The Lattigo lattice-based cryptographic library
Jean-Philippe Bossuat, Christian Mouchet, Juan Troncoso-Pastoriza, Jean-Pierre Hubaux

Motivation

• The tremendous performance improvement of homomorphic encryption makes it now usable
• Its greatest potential is not in privacy-preserving cloud outsourcing, but for Secure Multiparty Computation (SMC)
• This requires a lattice-based crypto library that:
  ‣ is easy to deploy in modern software stacks
  ‣ can support quick academic prototyping

Lattigo at a glance

✓ Pure Go (1.12)
✓ Optimized math layer
✓ Encrypted integer-arithmetic (BFV[1])
✓ Encrypted complex/float arithmetic (CKKS[2])
✓ Automatic Chebyshev approximation
✓ Distributed/Threshold key & SMC support

SMC Framework

• Secret shared keys / collective encryption
• Collective key-switching/re-encryption

Upcoming

• Bootstrapping for CKKS
• Post-quantum key exchange & signatures
• Zero-knowledge proofs
• Network layer implementation of SMC

The Lattigo library unleashes the potential of lattice-based cryptography in secure multiparty computation for modern software stacks.

Library overview

lattigo/dbfv
• Collective key generation
• Relinearization key generation
• Collective key-switching

lattigo/bfv
• Encrypted integer arithmetic
• Full-RNS operations

lattigo/ckks
• Encrypted complex arithmetic
•Full-RNS operations
• Automatic Chebyshev approx.

lattigo/ring
• NTT-polynomial and RNS-coefficients representations & arithmetic
• Efficient 64b modular arithmetic
  Short Barrett reduction for 64b integers
  Barrett reduction for arbitrary 64b×64b product
  Montgomery reduction
• Efficient sampling in Gaussian distribution
  RNS and Montgomery forms

Benchmarks

Processor: Intel Xeon Gold 6132 @ 2.6 GHz
Memory: 256 GB
Go version: Go1.12.7 linux/amd
Baseline: SEAL 3.2.0[3]

<table>
<thead>
<tr>
<th>Operation timing [ms]</th>
<th>(baseline ratio)</th>
</tr>
</thead>
<tbody>
<tr>
<td>lattigo/bfv</td>
<td></td>
</tr>
<tr>
<td>Encrypt</td>
<td>13.3 (0.6)</td>
</tr>
<tr>
<td>Decrypt</td>
<td>2.3 (0.4)</td>
</tr>
<tr>
<td>Add</td>
<td>0.9 (0.9)</td>
</tr>
<tr>
<td>Multiply</td>
<td>47.4 (2.6)</td>
</tr>
<tr>
<td>Scale</td>
<td>7.0 (5.8)</td>
</tr>
<tr>
<td>RotateScale</td>
<td>6.9 (5.5)</td>
</tr>
<tr>
<td>RotateValue</td>
<td>6.9 (5.3)</td>
</tr>
<tr>
<td>lattigo/ckks</td>
<td></td>
</tr>
<tr>
<td>Encode</td>
<td>3.0 (0.6)</td>
</tr>
<tr>
<td>Decode</td>
<td>8.3 (3.1)</td>
</tr>
<tr>
<td>Encrypt</td>
<td>13.2 (1.8)</td>
</tr>
<tr>
<td>Decrypt</td>
<td>0.4 (0.5)</td>
</tr>
<tr>
<td>Add</td>
<td>0.3 (0.3)</td>
</tr>
<tr>
<td>Multiply</td>
<td>0.8 (0.4)</td>
</tr>
<tr>
<td>Scale</td>
<td>5.0 (0.5)</td>
</tr>
<tr>
<td>Rescale</td>
<td>1.6 (0.6)</td>
</tr>
<tr>
<td>Dot/Transpose</td>
<td>5.7 (0.9)</td>
</tr>
<tr>
<td>RotateValue</td>
<td>5.9 (0.9)</td>
</tr>
</tbody>
</table>


https://github.com/ica1/lattigo