Why Learning With Error, Homomorphic Encryption

Peizhao Hu
Rochester Institute of Technology
Lattice space, structure, points
Lattice bases

Bases, $b_1$ and $b_2$, also known as the shortest vectors
From base vectors to Lattice

Base vectors, $b_1$ and $b_2$
Lattices in different from

Base vectors, $b_1$ and $b_2$

Easy lattice

Reasonable lattice

Difficult lattice
Given two base vectors, \((b_1, b_2)\), we can generate two lattice points \((p_1, p_2)\).

Given two lattice points \((p_1, p_2)\), find the two base vectors, \((b_1, b_2)\), that can contain \((p_1, p_2)\).
Hardness problems

Shortest Vector Problem (SVP)

Given two base vectors, $(b_1, b_2)$, we can generate two lattice points $(p_1, p_2)$.

Given two lattice points $(p_1, p_2)$, find the two base vectors, $(b_1, b_2)$, that can contain $(p_1, p_2)$. 
Short Vector Problem (SVP)

Find the two base vectors, \((b_1, b_2)\), that can contain \((p_1, p_2)\).

Given two lattice points \((p_1, p_2)\)
Lattices in different forms

Base vectors, $b_1$ and $b_2$

Easy lattice

Reasonable lattice

Difficult lattice
Brute-force approach

• Try different combinations of base vectors, \((b_1, b_2)\).
  • **While loop**
    • *Generate two random vectors*
    • *Construct a Lattice \(L\)*
    • *Test whether two lattice points \((p_1, p_2)\) are in \(L\)*
    • *Stop if yes*

• **Not very safe** if we can do brute-force faster!!!

• Similar in trying to break RSA where the modulo \(N=p \cdot q\), where \(p\) and \(q\) are two large (typically 2048-bit) primes.
  • The security of RSA is based the difficulty of factorizing a product of two large primes.
Making SVP harder (Post-Quantum)

Given two base vectors, \((b_1, b_2)\), we can generate two lattice points \((p_1, p_2)\).

**Standard SVP**

**Harder SVP**

Given two base vectors, \((b_1, b_2)\) and *small* noise \(e\), we can generate two lattice points \((p_1, p_2)\).
Generating two new vector points \((p_1, p_2)\)

Given two lattice points \((p_1, p_2)\) that have some error \(e\), find the two base vectors, \((b_1, b_2)\), which determine a lattice containing \((p_1, p_2)\).

Computationally indistinguishable from other points because of the small error \(e\)
SVP vs. Closest Vector Problem (CVP)

Shortest Vector Problem (SVP)
Given two lattice points \((p_1, p_2)\) that have some error \(e\), find the two base vectors, \((b_1, b_2)\), which determine a lattice containing \((p_1, p_2)\).

Closest Vector Problem (CVP)
Given two lattice points \((p_1, p_2)\) that have some error \(e\), find the closest vector, \(p\), that is on the lattice.
Example in Algebra

Given the following equations, find the values of X, Y, and Z.

\[33423X + 12693Y + 24634Z = 12921315\]
\[12236X - 79375Y + 12324Z = 47230832\]
\[76322X - 11111Y - 48918Z = 91873693\]
Example in Algebra

Given the following equations, find the values of X, Y, and Z.

\[
\begin{align*}
33423X + 12693Y + 24634Z & \approx 12921323 \\
12236X - 79375Y + 12324Z & \approx 47230822 \\
76322X - 11111Y - 48918Z & \approx 91873663
\end{align*}
\]

Note, numbers on the right have been masked with small error e.
A bit of HE Crypto

- **KeyGen(λ)**
  - Secret key, \( s \)
  - Public key, \(( -A, A \cdot s + t \cdot e )\)
- **Enc(m)**
  - Ciphertext, \( \text{Ctxt} = (-A, A \cdot s + t \cdot e + m) \)
- **Dec(\text{ctxt}, s)**
  - Message, \( m' = -A \cdot s + A \cdot s + t \cdot e + m \pmod{t} \)
- **Add(\text{ctxt}_1, \text{ctxt}_2)**
  - Sum, \( \text{Ctxt}' = (-A, A \cdot s + t \cdot e_1 + m_1) + (-A, A \cdot s + t \cdot e_2 + m_2) \)
    = \((-2A, 2A \cdot s + t \cdot (e_1 + e_2) + (m_1 + m_2))\)
- **Multi(\text{ctxt}_1, \text{ctxt}_2)** works in similar way but requires error reduction.