantine approximation (see, for example, [Lag]) to the fractions

$$\frac{S_i}{N} = \frac{A_i}{P} + \frac{B_i}{N}.$$

If the hints were chosen at random (each A_i and B_i chosen independently and uniformly at random from its range), then a bound like

$$K \leq Q^{1 - (1/i_{\max})}$$

should suffice. The tricks of this paper do not seem to improve this estimate. In the case of a single hint ($i_{\max} = 1$), we were able to use the fact that N was an exact multiple of P, together with the approximation $S/N \simeq A/P$, to improve the estimates. With multiple hints, the expression

$$P = xd + yf$$

is replaced by an expression

$$P = \sum_{i=1}^{i_{\max}+1} x_i d_i,$$

and the equation

$$xQ \equiv Nd^{-1} \bmod f$$

has no useful analogue. Efficient use of multiple hints remains an open question.

References

[Cop] D. Coppersmith, "Factoring with a Hint," IBM Research Report RC 19905, January 16, 1995.

[Has] J. Hastad, "On Using RSA with Low Exponent in a Public Key Network," *Advances in Cryptology — CRYPTO '85,* Hugh C. Williams (editor), Springer-Verlag LNCS **218** (1986), 403-408.

[Lag] J.C. Lagarias, "Knapsack Public Key Cryptosystems and Diophantine Approximation," *Advances in Cryptology, Proceedings of Crypto 83,* David Chaum (editor), Plenum Press, 1984, 3-23.

[LLL] A.K. Lenstra, H.W. Lenstra and L. Lovasz, "Factoring Polynomials with Integer Coefficients," *Matematische Annalen* **261** (1982), 513-534.

[Odl] A.M. Odlyzko, personal communication.

[RS] R.L. Rivest and A. Shamir, "Efficient factoring based on partial information," *Advances in Cryptology — EUROCRYPT '85,* Lecture Notes in Computer Science, Vol. 219, Berlin: Springer-Verlag, 1986, pp. 31-34.

[RSA] R.L. Rivest, A. Shamir and L. Adleman, "A method for obtaining digital signatures and public key cryptosystems," *Commun. ACM* **21** (April 1978), 294-299.

[VZ] S.A. Vanstone and R.J. Zuccherato, "Short RSA Keys and Their Generation," *J. Cryptology* vol 8 no 2 (Spring 1995), 101-114.