Motivation:
Different applications, different needs

Security & trust needs

Performance needs

Application 1
Application 2
Application 3
Application 4
Application 5
Different applications, different needs

Note: the applications in this figure are only examples to illustrate that different applications have different security & performance needs.
Main features

• One **unified design** to fit all use cases,
  – Ring and non-ring support.
  – Round2.KEM and Round.PKE with same building blocks.

• Fine-grained scaling of parameters to any required security level.

• Great bandwidth.

• Great computation speed.

• LWR, well-studied lattice problem.
Main features

LWR-based

• Builds on LWR problem:

Search LWR: public integers \( p, q \), public matrix \( A \in \mathbb{Z}_q^{d \times d} \), secret \( s \in \mathbb{Z}_q^d \), public vector \( b = \left\lfloor \frac{p}{q} As \right\rfloor \pmod{p} \). Find \( s \).

• Compared with LWE:
  – Improved bandwidth \((p < q)\).
  – Improved computation.
  – No noise sampling needed.
Main features

General LWR (GLWR) unifies LWR and RLWR

- Allows for unified design and implementation:
  - Ring $R_{n,q}$, for $n = 1$, $R_{n,q} \equiv \mathbb{Z}_q$.

- Fits applications with different trust needs (presence/absence of ring structure).
Main features

Common building blocks for INDCPA and INDCCA security

Round2.KEM and Round2.PKE support applications with different performance/security needs:
- Using common building blocks.
- Secure email can rely on Round2.PKE (INDCCA).
- IPSec VPN can use faster (~2x) Round2.KEM (INDCPA).
Main features
Common building blocks for INDCPA and INDCCA security

- Received official comment on INDCPA proof.
- Easily solvable as indicated by SABER team in their official comment.
- No change to parameters.
Main features
Prime cyclotomic ring

\[ R_n = \frac{x^{n+1} - 1}{x - 1} \]

• Security
  – Provable: Known reductions from RLWE and (Ideal) lattice problems.
  – Practical: Parameters chosen to avoid subrings (and thus, potential attacks).
• Scalable (bandwidth and security level) due to many choices for \( n \).

<table>
<thead>
<tr>
<th>( n )</th>
<th>418</th>
<th>676</th>
</tr>
</thead>
<tbody>
<tr>
<td>Public-key (Bytes)</td>
<td>435</td>
<td>709</td>
</tr>
<tr>
<td>Ciphertext (Bytes)</td>
<td>482</td>
<td>868</td>
</tr>
<tr>
<td>Failure probability (log2)</td>
<td>-81</td>
<td>-65</td>
</tr>
<tr>
<td>Best (quantum) attack (bits)</td>
<td>75</td>
<td>139</td>
</tr>
<tr>
<td>Best (classical) attack (bits)</td>
<td>79</td>
<td>144</td>
</tr>
</tbody>
</table>
Main features

GLWR and ring choice lead to great bandwidth performance

• For similar security level (bits), Round2 offers better performance.

• Round2 is scalable: parameters easily configured to offer any required security target.
Main features

Power of two moduli $q, p, t$

- $p, t$: Optimized bandwidth (transmit only $\log_2 p, \log_2 t$ bits).

- $t$: Allows to finely tune failure probability (depends on $t$).

- $q$: Optimized CPU performance in both ring and non-ring settings.
Main features

Generation of public parameter: $A \leftarrow f_n^\tau$

- **Static $A$**
  - CPU (1x)
  - No unified
  - Pre-computation attack

- **Dynamic $A$**
  - CPU (11.7x)
  - No unified
  - Pre-computation attack

- **Non-ring**
  - $a_{master}: d \ll \text{lenght} \leq d^2$
  - Permute

- **Ring**
  - $a_{master}: \text{lenght} = d$
  - Permute

- **CPU (1.4x)**
  - Unified
  - Pre-computation attack

- **CPU (< 1)**
  - Unified
  - Pre-computation attack
Main features

Sparse trinary secrets with fixed hamming weight

- Definition depends on $d$, and not on $n$, to enable unified implementation
  - Matrix-based multiplication involves always $d$ dimensional vectors, independently of ring or non-ring settings.
- Great performance.
- Low failure probability.
Main features

Parameter sets

- **uRound2**: unified implementation for ring and non-ring
  - Main submission.
  - One implementation, any set of parameters.
    - $q$ power of two.
    - Ring or non-ring.
    - Any security level.
    - Always, great performance.

- **nRound2**:
  - Specialized parameter set to support NTT.
  - Chooses prime $q$. 

Conclusions & Remarks

• Different applications have different security/performance needs.

• Round2 is an efficient & scalable scheme that fits needs of different applications.

• Lattice-based proposals should be compared based on same methodology to give security estimates.

• Explicit failure probability target required for comparing different proposals.

• Minimal KEM proposal by Mike Hamburg makes lots of sense.
Questions?
Thank you