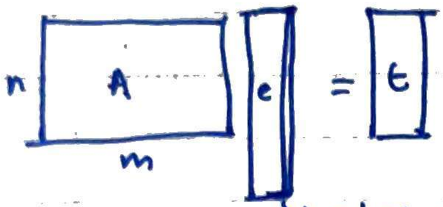
~~example~~ → LWE is being used in most cryptosystems

Why are they difficult?

→ e is small → Say it is binary



↳ e has m coefficients (typical ~~type~~ sizes are 256 ...)

we need to find 2^m trials to find the val of t → $2^{O(n)}$

~~2~~ 3 different systems

$$\begin{bmatrix} a_{00} & a_{01} & \cdots & a_{0n} \\ a_{10} & \ddots & & \\ \vdots & & \ddots & \\ a_{n0} & \cdots & & a_{nn} \end{bmatrix} \cdot \begin{bmatrix} s_0 \\ s_1 \\ \vdots \\ s_n \end{bmatrix} + \begin{bmatrix} e_0 \\ e_1 \\ \vdots \\ e_n \end{bmatrix} = \begin{bmatrix} t_0 \\ t_1 \\ \vdots \\ t_n \end{bmatrix}$$

Learning with errors → bad for large n

Storage ~~s~~ = $O(n^2)$

computation = $O(n^2)$

$$\begin{bmatrix} a_{0,0} & -a_{n,0} & \cdots & a_{1,0} \\ a_{1,0} & a_{0,0} & & \vdots \\ \vdots & a_{1,0} & & \vdots \\ a_{n-1,0} & \vdots & & \vdots \\ a_{n,0} & a_{n-1,0} & & a_{0,0} \end{bmatrix} \cdot \begin{bmatrix} s_0 \\ s_1 \\ \vdots \\ s_m \end{bmatrix} + \begin{bmatrix} e_0 \\ e_1 \\ \vdots \\ e_n \end{bmatrix} = \begin{bmatrix} t_0 \\ t_1 \\ \vdots \\ t_n \end{bmatrix}$$

Ring learning with errors → Very similar to fft

- Storage = $O(n)$
- Computation: $O(n \log n)$

$$\begin{bmatrix} A_{00}(x) & \cdots & A_{0,k}(x) \\ \vdots & \ddots & \vdots \\ A_{k,0}(x) & \cdots & A_{k,k}(x) \end{bmatrix} \begin{bmatrix} s_0 \\ \vdots \\ s_k \end{bmatrix} + \begin{bmatrix} e_0 \\ \vdots \\ e_k \end{bmatrix} = \begin{bmatrix} t_0 \\ t_1 \\ \vdots \\ t_k \end{bmatrix}$$

Module - learning with errors

- Storage ~~s~~ $O(k^2 n)$
- Computation $O(k^2 n \log n)$

Kyber uses module learning with errors becouse with different k's you can get different levels of security

→ kyber can provide different security levels