Positive Applications of Lattices to Cryptography*

Cynthia Dwork

IBM Almaden Research Center.

Abstract. We describe constructions of several cryptographic primitives, including hash functions, public key cryptosystems, pseudo-random bit generators, and digital signatures, whose security depends on the assumed worst-case or average-case hardness of problems involving lattices.

1 Introduction

Initiated by Ajtai's paper "Generating Hard Instances of Lattice Problems," a burgeoning effort to build cryptographic primitives based on the assumed hardness of worst-case or random instances of problems involving lattices has proved extremely fruitful. Prior to Ajtai's work, lattices, and in particular, the lattice basis reduction algorithm of Lenstra, Lenstra, and Lovász, were used in cryptography principally to prove cryptographic *in*security [1, 9, 10, 20, 22, 25]. We describe more positive applications of lattices: constructions for public key cryptosystems, cryptographically strong hash functions, and pseudo-random bit generators whose security depends only on the worst-case hardness of the underlying lattice problem; a digital signature scheme whose security depends on the average hardness of the underlying problem.

2 Definitions

Many of the definitions included here are *extremely* informal. References for precise definitions are included in every case.

2.1 Cryptography

A *one-way* function is easy to compute and hard to invert. A *trapdoor* function is a one-way function for which there exists some special "trapdoor" information, so that given the trapdoor information the function is easy to invert, but without the trapdoor information the function is hard to invert (see [12]).
A *public key cryptosystem* is a method of encrypting messages using publicly known information called the *public key*, in such a way that only the party knowing the corresponding *private key* can decrypt the ciphertext. Thus, encryption has a trapdoor nature: without the trapdoor information (the private key) decryption is hard, but decryption is easy given the private key (see [16] and [11]).
A *digital signature scheme* is a method of generating a (public key, private key) pair, together with a pair of procedures SIGN, and VERIFY. SIGN requires as input the message to be signed and the private key of the signer, while VERIFY, requires as input the message, its purported signature, and the public key of the claimed signer. Let (K, s) be a (public key, private key) pair. Let (m, α) be a claimed (message, signature) pair. Given (m, α, K) the VERIFY procedure, without knowing the secret s, verifies that $\alpha = \text{SIGN}(m, s)$ (see [17]).
A *one-way hash function* is a one-way function h mapping long strings to short strings, say, $h : \{0,1\}^n \rightarrow \{0,1\}^\ell$ for $n > \ell$. *One-way* hash functions have many uses in cryptography. In particular they are used to "shrink" long messages before signing (see [24]). Thus, what is actually signed is $h(m)$ rather than m ($h(m)$ is sometimes called a *message digest*). In this case the VERIFY procedure checks that $\alpha = \text{SIGN}(h(m), s)$. For this application it is essential that, given $h(m)$, it is hard to find a different message $m' \neq m$, for which $h(m') = h(m)$. A little more formally, a family of *universal one-way hash functions* is a collection $\mathcal{F}$ of functions $f : \{0,1\}^m \rightarrow \{0,1\}^{l(m)}$ with the property that for any element $x \in \{0,1\}^m$,

* This paper appeared in the proceedings of the 22nd International Symposium on Mathematical Foundations of Computer Science, *LNCS 1295*, Springer, 1997.

if f is chosen at random from the collection $\mathcal{F}$, then it is hard to find an element $y \neq x$ such that $f(y) = f(x)$. Each choice of $l(m)$ yields a class of hash functions. A slightly stronger notion is *collision-intractability*: for a randomly selected function $f \in \mathcal{F}$, it is hard to find x, y such that $x \neq y$ and $f(x) = f(y)$.

A *pseudorandom bit generator* is a (deterministic) function that takes as input a string $s \in \{0,1\}^n$ and produces as output a string $p \in \{0,1\}^m$ where $m > n$. Moreover, the strings produced in this way when the inputs s are random should be polynomial-time indistinguishable from truly random strings of length m. Thus these functions appear to manufacture some additional bits of randomness (see [6, 26]; extensive treatment appears in [23]).

The *subset sum problem* of dimensions m and l is: given m numbers $\mathbf{a} = (a_1, \ldots, a_m)$, each of length l, and a number T, find a subset $S \subset \{1, \ldots, m\}$ such that $\sum_{i \in S} a_i = T \bmod 2^l$. The subset sum problem can be viewed as that of inverting the function $f(\mathbf{a}, S) = \mathbf{a}, \sum_{i \in S} a_i \bmod 2^{l(n)}$.

2.2 Lattices

The fundamental concepts concerning lattices can be found in [8, 18, 19].

If $a_1, ..., a_n$ are linearly independent vectors in $\mathbb{R}^n$, then we say that the set $\{\sum_{i=1}^n k_i a_i | k_1, ..., k_n \in \mathbb{Z}\}$ is a lattice in $\mathbb{R}^n$. We will denote this lattice by $L(a_1, ..., a_n)$. The set $a_1, ..., a_n$ is called a basis of the lattice; its length is $\max_{1 \le i \le n} \|a_i\|$. The determinant of a lattice L will be the absolute value of the determinant of the matrix whose columns are the vectors $a_1, ..., a_n$. We let $\mathrm{bl}(L)$ denote the length of the shortest basis for L.

The *dual* lattice of L, denoted L^*, is defined as

$$L^* = \{x \in \mathbb{R}^n \mid x^T y \in \mathbb{Z} \text{ for all } y \in L\}.$$

If $(b_1, \ldots, b_n)$ is a basis of L then $(c_1, \ldots, c_n)$ is a basis for L^*, where

$$c_i^T b_j = \begin{cases} 1 \text{ if } i = j \\ 0 \text{ if } i \neq j \end{cases}$$

Thus, if we represent the lattice $L = L(b_1, \ldots, b_n)$ by a matrix B with columns $b_1, \ldots, b_n$, then the dual of L is the lattice spanned by the rows of B^{-1}. Each basis vector b_i in $L = L(b_1, \ldots, b_n)$ induces a collection of mutually parallel $(n-1)$-dimensional hyperplanes, where, for $k \in \mathbb{Z}$, the kth hyperplane in the collection is the set of all points whose inner product with b_i is equal to k. The distance between adjacent hyperplanes in the collection is $\|b_i\|^{-1}$. Thus, if $\|b_i\| < \|b_j\|$, then adjacent hyperplanes in the ith collection are farther apart than adjacent hyperplanes in the jth collection. As the formula for computing the basis for the dual makes clear, the dual lattice is the set of points that are intersections of n hyperplanes, one from each of the n collections.

Assume n is a positive integer, $M > 0$, $d > 0$ are real numbers, and $L \subseteq \mathbb{Z}^n$ is a lattice which has an $n-1$ dimensional sublattice L' with the following properties:

1. L' has a basis of length at most M;
2. if H is the $n-1$ dimensional subspace of $\mathbb{R}^n$ containing L' and $H' \neq H$ is a coset of H intersecting L, then the distance of H and H' is at least d.

We say that L is a (d, M)-lattice. If $d > M$, then L' is unique. In this case L' will be denoted by $L^{(d,M)}$.

If $a_1, \ldots, a_n \in \mathbb{R}^n$ are linearly independent vectors, then $\mathcal{P}^-(a, \ldots, a_n)$ denotes the half-closed parallelepiped $\{\sum_{i=1}^n \gamma_i a_i | 0 \le \gamma_i < 1, i = 1, \ldots, n\}$. By "$x \bmod \mathcal{P}$" we mean the unique vector $x' \in \mathcal{P}^-(a_1, \ldots, a_n)$ so that $x - x'$ is an integer linear combination of the vectors $a_1, \ldots, a_n$.

The *orthogonality defect* of an $n \times n$ matrix B is the quantity $\frac{1}{det(B)} \prod_{i=1}^n \|b_i\|$. The *dual orthogonality defect* of B is the quantity $\frac{1}{det(B^{-1})} \prod_{i=1}^n \|\hat{b_i}\|$, where for $1 \le i \le n$, $\hat{b_i}$ is the ith row of B^{-1}.

3 Generating Hard Instances of Lattice Problems

Cryptographic constructions necessarily require random choices: if, for example, the choice of a key were deterministic, then the key could not be secret. Thus, the security of the construction relies on the intractability of a *random* instance of the problem on which the construction is based. It has therefore been a longstanding goal in cryptography to find a "hard" problem for which one can establish an explicit connection between the hardness of random instances and the hardness of the hardest, or worst-case, instances.

Such a connection is the contribution of the celebrated paper of Ajtai, "Generating Hard Instances of Lattice Problems" [2]. Specifically, the paper presents a random problem whose solution would imply the solution of three famous worst-case problems:

1. Find the length of a shortest nonzero vector in an n-dimensional lattice approximately, up to a polynomial factor.
2. Find the shortest nonzero vector in an n-dimensional lattice L where the shortest vector v is unique in the sense than any other vector whose length is at most $n^c\|v\|$ is parallel to v, where c is a sufficiently large absolute constant.
3. Find a basis $b_1, \ldots, b_n$ in the n-dimensional lattice L whose length, defined as $\max_{i=1}^n \|b_i\|$, is the smallest possible up to a polynomial factor.

Ajtai's Random Lattice Problem. For $n, m, q \in \mathcal{N}$ such that $n \log q < m \leq \frac{q}{2n^4}$ and $q = O(n^c)$ for a fixed $c > 0$, given a matrix $M \in \mathbb{Z}_q^{n \times m}$ (that is, an $n \times m$ matrix of integers in [0,q-1] of a certain form described below), find a vector $x \neq 0 \in \mathbb{Z}_q^m$ so that $Mx \equiv 0 \bmod q$ and $\|x\| < n$. The lattices are defined modulo q, in the sense that if two vectors are congruent modulo q then either both are in the lattice or neither is in the lattice. Thus the matrix M and the integer q define the lattice: $x \in \Lambda(M, q)$ iff $Mx \equiv 0 (\bmod q)$.

The matrix M is obtained as follows. Randomize vectors $v_1, \ldots, v_{m-1}$ independently and with uniform distribution on the set of all vectors $\langle x_1, \ldots, x_n \rangle \in \mathbb{Z}_q^n$. Independently randomize a 0, 1 sequence $\delta_1, \ldots, \delta_{m-1}$, where the numbers δ_i are chosen independently and uniformly. Then define $v_m = -\sum_{i=1}^{m-1} \delta_i v_i \bmod q$ with the additional constraint that each component of v_m is an integer in $[0, q-1]$. The matrix M has columns $v_1, \ldots, v_m$. The class of lattices $\Lambda(M, q)$ defined by matrices of this type will be called λ. The random problem is to find a vector in $\Lambda(M, q)$ of length less than n. Note that $(\delta_1, \ldots, \delta_{m-1}, 1) \in \Lambda(M, q)$ and its length is $O(\sqrt{m})$, so this vector is a solution when $m < n^2$.

Let L be an n-dimensional lattice, let $a_1, \ldots, a_n$ be a set of linearly independent vectors in L and let $M = \max_{i=1}^n \|a_i\|$. The heart of Ajtai's work is a procedure which, if $M > n^c \mathrm{bl}(L)$ for a fixed consant c, uses an oracle for the random lattice problem just defined to obtain another set of linearly independent elements in L whose maximum length is at most $\frac{1}{2} \max_{1 \leq i \leq n} \|a_i\|$.

In rough outline the procedure works as follows. Starting from $a_1, \ldots, a_n$, construct a set of linearly independent lattice vectors $f_1, \ldots, f_n$ such that $\max_{i=1}^n \|f_i\| \leq n^3 M$ and $W = \mathcal{P}(f_1, \ldots, f_n)$ is close to a cube, in the sense that each vertex of W will be at most distance nM from a fixed cube. If the space is covered with the cells of a lattice determined by a short basis, then most of the cells intersecting W lie completely in the interior of W. This implies that every parallelepiped of the form $u + W$, $u \in \mathbb{R}^n$, has roughly the same number of lattice points. Moreover, this also holds for parallelepipeds of the form $u + \frac{1}{q}W$ for $q = [n^{c_2}]$, where c_2 is sufficiently small with respect to c. Thus, if we pick a lattice point v at random from a set D of parallelepipeds of the form $u + \frac{1}{q}W$ with non-overlapping interiors, then the distribution induced on D – that is, the choice of which element in D contains v – is very close to the uniform distribution.

The set D of parallelepipeds $u + \frac{1}{q}W$ that is of interest to us is that obtained by cutting W into q^n small parallelepipeds by dividing each of the vectors f_i into q pieces of equal length. Thus each of the small parallelepipeds is of the form $(\sum_{i=1}^n t_i \frac{f_i}{q}) + \frac{1}{q}W$, where $0 \leq t_i < q$, $i = 1, \ldots, n$ is a sequence of integers; that is, $\langle t_1, \ldots t_n \rangle \in \mathbb{Z}_q^n$. Let us call the vector $o = \sum_{i=1}^n t_i \frac{f_i}{q}$ the *origin* of the parallelepiped. We will name an element of D by the vector $t(o) = \langle t_1, \ldots, t_n \rangle$ of coefficients of the $\frac{f_i}{q}$ defining its origin. If we choose a random set of lattice points $\xi_1 \ldots, \xi_m$ in W and look at, for each ξ_j, the name $t(o_j)$ of the parallelepiped containing ξ_j, then we get a sequence $t(o_1), \ldots, t(o_m)$ of elements chosen almost uniformly from D. Express each ξ_j as the sum of the origin o_j and an offset $\delta_j \in \frac{1}{q}W$. Note that the offset is relatively short: since δ_j is contained in $\frac{1}{q}W$, $\|\delta_j\|$ is bounded by n times the length of the longest side of W. That is, $\max_{1 \leq j \leq m} \|\delta_j\| \leq n(\frac{1}{q} n^3 M)$.

By definition of D and the fact that the distribution induced on D by the choice of ξ is almost uniform, each $t(o_j)$ is distributed almost uniformly in $\mathbb{Z}_q^n$. Let $m = [c_1 n \log n]$. Consider the sequence $t(o_1), \ldots, t(o_m)$ as a value of the random variable λ (it is shown in [2] that the distribution of

$t(o_1), \ldots, t(o_m)$ is extremely close to that of λ). If there exists an algorithm $\mathcal{A}$ that can solve Ajtai's random lattice problem, then using $\mathcal{A}$ we can find a short (length at most n) vector $h = \langle h_1, \ldots, h_m \rangle \in \mathbb{Z}^m$ satisfying $\sum_{j=1}^m h_j t(o_j) \equiv 0 \bmod q$.
Writing the lattice vector $\sum_{j=1}^m h_j \xi_j$ as the weighted sum of origins and offsets, we get

$$w = \sum_{j=1}^m h_j \xi_j = \sum_{j=1}^m h_j o_j + \sum_{j=1}^m h_j \delta_j \ .$$

Critically, since $\sum_j h_j t(o_j) \equiv 0 \bmod q$, we have that $\sum_j h_j o_j$ is an integer linear combination of the vectors $(f_1, \ldots, f_n)$. Since the f_i are lattice vectors, so is $\sum_j h_j o_j$. Since w is also in L the difference $w - \sum_j h_j o_j = \sum_j h_j \delta_j \in L$. Finally, since $|\sum_{j=1}^m h_j^2| \leq n^2$ and, as noted above, each of the offsets is also relatively short, the lattice vector $\sum_{1 \leq j \leq n} h_j \delta_j$ is relatively short: $\| \sum_{1 \leq j \leq n} h_j \delta_j \| \leq n^2 (n^4 M \frac{1}{q})$, which is less than $\frac{M}{2}$ if q is sufficiently large (say, $q \geq n^7$).
Recently, Ajtai's results have been tightened by Cai and Nerurkar [7]. Through a number of technical steps, Cai and Nerurkar are able to shrink the constant c in Ajtai's reduction, slightly better than halving it.
Based solely on the results in [2], it is possible to design a number of *interactive* cryptographic procedures, including schemes for identification, bit commitment, and coin flipping [3].

4 Hashing

The reduction described in the previous section has implications for the security of the following family of hash functions, studied by Impagliazzo and Naor [21]:
Let $l(m) = (1 - c)m$ for $c > 0$. For $a_1, \ldots, a_m \in \{0,1\}^{l(m)}$ the function $f_{\mathbf{a}} = f_{a_1, \ldots, a_m} : \{0,1\}^m \rightarrow \{0,1\}^{l(m)}$ is defined as follows. Let the m-bit number x be written $x = x_1 x_2 \ldots x_m$ where each $x_i \in \{0,1\}$. Then $f_{\mathbf{a}}(x) = \sum_{i=1}^m x_i a_i \bmod 2^{l(m)}$.
The bits of x act as selectors to determine which of the a_i are summed. We can represent the function as a $1 \times m$ matrix M with columns $a_1, \ldots, a_m$. Given $x \in \{0,1\}^m$, the value of the function is $Mx \bmod 2^{l(m)}$.
As we next explain, Ajtai's proof shows that the ability to solve a random instance of the subset sum problem implies the ability to solve the worst-case lattice problems listed in Section 3 (additional details appear in [2]). So if we assume that these worst-case problems are hard for dimension n, then these randomized subset sum problems will be hard as well. To illustrate this connection, let $q = [n^{c_2}]$ and $m = [c_1 n \log n]$ as in the discussion of Ajtai's reduction in Section 3. Let $N = q_1 q_2 \ldots q_n$ where each q_i is a distinct prime in $[q, 2q]$. Let $a_1, \ldots, a_m, b$ be random integers modulo N. Consider the subset sum problem of finding $x \in \{0,1\}^m$ such that $\sum_{i=1}^m x_i a_i \equiv b \bmod N$.

Remarks.
(1) The numbers a_i are of length $l(m) \approx n \log q = (1 - c)m$ for some $c > 0$ if $c_1 > c_2$. So subset sum problems of this type are essentially those in the Impagliazzo-Naor family of hash functions.
(2) If $x \in \{0,1\}^m$ then $\|x\| \leq \sqrt{m} < n$.

We may express each a_i as a vector of remainders modulo the primes $q_1, \ldots, q_n$: $a_i' = (a_1^i, \ldots, a_n^i)$, where $a_j^i \in \mathbb{Z}_{q_j}$, for $1 \leq i \leq m$ and $1 \leq j \leq n$. Note that if a_i is chosen uniformly from $\mathbb{Z}_N$ then a_i' is implicitly chosen uniformly from $\mathbb{Z}_{q_1} \times \ldots \times \mathbb{Z}_{q_n}$. Similarly, let b' be the Chinese remainder decomposition of b. Let M be the $n \times m$ matrix with columns $a_1', \ldots, a_n'$. If we can find $x \in \{0,1\}^m$ satisfying $\sum_{i=1}^m x_i a_i \equiv b \bmod N$, then $Mx \equiv b'$ (where the jth component of the product is reduced modulo q_j, $1 \leq j \leq n$).
The hardness of this problem follows from Ajtai's proof. The key modification is as follows. Recall that $W = \mathcal{P}(f_1, \ldots, f_n)$. Rather than cutting each vector f_i, $1 \leq i \leq n$, into q equal pieces (for a fixed q), instead for each $1 \leq i \leq n$, cut f_i into q_i pieces. Thus, instead of having q^n little parallelepipeds we will have $N = q_1 \ldots q_n$ of them. Any solution x plays the role of the solution $h = \langle h_1, \ldots, h_m \rangle$ in the original proof. See [2] for more details and extensions of these results.
Impagliazzo and Naor proved that if the subset sum function for length $(1 - c)m$, $c > 0$, is one-way in the sense that no polynomial time algorithm can invert the function on a random input, then it is also

a family of universal one-way hash functions [21]. Since this class of subset sum problem is hard on average (assuming the worst-case lattice problems are difficult for dimension n), the Impagliazzo and Naor construction yields a family of universal one-way hash functions.

In a related note, Goldreich, Goldwasser, and Halevi [13] observed that these hash functions are actually *collision-intractable*. Specifically, they show that if M is a random matrix in $Z_q^{n\times m}$, then finding collisions of the function $h(x) = Mx \bmod q$ is hard provided a slight modification of Ajtai's random lattice problem is hard. The modification is to only require that the vector x have coefficients in $\{-1, 0, 1\}$ (rather than to require $x \in \mathbb{Z}_q^m$ and $\|x\| < n$), and the proof of collision-intractability relies on the fact that Ajtai's results hold even if the random lattice problem is relaxed so that $\|x\|$ is bounded by a polynomial in n. (The more relaxed version incurs a cost in the quality of the approximation obtained in Ajtai's reduction.) Collision-intractability follows from the fact that if it were easy to find $x, y \in \{0,1\}^m$ such that $Mx \equiv My \bmod q$ then $M(x-y) \equiv 0 \bmod q$. Since $x - y \in \{-1, 0, 1\}^m$, finding such a pair x, y is difficult.

5 Public Key Cryptography

Ajtai and Dwork constructed a public key cryptosystem generator with the property that if a random instance of the cryptosystem can be broken, that is, if for a random instance the probability that an encryption of a zero can be distinguished from an encryption of a one (without the private key) in polynomial time is at least $\frac{1}{2} + n^{-c_1}$ for some absoloute constant $c_1 > 0$, then the worst-case unique shortest vector problem has a probabilistic polynomial time solution. Intuitively, this worst-case/average-case equivalence means that there are essentially no "bad" instances of the cryptosystem.

In this discussion we will work with real numbers, ignoring issues of finite precision. The private key is a vector $u \in \mathbb{R}^n$ chosen uniformly at random from the n-dimensional unit ball. u induces a collection of $(n-1)$-dimensional hyperplanes, where for $i \in \mathbb{Z}$ the ith hyperplane is the set of vectors v whose inner product satisfy $u \cdot v = i$. Very roughly speaking, the public key is a method of generating a point guaranteed to be near one of the hyperplanes in the collection. The public key is chosen so as not to reveal the collection of hyperplanes – indeed, Ajtai and Dwork prove that any ability, given only the public key, to discover the collection implies the ability to solve the worst-case unique shortest vector problem. Encryption is bit-by-bit: zero is encrypted by using the public key to find a random vector $v \in \mathbb{R}^n$ near one of the hyperplanes – the ciphertext is v; one is encrypted by choosing a random vector u uniformly from $\mathbb{R}^n$ – the ciphertext is simply u. Decryption of a ciphertext x is simple using the private key u: if $u \cdot x$ is close to an integer then x is by definition near one of the hidden hyperplanes, and so x is interpreted as zero; otherwise x is interpreted as one.

If a lattice Λ has an n^c-unique shortest vector v, then $L = \Lambda^*$ is a $(\|v\|^{-1}, n^{-c'}\|v\|^{-1})$ lattice, where c' is roughly $c - 2$ (a proof appears in [2]). Moverover, v is orthogonal to the $(n-1)$-dimensional space containing $L' = L^{(\|v\|^{-1}, n^{-c'}\|v\|^{-1})}$, and if H is the $(n-1)$-dimensional subspace of $\mathbb{R}^n$ containing L', then the hyperplanes induced by v are the cosets of H intersecting L (recall the discussion of the dual in Section 2).

Define $\mathrm{pert}(R)$ to be a random variable that, roughly speaking, is normally distributed about the origin in a ball of radius R. Let $\mathcal{K}$ be a very large cube, and let $R = n^c$. It is first shown that if the n^{c_1}-unique shortest vector problem is hard, for c_1 sufficiently larger than c, then the distribution obtained by choosing a random lattice point in $\mathcal{K}$ and perturbing it by adding a value of $\mathrm{pert}(R)$ (for sufficiently large R) is polynomially indistinguishable from the distribution obtained by choosing a vector uniformly at random from $\mathcal{K}$.

To see this, suppose we are given a *random* lattice Λ with an n^{c_1}-unique shortest vector v, and let $L = \Lambda^*$. Let $d = \|v\|^{-1}$ and $M = n^{-c_1'}\|v\|^{-1}$, where c_1' is roughly $c_1 - 2$. Then L is a (d, M) lattice. Let $L' = L^{(d,M)}$ have basis $b_1, \ldots, b_{n-1}$. Let $H = H_0$ be the $(n-1)$-dimensional hyperplane containing L'. If R is sufficiently large with respect to $b_1, \ldots, b_{n-1}$, then the random variable obtained by sampling $\mathrm{pert}(R)$, projecting the result onto the $(n-1)$-dimensional hyperplane containing L', and taking the projection modulo $\mathcal{P}^-(b_1, \ldots, b_{n-1})$ is extremely close to the value obtained by choosing a point uniformly in $\mathcal{P}^-(b_1, \ldots, b_{n-1})$.

Intuitively, this means that any algorithm distinguishing between "lattice point + $\mathrm{pert}(R)$" and the uniform distribution is really distinguishing between points close to the cosets of H intersecting L and random points. ¿From this it is possible (with some effort – see [4]) to find H. Finally, given H we can recover v, the unique shortest vector in $\Lambda = L^*$ as follows. As noted above, v is perpendicular to H. By definition $\|v\| = d^{-1}$; given a basis for L (computable from the given basis for Λ), we can sample points from L and compute for each its distance from H. By taking the gcd of many random such distances we can find d.

The next step is to dispense with the lattice L. Let u be chosen uniformly at random from the n-dimensional unit ball and let $\mathcal{H}_u$ be the collection of hyperplanes induced by u. The distribution obtained by choosing a random point in $\mathcal{H}_u \cap \mathcal{K}$ and then sufficiently perturbing the chosen point, is indistinguishable from the uniform distribution in $\mathcal{K}$ – otherwise there would be a way of distinguishing points close to the hyperplanes from random points. The scheme is therefore as follows.

Private Key: vector u chosen at random from the n-dimensional unit ball

Public Key: $v_1, \ldots, v_m$: a collection of perturbations of points chosen uniformly from $\mathcal{H}_u \cap \mathcal{K}$, and a parallelepiped $\mathcal{P}$

Encryption: To encrypt zero, choose $\delta_1, \ldots, \delta_m$, each $\delta_i \in_R \{0, 1\}$. The ciphertext is $\sum_{i=1}^{m} \delta_i \bmod \mathcal{P}$. To encrypt one, choose a random point in $\mathcal{P}^-$.

Decryption: given ciphertext x, compute $x \cdot u$. If the result is sufficiently close (as a function of R) to an integer, then decrypt x as zero; else decrypt x as one.

There is some chance of a decryption error. This can be avoided by including in the public key a point B obtained by averaging two encryptions of zero lying on hyperplanes of different parity. (A related solution appears in [15].) The procedure for encrypting one becomes: follow the procedure for encrypting zero but add B before modding out by $\mathcal{P}$.

Very roughly, worst-case/average-case equivalence is shown as follows. Suppose we have an algorithm $\mathcal{A}$ that can break random instances of the cryptosystem with non-negligible probability over the choice of u. Given any instance L of the unique shortest vector problem, we convert it to an instance of the cryptosystem by choosing a number of random linear transformations $U = \theta\nu$ where $\theta \in \mathbb{R}$ and ν is an orthogonal linear transformation. Intuitively, ν rotates the lattice L leaving the lengths of the basis vectors unchanged, while θ scales the rotated basis. If v is the unique shortest vector and we choose enough transformations, then for one of them $\|Uv\| < 1$ and $\mathcal{A}$ can crack the instance of the cryptosystem defined by u. Note that v is the n^c-unique shortest vector of L if and only if Uv is the n^c-unique shortest vector of UL. It follows that J, the dual lattice of UL, is a $(1, n^{-c'})$ lattice, where $c' \sim c - 2$. Moreover, the distribution obtained by perturbing points of J is exponentially close to the distribution obtained by perturbing points in the hyperplanes induced by Uv. But Uv describes (the private key of) a *random* instance of the Ajtai-Dwork cryptosystem: it is random because U is random. Moreover, the ability to distinguish zeros – points close to the hyperplanes induced by Uv – from ones – random points– would imply the ability to distinguish perturbations of lattice points in J from random points. As argued above, this ability would yield Uv, the unique shortest vector in J^*, and hence, by the invertibility of U, v.

6 Pseudorandom Bit Generators

The Ajtai-Dwork construction suggests a pseudorandom bit generator with a very natural geometrical interpretation. Note that, given the secret information u, it requires fewer bits to describe a point that is close to one of the hyperplanes induced by u than to describe a point chosen at random from $\mathbb{R}^n$. To see this, consider a basis $b_1, \ldots, b_n$ for $\mathbb{R}^n$ in which the first $n-1$ vectors lie in H_0, the $(n-1)$-dimensional space orthogonal to u, and b_n is parallel to u. Using this basis it is easy to see that to describe a random point requires more bits than to describe a point close to one of the hyperplanes because, intuitively, there are more choices for the random point (the distance of a random point to the nearest hyperplane can be any value in $[0, \|u\|/2]$, while the distance of a point close to the hyperplane is in $[0, n^{-c}]$ for a fixed constant $c > 0$).

7 Digital Signatures

Goldreich, Goldwasser, and Halevi have suggested a digital signature scheme based on a trapdoor function related to the problem of finding the lattice vector closest to a given vector v [14]. Their approach, which also yields a public-key cryptosystem, depends on the hardness of random instances of the underlying problem (rather than worst-case instances). Naor and Yung have shown how to obtain a digital signature scheme from any one-way function [24]. Other than schemes obtained by applying this general construction to the one-way functions of [2, 4], we know of no proposed digital signature scheme with worst-case/average-case equivalence.

The trapdoor function proposed by Goldreich, Goldwasser, and Halevi relies on the difficulty, given a basis B for a lattice L, of finding a basis for L with small dual orthogonality defect. Call such a basis *reduced.*

The trapdoor information is a reduced basis R for an n-dimensional lattice (defined implicitly by R). Given R, it is possible to generate a second basis B for $L = L(R)$ so that B has high dual orthogonality defect. The trapdoor function is specified by B and a real parameter $\sigma \in \mathbb{R}$. Given vectors $v, e \in \mathbb{R}^n$, the function $f_{(B,\sigma)}(v, e) = Bv + e$. Note that the value σ does not appear in the definition of the function. Rather, σ governs the selection of e: each entry in e is chosen at random according to a distribution with zero mean and variance σ^2. For example, each entry in e can be chosen uniformly from $\{\sigma, -\sigma\}$.

Assume e is chosen as described and each component of v is chosen uniformly from, say, $\{-n^2, -n^2 + 1, \ldots, n^2 - 1, n^2\}$. Let $c = f_{(B,\sigma)}(v, e) = Bv + e$. If σ is chosen carefully, the function can be inverted using R by applying Babai's rounding technique [5]: represent c as a linear combination of the columns of R and then round the coefficients in the linear combination to the nearest integers to obtain a lattice point (integer linear combination of the columns of R). Once v is recovered we find $e = c - Bv$.

In the Goldreich, Goldwasser, and Halevi digital signature scheme, the private key is a reduced basis R and the public key is a non-reduced basis B. To sign a message m encoded as a vector $v \in \mathbb{R}^n$, the signer computes, using the reduced basis, a lattice vector w close to v. The public verification key is a threshold τ and the non-reduced basis B; the signature is verified by checking that $\|v - w\| \leq \tau$. As the authors point out, if $u, u' \in \mathbb{R}^n$ are sufficiently close, then a signature on u is likely also to be a signature on u'; it is therefore important to use a "good hash function" to hash a message before interpreting it as a vector in $\mathbb{R}^n$ [14].

References

1. L. Adleman, On Breaking Generalized Knapsack Public Key Cryptosystems, Proceedings 15th Annual ACM Symposium on Theory of Computing, 1983, pp. 402–412
2. M. Ajtai, Generating Hard Instances of Lattice Problems, Proceedings 28th Annual ACM Symposium on Theory of Computing, 1996, pp. 99–108 *Electronic Colloquium on Computational Complexity TR96-007*, http://www.eccc.uni-trier.de/eccc-local/Lists/TR-1996.html
3. M. Ajtai, *discussion with the author*, 1996
4. M. Ajtai, C. Dwork, A Public-Key Cryptosystem with Average-Case/Worst-Case Equivalence, Proceedings 29th Annual ACM Symposium on Theory of Computing, 1997; see also *Electronic Colloquium on Computational Complexity TR96-065*, http://www.eccc.uni-trier.de/eccc-local/Lists/TR-1996.html
5. L. Babai, On Lovász' Lattice Reduction and the Nearest Lattice Point Problem, *Combinatorica 6*(1), 1986, pp. 1–13
6. M. Blum and S. Micali, How to Generate Cryptographically Strong Sequences of Pseudo-Random Bits, *SIAM J. Computing 13*, 1984, pp. 850–864
7. J.-Y. Cai and A. P. Nerurkar, An Improved Worst-Case to Average-Case Connection for Lattice Problems, *private communication*, 1997
8. J.W.S. Cassels, *An Introduction to the Geometry of Numbers*, Springer, 1959
9. D. Coppersmith, Finding a Small Root of a Univariate Modular Equation, *Proc. EUROCRYPT'96*
10. D. Coppersmith, M. Franklin, J. Patarin, and M. Reiter, Low Exponent RSA with Related Messages, *Proc. EUROCRYPT'96*

11. D. Dolev, C. Dwork, and M. Naor, Non-Malleable Cryptography, Proceedings 23th Annual ACM Symposium on Theory of Computing, 1991, pp. 542–550
12. O. Goldreich, *Foundations of Cryptography (Fragments of a Book)*, http://www.wisdom.weizmann.ac.il/people/homepages/oded/frag.html
13. O. Goldreich, S. Goldwasser, and S. Halevi, Collision-Free Hashing from Lattice Problems, *Electronic Colloquium on Computational Complexity TR96-042*, http://www.eccc.uni-trier.de/eccc-local/Lists/TR-1996.html
14. O. Goldreich, S. Goldwasser, and S. Halevi, Public-Key Cryptosystems from Lattice Reduction Problems, *Electronic Colloquium on Computational Complexity TR96-056*, http://www.eccc.uni-trier.de/eccc-local/Lists/TR-1996.html
15. O. Goldreich, S. Goldwasser, and S. Halevi, Eliminating the Decryption Error in the Ajtai-Dwork Cryptosystem, *to appear, Proc. CRYPTO'97*
16. S. Goldwasser and S. Micali, Probabilistic Encryption, *J. Comput. System Sci. 28*, 1984, pp. 270–299
17. S. Goldwasser, S. Micali, and R. Rivest, A "Paradoxical" Solution to the Signature Problem, *SIAM J. Computing 17*, 1988, pp. 281–308
18. M. Grötschel, Lovász, A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*, Springer, Algorithms and Combinatorics 2, 1988
19. P.M. Gruber, C.G. Lekkerkerker, *Geometry of Numbers*, North-Holland, 1987
20. J. Hastad, Solving Simultaneous Modular Equations of Low Degree, *SIAM J. Computing 17*(2), pp.336–341, 1988
21. R. Impagliazzo and M. Naor, Efficient Cryptographic Schemes Provably as Secure as Subset Sum, *J. Cryptology 9*, pp. 199–216, 1996
22. J.C. Lagarias, A.M. Odlyzko, Solving low-density subset sum problems, *Journal of the Association for Computing Machinery 32* pp. 229-246, 1985. An earlier version appeared in *Proc. 24th Annual Symposium on Foundations of Computer Science*, 1983
23. M. Luby, **Pseudo-randomness and applications**, Princeton University Press, 1996.
24. M. Naor and M. Yung, Universal One-Way Hash Functions and Their Cryptographic Applications, Proceedings 21th Annual ACM Symposium on Theory of Computing, 1989, pp. 33–43
25. A. Shamir, A Polynomial-Time Algorithm for Breaking the Basic Merkle-Hellman Cryptosystem, *Proc. 23rd Annual Symposium on Foundations of Computer Science*, 1982, pp. 145–152
26. A. C. Yao, Theory and Applications of Trapdoor Functions, *Proc. 23rd Annual Symposium on Foundations of Computer Science*, 1982, pp. 80–91

This article was processed using the LaTeX macro package with LLNCS style