Why we couldn’t prove SETH hardness of the Closest Vector Problem for even norms!

Divesh Aggarwal
Centre for Quantum Technologies
National University of Singapore
Singapore
dcsdiva@nus.edu.sg

Rajendra Kumar
Computer Science and Applied Mathematics
Weizmann Institute of Science
Rehovot, Israel
rjndr2503@gmail.com

Abstract—Recent work has shown SETH hardness of CVP in the \( \ell_p \) norm for any \( p \) that is not an even integer. This result was shown by giving a Karp reduction from \( k\)-SAT on \( n \) variables to CVP on a lattice of rank \( n \). In this work, we show a barrier towards proving a similar result for CVP in the \( \ell_p \) norm where \( p \) is an even integer. We show that for any \( \epsilon > 0 \), if for every \( k > 0 \), there exists an efficient reduction that maps a \( k\)-SAT instance on \( n \) variables to a CVP instance for a lattice of rank at most \( n^3 \) in the Euclidean norm, then coNP \( \subset \) NP/Poly. We prove a similar result for CVP for all even norms under a mild additional promise that the ratio of the distance of the target from the lattice and the shortest non-zero vector in the lattice is bounded by \( \exp(n^{\Omega(1)}) \).

Furthermore, we show that for any \( \epsilon > 0 \), and any even integer \( p \), if for every \( k > 0 \), there exists an efficient reduction that maps a \( k\)-SAT instance on \( n \) variables to a SVP instance for a lattice of rank at most \( n^3 \), then coNP \( \subset \) NP/Poly.

While prior results have indicated that lattice problems in the \( \ell_2 \) norm (Euclidean norm) are easier than lattice problems in other norms, this is the first result that shows a separation between these problems.

We achieve this by using a result by Dell and van Melkebeek on the impossibility of the existence of a reduction that compresses between these problems.

In addition to CVP, we also show that the same result holds for the Subset-Sum problem using similar techniques.

Index Terms—Lattices, Fine-grained hardness, Post-Quantum Cryptography, Closest Vector Problem, Shortest Vector Problem

I. INTRODUCTION

A. Lattice Problems

A lattice \( \mathcal{L} \) is the set of integer linear combination of \( n \) linearly independent vectors \( b_1, b_2, \ldots, b_n \in \mathbb{R}^m \), i.e.

\[
\mathcal{L}(b_1, b_2, \ldots, b_n) := \left\{ \sum_{i=1}^{n} z_i b_i : \forall i \in [n], z_i \in \mathbb{Z} \right\}.
\]

We call \( n \) as the rank of the lattice, \( m \) as the dimension of the lattice and \( \mathcal{B} = \{b_1, b_2, \ldots, b_n\} \) is a basis of the lattice. There exist many different basis of a lattice.

The two most important computational problems on lattices are the Shortest Vector Problem (SVP) and the Closest Vector Problem (CVP). In the Shortest Vector Problem, given a basis for a lattice the goal is to output a shortest non-zero lattice vector. For \( \gamma > 1 \), in the \( \gamma\)-approximation of SVP (\( \gamma\)-SVP) the goal is to output a nonzero lattice vector whose length is at most \( \gamma \) times the length of shortest non-zero lattice vector.

In the Closest Vector Problem, given a target vector and basis for a lattice, the goal is to output a closest lattice vector to the target vector. For \( \gamma > 1 \), in the \( \gamma\)-approximation of CVP, (\( \gamma\)-CVP) the goal is to output a lattice vector whose distance from target vector is at most \( \gamma \) times the minimum distance between the target vector and lattice. In this work, we only consider the length and distance in the \( \ell_p \) norms, defined as follows. For \( 1 \leq p < \infty \)

\[
\|x\|_p := \left( \sum_{i=1}^{m} |x_i|^{p} \right)^{1/p},
\]

and for \( p = \infty \),

\[
\|x\|_\infty := \max_{i=1}^{m} \{ |x_i| \}.
\]

We write SVP\(_p\) and CVP\(_p\) for the respective problems in \( \ell_p \) norm. The most commonly used norm is the \( \ell_2 \) norm which is also called the Euclidean norm. The CVP is known to be at least as hard as SVP (in the same norm). More specifically, there is an efficient polynomial-time reduction from SVP\(_p\) to CVP\(_p\), which preserves the dimension, rank, and approximation factor [1].

Computational problems on lattices are particularly important due to their connection to lattice-based cryptography. Most specifically, the security of many cryptosystems [2]–[8] is based on the hardness of polynomial approximation of lattice problems. Other than the design of cryptosystems, from the 80’s the solvers for lattice problems has its application in algorithmic number theory [9], convex optimization [10], [11], and cryptanalytic tools [12]–[14].

Algorithms for CVP have been extensively studied for a long time. Kannan gave an enumeration algorithm [10] for CVP that works for any norm and takes \( n^{O(n)} \) time and requires \( poly(n) \) space. Micciancio and Voulgaris gave a deterministic algorithm for CVP\(_2\) that takes \( 2^{2n+o(n)} \) time.
and requires $2^{n+o(n)}$ space. Aggarwal, Dadush, and Stephens-Davidowitz [15] gave the current fastest known algorithm for CVP; it takes $2^{n+o(n)}$ time and space. However, there is no progress in solving exact CVP in $\ell_p$ norm; still, Kannan’s algorithm is the fastest for exact CVP for any arbitrary $p$. For a small approximation of CVP, Blomer and Seifert [16] gave an algorithm that runs in $2^{O(n)}$ time. Later, Dadush [17] improved it by giving a $2^{O(n)}$ time algorithm. For $p = \infty$, Aggarwal and Mukhopadhyay [18] gave a $2^{2n+o(m)}$ time algorithm. Recently, Eisenbrand and Venzin [19], gave an algorithm for a (large enough) constant-factor approximation of CVP for any $\ell_p$ norm in $2^{n\cdot 0.802^{m+o(m)}}$ time.

The CVP in any norm is NP-hard [20]. It is also known to be NP-hard for almost polynomial approximation $n^{c'/\log \log n}$ for some constant $c > 0$ [21], [22]. All of these hardness results do not say anything about the fine-grained hardness of lattice problems. In particular, it is not possible to say anything about possibility/impossibility of a $2^{\sqrt{n}}$ or a $2^{2^{n/100}}$ time algorithm for CVP. All known cryptanalytic attacks on lattice-based cryptosystems proceed via solving near exact lattice problems in a small dimensional lattice, and so, if one were to find an algorithm for CVP or SVP that runs in, say, $2^{2^{n/100}}$ time, then it will break all lattice-based cryptosystems currently considered to be practical. This immediately leads to the question whether such attacks can be ruled out by giving appropriate lower bounds for lattice problems under reasonable assumptions.

Motivated by the above question, Bennett, Golovnev, and Stephens-Davidowitz [23] initiated the study of the fine-grained hardness of CVP and its variants. They showed that, for any constant $\epsilon > 0$ and $p \notin 2\mathbb{Z}$, if there exists an algorithm for CVP that runs in time $2^{(1-\epsilon)n}$ then it refutes the Strong Exponential Time Hypothesis (SETH). The authors additionally also showed that for any $p \geq 1$, there exists no algorithm for CVP that runs in time $2^{O(n)}$ under the Exponential Time Hypothesis. Later, these results were extended to other lattice problems [24]–[27]. Aggarwal, Bennett, Golovnev, and Stephens-Davidowitz [28], improved the fine-grained hardness of CVP and showed that even approximating CVP to a factor slightly bigger than 1 is not possible in time $2^{(1-\epsilon)n}$ under the Gap variant of the Strong Exponential Time Hypothesis. This result was again only shown for $p \notin 2\mathbb{Z}$. This immediately leads to the question whether such a hardness result is possible for $p \in 2\mathbb{Z}$, or if there is a fundamental barrier that does not allow such a result. This is particularly important/interesting for the Euclidean norm (i.e., $p = 2$) since the security of lattice-based cryptosystems is typically based on the hardness of lattice problems in the Euclidean norm. The authors [28] made a small progress towards answering this question by proving that there are no “natural” reductions from $k$-SAT on $n$ variables to CVP on a lattice of rank at most $4n/3$.2

Motivated by the fact that the computational problems in lattices run in time exponential in the dimension of the lattice, [29] initiated the study of exponential time reductions (reductions that run in time $2^m$, for some small constant $\epsilon$) between CVP and SVP in $\ell_p$ norms for different $p$. The techniques used to obtain the results in this work were based on [30]. Together, these results have shown that for large constant approximation factors, both CVP and SVP are not approximable in time $2^{\epsilon n}$ for any $p \geq 1$ are almost equivalent.

Unfortunately, our main result (almost) rules out the possibility of an efficient reduction from $k$-SAT on $n$ variables to CVP for any even integer $p$ with rank bounded by a fixed polynomial in $n$. We achieve this result by proving that a CVP instance for any even integer $p$ can be compressed to size that is roughly a fixed polynomial in the rank of the lattice, while $k$-SAT is only known to be compressible in $O(n^k)$ bits.

### B. Subset Sum Problem

Motivated by the difficulty in proving SETH hardness for CVP for even norms, we consider the closely related but relatively simpler Subset Sum problem. The Subset Sum problem is among the most fundamental computational problems described as follows. Given $n$ positive integers $x_1, \ldots, x_n$, and a target value $t$, the goal is to decide whether there exists a subset of the $n$ positive integers that sums to $t$.

The classical algorithms for this problem run in time $O((tn))$ via dynamic programming [31], and in time $2^{n/2} \cdot poly(\log t, n)$ using meet-in-the-middle approach [32]. An important open question in the theory of exact exponential time algorithms for hard problems [33] is whether any of these algorithms for Subset Sum can be improved.

It has been shown in [34], [35] that under the Exponential Time Hypothesis, there is no $2^{(n^{0.5})} \cdot poly(1)$ algorithm for Subset Sum. In another line of work, it has been shown that there is no $O(t^{1-\epsilon}) \cdot poly(n)$-time algorithm under the Set cover conjecture [36], and also under SETH [37].

This still leaves open the question whether one can rule out the possibility of a $2^{\log t} \cdot poly(\log t)$-time algorithm for Subset Sum for some fixed constant $c < 1/2$ under a reasonable conjecture such as the Set Cover conjecture or SETH.

### C. Instance Compression

Harnik and Naor [38] studied instance compression of NP decision problems. They focus on the problem with long instances but relatively small witness sizes, and ask whether the instance size can be reduced while preserving the information whether the input instance is in the language or not. Moreover, a compressed instance may not be of the same problem. This compressed instance can then be used to solve the problem in the future, maybe by using technological advances or with some algorithmic improvement. A problem is said to be efficiently instance compressible if there exists a compression in size polynomial in the witness size, and polylogarithmic in the length of input size. They introduce new subclasses of NP depending on the compression size. They also study the implication of compression in cryptography. If we can have instance compression of problems like OR-SAT,
we can use these compressions to get cryptographic primitives, for example, collision-resistant hash functions.

It is easy to construct an efficient compression of a $\text{GapSAT}$ instance when the gap is $\left(1 - \frac{1}{\text{poly}(n)}\right)$. Create an instance of SAT by sampling a subset of clauses of $\text{poly}(n)$ size from $\text{GapSAT}$ instance uniformly at random. This instance is satisfiable if GapSAT instance is satisfiable; otherwise, it is unsatisfiable with high probability. Inspired by this compression, Harnik and Naor proposed the following problem: “Can we have an efficient reduction from SAT to GapSAT for which the number of variables in GapSAT depends only on the number of variables in the input SAT instance?” Note that a positive answer to this question will give a succinct PCP for the SAT problem. Later, Fortnow and Santhanam [39] gave a coNP by sampling a subset of clauses of instance that the target’s distance from the lattice is at SAT to these problems. It is easy to see that $O \ell$ problem. Later, Fortnow and Santhanam [39] gave a coNP by sampling a subset of clauses of instance that the target’s distance from the lattice is at

Later, Fortnow and Santhanam [39] gave a coNP by sampling a subset of clauses of instance that the target’s distance from the lattice is at

SAT to these problems. It is easy to see that $O \ell$ problem. Later, Fortnow and Santhanam [39] gave a coNP by sampling a subset of clauses of instance that the target’s distance from the lattice is at

We focus on the fine-grained hardness of CVP in the $\ell_p$ norm for $p \in \mathbb{Z}^+$. We say that a probabilistic reduction does not have false negatives if, for any oracle call that is made with a YES instance, the oracle responds YES with probability 1. We show the following impossibility result about reduction from satisfiability problem:

**Theorem 1.1** (Informal, see Theorems VII.1 and VII.2). For any even positive integer $p$ and constant $c > 0$, there exists a constant $k_0$ such that for all $k > k_0$, there is no polynomial time probabilistic reduction without false negatives from $k$-SAT on $n$-variables to CVP$_p$ on $O(n^c)$ rank lattice that make at most $O(n^c)$ calls to CVP$_p$ oracle, unless coNP $\subseteq$ NP/Poly.

For even $p$ greater than 2, we need an additional promise on CVP$_p$ instance that the target’s distance from the lattice is at most $\exp(n^{O(1)})$ factor large than the shortest non-zero vector in the lattice. For SVP$_p$ for all even $p$, we show the barrier for exact-SVP$_p$(without any additional promise needed).

Our result says that for any even integer $p$ and constant $c > 0$, it is not possible to get $2^{cn}$ SETH-hardness by a deterministic polynomial time Karp reduction from $k$-SAT to CVP$_p$, unless the polynomial hierarchy collapses to the third level. We also rule out Turing reductions, which makes less than $n^k$ calls to CVP$_p$ oracle. These impossibility results also hold for a probabilistic reduction, as long as the oracle does not output false negatives. This result, in particular, explains why [23], [28] could not prove SETH-hardness of CVP$_p$ for even $p$. They showed SETH-hardness of CVP$_p$ for non-even $p$ by a Karp-reduction (shown in Figure 1) from $k$-SAT on $n$-variables to CVP$_p$ on lattice of rank $n$. The above theorem says (under a complexity-theoretic assumption) that there exists a constant $k$ for which it is impossible to get a polynomial time-reduction Karp reduction from $k$-SAT to CVP$_p$ on $O(n)$ rank lattice for even $p$. On the contrary, our result does not rule out any fine-grained polynomial-time reductions. For example, it does not mention the possibility of Turing reductions, which make $n^k$ calls to CVP$_p$ oracle. Note that there also exists a polynomial time reduction from $k$-SAT to CVP$_p$ on the lattice of rank $n^{O(k)}$, but this reduction does not say anything about $2^{cn}$ fine-grained hardness of CVP.

Notice that the above theorem does not say anything about the possibility of a super-polynomial time reduction. We know that CVP$_p$ for any $p \geq 1$ is $2^{o(n)}$-hard (shown in
Figure 3) assuming the Exponential Time Hypothesis [23]. This reduction from \( k \)-SAT makes exponential number of calls to the CVP\(_p\) oracle.

If we believe that coNP is not contained in NP/poly, then we can only hope to get SETH-hardness for CVP\(_p\) for even \( p \) by a super-polynomial time reduction. Moreover, we give a barrier for a specific class of super-polynomial time reductions. We generalize our result for a range of different running times.

**Theorem 1.2** (Informal, see Theorems VII.4 and VII.5). For any even positive integer \( p \) and constant \( c > 0 \), there exists a constant \( k_0 \) such that for all \( k > k_0 \) and \( T \), there is no \( T \)-time probabilistic reduction with no false negatives from \( k \)-SAT on \( n \)-variables to CVP\(_p\) on \( O(n^c) \) rank lattice that make at most \( O(n^c) \) calls to a CVP\(_p\) oracle, unless coNP is in \( \text{NTime}(\text{poly}(n) \cdot T) \).

Notice that the above result shows a barrier for super-polynomial time-reductions that make only polynomial number of calls to the CVP\(_p\) oracle. We also study the barriers for probabilistic polynomial time reductions. However, we are only able to show this barrier for a non-adaptive reduction from \( k \)-SAT to CVP\(_p\) for even \( p \).

**Theorem 1.3** (Informal, see Theorems VII.7 and VII.8). For any even positive integer \( p \) and constant \( c > 0 \), there exists a constant \( k_0 \) such that for all \( k > k_0 \), there is no polynomial time probabilistic non-adaptive reduction from \( k \)-SAT on \( n \)-variables to CVP\(_p\) on \( O(n^c) \) rank lattice that make at most \( O(n^c) \) calls to the CVP\(_p\) oracle, unless there are non-uniform, statistical zero-knowledge proofs for all languages in NP.

We also observe that we can conclude the (im)possibility of SETH-hardness of Subset-Sum to get the following result.

**Theorem 1.4** (Informal, see Theorem VIII.2). For any constant \( c > 0 \), there exists a constant \( k_0 \) such that for all \( k > k_0 \), there is no polynomial time probabilistic reduction without false negatives from \( k \)-SAT on \( n \)-variables to Subset-Sum on \( O(n^c) \) numbers that make at most \( O(n^c) \) calls to Subset-Sum oracle, unless coNP is in NP/Poly.

**E. Our Techniques**

Instance compression of computational problems has interesting connection in fixed-parameter-tractable algorithms [42], [43] and Cryptography [38]. In this work, we initiate the study of instance compression for lattice problems, and shows its consequences to SETH hardness.

**a) Instance compression and Fine-grained hardness:**

We show a connection between the SETH-hardness and instance compression. Let’s say we are interested in fine-grained SETH-hardness of problem \( A \) for which there exists a polynomial time algorithm that gives instance compression of polynomial size. Note that, to show SETH-hardness we need efficient reduction from \( k \)-SAT for all constant \( k \). If there exists a fine-grained polynomial time reduction from \( k \)-SAT to problem \( A \) then it will immediately give an algorithm for polynomial size compression (independent of \( k \)) of \( k \)-SAT. However, Dell and van Melkebeek [40] showed that there does not exist any non-trivial compression for \( k \)-SAT problems unless the polynomial hierarchy collapses to the third level. In other words, if a computational problem has a polynomial size compression then it is impossible to get SETH-hardness by a polynomial time reduction unless the polynomial hierarchy collapses to the third level. Moreover, [40] gives the barrier for non-trivial oracle communication protocol for \( k \)-SAT. Oracle communication protocol (Definition II.11) can be seen as a generalized notion of instance compression. Using the impossibility of oracle communication protocol, we show a barrier for adaptive fine-grained reductions from \( k \)-SAT to a problem with polynomial size compression.

We also give barriers for SETH hardness of instance compressible problem by bounded error polynomial time probabilistic reduction. Drucker [41] showed that it is impossible to get a non-trivial probabilistic instance compression of \( k \)-SAT unless there are non-uniform, statistical zero-knowledge proofs for all languages in NP. Using a similar argument as above, we show a barrier for polynomial time probabilistic reduction from \( k \)-SAT to a problem that has a probabilistic algorithm for instance compression. However, we don’t know any barrier for the probabilistic oracle communication protocol, so we can only show a barrier for probabilistic non-adaptive fine-grained reduction.

Moreover, our work suggests that all computational problems can be classified into two classes: (i) problems that have a fixed polynomial size instance compression and (ii) problems for which it is not possible to find such compression. It is impossible to get polynomial-time fine-grained reductions from problems in (ii) class to problems in (i) by a polynomial time reduction.

**b) Instance compression of CVP in even norm:**

We present a polynomial time algorithm that gives \( O(n^{p+3}) \) bit-length instance compression for almost exact CVP\(_p\) for even \( p \) on rank \( n \) lattice. For this purpose, we introduce variants of CVP\(_p\); CVP\(^{\text{IP}}\) and CVP\(^{\text{mvp}}\). In CVP\(^{\text{IP}}\), given the inner product of basis vectors and target vector, the goal is to find the coefficient of a closest lattice vector to the target. This problem is well-defined from the fact that the euclidean distance between any lattice point and target vector can be computed, given the inner products of basis vectors and target, and coefficient of the lattice point in the underlying basis. So, there is also a trivial reduction from CVP\(_2\) to CVP\(^{\text{IP}}\). CVP\(^{\text{mvp}}\) is an extension of the CVP\(^{\text{IP}}\) for \( t_p \) norms when \( p \) is even. We show a polynomial time algorithm that reduces arbitrary CVP\(_p\) instance for even \( p \) to CVP\(^{\text{IP}}\)/CVP\(^{\text{mvp}}\) instance of fixed polynomial size.

For simplicity, here we will only present a sketch of compression algorithm for Euclidean norm. As mentioned above any CVP\(_2\) instance can be compressed by just storing the \( (n+1)^2 \) pairwise (with repetitions) inner products of the basis vectors \( b_1, \ldots, b_n \), and the target vector \( t \). The instance compression for CVP would be immediate from this if all co-ordinates of the vectors are bounded by \( 2^n \) for some constant \( c \), since that would imply a compression of (exact)
CVP to an instance of \(\text{CVP}^{\text{IP}}\) of size \((n + 1)^2 \log (m \cdot 2^{2n^2}) = (n + 1)^2 (2n^2 + \log n)\). In the following, we show how to decrease all the co-ordinates of the CVP instance while still retaining information of whether the instance is a YES/NO. For this, we need to consider approximate-CVP (with an approximation factor very close to 1) rather than exact CVP.

First, we transform the basis and target vector such that the coefficients of the closest lattice vector to the target vector are bounded by \(2^{n^2}\). To do this, we use the LLL algorithm [9]. Next, we want to bound the distance from the closest lattice vector by \(2^{n^2}\). To achieve this, we divide the lattice basis vectors and target vector by a carefully chosen large integer and ignore the fractional part of the vectors. As long as we only allow lattice vectors with coefficients bounded by \(2^{n^2}\), we show that this truncation of lower-order bits does not introduce any other close vectors. This implies that CVP reduces to a variant of CVP where the goal is to find a closest lattice vector whose coefficient is bounded by \(2^{n^2}\), and the distance from target is also bounded by \(2^{n^2}\). This step essentially uses the (small) gap between the Yes and NO instance\(^1\). Then, we reduce this to an instance where the basis and target vector coordinates are bounded by \(2^{n^2}\). For this, we choose a prime number significantly larger than the distance of the target from closest lattice vector and reduce all coordinates of the basis and target vector modulo the prime. The randomness of the prime is sufficient to guarantee that, with high probability, we do not introduce any new close vectors with bounded coefficients. So, finally, we reduce CVP to a variant of CVP with all coordinates bounded by \(O(n^2)\). More specifically, we reduce it to the problem of finding closest lattice vector with bounded coefficients. Now we reduce it to \(\text{CVP}^{\text{IP}}\) by computing the inner products of basis vectors and target.

We also demonstrate another technique for instance compression for \(\text{CVP}_2\) by utilizing a theorem from [11]. First, we apply a trivial reduction from \(\text{CVP}_2\) to \(\text{CVP}^{\text{IP}}\). Then, we employ a result from the theorem in [11] as a black-box to reduce it into a \(\text{CVP}^{\text{IP}}\) instance with inner products bounded by \(2^{O(n^2)}\). It’s worth noting that this instance compression technique is applicable to exact-CVP\(_2\). Furthermore, this technique can be extended to any even norm but requires an additional promise: that the distance of the target from the lattice is at most \(\exp(n^c)\) times the length of the shortest non-zero lattice vector.

### F. Comparison to previous works

In literature, there are results [44]–[46] that show the barrier for getting SETH-hardness of problems. In [44], authors propose Non-deterministic Strong Exponential Time Hypothesis (NSETH), which states that for every \(\varepsilon > 0\) there exists a \(k\) so that \(k\)-taut is not in \(\text{NTIME}(\exp((1-\varepsilon)^{n}))\), where \(k\)-taut is the language of all \(k\)-DNF which are tautologies. They gave faster co-nondeterministic algorithms for 3-SUM, APSP and model checking of a large class of first-order graph properties.

\(^1\) Note that if the distance from target vector is bounded by \(2^{n^2}\) then we get instance compression for exact-CVP\(_2\). They show that it is unlikely to get a fine-grained deterministic reduction from \(k\)-SAT to these problems. If there is a fine-grained reduction then it implies that \(k\)-taut has faster nondeterministic algorithm which contradicts NSETH.

In [45], the authors investigate the barriers to proving the SETH-hardness of Hamiltonian Path, Graph Coloring, Set Cover, Independent Set, Clique, Vertex Cover, and 3d-Matching. Specifically, they show that if a fine-grained reduction exists from \(k\)-SAT to any of these problems, it would imply new circuit lower bounds. In comparison to these results, our work focuses on ruling out fine-grained reductions for lattice problems and Subset-Sum under weaker conditions than those used in previous techniques. However, it should be noted that our conclusion is relatively weaker, as we cannot rule out Fine-grained Turing reductions that make superpolynomial calls.

Recently, in [46], authors shows barriers for SETH-hardness of constant approximation of CVP. This result does not say anything about SETH-hardness of near exact-CVP.

### G. Other Conclusion and Open Questions

There are several interesting observations that can be made about our main result in light of prior work. In the following, let \(q\) be quantified over \([1, \infty) \setminus 2\mathbb{Z}\), and \(p\) be quantified over \(2\mathbb{Z}^+\).

- It was shown in [28] that for all \(q\), \((1+\varepsilon)\)-approximate \(k\)-SAT on \(n\) variables can be reduced to \((1+\varepsilon/poly(k))\)-\(\text{CVP}_q\) on a rank \(n\) lattice. Notice that without loss of generality, one may assume that the number of clauses of a \(k\)-SAT instance is at most \(O(n^k)\), and thus \((1+1/n^k)\)-approximate \(k\)-SAT is the same as \(k\)-SAT. This implies that, from our result, one can conclude that there does not exist a \(\text{poly}(n)\)-time reduction from \((1+1/poly(n))\)-\(\text{CVP}_q\) on a rank \(n\) lattice to \(\text{CVP}_p\) on a \(\text{poly}(n)\)-rank lattice for any \(q \in [1, \infty) \setminus 2\mathbb{Z}\). Our result provides evidence that shows that \(\text{CVP}_2\) might be easier than \(\text{CVP}_q\) for \(q \in [1, \infty) \setminus 2\mathbb{Z}\). This conclusion can also be made with \(\text{CVP}_2\) replaced by \(\text{CVP}_p\), with a mild caveat that the \(\text{CVP}_p\) instance must satisfy the promise that the distance of the target from the lattice is at most \(\exp(n^{O(1)})\) factor larger than the shortest non-zero vector in the lattice.

- This result should be contrasted with [47] which showed that, approximate \(\text{CVP}_2\) is reducible to \(\text{CVP}_p\) with almost the same approximation factor, which also gave evidence that \(\text{CVP}_2\) might be easier than \(\text{CVP}_p\) in other \(\ell_p\) norms.

Our work helps take further our understanding of the limitations of the fine-grained hardness of CVP and SVP under the Strong Exponential Time Hypothesis. Some of the questions left open by our work are as follows.

- One interesting question that emerges from our work is the following. Prior work has shown that it is much easier to make algorithmic progress for CVP, SVP in the Euclidean norm [15], [48], [49], as opposed to the
corresponding problems in other $\ell_p$ norms. To our understanding, this was partially because our understanding of the Euclidean norm is much better than that for other norms. This work suggests that perhaps computational problems in the $\ell_2$ norm (and other even norms) are inherently easier, which suggests trying to find faster algorithms for SVP, CVP in $\ell_p$ norms where $p$ is an even integer.

- We need to introduce an additional promise on the CVP$_p$ instance for $p > 2$ for our compression algorithm to work. This doesn’t seem inherent, and is likely just a consequence of our techniques. The problem of removing this restriction is left open.

- While our work rules out the possibility of poly$(n)$-time reduction from $k$-SAT to CVP$_p$ and to the Subset-Sum problem, we do not rule out the possibility of such a reduction that makes more than poly$(n)$ calls to the oracle for the restricted problems. Ruling out such a reduction under a reasonable conjecture is a very interesting open question.

**Organization:** We give the preliminaries in Section II. In Section III, we propose the problems CVP$^\text{IP}$ and CVP$^\text{mvp}$. In Section IV, we give an instance compression algorithm for CVP in the Euclidean norm. We extend this and give an instance compression algorithm for CVP$_p$ for even $p$ in Section V. In Section VI, we present compression algorithm for exact-CVP in even norm. These results are significantly better than previous two sections but uses a theorem from [11] as black-box. We present the barriers for SETH-hardness of CVP in $\ell_p$ norms for even $p$ in Section VII. We show barrier for SETH hardness of Subset-Sum problem in Section VIII.

II. PRELIMINARIES

We use the notation $\mathbb{R}$, $\mathbb{Q}$ and $\mathbb{Z}$ to denote the set of real numbers, rational numbers and integers respectively. For any integer $n > 0$, we use $[n]$ to denote the set $\{1, \ldots, n\}$. For any integers $a, b (a > b + 1)$, we use $[a, b]$ to denote the set $\{a + 1, \ldots, b - 1\}$. We will use the boldfaced letters (for example $x$) to denote a vector and, denote $x$’s coordinate by $x_i$. We use bold capital letters (for example $B$) to denote a matrix. We will use the notation $[a]$ to denote the closest integer to $a$, and $\left\lfloor a \right\rfloor$ to denote the greatest integer less than equal to $a$. For any vector $v$, we will use the notation $[v]$ to denote the vector representing the floor of each coordinate of the vector. For any $p \in [1, \infty)$, the $\ell_p$ norm on $\mathbb{R}^m$ is defined as follows:

$$\|x\|_p := \left(\sum_{i=1}^{m} |x_i|^p\right)^{1/p},$$

and $\ell_\infty$ norm is defined as

$$\|x\|_\infty := \max\{|x_i|\}.$$  

We will use the following inequality between different $\ell_p$ norms,

$$\text{for any } p \leq q, \ x \in \mathbb{R}^m, \ |x|_q \leq |x|_p \leq m^{\frac{q}{p} - 1} |x|_q.$$  

We will usually drop the subscript and use $\|x\|$ to denote $\|x\|_2$. We often shorthand $p \in [1, \infty) \cup \{\infty\}$ by $p \in [1, \infty]$.

For any $v_1, \ldots, v_i \in \mathbb{R}^m$, we denote by $\text{proj}_{\{v_1, \ldots, v_{i-1}\}} v_i$, the vector formed by projecting $v_i$ orthogonal to the subspace spanned by $v_1, \ldots, v_{i-1}$.

**a) Lattices:** Let $B = \{b_1, \ldots, b_n\}$ be a set of $n$ linearly independent vectors from $\mathbb{R}^m$ for some positive integers $m, n$ with $m \geq n$. The lattice $L$ generated by basis $B$ is defined as follows:

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

Here $n$ is called the rank of the lattice and, $m$ is called the dimension of the lattice. Note that a lattice has infinitely many bases. We often write the basis $B$ as a matrix in $\mathbb{R}^{m \times n}$. For any $p \in [1, \infty]$, we use $\lambda_{\gamma}^p(L)$ to denote the length of a shortest non-zero vector in $\ell_p$ norm,

$$\lambda_{\gamma}^p(L) := \min \{ \|v\|_p : v \in L \text{ and } v \neq 0 \}.$$  

For any vector $t \in \mathbb{R}^m$, we use $\text{dist}_{\gamma}(L, t)$ to denote the distance in $\ell_p$ norm of the target vector from the lattice $L$,

$$\text{dist}_{\gamma}(L, t) := \min \{ \|v - t\|_p : v \in L \}.$$  

For the purpose of computational problems (and hence for the rest of the paper), we restrict our attention to lattices with basis entries in $\mathbb{Q}$.

A. Lattice Problems

In the following, we introduce lattice problems that we study in this paper.

**Definition II.1** ($\gamma$-GapCVP$_p$). For any $\gamma = \gamma(m, n) \geq 1$ and $p \in [1, \infty]$, the $\gamma$-GapCVP$_p$ (Closest Vector Problem) is the decision problem defined as follows: Given a basis $B \in \mathbb{Q}^{m \times n}$ of lattice $L$, a target vector $t \in \mathbb{Q}^m$ and a number $d > 0$, the goal is to distinguish between a YES instance, where $\text{dist}_{\gamma}(L, t) \leq d$ and a NO instance, where $\text{dist}_{\gamma}(L, t) > \gamma d$.

**Definition II.2** ($\gamma$-GapCVP$^\text{IP}$). For any $\gamma = \gamma(m, n) \geq 1$, $p \in [1, \infty]$ and a positive real-valued function $\phi$ on lattice, the $\gamma$-GapCVP$^\text{IP}$ (Closest Vector Problem) is the decision problem defined as follows: Given a basis $B \in \mathbb{Q}^{m \times n}$ of lattice $L$, a target vector $t \in \mathbb{Q}^m$ and a number $d > 0$ with the promise that $\text{dist}_{\gamma}(L, t) \leq \phi(L)$, the goal is to distinguish between a YES instance, where $\text{dist}_{\gamma}(L, t) \leq d$ and a NO instance, where $\text{dist}_{\gamma}(L, t) > \gamma d$.

**Definition II.3** ($\gamma$-GapSVP$_p$). For any $\gamma = \gamma(m, n) \geq 1$ and $p \in [1, \infty]$, the $\gamma$-GapSVP$_p$ (Shortest Vector Problem) is the decision problem defined as follows: Given a basis $B \in \mathbb{Q}^{m \times n}$ of lattice $L$, and a number $d > 0$, the goal is to distinguish between a YES instance, where $\lambda_{\gamma}^p(L) \leq d$ and a NO instance, where $\lambda_{\gamma}^p(L) > \gamma d$.

We omit the parameter $\gamma$ if $\gamma = 1$ and the parameter $p$ if $p = 2$.  

2218
B. LLL Algorithm

For any set of vectors $B = \{b_1, \ldots, b_n\} \in \mathbb{Q}^{m \times n}$, we define the Gram Schmidt Orthogonalization (GSO) of $B$ as $b_1^*, \ldots, b_n^*$, where

$$b_i^* := \text{proj}_{\{b_1^*, \ldots, b_{i-1}^*\}}b_i,$$

and the Gram Schmidt coefficients as

$$\mu_{ij} := \frac{\langle b_i, b_j^* \rangle}{||b_j^*||^2}.$$

Here $\{b_1^*, \ldots, b_{i-1}^*\}^\perp$ denotes the subspace of $\mathbb{R}^m$ which is orthogonal to space formed by $b_1^*, \ldots, b_{i-1}^*$.

**Theorem II.4 ([9]).** For any positive integers $m$ and $n$, there exists an algorithm that given a basis $C = \{c_1, \ldots, c_n\} \in \mathbb{Q}^{m \times n}$ of lattice $\mathcal{L}$, outputs a basis $B = \{b_1, \ldots, b_n\} \in \mathbb{Q}^{m \times n}$ of lattice $\mathcal{L}$ whose GSO vectors and coefficients satisfy the following conditions:

1. $\forall n \geq 1, \sum_{j=1}^{n} \mu_{ij} \leq \frac{1}{2}$,
2. $\forall n \geq 1, \frac{1}{2} \sum_{j=1}^{n} \mu_{ij} \leq \frac{1}{2} ||b_j^*||^2$,
3. $\sum_{j=1}^{n} \mu_{ij} \leq 2^{m/2} v_{\lambda_1}(\mathcal{L})$.

We call the output basis as an LLL-reduced basis. The algorithm runs in time polynomial in the size of the input basis.

C. $k$-SAT and Subset Sum

**Definition II.5 ($k$-SAT).** For any positive integers $k, n, m$, given a CNF formula $\Psi$ of $m$ clauses over $n$ Boolean variables and each clause of $\Psi$ contains at most $k$ literals, the goal is to distinguish between a YES instance, where there exists an assignment that satisfies $\Psi$ and a NO instance, where $\Psi$ does not exist any satisfying assignment.

Impagliazzo and Paturi [50] gave the following two hypotheses about the hardness of $k$-SAT. These two hypotheses are widely used to prove the fine-grained hardness of many computational problems.

**Definition II.6 (Exponential Time Hypothesis (ETH)).** The Exponential Time Hypothesis says the following: for any $k \geq 3$ there exists a constant $\epsilon > 0$ such that $k$-SAT cannot be solved in $2^{\epsilon n}$ time. In particular, $3$-SAT cannot be solved in $2^{O(n)}$ time.

**Definition II.7 (Strong Exponential Time Hypothesis (SETH)).** The Strong Exponential Time Hypothesis says the following: for any constant $\epsilon > 0$, there exists a constant $k$ such that $k$-SAT cannot be solved in $2^{(1-\epsilon) n}$ time.

We also define the Subset-Sum problem.

**Definition II.8 (Subset-Sum).** For any positive integer $n > 0$, given a set of integers $S = \{a_1, \ldots, a_n\}$ and target $t$, the goal is to decide whether there exists a subset of $S$, whose elements sums to $t$.

D. Instance Compression

In this work, we will use two variants of instance compression, probabilistic instance compression with no false negative and probabilistic instance compression that allows both side error. We will use instance compression for probabilistic instance compression with no false negative.

**Definition II.9 (Instance Compression).** A decision problem $P$ is $(f, g, \xi)$ instance compressible with soundness error bound $\xi$ if there exists an $f$-time randomized reduction from any arbitrary instance $x$ of $P$ to some instance $x'$ of size $g$ of decision problem $P'$ such that, if $x$ is a YES instance of $P$ then $x'$ is a YES instance of $P'$, and if $x$ is a NO instance of $P$ then $x'$ is a NO instance of $P'$ with probability at least $1 - \xi$. Here, $f$ and $g$ can be a functions of the witness size and the bit length of $x$.

**Definition II.10 (Probabilistic Instance Compression).** A decision problem $P$ is $(f, g, \xi)$ probabilistic instance compressible with error bound $\xi$ if there exists an $f$-time randomized reduction from any arbitrary instance $x$ of $P$ to some instance $x'$ of size $g$ of decision problem $P'$ such that, if $x$ is a YES instance of $P$ then $x'$ is a YES instance of $P'$ with probability at least $1 - \xi$, and if $x$ is a NO instance of $P$ then $x'$ is a NO instance of $P'$ with probability at least $1 - \xi$. Here, $f$ and $g$ can be a functions of the witness size and the bit length of $x$.

We use the definition of Oracle Communication protocol from [40]. It can also be seen as a generalization of instance compression.

**Definition II.11 (Oracle Communication protocol).** An $(f, g)$ oracle communication protocol for a language $L$ is a communication protocol between two players. The first player is given the input $x$ and has to run in time $f$: the second player is computationally unbounded but not given any part of $x$. At the end of the protocol, the first player outputs YES or NO. It always outputs YES if $x \in L$ and outputs NO with probability at least constant if $x \notin L$. The cost of the protocol $g$ is the number of bits of communication from the first player to the second player. Again, $f$ and $g$ can be a functions of the witness size and the bit length of $x$. The first player is allowed to use randomness but the output by the first player is assumed to be a deterministic function of the communication transcript.4

We will use the following theorem given by the Dell and van Melkebeek [40] about the sparsification of the Satisfiability problem.

**Theorem II.12.** Let $k \geq 3$ and $\epsilon > 0$ a positive real. There is no oracle communication protocol for $k$-SAT of cost $O(n^{k-\epsilon})$ that runs in time $poly(n)$, unless $\text{coNP} \subseteq \text{NP/poly}$.

Our definition of Oracle communication protocol allows the first player to do randomized operations. The Dell and van Melkebeek’s proof still holds for this generalized definition.

4We include randomness in our definition of the oracle communication protocol, since our instance compression algorithms require randomness.
We generalize the impossibility of oracle communication protocol for $k$-SAT within any time $T$.

**Theorem II.13.** Let $k \geq 3$ and $\epsilon > 0$ a positive real. For any $T = T(n) > 0$, there is no oracle communication protocol for $k$-SAT, of cost $O(n^{k-\epsilon})$ that runs in time $T$, unless $\text{coNP} \subseteq \text{NTIME}(\text{poly}(n) \cdot T)/\text{poly}$.

We defer the proof of generalized theorem to Section A.

It is considered unlikely that $\text{coNP} \subseteq \text{NP}/\text{Poly}$, it also implies that polynomial hierarchy collapses to third level.

**Theorem II.14 ( [51]).** If $\text{coNP} \subseteq \text{NP}/\text{Poly}$, then polynomial hierarchy collapses to third level.

We will require the following results from [40].

**Lemma II.15.** [40][Lemma 2] For any integer $k \geq 2$, there is a polynomial time reduction from $\text{OR}(3\text{-SAT})$ to $k$-Clique that maps $t$ tuples of instances of bitlength $n$ each to an instance of $O(n \cdot \text{max}(n,k^{1/k-o(1)}))$ vertices.

**Lemma II.16.** [40][Lemma 5] For any $k \geq 3$, there is a polynomial time reduction from $k$-Vertex Cover to $k$-SAT that maps a $k$-uniform hypergraph on $n$ vertices to $k$-CNF formula on $O(n)$ variables.

For the definitions of Hypergraph problems, $k$-Vertex Cover and $k$-Clique, we refer the reader to the preliminaries section of [40]. Note that there is a trivial reduction from $k$-Clique to $k$-Vertex Cover, which also preserves the number of vertices.

III. CVP INNER PRODUCT, CVP MULTI VECTOR PRODUCT: VARIANTS OF CVP

We introduce the following new variant of the problems, $\gamma$-GapCVP$^\text{IP}$, where the input lattice and the target vectors are not given directly but as pairwise inner products. More precisely, the input is $\alpha_{i,j} \in \mathbb{Q}$ for $1 \leq j < i \leq n$, and $\beta_i \in \mathbb{Q}$ for $1 \leq i \leq n+1$ such that the corresponding lattice and the target vector satisfies the following. For $1 \leq j \leq i \leq n$

$$\alpha_{i,j} = (b_i, b_j),$$

$$\beta_i = (b_i, t),$$

and

$$\beta_{n+1} = (t, t).$$

Note that the square of the $\ell_2$ distance between any integer combinations of the basis vectors and the target is an integer linear combination of the $(n+1)^p$ integers $\alpha_{i_1, i_2, \ldots, i_p}$'s. Hence, for even $p$, we can efficiently reduce $\gamma$-GapCVP$^\text{IP}$ to $\gamma$-GapCVP$^\text{mvp}$.

We define variants of CVP$^\text{mvp}$, $(r,q)$-CVP$^\text{mvp}$ and $r$-CVP$^\text{mvp}$.

**Definition III.2 ((r,q)-CVP$^\text{mvp}$).** For any positive integers $q = q(m,n)$ and $r = r(m,n)$, the $(r,q)$-CVP$^\text{mvp}$ is the promise problem defined as follows: Given $\text{mvp}$ form of a basis $B \in \mathbb{Q}^{m \times n}$ and a target vector $t \in \mathbb{Q}^m$ and a number $d > 0$, the goal is to distinguish between a ‘YES’ instance, where $\exists z \in [-r,r]^n \|Bz - t\|_p^q \mod q \leq d$ and a ‘NO’ instance, where $\forall z \in [-r,r]^n \|Bz - t\|_p^q \mod q > d$.

When $p = 2$ we will also denote $(r,q)$-CVP$^\text{mvp}$ by $(r,q)$-CVP$^\text{IP}$.

**Definition III.3 (r-CVP$^\text{mvp}$).** For any positive integer $r = r(m,n)$, the $(r)$-CVP$^\text{mvp}$ is the promise problem defined as follows: Given $\text{mvp}$ form of a basis $B \in \mathbb{Q}^{m \times n}$ and a target vector $t \in \mathbb{Q}^m$ and a number $d > 0$, the goal is to distinguish between a ‘YES’ instance, where $\exists z \in [-r,r]^n \|Bz - t\|_p^r \mod q \leq d$ and a ‘NO’ instance, where $\forall z \in [-r,r]^n \|Bz - t\|_p^r > d$. 

The lemma follows from the above equation. 

Notice that the above lemma implies that for even $p$, $p^\text{th}$ power of the $\ell_p$ distance between any integer combinations of the basis vectors and the target is an integer linear combination of the $(n+1)^p$ integers $\alpha_{i_1, i_2, \ldots, i_p}$'s. Hence, for even $p$, we can efficiently reduce $\gamma$-GapCVP$^\text{IP}$ to $\gamma$-GapCVP$^\text{mvp}$.
When \( p = 2 \) we will also denote \( r\text{-CVVP} \) by \( r\text{-CVP} \).

IV. INSTANCE COMPRESSION FOR ALMOST EXACT-CVP

In this section, we present an instance compression algorithm for CVP in the Euclidean norm. We show that, for any \( c > 0 \), and for any \( \varepsilon \geq 2^{-n} \), we can reduce an instance of \((1 + \varepsilon)\text{-GapCVP} \) with rank \( n \), and dimension \( m \) to an instance of \((r, q)\text{-CVVP} \), such that the size of the \((r, q)\text{-CVVP} \) instance is \( n^{c+} \) for some \( c > 0 \). The reduction takes time polynomial in the input size.

In the following lemma, we show using Babai’s algorithm [52] that without loss of generality, we may assume that given a CVP instance \((B, t)\), the coefficient vector of the vector closest to \( t \) has all co-ordinates bounded by \( 2^{O(n^2)} \).

**Lemma IV.1** (Bounding the coefficients of closest lattice vector). For any positive integers \( m, n \), there exists an algorithm that given a basis \( B \in \mathbb{Q}^{m \times n} \) and target vector \( t \in \mathbb{Q}^m \) of total bitlength \( \eta \) as output, outputs a basis \( C \in \mathbb{Q}^{m \times n} \), a target vector \( t' \in \mathbb{Q}^m \) such that \( t' = t + \mathcal{L}(B), \mathcal{L}(C) = \mathcal{L}(B) \), the total bitlength of \( C, t' \) is at most \( \text{poly}(\eta, m, n) \), and for any \( z \in \mathbb{Z}^n \) if

\[
||Cz - t'||_2 = \text{dist}_2(\mathcal{L}(C), t')
\]

then

\[
||z||_\infty < 2^{n^2}.
\]

The algorithm runs in \( \text{poly}(\eta, n, m) \) time and requires \( \text{poly}(\eta, n, m) \) space. Furthermore, \( C \) is a LLL-reduced basis of \( B \).

**Proof.** The algorithm does the following. The algorithm runs the LLL algorithm (from Theorem II.4) on basis \( B \), and gets an LLL reduced basis \( C = [c_1 \ldots c_n] \in \mathbb{Q}^{m \times n} \) as output. Then the algorithm computes the Gram-Schmidt Orthogonalization(GSO) of basis \( C \) as \( C^* = [c_1^* \ldots c_n^*] \). The algorithm then computes \( x_i = (c_i^* v_1, \ldots c_i^* v_m) \in \mathbb{Q}^m \) for all \( i \in [n] \). Finally, the algorithm computes

\[
t' = t - \sum_{i=1}^n w_i c_i, \text{ where } w_n = \lfloor x_n \rfloor,
\]

\[
\forall i < n, w_i = \left| x_i + \sum_{k=1}^n w_k k x_i \right|.
\]

The algorithm then outputs \( C, t' \).

The algorithm runs in \( \text{poly}(\eta, n, m) \) time and all vectors \( c_i \) for \( 1 \leq i \leq n \), and \( t' \) can be represented in \( \text{poly}(\eta, n, m) \) bits. For more details on the computation time of LLL basis, we refer the reader to [53]. We will now prove that \( t' \in t + \mathcal{L}(B) \) and the coefficients of the closest vector in \( \mathcal{L}(C) \) to \( t' \) are bounded.

As \( C \) is an LLL reduced basis, from Theorem II.4 we get the following conditions:

\[
\forall i \in [n - 1], ||c_i^*||^2 \leq 2||c_{i+1}^*||^2 \tag{1}
\]

and

\[
\forall i > j, |\mu_{i,j}| \leq \frac{1}{2} \tag{2}
\]

using Equations (1) and (2). Notice that we can represent the target vector \( t' \) as \( t' = \sum_{i=1}^n x_i^* c_i^* + c_{n+1}^* \), where \( c_{n+1}^* \) lies in a vector space orthogonal to \( c_1^*, \ldots, c_n^* \) in \( \mathbb{R}^m \). We emphasize here that \( c_{n+1}^* \) could be \( 0 \) if the target vector lies in the linear span of the basis vectors. We note that the coefficients \( w_i \) are chosen so that, we get that \( \forall i \leq n, x_i^* \in (-1/2, 1/2) \). Also, \( t - t' \in \mathcal{L}(C) \) as \( w_i \)'s are integers, and

\[
\text{dist}(\mathcal{L}(B), t) = \text{dist}(\mathcal{L}(C), t) = \text{dist}(\mathcal{L}(C), t') \leq ||t'|| \tag{3}
\]

Let \( v = \sum_{i=1}^n z_i c_i \) be a closest lattice vector to target \( t' \). We prove by induction on \( i \) that \( |z_{n-i+1}| \leq 2^{n-i} \). Without loss of generality, we assume \( c_{n+1}^* = 0 \). This does not change the closest vector in the lattice since \( c_{n+1}^* \) is orthogonal to the lattice.

We first show that \( z_n \) is bounded. To see this, note that

\[
|z_n - x_n^*||c_n^*|| \leq ||v - t'||^2 \leq \left( \frac{1}{2} \sum_{i=1}^n ||c_i^*||^2 \right)^{1/2}
\]

\[
\leq \frac{1}{2} ||c_n^*||^2 (2^{n-1} + 2^{n-2} + \ldots + 1)
\]

\[
< 2^{n-1} ||c_n^*||^2.
\]

Thus, \( |z_n| \leq \frac{1}{2} + 2^{n-1} < 2^n \), thereby proving the base case \( i = 1 \). We now assume that \( |z_{n-j+1}| < 2^{n-j} \) for \( j \leq i \).

For any fixed \( z_n, \ldots, z_{n-i+1} \), we now bound \( |z_{n-i}| \) corresponding to the vector \( v \) that minimizes \( ||v - t'|| \). Let \( u = \sum_{i=n-i+1}^n z_i c_i \). For any vector \( x \), let \( \pi(x) \) denote the projection of \( x \) in the linear span of \( c_1^*, \ldots, c_{n-i}^* \). Note that the projection of \( v - t' \) in the linear span of \( c_1^*, \ldots, c_{n-i}^* \) is the same as that of \( u - t' \). Thus, \( ||\pi(v - t')|| \leq ||\pi(u - t')|| \).

This implies that

\[
\left| z_{n-i} + \left( \sum_{k=n-i+1}^n \mu_{n-i,k} z_k \right) - x_{n-i} \right|^2 ||c_{n-i}^*||^2 \leq ||\pi(v - t')||^2
\]

\[
\leq \sum_{k=1}^{n-i} \left( \sum_{k=n-i+1}^n \mu_{k,k} z_k - x_k \right)^2 ||c_k^*||^2
\]

\[
< \sum_{k=1}^{n-i} \left( \frac{2^{n} + 2^{2n} + \ldots + 2^{n-i} + 1}{2} \right) ||c_k^*||^2
\]

\[
\leq \sum_{k=1}^{n-i} \left( 2^n \right) ||c_k^*||^2
\]

\[
< \left( 2^n \right)^2 \cdot (1 + 2 + \ldots + 2^{n-i-1}) ||c_{n-i}^*||^2
\]

\[
< \left( 2^n \right)^2 \cdot 2^n \cdot ||c_{n-i}^*||^2.
\]
Thus, by triangle inequality,
\[ |z_{n-i}| < \frac{|z_n| + \cdots + |z_{n-i+1}|}{2} + \frac{1}{2} + 2^{n+n/2} \]
\[ \leq 2^n + \cdots + 2^{n-i} \]
\[ \leq 2^n + 2^{n+n/2} \]
\[ \leq 2^{i+1}n. \]

In the following theorem, we give an instance compression algorithm for CVP2. We also show a instance compression with better parameters for CVP2 in Theorem VI.2 using a Theorem from [11].

**Theorem IV.2.** For any positive integers \( m, n, \) and constant \( c_1 \in \mathbb{R}^+ \), given a \((1 + 2^{-n^2})\)-GapCVP\((B, t, d)\) instance where \( B \in \mathbb{Q}^{m \times n} \) is a basis of a lattice \( L \), target \( t \in \mathbb{Q}^m \) and \( d > 0 \). The bit-length of the input is at most \( \eta \). There exists a poly\((n, m, \eta)\) time probabilistic algorithm that reduces it to a \((r, q)\)-CVP\(^p\) instance of size at most \( O(n^{2}\log^2(n+m+T)) \) for constant \( c_2 = \max\{c_1 + 3, 5\} \).

Furthermore, ‘YES’ instance always reduces to ‘YES’ instance and ‘NO’ instance reduces to ‘NO’ instance with at least \( 1 - 2^{-n^3} \) probability i.e. the reduction does not give false negative.

**Proof.** Let \( \gamma = 1 + 2^{-n^3} \) and \( r = 2n^2 \). We are given a basis \( B \in \mathbb{Q}^{m \times n} \), target \( t \in \mathbb{Q}^m \) and a distance \( d \) with a promise that either \( \text{dist}(B, t) > \gamma d \) or \( \text{dist}(B, t) \leq d \). From Lemma IV.1, we can assume that we are given a \( \gamma \)-GapCVP\((C, \tilde{t}, d)\) instance such that \( \tilde{t} \in t + L(B) \), \( L(C) = L(B) \) and for all \( z \in \mathbb{Z}^n \), if
\[ \|Cz - \tilde{t}\| = \text{dist}(C, \tilde{t}) \]
then
\[ z \in [-r, r]^n. \]

As \( C \in \mathbb{Q}^{m \times n} \) and \( \tilde{t} \in \mathbb{Q}^m \), we can scale the basis vector to make all the coordinates integers and it will not even increase the bit-representations. So, without loss of generality, we assume that \( C \in \mathbb{Z}^{m \times n} \) and \( \tilde{t} \in \mathbb{Z}^n \). Let \( c := \max\{c_1 + 1, 3\} \).

First, we reduce the problem to one where in YES instance the distance of target from lattice is at most \( 2^{4n^3} \). Let \( n' \) be an integer such that \( 2^{4n^3+1} > d \geq 2^{n^3} \). Let’s assume that \( n' > 4n^3 \).

Later, we will analyze the case when \( n' \leq 4n^3 \). We remove the \( n' - 4n^3 \) least significant bits of basis vectors \( c_i \)'s and target vector \( t \). Consider \( C' = \{c'_1, \ldots, c'_n\} \) and \( t' \) be the vectors after removing the least significant bits i.e.
\[ \forall i \in [n], c'_i = \left\lfloor \frac{1}{2^{n-4n^3}} \cdot c_i \right\rfloor \]
and \( t' = \left\lfloor \frac{1}{2^{n-4n^3}} \cdot t \right\rfloor \).

We define a new measure of distance from the target, where we only focus on the distance of the target vector from the integer combination of basis vectors whose coefficients are less than \( 2^{n^3} \).
\[ \text{dist}^*(C', t') := \min_{z \in [-r, r]^n} \{\|C'z - t'\|\}. \]

We know that closest lattice vector of target vector \( \tilde{t} \) in lattice \( L(C) \) is of form \( v = \sum_{i=1}^n z_i c_i \) where \( \forall i \leq n, |z_i| < r \). Therefore, we get
\[ \left| \left( 2^{n'-4n^3} \cdot \text{dist}^*(C', t') \right) - \text{dist}(L(C), \tilde{t}) \right| \]
\[ \leq \max_{z \in [-r, r]^n} \{\left| (2^{r} \cdot \|C'z - t'\|) - \|Cz - \tilde{t}\| \right| \}
\[ \leq m \cdot n \cdot 2^{n'-4n^3} \cdot 2^{n^3} \]
\[ < 2^{n'-2n^3}. \]

Now, we introduce it to a CVP instance with explicit bounds on coefficients. Let \( q \) be a prime chosen uniformly at random from \( \left[ 2^{10n^3+\alpha} \cdot 2^{20n^3+\alpha} \right] \) where \( \alpha = \log^2(n + m + \eta) \). Let
\[ \forall i \leq n, h_i := c'_i \mod q. H = \{h_1, \ldots, h_n\} \]
and \( t'' = t' \mod q \).

We will show that if \( \text{dist}^*(C', t') \leq d' \) then there exists a vector \( z \in [-r, r]^n \) such that \( \|Hz - t''\|^2 \mod q \leq (d')^2 \).

Otherwise (when \( \text{dist}^*(C', t') > d' \)), for all \( z \in [-r, r]^n \), \( \|Hz - t''\|^2 \mod q > (d')^2 \).

First, let’s assume that \( \text{dist}^*(C', t') < d' \). Let \( z \in \mathbb{Z}^n \) be a vector such that \( \|C'z - t''\| = \text{dist}^*(C', t') \) and \( \|z\| < 2^{n^3} \).

From the definition of \( \text{dist}^* \) there exist such a vector \( z \). Therefore, we get \( \|Hz - (t'' \mod q)\|^2 \mod q \leq \|C'z - (t'' \mod q)\|^2 \mod q \leq (d')^2 \).

Now, let’s assume that \( \text{dist}^*(C', t') > d' \) i.e. for all \( z \in [-r, r]^n \), \( \|C'z - t''\| > d' \). From a lower bound on the prime number theorem [54] we have that number of primes in range \( \left[ 2^{10n^3+\alpha} \cdot 2^{20n^3+\alpha} \right] \) is at least
\[ \frac{2^{20n^3+\alpha}}{2(20n^3+\alpha)} \cdot \log 2 = \frac{2^{10n^3+\alpha}}{2^{10n^3+\alpha}/2}. \]

Also, for any fixed \( z \in [-r, r]^n \) and \( w \leq (d')^2 \),
\[ \left\| \sum_{i=1}^n z_i c'_i - t' \right\|_p^p - w \leq m \cdot (n+1) \cdot 2^{r} \cdot 2^{\delta} \cdot (d')^2 \leq 2^{\text{poly}(n, m, \eta)}, \]
where
\[ \delta = \max \{\log |c'_1|, \log |c'_2|, \ldots, \log |c'_m|, \log |t'_1|, \ldots, \log |t'_m|\} \]
\[ \leq \text{poly}(n, m, \eta). \]
Hence, there are at most \( \text{poly}(n, m, \eta) \) distinct primes that divide \( \| \sum_{i=1}^n z_i c_i - t' \|_p - w \). Hence, with probability, at most
\[
\frac{\text{poly}(n, m, \eta)}{2^{19n^c + n/2}} \leq 2^{-19n^c},
\]
the prime \( q \) is such that \( \| C z - t' \|_p - w = 0 \) mod \( q \). Therefore, by union bound over all \( w \leq (d')^2 \) and \( z \in [-r, r]^n \), for uniformly sampled prime \( q \) we get
\[
\Pr \left[ \min_{\| z \|_\infty < 2^{n^c}} \{ \| (C z - t') \| \} \leq (d')^2 \right] \leq 2(n^c)n \cdot 2^{4n^c+1} \cdot 2^{-19n^c} < 2^{-13n^c}. \tag{4}
\]

It implies that with overwhelming probability, for all \( z \in [-r, r]^n \), \( \| H z - t' \|_p^2 \) mod \( q = \| (C z - t') \|^2 \) mod \( q > (d')^2 \).

Now, we construct \((r, q, c)\)-CVP instance of \((H, t')\) by storing the inner products of basis \( H \) and target \( t' \). We reduce the \( \gamma \)-GapCVP instance \((B, t, d)\) to \((r, q, c)\)-CVP instance, where given inner product form of \((H, t')\), integers \( d'' = (d')^2 \), \( r \) and \( q \), the goal is to distinguish between a YES instance where there exists a \( z \in [-r, r]^n \) for which \( \| H z - t' \|_p^2 \) mod \( q \) is at most \( d'' \) and a NO instance where for all vector \( z \in [-r, r]^n \), \( \| H z - t' \|_p^2 \) mod \( q \) is greater than \( d'' \). Notice that, in this reduction we don’t get false negative. The instance size is at most \((n + 1)^2 \cdot \log (m \cdot q^2) = \mathcal{O}((n^c + 2) \log^2 (n + m + \eta))\) bits because each coordinate of basis \( H \) and target \( t' \) is less than \( q \). It completes the proof.

\[ \square \]

V. INSTANCE COMPRESSION FOR ALL EVEN NORMS

In this section, we present an instance compression algorithm for CVP\(p\) for all even \( p \). We show this for any constant \( c > 0 \), \((1 + \exp(-n^c))\) approximation of CVP\(p\) problem with additional promise that distance of the target from lattice is bounded by \( \exp(n^c) \cdot \lambda_1^{(p)} \). We will first show that, we can bound the coefficient of the closest lattice vector by using this additional promise.

Lemma V.1 (Bounding the coefficients of CVP\(p\)). For any \( m, n \in \mathbb{Z}^+ \), (efficiently computable) \( \tau = \tau(m, n) > 0 \), and \( p \in [1, \infty] \), there exists a randomized algorithm that given an instance of CVP\(p\) basis \( B \in \mathbb{Q}^{m \times n} \) and target vector \( t \in \mathbb{Q}^m \) of bilineat \( \eta \) as input, outputs a basis \( C' \in \mathbb{Q}^{m \times n} \) and target vector \( t' \in \mathbb{Q}^m \) such that \( \lambda_1(C') = \lambda_1(B) \), \( t-t' \in \mathcal{L}(B) \) and for all vector \( z \in \mathbb{Z}^n \) which satisfies
\[
\| C z - t' \|_p = \text{dist}_p(L(C'), t'),
\]
we have
\[
\| z \|_\infty < \tau \cdot m \cdot 2^{3n^c/2}.
\]

The algorithm runs in \( \text{poly}(n, m, \eta) \) time and requires \( \text{poly}(n, m, \eta) \) space.

Proof. We are given a basis \( B \in \mathbb{Q}^{m \times n} \) and target vector \( t \in \mathbb{Q}^m \) which satisfy
\[
\text{dist}_p(L(B), t) \leq \tau \lambda_1^{(p)}(L(B)) \tag{5}
\]
The algorithm runs the LLL algorithm (from Theorem II.4) on basis \( B \), and gets an LLL reduced basis \( C = [c_1 \ldots c_n] \in \mathbb{Q}^{m \times n} \) as output. Then the algorithm computes Gram-Schmidt Orthogonalization (GSO) of basis \( C \) as \( C' = [c_1', \ldots, c_n'] \).

The algorithm then computes \( x_i = \langle t, c_i' \rangle \in \mathbb{Q} \) for all \( i \in [n] \). Finally, the algorithm computes
\[
t' = t - \sum_{i=1}^n w_i c_i, \quad \text{where } w_i = \langle x_i, c_i \rangle.
\]
The algorithm then outputs \( C, t' \). The algorithm runs in \( \text{poly}(n, m, \eta) \) time and all vectors \( c_i \) for \( 1 \leq i \leq n \), and \( t' \) can be represented in \( \text{poly}(n, m, \eta) \) bits. For more details on this, we refer the reader to [53]. We will now prove that \( \text{dist}_p(L(B), t) = \text{dist}_p(L(C, t')) \) and the coefficients of the closest lattice vector with respect to basis \( C \) to target \( t' \) are bounded.

As \( C \) is a LLL reduced basis, from Theorem II.4 we get the following conditions:
\[
\forall i \in [n-1], \| c_i^* \|_2^2 < 2\| c_{i+1}^* \|_2^2 \tag{6}
\]
and
\[
c_i = c_i^* + \sum_{j=1}^{i-1} \mu_{ij} c_j^* \quad \text{and} \quad \forall j > i \mid \mu_{ij} \mid \leq \frac{1}{2}. \tag{7}
\]

Notice that we can represent the target vector \( t' \) as \( t' = \sum_{i=1}^n x_i^* c_i^* + c_{n+1}^* \), where \( c_{n+1}^* \) lies in a vector space orthogonal to \( c_1^*, \ldots, c_n^* \) in \( \mathbb{R}^m \). We emphasize here that \( c_{n+1}^* \) could be \( 0 \) if the target vector lies in the linear span of the basis vectors. We note that the coefficients \( w_i \) are chosen so that, we get that \( \forall i \leq n, x_i^* \in (-1/2, 1/2] \). Also, \( t-t' \in \mathcal{L}(C) = \mathcal{L}(B) \) as \( w_i \)'s are integers, i.e. \( \text{dist}_p(L(C), t') = \text{dist}_p(L(B), t) \).

Let \( C' = L(C) \). As \( c_1^* \) is a non-zero lattice vector, we know that \( \lambda_1(C') \leq \| c_1^* \|_p \). From Equation 5, we get that
\[
\text{dist}_p(C', t') \leq (1 + \tau)\lambda_1^{(p)}(C') \leq \tau \cdot \| c_1^* \|_p \tag{8}
\]
Let \( v = \sum_{i=1}^n x_i c_i^* \) be a closest lattice vector to target vector \( t' \) in \( \ell_p \) norm. We prove by induction on \( i \) that \( |z_{n-i+1}| \leq \tau \cdot m \cdot 2^{2n^c/2+1} \). For interchanging between different \( \ell_p \) norms, we will be using the fact that for any \( u \in \mathbb{R}^m \) and \( q \geq p \geq 1 \), \( \| u \|_q \leq m^{1/p-1/q} \| u \|_p \). We rewrite \( v = \sum_{i=1}^n y_i c_i^* \) and get
\[
\| v-t' \|_2^2 = \sum_{i=1}^n (y_i - x_i^*)^2 \| c_i^* \|_2^2 + \| c_{n+1}^* \|_2^2 \leq m (\tau \cdot \| c_1^* \|_p)^2 \leq m^2 (\tau \cdot \| c_1 \|_2)^2,
\]
\[\square\]
where second last inequality follows from Equation (8). Therefore we get
\[ \forall i \in [n], |y_i| \leq \frac{\tau m \cdot \|c_1\|_2}{\|c_1\|_2^2} + \frac{1}{2} \leq \tau m \cdot \frac{2^{i-1}}{2} + \frac{1}{2} \leq \tau m \cdot 2^{n/2}, \] using Equation (6). By using the fact that \( z_n = y_n \) we get \( |z_n| < \tau m \cdot 2^{n/2} \), thereby proving the base case \( i = 1 \). We now assume that \( |z_{n-i+1}| < \tau m \cdot 2^{n/2+j} \) for \( j < i \). By using Equation (7), we get
\[ |z_{n-i+1}| \leq |y_{n-i+1}| + \sum_{k > n-i+1} |\mu_k n_{n-i+1} \cdot z_k| \leq |y_{n-i+1}| + \frac{1}{2} \sum_{k > n-i+1} |z_k| \leq \tau m \cdot 2^{n/2} + \frac{\tau m 2^{n/2}}{2} (2 + 2^2 + 2^3 + \ldots + 2^{i-1}) < \tau m \cdot 2^{n/2+i}. \]

Hence, the lemma follows. \( \square \)

Now, we show an instance compression algorithm for \( \text{CVP}_p \) when \( p \) is an even integer. It follows the proof technique of Theorem 4.2. For completeness, we also give a proof here. We also show a instance compression with better parameters for \( \text{CVP}_p \) in Theorem 4.3 using a Theorem from [11].

**Theorem V.2.** For any \( m, n \in \mathbb{Z}^+ \), \( p = 2^k \) and constant \( c_1 \in \mathbb{R}^+ \), given a \((1 + 2^{-p})\)-Gap \( \text{CVP}^p \) \((B, t, d)\) instance where \( B \in \mathbb{Z}^{m \times n} \) is basis of lattice \( L \), target \( t \in \mathbb{Q}^m \), \( d > 0 \) and \( \phi = 2^n \cdot \lambda_1(B) \). The bit-length of the input is at most \( \eta \). There exists a poly\((n, m, \eta)\) time randomized algorithm that reduces it to a \((r, q)\)-Gap \( \text{CVP}^p \) instance of size \( O(p \cdot n^{p+2}) \cdot \log(n + m + \eta) \) for constant \( c_2 = \max\{c_1, 2, 3\} \).

Furthermore, ‘YES’ instance always reduces to ‘YES’ instance and ‘NO’ instance reduces to ‘NO’ instance with at least \( 1 - 2^{-n^2} \) probability i.e. the reduction does not give false negatives.

**Proof.** Let \( \gamma := 1 + 2^{-n^2} \). We are given a basis \( B \in \mathbb{Q}^{m \times n} \), target \( t \in \mathbb{Q}^m \) and a distance \( d \geq 0 \) with a promise that either \( \text{dist}_p(B, t) \leq d \) or \( \text{dist}_p(B, t) > \gamma d \). We are also given a guarantee that \( \text{dist}_p(B, t) \leq 2^{\gamma n} \lambda_1(B) = (\gamma - 1)^{-1} \lambda_1(B) \). We first apply the algorithm from Lemma V.1 with \( \tau = \frac{1}{\gamma - 1} \), and get basis \( C \in \mathbb{Q}^{m \times n} \) and target vector \( \tilde{t} \in \mathbb{Q}^m \) which satisfies \( L(C) = L(B) \), \( \tilde{t} - t \in L(B) \), and for all vector \( z \in \mathbb{Z}^n \) which satisfies
\[ \|Cz - \tilde{t}\|_p = \text{dist}_p(L(C), \tilde{t}), \]
we have
\[ \|z\|_\infty < \frac{1}{\gamma - 1} \cdot m \cdot 2^{3n/2} < 2^{n^2+1}. \] (11)

As \( C \in \mathbb{Q}^{m \times n} \) and \( \tilde{t} \in \mathbb{Q}^m \), we can scale the basis vector to make all the coordinates integers and it will not even increase the bit-representations. So, without loss of generality, we assume that \( C \in \mathbb{Z}^{m \times n} \) and \( \tilde{t} \in \mathbb{Z}^m \). Let \( c := \max\{c_1, 1, 2\} \).

We define a new measure of distance from the target, where we only focus on the distance of the target vector from the integer combination of basis vector whose coefficients are less than \( 2^n \).
\[ \text{dist}_p^*(C, \tilde{t}) := \min_{z \in \mathbb{Z}^n} \|Cz - \tilde{t}\|_p. \]

Hence, if \( \text{dist}_p(B, t) \leq d \) then \( \text{dist}_p^*(C, \tilde{t}) \leq d \), otherwise \( \text{dist}_p(B, t) > \gamma d \) then \( \text{dist}_p^*(C, \tilde{t}) > \gamma d \).

We first reduce the problem to one where the distance from the target is at most \( 2^{n^2} \). Let \( n' \) be an integer such that \( 2^{n'+1} > d \geq 2^{n'} \). Let’s assume that \( n' > 4n^2 \). Later, we will analyze the case when \( n' \leq 4n^2 \). We remove the \( n' - 4n^2 \) least significant bits of the entries of basis vectors and target vector. Consider the basis \( C' = \{c'_1, \ldots, c'_n\} \) and \( t' \) be the lattice and target vector after removing the least significant bits i.e.
\[ \forall i \in [n], c'_i = \left[ \frac{1}{2^{n'-4n^2}} \cdot c_i \right] \text{ and } t' = \left[ \frac{1}{2^{n'-4n^2}} \cdot \tilde{t} \right]. \]

Therefore, we get
\[ \left( 2^{n'-4n^2} \cdot \text{dist}_p^*(C', t') \right) - \text{dist}_p^*(L(C), \tilde{t}) \leq m \cdot (n+1) \cdot 2^{n'-4n^2} \cdot 2^{n'-4n^2} \leq 2^{n'-2n^2}. \]

Hence, if \( \text{dist}_p(C, \tilde{t}) \leq d \) then \( \text{dist}_p^*(C', t') \leq \frac{d+2^{n'-2n^2}}{2^{n'-2n^2}} \), and when \( \text{dist}_p(C, \tilde{t}) > \gamma d \) then \( \text{dist}_p^*(C', t') \) is greater than \( \frac{\gamma d - 2^{n'-2n^2} - 1}{d - 2^{n'-2n^2}}. \mathrm{ Let } d' := \frac{d+2^{n'-2n^2}}{2^{n'-2n^2}} < 2^{n'-2n^2} \) and our choice of \( c \) implies that \( \gamma d - 2^{n'-2n^2} - 1 \)

When \( n' \leq 4n^2 \), we take \( C' = C, t' = \tilde{t} \) and \( d' = d < 2^{n'+1} \). Hence we get basis \( C' \in \mathbb{Z}^{m \times n} \), target \( t' \in \mathbb{Z}^m \) and number \( d' < 2^{n'+2} \) such that if \( \text{dist}(B, t) \leq d \) then \( \text{dist}_p^*(C', t') \leq d' \), and if \( \text{dist}(B, t) > d \) then \( \text{dist}_p^*(C', t') > d' \).

Now, we reduce to CVP instance with explicit bounds on basis and target vectors coordinate. Let \( q \) be a prime chosen uniformly at random from \( \left[ 2^{10n^{4-n^2}+\alpha}, 2^{20n^{4-n^2}+\alpha} \right] \), where \( \alpha = \log^2(n + m + \eta) \). Let \( \forall i \leq n, h_i := c'_i \mod q \) and \( H = \{h_1, \ldots, h_n\} \)

Let \( r := 2^n \). We will show that if \( \text{dist}_p^*(C', t') \leq d' \) then there exist a vector \( z \in [-r, r]^n \) such that \( \|Hz - t'\|_p^p \mod q \leq (d')^p \). Otherwise (when \( \text{dist}_p^*(C', t') > d' \)), for all \( z \in [-r, r]^n \), \( \|Hz - t'\|_p^p \mod q > (d')^p \).

First, let’s assume that \( \text{dist}_p^*(C', \tilde{t}) \leq d' \). Let \( z \in \mathbb{Z}^n \) be a vector such that \( \|Cz - t'\|_p = \text{dist}_p^*(C', t') \) and \( \|z\|_\infty < \mathcal{r} \). From the definition of \( \text{dist}_p^* \) there exist such a vector \( z \). Hence we get
\[ \|Hz - (t')^p\|_p^p \mod q = \|Cz - t'\|_p^p \mod q \leq (d')^p. \]

Now, let’s assume that \( \text{dist}_p^*(C', \tilde{t}) > d' \) i.e. for all \( z \in [-r, r]^n \), \( \|Cz - t'\|_p > d' \). From a lower bound on the prime
number theorem \cite{54} we know that number of primes in range
\[ [2^{10n^{1+\alpha}}, 2^{20n^{1+\alpha}}] \] is at least
\[ 2^{20n^{1+\alpha}} \cdot \log 2 \cdot \frac{1}{2 \cdot (20n^{1+\alpha}) - \alpha} \geq 2^{10n^{1+\alpha/2}}. \]
Also, for any fixed \( z \in [-r, r]^n \) and \( w \leq (d')^p \),
\[ \left\| \sum_{i=1}^{n} z_i c_i - t' \right\|_p^w \leq m \cdot (n+1) \cdot 2^d \cdot (d')^p \leq 2^{\text{poly}(n, m, T)}, \]
where
\[ \delta = \max \{ \log |c_1|, \log |c_2|, \ldots, \log |c_m|, \log |t_1|, \ldots, \log |t_n| \} \]
is bounded by \( \text{poly}(n, m, \eta) \). Hence, there are at most \( \text{poly}(n, m, \eta) \) distinct primes that divide
\[ \left\| \sum_{i=1}^{n} z_i c_i - t' \right\|_p^w \]. Hence, with probability, at most
\[ \text{poly}(n, m, \eta) \cdot 2^{10n^{1+\alpha/2}} \leq 2^{-10n^{1+\alpha}} \]
the prime \( p \) is such that \( \left| C' z - t' \right|_p^w \leq w \mod p \).

By further union bound over all \( z \in [-r, r]^n \) and \( w \leq (d')^p \), we get for uniformly sampled \( q \),
\[ \Pr \left[ \left\| C' z - t' \right\|_p^w \mod q \leq (d')^p \right] \]
\[ \leq 2^{(n^2) + 2(4n+1)p} \cdot 2^{-10n^{1+\alpha}} \leq 2^{-3n^{1+\alpha}}. \]

Last inequality uses the assumption that \( p = o(n) \). It implies that
with overwhelming probability, for all \( z \in [-r, r]^n \),
\[ \| H z - t'' \|_p^w \mod q = \| C' z - t' \|_p^w \mod q > (d')^p \]
Now, we construct a \((r, q), \text{CVP}^\text{mp}\) instance of \((H, t'')\) by storing \text{mpv} form of basis \( H \) and target \( t'' \). Therefore, we get a randomized reduction from the \( \gamma\)-GapCVP\(_p\) instance \((B, t, d)\) to \((r, q), \text{CVP}^\text{mp}\) where a \( \gamma\)-GapCVP\(_p\) instance of \( \gamma\)-GapCVP always reduces to a \( \gamma\)-YES instance and a \( \gamma\)-NO instance of \( \gamma\)-GapCVP\(_p\) reduces to a \( \gamma\)-NO instance with probability \( 1 - 2^{-10n^{1+\alpha}} \).

In \((r, q), \text{CVP}^\text{mp}\) given \text{mpv} form of \((H, t'')\), with integers \( d'' = (d')^p, r \) and \( q \), the goal is to distinguish between a \( \gamma\)-YES instance where there exists a \( z \in [-r, r]^n \) for which
\[ \| H z - t'' \|_p^w \mod q \leq m \cdot (n+1)^{1+p} \cdot \log(m \cdot q^p) = O(p \cdot n^{p+1} \cdot \log(n + m + \eta)). \]
It completes the proof.

\[ \square \]

A. Instance compression for SVP

We show a instance compression algorithm for SVP\(_p\) for even \( p \). For SVP we don’t require any additional promise. For this, we will use a reduction from \( \gamma\)-SVP\(_p\) to \( \gamma\)-CVP\(_p\) from [1].

**Theorem V.3.** For any \( m, n \in \mathbb{Z}_+ \), \( p \in 2\mathbb{Z}_+ \) and constant \( c_1 \geq 1 \), given a \((1 + 2^{-n^{1+\alpha}})\)-GapSVP\(_p\)(\(B, d)\) instance where \( B \in \mathbb{Q}^{m \times n} \) is basis of lattice \( \mathcal{L} \), and \( d > 0 \). The bit-length of the input is at most \( \eta \). There exists a \( \text{poly}(n, m, \eta) \) time randomized algorithm that reduces it to a \( \text{OR} \) of \( n \) \((r, q)\)-CVP\(_p\)\(_\text{mp}\) instances of size \( O(p \cdot n^{p+1} \cdot \log(n + m + \eta)) \) for constant \( c_2 = \max\{c_1 + 2, 3\} \).

Furthermore, ‘YES’ instance always reduces to ‘YES’ instance and ‘NO’ instance reduces to ‘NO’ instance with at least \( 1 - \exp(-n) \) probability.

**Proof.** From Theorem V.2, we known an instance compression algorithm for \((1 + 2^{-n^{1+\alpha}})\)-GapCVP\(_p\)(\(B', d)\) where \( \phi = 2^{-n^{1+\alpha}} \lambda_{\mathcal{L}}(\mathcal{L}(B')) \). We will use the reduction from [1] and show that it reduces to OR of \( n \) instances of \((1 + 2^{-n^{1+\alpha}})\)-GapCVP\(_p\)(\(B', d)\). It is a \( \gamma\)-GapCVP\(_p\) instance of \( \gamma\)-GapCVP\(_p\) is a \( \gamma\)-YES instance. It is enough to just store \( n \) compressed CVP\(_p\)\(_\text{mp}\) instances.

Without loss of generality assume that \( B \) is a LLL reduced basis. Let \( B_i = \{b_1, \ldots, b_n\} \) and \( t_i = b_i \). We will show that \((B_i, t_i, d)\) is a valid instance of GapCVP\(_p\) where \( \phi \geq m \sqrt{n} \cdot 2^{3n/4} \lambda_{\mathcal{L}}(\mathcal{L}(B_i)). \) Notice that for all \( i \in [n] \) and \( z \in \mathbb{Z}^n, B_i z - t_i \in \mathcal{L}(B_i) \) and \( \lambda_{\mathcal{L}}(\mathcal{L}(B_i)) \geq \lambda_{\mathcal{L}}(\mathcal{L}(B)) \). Therefore, it is enough to show that \( \text{dist}_p(\mathcal{L}(B_i), t_i) \leq \|b_i\|_p \leq 2^{-n^{1+\alpha}} \lambda_{\mathcal{L}}(\mathcal{L}(B_i)). \) From Theorem II.4, we get that \( \|b_i\|_p \leq 2^{-n^{1+\alpha}} \lambda_{\mathcal{L}}(\mathcal{L}(B_i)) \). Hence, \( B_i, t_i, d \) is a valid instance of GapCVP\(_p\) where \( \phi \geq m \sqrt{n} \cdot 2^{3n/4} \lambda_{\mathcal{L}}(\mathcal{L}(B_i)). \)

Now we show that if \( \lambda_{\mathcal{L}}(\mathcal{L}(B_i)) \leq d \) then there exist an \( i \in [n] \) for which \( \text{dist}_p(\mathcal{L}(B_i), t_i) \leq d \). Let \( \nu = \sum z_i b_i \) be a shortest non-zero lattice vector in \( \ell_{\mathcal{L}} \) norm. We use the fact that the fact that there exists an index \( i \) such that \( z_i \) is odd integer. Otherwise \( \nu/2 \) is also a lattice vector which contradicts the assumption that \( \nu \) is shortest lattice vector. Therefore, we get that \( \text{dist}_p(\mathcal{L}(B_i), t_i) = \lambda_{\mathcal{L}}(\mathcal{L}(B_i)) \leq d \). For other direction, if \( \lambda_{\mathcal{L}}(\mathcal{L}(B_i)) > d \) then for all \( i, \text{dist}_p(\mathcal{L}(B_i), t_i) > d \). It directly follows from the fact that for all \( z \in \mathbb{Z}^n, B, x - t_i \in \mathcal{L}(\mathcal{L}(B_i)) \). We use Theorem V.2 and store \( n \) instances of a variant of CVP\(_p\)\(_\text{mp}\). It completes the proof.

\[ \square \]

VI. COMPRESSION FOR EXACT-CVP IN EVEN NORMS

We will use the following theorem from [11].

**Theorem VI.1.** \[11\text{ [Theorem 3]}\] For any positive integers \( m, N \), given set of integers \( \Sigma = \{a_1, \ldots, a_m\} \) there exists a
polynomial time algorithm that outputs $\Sigma' = \{a'_1, \ldots, a'_m\}$ where $\forall i \in [m], |a'_i| \leq 2^{3n^2} N^{(m+2)}$ and satisfy the following: for all $z \in \mathbb{Z}^n$, if $\|z\|_1 \leq N - 1$ then
\[
\text{sign} \left( \sum_{i=1}^{m} z_i a_i \right) = \text{sign} \left( \sum_{i=1}^{m} z_i a'_i \right).
\]

Now, we will show a deterministic compression for exact-CVP.

**Theorem VI.2.** For any positive integers $m, n$, given a CVP($B, t, d$) instance of bitleth $\eta$ where $B \in \mathbb{Z}^{m \times n}$ is a basis of a lattice $L$, target $t \in \mathbb{Z}^m$ and $d > 0$. There exists a poly($n, m, \eta$) time algorithm that reduces it to a $2^{n^2}$-CVP$^\text{IP}$ instance of size $O(n^3)$ bits.

**Proof.** We are given a basis $B \in \mathbb{Z}^{m \times n}$, target $t \in \mathbb{Z}^m$ and a distance $d > 0$. Without loss of generality, by applying Lemma IV.1, we can assume that for any $z \in \mathbb{Z}^n$,
\[
\|Bz - t\| = \text{dist}(B, t) \implies z \in \left[-2^{n^2}, 2^{n^2}\right]^n.
\]

Let $\forall i, j \in [n], \alpha_{i,j} := (a_i, b_j) \text{ and } \beta_i := -(b_i, t), \beta_{n+1} := (t, t)$. It is easy to see that for all $z \in \mathbb{Z}^n$,
\[
\|Bz - t\|_2^2 = \sum_{i=1}^{n} \left( \sum_{j=1}^{n} (z_i \cdot z_j) \alpha_{i,j} + z_i \beta_i \right) + \beta_{n+1}.
\]

Notice that this expression is linear in variables $\{z_1, z_2, \ldots, z_n\} \in \mathbb{Z}^{n^2 + n}$ and from Equation (12) we are only interested in the evaluation of above expression when each component of this vector has absolute value at most $2^{n^2}n$.

Now we will apply the deterministic compression algorithm from [11]. Let
\[
\Sigma = \{\alpha_{i,j}, \beta_i, \beta_{n+1}, d \mid \forall i, j \in [n]\} \in \mathbb{Z}^{n^2 + n + n^2 + 1}
\]
and $N = (n^2 + n + 2)2^{n^2} + 1$.

By using Theorem VI.1, we get $\Sigma' = \{\alpha'_{i,j}, \beta'_i, \beta'_{n+1}, d' \mid \forall i, j \in [n]\} \in \mathbb{Z}^{n^2 + n + n^2 + 1}$ which satisfies, for any $z \in \left[-2^{n^2}, 2^{n^2}\right]^n$,
\[
\sum_{i=1}^{n} \left( \sum_{j=1}^{n} (z_i \cdot z_j) \alpha'_{i,j} + z_i \beta'_i \right) + \beta'_{n+1} \leq d
\]
if and only if
\[
\sum_{i=1}^{n} \left( \sum_{j=1}^{n} (z_i \cdot z_j) \alpha_{i,j} + z_i \beta_i \right) + \beta_{n+1} \leq d'.
\]

Hence it reduces to $2^{n^2}$-CVP$^\text{IP}$ instance with input as inner products $\forall i, j \in [n], \alpha'_{i,j}, \beta'_i, \beta_{n+1}$ and number $d'$. It is also easy to compute that instance size is $O(n^8)$ bits.

We also generalize this result to compression of CVP$_p$ instance for even integer $p$. It requires an additional condition that the distance from the lattice is bounded by some factor times the length of shortest non-zero lattice vector.

**Theorem VI.3.** For any $m, n \in \mathbb{Z}^+, p \in \mathbb{Z}^+$ and constant $c_1 \in \mathbb{R}^+$, given a CVP$^\text{IP}_p(B, t, d)$ instance where $B \in \mathbb{Q}^{m \times n}$ is basis of lattice $L$, target $t \in \mathbb{Z}^m$, $d > 0$ and $\phi = 2^{c_1} \cdot \lambda_1^p$. The bit-length of the input is at most $\eta$. There exists a poly($n, m, \eta$) time algorithm that reduces it to a $\tau$-CVP$^\text{mpv}$ instance of size $O(n^{3p} + \eta^{3p}(n^{c_1} + \log m))$ where $\tau = 2^{n^{1/2}} \cdot m \cdot 2^{3n^2}$.

**Proof.** We are given a basis $B \in \mathbb{Z}^{m \times n}$, target $t \in \mathbb{Z}^m$ and a distance $d > 0$. Without loss of generality, by applying Lemma V.1, we can assume that for any $z \in \mathbb{Z}^n$,
\[
\|Bz - t\| = \text{dist}(B, t) \implies z \in [-\tau, \tau]^n,
\]

where $\tau = 2^{n^2} \cdot m \cdot 2^{3n^2}$. Let $B_{n+1} = t, \forall (i_1, \ldots, i_p) \in [n + 1]^p$
\[
\text{mvp}(i_1, \ldots, i_p) = \sum_{k=1}^{n} (\prod_{j=1}^{p} b_{kj}).
\]

From Lemma III.1, we get that for all $z \in \mathbb{Z}^n$,
\[
\|Bz - t\|^2_p = \sum_{(j_1, \ldots, j_p) \in [n+1]^p} (z_{j_1} \cdots z_{j_p}) \text{mvp}(j_1, \ldots, j_p),
\]

where $z_{n+1} = -1$. Notice that this expression is linear in variables $\forall (j_1, \ldots, j_p) \in [n+1]^p, z_{j_1} \cdots z_{j_p} \in \mathbb{Z}^{n+1}$. From Equation (13) we are only interested in the evaluation of above expression when each coordinate of coefficient vector has absolute value at most $\tau^p$ i.e. $z_{j_1} \cdots z_{j_p} \leq \tau^p$.

Now we will apply the deterministic compression algorithm from [11]. Let
\[
\Sigma = \{\text{mvp}(j_1, \ldots, j_p), d \mid \forall (j_1, \ldots, j_p) \in [n+1]^p\} \in \mathbb{Z}^{(n+1)^p + 1}
\]
and $N = ((n + 1)^p + 1) \cdot \tau^p + 1$.

Let $M = 2^{4((n+1)^p + 1) + N}((n+1)^p+1)((n+1)^p+3)$. By using Theorem VI.1, we get
\[
\Sigma' = \{\alpha_{i_1, \ldots, i_p}, d \mid \forall (i_1, \ldots, i_p) \in [n+1]^p\} \in [M]^{(n+1)^p + 1}
\]
which satisfies, for any $z \in [-\tau, \tau]^n$, $z_{n+1} = -1$,
\[
\sum_{(j_1, \ldots, j_p) \in [n+1]^p} (z_{j_1} \cdots z_{j_p}) \text{mvp}(j_1, \ldots, j_p) \leq d
\]
if and only if
\[
\sum_{(j_1, \ldots, j_p) \in [n+1]^p} (z_{j_1} \cdots z_{j_p}) \alpha_{j_1, \ldots, j_p} \leq d'.
\]

Hence it reduces to $\tau$-CVP$^\text{mpv}$ instance with input as multi-vector product $\forall (j_1, \ldots, j_p) \in [n+1]^p, \alpha_{j_1, \ldots, j_p}$ and number $d'$. It is also easy to compute that instance size is $(\log M) \cdot ((n + 1)^p + 1) = O(n^{3p} + n^{3p}(n^{c_1} + \log m))$ bits.

**Corollary VI.4.** For any $m, n \in \mathbb{Z}^+, p \in \mathbb{Z}^+$ and constant $c_1 \geq 1$, given a SVP$^\text{IP}_p(B, d)$ instance where $B \in \mathbb{Q}^{m \times n}$ is basis of lattice $L$, and $d > 0$. The bit-length of the input is
at most \(\eta\). There exists a \(\text{poly}(n, m, \eta)\) time algorithm that reduces it to a OR of \(n\) \(\tau\)-\(CVP^p\)\(^{\text{nop}}\) instances of size \(O(n^{c_p} + n^{3p}(n + \log m))\) where \(\tau = m^{c_p} \sqrt{\pi}2^{3m}/4\).

Proof exactly follows the arguments of Theorem V.3 and uses Theorem VI.3.

**VII. IMPLICATION TO SETH HARDNESS OF CVP**

In this section, we show that it is not possible to get a polynomial time Turing reduction from \(k\)-SAT to \(CVP_2\) instance of \(n^c\) rank lattice unless polynomial hierarchy collapses to third level. We also generalize this result for \(CVP_p\) (distance guarantee) for even \(p\). We extend this result for exponential time reduction with the restriction that reduction will only make fixed polynomial number of calls to \(CVP_p\) oracle. We also give barriers for randomized polynomial time reductions. However, we are only able to show it for non-adaptive reductions.

**Theorem VII.1.** For any constants \(c, c_1 > 0\), there exists a constant \(k_0\) such that for any \(k > k_0\) there does not exist a polynomial time probabilistic reduction with no false negatives from \(k\text{-SAT}\) on \(n\)-variables to \(CVP_2\) on \(n^{c_1}\) rank lattice that makes at most \(n^c\) calls to \(CVP_2\) oracle, unless \(\text{coNP} \subseteq \text{NP/Poly}\).

**Proof.** Given a \(CVP_2\) instance of \(n^{c_1}\) rank lattice, by Theorem VI.2, we get a compressed instance of \(CVP_p^{\text{nop}}\) of size \(\tilde{O}(n^{c_1})\). Let \(k_0 = c + 8c_1\). Let’s assume that for some \(k > k_0\) there exist a polynomial time probabilistic reduction without false negative from \(k\text{-SAT}\) on \(n\)-variables to \(CVP_2\) on \(n^{c_1}\) rank lattice and it makes at most \(n^c\) calls to \(CVP_2\) oracle. This can also be seen as an oracle communication protocol for \(k\text{-SAT}\) where for each call to \(CVP_2\) instance, first player sends the compressed instance to second player (which is computationally unbounded). From definition of oracle communication protocol, the cost of this protocol is at most \(n^c \cdot \tilde{O}(n^{c_1})\). Therefore, by Theorem II.12, we get \(\text{coNP} \subseteq \text{NP/Poly}\) as \(k > k_0 = c + 8c_1\).

Notice that, the above arguments holds for adaptive reduction for \(k\text{-SAT}\) to \(CVP_2\). This completes the proof.

We also get the similar result for \(CVP_p\) for all even positive integer \(p\).

**Theorem VII.2.** For any constants \(c, c_1, c_2 > 0\) and \(p \in 2\mathbb{Z}^+\), there exists a constant \(k_0\) such that for any \(k > k_0\) there does not exist a polynomial time probabilistic reduction with no false negatives from \(k\text{-SAT}\) on \(n\)-variables to \(CVP_p\) on \(n^{c_2}\) rank lattice where \(\phi = 2^{n^{c_1}} \cdot \lambda_1^{(p)},\) that makes at most \(n^c\) calls to \(CVP_p\) oracle, unless \(\text{coNP} \subseteq \text{NP/Poly}\).

**Proof.** Given a \(CVP_p\) instance of \(n^{c_2}\) rank lattice, by Theorem VI.3, we get a compressed instance of \(CVP_p^{\text{nop}}\) of size \(\tilde{O}(n^{c_2})\) where \(c_1 = c_2, \max\{4p, 3p + c_1\}\). Let \(k_0 = c + c_3\). Let’s assume there exist a polynomial time probabilistic reduction without false negative from \(k\text{-SAT}\) on \(n\)-variables to \(CVP_p\) on \(n^{c_2}\) rank lattice and it makes at most \(n^c\) calls to \(CVP_p\) oracle. The reduction can also be seen as an oracle communication protocol for \(k\text{-SAT}\) for each call to \(CVP_p\) instance, first player sends the compressed instance to second player (which is computationally unbounded). From definition of oracle communication protocol, the cost of this protocol is at most \(n^c \cdot \tilde{O}(n^{c_2})\). Therefore, by Theorem II.12, we get \(\text{coNP} \subseteq \text{NP/Poly}\) as \(k > k_0 = c + c_3\).

Notice that, the above arguments also holds for adaptive reduction for \(k\text{-SAT}\) to \(CVP_p\). This completes the proof.

For \(SVP_p\), we get the following result.

**Theorem VII.3.** For any constants \(c, c_1 > 0\) and \(p \in 2\mathbb{Z}^+\), there exists a constant \(k_0\) such that for any \(k > k_0\) there does not exist a polynomial time probabilistic reduction with no false negatives from \(k\text{-SAT}\) on \(n\)-variables to \(SVP_p\) on \(n^{c_1}\) rank lattice that makes at most \(n^c\) calls to \(SVP_p\) oracle, unless \(\text{coNP} \subseteq \text{NP/Poly}\).

Proof directly follows from Corollary VI.4 and Theorem II.12; it follows the same arguments as above.

We extend these barrier for exponential time reduction. Specifically, we get the following results.

**Theorem VII.4.** For any constants \(c, c_1 > 0\), there exists a constant \(k_0\) such that for any \(k > k_0\) and \(T > 0\), there does not exist a probabilistic reduction without false negative from \(k\text{-SAT}\) on \(n\)-variables to \(CVP_2\) on \(n^{c_1}\) rank lattice, in time \(T\) and reduction makes at most \(n^c\) calls to \(CVP_2\) oracle, unless \(\text{coNP} \subseteq \text{NTIME}(\widetilde{T\text{poly}})\).

Proof follows directly from Theorem II.13 and Theorem VI.2; it follows the same arguments as Theorem VII.1.

By Theorem VI.3, we also get following result for \(\ell_p\) norms for all positive even integer \(p\).

**Theorem VII.5.** For any constants \(c, c_1, c_2 > 0\) and \(p \in 2\mathbb{Z}^+\), there exists a constant \(k_0\) such that for any \(k > k_0\) and \(T > 0\), there does not exist a probabilistic reduction without false negative from \(k\text{-SAT}\) on \(n\)-variables to \(CVP_p\) on \(n^{c_2}\) rank lattice where \(\phi = 2^{n^{c_1}} \cdot \lambda_1^{(p)},\) in time \(T\) and reduction makes at most \(n^c\) calls to \(CVP_p\) oracle, unless \(\text{coNP} \subseteq \text{NTIME}(\widetilde{T\text{poly}})\).

We also give a barrier for randomized polynomial time reduction with both side error from \(k\text{-SAT}\) to \(CVP_p\) in even norm. We are only able to show this barrier for non-adaptive reductions. We require the following result from [41].

**Theorem VII.6.** [41][Theorem 7.1] Let \(L\) be any language and \(t_1 = t_1(n), t_2 = t - 2(n) > 0\). Suppose that there exists a probabilistic polynomial time instance compression of \(OR(L)\) such that it reduces problem of \(OR\) of \(t_1\) instances of \(L\) of length \(n\) to instance of some language \(L'\) of bitlength \(t_2\) with error bound \(\varepsilon(n) < 1/2\). Let

\[
\delta = \min \left\{ \sqrt{\frac{\ln 2}{2} - t_2 + 1}, \frac{1}{t_1}, 1 - 2^{-\frac{t_2 - 3}{2}} \right\}
\]

If for some constant \(c > 0\), we have

\[
(1 - 2^{c}(n))^2 - \delta \geq \frac{1}{n^c}
\]
then there is a Karp reduction from $L$ to a problem in $\text{SZK}$. The reduction is computable in non-uniform polynomial time; in particular this implies $L \in \text{NP}/\text{Poly} \cap \text{coNP}/\text{Poly}$.

**Theorem VII.7.** For any constants $c, c_1 > 0$, there exists a constant $k_0$ such that for any $k > k_0$ there does not exist a (non-adaptive) randomized polynomial time reduction from $k$-$\text{SAT}$ on $n$-variables to $\text{CVP}_p$ on $n^c$ rank lattice with constant error bound and reduction make at most $n^{c_1}$ calls to $\text{CVP}_p$ oracle, unless there are non-uniform, statistical zero-knowledge proofs for all languages in NP.

**Proof.** Let $c_2 := 8c_1$ and $k_0 = c + c_2$. Let’s assume that there exists a non-adaptive randomized polynomial time reduction from $k$-$\text{SAT}$ on $n$-variables to $\text{CVP}_p$ on $n^{c_1}$ rank lattice with constant error bound and the reduction make at most $n^{c_2}$ calls to $\text{CVP}_p$ oracle. Using Theorem VI.2, for $\text{CVP}_p$ on $n^{c_1}$ rank lattice, we get instance compression of size $O(n^{c_2})$. As the reduction from $k$-$\text{SAT}$ to $\text{CVP}_p$ is non-adaptive, in polynomial time we can identify the $\text{CVP}_p$ instances to which reduction will make oracle calls. Therefore, by storing corresponding compressed $\text{CVP}_p$ instances, we get probabilistic instance compression of $k$-$\text{SAT}$ of size $O(n^{c_2}) = O(n^{k_0})$.

Now, from Lemma II.16, we get an instance compression for $k$-Vertex cover of $O(n^{k_0})$ size. Notice that, by a trivial reduction between $k$-Vertex cover and $k$-Clique, we also get $O(n^{k_0})$ bit-size instance compression for $k$-Clique. Therefore, by Lemma II.15, there is an instance compression for $\text{OR}(3$-SAT) of bitlenth $O\left(\left(n \cdot \max\{n, t^{1/k+o(1)}\}\right)^{k_0}\right)$. As $k_0$ is a constant less than $k$, if we take $t$ as sufficiently large polynomial in $n$ we get a $O(t \log t)$ bitlenth instance compression of $\text{OR}(3$-SAT) with constant error bound. Notice that this sequence of reductions are deterministic and preserves the error bound. Therefore, by Theorem VII.6, we get a non-uniform polynomial time Karp reduction from 3-SAT to a problem in $\text{SZK}$. Hence, it implies that there are non-uniform, statistical zero-knowledge proofs for all languages in NP. It completes the proof.

By Theorem VI.3 and following same arguments as above we get the following result for $\text{CVP}_p$ for all even positive integers $p$.

**Theorem VII.8.** For any constants $c, c_1, c_2 > 0$, there exists a constant $k_0$ such that for any $k > k_0$ there does not exist a (non-adaptive) randomized polynomial time reduction from $k$-$\text{SAT}$ on $n$-variables to $\text{CVP}_p$ on $n^{c_1}$ rank lattice with constant error bound where $\phi = 2^{n^{c_2}} \cdot 1^{(p)}$, and reduction make at most $n^{c_2}$ calls to $\text{CVP}_p$ oracle, unless there are non-uniform, statistical zero-knowledge proofs for all languages in NP.

**Remark 1.** Note that, in this paper, all the barriers for reduction from $k$-$\text{SAT}$ to $\text{CVP}_p$ also holds for reduction from $k$-$\text{SAT}$ to $\text{SVP}_p$ because of an efficient reduction from $\text{SVP}_p$ to $\text{CVP}_p$ [11].

**VIII. Barrier for SETH-hardness of Subset-Sum**

Subset-Sum is one of the most extensively studied problem in computer science. Showing a fine-grained hardness of Subset-Sum based on SETH is an important open problem. Harnik and Naor [38], gave an algorithm for instance compression of arbitrary Subset-Sum instance. In this section, we will describe the consequences of this compression to get a reduction from $k$-$\text{SAT}$ to Subset-Sum.

**Lemma VIII.1.** [38][Claim 2.7] For any positive integers $n, m$, there exists a randomized polynomial time algorithm that compresses any arbitrary Subset-Sum instance $(a_1, \ldots, a_n, t)$ on $n$ numbers with

$$\eta := \left\lceil \max \{\log_2|a_1|, \ldots, \log_2|a_n|, \log_2|t|\} \right\rceil,$$

to $O(n^2 + n \log \eta)$ bits. Furthermore, the reduction does not give false negative.

**Theorem VIII.2.** For any constants $c, c' > 0$, there exists a constant $k_0$ such that for any $k > k_0$ there does not exist a polynomial time probabilistic reduction with no false negatives from $k$-$\text{SAT}$ on $n$-variable to Subset-Sum on $n^{c'}$ numbers which makes at most $O(n^{c'})$ calls to Subset-Sum oracle, unless coNP $\subseteq$ NP/$\text{Poly}$.

**Proof.** Given a Subset-Sum instance of $n^{c'}$ numbers, by Lemma VIII.1, we can get a compressed instance of size $O(n^{2c'})$. Let $k_0 = c + 2c'$. Let’s assume there exist a polynomial time probabilistic reduction without false negative from $k$-$\text{SAT}$ on $n$-variables to Subset-Sum instance on $n^{c'}$ numbers and the reduction only makes $O(n^{c'})$ calls to Subset-Sum oracle. Notice that, the reduction can also be seen as an oracle communication protocol where for each call to Subset-Sum instance, first player send the compressed instance to second player (which is computationally unbounded). From definition of oracle communication protocol, the cost of this protocol is at most $n^{c'} \cdot O(n^{2c'}) = O(n^{k_0})$. Therefore, by Theorem II.12, we get coNP $\subseteq$ NP/$\text{Poly}$ as $k > c + c'$.

Notice that, the above arguments holds for adaptive reductions from $k$-$\text{SAT}$ to Subset-Sum. This completes the proof.

**Remark 2.** Note that, similar to Theorems VII.4 and VII.7, we can also get the same barriers for reductions from $k$-$\text{SAT}$ to Subset-Sum.

**Acknowledgment**

The authors thank Huck Bennett, Zvika Brakerski, Alexander Golovnev, Zeyong Li, Noah Stephens-Davidowitz, and Prashant Nalini Vasudevan for helpful discussions. In particular, we thank Alexander Golovnev for many helpful comments on an earlier draft of this work. We also thank FOCS reviewers for pointing out the instance compression algorithm for exact-CVP.
REFERENCES

[1] O. Goldreich, D. Micciancio, S. Safra, and J. Seifert, “Approximating shortest lattice vectors is not harder than approximating closest lattice vectors,” Inf. Process. Lett., vol. 71, no. 2, pp. 55–61, 1999. [Online]. Available: https://doi.org/10.1016/S0020-0190(99)00083-6

[2] M. Ajtai, “Generating hard instances of lattice problems,” in STOC, 1996.

[3] D. Micciancio and O. Regev, “Worst-case to average-case reductions based on Gaussian measures,” in FOCS, 2004.

[4] O. Regev, “On lattices, learning with errors, random linear codes, and cryptography,” J. ACM, vol. 56, no. 6, pp. 3–34:1–34:40, Sep. 2009. [Online]. Available: http://doi.acm.org/10.1145/1568318.1568324

[5] ——, “Lattice-based cryptography,” in Advances in Cryptology - CRYPTO 2006, 26th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 2006, Proceedings, 2006, pp. 131–141. [Online]. Available: https://doi.org/10.1109/11818175_8

[6] D. Micciancio and O. Regev, “Lattice-based cryptography,” 2008.

[7] C. Gentry, “Fully homomorphic encryption using ideal lattices,” in STOC, 2009.

[8] Z. Brakerski and V. Vaikuntanathan, “Lattice-based FHE as secure as PKE,” in ITCS, 2014.

[9] A. Lenstra, H. Lenstra, and L. Lovász, “Factoring polynomials with rational coefficients,” Math. Ann., vol. 261, pp. 515–534, 1982.

[10] R. Kannan, “Minkowski’s convex body theorem and integer programming,” Math. Oper. Res., vol. 12, no. 3, pp. 415–440, 1987. [Online]. Available: https://doi.org/10.1287/moor.12.3.415

[11] A. Frank and É. Tardos, “An application of simultaneous Diophantine approximation in combinatorial optimization,” Combinatorica, vol. 7, no. 1, pp. 49–65, 1987. [Online]. Available: https://doi.org/10.1007/BF02579200

[12] A. Shamir, “A polynomial-time algorithm for breaking the basic Merkle-Hellman cryptosystem,” IEEE Trans. Information Theory, vol. 30, no. 5, pp. 699–704, 1984. [Online]. Available: https://doi.org/10.1109/TIT.1984.1056964

[13] E. F. Brickell, “Breaking iterated knapsacks,” in CRYPTO, 1984.

[14] J. C. Lagarias and A. M. Odlyzko, “Solving low-density subset sum problems,” J. ACM, vol. 30, no. 2, pp. 229–246, 1983. [Online]. Available: https://doi.org/10.1145/2452.2446

[15] D. Aggarwal, D. Dadush, and N. Stephens-Davidowitz, “Solving the closest vector problem in 2 n time—the discrete gaussian strikes again!” in 2015 IEEE 56th Annual Symposium on Foundations of Computer Science. IEEE, 2015, pp. 563–582.

[16] J. Blömer and S. Naewe, “Sampling methods for shortest vectors, closest vectors and successive minima,” Theoretical Computer Science, vol. 410, no. 18, pp. 1648–1665, 2009.

[17] D. Dadush, “A (1/2+ε)-time sieving algorithm for approximate integer programming,” LATIN 2012: Theoretical Informatics, p. 207, 2011.

[18] D. Aggarwal and P. Mukhopadhyay, “Improved algorithms for the shortest vector problem and the closest vector problem in the infinity norm,” in 29th International Symposium on Algorithms and Computation (ISAAC 2018). Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

[19] F. Eisenbrand and M. Venzin, “Approximate cvp in time 20.802 n,” Journal of Computer and System Sciences, vol. 124, pp. 129–139, 2022.

[20] D. Aggarwal, Y. Chen, R. Kumar, Z. Li, and N. Stephens-Davidowitz, “Approximating CVP to within almost-polynomial factors is NP-hard,” in 2013 IEEE 54th Annual Symposium on Foundations of Computer Science, 2013, pp. 719–727.

[21] F. Eisenbrand and M. Venzin, “Approximate CVPs in time 20.902 n,” in ESA, 2020, https://arxiv.org/abs/2005.04957.

[22] R. Kannan, “Dynamic programming,” Science, vol. 153, no. 3731, pp. 34–37, 1966.

[23] A. Shamir, “A polynomial-time algorithm for breaking the basic Merkle–Hellman cryptosystem,” IEEE Trans. Information Theory, vol. 30, no. 5, pp. 699–704, 1984. [Online]. Available: https://doi.org/10.1109/TIT.1984.1056964

[24] E. F. Brickell, “Breaking iterated knapsacks,” in CRYPTO, 1984.

[25] J. C. Lagarias and A. M. Odlyzko, “Solving low-density subset sum problems,” J. ACM, vol. 30, no. 2, pp. 229–246, 1983. [Online]. Available: https://doi.org/10.1145/2452.2446

[26] D. Aggarwal, D. Dadush, and N. Stephens-Davidowitz, “Solving the closest vector problem in 2 n time—the discrete gaussian strikes again!” in 2015 IEEE 56th Annual Symposium on Foundations of Computer Science. IEEE, 2015, pp. 563–582.

[27] B. Wraggles and S. Naewe, “Sampling methods for shortest vectors, closest vectors and successive minima,” Theoretical Computer Science, vol. 410, no. 18, pp. 1648–1665, 2009.

[28] D. Aggarwal, Y. Chen, R. Kumar, Z. Li, and N. Stephens-Davidowitz, “Approximating CVP to within almost-polynomial factors is NP-hard,” in 2013 IEEE 54th Annual Symposium on Foundations of Computer Science, 2013, pp. 719–727.

[29] F. Eisenbrand and M. Venzin, “Approximate cvp in time 20.802 n,” Journal of Computer and System Sciences, vol. 124, pp. 129–139, 2022.

[30] D. Aggarwal, Y. Chen, R. Kumar, Z. Li, and N. Stephens-Davidowitz, “Approximating CVP to within almost-polynomial factors is NP-hard,” in 2013 IEEE 54th Annual Symposium on Foundations of Computer Science, 2013, pp. 719–727.
Let makes queries to the second party (which runs in time such that the output is \( x \in L \) belongs to \( N \) such that the problem of \( N \) be a language and has an oracle communication protocol, of inputs of length at most \( \ell \) of protocol \( P(x, x_2, \ldots, x_t; r) \) We will iteratively add rejecting transcripts to \( A_n \) such that they eventually cover all \( x \in L \).

Let \( F \) be the set of all \( x \in L \) that have so far not been covered. We know that for every \( t \)-tuple of instances in \( F^t \) for some randomness \( r \), there exists a rejecting transcript (in fact, there exist many rejecting transcripts). So, since there are at most \( 2^n \) transcripts in total, there exists a transcript \( \tau \) that is a rejecting transcript for at least \( |F|^t/2^n \) tuples of instances. Let \( G \) be the subset of \( F \) that is covered by \( \tau \). Thus, any tuple in \( F^t \) for which \( \tau \) is a rejecting transcript must be in \( G^t \). This implies that

\[
|G|^t \geq |F|^t 2^{-c \ell} \quad \Longrightarrow \quad |G| \geq |F| 2^{-c/t}.
\]

Include the transcript \( \tau \) in \( A_n \) and repeat this step by taking \( F = F \setminus G \) until there is no more \( x \in L \) that is not covered by some \( \tau \) in \( A_n \).

There are at most \( 2^n \) instance of \( x \) of bilen length \( n \). It is easy to see that by repeating the above procedure \( \ell \) times, the set \( A_n \) covers at least \((1 - (1 - 2^{-c})^{1/\ell})^\ell \cdot 2^n \geq (1 - e^{-\ell 2^{-c/t}})^\ell \cdot 2^n \) inputs. It implies that all instances will be covered after \( O\left(\frac{n}{2^{-c/t}}\right) \) repetitions of the above step. As \( c = O(t \log t) \) where \( t \) is some polynomial in \( n \), we get that the size of the set \( A_n \) is \( poly(n) \). Notice that each transcript is also polynomially bounded in \( n \) and the running time is also bounded in \( poly(n) \cdot T \). The resulting algorithm for \( L \) runs in \( NTIME(poly(n) \cdot T) / poly \).

Theorem II.13. Let \( k \geq 3 \) and \( \epsilon > 0 \) a positive real. For any \( T = T(n) > 0 \), there is no oracle communication protocol for \( k\text{-SAT} \), of cost \( O(n^{k-\epsilon}) \) that runs in time \( T \), unless \( coNP \subseteq NTIME(poly(n) \cdot T) / poly \).

Proof. Let’s assume that there is a randomized oracle communication protocol of \( k\text{-SAT} \) of cost \( O(n^{c}) \) for some constant \( c < k \) and runs in time \( T \). From Lemma II.16, it gives an oracle communication protocol for \( k\text{-Clique} \) of cost \( O(n^c) \). Notice that, by a trivial reduction from \( k\text{-Clique} \) to \( k\text{-Clique} \), it gives a \( O(n^c) \) cost protocol for \( k\text{-Clique} \). Therefore, by Lemma II.15, there is an oracle communication protocol for \( OR(3\text{-SAT}) \) of cost \( O\left(\left(n \cdot \max\{n, t^{1/k + o(1)}\}\right)^c\right) \). As \( c \) is a constant less than \( k \), if we take \( t \) as sufficiently large polynomial in \( n \) it gives a \( O(t \log t) \) cost randomized oracle communication protocol for \( OR(3\text{-SAT}) \). Hence, if \( coNP \not\subseteq NTIME(poly(n) \cdot T) / poly \), then we get a contradiction from Lemma A.1. It completes the proof.

APPENDIX

Lemma A.1. Let \( L \) be a language and \( t : \mathbb{N} \to \mathbb{N} \setminus \{0\} \) be polynomially bounded in \( n \) such that the problem of deciding whether at least one of \( t \) inputs of length at most \( n \) belongs to \( L \) has an oracle communication protocol, of cost \( O(t \log t) \), where the first player runs in time \( T \). Then \( L \in NTIME(poly(n) \cdot T) / poly \).

Proof. Let \( P \) be the oracle communication protocol for language \( L \) that runs in time \( poly(n) \cdot T \) such that the output is a deterministic function of the communication transcript. For input \( x \), protocol \( P \) makes queries to the second party (which we sometimes call the oracle) and receives some outputs from the second party. We use \( C \) to denote the communication transcript between the two parties. The cost of the protocol is the number of bits of communication from the first player to the second player. First player is allowed to use randomness and protocol always output accept (outputs \( YES \)) if instance belongs to the language \( L \).

We will use the following equivalence: an input \( x \) is in \( L \) if and only if there exist a sequence \( (x_2, \ldots, x_t) \) and randomness \( r \) such that \( P(x, x_2, \ldots, x_t; r) \) rejects (outputs \( NO \)). It follows from the fact that protocol does not give false negative. Let \( s = poly(n, t) \) be the number of random bits required to execute the oracle communication protocol. We will show existence of a polynomial size advice string \( A_n \) which contains a subset of the transcripts of the protocol \( P \) such that for every \( x \in L \) there exists \( x_2, \ldots, x_t \) and randomness \( r \) such that the communication transcript \( C \) of protocol \( P \) on input \( (x, x_2, \ldots, x_t; r) \) is in \( A_n \) and \( P(x, x_2, \ldots, x_t; r) \) rejects. Before showing the existence of the advice string, we show why this is sufficient. By using such an advice string \( A_n \), we get the following algorithm for input \( x \):

1. Guess a sequence \( (r, x_2, \ldots, x_t) \), where each \( x_i \) is of length \( n \) and randomness \( r \) is of length \( s \).
2. Check whether the communication transcript \( C \) of protocol \( P \) on input \( (x, x_2, \ldots, x_t; r) \) is consistent with a transcript in \( A_n \) and \( P(x, x_2, \ldots, x_t; r) \) rejects. If so then accept, otherwise reject.

To check the consistency of protocol \( P \) on input \( (x, x_2, \ldots, x_t; r) \) with any transcript \( \tau \), we check the input of the first player to the second player is consistent with transcript \( \tau \) and whenever the first player expects an output from the second player we give the desired output by using the transcript \( \tau \). The correctness of the protocol follows from the equivalence as mentioned earlier.

In the rest of the proof, we will show that there exists a polynomial-size advice string \( A_n \).

We find the set \( A_n \) in the greedy way. Notice that any transcript \( \tau \) of the protocol \( P \) on input \( (x, x_2, \ldots, x_t; r) \) is a deterministic function of the bits sent by the first player to second player. Let us assume that the cost of oracle communication protocol is \( c = O(t \log t) \). It implies that there are at most \( 2^n \) many distinct rejecting transcripts.

For any rejecting transcript \( \tau \), and any \( x \in L \), we say that \( \tau \) covers \( x \) if there exists \( x_2, \ldots, x_t \) and \( r \) where \( x_i \in L \) and \( r_i \in \{0,1\}^\epsilon \) such that \( \tau \) is a transcript for \( P(x, x_2, \ldots, x_t; r) \).