Encrypted Arithmetic Operations Using a Dynamic Key

ABSTRACT

With the rising concern of malicious agents, it is becoming more important to research hardware security approaches that embed security throughout the computation. Though there is existing work at creating new encryption schemes (e.g., homomorphic encryption) the interface between these schemes and most modern-day architecture is incompatible. Thus, we focus on creating secure Arithmetic Logic Unit (ALU) Components; such as add, subtract, multiply, divide, etc. that are not only encrypted but implement a dynamic key that changes at certain intervals. This research investigates the question on whether it is possible to implement such an idea in a multitude of common ALU components. By achieving hardware encryption with a dynamic key, it will prove to be a multitude of times harder to steal data from systems that integrate this technology compared to systems without it. If possible, to implement certain components research will go into benchmarking them versus common designs.

Motivation/Problem

As malicious agents continue to exist in the world today novel solutions must be investigated to protect data. Fully homomorphic encryption is one method that has been looked into but leads to complex hardware that does not fully compatible with standard systems and long execution times due to algorithm complexity. To integrate better with modern systems and lower the computation time we look into using a dynamic key with partially or somewhat homomorphic encryption. Not only will this be a different approach than fully homomorphic encryption it will hopefully mitigate the risk of side-channel attacks that can extract the key. This is do the constant changing of the key at predetermined intervals.

With this idea in mind, we aim to solve whether the implementation of a somewhat/fully homomorphic method with a dynamic key is possible. If it is possible, how will it compare to other standard designs in areas such as latency?

Hypothesis & Methods

It is assumed that using a **somewhat homomorphic encryption** method we will be able to implement common arithmetic operations in hardware. From there we would like to investigate if implementation of a key that changes on specified intervals is possible and can be used.

We will work to find functions to represent encrypted arithmetic operations using a simple shift cipher to prove the concept in a reasonable amount of time and attempt to implement the encrypted operations in a 4-bit ALU using Verilog HDL.

The shift encryption is defined as $\mathrm{E}(\mathrm{x}, \mathrm{k}) = F(x) + k$, where k is the encryption key. And E(x,k) represents the encrypted function a F(x) represents the standard function.

Current Work & Results

Functions for addition and multiplication based arithmetic operations have been found. The following equations should hold for any real number.

Addition

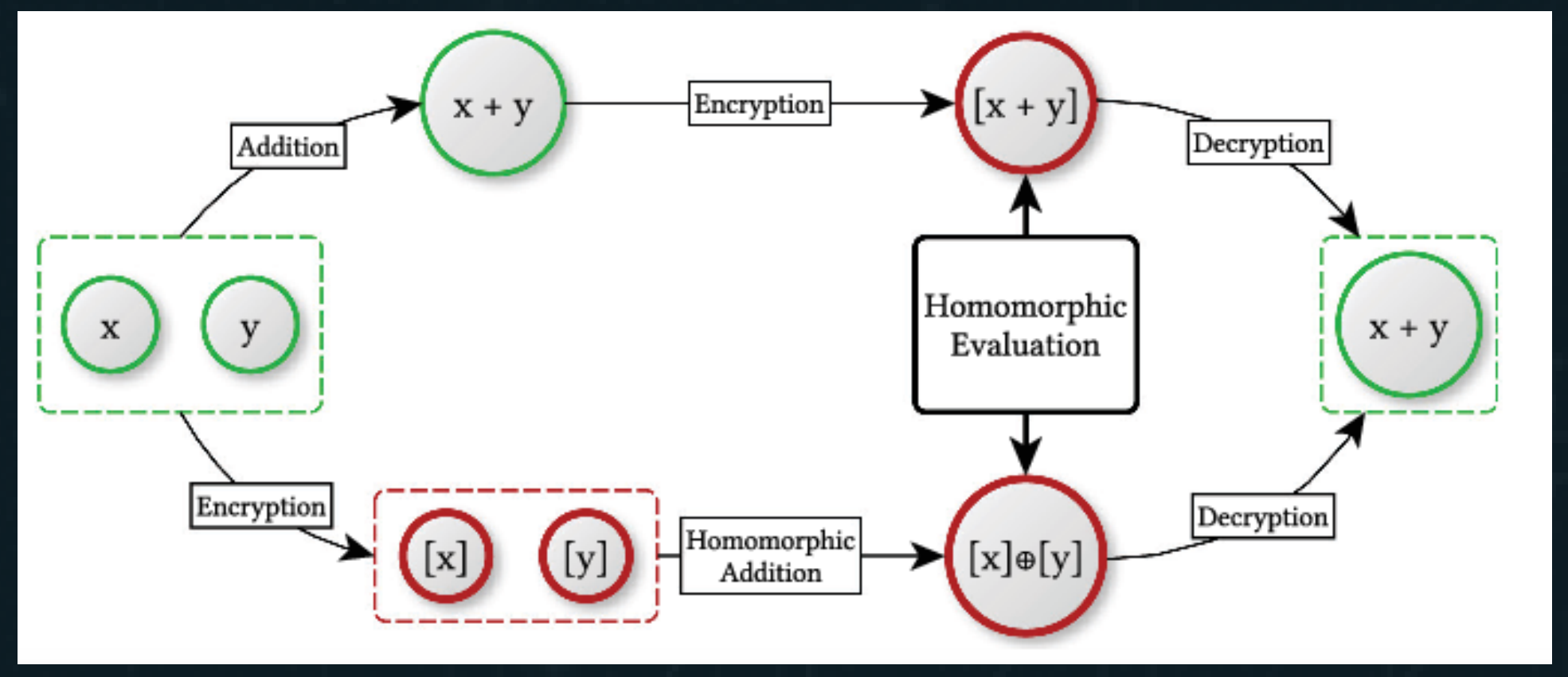
$$F(A, B) = A + B$$

$E(A_e, B_e, k) = A_e + B_e - k$, where A_e & B_e $represents\ encypted\ A\ and\ B.$

Multiplication

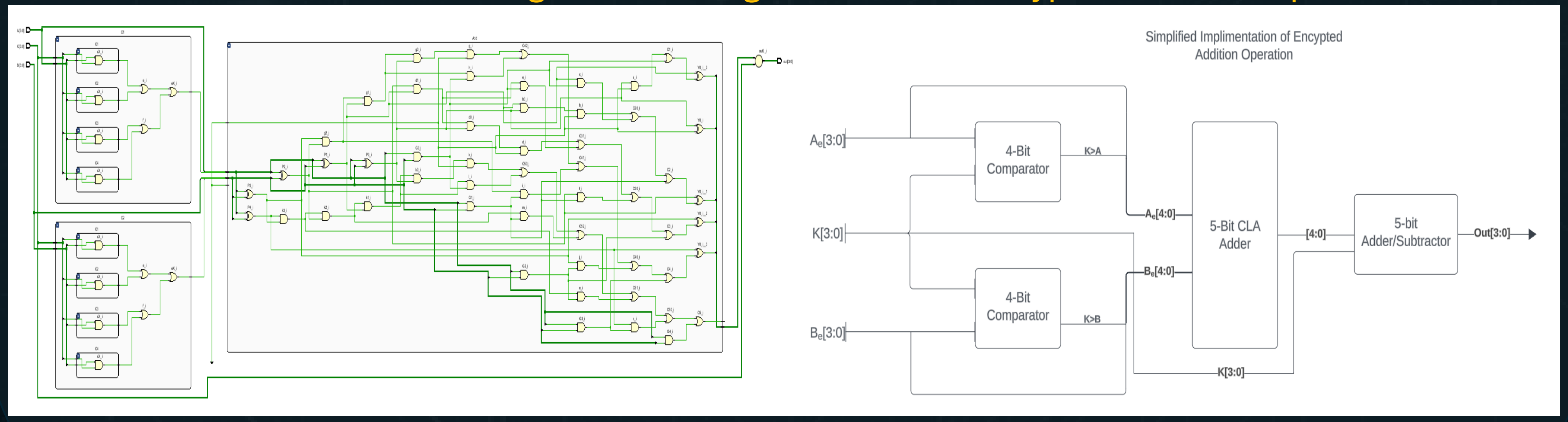
$$F(A, B) = A * B$$

$$E(A_e, B_e, k) = A_e B_e - k(A_e + B_e - k - 1$$



Wood, A., Najarian, K., & Kahrobaei, D. (2020). Homomorphic Encryption for Machine Learning in Medicine and Bioinformatics. ACM Computing Surveys (CSUR), 53, 1 - 35.

Hardware has been designed in Verilog HDL for the encrypted addition operation.



Discussion

Originally work investigated the use of an XOR cipher but it is not even partially homomorphic and so we switch to a shift cipher for our proof-of-concept encryption methodology.

Since the functions we found work with any real numbered inputs, the system can change the key when power is cycled. The new data will now use the new key. Issues arise with cache and other data storage invalidation from key changes. This needs to be investigated in the future.

Future Work

In the future we will look into the possibility of the logical functions in an ALU staying secure using encryption. As well as the implementation of multiplication in hardware.

After a working ALU with the working encrypted components has been designed and verified it will be benchmarked against other non-encrypted ALU designs to compare the latency.

Long term work is to implement more secure encryption like RSA if the results from the proof of concept look promising.

Acknowledgements

This work is funded by the DRACO Lab at the University of Central Florida.
Special thanks to Dr.Mike Borowczak and Andey Robins

Jarred Long
jarred.long@UCF.EDU

Arturo Lara
ar076023@UCF.EDU

DRACO LAB | www.ece.ucf.edu/DRACO
Director: Dr. Mike Borowczak
Dept. Of Electrical & Computer Engineering
College of Engineering and Computer Science
University of Central Florida

UCF

Student Scholars Symposium – March 26-27th 2024

DRACO