Verification

C code:

- GCC-4.8 is used to compile the c code
- Since there were no available test vectors for the evaluation, some random numbers were manually assigned to variables (x, y, and q are selected randomly) in the code for testing the algorithm.
- For evaluating, the outputs of all algorithms are printed out on a terminal based on the formula mentioned in the paper. Take A2B as an example. The output of this function is stored in the z variable. For comparing the correctness of the output, based on the formula in the paper, it should be $\oplus_{i=0}^{n-1} x + \oplus_{i=0}^{n-1} y = \oplus_{i=0}^{n-1} z$. shares of z and the formula is printed out in the terminal.

VHDL Code:

- Vivado 2021.2 has been used to simulate and synthesize VHDL codes.
- xc7a12tcsg325-3 has been selected as the target FPGA
- For evaluating the correctness of the output, the output of every component is compared with shares of the output in the corresponding c function.

Simulation Results:

For n = 4 and k = 8

```
x[i] &   y[i] =  48 ------------- SecAnd(x,y)            =  48
x[i] +   y[i] =  47 ------------- SecAdd(x,y)            =  47
x[i] +   y[i] =  47 ------------- SecAddGoubin(x,y)      =  47
x[i] +   y[i] =  47 --------------SecAddQ(x,y)           =  47
x[i] +   y[i] =  47 --------------SecAddQSimplified(x,y) =  47
x[i] +   y[i] =  47 --------------A2B(x,y)               =  47
x[i] +   y[i] = 207 --------------B2A(x,y)               = 207

########################################################################
########################################################################

SecAnd Shares        = {0x00,0x34,0x54,0x50}
SecAdd Shares        = {0x02,0xcc,0x2f,0xce}
SecAddQ Shares       = {0x2f,0x00,0x00,0x00}
SecAddQSimple Shares = {0x1e,0xdc,0x23,0xce}
A2B Shares           = {0x00,0xce,0x2d,0xcc}
B2A Shares           = {0xcd,0x00,0x00,0x02}
Program ended with exit code: 0
```

output of c code for n = 4, k = 8

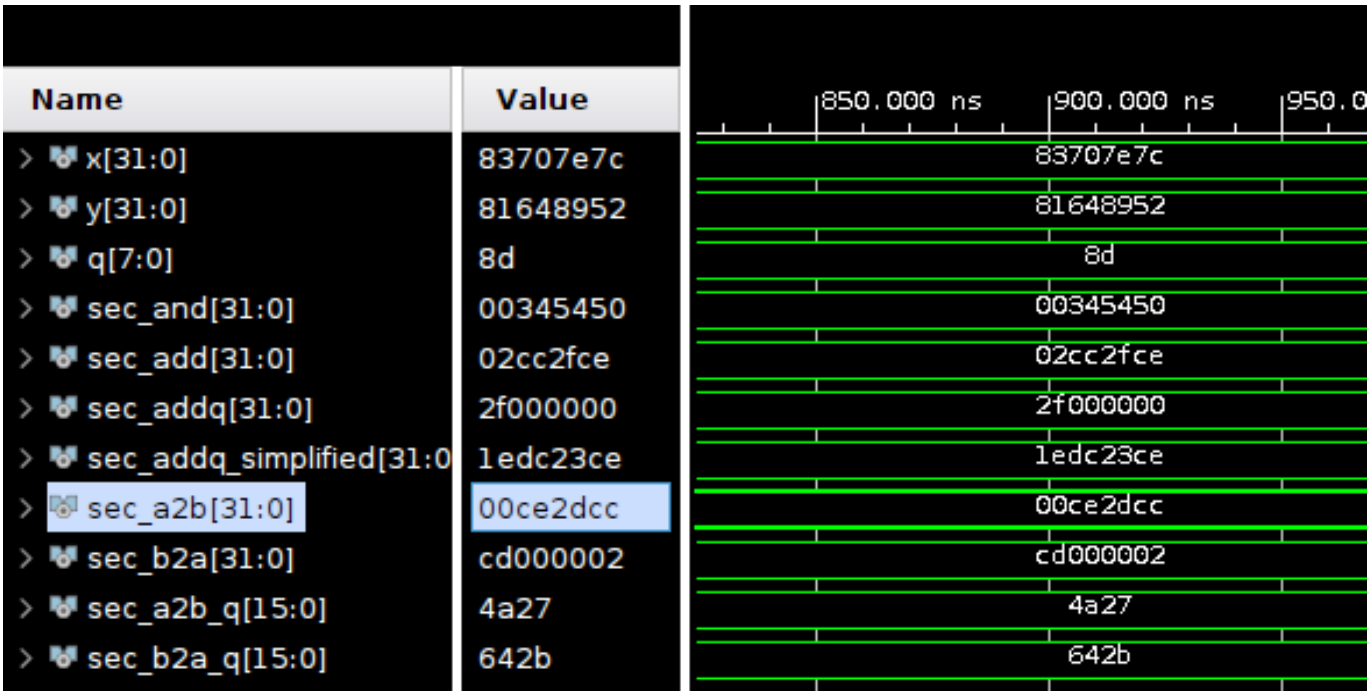
For n = 2 and k = 8. A2Bq and B2Aq are generated based on shares on x variable.

```
x[i] &   y[i] =   2 ------------- SecAnd(x,y)            =   2
x[i] +   y[i] = 221 ------------- SecAdd(x,y)            = 221
x[i] +   y[i] = 221 ------------- SecAddGoubin(x,y)      = 221
x[i] +   y[i] =  80 --------------SecAddQ(x,y)           =  80
x[i] +   y[i] = 106 --------------SecAddQSimplified(x,y) = 106
x[i] +   y[i] = 221 --------------A2B(x,y)               = 221
x[i] +   y[i] = 217 --------------B2A(x,y)               = 217
A0   +   A1   = 109 --------------A2Bq(x,y)              = 109
B0   ^   B1   =   2 --------------B2Aq(x,y)              =   2

########################################################################
########################################################################

SecAnd Shares        = {0x53,0x51}
SecAdd Shares        = {0x0d,0xd0}
SecAddQ Shares       = {0x50,0x00}
SecAddQSimple Shares = {0xb8,0xd2}
A2B Shares           = {0x13,0xce}
B2A Shares           = {0xd7,0x02}
A2Bq Shares          = {0x4a,0x27}
B2Aq Shares          = {0x64,0x2b}
Program ended with exit code: 0
```

output of c code for n = 2, k = 8



VHDL simulation