

# Generalized Compact Knapsacks are Collision Resistant <sup>\*</sup>

Vadim Lyubashevsky      Daniele Micciancio

University of California, San Diego  
9500 Gilman Drive, La Jolla, CA 92093-0404, USA  
{vlyubash,daniele}@cs.ucsd.edu

**Abstract.** In (Micciancio, FOCS 2002), it was proved that solving the generalized compact knapsack problem *on the average* is as hard as solving certain *worst-case* problems for cyclic lattices. This result immediately yielded very efficient one-way functions whose security was based on worst-case hardness assumptions. In this work, we show that, while the function proposed by Micciancio is not collision resistant, it can be easily modified to achieve collision resistance under essentially the same complexity assumptions on cyclic lattices. Our modified function is obtained as a special case of a more general result, which yields efficient collision-resistant hash functions based on the worst-case hardness of various new problems. These include new problems from algebraic number theory as well as classic lattice problems (e.g., the shortest vector problem) over *ideal lattices*, a class of lattices that includes cyclic lattices as a special case.

## 1 Introduction

Ever since Ajtai's discovery of a function whose average-case hardness can be proved based on worst-case complexity assumptions about lattices [2], the possibility of building cryptographic functions whose security is based on worst-case problems has been very alluring. Ajtai's initial discovery [2] and subsequent developments [5, 15, 17] are very interesting from a theoretical point of view because they are essentially the only problems for which such a worst-case / average-case connection is known. Unfortunately, the cryptographic functions proposed in these works are not efficient enough to be practical. The source of impracticality is the use of lattices, which are described as  $n \times n$  integer matrices. This results in cryptographic functions with key size and computation time at least quadratic in the security parameter  $n$ .

A step in the direction of creating efficient cryptographic functions based on worst-case hardness was taken by Micciancio [14]. He showed how to create a family of efficiently computable *one-way functions*, namely, the generalized

---

<sup>\*</sup> The full version of this extended abstract appears in ECCV TR05-142. Research supported by NSF CAREER 0093029 and NSF ITR 0313241

compact knapsack functions, whose security is based on a certain problem for a particular class of lattices, called cyclic lattices. These lattices admit a much more compact representation than general ones, and the resulting functions can be described and evaluated in time almost linear in  $n$ . However, one-wayness is a rather weak security property, interesting mostly from a theoretical point of view, because it is sufficient to prove the existence (via polynomial time, but rather impractical, constructions) of other cryptographic primitives, like commitment schemes, digital signatures, and private-key encryption. By contrast, the (inefficient) functions based on general lattices considered in [2, 5, 15, 17] are collision-resistant hash functions, a much more useful cryptographic primitive.

In this work, we take the next step in creating efficient cryptographic functions based on worst-case assumptions. We show how to create efficient, collision-resistant *hash functions* whose security is based on standard lattice problems for *ideal lattices* (i.e., lattices that can be described as ideals of certain polynomial rings). With current hash functions that are not based on any hardness assumptions, but used in practice, being broken [23, 24, 4], we believe that it may be an appropriate time to consider using efficient hash functions which do have an underlying hardness assumption, especially worst-case ones.

*Our contributions and comparison with related work.* The generalized knapsack problem is the following: given  $m$  random elements  $a_1, \dots, a_m$  in a ring  $R$ , and a target  $t \in R$ , find  $z_1, \dots, z_m \in D$  such that  $\sum a_i z_i = t$ , where  $D$  is some fixed subset of  $R$ . In [14], it was shown that for appropriate choices of  $R$  and  $D$ , the generalized compact knapsack problem is a one-way function with security based on the worst-case hardness of problems for lattices that can be represented as ideals in the ring  $\mathbb{Z}[x]/\langle x^n - 1 \rangle$  (i.e. cyclic lattices). In this work, we show how to construct collision-resistant hash functions based on the hardness of problems for lattices that can be represented as ideals in the ring  $\mathbb{Z}[x]/\langle f \rangle$ , where  $f$  can be one of infinitely many polynomials, including  $x^n - 1$ . Thus our result has two desirable features: it weakens the complexity assumption while strengthening the cryptographic primitive. As in [14], our functions are an instance of the generalized compact knapsack problem, but with ring  $R$  and subset  $D$  instantiated in a different way. The way we change ring  $R$  and subset  $D$  is simple, but essential, as we can show that the generalized compact knapsack instances considered in [14] are not collision resistant.

Concurrently with, and independently from our work, Peikert and Rosen [18] have shown, using very similar techniques, that the one-way function in [14] is not collision resistant and showed how to construct collision-resistant hash functions based on the hardness of finding the shortest vector for lattices which correspond to ideals in the ring  $\mathbb{Z}[x]/\langle x^n - 1 \rangle$ . While our more general result is interesting from a purely theoretical standpoint, it turns out that choices of certain  $f$  other than  $x^n - 1$  result in somewhat better hash functions, making our generalization also of practical use. Also, our hardness assumptions are formulated in a way that leads to natural connections with algebraic number theory, and we are able to relate our complexity assumptions to problems from that area. We believe that this will further our understanding of ideal lattices.

There have been many proposed cryptographic primitives whose hardness relied on the knapsack problem (e.g., [13, 7, 6]), but attacks against them (e.g., [21, 11, 22]) rendered the primitives impractical. These attacks, however, were applied to a group-based knapsack problem, and it is unclear how to apply them to our ring-based one. Also, none of those primitives had a reduction to worst-case instances of lattice problems, and, to the best of our knowledge, there are no known efficient algorithms that are able to solve lattice problems in the worst case (such as shortest vector) for lattices of dimension  $\approx 100$ . Of course, the hardness of our primitive is based on worst-case problems for *ideal* lattices, and very little is known about these. Still, currently there appear to be no algorithms able to take advantage of the ring structure that these lattices possess (see [14] for a discussion of known algorithms for cyclic lattices). Determining the worst-case hardness of lattice problems for ideal lattices is a very interesting open problem.

The ring-based cryptosystem NTRU [10] uses lattices that are similar to ours. While that cryptosystem has no known security proofs (not even one based on average-case assumptions), it has resisted attacks. This is perhaps due to the inherent hardness of ring-based cryptographic constructions that are used in [10] as well as in our work. While we only construct a hash function, our work may be viewed as a strong justification for using such ring based constructions. Our hope is that we have taken another step in the direction of constructing provably secure and *efficient* cryptosystems based on worst case hardness of lattice problems.

*The hash function.* We now give an informal description of the hash function families that we will be proving collision resistant. Given a ring  $R = \mathbb{Z}_p[x]/\langle f \rangle$ , where  $f \in \mathbb{Z}[x]$  is a monic, irreducible polynomial of degree  $n$  and  $p$  is an integer of order roughly  $n^2$ , generate  $m$  random elements  $a_1, \dots, a_m \in R$ , where  $m$  is a constant. The ordered  $m$ -tuple  $h = (a_1, \dots, a_m) \in R^m$  is our hash function. It will map elements in  $D^m$ , where  $D$  is a strategically chosen subset of  $R$ , to  $R$ . For an element  $b = (b_1, \dots, b_m) \in D^m$ , the hash is  $h(b) = \sum_{i=1}^m a_i \cdot b_i$ . Notice that the size of the key (the hash function) is  $O(mn \log p) = O(n \log n)$ , and the operation  $a_i \cdot b_i$  can be done in time  $O(n \log n \log \log n)$  by using the fast Fourier transform, for appropriate choice of the polynomial  $f$ . Since  $m$  is a constant, hashing requires time  $O(n \log n \log \log n)$ . To prove that our hash function family is collision resistant, we will show that if there is a polynomial-time algorithm that succeeds with non-negligible probability in finding  $b \neq b' \in D^m$  such that  $h(b) = h(b')$ , for a randomly chosen hash function  $h \in R^m$ , then a certain problem called the “shortest polynomial problem” is solvable in polynomial time for *every* ideal of the ring  $\mathbb{Z}[x]/\langle f \rangle$ . We then show that the shortest polynomial problem is equivalent to some lattice and algebraic number theory problems.

*Paper outline.* Our main result and techniques rely on a connection between lattices and ideals of certain rings, which we describe in section 3. In section 4, we define the worst case problems on which we will be basing the security of our hash function. We formally define the hash function families in section 5.1 and show the worst-case to average-case reduction in section 5.2.

## 2 Preliminaries

### 2.1 Algebra

Let  $\mathbb{Z}[x]$  and  $\mathbb{R}[x]$  be the sets of polynomials with integer and real coefficients respectively. We identify polynomials (of degree  $< n$ ) with the corresponding  $n$ -dimensional vectors having the coefficients of the polynomial as coordinates. We define the  $\ell_p$  norm  $\|g(x)\|_p$  of  $g(x) \in \mathbb{Z}[x]$  as the norm of the corresponding vector, and the product of two  $n$ -dimensional vectors  $\mathbf{x} \cdot \mathbf{y}$  as the  $(2n - 1)$ -dimensional vector associated to the product of the corresponding polynomials.

Let  $R$  be a ring. The smallest ideal of  $R$  containing a subset  $S \subseteq R$  is denoted  $\langle S \rangle$ . Much of our work deals with the rings  $\mathbb{Z}[x]/\langle f \rangle$  where  $f$  is monic and irreducible. When  $f$  is a monic polynomial of degree  $n$ , every equivalence class  $(g + \langle f \rangle) \in (\mathbb{Z}[x]/\langle f \rangle)$  has a unique representative  $g' \in (g + \langle f \rangle)$  of degree less than  $n$ . This representative is denoted  $(g \bmod f)$  and can be efficiently computed using the standard division algorithm. We endow the ring  $\mathbb{Z}[x]/\langle f \rangle$  with the (infinity) norm  $\|(g + \langle f \rangle)\|_f = \|g \bmod f\|_\infty$ . Notice that the function  $\|\cdot\|_f$  is well defined (i.e., it does not depend on the choice of representative  $g$ ) and it is indeed a norm (i.e., it satisfies the positivity and triangle inequality properties). As shorthand, we will sometimes write  $\|g\|_f$  instead of  $\|g + \langle f \rangle\|_f$ . Also, whenever there is no confusion from context, instead of writing  $g + \langle f \rangle$  for elements of  $\mathbb{Z}[x]/\langle f \rangle$ , we just write  $g$ .

### 2.2 Lattices

An  $n$ -dimensional *integer lattice* is a subgroup of  $\mathbb{Z}^n$  generated by linearly independent vectors  $\mathbf{b}_1, \dots, \mathbf{b}_n \in \mathbb{Z}^n$ . The set of vectors  $\mathbf{b}_1, \dots, \mathbf{b}_n$  is called a *basis* for the lattice, and can be compactly represented by the matrix  $\mathbf{B}$  having the basis vectors as columns. The lattice generated by  $\mathbf{B}$  is denoted  $\mathcal{L}(\mathbf{B})$ . The dual of this lattice, denoted  $\mathcal{L}(\mathbf{B})^*$ , is the lattice generated by the matrix  $\mathbf{B}^{-T}$ , and consists of all vectors that have integer scalar product with all lattice vectors. For any basis  $\mathbf{B}$ , we define the fundamental parallelepiped  $\mathcal{P}(\mathbf{B}) = \{\mathbf{B}\mathbf{x} : \forall i. 0 \leq x_i < 1\}$ . Sampling random lattice points from the fundamental parallelepiped associated to a given sublattice can be done in polynomial time [16, Proposition 8.2].

The *minimum distance* of a lattice  $\mathcal{L}(\mathbf{B})$  is the minimum distance between any two (distinct) lattice points and equals the length of the shortest nonzero lattice vector. The minimum distance can be defined with respect to any norm. For any  $p \geq 1$ , the  $\ell_p$  norm of a vector  $\mathbf{x}$  is defined by  $\|\mathbf{x}\|_p = \sqrt[p]{\sum_i |x_i|^p}$  and the corresponding minimum distance is denoted

$$\lambda_1^p(\mathcal{L}(\mathbf{B})) = \min\{\|\mathbf{x} - \mathbf{y}\|_p : \mathbf{x} \neq \mathbf{y} \in \mathcal{L}(\mathbf{B})\} = \min\{\|\mathbf{x}\|_p : \mathbf{x} \in \mathcal{L}(\mathbf{B}) \setminus \{\mathbf{0}\}\}.$$

Each norm gives rise to a corresponding computational problem  $SVPP_\gamma^p$  (the  $\gamma$ -approximate *Shortest Vector Problem* in the  $\ell_p$  norm): given a lattice  $\mathcal{L}(\mathbf{B})$ , find a nonzero vector  $\mathbf{v} \in \mathcal{L}(\mathbf{B})$  such that  $\|\mathbf{v}\|_p \leq \gamma \lambda_1^p(\mathcal{L}(\mathbf{B}))$ . We also consider the restriction of  $SVP$  to specific classes of lattices. The restriction of  $SVP$  to a class of lattices  $\Lambda$  is denoted  $\Lambda$ - $SVP$ . (E.g, [14] considers *Cyclic-SVP*).

The notion of minimum distance can be generalized to define the  $i$ th successive minimum (in the  $\ell_p$  norm)  $\lambda_i^p(\mathcal{L}(\mathbf{B}))$  as the smallest radius  $r$  such that the closed sphere  $\bar{\mathcal{B}}_p(r) = \{\mathbf{x} : \|\mathbf{x}\|_p \leq r\}$  contains  $i$  linearly independent lattice points:  $\lambda_i^p(\mathcal{L}(\mathbf{B})) = \min\{r : \dim(\text{span}(\mathcal{L}(\mathbf{B}) \cap \bar{\mathcal{B}}_p(r))) \geq i\}$ .

In this work, we focus on the infinity norm  $\|\mathbf{x}\|_\infty = \lim_{p \rightarrow \infty} \|\mathbf{x}\|_p = \max_i |x_i|$  since it is the most natural and convenient norm when dealing with polynomials, but most of our results are easily translated to other norms as well. The shortest vector problem in the infinity norm  $SV P_\gamma^\infty$  was shown to be  $NP$ -hard for factor up to  $\gamma(n) = n^{1/\log \log n}$  by Dinur [8]. The asymptotically fastest algorithm for computing the shortest vector exactly takes time  $2^{O(n)}$  [3] and the best polynomial time algorithm approximates the shortest vector to within a factor of  $2^{O(\frac{n \log \log n}{\log n})}$  [3],[20],[12]. It is conjectured that approximating  $SV P$  to within a polynomial factor is a hard problem, although it is shown that (under standard complexity assumptions) for small polynomial factors it is not  $NP$ -hard [1], [9].

### 2.3 Gaussian distribution

Let  $X$  and  $Y$  be random variables over a set  $A$  with probability density functions  $\delta_X$  and  $\delta_Y$ . We denote the statistical distance between  $X$  and  $Y$  by  $\Delta(X, Y)$ .

For any vectors  $\mathbf{c}, \mathbf{x}$  and any  $s > 0$ , let  $\rho_{s,\mathbf{c}}(\mathbf{x}) = e^{-\pi\|(\mathbf{x}-\mathbf{c})/s\|^2}$  be a Gaussian function centered in  $\mathbf{c}$  scaled by a factor of  $s$ . The total measure associated to  $\rho_{s,\mathbf{c}}$  is  $\int_{\mathbf{x} \in \mathbb{R}^n} \rho_{s,\mathbf{c}}(\mathbf{x}) d\mathbf{x} = s^n$ . So,  $\int_{\mathbf{x} \in \mathbb{R}^n} (\rho_{s,\mathbf{c}}(\mathbf{x})/s^n) d\mathbf{x} = 1$  and  $\rho_{s,\mathbf{c}}/s^n$  is a probability density function. The distribution  $\rho_{s,\mathbf{c}}/s^n$  can be efficiently approximated using standard techniques (see [17]), so in the rest of the paper we make the simplifying assumption that we can sample from  $\rho_{s,\mathbf{c}}/s^n$  exactly and work with real numbers.

Functions are extended to sets in the usual way; e.g.,  $\rho_{s,\mathbf{c}}(A) = \sum_{\mathbf{x} \in A} \rho_{s,\mathbf{c}}(\mathbf{x})$  for any countable set  $A$ . For any  $s, \mathbf{c}$  and lattice  $\Lambda$ , define the discrete probability distribution (over the lattice  $\Lambda$ )  $D_{\Lambda,s,\mathbf{c}}(\mathbf{x}) = \frac{\rho_{s,\mathbf{c}}(\mathbf{x})}{\rho_{s,\mathbf{c}}(\Lambda)}$ , where  $\mathbf{x} \in \Lambda$ . Intuitively,  $D_{\Lambda,s,\mathbf{c}}$  is the conditional probability<sup>1</sup> that  $(\rho_{s,\mathbf{c}}/s^n) = \mathbf{x}$  given  $(\rho_{s,\mathbf{c}}/s^n) \in \Lambda$ . For brevity, we sometimes omit  $s$  or  $\mathbf{c}$  from the notation  $\rho_{s,\mathbf{c}}$  and  $D_{\Lambda,s,\mathbf{c}}$ . When  $\mathbf{c}$  or  $s$  are not specified, we assume that they are the origin and 1 respectively.

In [17] Gaussian distributions are used to define a new lattice invariant (called the *smoothing parameter*) defined below, and many important properties of this parameter are established. The following properties will be used in this paper.

**Definition 1.** For an  $n$ -dimensional lattice  $\Lambda$ , and positive real  $\epsilon > 0$ , the smoothing parameter  $\eta_\epsilon(\Lambda)$  is the smallest  $s$  such that  $\rho_{1/s}(\Lambda^* \setminus \{\mathbf{0}\}) \leq \epsilon$ .

**Lemma 1 ([17, Lemma 4.1]).** Let  $\rho_s/s^n \bmod \mathbf{B}$  be the distribution obtained by sampling a point according to the probability density function  $\rho_s/s^n$  and reducing the result modulo  $\mathbf{B}$ . For any lattice  $\mathcal{L}(\mathbf{B})$ , the statistical distance between  $\rho_s/s^n \bmod \mathbf{B}$  and the uniform distribution over  $\mathcal{P}(\mathbf{B})$  is at most  $\frac{1}{2}\rho_{1/s}(\mathcal{L}(\mathbf{B})^* \setminus \{\mathbf{0}\})$ . In particular, if  $s \geq \eta_\epsilon(\mathcal{L}(\mathbf{B}))$ , then the distance  $\Delta(\rho_s/s^n \bmod \mathbf{B}, U(\mathcal{P}(\mathbf{B})))$  is at most  $\epsilon/2$ .

<sup>1</sup> We are conditioning on an event that has probability 0; this can be made rigorous by standard techniques.

**Lemma 2** ([17, Lemma 3.3]). *For any  $n$ -dimensional lattice  $\Lambda$  and positive real  $\epsilon > 0$ ,*

$$\eta_\epsilon(\Lambda) \leq \sqrt{\frac{\ln(2n(1+1/\epsilon))}{\pi}} \cdot \lambda_n^2(\Lambda) \leq \sqrt{\frac{n \ln(2n(1+1/\epsilon))}{\pi}} \cdot \lambda_n^\infty(\Lambda).$$

### 3 Generalized compact knapsacks and ideal lattices

In [14], Micciancio introduced the following generalization of the compact knapsack problem. Let  $R$  be a ring,  $D \subset R$  a subset, and  $m \geq 1$  a positive integer. The generalized knapsack function family  $\mathcal{H}(R, D, m)$  is the collection of all functions  $\mathfrak{h}_\mathbf{a} : D^m \rightarrow R$  indexed by  $\mathbf{a} \in R^m$  mapping  $\mathbf{b} \in D^m$  to  $\mathfrak{h}_\mathbf{a}(\mathbf{b}) = \sum_{i=1}^m b_i \cdot a_i \in R$ .

For any function family  $\mathcal{H}$ , define the problem  $Col_{\mathcal{H}}$  as follows: given a function  $\mathfrak{h} \in \mathcal{H}$ , find a collision, i.e., a pair of inputs  $\mathbf{b}, \mathbf{c} \in D^m$  such that  $\mathbf{b} \neq \mathbf{c}$  and  $\mathfrak{h}(\mathbf{b}) = \mathfrak{h}(\mathbf{c})$ . If there is no polynomial time algorithm that can solve  $Col_{\mathcal{H}}$  with non-negligible probability when given an  $\mathfrak{h}$  which is distributed uniformly at random in  $\mathcal{H}$ , then we say that  $\mathcal{H}$  is a collision resistant family of hash functions.

Let  $f \in \mathbb{Z}[x]$  be a monic polynomial of degree  $n$ , and consider the quotient ring  $\mathbb{Z}[x]/\langle f \rangle$ . Using the standard set of representatives  $\{(g \bmod f) : g \in \mathbb{Z}[x]\}$ , and our identification of polynomials with vectors, the quotient ring  $\mathbb{Z}[x]/\langle f \rangle$  is isomorphic (as an additive group) to the integer lattice  $\mathbb{Z}^n$ , and any ideal  $I \subseteq \mathbb{Z}[x]/\langle f \rangle$  defines a corresponding integer sublattice  $\mathcal{L}(I) \subseteq \mathbb{Z}^n$ . Notice that not every integer lattice  $\mathcal{L}(\mathbf{B}) \subseteq \mathbb{Z}^n$  can be represented this way.<sup>2</sup> We define ideal lattices as lattices that admit such a representation.

**Definition 2.** *An ideal lattice is an integer lattice  $\mathcal{L}(\mathbf{B}) \subseteq \mathbb{Z}^n$  such that  $\mathcal{L}(\mathbf{B}) = \{g \bmod f : g \in I\}$  for some monic polynomial  $f$  of degree  $n$  and ideal  $I \subseteq \mathbb{Z}[x]/\langle f \rangle$ .*

It turns out that the relevant properties of  $f$  for the resulting function to be collision resistant are:

- $f$  should be irreducible.
- the ring norm  $\|g\|_f$  is not much bigger than  $\|g\|_\infty$  for any polynomial  $g$ , in a quantitative sense to be explained later.

The first property implies that every ideal of the ring  $\mathbb{Z}[x]/\langle f \rangle$  defines a full-rank lattice in  $\mathbb{Z}^n$  and plays a fundamental role in our proofs.

**Lemma 3.** *Every ideal  $I$  of  $\mathbb{Z}[x]/\langle f \rangle$ , where  $f$  is a monic, irreducible integer polynomial of degree  $n$ , is isomorphic to a full-rank lattice in  $\mathbb{Z}^n$ .*

The second property affects the strength of our security proofs: the smaller the ratio  $\|g\|_f / \|g\|_\infty$  is, the harder to break our functions seems to be. We elaborate on the second property by defining a quantitative parameter (the expansion factor) that captures the relation between  $\|\cdot\|_\infty$  and  $\|\cdot\|_f$ .

<sup>2</sup> Take, for example, the 2-dimensional lattice generated by the vectors  $(2, 0)$  and  $(0, 1)$  (or in terms of polynomials, by  $2x$  and  $1$ ). This lattice cannot be represented by an ideal, because any ideal containing  $1$  must also contain the polynomial  $1 \cdot x$ , but the vector  $(1, 0)$  (corresponding to the polynomial  $x$ ) does not belong to the lattice.

### 3.1 The expansion factor

Notice that when we reduce a polynomial  $g$  modulo  $f$ , the maximum coefficient of  $g$  can increase by quite a bit, and thus  $\|g\|_f$  could be a lot bigger than  $\|g\|_\infty$ . For example if  $f = x^n - 2x^{n-1}$ , then  $x^{2n} \equiv 2^{n+1}x^{n-1}$  modulo  $f$ . On the other hand, if  $f = x^n - 1$ , we can never have such an exponential growth of coefficients. We capture this property of  $f$  by defining the *expansion factor* of  $f$  as

$$EF(f, k) = \max_{g \in \mathbb{Z}[x], \deg(g) \leq k(\deg(f)-1)} \|g\|_f / \|g\|_\infty$$

The below theorem gives tight bounds for the expansion factor of certain polynomials that have small expansion factors.

**Theorem 1.**

$$(1) EF(x^{n-1} + x^{n-2} + \dots + 1, k) \leq 2k \quad (2) EF(x^n + 1, k) \leq k$$

In the full version of this work, we also provide some general formulas that upper bound the expansion factors of arbitrary polynomials.

## 4 Worst case problems

In this section we define the worst case problems and provide reductions among them. Because of the correspondence between ideals and integer lattices, we can use the successive minima notation used for lattices for ideals as well. So for any ideal  $I$  of  $\mathbb{Z}[x]/\langle f \rangle$ , where  $f$  is a monic integer polynomial, we'll define  $\lambda_i^p(I)$  to be  $\lambda_i^p(\mathcal{L}(I))$ .

**Definition 3.** *In the approximate Shortest Polynomial Problem ( $SPP_\gamma(I)$ ), we are given an ideal  $I \subseteq \mathbb{Z}[x]/\langle f \rangle$  where  $f$  is a monic polynomial of degree  $n$ , and we are asked to find a  $g \in I$  such that  $g \neq 0$  and  $\|g\|_f \leq \gamma \lambda_1^\infty(I)$ .*

As for the shortest vector problem, we can consider the restriction of  $SPP$  to specific classes of ideals. We will write  $f$ - $SPP$  for  $SPP$  restricted to ideals of the ring  $\mathbb{Z}[x]/\langle f \rangle$ . The  $f$ - $SPP$  problem for any monic, irreducible  $f$  is the main worst-case problem of this work, as it is the problem upon which the security of our hash functions will be based. Since  $SPP$  is a new problem whose hardness has not been explored, we show that other better-known problems can be reduced to it. If we denote by  $\mathcal{I}(f)$  the set of lattices that are isomorphic (as additive groups) to ideals of  $\mathbb{Z}[x]/\langle f \rangle$  where  $f$  is monic, then there's a straightforward reduction from  $\mathcal{I}(f)$ - $SV P_\gamma$  to  $f$ - $SPP_\gamma$  (and also the other way around).

Lattices in the class  $\mathcal{I}(x^n - 1)$  (cyclic lattices) do not fall into the category of lattices that are isomorphic to ideals of  $\mathbb{Z}[x]/\langle f \rangle$  for an irreducible  $f$  (since  $x^n - 1$  is not irreducible). In the full version, we give a reduction from  $(x^n - 1)$ - $SPP_{2\gamma}$  to  $(x^{n-1} + x^{n-2} + \dots + 1)$ - $SPP_\gamma$ , thus establishing the security of hash functions based on the hardness of the shortest vector problem for cyclic lattices of prime dimension. Another problem that we reduce to  $SPP$  is the problem of finding

complex numbers with small conjugates in ideals of integers of certain number fields. This problem and the reduction is described in detail in the full version.

Now we state a lemma which shows that if  $I$  is an ideal of  $\mathbb{Z}[x]/\langle f \rangle$  where  $f$  is monic and irreducible, then  $\lambda_n^\infty(I)$  cannot be much bigger than  $\lambda_1^\infty(I)$ .

**Lemma 4.** *For all ideals  $I$  of  $\mathbb{Z}[x]/\langle f \rangle$  where  $f$  is a monic, irreducible polynomial of degree  $n$ , we have  $\lambda_n^\infty(I) \leq EF(f, 2)\lambda_1^\infty(I)$*

*Proof.* Let  $g$  be a polynomial in  $I$  of degree less than  $n$  such that  $\|g\|_\infty = \lambda_1^\infty(I)$ . Then consider the polynomials  $g, gx, \dots, gx^{n-1}$ . By lemma 3, the polynomials  $g, gx, \dots, gx^{n-1}$  are linearly independent. And since the maximum degree of any of these polynomials is  $2n - 2$ ,  $\|gx^i\|_f \leq EF(f, 2)\|gx^i\|_\infty \leq EF(f, 2)\|g\|_\infty = EF(f, 2)\lambda_1^\infty(I)$  for all  $0 \leq i \leq n - 1$ .

We now define the incremental version of *SPP*. In this version, we are not looking for the shortest polynomial, but for a polynomial that is smaller than the one given to us. We will be reducing this problem to the average-case problem.

**Definition 4.** *In the approximate Incremental Shortest Polynomial Problem ( $IncSPP_\gamma(I, g)$ ), we are given  $I$  and a  $g \in I$  such that  $\|g\|_f > \gamma\lambda_1^\infty(I)$  and are asked to return an  $h \in I$  such that  $\|h\|_f \neq 0$  and  $\|h\|_f \leq \|g\|_f/2$ .*

We define the restricted version of *IncSPP* in the same way as the restricted version for *SPP*.

**Lemma 5.** *There is a polynomial time reduction from  $f$ -*SPP* $_\gamma$  to  $f$ -*IncSPP* $_\gamma$ .*

## 5 Collision resistant hash function families

In this section, we define families of hash functions which are instances of generalized compact knapsacks and prove that finding collisions in these hash functions is at least as hard as solving the approximate shortest polynomial problem.

### 5.1 The hash function families

The hash function family  $\mathcal{H}(R, D, m)$  we will be considering in this paper will be instances of generalized knapsacks instantiated as follows. Let  $f \in \mathbb{Z}[x]$  be an irreducible, monic polynomial of degree  $n$  with expansion factor  $EF(f, 3) \leq \mathcal{E}$ . Let the ring  $R$  be  $\mathbb{Z}_p[x]/\langle f \rangle$  for some integer  $p$ , and let  $D = \{g \in R : \|g\|_f \leq d\}$  for some positive integer  $d$ . The family of functions  $\mathcal{H}$  is mapping elements from  $D^m$  to  $R$  where  $|D^m| = (2d + 1)^{nm}$  and  $|R| = p^n$ . So if  $m > \frac{\log p}{\log 2d}$ , then  $\mathcal{H}$  will be a family of functions that have collisions. We will only be interested in such families. We will now state the main theorem:

**Theorem 2.** *Let  $\mathcal{H}$  be a hash function family as above with  $m > \frac{\log p}{\log 2d}$  and  $p > 2\mathcal{E}dmn^{1.5} \log n$ . Then, for  $\gamma = 8\mathcal{E}^2dmn \log^2 n$ , there is a polynomial time reduction from  $f$ -*SPP* $_\gamma(I)$  for any  $I$  to  $Col_{\mathcal{H}}(\mathfrak{h})$  where  $\mathfrak{h}$  is chosen uniformly at random from  $\mathcal{H}$ .*

The proof of the theorem is given in the next subsection. To achieve the best approximation factor for  $f\text{-SPP}_\gamma(I)$ , we can set  $m = \Theta(\log n, \log \mathcal{E})$  and  $d = \Theta(\log n)$ . This makes  $\gamma = \tilde{O}(n)\mathcal{E}^2$ . For purposes of being able to compute the function faster, though, it is useful to have  $m$  be smaller than  $\Theta(\log n)$ . It is possible to make  $m$  constant at the expense of being able to approximate  $f\text{-SPP}$  only to a factor of  $\gamma = \tilde{O}(n^{1+\delta})\mathcal{E}^2$ . To be able to set  $m$  to a constant, we can set  $d = n^\delta$  for some  $\delta > 0$ . Then we can set  $m = \frac{\log(\mathcal{E})}{\delta \log n} + \frac{2+\delta}{\delta} + o(1)$ .

In order to get the “tightest” reduction, we should pick an  $f$  such that the bound  $\mathcal{E}$  on  $f$ 's expansion factor is small. In theorem 1, we show that we can set  $\mathcal{E}$  to be 3 and 6 for polynomials of the form  $x^n + 1$  and  $x^{n-1} + x^{n-2} + \dots + 1$  respectively. The polynomial  $x^n + 1$  is irreducible whenever  $n$  is a power of 2 and  $x^{n-1} + x^{n-2} + \dots + 1$  is irreducible for prime  $n$ , so those are good choices for  $f$ . Among other possible  $f$ 's with constant bounds for  $EF(f, 3)$  are polynomials of the form  $x^n \pm x \pm 1$  (see [19, Chapter 2.3.2] for sufficient conditions for the irreducibility of polynomials of this form).

*Some sample instantiations of the hash function.* If we let  $f = x^{126} + \dots + x + 1, n = 126, d = 8, m = 8$ , and  $p \approx 2^{23}$ , then our hash function is mapping  $|2d|^{mn} = 4032$  bits to  $|R_p| = p^n \approx 2900$  bits. If we want to base our hardness assumption on lattices of higher dimension, we can instantiate  $f = x^{256} + \dots + x + 1, n = 126, p \approx 2^{25}, d = 8, m = 8$ , and our hash function will be mapping 8192 bits to  $p^n \approx 6400$  bits. If we instead let  $f = x^{256} + 1$ , we can let  $p$  be half as small (because the expansion factor for  $x^n + 1$  is half of the expansion factor of  $x^n + \dots + x + 1$ ) and thus we will be mapping 8192 bits to around 6150 bits.

## 5.2 Finding collisions is hard

In this section, we will provide the proof of theorem 2. Let  $\mathcal{H}$  be the family of hash functions described in the last subsection with  $p > 2\mathcal{E}dmn^{1.5} \log n$ . We will show that if one can solve in polynomial time, with non-negligible probability, the problem  $Col_{\mathcal{H}}(\mathfrak{h})$  where  $\mathfrak{h}$  is chosen uniformly at random from  $\mathcal{H}$ , then one can also solve  $f\text{-IncSPP}_\gamma(I, g)$  for any ideal  $I$  for  $\gamma = 8\mathcal{E}^2dmn \log^2 n$ . And since by lemma 5,  $f\text{-SPP}_\gamma(I) \leq f\text{-IncSPP}_\gamma(I, g)$ , we will have a reduction from  $f\text{-SPP}_\gamma(I)$  for any  $I$  to  $Col_{\mathcal{H}}(\mathfrak{h})$  for a random  $\mathfrak{h}$ . Let  $\mathcal{C}$  be an oracle such that when given a uniformly random  $\mathfrak{h} \in \mathcal{H}$ ,  $\mathcal{C}(\mathfrak{h})$  returns a solution to  $Col_{\mathcal{H}}(\mathfrak{h})$  with non-negligible probability in polynomial time. Now we proceed with giving an algorithm for  $f\text{-IncSPP}_\gamma$  when given access to oracle  $\mathcal{C}$ .

Given:  $I, g \in I$  such that  $g \neq 0$  and  $\|g\|_f > 8\mathcal{E}^2dmn \log^2 n \lambda_1^\infty(I)$   
 Find:  $h \in I$ , such that  $h \neq 0$  and  $\|h\|_f \leq \|g\|_f/2$ .

Without loss of generality, assume that  $g$  has degree less than  $n$  and thus  $\|g\|_\infty = \|g\|_f$ . So we are looking for an  $h$  such that  $\|h\|_f \leq \|g\|_\infty/2$ . In this section, it will be helpful to think of ideals  $I$  and  $\langle g \rangle$  as subgroups of  $\mathbb{Z}^n$  (or

equivalently, as sublattices of  $\mathbb{Z}^n$ ). Define a number  $s$  as

$$s = \frac{\|g\|_\infty}{8\mathcal{E}\sqrt{n}\log ndm} \geq \mathcal{E}\sqrt{n}(\log n)\lambda_1^\infty(I) \geq \sqrt{n}(\log n)\lambda_n^\infty(I) \geq \eta_\epsilon(I)$$

for  $\epsilon = (\log n)^{-2\log n}$ , where the last inequality follows by lemma 2, and the inequality before that is due to lemma 4. By lemma 1, it follows that if  $y \in \mathbb{R}^n$  where  $y \sim \rho_s/s^n$ , then  $\Delta(y + I, U(\mathbb{R}^n/I)) \leq (\log n)^{-2\log n}/2$ . (That is,  $y$  is in an almost uniformly random coset of  $\mathbb{R}^n/I$ ). By our definition of  $s$ , we have that  $\|g\|_\infty = 8\mathcal{E}dm s \sqrt{n} \log n$ . Now we will try to create an  $h \in I$  which is smaller than  $g$  using the procedure below. In the procedure, it may not be obvious how each step is performed, and the reader is referred to lemma 6 for a detailed explanation of each step.

- (1) for  $i = 1$  to  $m$ 
  - (2) generate a uniformly random coset of  $I/\langle g \rangle$  and let  $v_i$  be a polynomial in that coset
  - (3) generate  $y_i \in \mathbb{R}^n$  such that  $y_i$  has distribution  $\rho_s/s^n$  and consider  $y_i$  as a polynomial in  $\mathbb{R}[x]$
  - (4) let  $w_i$  be the unique polynomial in  $\mathbb{R}[x]$  of degree less than  $n$  with coefficients in the range  $[0, p)$  such that  $p(v_i + y_i) \equiv gw_i$  in  $\mathbb{R}^n/\langle pg \rangle$
  - (5)  $a_i = [w_i] \bmod p$  (where  $[w_i]$  means round each coefficient of  $w_i$  to the nearest integer)
- (6) call oracle  $\mathcal{C}(a_1, \dots, a_m)$ , and using its output, find polynomials  $z_1, \dots, z_m$  such that  $\|z_i\|_f \leq 2d$  and  $\sum z_i a_i \equiv 0$  in the ring  $\mathbb{Z}_p[x]/\langle f \rangle$ .
- (7) output  $h = \left( \sum \left( \frac{g(w_i - [w_i])}{p} - y_i \right) z_i \right) \bmod f$ .

To complete the proof, we will have to show five things: first, we have to prove that the above procedure runs in polynomial time, which is done in lemma 6. Then, in lemma 7, we show that in step (6) we are feeding the oracle  $\mathcal{C}$  with an  $\mathfrak{h} \in \mathcal{H}$  where the distribution of  $\mathfrak{h}$  is statistically close to uniform over  $\mathcal{H}$ . In lemma 8, we show that the resulting polynomial  $h$  is in the ideal  $I$ . We then show that if  $\mathcal{C}$  outputted a collision, then with non-negligible probability,  $\|h\|_f \leq \|g\|_\infty/2$  and that  $h \neq 0$ . This is done in lemmas 9 and 10 respectively. These five things prove that with non-negligible probability, we will obtain a solution to  $IncSPP_\gamma$ . If we happen to fail, we repeat the procedure again. Since each run of the procedure is independent, we will obtain a solution to  $IncSPP_\gamma$  in polynomial time.

**Lemma 6.** *The above procedure runs in polynomial time.*

*Proof.* We will show that each step in the algorithm takes polynomial time. In step (2), we need to generate a random element of  $I/\langle g \rangle$ . By lemma 3, the ideals  $I$  and  $\langle g \rangle$  can be thought of as  $\mathbb{Z}$ -modules of dimension  $n$ . Since  $\langle g \rangle \subseteq I$ , the group  $I/\langle g \rangle$  is finite, and we can efficiently generate a random element of  $I/\langle g \rangle$ . Step (4) of the algorithm will be justified in lemma 7. In step (5), we are just rounding each coefficient of  $w_i$  to the nearest integer and then reducing modulo  $p$ . Now each  $a_i$  can be thought of as an element of  $\mathbb{Z}_p[x]/\langle f \rangle$ , so in step (6)

we can feed  $(a_1, \dots, a_m)$  to the algorithm that solves  $\text{Col}_{\mathcal{H}}(a_1, \dots, a_m)$ . The algorithm will return  $(\alpha_1, \dots, \alpha_m), (\beta_1, \dots, \beta_m)$  where  $\alpha_i, \beta_i \in \mathbb{Z}[x]/\langle f \rangle$  such that  $\|\alpha_i\|_f, \|\beta_i\|_f \leq d$  and  $\sum a_i \alpha_i \equiv \sum a_i \beta_i$  in the ring  $\mathbb{Z}_p[x]/\langle f \rangle$ . Thus if we set  $z_i = \alpha_i - \beta_i$ , we will have  $\|z_i\|_f \leq 2d$  and  $\sum z_i a_i \equiv 0$  in the ring  $\mathbb{Z}_p[x]/\langle f \rangle$ .

**Lemma 7.** *Consider the polynomials  $a_i$  as elements in  $\mathbb{Z}_p^n$ . Then,*

$$\Delta((a_1, \dots, a_m), U(\mathbb{Z}_p^{n \times m})) \leq m\epsilon/2.$$

*Proof.* We know that  $v_i$  is in a uniformly random coset of  $I/\langle g \rangle$  and let's assume for now that  $y_i$  is in a uniformly random coset of  $\mathbb{R}^n/I$ . This means that  $v_i + y_i$  is in a uniformly random coset of  $\mathbb{R}^n/\langle g \rangle$  and thus the distribution of  $p(v_i + y_i)$  is in a uniformly random coset of  $\mathbb{R}^n/\langle pg \rangle$ . A basis for the additive group  $\langle pg \rangle$  is  $pg, pgx, \dots, pgx^{n-1}$ , thus every element of  $\mathbb{R}^n/\langle pg \rangle$  has a unique representative of the form  $\alpha_0 pg + \alpha_1 pgx + \dots + \alpha_{n-1} pgx^{n-1} = g(p\alpha_0 + p\alpha_1 x + \dots + p\alpha_{n-1} x^{n-1})$  for  $\alpha_i \in [0, 1)$ . So step (4) of the algorithm is justified, and since  $p(v_i + y_i)$  is in a uniformly random coset of  $\mathbb{R}^n/\langle pg \rangle$ , the coefficients of the polynomial  $w_i = p\alpha_0 + p\alpha_1 x + \dots + p\alpha_{n-1} x^{n-1}$  are uniform over the interval  $[0, p)$ , and thus the coefficients of  $[w_i]$  are uniform over the integers modulo  $p$ . The caveat is that  $y_i$  is not really in a uniformly random coset of  $\mathbb{R}^n/I$ , but is very close to it. By our choice of  $s$ , we have that  $\Delta(\rho_s/s^n + I, U(\mathbb{R}^n/I)) \leq \epsilon/2$ , and since  $a_i$  is a function of  $y_i$ , by a property of statistical distance, we have that  $\Delta(a_i, U(\mathbb{Z}_p^n)) \leq \epsilon/2$ . And since all the  $a_i$ 's are independent, we get that  $\Delta((a_1, \dots, a_m), U(\mathbb{Z}_p^{n \times m})) \leq m\epsilon/2$ .

Due to space constraints, the proofs of the below lemmas are omitted, and we refer the interested reader to the full version of this work.

**Lemma 8.**  $h \in I$

**Lemma 9.** *With probability negligibly different from 1,  $\|h\|_f \leq \frac{\|g\|_\infty}{2}$ .*

**Lemma 10.**  $\text{Pr}[h = 0 | (a_1, \dots, a_m), (z_1, \dots, z_m)] = \Omega(1)$

## 6 Conclusions and open problems

We gave constructions of efficient collision-resistant hash functions that can be proven secure based on the conjectured worst-case hardness of the shortest vector problem for ideal lattices, i.e., lattices that can be represented as ideals of  $\mathbb{Z}[x]/\langle f \rangle$  for some monic, irreducible polynomial  $f$ . Moreover, our results can be extended to certain polynomials  $f$  that are not irreducible, e.g., the polynomial  $f = x^n - 1$  corresponding to the class of cyclic lattices.

The central question raised by our work is the hardness of  $\mathcal{I}(f)$ -SVP, or equivalently, the hardness of  $f$ -SPP for different  $f$ 's. It is known that SVP is hard in the general case, and it was conjectured in [14] that  $\mathcal{I}(x^n - 1)$ -SVP is hard as well. We show worst-case to average-case reductions that work for many other  $f$ 's, so, in essence, we are giving more "targets" that can be proved hard.

Almost nothing is currently known about the complexity of problems for ideal lattices. We hope that our constructions of efficient collision-resistant hash functions based on the worst-case hardness of these problems provides motivation for their further study.

## References

1. D. Aharonov and O. Regev. Lattice problems in  $NP \cap coNP$ . *Journal of the ACM*, 52(5):749–765, 2005.
2. M. Ajtai. Generating hard instances of lattice problems. In *STOC*, pages 99–108, 1996.
3. M. Ajtai, R. Kumar, and D. Sivakumar. A sieve algorithm for the shortest lattice vector problem. In *STOC*, pages 601–610, 2001.
4. E. Biham, R. Chen, A. Joux, P. Carribault, W. Jalby, and C. Lemuet. Collisions of SHA-0 and reduced SHA-1. In *EUROCRYPT*, 2005.
5. J. Cai and A. Nerurkar. An improved worst-case to average-case connection for lattice problems. In *FOCS*, pages 468–477, 1997.
6. B. Chor and R. L. Rivest. A knapsack type public-key cryptosystem based on arithmetic in finite fields. *IEEE Trans. Inform. Theory*, 34(5):901–909, 1988.
7. I. Damgard. A design principle for hash functions. In *CRYPTO '89*, pages 416–427.
8. I. Dinur. Approximating  $SVP_\infty$  to within almost-polynomial factors is NP-hard. *Theor. Comput. Sci.*, 285(1):55–71, 2002.
9. O. Goldreich and S. Goldwasser. On the limits of nonapproximability of lattice problems. *J. Comput. Syst. Sci.*, 60(3), 2000.
10. J. Hoffstein, J. Pipher, and J. H. Silverman. Ntru: A ring-based public key cryptosystem. In *ANTS*, pages 267–288, 1998.
11. A. Joux and L. Granboulan. A practical attack against knapsack based hash functions. In *EUROCRYPT'94*, pages 58–66, 1994.
12. A. K. Lenstra, H. W. Lenstra Jr., and L. Lovasz. Factoring polynomials with rational coefficients. *Mathematische Annalen*, (261):513–534, 1982.
13. R.C. Merkle and M.E. Hellman. Hiding information and signatures in trapdoor knapsacks. *IEEE Transactions on Information Theory*, IT-24:525–530, 1978.
14. D. Micciancio. Generalized compact knapsacks, cyclic lattices, and efficient one-way functions from worst-case complexity assumptions. *Computational Complexity*. (To appear. Preliminary version in FOCS 2002).
15. D. Micciancio. Almost perfect lattices, the covering radius problem, and applications to Ajtai's connection factor. *SIAM J. on Computing*, 34(1):118–169, 2004.
16. D. Micciancio and S. Goldwasser. *Complexity Of Lattice Problems: A Cryptographic Perspective*. Kluwer Academic Publishers, 2002.
17. D. Micciancio and O. Regev. Worst-case to average-case reductions based on Gaussian measures. *SIAM J. on Computing*. (To appear. Preliminary version in FOCS 2004).
18. C. Peikert and A. Rosen. Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices. In *TCC*, 2006.
19. V. V. Prasolov. *Polynomials*, volume 11 of *Algorithms and Computation in Mathematics*. Springer-Verlag Berlin Heidelberg, 2004.
20. C. P. Schnorr. A hierarchy of polynomial time basis reduction algorithms. *Theoretical Computer Science*, 53:201–224, 1987.
21. A. Shamir. A polynomial time algorithm for breaking the basic Merkle-Hellman cryptosystem. *IEEE Transactions on Information Theory*, IT-30(5):699–704, 1984.
22. S. Vaudenay. Cryptanalysis of the Chor–Rivest cryptosystem. *Journal of Cryptology*, 14(2):87–100, 2001.
23. X. Wang, X. Lai, D. Feng, H. Chen, and X. Yu. Cryptanalysis for hash functions MD4 and RIPEMD. In *EUROCRYPT*, 2005.
24. X. Wang and H. Yu. How to break MD5 and other hash functions. In *EUROCRYPT*, 2005.