Jin and Xue *Cybersecurity (2023)6:38*
https://doi.org/10.1186/s42400-023-00173-w

Cybersecurity

RESEARCH

Open Access

Improved lower bound for the complexity of unique shortest vector problem

Baolong Jin[1,2] and Rui Xue[1,2*]

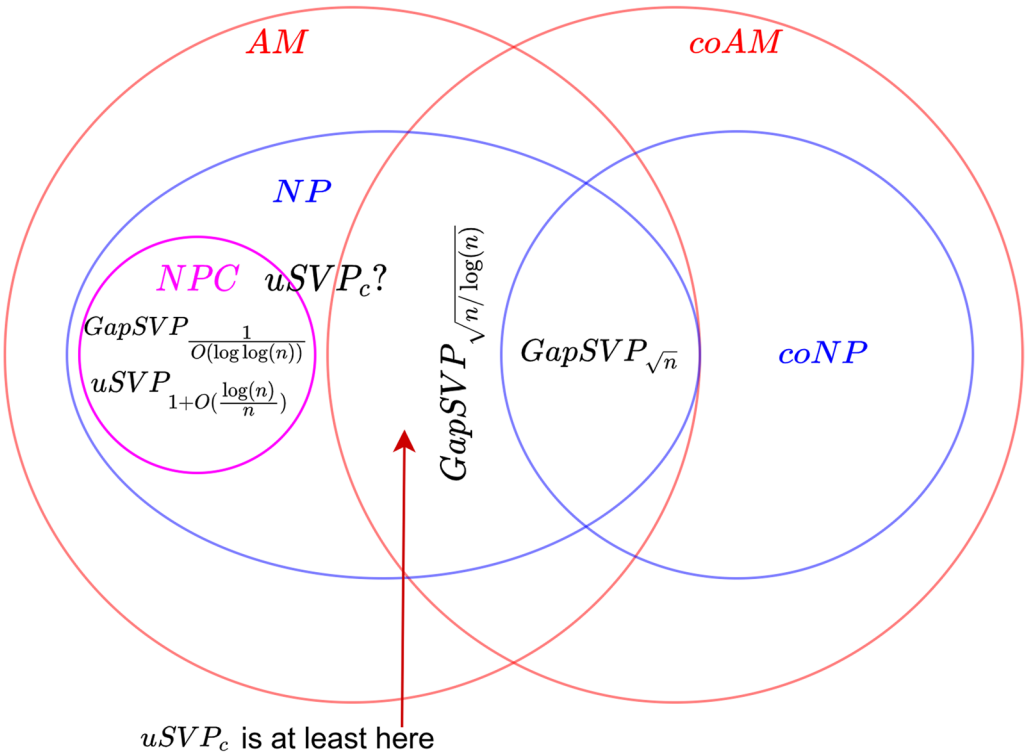
Abstract

Unique shortest vector problem (uSVP) plays an important role in lattice based cryptography. Many cryptographic schemes based their security on it. For the cofidence of those applications, it is essential to clarify the complexity of uSVP with different parameters. However, proving the NP-hardness of uSVP appears quite hard. To the state of the art, we are even not able to prove the NP-hardness of uSVP with constant parameters. In this work, we gave a lower bound for the hardness of uSVP with constant parameters, i.e. we proved that uSVP is at least as hard as gap shortest vector problem (GapSVP) with gap of $O(\sqrt{n/\log(n)})$, which is in $NP \cap coAM$. Unlike previous works, our reduction works for paramters in a bigger range, especially when the constant hidden by the big-O in GapSVP is smaller than 1.

Keywords Computational complexity, Unique shortest vector problem, Bounded distance decoding, Complexity reduction

*Correspondence:
Rui Xue
xuerui@iie.ac.cn
Full list of author information is available at the end of the article

Graphical abstract



Introduction

The Shortest Vector Problem (SVP) is one of the most important problems in lattice theory. From the perspective of complexity theory, it's very important to figure out the precise complexity of SVP with different parameters. The NP-hardness of SVP in l_2 norm was conjectured in 1981 by Boas (1981). However it remains to be an open problem for quite a long period. A breakthrough came up at Ajtai (1998), which proved that SVP in l_2 norm is NP-hard under randomized reductions. Actually, in Ajtai's work, he proved that approximate SVP in l_2 norm to within a factor of $1+2^{-n^{\varepsilon}}$ is NP-hard. This result answered the long standing question of the NP-hardness of SVP in l_2 norm, moreover it showed the possibility that approximating SVP to some factors beyond the fraction of exponential in n is still NP-hard, which turned out to be true. The NP-hardness result of SVP was later improved by Micciancio in Micciancio (1998) to within a constant approximation factor under a number theoretic assumption. Moreover, Micciancio's proof works for any l_p norms for approximation factor $\sqrt[p]{2}$. In Khot (2003), Khot improved the NP-hardness of SVP to approximation factor $p^{1-\varepsilon}$, which is stronger than Micciancio's result (Micciancio 1998). However this reduction only work for $p \geq p(\varepsilon)$ norms, especially, it don't apply to l_2 norm. Soon after (Khot 2003), Khot proposed another proof (Khot 2004) for the NP-hardness of approximating SVP, which stated that approximating SVP to within any constant factor is NP-hard assuming that $NP \nsubseteq RP$. Further, assuming $NP \nsubseteq RTIME(2^{poly(\log(n))})$ there is no polynomial-time algorithm approximates SVP to within factor of $2^{\log^{\frac{1}{2}-\varepsilon}(n)}$, which is almost polynomial in n. This result is way more stronger than Micciancio (1998) and Khot (2003). Later, Micciancio (2012) proposed another proof for the NP-hardness of approximating SVP. Micciancio (2012) used the same technique as the one used by Khot (2004), which is called the BCH code, in a different manner. He had also proved the NP-hardness of SVP for any constant approximation factor, moreover he proved that approximating SVP for subpolynomial factors $n^{\frac{1}{O(\log\log n)}}$ is NP-hard assuming that NP is not contained by subexponential time. The reduction in Micciancio (2012) contains significantly less probabilistic parts compared to the proof in Khot (2004), and it is potentially easier to be derandomized since the only random parts are the choosing of a vector and the famous Sauer's lemma due to Sauer (1972), Vapnik and Chervonenkis (2015) and Shelah (1972).

The so called Lattice based cryptography was inovated by Ajtai (1996), in which Ajtai constructed an average hard lattice problem called Short Integer Solution problem (SIS), and it is widely used in all kinds of lattice base cryptographic schemes. Follwoing Ajtai's work (Ajtai 1996), researches proposed various cryptographic schemes. The first one was due to Ajtai and Dwork (1997), their one-way function was based on the hardness of n^8-uSVP, the uniqueness factor of which is quite large. Following (Ajtai and Dwork 1997), many improved results (Cai and Cusick 1999; Micciancio 2004; Regev 2003) were proposed, and the security assumption was improved to $n^{1.5}$-uSVP. Apparently, a cryptosystem based on weaker assumption is way more attractive than those based on strong assumptions. As for uSVP based cryptosystems, we want to build them upon small uniqueness factor since $O(n)$-uSVP tends to be a lot easier than the corresponding $\text{GapSVP}_{O(n)}$, and both are far away from being NP-hard.

In the perspective of complexity theory, we want to build NP-hardness results for uSVP similar with what was done for SVP. However, things turned out to be extremly difficult, up to now, we don't even know whether uSVP is NP-hard for constant uniqueness factor. The first result proving the NP-hardness of uSVP was proposed by Kumar and Sivakumar (2001), without any guarantee for the uniqueness factor. Aggarwal and Dubey (2016) proposed a deterministic reduction from SVP to $O(1+2^{-O(n^2)})$-uSVP and a randomized reduction from SVP to $1+\frac{1}{poly(n)}$-uSVP. Aggarwal's reduction from SVP to $1+\frac{1}{poly(n)}$-uSVP used the same technique used by Kumar and Sivakumar (2001) to make the shortest vector unique. At the same time, another work (Stephens-Davidowitz 2016) gave a randomized polynomial-time reduction from SVP to $(1+O(\log(n)/n))$-uSVP, which showed us some hope for proving the NP-hardness of uSVP for bigger uniqueness factor. As for the l_∞ norm, Khoat and Tan (2008) gave a reduction from Knapsak Optimization problem to uSVP. On the other side, Cai (1998) proved that $n^{\frac{1}{4}}$-uSVP cannot be NP-hard, unless the polynomial hierarchy collapses. Lyubashevsky and Micciancio (2009) investigated the relation between the lattice problems GapSVP, BDD (Bounded Distance Decoding) and uSVP. Their results states that $\frac{1}{2\gamma}$-BDD reduces to γ-uSVP, γ-uSVP reduces to $\frac{1}{\gamma}$-BDD, and GapSVP_γ reduces to $\frac{1}{\gamma}\sqrt{n/\log(n)}$-BDD. The last reduction holds for any $\gamma > 2\sqrt{n/\log(n)}$. Combine them we have that GapSVP_γ reduces to $\frac{\gamma}{2}\sqrt{\log(n)/n}$-uSVP for any $\gamma > 2\sqrt{n/\log(n)}$. That is, $\text{GapSVP}_{c'\sqrt{n/\log(n)}}$ reduces to $\frac{c'}{2}$-uSVP, which states that uSVP with constant uniqueness factor is at least as hard as $\text{GapSVP}_{O(\sqrt{n/\log(n)})}$. Note that this result holds only for $c' > 2$.

Our contribution

The NP-hardness of uSVP with constant uniqueness factor still remains open. And it seems hard to establish reduction from NP-hard SVP instances to such uSVP instances. Instead of proving the NP-hardness of uSVP with constant uniqueness factor, we proved a result which is similar with the one obtained by combining results of Lyubashevsky and Micciancio (2009). We reduced $\text{GapSVP}_{c'\sqrt{n/\log(n)}}$ to $\frac{c'}{3\sqrt{2}c}$-uSVP for almost any constant c and $c' > 3\sqrt{2}c$. Compared to Lyubashevsky and Micciancio (2009), our reduction holds for any c', especially when $c' < 1$ is a small constant. Moreover, since the uniqueness factor of uSVP instance depends on the fraction of c' and c instead of only c', our result is way more flexible in the choice of uniqueness factor. Combine our reduction with the reduction from γ-uSVP to $\frac{1}{\gamma}$-BDD in Lyubashevsky and Micciancio (2009), we have that $\text{GapSVP}_{c'\sqrt{n/\log(n)}}$ reduces to $\frac{3\sqrt{2}c}{c'}$-BDD. Notice that, in the sence of parameters of BDD, this result from GapSVP to BDD is the same with that in Lyubashevsky and Micciancio (2009). However, due to the flexibility of the choice for c, c', the constant for GapSVP can be as small as you like, which gave a stronger guarantee for the hardness of BDD.

As an application of our result, one can directly convert an algorithm for uSVP with arbitrary constant uniqueness factor into an algorithm for GapSVP with parameter $o(\sqrt{n/\log(n)})$. According to the results of Liu et al. (2011); Wei et al. (2015), some lattice reduction or enumeration algorithms enjoy a better time and space complexity. In our reduction the constant hidden by $o(\sqrt{n/\log(n)})$ is almost irrelevant the uniqueness factor of uSVP. Hence we have the result that $\text{GapSVP}_{o(\sqrt{n/\log(n)})}$ is solvable within time $2^{0.8306n+o(n)}$.

Technique and limitation

The reduction used to establish our result is essentially the same one used by Lyubashevsky (2008) which was inspired by Peikert (2009). In order to solve GapSVP instance with the help of uSVP oracle, the reduction procedure construct a new basis from the input GapSVP instance. When the input is a NO instance, the uSVP oracle must answer the unique shortest vector generated by the reduction procedure. Meanwhile, if the input is a YES instance, the uSVP oracle won't be able to distinguish between the vector generated by the procedure and it's difference with some

vectors in the lattice spanned by the original basis of GapSVP instance. Actually this indistinguishability holds for any full power oracle, which is quite strong. One may want to use the same technique to prove similar results for uSVP with uniqueness factors beyond constant, which would be quite attractive. Unfortunately, this won't work due to the basic rules of high dimensional balls. The same situation arises when one tries to decrease c' to beyond constant. It should be emphasized that our reduction dosn't hold for of GapSVP$_{O(\sqrt{n/\log^k n})}$ for constant $k > 1$, which is a little closer to the NP-hardness bound $n^{\frac{1}{O(\log\log(n))}}$ proved by Micciancio (2012).

Roadmap

In "Preliminary" Section, we provided some basic knowledge about lattice. The main reduction is proved in "Hardness of uSVP" Section, it can be read alone since "Intersection of high dimensional balls" Section only provided a fact of high dimensional balls supporting the parameter settings in our reduction. Readers familiar with lattices and high dimensional balls can safely skip "Preliminary" and "Intersection of high dimensional balls" Sections.

Preliminary

Through out this paper, we use lowercase letters to denote numbers, variables and matrices, which can be told easily according to their contexts. Especially, e is used as the base of natural logrithm. We use $\log(a)$ to denote the logrithm of a with base 2. For a vector $v = (v_1, \cdots, v_n)$, we use $\|v\| = \sqrt{\sum_{i=1}^n v_i^2}$ to denote its Euclidean norm, which is usually called the length of v. Given a, b, with $a = (a_1, \ldots, a_n)$ being column vector, $b = (b_1, \ldots)$ being vector or number, we use (a, b) to denote the concatenation of a and b, i.e. $(a, b) = (a_1, \ldots, a_n, b_1, \ldots)$.

A lattice is the group generated by the integral combination of a finite subset of $\mathbb{R}^n$. Given a set of vectors $B = [b_1, \cdots, b_m] \in \mathbb{R}^{n\times m}$, the lattice generated by B is the group

$$\mathcal{L}(B) = \left\{ \sum_{i=1}^{m} z_i b_i | z_i \in \mathbb{Z} \right\}.$$

Take a vector $t \in \mathbb{R}^n$, we define the distance from t to lattice $\mathcal{L}(B)$ to be

$$dist(t, \mathcal{L}(B)) = \min_{v\in\mathcal{L}(B)} \{\|t - v\|\}.$$

For every lattice $\mathcal{L}(B)$, there is a very important sequence of constants $\{\lambda_i(\mathcal{L}(B))\}_{i\in[1,m]}$, which are called the successive minimums. They are defined as follows.

$$\lambda_i(\mathcal{L}(B)) = \inf\{r | dim(span(\mathcal{L}(B) \cap \mathcal{B}^n(r))) \geq i\},$$

where $\mathcal{B}^n(r) = \{v | v \in \mathbb{R}^n, \|v\| \leq r\}$ is the n-dimensional closed ball of radius r centered at 0 with respect to Euclidean norm. For simplicity, we use $\lambda_i(B)$ to denote $\lambda_i(\mathcal{L}(B))$. The most studied one of them is $\lambda_1(\mathcal{L}(B))$, which is usually denoted by $\lambda(B)$.

Definition 1 (*Bounded Distance Decoding Problem* (BDD_α)) Given basis B and a vector t with the promise that $dist(t, \mathcal{L}(B) < \alpha\lambda(B)$, the Bounded Distance Decoding problem is a promised search problem which asks for the vector $v \in \mathcal{L}(B)$ closest to t.

Definition 2 (*Shortest Vector Problem* (*SVP*)) Given basis B, the Shortest Vector problem is a search problem which asks for a vector $v \in \mathcal{L}(B)$ with length $\|v\| = \lambda(B)$.

Definition 3 (*Approximate Shortest Vector Problem* (SVP_γ)) For any real γ, given basis B, the Approximate Shortest Vector problem is a search problem which asks for a vector $v \in \mathcal{L}(B)$ with length $\|v\| \leq \gamma\lambda(B)$.

Definition 4 (*Gap Shortest Vector Problem* ($GapSVP_\gamma$)) For any real $\gamma \geq 1, d$, given basis B, the Gap Shortest Vector problem is a decisional problem which asks to tell the following

- (B, d) is a YES instance if $\lambda(B) \leq d$
- (B, d) is a NO instance if $\lambda(B) > \gamma d$.

Definition 5 (*Unique Shortest Vector Problem* ($\gamma - uSVP$)) For any real $\gamma \geq 1$, given basis B, the Unique Shortest Vector problem is a promised search problem with the promise that $\lambda_2(B) > \gamma\lambda_1(B)$, which asks for the unique vector $v \in \mathcal{L}(B)$ with length $\|v\| = \lambda(B)$.

Balls in n-dimension are defined as the set

$$\mathcal{B}^n(x, r) = \{v | v, x \in \mathbb{R}^n, \|v - x\| \leq r\},$$

where $x \in \mathbb{R}^n$ is the center of the ball, and $r \in \mathbb{R}$ is its radius. If the center of a ball is 0, we simplely write it as $\mathcal{B}^n(r) = \mathcal{B}^n(0, r)$. If the radius of a ball is 1, we simplely write it as $\mathcal{B}^n(x) = \mathcal{B}^n(x, 1)$. Especially, the ball centered at 0 with radius 1 is denoted by $\mathcal{B}^n$.

The (complete) gamma function is defined as $\Gamma(n) = (n-1)!$. Although there are much more interesting facts about the gamma function, knowing its basic definition is enough for our usage. Actually, for our reduction, it's not neccessory to know any detail about the gamma function.

Intersection of high dimensional balls

We are going to show some facts about high dimensional spheres in this section. First of all, using the famous notion of gamma function, the volume of n-dimensional unit ball can be write as

$$V(\mathcal{B}^n) = \frac{\pi^{n/2}}{\Gamma(n/2+1)}.$$

Hence the volume of n-dimensional ball with radius r is

$$V(\mathcal{B}^n(r)) = \frac{\pi^{n/2} r^n}{\Gamma(n/2+1)}.$$

Actually, for the purpose of supporting our proof, it's enough to show the relation between $V(\mathcal{B}^{n-1})$ and $V(\mathcal{B}^n)$. This can be done by integral of the volume of $(n-1)$ -dimensional ball. Formally, we have

$$\begin{aligned} V(\mathcal{B}^n) &= \int_{-1}^{1} V\left(\mathcal{B}^{n-1}(\sqrt{1-x^2})\right) dx \\ &= 2V(\mathcal{B}^{n-1}) \int_0^1 (1-x^2)^{(n-1)/2} dx. \end{aligned} \tag{1}$$

Now let's focus on the intersection of balls. Since we are dealing with lattice problems, there is a sibling for every lattice point v in the same lattice. If an unit ball centered at v intersects $\mathcal{B}^n$, there is another unit ball centered at $-v$ intersects $\mathcal{B}^n$, too. As an example we illustrated these balls in Fig. 1 when the dimension is 2. We want to bound the volume of the intersection of these 3 unit balls, i.e. the volume of $S = (\mathcal{B}^n(1) \cap \mathcal{B}^n(v,1)) \cup (\mathcal{B}^n(1) \cap \mathcal{B}^n(-v,1))$. Our redu ction fails in the situation where $\|v\|$ is such a constant that for sufficiently large dimension n, the volume of S is negligible. So we only consider the situation where $\|v\| = 2\varepsilon$ is sufficiently small. For convenience of analysis, let $k\varepsilon = 1, k \in \mathbb{Z}$. In the case $k' \notin \mathbb{Z}$, we can set $k = \lfloor k' \rfloor$, and all following inequalities still hold. We can rewrite the volume of $\mathcal{B}^n$ as follows

$$\begin{aligned} V(\mathcal{B}^n) &= 2V(\mathcal{B}^{n-1}) \sum_{i=1}^{k} \int_{(i-1)\varepsilon}^{i\varepsilon} (1-x^2)^{(n-1)/2} dx \\ &> 2V(\mathcal{B}^{n-1}) \sum_{i=1}^{k} \left(\frac{1-(i\varepsilon)^2}{1-\varepsilon^2}\right)^{(n-1)/2} \int_0^{\varepsilon} (1-x^2)^{(n-1)/2} dx \\ &> 2V(\mathcal{B}^{n-1}) \left(1 + (1-4\varepsilon^2)^{(n-1)/2}\right) \int_0^{\varepsilon} (1-x^2)^{(n-1)/2} dx. \end{aligned} \tag{2}$$

Instead of directly calculating the volume of S, we bound the volume of $V(\mathcal{B}^n(1)) - V(S)$ as follows

$$\begin{aligned} V(\mathcal{B}^n) - V(S) &= \int_{-\varepsilon}^{\varepsilon} V\left(\mathcal{B}^{n-1}(\sqrt{1-x^2})\right) - V\left(\mathcal{B}^{n-1}\left(\sqrt{1-(2\varepsilon-|x|)^2}\right)\right) dx \\ &= 2V(\mathcal{B}^{n-1}) \left(\int_0^{\varepsilon} (1-x^2)^{(n-1)/2} - \int_{\varepsilon}^{2\varepsilon} (1-x^2)^{(n-1)/2} dx \right) \\ &< 2V(\mathcal{B}^{n-1})(1-(1-4\varepsilon^2)^{(n-1)/2}) \int_0^{\varepsilon} (1-x^2)^{(n-1)/2} dx. \end{aligned} \tag{3}$$

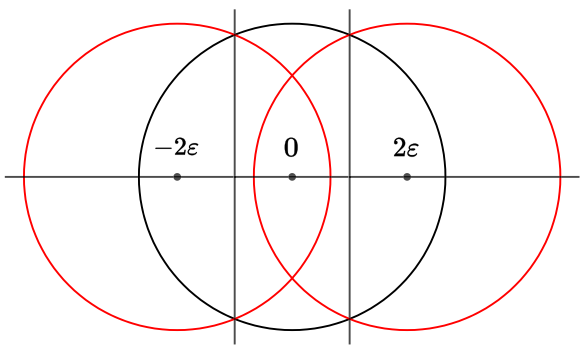


Fig. 1 Intersection of 3 balls in dimension 2

Let $\varepsilon = c_0\sqrt{\log(n-1)/(n-1)}$, for sufficiently large n we have

$$\begin{aligned} \frac{V(S)}{V(\mathcal{B}^n)} &= 1 - \frac{V(\mathcal{B}^n) - V(S)}{V(\mathcal{B}^n)} \\ &> 1 - \frac{1-(1-4\varepsilon^2)^{(n-1)/2}}{1+(1-4\varepsilon^2)^{(n-1)/2}} \\ &= 2\left(1 - \frac{1}{1+(1-4\varepsilon^2)^{(n-1)/2}}\right) \\ &\approx 2\left(1 - \frac{1}{1+e^{-2c_0^2\log(n-1)}}\right). \end{aligned} \tag{4}$$

Let c_0 be a constant such that $2c_0^2 \log(n-1)\log(e) < \log(n^k - 1)$, we have

$$\frac{V(S)}{V(\mathcal{B}^n)} > 1 - \frac{1}{1+(n^k-1)^{-1}} = \frac{2}{n^k} > \frac{1}{n^k}. \tag{5}$$

With this result, we have the following lemma for lattices

Lemma 1 *For any integer $k \geq 1$, let c_0 be a constant such that $2c_0^2 \log(n-1)\log(e) \leq \log(n^k-1)$, $\varepsilon \leq c_0\sqrt{\frac{\log(n-1)}{n-1}}$, and x be a vector in $\mathbb{R}^n$ such that $\|x\| \leq d$. If s is sampled uniform randomly form $\mathcal{B}^n(\frac{1}{2\varepsilon}d)$, then with probability at least $\frac{1}{n^k}$ we have the length of either $s-x$ or $s+x$ is at most $\frac{1}{2\varepsilon}d$.*

Collary 1 For $k \geq 2$, let c_0 be any constant, lemma 1 holds for all sufficiently large n. Especially, lemma 1 holds for

$$\varepsilon = c_0\sqrt{\frac{\log(n)}{n}} < c_0\sqrt{\frac{\log(n-1)}{n-1}}. \tag{6}$$

Hardness of uSVP

In this section we construct the reduction from GapSVP$_\gamma$ to γ'-uSVP, where $\gamma = O(\sqrt{n/\log(n)})$ and $\gamma' = O(1)$. Actually, we used the same reduction which was used by Lyubashevsky (2008) with different parameters. Lyubashevsky established the connection between GapSVP$_\gamma$ and $\frac{\gamma}{6\sqrt{n}}$-uSVP. Different with Lyubashevsky (2008), our reduction proved that γ'-uSVP, where γ' being any constant, is at least in NP $\cap$ coAM (Goldreich and Goldwasser 2000), which showed us some hope for proving the NP-hardness of γ'-uSVP. This is even better than a possible result mentioned by the author in Lyubashevsky (2008), where it was conjectured that the uniqueness factor of uSVP can be optimized to be $\gamma \cdot \sqrt{\log(n)/n}$ (the corresponding gap of GapSVP should be $O(\sqrt{n/\log(n)})$, this is the same with the parameter resulted by our reduction).

Algorithm 1 GapSVP$_\gamma$

Input: Basis $B_0 \in \mathbb{R}^{n\times n}$, $d \in \mathbb{R}$
Output: Whether (B_0, d) is a YES GapSVP$_\gamma$ instance or not

for $j = 1$ to n^3 **do**
  $s \xleftarrow{U} \mathcal{B}^n(c\sqrt{n/\log(n)}d)$, $t \leftarrow s \mod B_0$, $BetaWasOne \leftarrow false$
  for $i = 0$ to $\lceil \log(\|b_1\|) - \log(\gamma d)\rceil$ **do**
    $\alpha \leftarrow 2^i \cdot \frac{c}{2c'}\gamma d/2$
    $B \leftarrow \begin{bmatrix} B_0 & t \\ 0 & \alpha \end{bmatrix}$
    $w \leftarrow \frac{c'}{3\sqrt{2}c}$-uSVP$(B)$
    rewrite $w = (v - t\beta, -\alpha\beta)$, where $v \in \mathcal{L}(B_0), \beta > 0 \in \mathbb{Z}$
    if $\beta = 1, \|v-t\| \leq c\sqrt{n/\log(n)}d$ and $v \neq t - s$ **then**
      Output YES and terminate
    else if $\beta = 1, \|v-t\| \leq c\sqrt{n/\log(n)}d$ and $v = t - s$ **then**
      $BetaWasOne \leftarrow true$
    end if
  end for
  if $BetaWasOne = false$ **then**
    Output YES and terminate
  end if
end for
Output NO

The reduction procedure takes as input a basis $B_0 \in \mathbb{R}^{n\times n}$ and a real number d as a GapSVP$_\gamma$ instance. We will proved that this procedure output YES if $\lambda(B_0) \le d$ with probability exponentially close to 1, and output NO if $\lambda(B_0) > \gamma d$. The basic idea of this reduction is that we can distinguish between YES and NO instance of GapSVP$_\gamma$ with access to an oracle for γ'-uSVP. More specificly, a new basis B was constructed by adding an extra vector, say s, to B_0. Then we are able to proved that if $\lambda(B_0) > \gamma d$, with properly parameters, the procudure can find s. On the other hand, if $\lambda(B_0) \le d$, with reasonalble probability, NO procedure can tell s from some other vectors and hence may output any one of them. Hence we know that the original (B_0, d) is a YES instance once the procedure output a short vector other than s. Similar with Lyubashevsky (2008), we write the following theorem as a summary of this reduction.

Theorem 1 *For any constant c_0 satisfies lemma 1, let $c = \frac{1}{2c_0}$, $c' > 3\sqrt{2}c$ and $\gamma = c'\sqrt{n/\log(n)}$, for any integer $k \ge 2$ and all sufficiently large n, GapSVP$_\gamma$ reduces to $\frac{c'}{3\sqrt{2}c}$ -uSVP in polynomial time under randomized reduction.*

Proof of Theorem 1

Now let's prove that reduction 1 behaves right as expected under the situations where (B_0, d) is a YES and NO instance of GapSVP$_\gamma$.

On one hand, assume that (B_0, d) is a NO instance. In this case, we have $\lambda(B_0) > \gamma d$, and $dist(t, \mathcal{L}(B_0)) \le \|s\| \le c\sqrt{n/\log(n)}d \le \frac{c}{c'}\lambda(B_0)$. Notice that reduction 1 only output YES in two places. For the first place, we have $\beta = 1, \|v - t\| \le c\sqrt{n/\log(n)}d$ and $v \ne t - s$. Notice that $t - s \in \mathcal{L}(B_0)$, we can prove $\|v - (t - s)\| < \lambda(B_0)$ by the following

$$\begin{aligned}\|v - (t-s)\| &\le \|v - t\| + \|s\| \\ &\le 2c\sqrt{n/\log(n)}d \\ &\le c\sqrt{n/\log(n)}\lambda(B_0)/\gamma \\ &\le \frac{c}{c'}\lambda(B_0).\end{aligned} \tag{7}$$

This contradits the definition of $\lambda(B_0)$.

In the second place, we have that *BetaWasOne* was never set to true. According to lemma 2, there is an α such that

$$\begin{aligned}\|v - t, -\alpha\| &= \sqrt{\|v - t\|^2 + \alpha^2} \\ &\le \sqrt{2}c\sqrt{n/\log(n)}\lambda(B_0)/\gamma \\ &\le \frac{\sqrt{2}c}{c'}\lambda(B_0).\end{aligned} \tag{8}$$

Moreover, $(v - t, -\alpha)$ is the $\frac{c'}{3\sqrt{2}c}$-unique shortest vector in $\mathcal{L}(B)$. Notice that $\alpha = 2^i \cdot \frac{c}{2c'}\gamma d$, with $0 \le i \le \lceil \log(\|b_1\|) - \log(\gamma d)\rceil$. We have α ranges from

$$\alpha = 2^0 \cdot \frac{c}{2c'}\gamma d \le \frac{c}{2c'}\lambda(B_0) \tag{9}$$

to

$$\alpha = 2^{\lceil \log(\|b_1\|) - \log(\gamma d)\rceil} \cdot \frac{c}{2c'}\gamma d \ge \frac{c}{2c'}\|b_1\| \ge \frac{c}{2c'}\lambda(B_0). \tag{10}$$

Since α is multiplied by 2 in each loop, there exist an i makes $\frac{c}{2c'}\lambda(B_0) \le \alpha \le \frac{c}{c'}\lambda(B_0)$ holds. When calling the $\frac{c'}{3\sqrt{2}c}$-uSVP oracle with the corresponding matrix B as input, the oracle would return the unique vector $\|w\| = \|(v - t, -\alpha)\| = \lambda(B)$, which satisfies $\beta = 1, \|v - t\| \le c\sqrt{n/\log(n)}d$ and $v = t - s$. The variable *BetaWasOne* is set to be true, hence it won't output YES.

Combine all above, we proved that on input a NO instance (B_0, d), procedure 1 never output YES for all j. This proved the correctness of this reduction when (B_0, d) is a NO instance.

On the other hand, assume that (B_0, d) is a YES instance, we have $\lambda(B_0) \le d$. Obviously, on input a YES instance, with high probability, the constructed lattice B is not a $\frac{c'}{3\sqrt{2}c}$-uSVP instance. Hence, the $\frac{c'}{3\sqrt{2}c}$-uSVP oracle won't behave in any expected way. Notice that this procedure only output NO when *BetaWasOne* was set to be true for every sampled s. We can assume that the oracle always tries to prevent procedure 1 to output the correct answer. Let's now bound the probability of procedure 1 output NO, we denote this event as E. When E happens, *BetaWasOne* is set to be true for every s. This means that the oracle output a $w = (v - t, -\alpha)$ which satisfies $\|v - t\| \le c\sqrt{n/\log(n)}d$ and $v = t - s$. Notice that t is fixed once s is sampled from $\mathcal{B}(c\sqrt{n/\log(n)}d)$. Hence output such a w is equivalent with output s, which means that the oracle knows s. Howerver, by setting $k = 2$ in lemma 1, this only happens with negligible probability for the reason that in each loop (for each j) with probability at least $\frac{1}{n^2}$ there exists no algorithm that can tell s apart

from one of $\pm v_0 - s$. Where v_0 is one of the shortest vector in $\mathcal{L}(B_0)$, and $t \equiv s \equiv \pm v_0 - s \mod B_0$. Hence, the reduction procedure set *BetaWasOne* to true with probability at most $(1 - \frac{1}{n^2}) + \frac{1}{2n^2} = 1 - \frac{1}{2n^2}$. As a result, after n^3 iterations, $\Pr[E] < (1 - \frac{1}{2n^2})^{n^3} \approx e^{-n/2}$, which is negligible for all sufficiently large n.

Lemma 2 *Given $B_0 \in \mathbb{R}^{n\times n}, t \in \mathbb{R}^{n\times 1}$ and positve real number α, consider the following matrix*

$$B = \begin{bmatrix} B_0 & t \\ 0 & \alpha \end{bmatrix}. \tag{11}$$

For properly chosen constant $c, c' > 3\sqrt{2}c$, $\gamma = c'\sqrt{n/\log(n)}$, if

$$\begin{aligned} &\frac{c}{2c'}\lambda(B_0) \le \alpha \le \frac{c}{c'}\lambda(B_0), \\ &dist(t, \mathcal{L}(B_0)) \le \frac{c}{c'}\lambda(B_0), \end{aligned} \tag{12}$$

then $\mathcal{L}(B)$ has a $\frac{c'}{3\sqrt{2}c}$-unique shortest vector. Specifically, if $v \in \mathcal{L}(B_0)$ satisfies $\|v - t\| = dist(t, \mathcal{L}(B_0))$, the vector $w = (v - t, -\alpha) \in \mathcal{L}(B)$ is the $\frac{c'}{3\sqrt{2}c}$-unique shortest vector.

Proof of Lemma 2

We start by proving that $\lambda(B)$ is indeed smaller than $\lambda(B_0)/3$, then finish the proof by showing that the length of any vector other than the multiple of w is big, sepcifically, greater than $\lambda(B_0)/3$.

For the value of $\lambda(B)$ we have

$$\begin{aligned} \lambda(B) \le \|w\| &= \sqrt{\|v - t\|^2 + \alpha^2} \\ &\le \sqrt{\left(\frac{c}{c'}\lambda(B_0)\right)^2 + \left(\frac{c}{c'}\lambda(B_0)\right)^2} \\ &\le \frac{\sqrt{2}c}{c'}\lambda(B_0). \end{aligned} \tag{13}$$

Now let's finish this proof by showing that all vector $w' \neq kw, k \in \mathbb{Z}$ are long. For the sake of contradiction, assume that $\|w'\| \le \lambda(B_0)/3$. Write $w' = (v' - t\beta, -\beta\alpha)$, where $v' \in \mathcal{L}(B_0)$. If $\beta \ge \frac{2c'}{3c}$, we have $\beta\alpha \ge \beta\frac{c}{2c'}\lambda(B_0) \ge \lambda(B_0)/3$. If $\beta = 0$, $\|w'\| = \|v'\|$, since $v' \in \mathcal{L}(B_0), \|v'\| \ge \lambda(B_0)$. Hence we can limit $0 < \beta < \frac{2c'}{3c}$. By our assumption $\|v' - t\| < \|w'\| \le \lambda(B_0)/3$. Recall that $v \in \mathcal{L}(B_0)$ satisfies $\|v - t\| \le \frac{c}{c'}\lambda(B_0)$. We have the following

$$\begin{aligned} \|v' - \beta v\| &= \|(v' - t\beta) - (\beta v - t\beta)\| \\ &\le \|v' - t\beta\| + \beta\|v - t\| \\ &< \frac{1}{3}\lambda(B_0) + \frac{2}{3}\lambda(B_0) \\ &= \lambda(B_0). \end{aligned} \tag{14}$$

This is a contradiction since $v, v' \in \mathcal{L}(B_0)$.

As a conclusion, we have $\frac{c'}{3\sqrt{2}c}\lambda_1(B) < \lambda_2(B)$, and there is a unique vector w satisfies $\| \pm w\| = \lambda(B)$.

Conclusion

We have proved that, for any constant $\frac{c'}{3\sqrt{2}c}, \frac{c'}{3\sqrt{2}c}$-uSVP is at least as hard as $\text{GapSVP}_{c'\sqrt{n/\log(n)}}$, and hence $\frac{c'}{3\sqrt{2}c}$-uSVP lies at least in $NP \cap coAM$. Especially, the constant of the approximation factor of GapSVP is irrelevant with c. Our result established a hardness result for uSVP which allows one to choose its uniqueness factor at wish. From the perspecitve of complexity theory, we gave a support for the possibility that uSVP is NP-hard for constant uniqueness factors.

Combining our result for uSVP and the reduction in Lyubashevsky and Micciancio (2009), which reduce γ-uSVP to $\frac{1}{\gamma}$-BDD, we get a similar hardness result for appriximate BDD. Compared with Lyubashevsky and Micciancio (2009) our reduction provided more flexibility for the choice of parameters for GapSVP instance. Especially, we reduced $\text{GapSVP}_{c'\sqrt{n/\log n}}$ to $\frac{3\sqrt{2}c}{c'}$-BDD. T value of c' can be an arbitrary small constant, while it must be greater than 2 in the result of Lyubashevsky and Micciancio (2009).

At the end, we emphasize again that the reduction in this paper dosn't apply for $\text{GapSVP}_{O(\sqrt{n/\log^k n})}, k > 1$. New ideas are needed to obtain such a result.

Abbreviations

SVP	Shortest vector problem
GapSVP	Gap shortest vector problem
uSVP	Unique shortest vector problem
BDD	Bounded distance decoding problem.

Acknowledgements

There is no any third person/ organisation to acknowledge

Author contributions

All authors read and approved the final manuscript.

Funding

This work is funded by National Natural Science Foundation of China (Grants No. 62172405).

Availability of data and materials
Not applicable

Declarations

Ethics approval and consent to participate
Not applicable

Consent for publication
Not applicable

Competing interests
The authors declare that they have no competing interests.

Author details
[1]State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100085, China. [2]School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China.

Received: 30 March 2023 Accepted: 29 June 2023
Published online: 04 November 2023

References

Aggarwal D, Dubey CK (2016) Improved hardness results for unique shortest vector problem. Inf Process Lett 116(10):631–637. https://doi.org/10.1016/j.ipl.2016.05.003
Ajtai M (1996) Generating hard instances of lattice problems (extended abstract). In: Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing, Association for Computing Machinery, STOC '96 New York, pp 99–108
Ajtai M (1998) The shortest vector problem in l2 is np-hard for randomized reductions (extended abstract). In: Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing, Association for Computing Machinery, STOC '98 New York, pp 10–19
Ajtai M, Dwork C (1997) A public-key cryptosystem with worst-case/average-case equivalence. In: Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing, Association for Computing Machinery, STOC '97 New York, pp 284–293
Boas P (1981) Another NP-complete partition problem and the complexity of computing short vectors in a lattice. https://staff.fnwi.uva.nl/p.vanemdeboas/vectors/mi8104c.html
Cai J (1998) A relation of primal-dual lattices and the complexity of shortest lattice vector problem. Theor Comput Sci 207(1):105–116. https://doi.org/10.1016/S0304-3975(98)00058-9
Cai J, Cusick TW (1999) A lattice-based public-key cryptosystem. Inf Comput 151(1–2):17–31. https://doi.org/10.1006/inco.1998.2762
Goldreich O, Goldwasser S (2000) On the limits of nonapproximability of lattice problems. J Comput Syst Sci 60(3):540–563. https://doi.org/10.1006/jcss.1999.1686
Khoat TQ, Tan NH (2008) Unique shortest vector problem for max norm is NP-hard. Cryptology ePrint Archive, Paper 2008/366. https://eprint.iacr.org/2008/366
Khot S (2003) Hardness of approximating the shortest vector problem in high l_p norms. , In: 44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings. pp 290–297
Khot S (2004) Hardness of approximating the shortest vector problem in lattices. In: 45th Annual IEEE Symposium on Foundations of Computer Science pp 126–135
Kumar R, Sivakumar D (2001) On the unique shortest lattice vector problem. Theor Comput Sci 255(1–2):641–648. https://doi.org/10.1016/S0304-3975(00)00387-X
Liu M, Wang X, Xu G, Zheng X (2011) Shortest lattice vectors in the presence of gaps. Cryptology ePrint Archive, Paper 2011/139. https://eprint.iacr.org/2011/139
Lyubashevsky V (2008) The n^c-unique shortest vector problem is hard. Cryptology ePrint Archive, Paper 2008/504. https://eprint.iacr.org/2008/504
Lyubashevsky V, Micciancio D (2009) On bounded distance decoding, unique shortest vectors, and the minimum distance problem. In: Halevi S (ed) Advances in cryptology - CRYPTO 2009. Lecture Notes in Computer Science Berlin, vol 5677, Springer, Heidelberg, pp 577–594
Micciancio D (1998) The shortest vector in a lattice is hard to approximate to within some constant. , In: Proceedings 39th Annual Symposium on Foundations of Computer Science (Cat. No.98CB36280), pp 92–98
Micciancio D (2004) Almost perfect lattices, the covering radius problem, and applications to ajtai's connection factor. SIAM J Comput 34(1):118–169. https://doi.org/10.1137/S0097539703433511
Micciancio D (2012) Inapproximability of the shortest vector problem: toward a deterministic reduction. Theory Comput 8(1):487–512. https://doi.org/10.4086/toc.2012.v008a022
Peikert C (2009) Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In: Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing, Association for Computing Machinery, STOC '09 New York, pp 333–342
Regev O (2003) New lattice based cryptographic constructions. In: Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing, Association for Computing Machinery, STOC '03 New York, pp 407–416
Sauer N (1972) On the density of families of sets. J Comb Theory Ser A 13(1):145–147. https://doi.org/10.1016/0097-3165(72)90019-2
Shelah S (1972) A combinatorial problem; stability and order for models and theories in infinitary languages. Pac J Math 41:247–261. https://doi.org/10.2140/pjm.1972.41.247
Stephens-Davidowitz N (2016) Search-to-decision reductions for lattice problems with approximation factors (slightly) greater than one. In: Jansen K, Mathieu C, Rolim JDP, Umans C (eds) Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2016). vol. 60, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Leibniz International Proceedings in Informatics (LIPIcs) Dagstuhl, Germany, pp. 19–11918
Vapnik VN, Chervonenkis AY (2015) On the uniform convergence of relative frequencies of events to their probabilities. Springer, Cham, pp 11–30
Wei W, Liu M, Wang X (2015) Finding shortest lattice vectors in the presence of gaps. In: Nyberg K (ed), Topics in cryptology—CT-RSA 2015 Cham, Springer, pp 239–257

Publisher's Note