The Cryptographic Security of Truncated Linearly Related Variables

Johan Hastad* and Adi Shamir**

Abstract

In this paper we describe a polynomial time algorithm for computing the values of variables $x_1, \ldots, x_k$ when some of their bits and some linear relationships between them are known. The algorithm is essentially optimal in its use of information in the sense that it can be applied as soon as the values of the x_i become uniquely determined by the constraints. Its cryptanalytic significance is demonstrated by two applications: breaking linear congruential generators whose outputs are truncated, and breaking Blum's protocol for exchanging secrets.

1. Introduction

The design and analysis of cryptographic schemes and protocols has been a very active research area in recent years. A large number of cryptosystems, signature schemes, key distribution schemes, key sharing schemes, secret exchange schemes, pseudo random bit generators, and a variety of protocols were proposed in the literature. However, progress in cryptographic research was accompanied by progress in cryptanalytic research, and some of these schemes and protocols were shown to be insecure. This game of proposing and breaking schemes might seem a bit strange, but in the absence of any techniques for proving lower bounds on the complexity of problems in NP, it is probably unavoidable.

To make research in cryptography less empirical, proponents of new schemes and protocols should carefully list all the assumptions they make about the computational complexity of various problems, and rigorously prove the security of their proposal under these assumptions. The proof becomes particularly useful when the security of the scheme is shown to be equivalent to the complexity of a single well studied problem such as factoring or discrete log. An alternative approach is to develop schemes which are not based on a specific problem but on any one-way function. The large class of possible implementations of these schemes makes them less vulnerable to attacks (but still leaves the possibility that such functions do not exist or that P =NP).

One common idea which is used in several of these schemes and protocols is to hide the values of certain variables by applying (regular or modular) arithmetic operations to them and then truncating the binary representations of the results. A simple scheme of this type (which was used extensively on the early computers) generates a pseudo random sequence of numbers by alternately squaring the previous n bit value and discarding the top and bottom $n/2$ bits of the $2n$-bit result.

The purpose of this paper is to analyze the security of such procedures in a general setting. The main result is a novel cryptanalytic technique which can solve many of the instances of truncated linearly related variables in polynomial time. We demonstrate the power of this technique by describing two applications:

1. It simplifies, generalizes, and puts in a clearer perspective the recent result by Frieze, Kannan and Lagarias [3] that truncated linear congruential generators are insecure.
2. It shows that an assumption in Blum's protocol for exchanging secrets [1] about the infeasibility of a certain computational task is not correct and hence the protocol is insecure. This example is particularly surprising since at first glance the protocol appears to be inherently quadratic while our techniques are inherently linear.

* Department of Mathematics, MIT, Cambridge, MA 02139. Partially supported by NSF grant 8413577-DCR.

** Department of Applied Mathematics, The Weizmann Institute of Science, Rehovot, Israel. Partially supported by NSF grant no. MCS-80-06938.

We believe that other applications of the technique will be found in the future.

2. Formulation of the Problem

Let m be a given number and let $x_1, \ldots, x_k$ be k unknown values in the range $0 \leq x_i < m$ which are related by l independent homogeneous linear equations (mod m):

$$\sum_{i=1}^{k} a_i^j x_i = 0 \pmod{m}, \quad j = 1, \ldots, l \tag{1}$$

The coefficients a_i^j and the modulus m are assumed to be known, but $l < k$ and thus the x_i's are not uniquly defined. However, we are given (or somehow obtain) certain bits of each x_i, and our goal is to combine this partial knowledge with the given linear relationships in order to compute the remaining bits of all the x_i's. Our main tools will come from the geometry of numbers. Let us recall some facts:

A *lattice* is defined to be the set of points

$$L = \{\vec{y} \mid \vec{y} = \sum_{i=1}^{k} a_i \vec{b}_i, a_i \in \mathbf{Z}\}$$

where $\vec{b}_i$ are linearly independent vectors in $\mathbf{R}^k$. The set $\vec{b}_i$ is called a *basis* and k is the *dimension* of the lattice. The determinant of a lattice is defined to be the absolute value of the determinant of the matrix whose rows are $\vec{b}_i$. Geometrically the determinant can be interpreted as the volume of the parallelepiped spanned by the basis vectors. Using this interpretation it is possible to prove that the determinant is equal to the inverse of the density of the lattice (where the density is the average number of lattice points per unit volume). This characterization shows that the determinant is independent of the choice of basis. We define a set of k linearly independent vectors in L inductively:

$\vec{u}_1$ is the shortest vector in L.

$\vec{u}_{j+1}$ is the shortest vector in L which is linearly independent of $\vec{u}_1, \ldots, \vec{u}_j$.

(Ties are broken in an arbitrary way.)

The lengths $\lambda_j = \| \vec{u}_j \|$ of these vectors are called the successive minima of the lattice.

Definition: A lattice $L \subset \mathbf{R}^k$ with determinant D is *regular with constant* c if $\lambda_k \leq cD^{\frac{1}{k}}$.

Generally it is easier to work with λ_1 than with λ_k. Thus we will rely on the following lemma to transform lower bounds for λ_1 to upper bounds for λ_2.

Lemma 1: If $\lambda_1 \geq cD^{\frac{1}{k}}$ then $\lambda_k \leq \gamma_k^{\frac{k}{2}} c^{-(k-1)} D^{\frac{1}{k}}$ where γ_k is Hermite's constant.

Proof: Minkowski's second theorem [2] tells us that

$$\prod_{i=1}^{k} \lambda_i \leq \gamma_k^{\frac{k}{2}} D$$

and therefore by using the lower bound for λ_i, $i = 1, \ldots, k-1$ we get the desired upper bound for λ_k. ∎

Although the value of Hermite's constant is not known for $k > 9$, we have in [2] the upper estimate $\gamma_k \leq k$. This estimate is not far from the actual value, since it is known that γ_k is of size $O(k)$ [10] . In fact most lattices will satisfy $\lambda_1 > \sqrt{\frac{k}{2e\pi}} D^{\frac{1}{k}}$ [5] . This also shows that most lattices will be regular with constants which are not too large.

Let us return to our problem. Recall equation (1). Let L be the lattice in $\mathbf{R}^k$ spanned by the l vectors $\vec{a}^j$ (the coefficients of the known modular relations) and the k vectors $m\vec{e}_i$ where $\vec{e}_i$ are the unit vectors along the coordinate axes.

Observe that L consists precisely of those vectors $\vec{v} \in \mathbf{R}^k$ which are known to satisfy

$$\sum_{i=1}^{k} v_i x_i = 0 \pmod{m}$$

The determinant of L is m^{k-l}. To see this we use the characterization of the determinant as the inverse of the density. To calculate the density observe that in a hypercube with side m there will be m^l points in the lattice and hence the density is m^{l-k} and therefore the determinant is m^{k-l}.

We are now ready to state the main theorem.

Theorem 1: Suppose that the lattice L defined above has determinant D and dimension k. Assume that L is regular with constant c and define $s = ((\log D)/k) + \frac{k}{2} + \frac{1}{2}\log k + \log c + 1$. Then if we are either given as inputs

(i) the s most significant bits of all the x_i. Or

(ii) the s least significant bits of all the x_i and m is odd. Then we can recover the x_i completly in polynomial time.

Proof: We use a three stage algorithm: First apply a lattice reduction algorithm to the lattice of known modular relations L to get get modular equations with small coefficients. Now use the known bits of the x_i to transform these equations to equations over the integers. From these equations over the integers recover the exact values of the x_i.

We apply the algorithm of Lenstra, Lenstra and Lovasz in [8] to the lattice of modular relations to obtain a

good basis. They prove the following:

Theorem: (LLL) There exits a polynomial time algorithm that on input L finds a basis $\vec{b}_i$ such that $\| \vec{b}_i \| \leq 2^{\frac{k}{2}} \lambda_i$.

By our assumption on the λ_i this means that we can find linearly independent vectors $\vec{w}^j$ such that

$$\| \vec{w}^j \| \leq 2^{\frac{k}{2}} \lambda_j \leq 2^{\frac{k}{2}} c D^{\frac{1}{k}}.$$

As $\vec{w}^j$ are in L

$$\sum_{i=1}^{k} w_i^j x_i = 0 \pmod{m}, \quad j = 1, \ldots, l$$

These equations can be regarded as equations over the integers by letting

$$\sum_{i=1}^{k} w_i^j x_i = d_j m$$

where the d_j's are integers.

From this point on the proof depends on whether condition (i) or condition (ii) of the theorem holds. Assume first condition (i) i.e., that we know the s most significant bits of all the x_i's.

Decompose x_i into $x_i^{(1)} + x_i^{(2)}$ where $x_i^{(1)}$ is known and $|x_i^{(2)}| < m/2^s$.

This means that

$$\sum_{i=1}^{k} w_i^j x_i = \sum_{i=1}^{k} w_i^j x_i^{(1)} + \sum_{i=1}^{k} w_i^j x_i^{(2)}.$$

The first sum is known, and the second sum can be bounded since by the definition of s

$$\sum_{i=1}^{k} w_i^j x_i^{(2)} \leq \| \vec{w}^j \| \cdot \| \vec{x}^2 \| \leq 2^{\frac{k}{2}} c D^{\frac{1}{k}} \sqrt{k} \frac{m}{2^s} < m/2$$

Since the uncertainty is less than $m/2$ and we know that the total sum is a multiple of m we can calculate it exactly and hence determine d_j. This finishes the treatment of case (i).

Assume now that condition (ii) holds instead. Then:

$$|d_j| \leq |\sum_{i=1}^{k} w_i^j x_i|/m \leq \|\vec{w}^j\| \cdot \|\vec{x}\|/m \leq 2^{\frac{k}{2}} c D^{\frac{1}{k}} \sqrt{k} \leq 2^{s-1}$$

If we consider the equation

$$\sum_{i=1}^{k} w_i^j x_i = d_j m \pmod{2^s}$$

everything is known except d_j. Since m is odd we can calculate d_j uniquely $\pmod{2^s}$. Finally the size estimates of the d_j's determine the d_j's over the integers. Once the d_j are determined all that remains are k linearly independent equations in k unknowns over the rationals and these can be easily solved in polynomial time. ∎

Remark: It is also possible to consider the case when we know a window of consecutive bits in the x_i's, but not necessarily the most or least significant bits. However in this case the analysis is harder, and the conditions on m turn out to be complicated. Thus we omit this analysis.

Remark: The leading term in s is $\log D/k$, which equals $(k-l)\log m/k$ by the definition of D. When the k unknown values x_i are related by l independent linear equations $\pmod{m}$, at least $(k-l)\log m$ bits are required to specify a particular solution. Consequently, the problem cannot be solved uniquely if fewer than $((k-l)/k)\log m$ of the bits of each x_i are known. When k is fixed our algorithm matches this information-theoretic bound up to an additive constant, and thus it is essentially optimal in its use of information. Note in addition that for any fixed k we can replace the LLL algorithm by Kannan's algorithm [6] and thus reduce s by $\frac{k}{2}$.

3. Cryptanalysis of Truncated Linear Congruential Generators

A linear congruential generator is based on the recurrence

$$x_{i+1} = a x_i + c \pmod{m}$$

in which a, c and m are known and the seed x_0 is secret. Plumstead [11] has shown that if the entire numbers are published we can start predicting this sequence after having been given a short initial segment even if a,c and m are unknown. The case when a,c and m are known but only some of the bits of the x_i's are published was first considered by Knuth [7] . The first polynomial time algorithm for this case was given by Frieze, Kannan and Lagarias [3] . They show that with high probability it is possible to recover the seed if at least 2/5 of the leading bits of three consecutive numbers are known, and claim (without proof) a similar result for any fixed fraction of the leading bits whenever m is squarefree. In this section we show that this stronger result follows directly from our general cryptanalytic technique. We also improve 2/5 to any fraction greater than 1/3 for a general m. The proof in the case where m is square free involves analyzing the same number of theoretic problems that were considered in the unpublished proof of Frieze et. al. but our use of lattices avoids several other complications they encountered.

In this section we consider the case when some of the most significant bits are published, but our technique applies to the other cases as well. Without loss of generality we can assume that $c = 0$ (otherwise we can use $z_i = x_{i+1} - x_i$ requiring one extra number to be seen). The k variables x_i are related by the following system of $k-1$ independent homogeneous equations:

$$a^{i-1}x_1 - x_i = 0 \pmod{m} \quad i = 2, \ldots, k$$

The lattice spanned by the $k-1$ coefficient vectors

$$(a^{i-1}, 0, \cdots, 0, -1, 0, \cdots, 0)$$

is the set of vectors

$$(\sum_{i=2}^{k} a^{i-1}v_i, -v_2, \cdots, -v_k)$$

for all possible choices of $v_2, \cdots, v_k$ in $\mathbf{Z}$. If we define

$$v_1 = -\sum_{i=2}^{k} a^{i-1}v_i$$

then an alternative characterization of this lattice is the set of all vectors $\vec{v} = (v_1, \cdots, v_k)$ in $\mathbf{Z}^k$ for which

$$\sum_{i=1}^{k} a^{i-1}v_i = 0.$$

When the vectors $m\vec{e}_i$ are added to the basis of this lattice, we can change each v_i by arbitrary multiples of m, and thus the final lattice with which we have to deal is

$$L_a = \{\vec{v} \in \mathbf{Z}^k \mid \sum_{i=1}^{k} a^{i-1}v_i = 0 \pmod{m}\}.$$

The density of L_a is obviously $\frac{1}{m}$ and hence the determinant is m. Applying our general technique we get:

Theorem 2 Let m be squarefree, $\epsilon > 0$, and k be a given integer. Then knowledge of $\log m(\frac{1}{k}+\epsilon)+c_\epsilon$ leading bits of all the x_i suffices to compute the x_i completely in polynomial time for $1-O(m^{\frac{-\epsilon}{2}})$ of the possible coefficients a.

Remark: The fraction of bits that must be known can be made arbitrarily small, and this result is essentially optimal except for the presence of the ϵ.

The only nontrivial part of applying our general framework is the analysis of the lattice L_a. We have the following lemma:

Lemma 2: When m is squarefree, the lattice L_a satisfies the estimate $\lambda_k \leq c_\epsilon D^{\frac{1}{k}+\epsilon}$ for $1 - O(m^{\frac{-\epsilon}{2}})$ of the possible coefficient a.

Proof: A vector $\vec{v}$ is in L_a precisely when a satisfies the polynomial equation

$$\sum_{i=1}^{k} v_i a^{i-1} = 0 \pmod{m}$$

where $\vec{v} = (v_1, v_2 \ldots, v_k)$. We want to estimate the cardinality of the set of a which satisfy such an equation with small coefficients. To do this we estimate the number of such equations and the number of solutions to each equation. Let us make this precise.

Suppose that $m = \prod_{i=1}^{s} p_i$ where the p_i are different primes. We want to estimate the cardinality of the following set:

$$F(t) = \{a \mid \exists \|\vec{v}\| \leq m^t, \ \sum_{i=1}^{k} v_i a^{i-1} = 0\}$$

Estimating the size of $F(t)$ involves counting the number of lattice points in spheres. This is a complicated problem[9] . For this reason we trade constants for clarity and replace the sphere by the larger cube:

$$|v_i| < m^t, \quad i = 0, \ldots, k-1$$

Let d be the product of r of the prime factors of m. If $gcd(gcd(\vec{v}), m) = d$ the number of solutions to the equation

$$\sum_{i=1}^{k} v_i a^{i-1} = 0 \pmod{m}$$

is estimated by $d(k-1)^{s-r}$. The reason for this is that we have at most $k-1$ solutions modulo the primes which do not divide d. We therefore need to count the number of vectors that satisfy the condition $gcd(gcd(\vec{v}), m) = d$.

Lemma 3: If $d \mid m$ then the number of integer vectors satisfying $gcd(gcd(\vec{v}), m) = d$ and $|v_i| < h$, $i = 1, \ldots, k$ is less than $(\frac{3h}{d})^k$.

Proof: Dividing the v_i and m by d shows that it is enough to prove the lemma when $d = 1$. In this case the estimate follows from an estimate of the total number of points in the region considered. ∎

Therefore we have:

$$|F(t)| < \sum_{d|m} k^{s-r} d(\frac{3m^t}{d})^k < k^s 3^k m^{tk} \sum_{d|m} d^{1-k}$$

By elementary calculation $k^s = O(m^{\epsilon_1})$ for any $\epsilon_1 > 0$ and the same is true for the sum (the sum is of course bounded by a constant if $k > 2$). Putting $t = \frac{1}{k}(1-\epsilon)$ and $\epsilon_1 = \epsilon/4$ and using the proof of lemma 1 gives lemma 2.

Now theorem 2 follows from theorem 1 by the same proof. ∎

The above proof works for m which are almost squarefree. Define a number m to be δ-squarefree if $m = \prod_{i=1}^{s} p_i^{e_i}$ and $\prod_{i=1}^{s} p_i^{e_i-1} \leq m^{\delta}$. Then we have the following theorem.

Theorem 3: Let m be δ-squarefree $\epsilon > 0$ and k a constant. Then knowledge of $\log m(\frac{1}{k} + \epsilon + \delta) + c_\epsilon$ leading bits of all the x_i suffices to compute the x_i completely in polynomial time for $1 - m^{\frac{-\epsilon}{2}}$ of the possible coefficients a.

The proof is essentially the same as for theorem 2.

For $k = 3$ we can prove the following theorem, in which m need not be squarefree:

Theorem 4: For any m and given $\epsilon > 0$, knowledge of $\log m(\frac{1}{3} + \epsilon) + c_\epsilon$ leading bits of x_1, x_2 and x_3 suffices to compute the numbers in polynomial time for all a except a set of cardinality $m^{1-\frac{\epsilon}{2}}$.

As we have seen the hard part of the proof will be to count the number of solutions to second degree equations when the modulus is highly composite. To fix notation let $A(x) = a_0 + a_1 x + a_2 x^2$.

We want to estimate the size of the following set:

$$F(t) = \{x \mid \exists \|\vec{a}\| < m^t, A(x) = 0)\}$$

Lemma 4: For any $\epsilon > 0$ it is true that $|F(t)| \leq O(\max(m^{3t+\epsilon}, m^{.5+t+\epsilon}))$

Assume first that $gcd(gcd(\vec{a}), m) = 1$. Suppose $m = \prod_{i=1}^{s} p_i^{e_i}$ where p_i are different primes. If all the e_i are unity the theorem follows from theorem 2. We are therefore interested in the number of solutions to quadratic equations modulo prime powers. Let us remind ourselves that the discriminant of a quadratic polynomial is $4a_0a_2 - a_1^2 = D$ and that the discriminant is 0 iff the polynomial has a double root. We have the following (well known?) lemma.

Lemma 5: If p does not divide $gcd(\vec{a})$ then the number of solutions to $A(x) = 0 \pmod{p}^e$ is bounded by $\min(p^{\lfloor \frac{e}{2} \rfloor}, 2p^r)$ where r is the largest integer such that $D = 0 \pmod{p}^r$.

Proof: We can assume that the highest degree coefficient is not divisible by p since in that case the equation only has at most one solution. If $r \geq e$ $A(x)$ factors as $t(x+a)^2 \pmod{p^e}$ and we have $p^{\lfloor e/2 \rfloor}$ solutions. If $r < e$ then by the condition on the discriminant $A(x)$ does not have any quadratic factors $\pmod{p^{r+1}}$ but is a square $\pmod{p^r}$. We have two possibilities: either A does not factor $\pmod{p^{r+1}}$ or factors into different linear factors. In the first case we get no solutions and in the second we can lift the factorization $\pmod{p^{r+1}}$ to a factorization $\pmod{p^e}$ which can be written $t(x+a)(x+b)$ where $a = b \pmod{p^r}$ while $a \neq b \pmod{p^{r+1}}$. It is not hard to see that the number of solutions in this case is $2p^r$. The number of solutions will therefore be $\min(p^{\lfloor e/2 \rfloor}, 2p^r)$. ∎

It remains to estimate the frequency with which the condition in lemma 5 is satisfied.

Lemma 6: Given $\epsilon > 0$ and $d < m^{2t}$ the number of $\|\vec{a}\| \leq m^t$ that satisfy $D(A) = 0 \pmod{d}$ is $O(m^{3t+\epsilon}/d)$.

Proof: $4a_0a_2 - a_1^2 = 0 \pmod{d}$ splits into the $O(m^{2t}/d)$ equations $4a_0a_2 - a_1^2 = kd, |k| \leq \frac{m^{2t}}{d}$ over the integers. For each fixed a_1 the equation becomes $4a_0a_2 = c$. If $c \neq 0$ then this equation has as many solutions as divisors of c but it is not hard to see that this number is $O(m^\epsilon)$ since $c < m^{2t}$. $c = 0$ gives $4m^t$ possibilities for a and c but in this case b is determined by k and hence the total number of solutions is $O(m^{3t+\epsilon}/d)$. ∎

Back to the proof of lemma 4:

$|F(t)| \leq \sum_{d|m, d \leq m^{2t}} c_1 d \frac{m^{3t+\epsilon}}{d} + m^{1/2}$ (number of A:s such that $D = 0$) $\leq$

(number of divisors of m) $c_1 m^{3t+\epsilon} + c_2 m^{1/2+t+\epsilon} \leq c_3 m^{3t+\epsilon_1} + c_2 m^{1/2+t+\epsilon}$

We would like to remove the restriction that $gcd(gcd(\vec{a}), m) = 1$. Look at the set of a which satisfy $gcd(gcd(\vec{a}), m) = d$. Dividing the equation by d we get a polynomial with coefficients of size m^t/d and a modulus m/d. Looking at the corresponding F-set we see that it has cardinality at most $O(m^{3t+\epsilon}/d^3)$ and that each solution will have exactly d images when lifted to $\pmod{m}$. This leaves us with the bound $O(m^{3t+\epsilon}/d^2)$. The total count will therefore be:

$$m^{3t+\epsilon} \sum_{d|m} d^{-2} \leq O(m^{3t+\epsilon})$$

Lemma 4 implies theorem 4 by a proof similar to that of theorem 1.

4. Cryptanalysis of Blum's Protocol for Exchanging Secrets

Blum's protocol[1] was one of the first results which dealt with the issue of simultaneity in sequential processes. It enables two parties A and B to exchange the factorization of their published moduli m_A and m_B (which are the products of two large primes) in a fair and verifiable way. Let $n = \log m_A = \log m_B$ be the size parameter.

The protocol is symmetric, and the two parties alternately perform the following steps:

1. Choose k random numbers $y_1, \ldots, y_k$ and send their squares modulo the opponent's modulus to the other party.

2. Extract the four square roots modulo your own number of each number y_i^2 received from the other party. This is possible since you know the factorization. Now write the $4k$ square roots in a $4k \times n$ binary matrix where the least significant bits are in the last column.

3. Send the i-th column of the matrix to the other party. (For $i = 1, \ldots, n$).

The idea behind this procedure is that by having one of the square roots of y_i^2 at hand it is possible to check that what you receive is correct information. If B wants to cheat he can guess which square root A has and send that square root and its negation correctly while the rest are unrelated bits. The probability that such a technique would not be detected by A is 2^{-k}. The security of the protocol depends on the inability of the parties to factor efficiently before all (or almost all) the columns have been exchanged. Blum stated this as an assumption in the proof of correctness of his protocol. We show that this assumption is incorrect.

Theorem 5: There is a polynomial time algorithm which when given as input k random numbers y_i and $n/k + c_{k,\epsilon}$ most significant bits of all square roots of the $y_i^2 \pmod m$ factors with probability $1 - \epsilon$. The probability is taken over the probability distribution over the y_i and the running time is polynomial in n but not in k.

Proof: For each i, the four square roots of y_i^2 can be denoted by y_i, $-y_i$, $x_i = ry_i$ and $-ry_i \pmod m$ where r is a square root of 1 different from ± 1. Since y_i is known, the first two values can be easily identified. The selection of x_i from the remaining two values when only some of its bits are known requires guessing. However, for fixed k the total number of guesses is a constant (2^{k-1}). Observe that if we can recover any of the x_i we can factor m since $gcd(x_i - y_i, m)$ will be nontrivial.

Since the unknown values of the x_i are fixed multiples of the known values of the y_i, we can relate them by $k-1$ modular linear equations:

$$y_i x_1 - y_1 x_i = 0 \pmod m, \quad i = 2, \cdots k.$$

The lattice spanned by the $k-1$ coefficient vectors

$$(y_i, 0, \cdots, 0, -y_1, 0, \cdots, 0)$$

is the set of vectors

$$(\sum_{i=2}^{k} y_i v_i, -y_1 \; v_2, \cdots, -y_1 v_k)$$

for all possible choices of $v_2, \cdots, v_k$ in $\mathbf{Z}$. If we define

$$y_1 v_1 = -\sum_{i=2}^{k} y_i v_i$$

and add the k vectors $m\vec{e_i}$ to the basis, then this lattice becomes the set of all the vectors of the form $y_1 \vec{v}$ for $\vec{v} \in \mathbb{Z}^k$ which satisfy

$$\sum_{i=1}^{k} y_i v_i = 0 \pmod m.$$

Since y_1 is almost certainly invertible $\pmod m$, we can eliminate it and obtain the following characterization of the lattice:

$$L_y = \{\vec{v} \in \mathbb{Z}^k \mid \sum_{i=1}^{k} y_i v_i = 0 \pmod m\}.$$

To apply our general technique, we only have to prove that L_y is regular for almost all choices of $\vec{y}$.

Lemma 7: Given $\epsilon > 0$, with probability $1-\epsilon$ for random $y_1, \ldots, y_k$, the equation

$$\sum_{i=1}^{k} c_i y_i = 0 \pmod m$$

cannot be solved with c_i not all 0 satisfying $|c_i| < d_{k,\epsilon} m^{\frac{1}{k}}$.

Proof: The lemma is true for all m but we prove it only in the case we need it, namely when m is the product of two large primes. For each fixed set of c_i the proportion of the y_i which satisfy the linear equation is $\frac{1}{m}$. The number of sets of c_i is approximately $\omega_k d_{k,\epsilon}^k m$ where ω_k is the volume of the unit ball in $\mathbf{R}^k$. By making $d_{k,\epsilon}$ small we can make the proportion of y_i satisfying any equation in the set as small as we please. This implies the regularity of L_y, and theorem 5 is proved. ∎

Thus we can conclude that for almost all y_i we can recover the x_i when $n/k + c_k$ columns have been exchanged. As pointed out above this enables us to factor by calculating $gcd(x_i - y_i, m)$. The original protocol can therefore be broken by somebody who only deviates from the protocol by stopping early and using our algorithm—there is no need to control the choice of random bits or to lie to the other party.

Remark: The alternative protool in which the columns of the matrices are exchanged in reverse order (from least significant bits to most significant bits) is just as insecure, since m is odd.

Discussion:

Blum's paper is one of the best examples of thorough and responsible research in cryptography. Due to the extraordinary care with which Blum listed his assumptions, it is easy to trace the source of the problem to the following redundancy condition ([1] , pp. 187):

"Alice cannot use the $100 \times k$ most significant bits, $y_1^k, \cdots, y_{100}^k$, to split m_B any better than she can use just the k most significant bits y_1^k".

Blum's paper axiomatically assumes that this condition is true, and rigorously proves the security of the protocol modulo this assumption (and a few others). We did not find any error in Blum's proofs — we just showed that this assumption was too strong. This should not be taken as a sign of sloppiness since progress in the design of efficient algorithms is relentless and unpredictable. What is really important is to distinguish between facts, assumptions and proofs, and to identify all the possible sources of insecurity. Blum actually exceeds these minimal criteria. While he expresses his personal belief that

"these assumptions are not unreasonable, the protocol is hardier even than the assumptions that underie our proof, and consequently, one would be hard put to find a flaw in that protocol",

he also prepares alternatives to the protocol should something happen to his assumptions:

"While the redundancy condition is the most demanding of our assumptions, the protocol can be modified to work without this redundancy condition at all $\cdots$ If this is done, the applications at the end of this paper can still go through as before".

Blum's modification is based on a multi-moduli variant of his protocol and a redefinition of the secrets which are exchanged by the parties. Another possible modification is to ask the parties to exchange fewer columns from their matrices and to use our algorithm to factor the moduli at an earlier stage. None of these variants seems to be vulnerable to the cryptanalytic attack proposed in this paper, but its existence demonstrates once more the extremely delicate nature of proofs of security in cryptography.

Acknowledgements

We would like to thank Oded Goldreich and Silvio Micali for starting this research and for contributing generously to its progress. The first author would like to thank Shafi Goldwasser for her constant interest and helpful comments.

References

[1] M. Blum, "How to Exchange (Secret) Keys", *ACM Transactions on Computer Systems 1*, **2**, May 1983, 175–193.

[2] J.W.S. Cassels, "Geometry of Numbers", Springer Verlag 1959.

[3] A.M. Frieze, R. Kannan and J.C. Lagarias "Linear Congruential Generators do not Produce Random Sequences", FOCS 1984 480–484.

[4] S. Goldwasser, S. Micali and C. Rackoff "The Knowledge Complexity of Interactive Protocols". Unpublished manuscript.

[5] J. Hastad "On the Shortest Vector in Random Lattices". Manuscript in preparation.

[6] R. Kannan "Improved Algorithms for Integer Programming and Related Lattice Problems", STOC 1983, 193–206.

[7] D.E. Knuth "The Art of Computer Programming", Vol. 2 Addison Wesley 1980.

[8] A.K. Lenstra, H.W. Lenstra and L. Lovasz, "Factoring Polynomials With Integer coefficients", *Matematische Annalen*, **261**, (1982), 513–534.

[9] J.E. Mazo and A.M. Odlyzko, Lattice points in High-dimensional spheres. Paper in prerparation.

[10] Minkowski "Diskontinuitätsbereich für Arithmetische Äquivalenz", *Journal für Mathematik*, **129**, 1905, 220–274.

[11] J.B. Plumstead, "Inferring a Sequence Generated by a Linear Congruence", FOCS 1982, 153–159.