Compact-LWE: Lattice-based PKE without Concretely Relying on the Hardness of Lattice Problems

Dongxi Liu  Nan Li  Jongkil Kim  Surya Nepal
9 April 2018
Contents in Submitted Specification

- Private Key Recovery
- Ciphertext-Only Attack
- Security and Attack Analysis
- Performance Evaluation
- Hardness Discussion
- Compact-LWE Public Key Encryption (PKE)
- Compact-LWE Problem
Security Problem

- Ciphertext-only attacks to Compact-LWE PKE can be true
  - Found by Pan et al., and Boole et al.
  - A countermeasure provided below

- Hardness of Compact-LWE problem not affected
Outline

- Compact-LWE problem and its hardness
- Compact-LWE PKE
  - key generation, encryption, decryption
  - an instance (parameters, sizes of keys and ciphertexts)
- Explanation of Ciphertext-only Attack
- Countermeasure: Revision to Compact-LWE PKE
- Advantages
Compact-LWE problem

- Secret values: \( s, s' \in \mathbb{Z}_q^n, k, k' \in \mathbb{Z}_q, ck, ck' \in \mathbb{Z}_p, \) and \( p < q \)
  - All values randomly sampled from uniform distributions
- Compact-LWE samples
  - \((a_i, \langle a_i, s \rangle + k \ast (r_i + p \ast e_i)) \mod q, \langle a_i, s' \rangle + k' \ast (r'_i + p \ast e'_i) \mod q\)
    - \( r_i, r'_i \in \mathbb{Z}_p, \) satisfying \( ck \ast r_i + ck' \ast r'_i = 0 \mod p \)
    - \( e_i, e'_i \) are small error values
    - \( a_i \in \mathbb{Z}_b^n, \) with \( b \leq q \)
- Compact-LWE problem
  - finding secret values from Compact-LWE samples
Hardness of Compact-LWE Problem

- A LWE sample is a Compact-LWE sample with $k = 1$, $k' = 1$, $p = 1$, $ck = 0$, $ck' = 0$, and $b = q$.
- Smaller $b$ makes Compact-LWE resistant to lattice-based attacks to recover original $s$ or $s'$.
Compact-LWE PKE

- Public parameters: nine positive integers
  - $q, t, n, m, w, w', b, b', l$
- Generation of private keys
  - Private parameters: $sk_{max}, p_{size}, e_{min},$ and $e_{max}$
  - Private key: $(s, k, sk, ck, s', k', sk', ck', p)$
    - $p \in \{(w + w') * b', ..., (w + w') * b' + p_{size}\}$
    - $p$ coprime with $q$ and
    - $sk_{max} * b' + p + e_{max} * p < q / (w + w')$
    - $sk, sk' \in \mathbb{Z}_{sk_{max}}$, satisfying $sk * ck + sk' * ck'$ coprime with $p$
Compact-LWE PKE

- Generation of public keys
  - $m$ public key samples $(a_i, u_i, pk_i, pk'_i)$
    - $pk_i = \langle a_i, s \rangle + k_q^{-1} \times (sk \times u_i + r_i + e_i \times p) \mod q$
    - $pk'_i = \langle a_i, s' \rangle + k'_q^{-1} \times (sk' \times u_i + r'_i + e'_i \times p) \mod q$
    - $u_i \in \mathbb{Z}_{b_i}$, $e_i \in [e_{\text{min}}, e_{\text{max}}]$, and $e'_i \in [e_{\text{min}}, e_{\text{max}}]$

- Encryption
  - basic encryption: only encrypting messages in $\mathbb{Z}_t$
  - general encryption: relying on basic encryption to encrypt long messages
Compact-LWE PKE: Basic Encryption

- Generate the $m$-dimensional random vector $l$, such that
  - $w \leq \sum_{i=1}^{m} l[i] \leq w + w'$ for all $l[i] > 0$
  - $-w' \leq \sum_{i=1}^{m} l[i] \leq 0$ for all $l[i] < 0$
  - $\sum_{i=1}^{m} l[i] \cdot u_i > 0$

- Generate the ciphertext $c$
  - $(\sum_{i=1}^{m} l[i] \cdot a_i, f(v, \sum_{i=1}^{m} l[i] \cdot u_i), \sum_{i=1}^{m} l[i] \cdot pk_i \mod q, \sum_{i=1}^{m} l[i] \cdot pk'_i \mod q)$
    - where
      \[
      f(v, \sum_{i=1}^{m} l[i] \cdot u_i) = (v \oplus \text{rol}(u, \log_2(t)/2)) \cdot u' \mod t,
      \]
      \[
      u = (\sum_{i=1}^{m} l[i] \cdot u_i) \mod t, \text{ and}
      \]
      \[
      u' \geq (\sum_{i=1}^{m} l[i] \cdot u_i)/t \text{ is the smallest integer coprime with } t.
      \]
Compact-LWE PKE: Basic Decryption

- Let $c = (a, d, pk, pk')$ be the ciphertext.
- With the private key, $v$ is recovered by using the steps below:
  - Calculate $d_1 = (pk - \langle a, s \rangle) \ast k \mod q$, and $d'_1 = (pk' - \langle a, s' \rangle) \ast k' \mod q$.
  - Let $d_2 = ck \ast d_1 + ck' \ast d'_1 \mod p$.
  - Calculate $d_3 = sckInv \ast d_2 \mod p$, where $sckInv$ is determined by $sckInv \ast (sk \ast ck + sk' \ast ck') = 1 \mod p$.
  - Obtain $v = f^{-1}(d, d_3)$, where

$$f^{-1}(d, d_3) = (u_p'^{-1} \ast d \mod t) \oplus \text{ro}(u, \log_2(t)/2),$$

$$u = d_3 \mod t,$$

$u' \geq d_3/t$ is the smallest integer coprime with $t$, and

$$u_p'^{-1} \ast u' = 1 \mod t.$$
An Instance: parameters

- 192-bit search space for private keys

<table>
<thead>
<tr>
<th>$q$</th>
<th>$t$</th>
<th>$n$</th>
<th>$m$</th>
<th>$w$</th>
<th>$w'$</th>
<th>$b$</th>
<th>$b'$</th>
<th>$l$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$2^{64}$</td>
<td>$2^{32}$</td>
<td>8</td>
<td>128</td>
<td>224</td>
<td>32</td>
<td>16</td>
<td>68719476736</td>
<td>8</td>
</tr>
</tbody>
</table>

Table: Public Parameters

<table>
<thead>
<tr>
<th>$sk_{max}$</th>
<th>$p_{size}$</th>
<th>$e_{min}$</th>
<th>$e_{max}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>229119</td>
<td>16777216</td>
<td>457</td>
<td>3200</td>
</tr>
</tbody>
</table>

Table: Private Parameters
An Instance: sizes and performance

- 232 bytes for a private keys and 2064 bytes for a public key

<table>
<thead>
<tr>
<th>Message (B)</th>
<th>32</th>
<th>64</th>
<th>128</th>
<th>256</th>
<th>512</th>
<th>1024</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ciphertext (B)</td>
<td>360</td>
<td>648</td>
<td>1224</td>
<td>2376</td>
<td>4680</td>
<td>9288</td>
</tr>
</tbody>
</table>

Table: Ciphertext Size

<table>
<thead>
<tr>
<th>Message (B)</th>
<th>32</th>
<th>64</th>
<th>128</th>
<th>256</th>
<th>512</th>
<th>1024</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enc (sec)</td>
<td>1.29</td>
<td>2.15</td>
<td>4.36</td>
<td>7.56</td>
<td>14.81</td>
<td>28.7</td>
</tr>
<tr>
<td>Dec (sec)</td>
<td>0.18</td>
<td>0.27</td>
<td>0.43</td>
<td>0.88</td>
<td>1.78</td>
<td>3.50</td>
</tr>
</tbody>
</table>

Table: Performance of 100000 Encryptions and Decryptions

- Note that the evaluation will change in the revised version of Compact-LWE encryption scheme.
Explanation of Ciphertext-only Attack

- Given ciphertext \( c = (a, d, pk, pk') \), we have
  
  \[
  a = \sum_{i=1}^{m} l[i] \ast a_i \\
  pk = \sum_{i=1}^{m} l[i] \ast pk_i \mod q \\
  pk' = \sum_{i=1}^{m} l[i] \ast pk'_i \mod q \\
  d = f(v, \sum_{i=1}^{m} l[i] \ast u_i)
  \]

- From the first three equations, a short vector \( l' \) can be obtained.

- The ciphertext-only attack can succeed, due to
  
  \[
  \sum_{i=1}^{m} l[i] \ast u_i = \sum_{i=1}^{m} l'[i] \ast u_i.
  \]
Compact-LWE PKE - revised

- Changes indicated in red.
- Public parameters: ten positive integers
  - \( q, t, n, m, w, w', b, b', l, n' \)
- Generation of private keys
  - Private parameters: \( sk_{max}, p_{size}, e_{min}, \) and \( e_{max} \)
  - Private key: \((s, k, sk, ck, s', k', sk', ck', p, s'')\)
    - \( p \in \{(w + w') \cdot b', ..., (w + w') \cdot b' + p_{size}\} \)
    - \( p \) coprime with \( q \) and \( p + p + e_{max} \cdot p < q/(w + w') \)
    - \( sk, sk' \in \mathbb{Z}_p \), satisfying \( sk \cdot ck = sk' \cdot ck' \) and \( sk \cdot ck \) coprime with \( p \)
    - \( s'' = (s''[1], ..., s''[n']) \in \mathbb{Z}_{b'}^n \), with \( s''[1] \) and \( s''[2] \) co-prime with \( b' \)
Compact-LWE PKE - revised

- Generation of public keys
  - $m$ public key samples $(a_i, a'_i, pk_i, pk'_i)$
    - $pk_i = \langle a_i, s \rangle + k_q^{-1} \times ((sk \times u_i \text{ mod } p) + r_i + e_i \times p) \text{ mod } q$
    - $pk'_i = \langle a_i, s' \rangle + k'_q^{-1} \times ((sk' \times u'_i \text{ mod } p) + r'_i + e'_i \times p) \text{ mod } q$
    - $u_i, u'_i \in \mathbb{Z}_p$, and $(u_i + u'_i) \text{ mod } p \in \mathbb{Z}_{b'}$
    - $e_i \in [0, e_{\text{max}}]$, and $e'_i \in [0, e_{\text{max}}]$
    - $a'_i \in \mathbb{Z}_{b'}$, and $\langle a'_i, s'' \rangle = ((u_i + u'_i) \text{ mod } p) \text{ mod } b'$
  - Let $s2'' = (s''[1] \times s''[1], ..., s''[n'] \times s''[n']) \in \mathbb{Z}_{b'}^{n'}$.
  - For $1 \leq i < j \leq n'$, $a''_{ij} \in \mathbb{Z}_{b'}$ are included in the public key,
    - Satisfying $\langle a''_{ij}, s2'' \rangle = s''[i] \times s''[j] \text{ mod } b'$

- Encryption
  - basic encryption: only encrypting messages in $\mathbb{Z}_t$
  - general encryption: relying on basic encryption to encrypt long messages
Compact-LWE PKE: Basic Encryption - revised

• Generate the $m$-dimensional random vector $l$, such that
  - $w \leq \sum_{i=1}^{m} l[i] \leq w + w'$ and $l[i] > 0$ for $1 \leq i \leq m$
  - $-w' \leq \sum_{i=1}^{m} l[i] \leq 0$ for all $l[i] < 0$
  - $\sum_{i=1}^{m} l[i] \ast u_i > 0$

• Generate the ciphertext $c$
  - $(\sum_{i=1}^{m} l[i] \ast a_i, a', f(v, u), \sum_{i=1}^{m} l[i] \ast pk_i \mod q, \sum_{i=1}^{m} l[i] \ast pk'_i \mod q)$, where
    - $u$ randomly sampled from $\mathbb{Z}_{b'}$
    - let $(a_1, ..., a_{n'}) = \sum_{i=1}^{m} l[i] \ast a'_i$
    - $a' = (a_1^2, ..., a_{n'}^2) + \sum_{i=1}^{n'-1} \sum_{j=i+1}^{n'} 2 \ast a_i \ast a_j \ast a''_{ij} + u \ast a''_{12} \in \mathbb{Z}_{b'}$
    - no change to $f$

• More random $u$ (e.g., $u' \ast a_{13} + u'' \ast a_{14} + ...$) can be added into $a'$ if general encryption and decryption are also revised.
Compact-LWE PKE: Basic Decryption - revised

- Let \( c = (a, a', d, pk, pk') \) be the ciphertext.
- With the private key, \( v \) is recovered by using the steps below:
  - Calculate \( d_1 = (pk - \langle a, s \rangle) \cdot k \mod q \), and \( d_1' = (pk' - \langle a, s' \rangle) \cdot k' \mod q \).
  - Let \( d_2 = ck \cdot d_1 + ck' \cdot d_1' \mod p \).
  - Calculate \( d_3 = sckInv \cdot d_2 \mod p \), where \( sckInv \) is determined by \( sckInv \cdot (sk \cdot ck + sk' \cdot ck') = 1 \mod p \).
  - Let \( s2'' = (s''[1] \cdot s''[1], ..., s''[n'] \cdot s''[n']) \in \mathbb{Z}_{b'}^{n'} \).
  - Calculate \( u = (s''[1] \cdot s''[2])^{-1} \cdot (\langle a', s2'' \rangle - d_3 \cdot d_3) \mod b' \).
  - Obtain \( v = f^{-1}(d, u) \).
Evaluation of Countermeasure

- Implementation of basic encryption and decryption in Sage.
- \((n' - 1) \times \log_2 b'\) should be greater than the declared security level.
  - \(n' = 6\) and \(b' = 2^{39}\) used in our evaluation (5 \times 39 > 192)
- \(n' - 1\) elements in \(a'_i\) are independently and randomly sampled.
  - The idea of ciphertext-only attack explained before is not applicable.
  - i.e., \(\sum_{i=1}^{m} l[i] \times a'_i \neq \sum_{i=1}^{m} l'[i] \times a'_i\) when \(n' > 1\).
- \(b'\) must be a composite number.
  - \(\mathbb{Z}_{b'}\) is not a field, and thus \(s''\) cannot be recovered from \(a''_{ij}\) by solving MQ equations.
  - \(a''_{ij}\) is used to reduce ciphertext size; i.e., without \(a''_{ij}\), ciphertexts become bigger.
Advantages

- Simple to understand and implement
  - Constructed with integers and modular arithmetic
  - All random values sampled from uniform distribution
- Assuming hard problems in lattices can be efficiently solved, with small parameters (e.g., n=8) selected
  - Detect design flaws if there are easily with concrete attacks (good for preventing deeply hidden design flaws)
  - Mitigate the impact when hard problems in lattices become not hard in future
- Relatively small ciphertexts
  - A ciphertext has about 700 bytes for a 32-byte message in the revised version.