SIAM J. COMPUT.
Vol. 17, No. 2, April 1988

005

RECONSTRUCTING TRUNCATED INTEGER VARIABLES SATISFYING LINEAR CONGRUENCES*

ALAN M. FRIEZE†, JOHAN HASTAD‡, RAVI KANNAN§, JEFFREY C. LAGARIAS¶ AND ADI SHAMIR‖

Abstract. We propose a general polynomial time algorithm to find small integer solutions to systems of linear congruences. We use this algorithm to obtain two polynomial time algorithms for reconstructing the values of variables $x_1, \cdots, x_k$ when we are given some linear congruences relating them together with some bits obtained by truncating the binary expansions of the variables. The first algorithm reconstructs the variables when either the high order bits or the low order bits of the x_i are known. It is essentially optimal in its use of information in the sense that it will solve most problems almost as soon as the variables become uniquely determined by their constraints. The second algorithm reconstructs the variables when an arbitrary window of consecutive bits of the variables is known. This algorithm will solve most problems when twice as much information as that necessary to uniquely determine the variables is available. Two cryptanalytic applications of the algorithms are given: predicting linear congruential generators whose outputs are truncated and breaking the simplest version of Blum's protocol for exchanging secrets.

Key words. pseudorandom numbers, linear congruential generators, lattice basis reduction algorithm, cryptography

AMS(MOS) subject classifications. 11T71, 11K45

1. Introduction. The basic techniques of cryptanalysis are methods for solving various sorts of reconstruction problems. Given diverse kinds of information about a cryptosystem together with some enciphered messages, the cryptanalyst wishes to combine this information to recover the original plaintext messages, which is the *message reconstruction problem.* The cryptanalyst often accomplishes this by solving the possibly harder problem of finding the key used by the encipherer, which is the *key reconstruction problem.* From this perspective a general method of cryptanalysis is one that solves a wide class of reconstruction problems. General reconstruction methods serve as building blocks in the cryptanalysis of complex cryptosystems and also serve to set limitations of the possible design of secure cryptosystems.

This paper studies a reconstruction problem arising from the combination of two basic operations used in the design of pseudorandom number generators and cryptosystems. These two operations consist of modular arithmetic operations used as a computationally efficient way to "mix" the values of certain variables and the (nonlinear) operation of truncating the binary representation of the results. A simple scheme of this type (which was used extensively on early computers) generates a pseudorandom sequence of integers by alternately squaring the previous n-bit value and discarding the top and bottom $n/2$ bits of the $2n$-bit result. A related scheme is that of using the high-order bits of a linear congruential sequence, which is generally called a *truncated linear congruential pseudorandom number generator.* This was proposed by Knuth [10].

* Received by the editors October 15, 1985; accepted for publication (in revised form) May 18, 1987.
† Carnegie-Mellon University, Pittsburgh, Pennsylvania 15213.
‡ Massachusetts Institute of Technology, Cambridge, Massachusetts 02139. The work of this author was supported by an IBM fellowship.
§ Carnegie-Mellon University, Pittsburgh, Pennsylvania 15213. The work of this author was supported in part by National Science Foundation grant ACS-8418392.
¶ AT&T Bell Laboratories, Murray Hill, New Jersey 07974.
‖ Weizmann Institute of Science, Rehovot, Israel.

The problem we consider is that of reconstructing a set $(x_1, x_2, \cdots, x_k)$ of integer variables given two sorts of information about them. First we are given a set of l modular equations

$$\sum_{j=1}^{k} a_{ij}x_j \equiv c_i \pmod{M} \quad \text{for } 1 \leqq i \leqq l, \tag{1.1}$$

that the variables satisfy. We assume that the a_{ij}, c_i and M are known, and that the unknowns x_i satisfy the bounds

$$0 \leqq x_j < M \quad \text{for } 1 \leqq j \leqq k. \tag{1.2}$$

Second, we are given side information about some of the bits of the x_j, which consists of knowledge of blocks of consecutive binary digits of the variables x_j. More precisely, this partial information consists of knowledge of

$$y_j \equiv \left[\frac{x_j}{2^{l_2}}\right] \pmod{2^{l_1}} \quad \text{for } 1 \leqq j \leqq k. \tag{1.3}$$

Here $[z]$ denotes the greatest integer of a real number z. In this case we know a fraction δ of the bits of each x_i, where δ is given by

$$\delta = \frac{l_1}{[\log_2 M]}$$

whenever $l_1 + l_2 \leqq [\log_2 M]$. The interesting case for cryptanalysis occurs when the number of equations l is less than the number of unknowns k. In this case the congruences (mod M) taken by themselves constrain the variables x_j (mod M) but do not determine them uniquely.

The first question to deal with is: how much side information is needed to make unique reconstruction possible? We may obtain an information-theoretic lower bound for the amount of side information required as follows. Suppose $2^{n-1} \leqq M < 2^n$ so that the x_j are n bit integers, and that we know a block of δn bits of each x_j. Now l modular equations (mod M) with side conditions $0 \leqq x_i < M$ can normally be used to eliminate l of the variables. The remaining $k-l$ variables contain $(k-l)n$ unknown bits of information which must be uniquely determined by the $k\delta n$ bits given by the variables y_j. Consequently we infer that a necessary condition for unique reconstruction is that $k\delta n \geqq (k-l)n$, which is

$$\delta \geqq 1 - \frac{l}{k}. \tag{1.4}$$

In fact it turns out that the fraction $\delta = 1 - (l/k) + \varepsilon$ of the highest-order bits of each x_j suffice to guarantee unique reconstruction for the overwhelming majority of systems (1.1)–(1.3), in a sense made precise in § 2. However, there does exist a small minority of such systems which require a larger δ than given by (1.4) to guarantee unique reconstructibility.

The main result of this paper is a general technique for solving this type of problem. It uses lattice basis reduction ideas and is guaranteed to run in polynomial time, but is not always guaranteed to produce a reconstruction. In the problems we are considering there are two major cases. The first case applies when the truncated variables y_j either consist of the highest-order bits of the x_j, or consist of the lowest-order bits of

the x_j and M is odd. We show in Theorem 2.1 that our general algorithm succeeds for "most" instances when

$$\delta > 1 - \frac{l}{k} + \varepsilon,$$

where

$$\varepsilon = \frac{c_k}{\log M}.$$

We will quantify what "most" means in § 2. The constants c_k are of size $O(k)$. Our second case applies to systems with arbitrarily truncated variables y_j. To be effective it requires that twice as much side information be given as that needed to guarantee uniqueness, that is,

$$\delta > 2\left(1 - \frac{l}{k}\right) + \varepsilon,$$

and it then works for "most" systems (Theorem 2.3). There is a small fraction of problems on which the algorithms fail, and there is a smaller fraction of exceptional problems for which unique reconstruction is not possible.

We demonstrate the usefulness of these reconstruction procedures with two applications:

(1) We show that truncated linear congruential pseudorandom number generators are cryptographically insecure in most cases.

(2) We show that the simplest version of Blum's protocol [1] for exchanging secrets is insecure. We remark that Blum suggests other implementations of this protocol which do not seem vulnerable to the attack described here.

The two applications are described in §§ 3 and 4, respectively, where we give an analysis of our algorithm applied to these cases. The main technical difficulty is to analyze the behavior of the lattices arising in the special problems.

Some of the results of this paper appeared in preliminary form in Frieze, Kannan and Lagarias [5] and Hastad and Shamir [6].

2. Reconstructing truncated variables satisfying linear congruences. Let M be a given modulus and $x_1, \cdots, x_k$ unknown values in the range $0 \leqq x_j < M$ satisfying l independent linear congruences (mod M):

$$\sum_{j=1}^{k} a_{ij} x_j \equiv c_i \pmod{M} \quad \text{for } 1 \leqq i \leqq l, \tag{2.1}$$

where $l \leqq k$. The coefficients a_{ij} and c_i and the modulus M are assumed to be known. We are given (or somehow obtain) certain bits y_j of each x_j where

$$y_j \equiv \left[\frac{x_j}{2^{l_2}}\right] \pmod{2^{l_1}}, \tag{2.2}$$

and our goal is to combine this partial knowledge with the given linear relationship to compute the remaining bits of all the x_j's. Our main tools to do this will come from the geometry of numbers, see [3], [4]. Let us recall some facts. A (full rank) *lattice* L is defined to be the set of points

$$L = \left\{ \mathbf{y} : \mathbf{y} = \sum_{i=1}^{k} n_i \mathbf{b}_i,\ n_i \in \mathbb{Z} \right\}$$

where the $\mathbf{b}_i$ are linearly independent vectors in $\mathbb{R}^k$. The set $\{\mathbf{b}_i: 1 \leqq i \leqq k\}$ is called a *basis* of L and k is the *dimension* of the lattice. The *determinant* $d(L)$ of a lattice L is defined to be the absolute value of the determinant of a matrix whose rows are the $\mathbf{b}_i$. 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 the ith *successive minimum* $\lambda_i = \lambda_i(L)$ of a lattice L to be the smallest radius r such that the sphere or radius r around 0 in $\mathbb{R}^k$ contains i linearly independent points of L in it or on its boundary. We will be interested in lattices whose successive minima are roughly the same size.

We will be interested in bounding λ_k from above. To do so, we use the *dual lattice* L^* which is defined to be

$$L^* = \{\mathbf{y} \mid \langle \mathbf{y}, \mathbf{x} \rangle \in \mathbb{Z} \quad \text{for all } \mathbf{x} \in L\}.$$

It is well known that $d(L^*) = d(L)^{-1}$. A classical result asserts that $\lambda_1 \lambda_k^* \leqq k!$ [3, p. 371], and a recent result of Lagarias, Lenstra and Schnorr [12, Thm. 4.4] shows that $\lambda_1^* \lambda_k \leqq k^2/6$ for $k \geqq 7$, where λ_1^* denotes the length of the shortest vector of L^*, and $\lambda_1^* \lambda_k \leqq k^2$ for all k. Thus a lower bound for λ_1^* gives the desired upper bound for λ_k.

The idea to use the dual lattice was suggested to us by C. P. Schnorr [17]. Our original method gave constants having a worse dependence on k.

Let us return to our problem. Let $L(\mathbf{a}, M)$ be the lattice in $\mathbb{R}^k$ spanned by the l vectors $\mathbf{a}_i = (a_{i,1}, \cdots, a_{i,k})$ (the coefficients of the known modular relations) in (2.1) and by the k vectors $M\mathbf{e}_i$ where $\mathbf{e}_i$ are the unit vectors along the coordinate axes. We will use this lattice in our algorithm, and the performance of the algorithm depends on properties of this lattice. Observe that the dual lattice $L^* = L^*(\mathbf{a}, M)$ is

$$\left\{ \frac{1}{M}\mathbf{y} \,\middle|\, \langle \mathbf{y}, \mathbf{a}_i \rangle \equiv 0 \pmod{M} \text{ for } 1 \leqq i \leqq l \right\},$$

where $\langle \mathbf{y}, \mathbf{a}_i \rangle$ is the Euclidean inner product.

Let us start by giving the theorem which will be our main tool.

THEOREM 2.1. *The system of modular equations*

$$\sum_{j=1}^{k} a_{ij} x_j \equiv c_i \pmod{M}, \qquad i = 1, 2, \cdots, l$$

has at most one solution $\mathbf{x} \in \mathbb{Z}^k$ *satisfying the bound*

$$\|\mathbf{x}\| \leqq M\lambda_k^{-1} 2^{-(k/2)-1}, \tag{2.3}$$

where λ_k *is the largest successive minimum of the lattice* $L(\mathbf{a}, M)$. *If the* a_{ij}, c_i *and* M *are known then there is a polynomial time algorithm that either finds* $\mathbf{x}$ *or proves that no such* $\mathbf{x}$ *exists.*

Proof. We use a three-stage algorithm. First, we apply a lattice basis reduction algorithm to the lattice $L = L(\mathbf{a}, M)$ of known modular relations to get modular relations with small coefficients. Second, we use size constraints on the x_j to transform these equations to equations over the integers. Third, we use these equations over the integers to recover the exact values of the x_j.

We apply the lattice basis reduction algorithm of Lenstra, Lenstra and Lovász [13] to the lattice L of modular relations to obtain a good basis. They prove the following result.

THEOREM 2.2. *There exists an algorithm, the L^3-algorithm, that when given as input a basis $\{\mathbf{b}_i: 1 \leqq i \leqq k\}$ of an integer lattice $L \subseteq \mathbb{R}^k$ finds a basis $\{\mathbf{b}_i^*: 1 \leqq i \leqq k\}$ such that*

$$\|\mathbf{b}_i^*\| \leqq 2^{k/2}\lambda_i(L) \quad \text{for } 1 \leqq i \leqq k. \tag{2.4}$$

This algorithm always halts in $O(n^6(\log B)^3)$ bit operations, where $B^2 = \sum_{i=1}^k \|\mathbf{b}_i\|^2 = \sum_{i=1}^k \sum_{j=1}^k b_{ij}^2$.

We do not have a basis for the lattice $L(\mathbf{a}, M)$ but we can obtain one as follows. Using a Hermite normal form reduction algorithm (see [8]) we obtain in polynomial time an integer matrix V in $GL(k+l, \mathbb{Z})$ such that

$$\begin{bmatrix} \mathbf{w}_1' \\ \mathbf{w}_k' \\ 0 \\ \vdots \\ 0 \end{bmatrix} = V \begin{bmatrix} \mathbf{a}_1 \\ \vdots \\ \mathbf{a}_l \\ M\mathbf{e}_1 \\ \vdots \\ M\mathbf{e}_k \end{bmatrix},$$

where the matrix on the left is in Hermite normal form and $\{\mathbf{w}_i': 1 \leqq i \leqq k\}$ is a basis of $L(\mathbf{a}, M)$. Now the L^3-algorithm applied to this basis produces an L^3-reduced basis $\{\mathbf{w}_i: 1 \leqq i \leqq k\}$ and a unimodular matrix U in $GL(k, \mathbb{Z})$ such that

$$\begin{bmatrix} \mathbf{w}_1 \\ \vdots \\ \mathbf{w}_k \end{bmatrix} = U \begin{bmatrix} \mathbf{w}' \\ \vdots \\ \mathbf{w}_k' \end{bmatrix}$$

and

$$\|\mathbf{w}_i\| \leqq 2^{k/2}\lambda_k, \qquad 1 \leqq i \leqq k.$$

Combining these steps we obtain an integer matrix Y such that

$$\begin{bmatrix} \mathbf{w}_1 \\ \vdots \\ \mathbf{w}_k \end{bmatrix} = Y \begin{bmatrix} \mathbf{a}_1 \\ \vdots \\ \mathbf{a}_l \\ M\mathbf{e}_1 \\ \vdots \\ M\mathbf{e}_k \end{bmatrix} \tag{2.5}$$

(Alternatively the L^3-algorithm can be adapted to work on a set of generators of a lattice and produce (2.5) directly.) Now by multiplying (2.5) on the right by $\mathbf{x}$ and reducing (mod M) using (2.1) we obtain modular relations with small coefficients:

$$\sum_{j=1}^k w_{ij}x_j \equiv c_i' \pmod{M}, \qquad 1 \leqq i \leqq k. \tag{2.6}$$

Note that, although we started with l modular equations in (2.1), we have now obtained a full set of k modular relations which are independent over the integers.

To perform the second stage of the algorithm we observe that

$$\left| \sum_{j=1}^k w_{ij}x_j \right| \leqq \|\mathbf{w}_i\| \, \|\mathbf{x}\| < 2^{k/2}\lambda_k M \lambda_k^{-1} 2^{-(k/2)-1} < \frac{M}{2}.$$

Thus if we choose c_i' to satisfy $|c_i'| < M/2$ we know that

$$\sum_{j=1}^k w_{ij}x_j = c_i', \qquad 1 \leqq i \leqq k,$$

holds over the integers.

Finally we solve this system of k linearly independent equations in k unknowns. □

Let us see how to use Theorem 2.1 if the variables are not small but some of the bits are known. Define for convenience

$$s_0 = \log \lambda_k + \frac{k}{2} + \frac{1}{2}\log k + 1, \tag{2.7}$$

where $\lambda_k = \lambda_k(L(\mathbf{a}, M))$.

COROLLARY 2.3. *The system of modular equations*

$$\sum_{j=1}^{k} a_{ij}x_j \equiv c_i \pmod{M}, \qquad i = 1, 2, \cdots, l$$

has at most one solution $\mathbf{x}$ *in which either of the following conditions holds*:

(i) *The* s_0 *most significant bits of each* x_j *are specified.*

(ii) *The* s_0 *least significant bits of each* x_j *are specified, and* M *is odd.*

If the a_{ij}, c_i *and* M *are known there is a polynomial time algorithm that either finds* $\mathbf{x}$ *or proves that no such* $\mathbf{x}$ *exists.*

Proof. To prove case (i) we just observe that $x_i = x_i^{(1)} + x_i^{(2)}$ where $x_i^{(1)}$ are the known most significant bits and $|x_i^{(2)}| \leqq M\lambda_k^{-1}2^{-(k/2)-1}\sqrt{k^{-1}}$. Substituting the known $x_i^{(1)}$ we come into position to use Theorem 2.1.

For the case (ii) write $x_i = 2^{s_0}x_i^{(1)} + x_i^{(2)}$ where $x_i^{(2)}$ are known and $x_i^{(1)}$ satisfies the same size bounds. Since M is odd $(2^{s_0})^{-1} \pmod{M}$ is defined, and after multiplying the equations by $(2^{s_0})^{-1} \pmod{M}$ we can use Theorem 2.1. □

Note that the algorithm of Theorem 2.1 can be applied without knowing the value of λ_k or whether or not the bound (2.3) holds. To explain this, we associate with any basis $\{\mathbf{b}_i : 1 \leqq i \leqq k\}$ of a lattice L in $\mathbb{R}^k$ the quantity

$$\Delta_k(\mathbf{b}_1, \cdots, \mathbf{b}_k) = \max_{1 \leqq i \leqq k} (\|\mathbf{b}_i\|). \tag{2.8}$$

Then by the proof of Theorem 2.1 the algorithm succeeds whenever

$$s_0 \geqq \log \Delta_k(\mathbf{b}_1^*, \cdots, \mathbf{b}_k^*) + \tfrac{1}{2}\log k + 1, \tag{2.9}$$

where $\{\mathbf{b}_i^* : 1 \leqq i \leqq k\}$ is the L^3-reduced basis of the lattice $L(\mathbf{a}, M)$ obtained in the algorithm. The bound (2.9) can be checked during the algorithm.

When can the algorithms of Corollary 2.2 be expected to succeed? This depends on the value of λ_k, and to get an idea how large it usually is we estimate it in the case where the modulus M is a fixed prime and the coefficients a_{ij} of the modular relations (2.1) are drawn independently from the uniform distribution on $[0, M-1]$.

THEOREM 2.4. *Let* p *be prime. For the* p^{kl} *possible systems* $\mathbf{A}$ *of modular equations*

$$\sum_{j=1}^{k} a_{ij}x_j \equiv 0 \pmod{p} \quad \textit{for } 1 \leqq i \leqq l$$

arising by choosing

$$0 \leqq a_{ij} < p \quad \textit{for } 1 \leqq i \leqq l \textit{ and } 1 \leqq j \leqq k$$

at least $(1 - \varepsilon - O(p^{-l/k}))p^{kl}$ *of these give rise to lattices* $L(\mathbf{A}, p)$ *which have*

$$\lambda_k < 5k^{3/2}\varepsilon^{-1/k}p^{1-l/k}. \tag{2.10}$$

Proof. We will use the previously mentioned result by Lagarias, Lenstra and Schnorr [12] that $\lambda_1^*\lambda_k \leqq k^2$ for all $k \geqq 1$.

We estimate the probability that L^* contains a short vector. We know that pL^* is the integer lattice $\{\mathbf{y} | \langle \mathbf{y}, \mathbf{a}_i \rangle \equiv 0 \pmod p;\ 1 \leqq i \leqq l\}$. Take a sphere S centered around 0 of radius R where $R < p$. For any nonzero point $\mathbf{z}$ in S the probability that $\mathbf{z} \in pL^*$ is p^{-l}. Thus the probability that any point inside S is in pL^* is bounded by $S_k(R) \cdot p^{-l}$, where $S_k(t)$ counts the number of lattice points in a k-dimensional sphere of radius t centered at the origin. Since $S_k(t) = V_k t^k + O(kV_k t^{k-1})$ with $V_k = \pi^{k/2}/\Gamma(k/2+1)$ as $t \to \infty$ we conclude that if we choose $R = (\pi^{k/2}/\Gamma(k/2+1))^{-1/k} \varepsilon^{1/k} p^{l/k}$ then $p^{-l} S_k(R) = \varepsilon + O(p^{-l/k})$ as $p \to \infty$. Hence we conclude that with probability $1 - \varepsilon - O(p^{-l/k})$ the inequality

$$\lambda_1^* \geqq \frac{1}{p} \left(\frac{\pi^{k/2}}{\Gamma(k/2+1)} \right)^{-1/k} \varepsilon^{1/k} p^{l/k} \geqq \frac{1}{5} \sqrt{k}\, \varepsilon^{1/k} p^{l/k-1}$$

holds, and thus

$$\lambda_k \leqq 5k^{3/2} \varepsilon^{-1/k} p^{1-l/k}.$$

This completes the proof. □

A slightly weaker result than Theorem 2.4 can be proved to hold for all moduli M. We omit the details.

We may now infer that the algorithm of Corollary 2.3 succeeds in most cases for a random system (2.1) whenever the number s of known bits exceeds the information bound $(1-(l/k)) \log M$ by a small amount. Indeed for M a prime p, for most lattices $L(\mathbf{a}, p)$ the bound (2.7) implies that this happens if s satisfies

$$s \geqq \left(1 - \frac{l}{k}\right) \log p + \frac{k}{2} + 2 \log k + \frac{1}{k} |\log \varepsilon| + 3, \tag{2.11}$$

which exceeds the information bound by a constant depending only on the dimension k and desired failure rate ε.

In cryptanalytic applications, the set of problems (2.1) that arises may be distributed in an entirely different way than the uniform distribution studied in Theorem 2.4. For this reason, in §§ 3 and 4 we separately analyze the distributions of λ_k arising in our two applications. However in the absence of other information, the bound (2.11) is a useful heuristic to use.

We now describe and analyze our second algorithm, which applies to the set of modular equations

$$\sum_{j=1}^{k} a_{ij} x_j \equiv c_i \pmod M, \qquad 1 \leqq i \leqq l \tag{2.12}$$

where we are given an arbitrarily located window of bits for each x_i. We suppose that the window of s truncated bits is from bit w to bit $w+s-1$, i.e.,

$$x_i = x_i^{(1)} + 2^w x_i^{(2)} + 2^{w+s} x_i^{(3)}$$

where $x_i^{(1)} < 2^w$, $x_i^{(2)} < 2^s$ and $x_i^{(3)} < M 2^{-w-s}$, and $x_i^{(2)}$ is assumed to be known. Thus the unknown is $x_i^{(1)} + 2^{w+s} x_i^{(3)}$.

To use Theorem 2.1 we want to transform (2.12) to an equation with small unknowns. To do this we find a which satisfies

$$|a| \leqq M 2^{-w-s/2} \quad \text{and} \quad |a 2^{w+s} \pmod M| \leqq 2^{w+s/2}. \tag{2.13}$$

Such an a always exists and we can find it in polynomial time using the result of Lenstra [14] that there is a polynomial time algorithm for solving integer programs in

a fixed number of variables. This is because (2.13) can be written as the integer program in three variables (a, y_1, y_2) given by

$$-M2^{-w-(s/2)} \leqq a - My_1 \leqq M2^{-w-(s/2)},$$
$$-2^{w+(s/2)} \leqq a2^{w+s} - My_2 \leqq 2^{w+(s/2)},$$
$$0 < a < M.$$

Multiplying the equation (2.12) by a and using the unknowns

$$z_i = ax_i^{(1)} + a2^{w+s}x_i^{(3)}$$

we obtain the modular equation

$$\sum_{j=1}^{k} a_{ij}z_i \equiv c_i' \pmod{M}, \qquad 1 \leqq i \leqq l,$$

where the quantities

$$c_i' = a\left(c_i - 2^w \sum_{j=1}^{k} a_{ij}x_i^{(2)}\right)$$

are known. Now we know by (2.13) that

$$|z_i| \leqq |a||x_1^{(i)}| + |a2^{w+s} \pmod{M}||x_i^{(3)}| \leqq M2^{-s/2+1}.$$

Thus provided that

$$s > 2\log \lambda_k + k + \log k + 4$$

holds that we can apply Theorem 2.1 and find z_i. If $(a, M) = 1$ then all that remains is to compute $a^{-1}z_i \pmod{M}$. If M is prime then $(a, M) = 1$ is guaranteed to hold and we have proved the following result.

COROLLARY 2.5. *Suppose M is prime and we are given a known system of modular equations*

$$\sum_{j=1}^{k} a_{ij}x_j \equiv c_i \pmod{M}, \qquad 1 \leqq i \leqq l,$$

and a window of s truncated bits of each x_i consisting of bits w to $w+s-1$ where

$$s > 2\log \lambda_k + k + \log k + 4.$$

Then there is a polynomial time algorithm that either finds a solution **x** *to the modular equations matching the truncated window data, or else proves that no such* **x** *exists.*

However in the case of a general modulus M we cannot assume that $(a, M) = 1$. There might not even exist an a such that $(a, M) = 1$ which has the desired properties. To get around this problem we will use a different approach. We will prove a result for general M which depends on Diophantine approximation properties of the number $M/2^{w+s}$. Define for a real number θ the quantity

$$\alpha(\theta, x) = \min_{\substack{m,n \in \mathbb{Z} \\ 1 \leqq n \leqq x}} |n\theta - m|. \tag{2.14}$$

We have the following result.

THEOREM 2.6. *Suppose that we are given a known system of modular equations*

$$\sum_{j=1}^{k} a_{ij}x_j \equiv c_i \pmod{M}, \qquad i = 1, 2, \cdots, l$$

and that we are given a window of s bits of each of the variables x_i whose largest bit is the $(w+s)$th bit. If

$$s \geqq \log \lambda_k + \frac{k}{2} + \left| \log \alpha\left(\frac{M}{2^{w+s}}, \sqrt{k}\, 2^{(k/2)+1}\lambda_k\right)\right| + \frac{1}{2}\log k + 1 \tag{2.15}$$

then the x_i are uniquely determined and can be found in polynomial time.

Proof. As in the proof of Theorem 2.1 we get a system of equations

$$\sum_{j=1}^{k} w_{ij}x_j \equiv c_i' \pmod{M}, \qquad 1 \leqq i \leqq k,$$

where $\|\mathbf{w}_i\| \leqq 2^{k/2}\lambda_k$. Over the integers we can write this as

$$\sum_{j=1}^{k} w_{ij}x_j = c_i' + d_i M, \qquad 1 \leqq i \leqq k, \tag{2.16}$$

where we know by the bound for the $\mathbf{w}_i$ that $|d_i| \leqq \sqrt{k}\, 2^{k/2}\lambda_k$. Using $x_j = x_j^{(1)} + x_j^{(2)}2^w + x_j^{(3)}2^{w+s}$ where $x_j^{(2)}$ are known we get

$$\sum_{j=1}^{k} w_{ij}x_j^{(1)} \equiv c_i'' + d_i M \pmod{2^{w+s}}, \tag{2.17}$$

where the integers c_i'' are known. Since $|x_j^{(1)}| \leqq 2^w$ the bounds on $\mathbf{w}_i$ imply that

$$\left|\sum_{j=1}^{k} w_{ij}x_j^{(1)}\right| \leqq 2^{(k/2)+w}\lambda_k\sqrt{k}. \tag{2.18}$$

Using (2.17) we obtain

$$c_i'' + d_i M \equiv z_i \pmod{2^{w+s}},$$

where $|z_i| \leqq \sqrt{k}\, 2^{(k/2)+w}\lambda_k$. Now we know that each d_i satisfies the integer program in two variables $(\tilde{d}_i, t_i)$ given by

$$-\sqrt{k}\, 2^{(k/2)+w}\lambda_k \leqq c_i' + \tilde{d}_i M + 2^{w+s}t_i \leqq \sqrt{k}\, 2^{(k/2)+w}\lambda_k, \tag{2.19a}$$

$$-\sqrt{k}\, 2^{k/2}\lambda_k \leqq \tilde{d}_i \leqq \sqrt{k}\, 2^{k/2}\lambda_k. \tag{2.19b}$$

We can find a solution $(\tilde{d}_i, t_i)$ to this integer program in polynomial time by Lenstra [14] (see also [7]). We claim that all solutions of (2.19) have $\tilde{d}_i = d_i$. Suppose not, and let $d_i^{(1)}, d_i^{(2)}$ be distinct solutions. Then their difference $\bar{d}_i = d_i^{(1)} - d_i^{(2)}$ satisfies

$$|\bar{d}_i M + 2^{w+s}(t_i^{(1)} - t_i^{(2)})| \leqq \sqrt{k}\, 2^{(k/2)+w+1}\lambda_k, \tag{2.20a}$$

$$|\bar{d}_i| \leqq \sqrt{k}\, 2^{(k/2)+1}\lambda_k. \tag{2.20b}$$

Then

$$\left|\frac{\bar{d}_i M}{2^{w+s}} - r\right| \leqq \sqrt{k}\, 2^{(k/2)-s+1}\lambda_k$$

for some integer $r = t_i^{(1)} - t_i^{(2)}$. Using the bound on $\bar{d}$ and the definition of $\alpha(M/2^{w+s}, \sqrt{k}\, 2^{(k/2)+1}\lambda_k)$ if $\bar{d} \neq 0$ this yields

$$\alpha\left(\frac{M}{2^{s+w}}, \sqrt{k}\, 2^{(k/2)+1}\lambda_k\right) \leqq \sqrt{k}\, 2^{(k/2)-s+1}\lambda_k.$$

This inequality contradicts the bound for s in (2.15). Consequently $\tilde{d}_i = d_i$. Hence we have found d_i by solving (2.19). Now we determine the x_j by solving the invertible linear system (2.16). □

To estimate the useful range of Theorem 2.6 we need information about the quantities $\alpha(M/2^{s+w}, \sqrt{k}\,2^{(k/2)+1}\lambda_k)$. Dirichlet's theorem for Diophantine approximation (see [3, p. 165], [9]) asserts that for all real θ one has

$$\alpha(\theta, x) \leqq \frac{1}{x}$$

for integer x, and it is known that for most pairs (θ, x) one has $\alpha(\theta, x)$ of size about $1/x$. Hence one expects that for most triples (M, w, s) one has $\alpha(M/2^{s+w}, \sqrt{k}\,2^{(k/2)+1}\lambda_k) \sim k^{-1/2}2^{-k/2}\lambda_k^{-1}$ and hence that the bound (2.15) is about twice that necessary to uniquely determine the variables x_i. The loss of efficiency of this algorithm in the information-theoretic sense arises in stage 2 of the algorithm. We are given a window of s truncated bits. The effect of the low-order bits y_j'' is inflated by the coefficients w_{ij} of the reduced basis and in the integer program (2.19) they destroy the information in the bottom $\log \lambda_k$ bits of the "window." Since we need about $\log \lambda_k$ information bits to recover the input, the window must contain this many undestroyed bits of information, so it must have at least $2 \log \lambda_k$ bits, i.e., its efficiency is halved.

3. Cryptanalysis of truncated linear congruential pseudorandom number generators. A linear congruential pseudorandom number generator is based on the recurrence

$$x_{i+1} \equiv ax_i + c \pmod{M}. \tag{3.1}$$

Several kinds of reconstruction problems relating to linear congruential generators have been studied previously. In the case where the parameters (a, c, M) are *unknown* and $\{x_i: 1 \leqq i \leqq k\}$ are known, J. Boyar [2] shows that one can start predicting subsequent values of the sequence with high accuracy given a short initial segment. Her method is to find parameters $(\hat{a}, \hat{c}, \hat{M})$ consistent with the available data (in polynomial time) and to extrapolate the sequence using these parameters. If a later disagreement occurs, the values $(\hat{a}, \hat{c}, \hat{M})$ are changed to remain consistent with the new data. She shows that at most $O(\log M)$ disagreements can ever occur, using this procedure. Knuth [11] considered problems arising when only truncated high-order bits y_i of the generator are known. He supposed that $M = 2^n$ is known and that the parameters (a, c) are unknown, and he gave an attack which, when given $\{y_i: 1 \leqq i \leqq k\}$ where $y_i \equiv [x_i/2^l] \pmod{M}$, will usually reconstruct the parameters (a, c) and seed x_0 in $O(n^2 2^{2l}/k^2)$ steps. This running time bound is exponential time, as may be seen for example in the case when half of the bits are truncated and when the number k of values y_i observed is small. Reeds [15], [16] was the first to study linear congruential generators from a cryptographic viewpoint. In [16] he studied a cryptosystem which in its simplest version enciphers the plaintext P_i as

$$E_i \equiv y_i + P_i \pmod{256},$$

where $y_i = [257x_i/M]$ and $x_i \equiv ax_{i-1} \pmod{M}$. He showed how to break it in a reasonable time when both the modulus $M = 2^{31} - 1$ and multiplier $a = 7^5$ are *known*, using a partially known plaintext attack. His attack appears to take exponential time for general parameters (M, a, c).

We consider here the situation in which the modulus M and multiplier a are *known*, the constant term c is *unknown* and a segment y_i of truncated high-order bits

$$y_i = \left[\frac{x_i}{2^l}\right] \quad \text{for } 1 \leqq i \leqq k \tag{3.2}$$

of the linear congruential generator are given as data. We give a polynomial time reconstruction procedure and prove that it succeeds on nearly all problems in which sufficient data is available to permit unique reconstruction (Theorem 3.1), provided the modulus M is squarefree. We also prove a similar result which applies to *all* moduli M, provided that any fraction greater than one third of the bits of the original x_i is given as input (Theorem 3.5). In our analysis for simplicity we treat only the case that high-order truncated bits y_i are given as data, but our technique applies to the cases where the low-order truncated bits are given, or where an interior window of truncated bits is given. In the interior window case we would achieve only 50 percent efficiency in the use of the available information.

We will show that unique reconstruction of the sequence x_k is usually possible in the case that the parameter $c=0$. The case $c\neq 0$ is different. In the case $c\neq 0$ we set $x_i'=x_{i+1}-x_i$ and $y_i'=y_{i+1}-y_i$ and observe that x_i' satisfies the recurrence

$$x_{i+1}'\equiv ax_i' \pmod{M}$$

with $c=0$ and y_i' is essentially a truncated version of x_i'. Now the methods of this section will show that x_i' can usually be uniquely reconstructed. However, x_i' does not determine the sequence x_i since x_i and x_i+d for any d will give the same x_i' and both are generated by linear congruential generators. In fact for small d the two sequences $\{x_i\}$ and $\{x_i+d\}$ will usually have the same s most significant bits, and so it is impossible to uniquely reconstruct the original $\{x_i\}$ in this case. What we *can* do in the general case that $c\neq 0$ is to *predict* future values of the truncated sequence $\{y_i\}$ with great accuracy, using the s most significant bits of x_1 together with the uniquely reconstructed sequence $\{x_i'\}$.

Now we consider the case where the parameter $c=0$. The k unknowns $\{x_i: 1\leqq i\leqq k\}$ satisfy

$$x_{i+1}\equiv ax_i \pmod{M}$$

and consequently are related by the following system of $k-1$ independent homogeneous congruences:

$$a^{i-1}x_1-x_i\equiv 0 \pmod{M} \quad \text{for } 2\leqq i\leqq k. \tag{3.3}$$

Since we are given the high-order bits y_i of the x_i as input, we have exactly a problem of the kind analyzed in Corollary 2.2. In this case $L(\mathbf{a}, M)=L_a$ is the lattice consisting of all vectors $(\nu_1,\cdots,\nu_k)\in\mathbb{Z}^k$ satisfying (3.3), which has as a basis the vectors

$$\begin{aligned}
\mathbf{b}_1&=(M,0,0,\cdots,0),\\
\mathbf{b}_2&=(a,-1,0,\cdots,0),\\
\mathbf{b}_3&=(a^2,0,-1,\cdots,0),\\
&\vdots\\
\mathbf{b}_k&=(a^{k-1},0,0,\cdots,-1).
\end{aligned}$$

The determinant $D=D(L_a)$ is given by

$$D(L_a)=M. \tag{3.4}$$

The analysis of the size of $\lambda_k(L_a)$ is the only problem in applying Corollary 2.2. In the case where M is squarefree we are able to prove that λ_k is small for most L_a.

THEOREM 3.1. *For squarefree $M > c(\varepsilon, k)$ there is an exceptional set $E(M, \varepsilon, k)$ of multipliers of cardinality $|E(M, \varepsilon, k)| \leqq M^{1-\varepsilon}$ such that for any multiplier not in $E(M, \varepsilon, k)$ the following is true. The x_i are uniquely determined by knowledge of the $(1/k+\varepsilon)\log M + c(k)$ leading bits of all $\{x_i: 1 \leqq i \leqq k\}$, where*

$$c_k = \frac{k}{2} + (k-1)\log 3 + \frac{7}{2}\log k + 2.$$

Furthermore, there is an algorithm which runs in time polynomial in $\log M + k$ which finds the x_i.

Remarks. (1) The number of bits needed for unique reconstruction is essentially optimal on information theory grounds, except for the presence of the ε.

(2) Some sort of exceptional set $E(m, \varepsilon, k)$ is necessary because $a = 1$ is always a "bad" multiplier. In the case $a = 1$, all the observed truncated bits y_i will be equal to the high-order bits of the seed x_0, and one never gets any information about the low-order bits of the seed. (Of course we can *extrapolate* future values of the generator very well in this case.) There are usually other multipliers a in the exceptional set $E(M, \varepsilon, k)$ defined below, though they are in general not easy to characterize.

(3) The proof actually shows that we could take ε to be a constant times $1/\log\log M$ as $M \to \infty$.

Proof. Our object is to apply Theorem 2.1, and our only problem is to bound the number of lattices L_a which have a large λ_k. We define the exceptional set

$$E(M, \varepsilon, k) = \{L_a: \lambda_k(L_a) > 2k^3 3^{k-1} M^{(1/k)+\varepsilon}\}$$

and our object will be to show that for squarefree $M \geqq c(\varepsilon, k)$ there are at most $M^{1-\varepsilon}$ lattices L_a in the exceptional set $E(M, \varepsilon, k)$.

We will study the dual lattice L_a^*, which is generated by $1/M(1, a, a^2, \cdots, a^{k-1})$ and the unit vectors $\mathbf{e}_i$, $i = 1, \cdots, k$. For notational simplicity let us study ML_a^*. A short vector in ML_a^* corresponds to an integer t such that $\{ta^i: 0 \leqq i \leqq k-1\}$ are all small (mod M). For a fixed R we are interested in estimating the size of the set

$$S_R = \{a \mid \exists t, 1 \leqq t < M, |ta^i (\text{mod } M)| < R \quad \text{for } 0 \leqq i \leqq k-1\}.$$

Define for d dividing M the sets

$$S_{R,d} = \{a: \exists t, 1 \leqq t < M, (M, t) = d, |ta^i (\text{mod } M)| < R \quad \text{for } 0 \leqq i \leqq k-1\}.$$

It is clearly true that $S_R = \bigcup_{d|M} S_{R,d}$. Let us first estimate the size of $S_{R,1}$.

LEMMA 3.2. *If $a \in S_{R,1}$ then a satisfies an equation*

$$\sum_{i=1}^{k} \nu_i a^{i-1} \equiv 0 \pmod{M} \tag{3.5}$$

with

$$|\nu_i| \leqq (2kR)^{1/(k-1)}. \tag{3.6}$$

Proof. By assumption we have t such that $(t, M) = 1$ and $|ta^i| < R$ for $0 \leqq i \leqq k-1$. Consider all linear combinations $\sum_{i=0}^{k-1} s_i ta^i$ with $0 \leqq s_i < (2kR)^{1/(k-1)}$ for $0 \leqq i \leqq k-1$. There are $(2kR)^{k/(k-1)}$ such combinations and the value of any such combination is bounded in absolute value by $kR(2kR)^{1/(k-1)}$. Thus by the pigeonhole principle there are two different sets of s_i's which give the same value. Subtracting the two expressions and dividing by t we get the desired solution to (3.5). □

To estimate the size of $S_{R,1}$ we must estimate how many numbers a satisfy an equation (3.5) with small coefficients. Let $M=\prod_{i=1}^{f} p_i$. We count the number of a's satisfying this equation having

$$d=\text{g.c.d.}\,(M, \nu_1 \cdots \nu_k). \tag{3.7}$$

If d is the product of g of the prime factors of M then an upper bound for the number of solutions is $d(k-1)^{f-g}$. The reason for this is that we have at most $k-1$ solutions to the congruence

$$\sum_{i=1}^{k} \nu_i a^{i-1} \equiv 0 \pmod{p_i}$$

for each of the $f-g$ primes p_i of M not dividing d. We next bound the number of vectors $\mathbf{v}=(\nu_1, \cdots, \nu_k)$ satisfying the condition (3.7), using the following lemma.

LEMMA 3.3. *If d divides M then the number of nonzero integer vectors satisfying*

$$\text{g.c.d.}\,(M, \nu_1, \cdots, \nu_k)=d$$

and $|\nu_i| \leqq T$ for $1 \leqq i \leqq k$ is less than $(3T/d)^k$.

Proof. Dividing all ν_i and M by d shows it is enough to prove the lemma for $d=1$. In this case the estimate follows from a trivial bound on the number of lattice points in the region considered. □

Combining the above results we get

$$|S_{R,1}| \leqq \sum_{d|M} k^{f-g} d \left(\frac{(2kR)^{1/(k-1)} \cdot 3}{d} \right)^k$$
$$< k^f (2kR)^{k/(k-1)} \cdot 3^k \sum_{d|M} d^{1-k}.$$

To simplify this further we use a well-known number-theoretic estimate valid for squarefree numbers M that shows that there is a constant c_0 such that for $M \geqq 20$ one has

$$f \leqq c_0 \frac{\log M}{\log \log M}.$$

Hence

$$k^f \leqq M^{(c_0 \log k)/(\log \log M)}$$

and

$$\sum_{d|M} d^{1-k} \leqq \sum_{d|M} 1 \leqq 2^f \leqq M^{(c_0 \log 2)/(\log \log M)},$$

giving

$$|S_{R,1}| < 3^k (2kR)^{k/(k-1)} M^{c_0 \log 2k/\log \log m}. \tag{3.8}$$

Next let us consider $S_{R,d}$ for $d>1$. Whether $a \in S_{R,d}$ only depends on $a \pmod{M/d}$. To be more precise $a \in S_{R,d}$ if and only if there exists an integer t with $1 \leqq t < M/d$, and $(t, M/d)=1$ with

$$|ta^i \pmod{M/d}| \leqq R/d, \qquad 0 \leqq i \leqq k-1.$$

Reasoning as in Lemma 3.2 we find that a solves an equation with small coefficients (mod M/d). Using that each solution (mod M/d) lifts to at most d solutions (mod M) we get

$$|S_{R,d}| < d3^k \left(\frac{2kR}{d}\right)^{k/(k-1)} M^{(c_0 \log 2k)/(\log\log M)}$$
$$= 3^k d^{-1/(k-1)} (2kR)^{k/(k-1)} M^{(c_0 \log 2k)/(\log\log M)}.$$

Thus we obtain

$$|S_R| = |\bigcup_d S_{R,d}| \leqq \sum_d |S_{R,d}| < 3^k (2kR)^{k/(k-1)} M^{(c_0 \log 2k)/(\log\log M)} \sum_{d|M} d^{-1/(k-1)}$$

$$\leqq 3^k (2kR)^{k/(k-1)} M^{(c_0 \log 2k)/(\log\log M)} 2^f \tag{3.9}$$

$$\leqq 3^k (2kR)^{k/(k-1)} M^{(c_0 \log 4k)/(\log\log M)}.$$

Now choose $R = M^{(k-1)/k} 3^{-(k-1)} (2k)^{-1}$. Then (3.9) yields for $M \geqq c(\varepsilon, k)$ that

$$|S_R| < M^{1-\varepsilon},$$

and that for all a not in S_R we have

$$\lambda_1(L_a^*) \geqq \frac{1}{2k} 3^{-(k-1)} M^{-1/k-\varepsilon}.$$

Then the inequality $\lambda_1^* \lambda_k \leqq k^2$ proved in [12] yields

$$\lambda_k(L_a) \leqq 2k^3 3^{k-1} M^{1/k+\varepsilon}. \tag{3.10}$$

Hence $E(M, \varepsilon, k) \subseteq S_R$ so $|E(M, \varepsilon, k)| \leqq M^{1-\varepsilon}$.

To complete the proof of Theorem 3.1, we choose

$$s \geqq \left(\frac{1}{k} + \varepsilon\right) \log M + c(k)$$

with $c(k) = k/2 + (k-1) \log 3 + \frac{7}{2} \log k + 2$, and apply Corollary 2.2, after observing that if $a \notin E(M, \varepsilon, k)$ then (3.10) implies that

$$s \geqq s_0 = \log \lambda_k + \frac{k}{2} + \frac{1}{2} \log k + 1. \qquad \square$$

This proof of Theorem 3.1 does not carry over very well to non-squarefree moduli M, the worst case being $M = p^e$ a prime power. The problem in that case is that polynomials (mod p^e) may have many roots, e.g.,

$$\sum_{i=1}^{k} \nu_i a^{i-1} \equiv 0 \pmod{p^e}$$

may have up to kp^{e-1} roots, and we get a much weaker estimate for $|S_R|$ in this case. We can still use this weaker bound to extend the proof of Theorem 3.1 to apply to moduli M which are almost squarefree. Define a number M to be *δ-squarefree* if

$$M = \prod_{i=1}^{f} p_i^{e_i} \quad \text{and} \quad \prod_{i=1}^{f} p_i^{e_i - 1} \leqq M^{\delta}.$$

Then we have the following result.

THEOREM 3.4. *Suppose that the modulus M is δ-squarefree and let the number of iterates k and a constant $\varepsilon > 0$ be given. Then there exists a constant $c_2(\varepsilon, \delta)$ such that*

for all such $M > c_2(\varepsilon, \delta)$ and all residues not in an exceptional set $E(\varepsilon, M)$ of cardinality at most $M^{1-\varepsilon}$ given knowledge of s leading bits of $\{x_i: 1 \leqq i \leqq k\}$ where

$$s > \left(\frac{1}{k} + \varepsilon + \delta\right) \log M + c(k)$$

suffices to determine the $\{x_i: 1 \leqq i \leqq k\}$ uniquely. Furthermore, there is an algorithm that runs in time polynomial in $\log M + k$ which always reconstructs all the x_i in this case.

The proof is essentially the same as that of Theorem 3.1, using the weaker bound for $|S_R|$, and we omit the details.

For the special case $k = 3$ we are able by a more careful argument to prove an essentially optimal bound valid for all moduli M.

THEOREM 3.5. *Given $\varepsilon > 0$, for any M the knowledge of $(\frac{1}{3} + \varepsilon) \log M + c(k)$ leading bits of x_1, x_2 and x_3 allows the recovery of x_1, x_2 and x_3 in polynomial time for all multipliers a except a set of cardinality $c(\varepsilon) M^{1-\varepsilon/2}$.*

Proof. As we have seen the hard part of the proof will be to count the number of solutions to second-degree congruences when the modulus is highly composite. To fix notation let $V(x) = \nu_0 + \nu_1 x + \nu_2 x^2$ and $\mathbf{v} = (\nu_0, \nu_1, \nu_2)$ with $\|\mathbf{v}\| = (\nu_0^2 + \nu_1^2 + \nu_2^2)^{1/2}$. We want to estimate the size of the following set:

$$F(\mu) = \{a: 1 \leqq a < M \text{ and } \exists \|\mathbf{v}\| \leqq M^{\mu} \text{ with } V(a) \equiv 0 \pmod M\}.$$

Assume first that g.c.d. $(\nu_0, \nu_1, \nu_2, M) = 1$. Suppose $M = \prod_{i=1}^{f} p_i^{e_i}$ where the p_i are distinct primes. We study the number of solutions to a quadratic equation modulo prime powers p^e. Let $D(V) = \nu_1^2 - 4\nu_0\nu_2$ denote the discriminant of the quadratic polynomial $V(x)$. Note that the polynomial has a double root (mod p) if and only if $D(V) \equiv 0 \pmod p$. We have the following lemma.

LEMMA 3.6. *If p does not divide* g.c.d. (ν_0, ν_1, ν_2) *then the number of solutions of*

$$\nu_2 x^2 + \nu_1 x + \nu_0 \equiv 0 \pmod{p^e}$$

is at most $2 \min (p^{\lfloor e/2 \rfloor}, p^{\lfloor r/2 \rfloor})$ *where r is the largest integer such that*

$$D(V) \equiv 0 \pmod{p^r}.$$

Proof. We can assume that the highest-degree coefficient of V is not divisible by p since otherwise the congruence has at most one solution.

Suppose now that there is at least one solution, so that V factors as $t(x+a)(x+b) \pmod{p^e}$ with $(t, p) = 1$. The discriminant $D(V) \pmod{p^e}$ is $t^2(a-b)^2$ and if s is the largest integer such that $a \equiv b \pmod{p^s}$ then $r \geqq \min(2s, e)$. We have two cases, depending on the size of $2s$.

If $2s < e$ then the solutions of the congruence are precisely those $x \equiv a \pmod{p^{e-s}}$ and those $x \equiv b \pmod{p^{e-s}}$. There are $2p^s$ solutions in this case.

If $2s \geqq e$ then the solutions of the congruence are exactly those $x \equiv a \equiv b \pmod{p^{\lceil e/2 \rceil}}$ and there are $p^{\lfloor e/2 \rfloor}$ such $x \pmod{p^e}$.

In both these cases the bound of the lemma holds. □

We next estimate the frequency with which the condition in Lemma 3.6 is satisfied.

LEMMA 3.7. *Given $\varepsilon > 0$ and $d < M^{2\mu}$ the number of $\|\mathbf{v}\| \leqq M^{\mu}$ that satisfy*

$$D(V) \equiv 0 \pmod d$$

is $O((1/d) M^{3\mu+\varepsilon})$.

Proof. The congruence $\nu_1^2 - 4\nu_0\nu_2 \equiv 0 \pmod d$ splits into the $O((1/d) M^{2\mu})$ equations

$$\nu_1^2 - 4\nu_0\nu_2 = kd \quad \text{for } |k| \leqq \frac{1}{d} M^{2\mu},$$

over the integers. For each fixed ν_1 the equation is of the form $4\nu_0\nu_2 = c$. If $c \neq 0$ this equation has as many solutions as divisors of c, and it is not hard to see that this number is $O(M^{\varepsilon})$ since $c \leqq M^{2\mu}$. The remaining case $c = 0$ gives $4M^{\mu}$ possibilities for ν_0 and ν_2 but in this case ν_1 is determined by k and hence the total number of solutions is $O((1/d)M^{3\mu+\varepsilon})$. □

Now we are able to estimate $F(\mu)$ in the general case.

LEMMA 3.8. *For any $\varepsilon > 0$ it is true that*

$$|F(\mu)| \leqq c(\varepsilon) \max (M^{3\mu+\varepsilon}, M^{1/2+(3\mu/2)+\varepsilon}).$$

Proof. First observe that by Lemma 3.6 if $(D(V), M) = d$ then the number of solutions to $V(x) = 0$ is bounded by $d2^f$, where f is the number of prime factors of M. The contribution $S_1(\mu)$ to $|F(\mu)|$ from $V(x)$ with g.c.d. $(\nu_0, \nu_1, \nu_2, M) = 1$ and $D(V) \neq 0$ can be estimated using Lemma 3.7 by

$$\begin{aligned} S_1(\mu) &\leqq \sum_{\substack{d \mid M \\ d \leqq M^{2\mu}}} c_1 d 2^f \left(\frac{1}{d} M^{3\mu+\varepsilon/2}\right) \\ &\leqq c_1 d(M) 2^f M^{3\mu+\varepsilon/2} \leqq c_3 M^{3\mu+\varepsilon}, \end{aligned} \tag{3.11}$$

where $d(M) = O(M^{f_0/(\log\log M)})$ denotes the number of divisors of M.

Now we let $S_d(\mu)$ count the set of a in $|F(\mu)|$ arising from polynomials $V(x)$ with g.c.d. $(M, \nu_0, \nu_1, \nu_2) = d$ and $D(V) \neq 0$. Dividing such an equation for $V(x)$ by d we get a polynomial $(1/d)V(x)$ with integer coefficients of size $(1/d)M^{\mu}$ and modulus M/d, and g.c.d. $(M/d, \nu_0/d, \nu_1/d, \nu_2/d) = 1$. Looking at the corresponding F-set $(\mathrm{mod}\,(M/d))$ we see by the argument giving (3.11) that the F-set has cardinality $O((1/d^3)M^{3\mu+\varepsilon})$ and furthermore that each solution to it $(\mathrm{mod}\, M/d)$ will lift to exactly d distinct solutions $(\mathrm{mod}\, M)$. This leaves us with the bound

$$S_d(\mu) = O\left(\frac{1}{d^2} M^{3\mu+\varepsilon}\right).$$

We need also to estimate the number of a that satisfy the congruence $V(x) \equiv 0$ $(\mathrm{mod}\, M)$ with $D(V) = 0$ and $\|\mathbf{v}\| < M^{\mu}$. If g.c.d. $(\nu_0, \nu_1, \nu_2, M) = d$ then the congruence is

$$\frac{\nu_2}{d} x^2 + \frac{\nu_1}{d} x + \frac{\nu_0}{d} \equiv 0 \quad \left(\mathrm{mod}\, \frac{M}{d}\right). \tag{3.12}$$

By the analysis of the second case in Lemma 3.6 this congruence has at most $2^{\omega(M/d)}(M/d)^{1/2}$ solutions $(\mathrm{mod}\, M/d)$, where $\omega(M/d)$ counts the number of distinct prime divisors of M/d. These lift to a total of at most $2^{\omega(M)}\sqrt{Md}$ solutions x $(\mathrm{mod}\, M)$. Since $d \leqq M^{\mu}$, we have at most $2^{\omega(M)} d(M) M^{1/2+\mu/2}$ solutions to (3.12). This is $O(M^{1/2+\mu/2+\varepsilon/2})$ for any fixed ε, as $M \to \infty$. Finally the number of polynomials V that satisfy $D(V) = 0$ and $\|\mathbf{v}\| < M^{\mu}$ is $O(M^{\mu+\varepsilon/2})$ for any fixed ε, by similar reasoning to that of Lemma 3.7.

The total count of solutions $F(\mu)$ therefore satisfies

$$\begin{aligned} |F(\mu)| &\leqq \sum_{d \mid M} S_d(\mu) + M^{1/2+\mu/2+\varepsilon/2} \#\{V : D(V) = 0\} \\ &\leqq c_3 M^{3\mu+\varepsilon} \left(\sum_{d \mid M} \frac{1}{d^2}\right) + c_2 M^{1/2+3\mu/2+\varepsilon} \\ &\leqq c_4 (M^{3\mu+\varepsilon} + M^{1/2+3\mu/2+\varepsilon}). \end{aligned}$$

This proves Lemma 3.8. □

Now Lemma 3.8 implies Theorem 3.5 by a proof similar to that of Theorem 3.1. □

4. Cryptanalysis of Blum's protocol for exchanging secrets. Blum's paper [1] was one of the first to deal with the issue of simultaneity in sequential processes. He proposed several versions of a protocol to enable two parties to exchange the factorizations of their two published moduli m_A and m_B, which are the products of two large primes in a fair and verifiable way. The simplest of these is as follows. 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 $y_1, \cdots, y_k$ 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 ith column of the matrix to the other party (for $i = 1, \cdots, n$).

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 cheating 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 in this protocol. We show that this assumption is incorrect.

THEOREM 4.1. *There is an algorithm which when given as input k random numbers y_i and the $n/k + c_{k,\varepsilon}$ most significant bits of all square roots of the $y_i^2 \pmod M$ factors M with probability $1 - \varepsilon$. The probability is taken over the probability distribution on the y_i and the running time of the algorithm 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, $-y_i$, $x_i \equiv ry_i$ and $-x_i \equiv -ry_i \pmod M$ where r is a square root of $1 \pmod M$ different from ± 1. Since y_i is known, we can easily pair y_i and $-y_i$ with their $N/k + c_{k,\varepsilon}$ most significant bits in the data received. However, we do not know how to pair the remaining two sets of most significant bits with x_i and $-x_i$, so we must guess. However, for fixed k the total number of guesses is the constant 2^k. When the correct guess is made we have paired each x_i with its $N/k + c_{k,\varepsilon}$ most significant bits. Observe that if we can recover any of the x_i we can factor M since g.c.d. $(x_i - y_i, M)$ will be nontrivial.

Since the unknown values of the x_i are fixed multiples of the known values of the y_i, they are related by $k-1$ modular linear equations:

$$y_i x_1 - y_1 x_i \equiv 0 \pmod M \quad \text{for } i = 2, \cdots, k.$$

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

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

together with the vectors $M\mathbf{e}_1, \cdots, M\mathbf{e}_k$ is the set of vectors

$$\left(\sum_{i=2}^{k} y_i \nu_i, -y_1 \nu_2, \cdots, -y_1 \nu_k \right) + (a_1 M, \cdots, a_k M)$$

for all possible choices of $\nu_2, \cdots, \nu_k$ and $a_1, \cdots, a_k$ in $\mathbb{Z}$. When y_1 is invertible $\pmod M$, as is usually the case, we obtain the following characterization of this lattice:

$$L_y = \left\{ \mathbf{v} \in Z^k \;\middle|\; \sum_{i=1}^{k} y_i \nu_i \equiv 0 \pmod M \right\},$$

and L_y has determinant M. To apply our general technique, we only have to bound $\lambda_k(L_y)$ from above for almost all choices of $\mathbf{y}$.

As usual we do this by bounding from below the length of the shortest vector in the dual lattice L_y^*. In this case L_y^* is the lattice spanned by $(1/M)(y_1, \cdots, y_k)$ and the unit vectors $\mathbf{e}_i$. We use the following lemma.

LEMMA 4.2. *Let $\varepsilon > 0$ and k be given. Then there is a positive constant $d_{k,\varepsilon}$ such that for a random drawing of integers $(y_1, \cdots, y_k)$ in $(\mathbb{Z}/M\mathbb{Z})^k$ where $M = p_1 p_2$ and $\min(p_1, p_2) \geqq \frac{1}{2} d_{k,\varepsilon} M^{1/k}$, then with probability at least $1-\varepsilon$ the inequalities*

$$|ty_i \pmod M| \leqq d_{k,\varepsilon} M^{(k-1)/k} \tag{4.1}$$

cannot be solved with $0 < t < M$, and so

$$\lambda_1(L_y^*) \geqq d_{k,\varepsilon} M^{-1/k}.$$

Proof. For each fixed t and a fixed index i the probability that

$$|ty_i \pmod M| \leqq d_{k,\varepsilon} M^{(k-1)/k}$$

is at most

$$\frac{(t, M)}{M} + 2d_{k,\varepsilon} M^{-1/k} \leqq 3d_{k,\varepsilon} M^{-1/k},$$

since

$$\frac{(t, M)}{M} \leqq \frac{M}{p_1} + \frac{M}{p_2} \leqq d_{k,\varepsilon} M^{(k-1)/k}.$$

Since the draws for different i are independent, the probability that (4.1) holds for fixed t is $\leqq (3d_{k,\varepsilon})^k M^{-1}$. Summing over $0 < t < M$ the total probability that (4.1) holds is at most $\leqq (3d_{k,\varepsilon})^k$, and we may choose $d_{k,\varepsilon} = \frac{1}{3}\varepsilon^{1/k}$. □

Now we complete the proof of Theorem 4.1. By Corollary 2.2 we can recover $(x_1, \cdots, x_k)$ if we know $s_0 = \log \lambda_k + (k/2) + \frac{1}{2}\log k + 1$ significant bits of each x_i. Using Lemma 4.2 and the bound $\lambda_1^* \lambda_k \leqq k$ we obtain

$$\lambda_k(L_y) \leqq k^2 d_{k,\varepsilon}^{-1} M^{1/k}.$$

Since $n = \log M$ this yields

$$s_0 \leqq \frac{n}{k} + \frac{k}{2} + \frac{3}{2}\log k - \log d_{k,\varepsilon} + 1.$$

We are given $s_1 = n/k + c_{k,\varepsilon}$ significant bits, so on choosing

$$c_{k,\varepsilon} = \frac{k}{2} + \frac{3}{2}\log k - \log d_{k,\varepsilon} + 1$$

we can find the x_i. As pointed out earlier, this enables us to factor M by calculating g.c.d. $(x_i - y_i, M)$ so the theorem follows. □

Theorem 4.1 shows that Blum's original protocol can be broken by somebody who only deviates from the protocol by stopping early and using this algorithm—there is no need to control the choice of random bits or to lie to the other party.

The alternative protocol in which the columns of the matrices are exchanged in reverse order (from least significant bits to most significant bits) is just as insecure, again using the algorithm of Corollary 2.2, noting that M is odd.

Discussion. Blum proved that his protocol is secure assuming the truth of certain (unproved) assumptions. Due to the care with which Blum listed his assumptions, it is easy to trace the source of the cryptographic weakness we exploit to the following one [1, p. 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_i^k."

Our attack shows that this assumption was too strong. Blum considered the possibility that his original protocol might be insecure, and in his paper he described a modified protocol in which the participants use several moduli each. A second 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. Neither of these variants seems to be vulnerable to the cryptanalytic attack proposed in this paper.

The existence of this cryptanalytic attack demonstrates once more the extremely delicate nature of proofs of security in cryptology. It also shows the importance when proposing cryptographic protocols to clearly distinguish sources of insecurity. Blum's paper certainly does this.

Acknowledgments. We thank M. Blum, J. Boyer, O. Goldreich and S. Micali for bringing this problem to our attention. We thank Shafi Goldwasser and Rick Statman for useful discussion and comments.

REFERENCES

[1] M. BLUM, *How to exchange (secret) keys*, ACM Trans. Comput. Systems, 1 (1983), pp. 175–193.
[2] J. BOYAR, *Inferring sequences produced by pseudo-random number generators*, Proc. 23rd IEEE Conference on Foundations of Computer Science, 1982, pp. 153–159.
[3] J. W. S. CASSELS, *Geometry of Numbers*, Springer-Verlag, New York, 1971.
[4] ———, *Rational Quadratic Forms*, Academic Press, London, 1978.
[5] A. M. FRIEZE, R. KANNAN AND J. C. LAGARIAS, *Linear congruential generators do not produce random sequences*, Proc. 25th IEEE Symposium on Theory of Computing, 1984, pp. 480–484.
[6] J. HASTAD AND A. SHAMIR, *The cryptographic security of truncated linearly related variables*, Proc. 17th Annual ACM Symposium on the Theory of Computing, 1985, pp. 356–362.
[7] R. KANNAN, *Improved algorithms for integer programming and related lattice problems*, Proc. 15th Annual ACM Symposium on the Theory of Computing, 1983, pp. 193–206.
[8] R. KANNAN AND A. BACHEM, *Polynomial algorithms for computing the Smith and Hermite normal forms of an integer matrix*, this Journal, 8 (1979), pp. 499–507.
[9] A. YA. KHINCHIN, *Continued Fractions*, University of Chicago Press, Chicago, 1964.
[10] D. E. KNUTH, *The Art of Computer Programming*, Vol. 2, Addison-Wesley, Reading, MA, 1980.
[11] ——— *Deciphering a linear congruential encryption*, IEEE Trans. Inform. Theory, IT-31 (1985), pp. 49–52.
[12] J. C. LAGARIAS, H. W. LENSTRA, JR., AND C. P. SCHNORR, *Korkine–Zolotarev bases and successive minima of a lattice and its reciprocal lattice*, preprint.
[13] A. K. LENSTRA, H. W. LENSTRA, JR., AND L. LOVÁSZ, *Factoring polynomials with integer coefficients*, Math. Ann., 261 (1982), pp. 513–534.
[14] H. W. LENSTRA, JR., *Integer programming in a fixed number of variables*, Math. Oper. Res., 8 (1983), pp. 538–548.
[15] J. REEDS, "*Cracking*" *a random number generator*, Cryptologia, 1 (1977), pp. 20–26.
[16] ——— *Cracking a multiplicative congruential encryption algorithm*, in Information Linkage Between Applied Mathematics and Industry, P. C. Wong, ed., Academic Press, New York, 1979, pp. 467–472.
[17] C. P. SCHNORR, private communication.