Lattices and Hard Problems

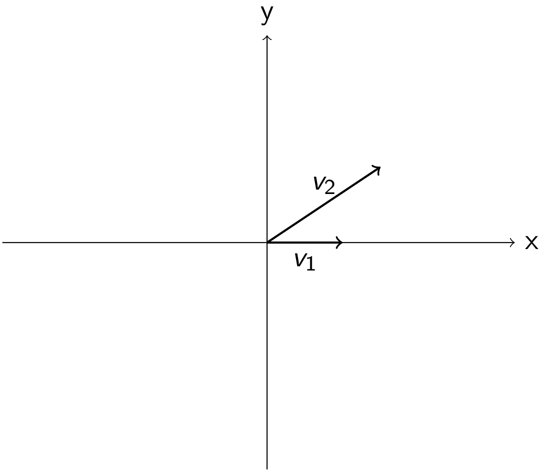
Robert Brede

February 23, 2023

Outline

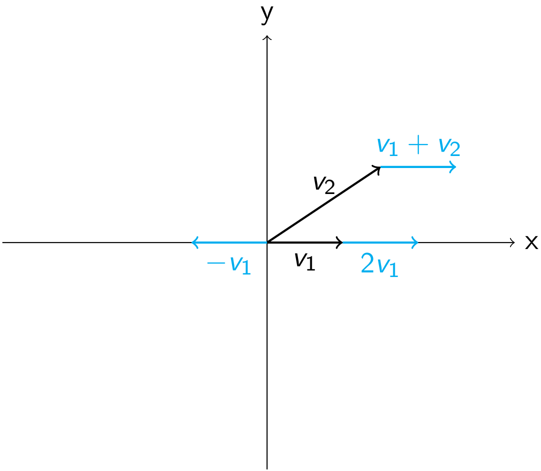
- Lattices
- Shortest Vector Problem, Closest Vector Problem
  - LLL - Algorithm
- Learning With Errors
  - Learning Parity with Noise
  - Reduction LWE to CVP
  - Usage of LWE

Lattices - Definition



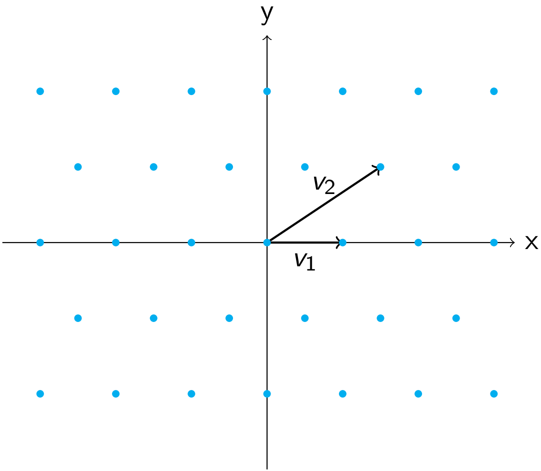
Generating Vectors v_1, v_2

Lattices - Definition



Linear combinations with integer coefficients

Lattices - Definition



Lattice spanned by v_1 and v_2

Lattices - Formal Definition

- Generating Vectors $\{v_1, \ldots, v_m\} \subset \mathbb{R}^n$
- Lattice $\Lambda = \{\sum_{i=1}^{m} x_i v_i \mid x_i \in \mathbb{Z}\}$

Lattices - Discret

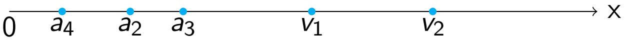
- There is smallest distance $\epsilon > 0$ between Points

Lattices - Discret

- There is smallest distance $\epsilon > 0$ between Points
- Counterexample: $v_1 = (1), v_2 = (\sqrt{2}) \in \mathbb{R}^1$

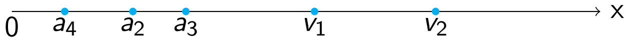
Lattices - Discret

- There is smallest distance $\epsilon > 0$ between Points
- Counterexample: $v_1 = (1), v_2 = (\sqrt{2}) \in \mathbb{R}^1$
- $a_n = |a_{n-2} - a_{n-1}|$, $a_0 = v_1$, $a_1 = v_2$



Lattices - Discret

- There is smallest distance $\epsilon > 0$ between Points
- Counterexample: $v_1 = (1), v_2 = (\sqrt{2}) \in \mathbb{R}^1$
- $a_n = |a_{n-2} - a_{n-1}|,\ a_0 = v_1,\ a_1 = v_2$



- Converges to 0
- No ϵ exists

Lattices - Properties

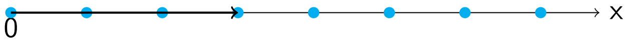
- Basis $B = \{b_1, ..., b_d\} \subset \mathbb{R}^n$ linear independent
- Lattice $\Lambda = \left\{\sum_{i=1}^{d} x_i b_i \mid x_i \in \mathbb{Z}\right\}$

Lattices - Properties

- Basis $B = \{b_1, ..., b_d\} \subset \mathbb{R}^n$ linear independent
- Lattice $\Lambda = \left\{\sum_{i=1}^{d} x_i b_i \mid x_i \in \mathbb{Z}\right\}$
- Similar to vector subspace
  - Each point unique combination of basis vectors
  - Rank of lattice $d \leq n$, full rank $d = n$
  - Change of basis Matrix B
  - But: for given Lattice Λ, cannot take any d linear independent vectors $\{b_1, ..., b_d\} \subset \Lambda$

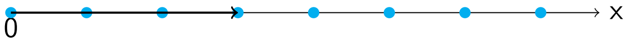
Lattices - Properties

- Basis $B = \{b_1, ..., b_d\} \subset \mathbb{R}^n$ linear independent
- Lattice $\Lambda = \left\{ \sum_{i=1}^{d} x_i b_i \mid x_i \in \mathbb{Z} \right\}$
- Similar to vector subspace
  - Each point unique combination of basis vectors
  - Rank of lattice $d \leq n$, full rank $d = n$
  - Change of basis Matrix B
  - But: for given Lattice Λ, cannot take any d linear independent vectors $\{b_1, ..., b_d\} \subset \Lambda$



Lattices - Properties

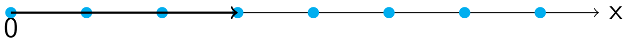
- Basis $B = \{b_1, ..., b_d\} \subset \mathbb{R}^n$ linear independent
- Lattice $\Lambda = \left\{\sum_{i=1}^{d} x_i b_i \mid x_i \in \mathbb{Z}\right\}$
- Similar to vector subspace
  - Each point unique combination of basis vectors
  - Rank of lattice $d \leq n$, full rank $d = n$
  - Change of basis Matrix B
  - But: for given Lattice Λ, cannot take any d linear independent vectors $\{b_1, ..., b_d\} \subset \Lambda$



- Dual lattice $\Lambda^* = \{y \in \mathbb{R}^n \mid \forall v \in \Lambda : \langle v, y \rangle \in \mathbb{Z}\}$

Lattices - Properties

- Basis $B = \{b_1, ..., b_d\} \subset \mathbb{R}^n$ linear independent
- Lattice $\Lambda = \left\{ \sum_{i=1}^{d} x_i b_i \mid x_i \in \mathbb{Z} \right\}$
- Similar to vector subspace
  - Each point unique combination of basis vectors
  - Rank of lattice $d \leq n$, full rank $d = n$
  - Change of basis Matrix B
  - But: for given Lattice Λ, cannot take any d linear independent vectors $\{b_1, ..., b_d\} \subset \Lambda$



- Dual lattice $\Lambda^* = \{y \in \mathbb{R}^n \mid \forall v \in \Lambda : \langle v, y \rangle \in \mathbb{Z}\}$
- Closed, countable, bounded subset S $\rightarrow$ L $\cap$ S finite
- λ_1: shortest distance between any two lattic points

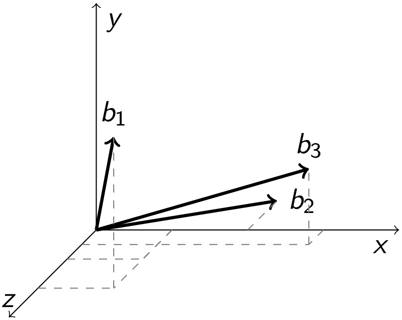
Closest Vector Problem (CVP) and Shortest Vector Problem (SVP)

Given a lattice basis B, find:

- SVP: the shortest nonzero lattice point
- CVP: the closest lattice point for a given point in $\mathbb{R}^n$

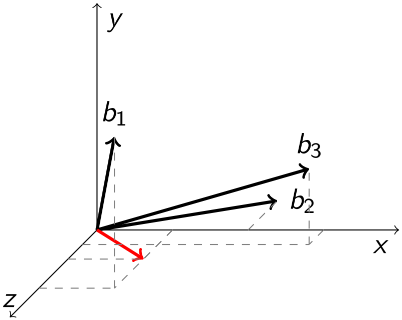
SVP - Example

$$b_1 = \begin{pmatrix} 1 \\ 2 \\ 2 \end{pmatrix}, b_2 = \begin{pmatrix} 2 \\ 0 \\ -1 \end{pmatrix}, b_3 = \begin{pmatrix} 3 \\ 1 \\ 0.5 \end{pmatrix}$$



SVP - Example

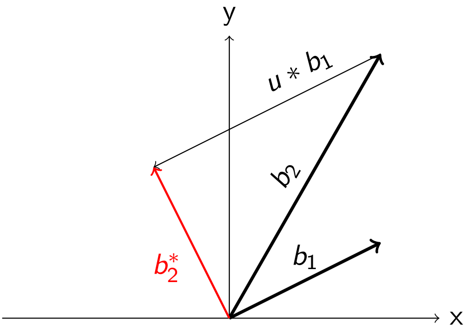
$$b_1 = \begin{pmatrix} 1 \\ 2 \\ 2 \end{pmatrix}, b_2 = \begin{pmatrix} 2 \\ 0 \\ -1 \end{pmatrix}, b_3 = \begin{pmatrix} 3 \\ 1 \\ 0.5 \end{pmatrix}, 2b_3 - 2b_2 - b_1 = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$$



LLL Algorithm

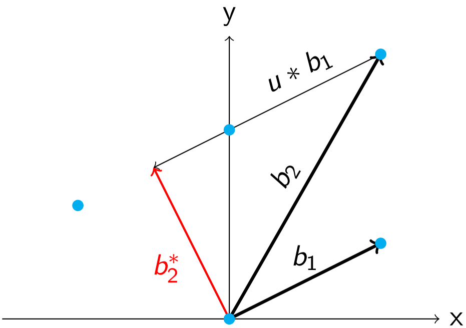
- Named after creators Lenstra, Lenstra, Lovász
- Approximates SVP
- Tries to calculate short orthogonal basis
- Adaptation of Gram-Schmidt algorithm

Gram-Schmidt Algorithm



- Orthogonalize $\{b_1, b_2\}$ to $\{b_1^*, b_2^*\}$
- Normalize

Gram-Schmidt - On Lattice



- If $u \notin \mathbb{Z}$, then $b_2^* \notin L$
- Normalized Vector Generally not in L

LLL Algorithm - Goal

LLL reduced latice basis $\{b_1, \ldots, b_n\}$ with $\{b_1^*, \ldots, b_n^*\}$ from Gram-Schmidt

- $\forall i \neq k : u_{i,k} = \frac{\langle b_k, b_i^* \rangle}{\langle b_i^*, b_i^* \rangle} \leq \frac{1}{2}$
- $\forall i : ||b_{i+1}^* + u_{i,i+1} b_i^*||^2 \geq \delta ||b_i^*||^2 \quad (\frac{1}{4} \leq \delta \leq 1)$

LLL Algorithm - Goal

LLL reduced latice basis $\{b_1, \ldots, b_n\}$ with $\{b_1^*, \ldots, b_n^*\}$ from Gram-Schmidt

- $\forall i \neq k : u_{i,k} = \frac{\langle b_k, b_i^* \rangle}{\langle b_i^*, b_i^* \rangle} \leq \frac{1}{2}$
- $\forall i : ||b_{i+1}^* + u_{i,i+1} b_i^*||^2 \geq \delta ||b_i^*||^2 \quad (\frac{1}{4} \leq \delta \leq 1)$

- Swapping b_i with $b_{i+1} \rightarrow b_{i,new}^* = b_{i+1}^* + u_{i,i+1} b_i^*$

LLL Algorithm - Goal

LLL reduced latice basis $\{b_1, \ldots, b_n\}$ with $\{b_1^*, \ldots, b_n^*\}$ from Gram-Schmidt

- $\forall i \neq k : u_{i,k} = \frac{\langle b_k, b_i^* \rangle}{\langle b_i^*, b_i^* \rangle} \leq \frac{1}{2}$
- $\forall i : ||b_{i+1}^* + u_{i,i+1} b_i^*||^2 \geq \delta ||b_i^*||^2 \quad (\frac{1}{4} \leq \delta \leq 1)$

- Swapping b_i with $b_{i+1} \rightarrow b_{i,new}^* = b_{i+1}^* + u_{i,i+1} b_i^*$
- Condition ensures not much smaller base when swapping

LLL Algorithm - Pseudocode

- Step 1: Gram-Schmidt with rounded $u_{i,k}$
- Step 2: If second condition not satisfied for index i, swap b_i and b_{i+1} and return to step 1

Learning Parity with Noise

- Find $s \in \{0,1\}^n$
- Given pairs $(a_i, b_i = \langle a_i, s \rangle + e_i \mod 2)$
- $a_i \in \{0,1\}^n$ uniformly chosen
- $e_i = 1$ with probability $\epsilon \in (0,1)$

Learning Parity with Noise

- Find $s \in \{0,1\}^n$
- Given pairs $(a_i, b_i = \langle a_i, s\rangle + e_i \mod 2)$
- $a_i \in \{0,1\}^n$ uniformly chosen
- $e_i = 1$ with probability $\epsilon \in (0,1)$
- Can be written $b = As + e$, with

$$A = \begin{bmatrix} a_1^T \\ a_2^T \\ \vdots \\ a_m^T \end{bmatrix}, \quad e = \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_m \end{pmatrix}, b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}$$

Learning Parity with Noise - Example

$$
\begin{aligned}
1s_0 + 0s_1 + 1s_2 &= 1 \mod 2 \\
1s_0 + 1s_1 + 0s_2 &= 0 \mod 2 \\
1s_0 + 1s_1 + 1s_2 &= 1 \mod 2
\end{aligned}
$$

Learning Parity with Noise - Example

$$\begin{aligned} 1s_0 + 0s_1 + 1s_2 &= 1 \mod 2 \\ 1s_0 + 1s_1 + 0s_2 &= 0 \mod 2 \\ 1s_0 + 1s_1 + 1s_2 &= 1 \mod 2 \end{aligned}$$

Likely Solution:

$$s = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

Learning Parity with Noise - Example

$$\begin{aligned} 1s_0 + 0s_1 + 1s_2 &= 1 \mod 2 \\ 1s_0 + 1s_1 + 0s_2 &= 0 \mod 2 \\ 1s_0 + 1s_1 + 1s_2 &= 1 \mod 2 \end{aligned}$$

Other Solution:

$$s = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

Gaussian Elimination

Look at only calculating first bit of the secret

- Use n equations with a_i adding to $(1, 0, \ldots, 0)$
- Add equations

Gaussian Elimination

Example with $\epsilon = \frac{1}{4}$

$$
\begin{aligned}
1s_0 + 0s_1 + 1s_2 &= 1 \mod 2 \\
&+ \\
1s_0 + 1s_1 + 0s_2 &= 0 \mod 2 \\
&+ \\
1s_0 + 1s_1 + 1s_2 &= 1 \mod 2 \\
&= \\
s_0 &= 0 \mod 2
\end{aligned}
$$

Gaussian Elimination

Example with $\epsilon = \frac{1}{4}$

$$\begin{aligned} 1s_0 + 0s_1 + 1s_2 &= 1 \quad \bmod 2 \\ &+ \\ 1s_0 + 1s_1 + 0s_2 &= 0 \quad \bmod 2 \\ &+ \\ 1s_0 + 1s_1 + 1s_2 &= 1 \quad \bmod 2 \\ &= \\ s_0 &= 0 \quad \bmod 2 \end{aligned}$$

Chance of Success:

$$\left(\frac{3}{4}\right)^3 + 3 * \left(\frac{1}{4}\right)^2 \cdot \frac{3}{4} = 0.5625$$

Learning with Errors (LWE)

- Generalization of Learning Parity with Noise
  - $\mathbb{Z}_q = \{0, 1, \ldots, q-1\}$ Instead of $\{0, 1\}$
  - Error $e_i \in \mathbb{Z}_q$ chosen from distribution χ

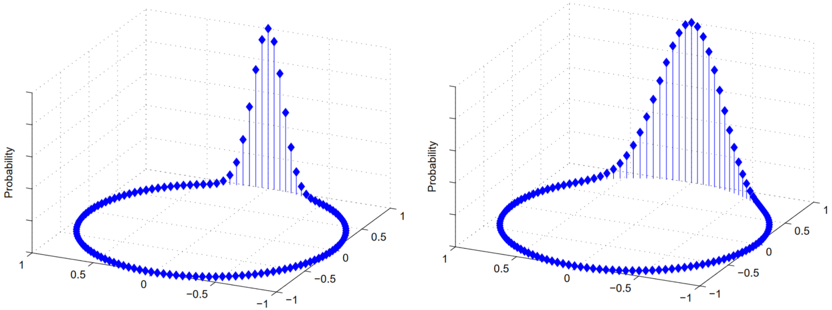
Learning with Errors (LWE)

- Generalization of Learning Parity with Noise
  - $\mathbb{Z}_q = \{0, 1, \ldots, q-1\}$ Instead of $\{0, 1\}$
  - Error $e_i \in \mathbb{Z}_q$ chosen from distribution χ
- Given pairs $(a_i, \langle a_i, s\rangle + e_i \mod q)$
- $a_i \in \mathbb{Z}_q^n$ uniformly chosen
- Recover $s \in \mathbb{Z}_q^n$

Error Distribution Ψ_α

- Used for error distribution χ
- Sample from $\mathcal{N}(0, \frac{\alpha^2}{2\pi})$
- Modulo 1 to value in $[0, 1)$
- Multiply with p and round to next integer
- Same as dividing $[0, 1)$ in $\{0, \frac{1}{p}, \frac{2}{p}, \ldots, \frac{p-1}{p}\}$

Error Distribution Ψ_α - Example



Ψ_α for $p = 127$ with $\alpha = 0.05$ (left) and $\alpha = 0.1$ (right)
Similar to discret gaussian distribution with standard deviation αp

[1] Source https://cims.nyu.edu/ regev/papers/qcrypto.pdf

Reduction LWE to CVP

- Use LWE oracle to solve CVP
- Given lattice Λ and point $x \in \mathbb{R}^n$ close to lattice point $v \in \Lambda$

Reduction LWE to CVP

- Use LWE oracle to solve CVP
- Given lattice Λ and point $x \in \mathbb{R}^n$ close to lattice point $v \in \Lambda$
- Given samples y_i on dual lattice Λ^*

Reduction LWE to CVP

- Use LWE oracle to solve CVP
- Given lattice Λ and point $x \in \mathbb{R}^n$ close to lattice point $v \in \Lambda$
- Given samples y_i on dual lattice Λ^*
- Use the samples to create LWE, v as secret
- Interprete x as v with error e

Reduction LWE to CVP

- Use LWE oracle to solve CVP
- Given lattice Λ and point $x \in \mathbb{R}^n$ close to lattice point $v \in \Lambda$
- Given samples y_i on dual lattice Λ^*
- Use the samples to create LWE, v as secret
- Interprete x as v with error e
- $a_i = (B^*)^{-1} y_i$, representation in base of Λ^*
- $(a_i \mod q, b_i = \langle y_i, x \rangle \mod q)$

Reduction LWE to CVP

- $b_i = \langle y_i, x \rangle = \langle y_i, v \rangle + \langle y_i, e \rangle \mod q$

Reduction LWE to CVP

- $b_i = \langle y_i, x \rangle = \langle y_i, v \rangle + \langle y_i, e \rangle \mod q$

- Change of basis matrix $B^T = (B^*)^{-1}$
- $\langle y_i, v \rangle = \langle (B^*)^{-1} y_i, B^{-1} v \rangle = \langle a_i, s \rangle \mod q$

Reduction LWE to CVP

- $b_i = \langle y_i, x \rangle = \langle y_i, v \rangle + \langle y_i, e \rangle \mod q$

- Change of basis matrix $B^T = (B^*)^{-1}$
- $\langle y_i, v \rangle = \langle (B^*)^{-1} y_i, B^{-1} v \rangle = \langle a_i, s \rangle \mod q$
- With $a_i \in \mathbb{Z}^n$, $s = B^{-1} v \in \mathbb{Z}^n$

Simple Encryption

Keygen

- Secret $s \in \mathbb{Z}_q^n$
- Public pairs $(a_i, b_i = \langle a_i, s \rangle + e_i)$

Simple Encryption

Keygen

- Secret $s \in \mathbb{Z}_q^n$
- Public pairs $(a_i, b_i = \langle a_i, s \rangle + e_i)$

Encrypt bit b

- Choose indices J
- $\mathrm{Enc(b)} = (\sum_{i \in J} a_i,\ \ \sum_{i \in J} b_i + b \cdot q/2) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$

Simple Encryption

Keygen

- Secret $s \in \mathbb{Z}_q^n$
- Public pairs $(a_i, b_i = \langle a_i, s \rangle + e_i)$

Encrypt bit b

- Choose indices J
- Enc(b) $= (\sum_{i \in J} a_i,\ \sum_{i \in J} b_i + b \cdot q/2) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$

Decrypt (c_1, c_2)

- calculate $e = \sum_{i \in J} b_i - \sum_{i \in J} \langle a_i, s \rangle = c_2 - \langle c_1, s \rangle$
- $b = 1$ if e closer to q/2 than 0 mod q
- Idea: sum of error terms stil small $(<< q/2)$

CRYSTALS Kyber - More practical Encryption

Ring LWE

- $R = Z_q[X]/(X^d + 1)$ instead of Z_q

CRYSTALS Kyber - More practical Encryption

Ring LWE

- $R = Z_q[X]/(X^d + 1)$ instead of Z_q
- Polynom reduced by $X^d + 1$

CRYSTALS Kyber - More practical Encryption

Ring LWE

- $R = Z_q[X]/(X^d + 1)$ instead of Z_q
- Polynom reduced by $X^d + 1$
- Coefficients in Z_q

CRYSTALS Kyber - More practical Encryption

Ring LWE

- $R = Z_q[X]/(X^d + 1)$ instead of Z_q
- Polynom reduced by $X^d + 1$
- Coefficients in Z_q
- $X^2 + 1 \cdot X + 0 \in Z_2[X]/(X^3 + 1)$

CRYSTALS Kyber - More practical Encryption

Ring LWE

- $R = Z_q[X]/(X^d + 1)$ instead of Z_q
- Polynom reduced by $X^d + 1$
- Coefficients in Z_q
- $X^2 + 1 \cdot X + 0 \in Z_2[X]/(X^3 + 1)$
- Can store multiple bits in one polynom

CRYSTALS Kyber - More practical Encryption

Keygen

- Secret $s \in R^n$
- Public Key $A \in R^{n \times n}, t = As + e \in R^n$

CRYSTALS Kyber - More practical Encryption

Keygen

- Secret $s \in R^n$
- Public Key $A \in R^{n \times n}, t = As + e \in R^n$

Encryption of m

- choose $e_1, r \in R^n, e_2 \in R$
- $u^T = r^T A + e_1$
- $v = r^T t + e_2 + q/2 \cdot m$
- $c = (u^T, v)$

CRYSTALS Kyber - More practical Encryption

Keygen

- Secret $s \in R^n$
- Public Key $A \in R^{n \times n}, t = As + e \in R^n$

Encryption of m

- choose $e_1, r \in R^n, e_2 \in R$
- $u^T = r^T A + e_1$
- $v = r^T t + e_2 + q/2 \cdot m$
- $c = (u^T, v)$

Decryption of $c = (u^T, v)$

- $w = v - u^T s$
- $m = \frac{\text{round}(w)}{q/2}$

CRYSTALS Kyber - More practical Encryption

$$v - u^T s = r^T t + e_2 + q/2 \cdot m - (r^T A + e_1)s$$

CRYSTALS Kyber - More practical Encryption

$$
\begin{aligned}
v - u^T s &= r^T t + e_2 + q/2 \cdot m - (r^T A + e_1)s \\
&= r^T A s + r^T e + e_2 + q/2 \cdot m - r^T A s + e_1 s
\end{aligned}
$$

CRYSTALS Kyber - More practical Encryption

$$
\begin{aligned}
v - u^T s &= r^T t + e_2 + q/2 \cdot m - (r^T A + e_1)s \\
&= {\color{red} r^T A s} + r^T e + e_2 + q/2 \cdot m - {\color{red} r^T A s} - e_1 s
\end{aligned}
$$

CRYSTALS Kyber - More practical Encryption

$$
\begin{aligned}
v - u^T s &= r^T t + e_2 + q/2 \cdot m - (r^T A + e_1)s \\
&= r^T As + r^T e + e_2 + q/2 \cdot m - r^T As + e_1 s \\
&= q/2 \cdot m + r^T e + e_2 + e_1 s \\
&= q/2 \cdot m + \text{small}
\end{aligned}
$$

CRYSTALS Kyber - More practical Encryption

$$
\begin{aligned}
v - u^T s &= r^T t + e_2 + q/2 \cdot m - (r^T A + e_1)s \\
&= r^T As + r^T e + e_2 + q/2 \cdot m - r^T As + e_1 s \\
&= q/2 \cdot m + r^T e + e_2 + e_1 s \\
&= q/2 \cdot m + \text{small}
\end{aligned}
$$

$\Rightarrow$ Rounded to $m \cdot q/2$

Outlook

- Signatures (Dilithium)
- Fully Homomorphic Encryption