Lattice Based Cryptography: A Global Improvement

Daniele Micciancio
Laboratory for Computer Science
Massachusetts Institute of Technology
545 Technology Square
Cambridge MA, USA
`miccianc@theory.lcs.mit.edu`*

March 4, 1999

Abstract

We describe a general technique to simplify as well as to improve several lattice based cryptographic protocols. The technique is rather straightforward and is easily applied to the protocols, and gives both a simpler analysis and better performance than the original protocols. The improvement is global: the modified protocols are simpler, faster, require less storage, use less bandwidth and need less random bits than the originals. Moreover, the improvement is achieved without any loss in security: we formally prove that the modified protocols are at least as secure as the original ones. In fact, the modified protocols might even be more secure as the adversary gets less information. We exemplify our technique on the Goldreich-Goldwasser zero-knowledge proof systems for lattice problems and the GGH public key cryptosystem.

*Partially supported by DARPA grant DABT63-96-C-0018 and NTT grant 67627-00.

1 Introduction

Various cryptographic protocols based on the hardness of lattice problems requires the selection of a random point from the lattice. An illustrative example is the following. Assume we want to generate a "solved" instance of the closest vector problem. Let $\mathcal{L}(B)$ be a lattice. We can choose a lattice point $\mathbf{v} \in \mathcal{L}(B)$ at random and then add a small random error η to it. The error must be sufficiently small to assure that $\mathbf{v}$ is the lattice vector closest to $\mathbf{v}' = \mathbf{v} + \eta$, but large enough to make recovering $\mathbf{v}$ from $\mathbf{v}'$ computationally hard. Since lattices are infinite sets, sampling a lattice point uniformly at random is neither computationally possible nor a mathematically well defined operation. Still, the security of the protocol often relies on the "randomness" of the lattice point we start with. In the previous example, if $\mathbf{v}$ is chosen from a small set of lattice vectors, then we can easily recover it from $\mathbf{v}'$ by exhaustive search. This difficulty is usually overcome by starting from a lattice point uniformly chosen from a large region (say a sphere of radius exponentially bigger than the longest vector in the lattice basis B). If the region is sufficiently large, this is essentially the same as starting from a "random" lattice point.

We suggest to replace the sampling operation which introduces vectors much larger than the vectors describing the original lattice, by a reduction operation modulo the basis of the lattice. In our example, we choose a random error η, and output $\mathbf{v}' = \eta \bmod B$ as the target vector (see figure 1). Although we always start from the origin, the lattice vector closest to $\mathbf{v}'$ is not in general the origin, and it can be formally proved that finding the lattice vector closest to $\mathbf{v}'$ is as hard as finding it when we start from a random lattice point chosen from arbitrarily large regions. This techniques can be applied to various cryptographic protocols based on the complexity of lattice problems. The result is usually a simpler protocol (no need to perform the lattice sampling operation) with better running time and smaller communication complexity (the output vector can be exponentially shorter than in the original protocol, resulting in an n^2 bits saving per vector). We use our techniques to obtained improved versions of the Goldreich-Goldwasser zero-knowledge interactive proof systems for the Closest Vector Problem and the Shortest Vector Problem [6], and the GGH public key cryptosystem [8]. In the case of the GGH cryptosystem we also show how a clever choice of the public basis may lead to a modified cryptosystem with keys and ciphertexts more than one order of magnitude shorter than the original scheme.

The rest of the paper is organized as follows. In section 2 we recall some basic definitions and properties of lattices and statistical distance. In section 3 we apply our technique to the Goldreich-Goldwasser interactive proof systems for the closest and shortest vector problem. Finally, in section 4 we show how the same technique can be applied to the GGH cryptosystem to get faster encryption algorithms and shorter ciphertexts and keys. In all cases, we prove that the modified protocols are at least as secure as the original ones.

2 Preliminaries

In this section we recall some basic facts about lattices and statistical distance.

2.1 Lattices

Let $B = \{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$ be a set of n linearly independent vectors. The *lattice* generated by B is the set $\mathcal{L}(B) = \{\sum_i x_i \mathbf{b}_i \mid x_i \in \mathbb{Z}\}$ of all *integer* linear combinations of the vectors in B. The set B is called a basis for the lattice $\mathcal{L}(B)$, and it is usually identified with a matrix having the vectors $\mathbf{b}_i$ as columns. In matrix notation $\mathcal{L}(B) = \{B\mathbf{x} \mid \mathbf{x} \in \mathbb{Z}^n\}$. The basis of a lattice is not unique.

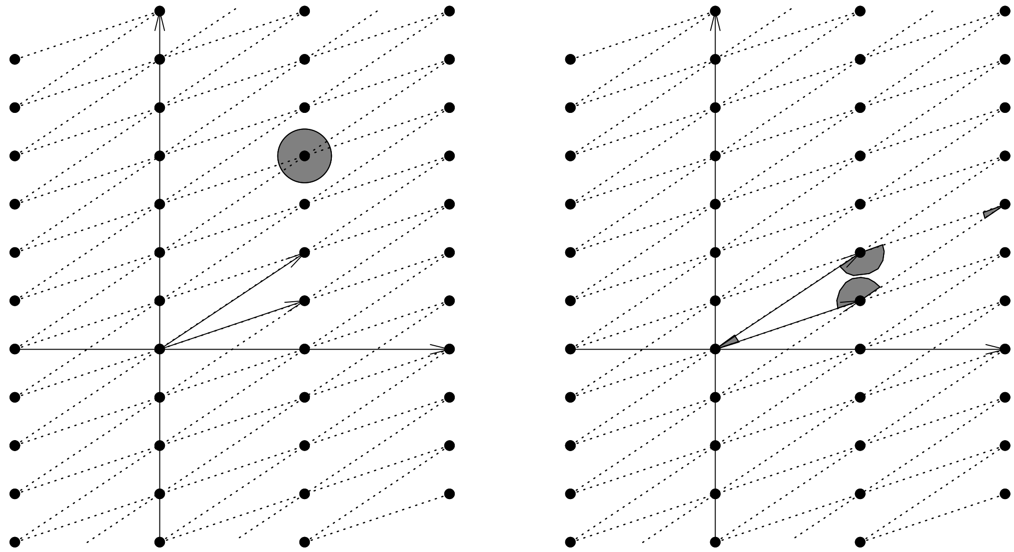

Figure 1: Sphere centered around a generic lattice point and sphere reduced modulo the lattice basis

However, not every basis of $\text{span}(B) = \{B\mathbf{x} \mid \mathbf{x} \in \mathbb{R}^n\}$ (as a vector space) is a basis of the lattice $\mathcal{L}(B)$. Every lattice $\mathcal{L}(B)$ induces an equivalence relation over $\text{span}(B)$ defined as follows: $\mathbf{v} \equiv \mathbf{w} \pmod{\mathcal{L}(B)}$ if and only if $\mathbf{v} - \mathbf{w} \in \mathcal{L}(B)$.

The *fundamental parallelepiped* spanned by the basis B is defined as the set $\mathcal{P}(B) = \{\sum_i x_i \mathbf{b}_i \mid 0 \le x_i < 1\}$. It is easy to see that for every basis B and every point $\mathbf{v} \in \text{span}(B)$, there exists a unique vector $\mathbf{w} \in \mathcal{P}(B)$ such that $\mathbf{v} \equiv \mathbf{w} \pmod{\mathcal{L}(B)}$. Vector $\mathbf{w}$ can be easily computed from $\mathbf{v}$ and B as follows. Let $\mathbf{x}$ be the solution to the linear system $B\mathbf{x} = \mathbf{v}$ (a solution always exists because $\mathbf{v} \in \text{span}(B)$ and is unique by linear independence of the vectors in B). Let $\mathbf{x}' = \mathbf{x} - \lfloor \mathbf{x} \rfloor$ be the vector obtained by replacing each coordinate of $\mathbf{x}$ by its fractional part. Then, it is easy to check that $\mathbf{w} = B\mathbf{x}' \in \mathcal{P}(B)$ and $\mathbf{v} \equiv \mathbf{w} \pmod{\mathcal{L}(B)}$. The unique element of $\mathcal{P}(B)$ congruent to $\mathbf{v}$ modulo $\mathcal{L}(B)$ is denoted $\mathbf{v} \bmod B$. Notice that although the equivalence relation $\mathbf{v} \equiv \mathbf{w} \pmod{\mathcal{L}(B)}$ does not depend on the particular choice of the basis B for the lattice $\mathcal{L}(B)$, the definition of the reduced vector ($\mathbf{v} \bmod B$) is basis dependent.

Pictorially, we can think the vector space $\text{span}(B)$ as partitioned into parallelepiped $\{\mathcal{P}(B) + \mathbf{w} \mid \mathbf{w} \in \mathcal{L}(B)\}$. Then, the reduced vector $\mathbf{v} \bmod B$ is the relative position of $\mathbf{v}$ in the parallelepiped $\mathcal{P}(B) + \mathbf{w}$ it belongs to. Notice that the reduced vector $\mathbf{v} \bmod B$ can be longer than the original vector $\mathbf{v}$, but it is never longer than the sum of the lengths of the basis vectors $\|\mathbf{b}_1\| + \cdots + \|\mathbf{b}_n\|$.

The technique presented in this paper easily adapts to any norm. However, for concreteness, we will concentrate on the Euclidean norm. The distance between two vectors $\mathbf{v}$ and $\mathbf{w}$ is defined by $\text{dist}(\mathbf{v}, \mathbf{w}) = \|\mathbf{v} - \mathbf{w}\| = \sqrt{\sum_i (v_i - w_i)^2}$. The distance function is extended to set of vectors as usual

$$\text{dist}(S_1, S_2) = \inf\{\|\mathbf{v} - \mathbf{w}\| : \mathbf{v} \in S_1, \mathbf{w} \in S_2\}.$$

In particular the distance of a vector from a lattice is given by $\text{dist}(\mathbf{v}, \mathcal{L}(B)) = \min\{\|\mathbf{v} - \mathbf{w}\| : \mathbf{w} \in \mathcal{L}(B)\}$. For every vector $\mathbf{v}$ and real $r \in \mathbb{R}$, we define the ball $\mathcal{B}(\mathbf{v}, r) = \{\mathbf{w} : \text{dist}(\mathbf{v}, \mathbf{w}) \le r\}$ of radius r centered in $\mathbf{w}$.

Two fundamental computational problems on lattices and the Shortest Vector Problem (SVP) and the Closest Vector Problem (CVP). In SVP, one is given a basis B and must find the shortest non-zero vector in $\mathcal{L}(B)$. In CVP, one is given a basis B and a target vector $\mathbf{v}$ (not necessarily in the lattice) and must find the lattice vector in $\mathcal{L}(B)$ closest to $\mathbf{v}$. Approximation versions of the above problems are easily defined. In the γ-approximate SVP one must find a non-zero lattice vector of length at most γ the shortest, and in the approximate CVP one must find a lattice vector at distance at most $\gamma \cdot \mathrm{dist}(\mathbf{v}, \mathcal{L}(B))$.

To date, the best polynomial time algorithms to approximate SVP and CVP achieve only a worst case approximation factor γ exponential in the dimension of the lattice [11, 4, 13]. On the other hand, CVP is NP-hard to approximate within a factor $\gamma = 2^{\ln^{1-\epsilon} n}$ [3, 5], and SVP is NP-hard (for randomized reductions) to approximate within any factor less than $\sqrt{2}$ [12]. The relation between the two problems has also been investigated, and in [9] it is proved that CVP is at least as hard as SVP. In general, finding good approximations to SVP and CVP seems to be computationally hard problems and have been used as the basis of various cryptographic protocols (e.g., [1, 2, 8]).

The approximation problems associated to the shortest vector problem and the closest vector problem are usually formalized in terms of the following promise problems [6].

Definition 1 (Approximate SVP) *The promise problem* $\mathtt{GapSVP}_\gamma$, *where* γ *(the gap function) is a function of the dimension, is defined as follows:*

- YES *instances are pairs* (B, d) *where* $B \in \mathbb{Z}^{n\times k}$, $d \in \mathbb{R}$ *and* $\|B\mathbf{z}\| \leq d$ *for some* $\mathbf{z} \in \mathbb{Z}^k \setminus \{\mathbf{0}\}$.
- NO *instances are pairs* (B, d) *where* $B \in \mathbb{Z}^{n\times k}$, $d \in \mathbb{R}$ *and* $\|B\mathbf{z}\| > \gamma d$ *for all* $\mathbf{z} \in \mathbb{Z}^n \setminus \{\mathbf{0}\}$.

Definition 2 (Approximate CVP) *The promise problem* $\mathtt{GapCVP}_\gamma$, *where* γ *(the gap function) is a function of the dimension, is defined as follows:*

- YES *instances are triples* $(B, \mathbf{y}, d)$ *where* $B \in \mathbb{Z}^{n\times k}$, $\mathbf{y} \in \mathbb{Z}^n$, $d \in \mathbb{R}$ *and* $\|B\mathbf{z} - \mathbf{y}\| \leq d$ *for some* $\mathbf{z} \in \mathbb{Z}^n$.
- NO *instances are triples* $(B, \mathbf{y}, d)$ *where* $B \in \mathbb{Z}^{n\times k}$, $\mathbf{y} \in \mathbb{Z}^n$, $d \in \mathbb{R}$ *and* $\|B\mathbf{z} - \mathbf{y}\| > \gamma d$ *for all* $\mathbf{z} \in \mathbb{Z}^k$.

2.2 Statistical Distance

Let X_0, X_1 be two random variables over the same set $\mathcal{X}$. The statistical distance between X_0 and X_1 is defined by

$$\Delta(X_0, X_1) = \max_{S \subseteq \mathcal{X}} |\Pr\{X_0 \in S\} - \Pr\{X_1 \in S\}|.$$

In the rest of the paper we will make extensive use of the following simple facts about the statistical distance.

- When $\mathcal{X}$ is countable, $\Delta(X_0, X_1) = \frac{1}{2}\sum_{x\in\mathcal{X}} |\Pr\{X_0 = x\} - \Pr\{X_1 = x\}|$.
- If each X_i is uniformly distributed over some set $\mathcal{X}_i \subseteq \mathcal{X}$, then $\Delta(X_0, X_1) = 1 - \frac{|\mathcal{X}_0 \cap \mathcal{X}_1|}{\max(|\mathcal{X}_0|, |\mathcal{X}_1|)}$.
- For any (possibly randomized) function f with domain $\mathcal{X}$, $\Delta(f(X_0), f(X_1)) \leq \Delta(X_0, X_1)$.

3 Interactive Protocols

In [6], Goldreich and Goldwasser describe interactive protocols to prove in zero-knowledge that a point is far from a lattice, or that the shortest vector in a lattice is long. More formally, they give honest verifier perfect zero-knowledge constant round one-sided error interactive proof systems for the complement of the promise problems $\mathtt{GapCVP}_\gamma$ and $\mathtt{GapSVP}_\gamma$ where $\gamma = \sqrt{\frac{n}{O(\ln n)}}$.

The protocols essentially work as follows. Let $(B, \mathbf{v})$ be an instance of $\mathtt{GapCVP}_\gamma$. The verifier select a lattice point r uniformly at random from a large subset of the lattice, a bit $\sigma \in \{0,1\}$ and an error vector η uniformly from a sphere of radius $\gamma d/2$. The vector $r + \eta + \sigma y$ is sent to the prover, who must guess the value of σ.

The protocol for $\mathtt{GapSVP}_\gamma$ is similar. A lattice point is chosen at random from a sufficiently large region of the lattice. A small error is added to it and the prover is left with the task of recovering the original lattice point.

In the next subsections we describe our modified protocols for CVP and SVP, obtained applying the technique outlined in the introduction to the original Goldreich-Goldwasser protocols.

3.1 Closest Vector Problem

Our modified interactive proof system for the closest vector problem works as follows. Let $(B, \mathbf{v}, d)$ be an instance of $\mathtt{GapCVP}_\gamma$, where $\gamma(n) = \sqrt{\frac{n}{O(\ln n)}}$.

1. The verifier uniformly selects a bit $\sigma \in \{0,1\}$ and an error vector η uniformly distributed in a sphere of radius $\gamma d/2$. The verifier sends $\mathbf{w} = (\eta + \sigma\mathbf{v}) \bmod B$ to the prover.
2. The prover responds with the value $\tau \in \{0,1\}$ such that $\mathrm{dist}(\tau\mathbf{v}, \mathbf{w} + \mathcal{L}(B))$ is minimized.
3. The verifier accepts if and only if $\tau = \sigma$.

Proposition 1 *When $\gamma = \sqrt{\frac{n}{O(\ln n)}}$, the above protocol is a honest-verifier zero-knowledge interactive proof system for the complement of* $\mathtt{GapCVP}_\gamma$, *with perfect completeness and soundness error bounded away from* 1.

In the rest of this subsection we prove the above Proposition.

Zero-Knowledge: The simulator simply executes the honest verifier protocol and return σ as the prover's answer.

Completeness: Assume $\mathrm{dist}(\mathbf{y}, \mathcal{L}(B)) > \gamma d$. We want to prove that

$$\mathrm{dist}(\sigma\mathbf{v}, \mathbf{w} + \mathcal{L}(B)) < \mathrm{dist}((1-\sigma)\mathbf{v}, \mathbf{w}) + \mathcal{L}(B).$$

Notice that $\mathbf{w} + \mathcal{L}(B) = (\eta + \sigma\mathbf{v}) \bmod B + \mathcal{L}(B) = \eta + \sigma\mathbf{v} + \mathcal{L}(B)$ and therefore

$$\mathrm{dist}(\sigma\mathbf{v}, \mathbf{w} + \mathcal{L}(B)) = \mathrm{dist}(\sigma\mathbf{v}, \sigma\mathbf{v} + \eta + \mathcal{L}(B)) \leq \|\eta\| \leq \gamma d/2.$$

On the other hand

$$\begin{aligned}
\mathrm{dist}((1-\sigma)\mathbf{v}, \mathbf{w} + \mathcal{L}(B)) &= \mathrm{dist}((1-\sigma)\mathbf{v}, \sigma\mathbf{v} + \eta + \mathcal{L}(B)) \\
&= \mathrm{dist}((1-2\sigma)\mathbf{v}, \eta + \mathcal{L}(B) \\
&\geq \mathrm{dist}(\pm v, \mathcal{L}(B)) - \|\eta\| \\
&> \gamma d - \frac{\gamma d}{2} = \frac{\gamma d}{2}.
\end{aligned}$$

Therefore the prover always output the correct value $\tau = \sigma$.

Soundness: Assume $\mathrm{dist}(\mathbf{v}, \mathcal{L}(B)) \leq d$ and let $\mathbf{x}$ be an integer vector such that $\|\mathbf{v} - B\mathbf{x}\| \leq d$. Let ξ_0 and ξ_1 be two random variables uniformly distributed on spheres $\mathcal{B}(0, \gamma d/2)$ and $\mathcal{B}(\mathbf{v} - B\mathbf{x}, \gamma d/2)$ respectively. Notice that when η is chosen uniformly at random, $((\eta + \sigma\mathbf{v}) \bmod B) = ((\eta + \sigma\mathbf{v} - B\mathbf{x}) \bmod B)$ is distributed identically to $(\xi_\sigma \bmod B)$ for $\sigma = 0, 1$. Therefore the protocol followed by the verifier is equivalent to choosing $\sigma \in \{0, 1\}$ at random and sending $(\xi_\sigma \bmod B)$ to the prover. We use the bound on the size of the relative intersection of two spheres from [6] to bound the statistical distance between ξ_0 and ξ_1:

$$\begin{aligned}\Delta(\xi_0, \xi_1) &\leq 1 - \frac{\mathrm{vol}(\mathcal{B}(0, \gamma d/2), \mathcal{B}(\mathbf{v} - B\mathbf{x}, \gamma d/2))}{\mathrm{vol}(\mathcal{B}(0, \gamma d/2))} \\ &\leq 1 - \frac{1}{\mathrm{poly}(n)}\end{aligned}$$

We can now bound the soundness error as follows:

$$\begin{aligned}\Pr\{P^*(\xi_\sigma \bmod B) = \sigma\} &= \frac{1}{2}(\Pr\{P^*(\xi_0 \bmod B) = 0\} + \Pr\{P^*(\xi_1 \bmod B) = 1\}) \\ &= \frac{1}{2}(1 + \Pr\{P^*(\xi_1 \bmod B) = 1\} - \Pr\{P^*(\xi_0 \bmod B) = 1\}) \\ &= \frac{1}{2}(1 + \Delta(\xi_0 \bmod B, \xi_1 \bmod B)) \\ &\leq \frac{1}{2}(1 + \Delta(\xi_0, \xi_1)) \\ &\leq \frac{1}{2}(2 - \frac{1}{\mathrm{poly}(n)}) \\ &= 1 - \frac{1}{\mathrm{poly}(n)}\end{aligned}$$

This concludes the proof of Proposition 1.

3.2 Shortest Vector Problem

The proof system for the Shortest Vector Problem uses a similar idea, but the reduction is made modulo $2B$, the basis of $2\mathcal{L}(B)$ obtained by doubling each vector in B. Let (B, d) an instance of $\mathtt{GapSVP}_\gamma$, where $\gamma = \sqrt{\frac{n}{O(\ln n)}}$.

1. The verifier uniformly selects a bit string $\mathbf{s} \in \{0, 1\}^n$ and an error vector η uniformly distributed in a sphere of radius $\gamma d/2$. The verifier sends $\mathbf{w} = (\eta + B\mathbf{s}) \bmod (2B)$ to the prover.

2. The prover finds $\mathbf{t} \in \{0, 1\}^n$ such that $\mathrm{dist}(B\mathbf{t}, \mathbf{w} + \mathcal{L}(2B))$ is minimized and sends it to the verifier.

3. The verifier accepts if and only if $\mathbf{t} = \mathbf{s}$.

Proposition 2 *The above protocol is a honest-verifier zero-knowledge interactive proof system for the complement of* $\mathtt{GapSVP}_\gamma$*, with perfect completeness and soundness error bounded away from* 1.

In the rest of this subsection we prove the above Proposition.

Zero-Knowledge: The simulator simply executes the honest verifier protocol and return $\mathbf{s}$ as the prover's answer.

Completeness: Assume $\mathrm{dist}(\mathbf{y}, \mathcal{L}(B)) > \gamma d$. We want to prove that $\mathrm{dist}(B\mathbf{s}, \mathbf{w} + \mathcal{L}(2B)) < \mathrm{dist}(B\mathbf{t}, \mathbf{w} + \mathcal{L}(2B))$ for any $\mathbf{t} \neq \mathbf{s}$. First of all notice that

$$\mathrm{dist}(B\mathbf{s}, \mathbf{w} + \mathcal{L}(2B)) = \mathrm{dist}(B\mathbf{s}, \eta + B\mathbf{s} + \mathcal{L}(2B)) \leq \|\eta\| \leq \gamma d/2.$$

We now prove that $\mathrm{dist}(B\mathbf{t}, \mathbf{w} + \mathcal{L}(2B) > \gamma d/2$ for all $\mathbf{t} \neq \mathbf{s}$. Notice that

$$\begin{aligned} \mathrm{dist}(B\mathbf{t}, \mathbf{w} + \mathcal{L}(2B)) &= \mathrm{dist}(B\mathbf{t}, \eta + B\mathbf{s} + \mathcal{L}(2B)) \\ &\geq \mathrm{dist}(B(\mathbf{t} - \mathbf{s}), \mathcal{L}(2B)) - \|\eta\| \\ &> \gamma d - \frac{\gamma d}{2} = \frac{\gamma d}{2} \end{aligned}$$

because for any vector $\mathbf{v} \in \mathcal{L}(2B)$ and for any $\mathbf{t} \in \{0,1\}^n \setminus \{\mathbf{s}\}$, $B(\mathbf{t} - \mathbf{s}) - \mathbf{v}$ is a non-zero vector in $\mathcal{L}(B)$, and therefore $\mathrm{dist}(B(\mathbf{t} - \mathbf{s}), \mathcal{L}(2B)) > \gamma d$ (here we are using the fact that $\mathbf{s} \neq \mathbf{t}$ and $\mathbf{s} - \mathbf{t}$ must have some odd component). This proves the prover always outputs the correct value $\mathbf{t} = \mathbf{s}$.

Soundness: Let P^* be an arbitrary prover and let p be the probability of success $p = \Pr\{P^*((\eta + B\mathbf{s}) \bmod 2B) = \mathbf{s}\}$ (probability computed with respect to the choice of $\mathbf{s} \in \{0,1\}^n$ and $\eta \in \mathcal{B}(0, \gamma d/2)$). Consider the following mental experiment. Let $B\mathbf{x}$ be a shortest non-zero vector in $\mathcal{L}(B)$ and assume $\|B\mathbf{x}\| \leq d$. Notice that $\mathbf{x} \neq \mathbf{0} \pmod 2$ because otherwise $B(\mathbf{x}/2)$ is a shorter non-zero vector in $\mathcal{L}(B)$. Choose $\mathbf{s} \in \{0,1\}^n$ at random and let $\mathbf{s}_\sigma = (\mathbf{s}_0 + \sigma\mathbf{x}) \bmod 2$ for $\sigma = 0, 1$. Choose $\sigma \in \{0,1\}$ at random and sends $B\mathbf{s}_\sigma \bmod (2B)$ to the prover. Since $\mathbf{s}_0$ and $\mathbf{s}_1$ are uniformly distributed on 2^n, this is equivalent to the protocol followed by the honest verifier. It follows by a simple averaging argument that there exists an $\mathbf{s}$ such that the prover succeed with probability

$$p_{\mathbf{s}} = \Pr\{P^*((\eta + B\mathbf{s}_\sigma) \bmod 2B) = \mathbf{s}_\sigma\} \geq p.$$

We now prove that $p_{\mathbf{s}}$ is bounded away from 1. Let ξ_0 and ξ_1 be two random variables uniformly distributed on spheres $\mathcal{B}(\mathbf{s}, \gamma d/2)$ and $\mathcal{B}(B(\mathbf{s} + x), \gamma d/2)$ respectively. Notice that $(\eta + B(\mathbf{s} + \sigma\mathbf{x})) \bmod (2B) = (\eta + B\mathbf{s}_\sigma) \bmod (2B)$ is distributed identically to $\xi_\sigma \bmod 2B$ for $\sigma = 0, 1$. As for the CVP proof system, the statistical distance between ξ_0 and ξ_1 is at most

$$\begin{aligned} \Delta(\xi_0, \xi_1) &\leq 1 - \frac{\mathrm{vol}(\mathcal{B}(B\mathbf{s}, \gamma d/2), \mathcal{B}(B(\mathbf{s} + \mathbf{x}), \gamma d/2))}{\mathrm{vol}(\mathcal{B}(B\mathbf{s}, \gamma d/2))} \\ &\leq 1 - \frac{1}{\mathrm{poly}(n)} \end{aligned}$$

Let P^* a prover that tries to guess the value of σ. We can bound the soundness error as follows:

$$\begin{aligned} \Pr\{P^*(\xi_\sigma \bmod B) = \mathbf{s}_\sigma\} &= \frac{1}{2}(\Pr\{P^*(\xi_0 \bmod B) = \mathbf{s}_0\} + \Pr\{P^*(\xi_1 \bmod B) = \mathbf{s}_1\}) \\ &\leq \frac{1}{2}(1 + \Delta(\xi_0 \bmod B, \xi_1 \bmod B)) \\ &\leq \frac{1}{2}(1 + \Delta(\xi_0, \xi_1)) \\ &\leq \frac{1}{2}(2 - \frac{1}{\mathrm{poly}(n)}) \\ &= 1 - \frac{1}{\mathrm{poly}(n)}. \end{aligned}$$

This prover that $p_\mathbf{s}$ is bounded away from 1, and therefore also the soundness error $p \leq p_\mathbf{s}$ is bounded away from 1.

This concludes the proof of Proposition 2.

Interestingly, the proof systems we just described for SVP and CVP are reminiscent of the connection between the two problems discovered in [9]. In that work, an SVP instance B is reduced to a CVP problem by removing some basis vector $\mathbf{b}_i$ from the lattice $\mathcal{L}(B)$ by doubling the corresponding basis element, and then looking for a lattice vector (in the doubled sub-lattice) closest to $\mathbf{b}_i$. Our protocols for SVP, doubles the basis vectors $2B$ removing all vectors in B from the lattice and then executes a protocols which can be thought as a multidimensional extension of the CVP protocol with lattice $2B$ and "targets" B. In a certain sense, the SVP protocol corresponds to first reducing SVP to CVP as in [9] and then running the interactive protocol for CVP.

4 The GGH encryption scheme

In [8] Goldreich, Goldwasser and Halevi propose a trapdoor permutation based on the hardness of the closest vector problem, and use it to construct encryption schemes. The trapdoor function and the corresponding trapdoor are described by two bases B, R of the same lattice $\mathcal{L}(B) = \mathcal{L}(R)$ (called the public and private basis respectively). The private key R is a particularly good basis that allows to solve the Closest Vector Problem in the lattice, when the distance of the target point from the lattice is sufficiently small. The function takes in input an integer vector $\mathbf{v}$ and a small error vector $\mathbf{r}$, and returns $B\mathbf{v} + \mathbf{r}$, i.e. the lattice vector with public coefficients $\mathbf{v}$ perturbed by $\mathbf{r}$. The error vector $\mathbf{r}$ must be sufficiently small to allow to recover $B\mathbf{v}$ using the private basis R. Once $B\mathbf{v}$ is recovered, one can easily compute $\mathbf{v}$ and $\mathbf{r}$ using simple linear algebra, therefore inverting the trapdoor function.

The above function is used to build two public key encryption schemes, depending on how the message is embedded into the input $(\mathbf{v}, \mathbf{r})$. In particular, one can either encode the message in the error vector $\mathbf{r}$, and choose $\mathbf{v}$ completely at random from a sufficiently large cube, or alternatively, choose $\mathbf{r}$ completely at random (from a sufficiently small sphere) and encode the message bits as the lowest order bit of the entries in $\mathbf{v}$.

In both cases the vector $\mathbf{v}$ must be chosen from a sufficiently large cube. The exact effect of the size of the cube on the security of the system is not clear, so for efficiency reasons [8] sets the size of the cube to a relatively small value (polynomial in n) which seems in practice sufficient to withstand known attacks. Using our technique, we can achieve the same effect of using an arbitrary large cube, and make the scheme more efficient at the same time.

The specific way we apply our technique to the trapdoor function depends on the encryption method we want to use, and is described in the next two subsections.

4.1 Embedding the message in the error vector

If the message is encoded in the error vector $\mathbf{r}$, then we don't need to consider $\mathbf{v}$ at all: we can just take the error vector $\mathbf{r}$ and output

$$E(\mathbf{m}) = \mathbf{r} \bmod B.$$

The reader can easily check that the same decryption procedure of the original scheme still works. Moreover the new scheme is at least as secure as the original one. That is, given a decryption oracle for our scheme, we can easily decrypt the original GGH cryptosystem as follows: let $\mathbf{v}$ be the ciphertext of the GGH cryptosystem. Compute $\mathbf{w} = \mathbf{v} \bmod B$ and call the decryption oracle for our scheme. It is easy to verify that the ciphertext is decrypted correctly if and only if the oracle returns the right answer. Therefore the attack to the GGH encryption function will succeed with the same probability as the original attack.

4.2 Embedding the message in the lattice vector

If we want to embed the message in the coefficient vector $\mathbf{v}$ we proceed as follows. Let $\mathbf{v}$ be the message itself. Compute $B\mathbf{v}$ and add a random error $\mathbf{r}$ (chosen as in the original protocol). Finally, reduce the result modulo $2B$:

$$E(\mathbf{m}) = (B\mathbf{m} + \mathbf{r}) \bmod 2B$$

Again, the same decryption algorithm will work, and the modified scheme is at least as secure as the original one. The proof is essentially the same as in the previous cryptosystem.

4.3 On the choice of the public basis

In the GGH cryptosystem the public basis B is obtained from the private basis by applying a random unimodular transformation (or alternatively, performing a sufficiently long sequence of elementary column operations). This results in a public basis B much bigger than the private basis. As a consequence, the public basis is fairly large even for moderate sizes of the parameters. We suggest a modification to the public key generation process analogous to the encryption function. Instead of choosing some "random" basis generating the same lattice as R, we always output some standard basis that depends only on the lattice generated by R (and not on the specific private basis we started from). A natural choice is to let B be the Hermite Normal Form (HNF) of B. Matrix B is the HNF of R[1] if

1. they generate the same lattice
2. B is upper triangular, i.e., $b_{i,j} = 0$ for all $i > j$
3. For all $i < j$, $0 \leq b_{i,j} < b_{i,i}$.

The HNF is unique and can be computed in polynomial time from any basis of the lattice (e.g., using the algorithm in [10]). Since the HNF of R can be computed in polynomial time from any other public basis B' generating the same lattice as R, choosing $B = HNF(R)$ as a public basis is the best possible choice from the security point of view: one can easily prove that any attack to the modified scheme using $B = HNF(R)$ as a public basis easily translates to an attack (with at least the same success probability) to the original scheme where B is chosen at random applying an arbitrarily long sequence of elementary column operations.

The triangular form of B also makes the encryption algorithm (i.e., the reduction modulo B or $2B$) extremely simple. Given $\mathbf{r}$, the reduced vector $\mathbf{r} \bmod B$ can be easily determined as follow. Compute the integer vector $\mathbf{x}$ one coordinate at a time (starting from x_n) using the formula

$$x_i = \left\lfloor \frac{r_i - \sum_{j>i} b_{i,j} x_j}{b_{i,i}} \right\rfloor .$$

The output of the encryption algorithm is $\mathbf{y} = \mathbf{r} - B\mathbf{x} \equiv \mathbf{r} \bmod B$. The reader can easily check that for every i, $0 \leq y_i < b_{i,i}$, i.e., the result is the unique point in the parallelepiped $\{\mathbf{w} \mid 0 \leq w_i < b_{i,i}\}$ which is congruent to $\mathbf{r}$ modulo $\mathcal{L}(B)$. Notice that this is slightly different, but equivalent, to the reduction operation modulo the basis described in the introduction.

[1] We are assuming R generate a full rank lattice.

dimension	Basis Size GGH	Basis Size New scheme	Ciphertext GGH	Ciphertext New scheme
200	250 KB	32 KB	2 KB	160 B
250	500 KB	50 KB	3 KB	200 B
300	750 KB	75 KB	4 KB	250 B
350	1250 KB	100 KB	5 KB	300 B
400	1850 KB	140 KB	6 KB	350 B

Figure 2: Comparison of the key and ciphertext sizes in the GGH scheme and the modified scheme. Sizes in kilobytes (KB) and bytes (B).

We now analyze the size of the public key and the ciphertext of the new encryption algorithm. First of all notice that the product of the elements on the diagonal of B equals the determinant of the lattice $\det(B) = \det(R)$. Therefore we can bound the bit-size of the ciphertexts and the basis vectors by

$$\sum \lg b_{i,i} = \lg \prod b_{i,i} = \lg \det(R).$$

A $n \lg \det(R)$ bound on the bit-size of the public basis immediately follows. The saving with respect to the original GGH encryption algorithm can be substantial. Estimates of the key and ciphertext sizes for the GGH and the modified scheme are shown in Figure 2. The estimates are based on the GGH challenges published at [7]. One can easily see that the modified scheme results in keys and ciphertexts more than an order of magnitude smaller than the original scheme. We remark that the sizes relative to the modified scheme are only upper bounds obtained using the Hadamard inequality to estimate the determinant of the lattice, and the actual sizes of the keys and ciphertexts of the modified cryptosystem can be even smaller than shown in the table.

5 Discussion

We presented a general technique that can be used to simplify and improve various cryptographic protocols based on the hardness of lattice problems. The improvement can be quite significant in practice, as demonstrated for the GGH encryption scheme. Moreover, the modified protocols perform better than the original ones from essentially all points of view: they are faster, more secure, require less storage, use less bandwidth and need less random bits. Finally, they are also simpler than the original protocols. This is clearly a significant advantage both in practice and in theory, because the simplified protocols are easier to implement, and their security can be better understood and analyzed.

References

[1] Miklós Ajtai. Generating hard instances of lattice problems (extended abstract). In *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing*, pages 99–108, Philadelphia, Pennsylvania, 22–24 May 1996.

[2] Miklós Ajtai and Cynthia Dwork. A public-key cryptosystem with worst-case/average-case equivalence. In *Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing*, pages 284–293, El Paso, Texas, 4–6 May 1997.

[3] Sanjeev Arora, László Babai, Jacques Stern, and Z. Sweedyk. The hardness of approximate optima in lattices, codes, and systems of linear equations. *Journal of Computer and System Sciences*, 54(2):317–331, April 1997.

[4] László Babai. On Lovasz' lattice reduction and the nearest lattice point problem. *Combinatorica*, 6, 1986.

[5] Irit Dinur, Guy Kindler, and Shmuel Safra. Approximating CVP to within almost-polynomial factors is NP-hard. In *39th Annual Symposium on Foundations of Computer Science*, Palo Alto, California, 7–10 November 1998. IEEE.

[6] Oded Goldreich and Shafi Goldwasser. On the limits of non-approximability of lattice problems. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pages 1–9. Dallas, Texas, 23–26 May 1998.

[7] Oded Goldreich, Shafi Goldwasser, and Shai Halevi. The GGH cryptosystem, challange page. `http://theory.lcs.mit.edu/~cis/lattice/challenge.html`.

[8] Oded Goldreich, Shafi Goldwasser, and Shai Halevi. Public-key cryptosystems from lattice reduction problems. In Burton S. Kaliski Jr., editor, *Advances in Cryptology—CRYPTO '97*, volume 1294 of *Lecture Notes in Computer Science*, pages 112–131. Springer-Verlag, 17–21 August 1997.

[9] Oded Goldreich, Daniele Micciancio, Shmuel Safra, and Jean-Pierre Seifert. Approximating shortest lattice vectors is not harder than approximating closest lattice vectors. In *Electronic Colloquium on Computational Complexity, technical reports*. ECCC, 1999.

[10] Ravi Kannan and Achim Bachem. Polynomial algorithms for computing the Smith and Hermite normal forms of an integer matrix. *SIAM Journal on Computing*, 8(4):499–507, November 1979.

[11] Arjen K. Lenstra, Hendrik W. Lenstra, Jr., and László Lovász. Factoring polynomials with rational coefficients. *Mathematische Ann.*, 261:513–534, 1982.

[12] Daniele Micciancio. The shortest vector problem is NP-hard to approximate to within some constant. In *39th Annual Symposium on Foundations of Computer Science*, Palo Alto, California, 7–10 November 1998. IEEE.

[13] Claus P. Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. *Theoretical Computer Science*, 53(2–3):201–224, 1987.