1. **Computational Analysis**

n is the security parameter. For our implementation, n =256.

wSize is the size of the word used int the implementation. It can be 8,16,32,64.

We used wSize = 64 (64 bits = 1 word)

**toeplitzbyvector**

* This function performs expand Toeplitz + matrix by vector
* Since it’s packed in words, there are only 256X4 multiplications

Number of Operations of each type

|  |  |  |
| --- | --- | --- |
| **Operation** | **Number of times** | **Asymptotic value** |
| **Shift** | (256 X 2) + (256 X 4 X 2) | **10n** |
| **AND** | (256 X 2) + (256 X 4) | **6n** |
| **XOR** | (256 X 4) | **4n** |
| **OR** | (256 X 4) | **4n** |
| **neg (-)** | 256 | **n** |

**toeplitzMatrix (Expansion**):

|  |  |  |
| --- | --- | --- |
| **Operation** | **Number of times** | **Asymptotic value** |
| **Shift** | (256) + (256 X 4 X 2) | **9n** |
| **AND** | (256) | **n** |
| **OR** | (256 X 4) | **4n** |

**matByVec**:

10 x 256 operaitons

|  |  |  |
| --- | --- | --- |
| **Operation** | **Number of times** | **Asymptotic value** |
| **Shift** | (256 X 2) | **2n** |
| **AND** | (256 X 2) + (256 X 4) | **6n** |
| **XOR** | 256 | **n** |
| **neg (-)** | 256 | **n** |

**Add**:

|  |  |  |
| --- | --- | --- |
| **Operation** | **Number of times** | **Asymptotic value** |
| **XOR** | 4 | **(n/wSize)** |

**Lsbs / subtract / -= / add / +=** (packed mod3):

|  |  |  |
| --- | --- | --- |
| **Operation** | **Number of times** | **Asymptotic value** |
| **XOR** | 4 X 3 | **3\*(n/wSize)** |
| **OR** | 4 X 4 | **4\*(n/wSize)** |

**at()**

|  |  |  |
| --- | --- | --- |
| **Operation** | **Number of times** | **Asymptotic value** |
| **XOR** | 4 | **const** |
| **OR** | 1 | **const** |
| **AND** | 3 | **const** |
| **neg (-)** | 256 | **n** |

**Toeplitz\_party1**:

|  |  |  |
| --- | --- | --- |
| **Operation** | **Number of times** | **Asymptotic value** |
| **Shift** | (256 X 2) + (256 X 4 X 2) | **10n** |
| **AND** | (256 X 2) + (256 X 4) | **6n** |
| **XOR** | (256 X 4) + 256 + 4 + 4 | **5n + (2\*(n/wSize))** |
| **OR** | (256 X 4) | **4n** |
| **neg (-)** | 256 | **n** |

**Toeplitz\_party2**:

|  |  |  |
| --- | --- | --- |
| **Operation** | **Number of times** | **Asymptotic value** |
| **Shift** | (256 X 2) + (256 X 4 X 2) | **10n** |
| **AND** | (256 X 2) + (256 X 4) | **6n** |
| **XOR** | (256 X 4) + 4 + 4 + 4 | **4n + (3\*(n/wSize))** |
| **OR** | (256 X 4) | **4n** |
| **neg (-)** | 256 | **n** |

**OT\_Party2:**

|  |  |  |
| --- | --- | --- |
| **Operation** | **Number of times** | **Asymptotic value** |
| **AND** | 2 | **const** |
| **XOR** | 3 | **const** |

**OT\_Party1:**

|  |  |  |
| --- | --- | --- |
| **Operation** | **Number of times** | **Asymptotic value** |
| **AND** | 2 + 1 + 1 + 2 | **const** |
| **XOR** | 1 + 1 | **const** |
| **negate (~)** | 4 | **n/wSize** |

**SC\_Party1: (**lsbs, negate, -=, OT\_party\_1)

|  |  |  |
| --- | --- | --- |
| **Operation** | **Number of times** | **Asymptotic value** |
| **AND** | 2 + 1 + 1 + 2 | **const** |
| **XOR** | 2 X (4 X 3) + 2 | **6 \* (n/wSize)** |
| **negate (~)** | 2 X 4 | **n/wSize** |

**SC\_Party2: (**OT\_Party\_2\_1, OT\_Party\_2\_2)

|  |  |  |
| --- | --- | --- |
| **Operation** | **Number of times** | **Asymptotic value** |
| **AND** | 8 | **const** |
| **XOR** | 5 | **const** |
| **negate (~)** | 4 | **n/wSize** |

**Third phase (2 X matByVec) – PakcedMod3**

**TO FIX/REMOVE:**

* at() -> 256
* (-) -> 2 X 256
* >> -> 256
* AND -> 256 + (256 X 4 X 4)
* ^ -> 2 X 256 X 4
* += -> 256 X (+=)
* Add – 256\*14

|  |  |  |
| --- | --- | --- |
| **Operation** | **Number of times** | **Asymptotic value** |
| **Shift** | 2 X 256 | **2n** |
| **AND** | 2 X {256 X 3 + [256 + (256 X 4 X 4)]} | **2\*(3n + (n+ n\*(n/wSize)2)** |
| **XOR** | 2 X {256 X 4 + [256 + (256 X 4 X 4)]} | **2\*(4n + (n+ n\*(n/wSize)2)** |
| **OR** | 2 X (256 X 4 X 4) | **2\*(n\*(n/wSize)2)** |
| **neg (-)** | 2 X 256 | **2n** |

**Add:**

14 operationsfor each columns

**matByVec:**

30 \* 256 operations

1. **Communication Analysis**

**topelitz\_Party1**:

Receive 256 uint64\_t bits (mx)

Send 2 x 256 uint64\_t bits (ma,mb)