On the Design of Hardware Building Blocks for Modern Lattice-Based Encryption Schemes

Norman Göttert, Thomas Feller, Michael Schneider, Johannes Buchmann, and Sorin A. Huss
Motivation

- Alternative security background required
- Current systems may be broken by quantum computers
- No sub-exponential attacks on commodity computers

Learning With Errors (LWE)
[LP11] Lindner, Peikert CT-RSA 2011
Contributions

- Implementation of three variants
- LWE-Matrix and LWE-Polynomial in software
- Polynomial variant in Hardware
- Fundamental Building Blocks for Lattice-Based Cryptography
- Performance evaluation
LWE Basics

- Strong security proofs for lattice-based crypto exist
- Encryption introduces Gaussian distributed errors
- Decoding works if errors are below threshold
- Error tolerant message encoding
Gaussian Error Sampling

- Software: Rejection Sampling
- Hardware: Look-Up-Table

Rejection Sampling.

Address-based Saves >90% of bits
Gaussian Error Sampling

- Rejection Sampling performance
- Devroye Sampler

Histogram of samples.
Preliminaries

Public Key \((a, p)\) | Private Key \(r_2\)

\[ a \in \mathbb{Z}_q[X]/\langle f(x) \rangle \] chosen uniformly at random

\[ r_1, r_2 \leftarrow \chi \] a Gaussian distribution

\[ p = r_1 - a \cdot r_2 \]

Decoding works if errors are below threshold

\[ e_1, e_2, e_3 \leftarrow \chi \hspace{1cm} \delta = |e_1 \cdot r_1 + e_2 \cdot r_2 + e_3| \leq t \]
LWE-based Cryptosystem

\[
\begin{align*}
(a, p) & \quad \text{public key} \\
\text{m} & \quad \text{message} \\
(r_2) & \quad \text{private key} \\
(c_1, c_2) & \quad \text{ciphertext}
\end{align*}
\]
Fast Fourier Transform-Based Polynomial Multiplication I

Algorithm:

1. $A = \text{FFT}(a, \omega_m)$
2. $B = \text{FFT}(b, \omega_m)$
3. for $i=0$ to $2n-1$ do
5. end
6. $c = \text{FFT}^{-1}(C, \omega_m)$
7. return $c$
Fast Fourier Transform-Based Polynomial Multiplication II

- Polynomial multiplication using coefficients $O(n^2)$
- Point-value representation
  - Serial $O(n \log(n))$
  - Parallel $O(\log(n))$
- FFT-based approach can be utilized for other lattice-based encryption schemes
# Message Expansion Factors

<table>
<thead>
<tr>
<th>n</th>
<th>LWE-Matrix</th>
<th>LWE-Polynomial</th>
<th>LWE-Hardware</th>
<th>[LP11]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>cipher</td>
<td>cipher/plain</td>
<td>cipher</td>
<td>cipher/plain</td>
</tr>
<tr>
<td>128</td>
<td>512</td>
<td>32</td>
<td>512</td>
<td>32</td>
</tr>
<tr>
<td>192</td>
<td>640</td>
<td>40</td>
<td>768</td>
<td>48</td>
</tr>
<tr>
<td>256</td>
<td>768</td>
<td>48</td>
<td>1024</td>
<td>64</td>
</tr>
<tr>
<td>320</td>
<td>896</td>
<td>56</td>
<td>1280</td>
<td>80</td>
</tr>
<tr>
<td>384</td>
<td>1024</td>
<td>64</td>
<td>1536</td>
<td>96</td>
</tr>
<tr>
<td>448</td>
<td>1152</td>
<td>72</td>
<td>1792</td>
<td>112</td>
</tr>
<tr>
<td>512</td>
<td>1280</td>
<td>80</td>
<td>2048</td>
<td>128</td>
</tr>
</tbody>
</table>
Evaluation Results I

Throughput

(a) Software

(b) Hardware
Evaluation Results II

LWE-Matrix vs. LWE-Polynomial
Execution times in ms
Evaluation Results III

Decryption Error Rate

- LWE-Matrix, $\delta=10^{-2}$
- LWE-Matrix, $\delta=10^{-3}$
- LWE-Matrix, $\delta=10^{-4}$
- LWE-Poly, $\delta=10^{-2}$
- LWE-Poly, $\delta=10^{-3}$
- LWE-Poly, $\delta=10^{-4}$

Error Rate vs. $n$ (number of elements)
## Evaluation Results IV

<table>
<thead>
<tr>
<th></th>
<th>Virtex-6 LX240T</th>
<th>Virtex-7 2000T</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>n=128 %</td>
<td>n=512 %</td>
</tr>
<tr>
<td># Register</td>
<td>37918</td>
<td>174757</td>
</tr>
<tr>
<td># LUTs</td>
<td>64804</td>
<td>348204</td>
</tr>
</tbody>
</table>

### KeyGen

<table>
<thead>
<tr>
<th></th>
<th>Virtex-6 LX240T</th>
<th>Virtex-7 2000T</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>n=128 %</td>
<td>n=512 %</td>
</tr>
<tr>
<td># Register</td>
<td>64680</td>
<td>296207</td>
</tr>
<tr>
<td># LUTs</td>
<td>131254</td>
<td>634893</td>
</tr>
</tbody>
</table>

### Encrypt

<table>
<thead>
<tr>
<th></th>
<th>Virtex-6 LX240T</th>
<th>Virtex-7 2000T</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>n=128 %</td>
<td>n=512 %</td>
</tr>
<tr>
<td># Register</td>
<td>31884</td>
<td>134036</td>
</tr>
<tr>
<td># LUTs</td>
<td>56311</td>
<td>260772</td>
</tr>
</tbody>
</table>

### Decrypt

<table>
<thead>
<tr>
<th>Hardware Resource Utilization</th>
<th>KeyGen</th>
<th>Encrypt</th>
<th>Decrypt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case</td>
<td>Thomas Feller</td>
<td>Cyberphysical Systems and Networks Group</td>
<td>15</td>
</tr>
</tbody>
</table>
Conclusion

- Sampling of Gaussian distributed numbers
- HW/SW Implementation of LWE-Scheme
- Extensive evaluation

FFT-based polynomial multiplication as building block for Lattice-based Schemes
TOP SECRET
Message Encoding

- Message $m \in \Sigma$, alphabet $\Sigma = \{0,1\}$
- $m \rightarrow \overline{m} \in \mathbb{Z}_q$
- Encoder function:
  $$\text{encode}(m_i) = m_i \cdot \left\lfloor \frac{a}{2} \right\rfloor = \overline{m_i}, \text{for } 0 \leq i < 1$$
- Decoder function:
  $$\text{decode}(\overline{m_i}) = \begin{cases} 0, \text{if } \overline{m_i} \in [-\left\lfloor \frac{a}{2} \right\rfloor, \left\lceil \frac{a}{2} \right\rceil) \\ 1, \text{otherwise} \end{cases} = m_i, \text{for } 0 \leq i < 1$$
LWE Key Generation

\[(a, p) – \text{public key} \quad r_2 – \text{private key}\]