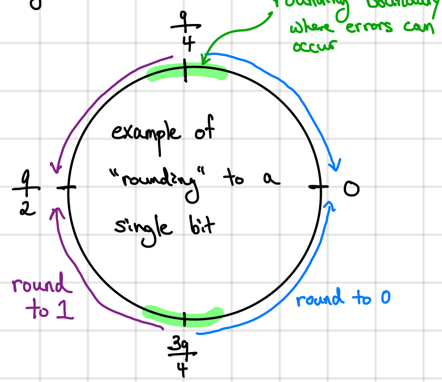
So far ... we have developed public-key encryption + signatures from lattices — what remains: analog of Diffie-Hellman?



Alice | **Bob**

Alice: $A \xleftarrow{R} \mathbb{Z}_q^{n \times n}$

Alice: $S, E \leftarrow \chi^{n \times n}$

Bob: $S', E' \leftarrow \chi^{n \times n}$

Bob: $E'' \leftarrow \chi^{n \times n}$

Alice → Bob: $A, \overbrace{AS + E}^{B}$

Bob → Alice: $S'A + E'$

Alice: compute $(S'A + E')S$ and "round"

Bob: compute $S'B + E''$ and "round"
↳ drop least significant bits (similar to decryption in Regev encryption)

Under the LWE assumption:

$$(A, AS+E) \approx \mathcal{U} \text{ where } \mathcal{U} \xleftarrow{R} \mathbb{Z}_q^{n \times m}$$

[note: requires that LWE holds even if S is sampled from error distribution]

⟶ shared key then derived by $S'B + E''$ ⟶ by LWE, $(B, S'B + E'') \approx (B, \mathcal{U}')$

⟶ shared key is derived from random matrix (similar to Diffie-Hellman, the key material is hashed to derive a symmetric key)

Practical considerations:

- Key reconciliation: presence of noise means Alice and Bob may end up with inconsistent keys
  Bob sends a "hint" with his message to reconcile any errors and ensure exact key agreement
- Message size: large matrix A is uniform — can be derived from a short seed (using PRG)
  ↳ justifiable using the random oracle model

Above construction relies on security of LWE where the secret key is sampled from error distribution
↳ This is LWE in "Hermite normal form" and is just as hard as standard LWE

Consider LWE challenge (A, b) where $A \xleftarrow{R} \mathbb{Z}_q^{n \times m}$, $b \in \mathbb{Z}_q^m$

Write $A = [A_1 \mid A_2]$ where $A_1 \in \mathbb{Z}_q^{n \times n}$, $A_2 \in \mathbb{Z}_q^{n \times (m-n)}$; similarly write $b^T = [b_1^T \mid b_2^T]$ where $b_1 \in \mathbb{Z}_q^n$, $b_2 \in \mathbb{Z}_q^{m-n}$

If A_1 is invertible, then define

$$\bar{A} = -A_1^{-1} A_2 \qquad \bar{b}^T = b_1^T \bar{A} + b_2^T$$

If $b^T = s^T A + e^T$, then

$$s^T [A_1 \mid A_2] + [e_1^T \mid e_2^T] = [b_1^T \mid b_2^T]$$

$$\begin{aligned}
\Rightarrow \bar{b}^T = b_1^T \bar{A} + b_2^T &= (s^T A_1 + e_1^T)\bar{A} + s^T A_2 + e_2^T \\
&= s^T A_1 \bar{A} + s^T A_2 + e_1^T \bar{A} + e_2^T \\
&= s^T A_1 (-A_1^{-1} A_2) + s^T A_2 + e_1^T \bar{A} + e_2^T \\
&= e_1^T \bar{A} + e_2^T
\end{aligned}$$

$\Rightarrow (\bar{A}, \bar{b})$ is an instance of normal-form LWE
($A_1^{-1} A_2$ is uniform since A_2 is uniform and A_1^{-1} is invertible)

If $b^T \xleftarrow{R} \mathbb{Z}_q^m$, then $b_1^T \bar{A} + b_2^T$ is uniform (by b_2)

Thus, normal-form LWE at least as hard as standard LWE: $(\bar{A}, \bar{b})$ is a normal-form LWE challenge

Further optimizations for LWE-based cryptography:

$$(A, s^T A + e^T)$$

typically a large matrix (e.g., $n \times m$ where $m \sim n \log q$)

computing $s^T A + e^T$ requires $O(mn)$ time (m inner products)

<u>Idea</u>: Replace the ring $\mathbb{Z}$ with a different ring (algebraic structure that supports addition + multiplication)

Specifically, we consider polynomial rings:

$\mathbb{Z}[x]$: polynomials with integer coefficients in the variable x

$\mathbb{Z}[x]/f(x)$: polynomials with integer coefficients modulo the polynomial $f(x)$

<u>Ring-LWE</u>: <u>Common choice of ring</u>: Power-of-two cyclotomic ring $\mathbb{Z}[x]/(x^{2^d}+1)$

General cyclotomic ring $\mathbb{Z}[x]/\Phi_m(x)$ where Φ_m denotes m^{th} cyclotomic polynomial

equivalent formulation of ring LWE over ring $R = \mathbb{Z}[x]/f(x)$:

More generally, can consider secrets over R_q^d (module LWE)

$$\left\{\begin{array}{ll} s \xleftarrow{R} R_q & (R_q = \mathbb{Z}_q[x]/f(x) \text{ — coefficients reduced modulo } q) \\ a \xleftarrow{R} R_q & \\ e \leftarrow \chi & (\chi : \text{error distribution over } R) \\ u \leftarrow R_q & \end{array}\right.$$

LWE: ($d = n$, $R = \mathbb{Z}$)

RLWE: ($d = 1$, $R = \mathbb{Z}/f(x)$)

$$\Longrightarrow (a, \underbrace{sa+e}) \approx (a, u)$$

computing can be done in time $O(m \lg m)$ using FFT (for properly chosen q)

if f has degree m, then $sa+e$ is still a vector of m values

a is a <u>vector</u> of m values, <u>not</u> a matrix

advantages over LWE: more compact public parameters, faster computation

disadvantage: more structured assumption $\Rightarrow$ easier to break?

Some rings are insecure but for the rings proposed for standardization, does not seem significantly less secure than LWE

Google piloted an experiment using RLWE-based key exchange in Chrome in 2016 (using a "best of both worlds" approach)

We can also view ring LWE as LWE with <u>structured</u> matrices

<u>Example</u>: $\mathbb{Z}[x]/(x^2+1) = \{ax + b : a, b \in \mathbb{Z}\} \cong \mathbb{Z}^2$

Every polynomial $f \in \mathbb{Z}[x]/(x^2+1)$ can be associated with a matrix M_f that "represents" polynomial multiplication

Suppose $f(x) = ax + b$, $g(x) = cx + d$

$$\Rightarrow f(x) \cdot g(x) = (ax+b)(cx+d)$$

$$= (ac)x^2 + (ad+bc)x + (bd)$$

$$= (ad+bc)x + (bd-ac) \quad \text{since } x^2 = -1 \pmod{x^2+1}$$

$$\underset{M_f}{\begin{bmatrix} b & a \\ -a & b \end{bmatrix}} \underset{g}{\begin{bmatrix} c \\ d \end{bmatrix}} = \underset{fg}{\begin{bmatrix} ad+bc \\ bd-ac \end{bmatrix}}$$

Computing this matrix-vector product is <u>equivalent</u> to computing a product of polynomials

- Advantage 1: polynomial multiplication can use FFT ($n \log n$ for degree n polynomials)
- Advantage 2: Matrix M_f can be described by n values rather than n^2 values

LWE is a versatile assumption: yields key exchange, public-key cryptography, signatures

also enables advanced primitives like

- fully homomorphic encryption: arbitrary computation on ciphertexts
- identity-based encryption: public-key encryption scheme where public keys can be arbitrary strings
- functional encryption: fine-grained control of data access
- and many more!

→ also plausibly post-quantum resilient!