(Gap/S)ETH Hardness of SVP

Divesh Aggarwal and Noah Stephens-Davidowitz

National University of Singapore

December 15, 2017

Talk Outline

- A very brief introduction to lattices
- An introduction to the Exponential Time Hypotheses
- Hardness of SVP_p for $p \geq 2.14$ under SETH
- Summary of Other Results
- Conclusions and open questions

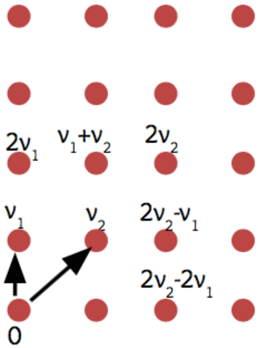
Talk Outline

- A very brief introduction to lattices
- An introduction to the Exponential Time Hypotheses
- Hardness of SVP_p for $p \geq 2.14$ under SETH
- Summary of Other Results
- Conclusions and open questions

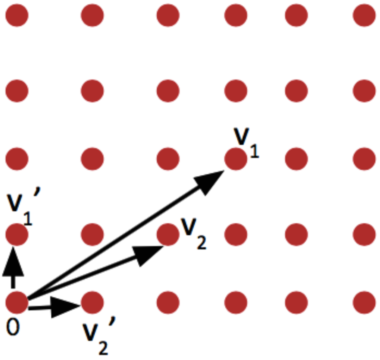
Lattices

- A lattice is a set of points
- $\mathcal{L} = \{a_1 v_1 + \cdots + a_n v_n \mid a_i \text{ integers}\}$.

  for some linearly independent vectors $v_1, \ldots, v_n \in \mathbb{R}^d$.
- We call $v_1, \ldots, v_n$ a basis, n the rank, and d the dimension of the lattice $\mathcal{L}$.



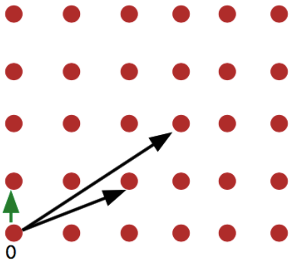
Basis is Not Unique



Good Basis: $v_1', \; v_2'$

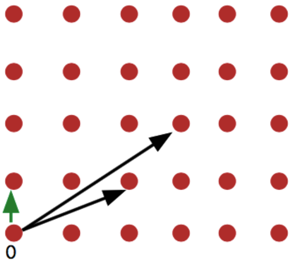
Bad Basis: $v_1, \; v_2$

Lattice Problems



- SVP: Given a lattice basis and a length $r > 0$, decide whether $\lambda_1 \leq r$ or $\lambda_1 > r$, where λ_1 is the length of a shortest non-zero vector.

Lattice Problems



- SVP: Given a lattice basis and a length $r > 0$, decide whether $\lambda_1 \leq r$ or $\lambda_1 > r$, where λ_1 is the length of a shortest non-zero vector.
- CVP: Given a basis of $\mathcal{L}$, a vector $\vec{t} \in \mathbb{R}^n$ and a length $r > 0$, decide whether $\text{dist}(\vec{t}, \mathcal{L}) \leq r$ or $\text{dist}(\vec{t}, \mathcal{L}) > r$, where $\text{dist}(\vec{t}, \mathcal{L})$ is the shortest distance of the vector $\vec{t}$ from the lattice.

ℓ_p norms

Typically, we define length in terms of the ℓ_p norm for some $1 \leq p \leq \infty$ defined as

$$\|\vec{x}\|_p := (|x_1|^p + |x_2|^p + \cdots + |x_d|^p)^{1/p}$$

for finite p and

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

We write SVP_p for SVP in the ℓ_p norm.

The LLL Algorithm [LLL82]

- An efficient algorithm that outputs a “somewhat short" lattice vector
- Applications include:
  - Solving integer programs in a fixed dimension
  - Factoring polynomials over rationals
  - Finding integer relations:

$$5.709975946676696\ldots \stackrel{?}{=} 4 + 3\sqrt{5}$$

  - Attacking knapsack-based cryptosystems [LagOdl85] and variants of RSA [Has85,Cop01]

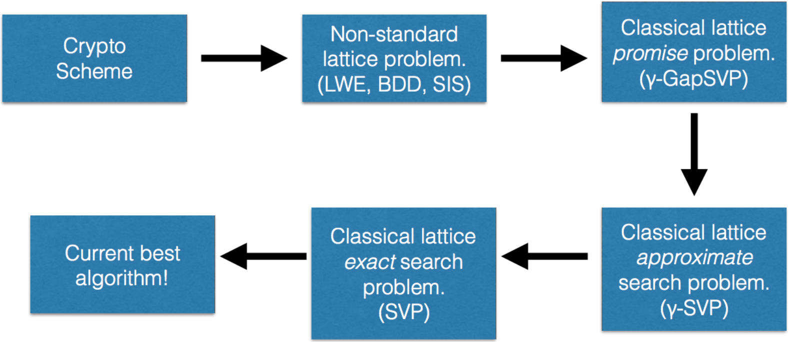
Lattices and Cryptography

- Lattices can also be used to create cryptosystems.
- This started with a breakthrough of Ajtai[Ajt96].
- Cryptography based on lattices has many advantages compared with 'traditional' cryptography like RSA:
  - It has strong, mathematically proven, security.
  - It is believed to be resistant to quantum computers.
  - In some cases, it is much faster.
  - It can do more, e.g., fully homomorphic encryption, which is one of the most important cryptographic primitives.

Lattice-based Crypto

- Public-key Encryption [Reg05,KTX07,PKW08]
- CCA-Secure PKE [PW08,Pei09].
- Identity-based Encryption [GPV08]
- Oblivious Transfer [PVW08]
- Circular Secure Encryption [ACPS09]
- Hierarchical Identity-based Encryption [Gen09,CHKP09,ABB09].
- Fully Homomorphic Encryption [Gen09,BV11,Bra12].
- And more...

Faster Algorithms for SVP – A Threat to Cryptography



Best Known Algorithms for SVP

	Norm	Time	Space
[Kan86]	Euclidean (ℓ_2)	$n^{O(n)}$	poly(n)
[ADRS15,AS18]	Euclidean (ℓ_2)	$2^{n+o(n)}$	$2^{n+o(n)}$
[BN07,AJ08]	All norms	$2^{O(n)}$	$2^{O(n)}$

Hardness of SVP

- SVP is known to be NP-hard under randomized reductions [Ajt98].

Hardness of SVP

- SVP is known to be NP-hard under randomized reductions [Ajt98].
- Implication: There is no polynomial time algorithm unless P $=$ NP.

Hardness of SVP

- SVP is known to be NP-hard under randomized reductions [Ajt98].
- Implication: There is no polynomial time algorithm unless $\mathrm{P} = \mathrm{NP}$.
- Does not rule out the possibility of a $2^{n/100}$ or a $2^{n^{1/10}}$ algorithm.

Hardness of SVP

- SVP is known to be NP-hard under randomized reductions [Ajt98].
- Implication: There is no polynomial time algorithm unless $P = NP$.
- Does not rule out the possibility of a $2^{n/100}$ or a $2^{n^{1/10}}$ algorithm.
  - This will still break cryptosystems in practice.
- Question: Can we show a 2^{cn} lower bound for some constant c under a reasonable complexity-theoretic assumption?

Hardness of SVP

- SVP is known to be NP-hard under randomized reductions [Ajt98].
- Implication: There is no polynomial time algorithm unless $P = NP$.
- Does not rule out the possibility of a $2^{n/100}$ or a $2^{n^{1/10}}$ algorithm.
  - This will still break cryptosystems in practice.
- Question: Can we show a 2^{cn} lower bound for some constant c under a reasonable complexity-theoretic assumption?
  - YES (this talk)

Talk Outline

- A very brief introduction to lattices
- An introduction to the Exponential Time Hypotheses
- Hardness of SVP_p for $p \geq 2.14$ under SETH
- Summary of Other Results
- Conclusions and open questions

3-SAT and k-SAT

- 3-SAT: Given a formula ϕ in 3-CNF with n variables and m clauses, decide whether there is a satisfying assignment.
- 3-CNF: ϕ is a conjunction of clauses, with each clause being a disjunction of 3 literals – variables or their negations

$$(x_1 \vee x_7 \vee \neg x_{17}) \wedge (x_{12} \vee \neg x_{15}) \wedge (\neg x_4 \vee x_6 \vee x_{12}) \cdots$$

3-SAT and k-SAT

- 3-SAT: Given a formula ϕ in 3-CNF with n variables and m clauses, decide whether there is a satisfying assignment.

- 3-CNF: ϕ is a conjunction of clauses, with each clause being a disjunction of 3 literals – variables or their negations

$$(x_1 \vee x_7 \vee \neg x_{17}) \wedge (x_{12} \vee \neg x_{15}) \wedge (\neg x_4 \vee x_6 \vee x_{12}) \cdots$$

- Trivial algorithm: $\widetilde{O}(2^n)$ time.

3-SAT and k-SAT

- 3-SAT: Given a formula ϕ in 3-CNF with n variables and m clauses, decide whether there is a satisfying assignment.

- 3-CNF: ϕ is a conjunction of clauses, with each clause being a disjunction of 3 literals – variables or their negations

$$(x_1 \vee x_7 \vee \neg x_{17}) \wedge (x_{12} \vee \neg x_{15}) \wedge (\neg x_4 \vee x_6 \vee x_{12}) \cdots$$

- Trivial algorithm: $\widetilde{O}(2^n)$ time.

- Smarter algorithm: Take any clause not satisfied so far, and branch on the evaluations of the variables satisfying the clause.
  - $$T(n) \leq \max(7 \cdot T(n-3)\ ,\ 3 \cdot T(n-2)\ ,\ T(n-1)) \leq 7^{n/3} < 2^{0.936n}\ .$$

3-SAT and k-SAT

- 3-SAT: Given a formula ϕ in 3-CNF with n variables and m clauses, decide whether there is a satisfying assignment.

- 3-CNF: ϕ is a conjunction of clauses, with each clause being a disjunction of 3 literals – variables or their negations

$$(x_1 \vee x_7 \vee \neg x_{17}) \wedge (x_{12} \vee \neg x_{15}) \wedge (\neg x_4 \vee x_6 \vee x_{12}) \cdots$$

- Trivial algorithm: $\widetilde{O}(2^n)$ time.

- Smarter algorithm: Take any clause not satisfied so far, and branch on the evaluations of the variables satisfying the clause.
  - $$T(n) \leq \max(7 \cdot T(n-3)\ ,\ 3 \cdot T(n-2)\ ,\ T(n-1)) \leq 7^{n/3} < 2^{0.936n}\ .$$

- Current Best: $2^{0.388n}$ [Her14].

3-SAT and k-SAT

- 3-SAT: Given a formula ϕ in 3-CNF with n variables and m clauses, decide whether there is a satisfying assignment.

- 3-CNF: ϕ is a conjunction of clauses, with each clause being a disjunction of 3 literals – variables or their negations

$$(x_1 \vee x_7 \vee \neg x_{17}) \wedge (x_{12} \vee \neg x_{15}) \wedge (\neg x_4 \vee x_6 \vee x_{12}) \cdots$$

- Trivial algorithm: $\widetilde{O}(2^n)$ time.

- Smarter algorithm: Take any clause not satisfied so far, and branch on the evaluations of the variables satisfying the clause.
  - $$T(n) \leq \max(7 \cdot T(n-3)\ ,\ 3 \cdot T(n-2)\ ,\ T(n-1)) \leq 7^{n/3} < 2^{0.936n}\ .$$

- Current Best: $2^{0.388n}$ [Her14].

- For every k, we can solve k-SAT in $2^{(1-\varepsilon_k)n}$, but $\lim_{k\to\infty} \varepsilon_k = 0$.

ETH and SETH

- Can we do significantly better, i.e., find a $2^{o(n)}$ algorithm for solving 3-SAT?

ETH and SETH

- Can we do significantly better, i.e., find a $2^{o(n)}$ algorithm for solving 3-SAT?
- Can we solve SAT in time α^n for $\alpha < 2$?

ETH and SETH

- Can we do significantly better, i.e., find a $2^{o(n)}$ algorithm for solving 3-SAT?
- Can we solve SAT in time α^n for $\alpha < 2$?
- At the moment, we are very very far from answering these questions.

ETH and SETH

- Can we do significantly better, i.e., find a $2^{o(n)}$ algorithm for solving 3-SAT?
- Can we solve SAT in time α^n for $\alpha < 2$?
- At the moment, we are very very far from answering these questions.

Definition (ETH and SETH: Informal Definitions)

ETH: 3-SAT cannot be solved in time $2^{o(n)}$.

SETH: For all $\varepsilon > 0$, there exists $k > 0$ such that k-SAT cannot be solved in $2^{(1-\varepsilon)n}$ time.

ETH and SETH

- Can we do significantly better, i.e., find a $2^{o(n)}$ algorithm for solving 3-SAT?
- Can we solve SAT in time α^n for $\alpha < 2$?
- At the moment, we are very very far from answering these questions.

Definition (ETH and SETH: Informal Definitions)

ETH: 3-SAT cannot be solved in time $2^{o(n)}$.

SETH: For all $\varepsilon > 0$, there exists $k > 0$ such that k-SAT cannot be solved in $2^{(1-\varepsilon)n}$ time.

- Formulated by Impagliazzo, Paturi, and Zane in 2001.
- It is now a fairly standard assumption for fine-grained complexity theory.

Implication for SVP/CVP

- We would like to conclude lower bounds via reductions.
- A reduction from k-SAT to L and a very fast algorithm for L will imply a very fast algorithm for k-SAT.
- Closest Vector Problem
  - Standard NP-Hardness reductions are linear and will give a $2^{\Omega(n)}$ bound under ETH.
  - A recent result showed a lower bound of 2^n for almost all ℓ_p norms under SETH [BGS17].
- Shortest Vector Problem
  - The reduction from [Kho05] is a reduction from 3-SAT on n' variables to SVP on a lattice of rank $n = O(n'^3)$.
  - This implies a $2^{n^{1/3}}$ lower bound for SVP under ETH.
  - Other known NP Hardness reductions likely yield worse results.
  - Desired to find a reduction with $n = O(n')$.

Talk Outline

- A very brief introduction to lattices
- An introduction to the Exponential Time Hypotheses
- Hardness of SVP_p for $p \geq 2.14$ under SETH
- Summary of Other Results
- Conclusions and open questions

A naïve reduction from CVP

- Using [BGS17], there is a lower bound of 2^n under SETH.

A naïve reduction from CVP

- Using [BGS17], there is a lower bound of 2^n under SETH.
- In order to prove hardness of SVP, we want a reduction from CVP to SVP.

A naïve reduction from CVP

- Using [BGS17], there is a lower bound of 2^n under SETH.
- In order to prove hardness of SVP, we want a reduction from CVP to SVP.
- Naïve idea: Given a CVP_p instance $(\mathbf{B}, \vec{t}, r)$, construct the SVP_p instance given by the basis of a lattice $\mathcal{L}^*$ of the form

$$\mathbf{B}^* := \begin{pmatrix} \mathbf{B} & \vec{t} \\ 0 & s \end{pmatrix} ,$$

  for some (small) parameter s (say $s = 1$) and $r^* = (r^p + s^p)^{1/p}$.

A naïve reduction from CVP

- Using [BGS17], there is a lower bound of 2^n under SETH.
- In order to prove hardness of SVP, we want a reduction from CVP to SVP.
- Naïve idea: Given a CVP_p instance $(\mathbf{B}, \vec{t}, r)$, construct the SVP_p instance given by the basis of a lattice $\mathcal{L}^*$ of the form

$$\mathbf{B}^* := \begin{pmatrix} \mathbf{B} & \vec{t} \\ 0 & s \end{pmatrix} ,$$

  for some (small) parameter s (say $s = 1$) and $r^* = (r^p + s^p)^{1/p}$.
- Is this a valid reduction?

A naïve reduction from CVP

- Using [BGS17], there is a lower bound of 2^n under SETH.
- In order to prove hardness of SVP, we want a reduction from CVP to SVP.
- Naïve idea: Given a CVP_p instance $(\mathbf{B}, \vec{t}, r)$, construct the SVP_p instance given by the basis of a lattice $\mathcal{L}^*$ of the form

$$\mathbf{B}^* := \begin{pmatrix} \mathbf{B} & \vec{t} \\ 0 & s \end{pmatrix} ,$$

  for some (small) parameter s (say $s = 1$) and $r^* = (r^p + s^p)^{1/p}$.

- Is this a valid reduction?
  - If the CVP instance is a YES instance ($\|\vec{v} - \vec{t}\|_p \leq r$) then the SVP instance is a YES instance: $(\vec{v} - \vec{t}, -s)$ is a short vector.

A naïve reduction from CVP

- Using [BGS17], there is a lower bound of 2^n under SETH.
- In order to prove hardness of SVP, we want a reduction from CVP to SVP.
- Naïve idea: Given a CVP_p instance $(\mathbf{B}, \vec{t}, r)$, construct the SVP_p instance given by the basis of a lattice $\mathcal{L}^*$ of the form

$$\mathbf{B}^* := \begin{pmatrix} \mathbf{B} & \vec{t} \\ 0 & s \end{pmatrix} ,$$

  for some (small) parameter s (say $s = 1$) and $r^* = (r^p + s^p)^{1/p}$.
- Is this a valid reduction?
  - If the CVP instance is a YES instance ($\|\vec{v} - \vec{t}\|_p \leq r$) then the SVP instance is a YES instance: $(\vec{v} - \vec{t}, -s)$ is a short vector.
  - If CVP instance is a NO instance, there might still be short vectors

$$(\vec{v} - k \cdot \vec{t},\ -k \cdot s)^T$$

  for $\vec{v} \in \mathcal{L}(\mathbf{B})$, $k \neq \pm 1$.

Sparsification Lemma [Khot05]

For prime q, and $\vec{z} \in \mathbb{Z}_q^n$, we write

$$\mathcal{L}_{\vec{z}} = \mathcal{L}_{\mathbf{B},\vec{z},q} := \{\mathbf{B}\vec{y} \in \mathcal{L} \ : \ \vec{y} \in \mathbb{Z}^n \ , \ \langle \vec{z}, \vec{y} \rangle \equiv 0 \bmod q\} \ .$$

Theorem

Let $\vec{z} \in \mathbb{Z}_q^n$ be chosen uniformly at random. Consider lattice vectors $\vec{y}_1, \ldots, \vec{y}_N \in \mathcal{L}$ that are non-zero modulo q. Then,

$$\Pr\left[\forall i > 0,\ \vec{y}_i \notin \mathcal{L}_{\vec{z}}\right] \geq 1 - \frac{N}{q} \ ,$$

Furthermore, if for all distinct $i, j \in [N]$, $\vec{y}_i$ is not an integer multiple of $\vec{y}_j$ modulo q, then

$$\Pr\left[\exists i,\ \vec{y}_i \in \mathcal{L}_{\vec{z}}\right] \geq 1 - \frac{q}{N} \ .$$

Sparsification Lemma [Khot05]

For prime q, and $\vec{z} \in \mathbb{Z}_q^n$, we write

$$\mathcal{L}_{\vec{z}} = \mathcal{L}_{\mathbf{B},\vec{z},q} := \{\mathbf{B}\vec{y} \in \mathcal{L} \ : \ \vec{y} \in \mathbb{Z}^n \ , \ \langle \vec{z}, \vec{y} \rangle \equiv 0 \bmod q\} \ .$$

Theorem

Let $\vec{z} \in \mathbb{Z}_q^n$ be chosen uniformly at random. Consider lattice vectors $\vec{y}_1, \ldots, \vec{y}_N \in \mathcal{L}$ that are non-zero modulo q. Then,

$$\Pr\left[\forall i > 0,\ \vec{y}_i \notin \mathcal{L}_{\vec{z}}\right] \geq 1 - \frac{N}{q} \ ,$$

Furthermore, if for all distinct $i, j \in [N]$, $\vec{y}_i$ is not an integer multiple of $\vec{y}_j$ modulo q, then

$$\Pr\left[\exists i,\ \vec{y}_i \in \mathcal{L}_{\vec{z}}\right] \geq 1 - \frac{q}{N} \ .$$

i.e., if $N \ll q$, then w.h.p. none of the vectors is in $\mathcal{L}_{\vec{z}}$,

and if $N \gg q$, then w.h.p. one of the vectors is in $\mathcal{L}_{\vec{z}}$.

How does the sparsification lemma help

- Given the CVP instance, we construct a lattice $\mathcal{L}^*$ and choose $r^* > 0$, such that $N_{\text{YES}} \gg N_{\text{NO}}$, where
  - N_{YES} is a lower bound on the number of vectors in $\mathcal{L}^*$ of length at most r^* if the input instance is a YES instance.
  - N_{NO} is an upper bound on the number of vectors in $\mathcal{L}^*$ of length at most r^* if the input instance is a NO instance.

How does the sparsification lemma help

- Given the CVP instance, we construct a lattice $\mathcal{L}^*$ and choose $r^* > 0$, such that $N_{\text{YES}} \gg N_{\text{NO}}$, where
  - N_{YES} is a lower bound on the number of vectors in $\mathcal{L}^*$ of length at most r^* if the input instance is a YES instance.
  - N_{NO} is an upper bound on the number of vectors in $\mathcal{L}^*$ of length at most r^* if the input instance is a NO instance.
- We then choose $q \approx \sqrt{N_{\text{YES}} \cdot N_{\text{NO}}}$ and sparsify the lattice.

Modifying the naïve reduction

Consider the CVP instance $(\mathbf{B}, \vec{t}, r)$ from [BGS17]. It has the form

$$\mathbf{B} = \begin{pmatrix} \Phi \\ I_n \end{pmatrix} \in \mathbb{R}^{d \times n}, \qquad \vec{t} = \begin{pmatrix} \vec{t}_1 \\ 1/2 \\ \vdots \\ 1/2 \end{pmatrix} \in \mathbb{R}^{d},$$

for some $\Phi \in \mathbb{R}^{(d-n)\times n}$, $\vec{t}_1 \in \mathbb{R}^{d-n}$, and $r = \frac{(n+1)^{1/p}}{2}$.

Consider the lattice basis obtained by adding the gadget lattice $\mathbb{Z}^{n^\dagger}$.

$$\mathbf{B}^* = \begin{pmatrix} \mathbf{B} & \mathbf{0} & \vec{t} \\ \mathbf{0} & \mathbb{Z}^{n^\dagger} & \vec{t}^\dagger \\ \mathbf{0} & \mathbf{0} & s \end{pmatrix} \in \mathbb{R}^{(d+n^\dagger)\times(n+n^\dagger+1)},$$

where $\vec{t}^\dagger = (1/2, \ldots, 1/2) \in \mathbb{R}^{n^\dagger}$, and $r^* = \left(r^p + \frac{n^\dagger}{2^p} + s^p\right)^{1/p}$.

Recall

- Given the CVP instance, we wanted to construct a lattice $\mathcal{L}^*$ and choose $r^* > 0$, such that $N_{\text{YES}} \gg N_{\text{NO}}$, where
  - N_{YES} is a lower bound on the number of vectors in $\mathcal{L}^*$ of length at most r^* if the input instance is a YES instance.
  - N_{NO} is an upper bound on the number of vectors in $\mathcal{L}^*$ of length at most r^* if the input instance is a NO instance.
- We then choose $q \approx \sqrt{N_{\text{YES}} \cdot N_{\text{NO}}}$ and sparsify the lattice.

Our reduction

We have constructed the lattice basis

$$\mathbf{B}^* = \begin{pmatrix} \Phi & \mathbf{0} & \vec{t}_1 \\ \mathbb{Z}^n & \mathbf{0} & \vec{t}_2 \\ \mathbf{0} & \mathbb{Z}^{n^\dagger} & \vec{t}^\dagger \\ \mathbf{0} & \mathbf{0} & s \end{pmatrix},$$

where $\vec{t}_2 = (1/2, \ldots, 1/2) \in \mathbb{R}^n$, $\vec{t}^\dagger = (1/2, \ldots, 1/2) \in \mathbb{R}^{n^\dagger}$, and

$r^* = \left(r^p + \frac{n^\dagger}{2^p} + s^p\right)^{1/p} \approx \frac{(n+n^\dagger)^{1/p}}{2}$.

Our reduction

We have constructed the lattice basis

$$\mathbf{B}^* = \begin{pmatrix} \Phi & \mathbf{0} & \vec{t}_1 \\ \mathbb{Z}^n & \mathbf{0} & \vec{t}_2 \\ \mathbf{0} & \mathbb{Z}^{n^\dagger} & \vec{t}^\dagger \\ \mathbf{0} & \mathbf{0} & s \end{pmatrix},$$

where $\vec{t}_2 = (1/2, \ldots, 1/2) \in \mathbb{R}^n$, $\vec{t}^\dagger = (1/2, \ldots, 1/2) \in \mathbb{R}^{n^\dagger}$, and

$$r^* = \left(r^p + \frac{n^\dagger}{2^p} + s^p\right)^{1/p} \approx \frac{(n+n^\dagger)^{1/p}}{2}.$$

Clearly, $N_{\text{YES}} \geq 2^{n^\dagger}$ (Choose $0/1$ coefficients in the gadget lattice).

Our reduction

We have constructed the lattice basis

$$\mathbf{B}^* = \begin{pmatrix} \Phi & \mathbf{0} & \vec{t}_1 \\ \mathbb{Z}^n & \mathbf{0} & \vec{t}_2 \\ \mathbf{0} & \mathbb{Z}^{n^\dagger} & \vec{t}^\dagger \\ \mathbf{0} & \mathbf{0} & s \end{pmatrix} ,$$

where $\vec{t}_2 = (1/2, \ldots, 1/2) \in \mathbb{R}^n$, $\vec{t}^\dagger = (1/2, \ldots, 1/2) \in \mathbb{R}^{n^\dagger}$, and

$r^* = \left(r^p + \frac{n^\dagger}{2^p} + s^p\right)^{1/p} \approx \frac{(n+n^\dagger)^{1/p}}{2}$.

Clearly, $N_{\mathrm{YES}} \geq 2^{n^\dagger}$ (Choose $0/1$ coefficients in the gadget lattice).

Also, we can show that

$$N_{\mathrm{NO}} \leq \mathrm{poly}(n) \cdot N_p\left(\mathbb{Z}^{n+n^\dagger} , \frac{(n+n^\dagger)^{1/p}}{2}\right) ,$$

where $N_p(\mathcal{L}, r)$ denotes the number of vectors of length at most r in $\mathcal{L}$.

Our reduction

We have constructed the lattice basis

$$\mathbf{B}^* = \begin{pmatrix} \Phi & \mathbf{0} & \vec{t}_1 \\ \mathbb{Z}^n & \mathbf{0} & \vec{t}_2 \\ \mathbf{0} & \mathbb{Z}^{n^\dagger} & \vec{t}^\dagger \\ \mathbf{0} & \mathbf{0} & s \end{pmatrix} ,$$

where $\vec{t}_2 = (1/2, \ldots, 1/2) \in \mathbb{R}^n$, $\vec{t}^\dagger = (1/2, \ldots, 1/2) \in \mathbb{R}^{n^\dagger}$, and

- The last coefficient k odd is “like" $k = 1$ and does not give a vector of length less than r^* since it is a NO instance.
- The last coefficient k even is “like" $k = 0$, and only contributes for $|k| < \text{poly}(n)$.

$$N_{\text{NO}} \leq \text{poly}(n) \cdot N_p\left(\mathbb{Z}^{n+n^\dagger} , \frac{(n + n^\dagger)^{1/p}}{2}\right) ,$$

where $N_p(\mathcal{L}, r)$ denotes the number of vectors of length at most r in $\mathcal{L}$.

Our reduction

We have constructed the lattice basis

$$\mathbf{B}^* = \begin{pmatrix} \Phi & \mathbf{0} & \vec{t}_1 \\ \mathbb{Z}^n & \mathbf{0} & \vec{t}_2 \\ \mathbf{0} & \mathbb{Z}^{n^\dagger} & \vec{t}^\dagger \\ \mathbf{0} & \mathbf{0} & s \end{pmatrix},$$

where $\vec{t}_2 = (1/2, \ldots, 1/2) \in \mathbb{R}^n$, $\vec{t}^\dagger = (1/2, \ldots, 1/2) \in \mathbb{R}^{n^\dagger}$, and

$r^* = \left(r^p + \frac{n^\dagger}{2^p} + s^p\right)^{1/p} \approx \frac{(n+n^\dagger)^{1/p}}{2}$.

Clearly, $N_{\text{YES}} \geq 2^{n^\dagger}$ (Choose $0/1$ coefficients in the gadget lattice).

Also, we can show that

$$N_{\text{NO}} \leq \text{poly}(n) \cdot N_p\left(\mathbb{Z}^{n+n^\dagger}, \frac{(n+n^\dagger)^{1/p}}{2}\right),$$

where $N_p(\mathcal{L}, r)$ denotes the number of vectors of length at most r in $\mathcal{L}$.

We need to bound $N_p\left(\mathbb{Z}^{n+n^\dagger}, \frac{(n+n^\dagger)^{1/p}}{2}\right)$ by $2^{n^\dagger}$.

Finishing the proof

Let $m = n + n^{\dagger}$. We need to bound $N_p\left(\mathbb{Z}^m, \frac{m^{1/p}}{2}\right)$. As an example, consider $p = 2$. Then, any vector with $m/4$ ± 1's, and $3m/4$ 0's has norm $\sqrt{m}/2$.

Finishing the proof

Let $m = n + n^{\dagger}$. We need to bound $N_p\left(\mathbb{Z}^m, \frac{m^{1/p}}{2}\right)$. As an example, consider $p = 2$. Then, any vector with $m/4$ ± 1's, and $3m/4$ 0's has norm $\sqrt{m}/2$. Thus,

$$N_2\left(\mathbb{Z}^m, \frac{\sqrt{m}}{2}\right) \geq \binom{m}{m/4} \cdot 2^{m/4} > 2.086^m > 2^{n^{\dagger}} .$$

Finishing the proof

Let $m = n + n^{\dagger}$. We need to bound $N_p\left(\mathbb{Z}^m, \frac{m^{1/p}}{2}\right)$. As an example, consider $p = 2$. Then, any vector with $m/4$ ± 1's, and $3m/4$ 0's has norm $\sqrt{m}/2$. Thus,

$$N_2\left(\mathbb{Z}^m, \frac{\sqrt{m}}{2}\right) \geq \binom{m}{m/4} \cdot 2^{m/4} > 2.086^m > 2^{n^{\dagger}} .$$

- The above is a reasonable estimate of $N_2\left(\mathbb{Z}^m, \frac{m^{1/p}}{2}\right)$. We show in the paper that $N_2\left(\mathbb{Z}^m, \frac{m^{1/2}}{2}\right) \approx 2.089^m$.

Finishing the proof

Let $m = n + n^\dagger$. We need to bound $N_p\left(\mathbb{Z}^m, \frac{m^{1/p}}{2}\right)$. As an example, consider $p = 2$. Then, any vector with $m/4$ ± 1's, and $3m/4$ 0's has norm $\sqrt{m}/2$. Thus,

$$N_2\left(\mathbb{Z}^m, \frac{\sqrt{m}}{2}\right) \geq \binom{m}{m/4} \cdot 2^{m/4} > 2.086^m > 2^{n^\dagger} .$$

- The above is a reasonable estimate of $N_2\left(\mathbb{Z}^m, \frac{m^{1/p}}{2}\right)$. We show in the paper that $N_2\left(\mathbb{Z}^m, \frac{m^{1/2}}{2}\right) \approx 2.089^m$.
- It is easy to see that $N_p\left(\mathbb{Z}^m, \frac{m^{1/p}}{2}\right)$ decreases with increase in p.

Finishing the proof

Let $m = n + n^\dagger$. We need to bound $N_p\left(\mathbb{Z}^m\ ,\ \frac{m^{1/p}}{2}\right)$. As an example, consider $p = 2$. Then, any vector with $m/4$ ± 1's, and $3m/4$ 0's has norm $\sqrt{m}/2$. Thus,

$$N_2\left(\mathbb{Z}^m\ ,\ \frac{\sqrt{m}}{2}\right) \geq \binom{m}{m/4} \cdot 2^{m/4} > 2.086^m > 2^{n^\dagger}\ .$$

- The above is a reasonable estimate of $N_2\left(\mathbb{Z}^m\ ,\ \frac{m^{1/p}}{2}\right)$. We show in the paper that $N_2\left(\mathbb{Z}^m\ ,\ \frac{m^{1/2}}{2}\right) \approx 2.089^m$.
- It is easy to see that $N_p\left(\mathbb{Z}^m\ ,\ \frac{m^{1/p}}{2}\right)$ decreases with increase in p.
- So, we expect $N_p\left(\mathbb{Z}^m\ ,\ \frac{m^{1/p}}{2}\right) \ll 2^m$, for a large enough p. If this is true, then we can choose $n^\dagger = C^\dagger n$ for a large enough constant $C^\dagger$ to get

$$N_p\left(\mathbb{Z}^{n+n^\dagger}\ ,\ \frac{(n+n^\dagger)^{1/p}}{2}\right) \ll 2^{n^\dagger}$$

Estimating $N_p\left(\mathbb{Z}^m, \frac{m^{1/p}}{2}\right)$

For any $\tau > 0$, we define

$$\Theta_p(\tau) := \sum_{z \in \mathbb{Z}} \exp(-\tau |z|^p) .$$

Notice that we can write $\Theta_p(\tau)^m$ as a summation over $\mathbb{Z}^m$,

$$\Theta_p(\tau)^m = \sum_{\vec{z} \in \mathbb{Z}^m} \exp(-\tau \|\vec{z}\|_p^p) .$$

In particular, for any radius $r > 0$ and $\tau > 0$, we have

$$\Theta_p(\tau)^m \geq \sum_{\substack{\vec{z} \in \mathbb{Z}^m \\ \|\vec{z}\|_p \leq r}} \exp(-\tau \|\vec{z}\|_p^p) \geq \exp(-\tau r^p) \cdot N_p(\mathbb{Z}^m, r, \vec{0}) .$$

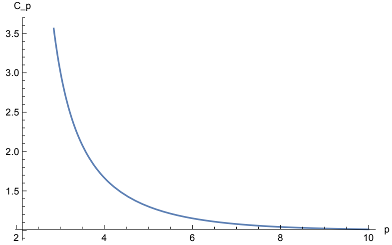
Rearranging and taking the minimum over all $\tau > 0$, we see that

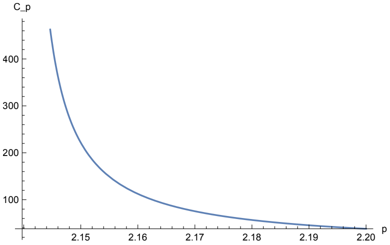
$$N_p(\mathbb{Z}^m, r) \leq \min_{\tau > 0} \exp(\tau r^p) \cdot \Theta_p(\tau)^m .$$

We show that this bound is quite tight. We cannot compute this analytically, but can estimate this numerically to any precision.

The final result: SETH Hardness

We get that for “almost" all $p \geq 2.14$, under randomized SETH, there is no algorithm for SVP_p that runs in time better than $2^{n/C_p}$. The following shows the dependence of C_p on p.





Talk Outline

- A very brief introduction to lattices
- An introduction to the Exponential Time Hypotheses
- Hardness of SVP_p for $p \geq 2.14$ under SETH
- Summary of Other Results
- Conclusions and open questions

Gap-ETH Hardness

Max-3-SAT$_\eta$: This is a promise problem. Given a formula ϕ in 3-CNF with n variables and m clauses

- YES instance: There is a satisfying assignment
- NO instance: Every assignment satisfies at most η fraction of the clauses.

Gap-ETH Hardness

Max-3-SAT$_\eta$: This is a promise problem. Given a formula ϕ in 3-CNF with n variables and m clauses

- YES instance: There is a satisfying assignment
- NO instance: Every assignment satisfies at most η fraction of the clauses.

The following definition is due to [MR16,Din16]. It is fast becoming a standard assumption.

Definition (Gap-ETH: Informal Definition)

Gap-ETH: There exist $\eta \in (0, 1)$ such that Max-3-SAT$_\eta$ cannot be solved in time $2^{o(n)}$.

Our Results under Gap-ETH

- For any $p > 2$, there is no $2^{o(n)}$-time algorithm for SVP_p under Gap-ETH Assumption.
  - For this, we show that for any $p > 2$, there exists a vector $\vec{t}$ and $r > 0$ such that
$$N_p(\mathbb{Z}^n, \vec{t}, r) \geq exp(n) \cdot N_p(\mathbb{Z}^n, \vec{0}, r) \ .$$
- There is no $2^{o(n)}$-time algorithm for SVP_2 under Gap-ETH Assumption and the assumption that there exists a family of lattices with exponential kissing number.

Our Results under Gap-ETH

- For any $p > 2$, there is no $2^{o(n)}$-time algorithm for SVP_p under Gap-ETH Assumption.
  - For this, we show that for any $p > 2$, there exists a vector $\vec{t}$ and $r > 0$ such that

$$N_p(\mathbb{Z}^n, \vec{t}, r) \geq exp(n) \cdot N_p(\mathbb{Z}^n, \vec{0}, r) \ .$$

- There is no $2^{o(n)}$-time algorithm for SVP_2 under Gap-ETH Assumption and the assumption that there exists a family of lattices with exponential kissing number.
  - For this, we show that if there is a family of lattices with exponential kissing number, then for any n, there exists an n-dimensional lattice $\mathcal{L}$, a vector $\vec{t}$, and $r > 0$ such that

$$N_2(\mathcal{L}, \vec{t}, r) \geq exp(n) \cdot N_2(\mathcal{L}, \vec{0}, r) \ .$$

Talk Outline

- A very brief introduction to lattices
- An introduction to the Exponential Time Hypotheses
- Hardness of SVP_p for $p \geq 2.14$ under SETH
- Summary of Other Results
- Conclusions and open questions

Conclusions and Open Questions

- Under SETH, we show that for “almost" all p, SVP_p cannot be solved in $2^{n/C_p}$ time.
  - Question 1: Improve the constant C_p, possibly by using a different gadget lattice.
  - Question 2: Remove the “almost", possibly via a direct reduction from k-SAT.

Conclusions and Open Questions

- Under SETH, we show that for "almost" all p, SVP_p cannot be solved in $2^{n/C_p}$ time.
  - Question 1: Improve the constant C_p, possibly by using a different gadget lattice.
  - Question 2: Remove the "almost", possibly via a direct reduction from k-SAT.
- Under Gap-ETH, we show that for all $p > 2$, SVP_p cannot be solved in $2^{o(n)}$ time.
  - Question 3: Can we show this under the more standard ETH.

Conclusions and Open Questions

- Under SETH, we show that for “almost" all p, SVP_p cannot be solved in $2^{n/C_p}$ time.
  - Question 1: Improve the constant C_p, possibly by using a different gadget lattice.
  - Question 2: Remove the “almost", possibly via a direct reduction from k-SAT.
- Under Gap-ETH, we show that for all $p > 2$, SVP_p cannot be solved in $2^{o(n)}$ time.
  - Question 3: Can we show this under the more standard ETH.
- Under Gap-ETH and the assumption that the lattice has exponential kissing number, we show that SVP_2 cannot be solved in $2^{o(n)}$ time.
  - Question 4: Replace Gap-ETH with ETH.
  - Question 5: Remove the assumption about exponential kissing number.

Questions?