COM S 424 Final Project Proposal

For my final project, my goal is to create an image encryption/decryption method that can be run in parallel. I have found multiple research papers providing the feasibility of this idea and the outcomes. I will demonstrate theoretical outcomes, expected outcomes, and actual outcomes. The benefit I expect to observe is for the encryption and decryption of images to be reduced from O(n^2) to O(n) at the maximum level of parallelization and to fit in the range of O(n)-O(n^2) for limited parallelization.

Two supporting articles:

<https://iopscience.iop.org/article/10.1088/1757-899X/981/2/022017/pdf>

<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9025433/>

Abstract

As the use of distributed cloud-network solutions rise, more and more data will be stored in distributed, external networks. It is important to practice safe encryption of data that will be stored on external devices. However, the cost of encrypting and decrypting massive datasets is extremely high. To mitigate these effects, it is in a user’s best interest to decrypt as infrequently as possible. To effectively secure data, and limit the need for decryption, we will be implementing homomorphic encryption on an image. To maximize efficiency, and prepare the data for distribution, we will also be implementing a chunking algorithm to demonstrate benefits in speed, and security with the ability to operate on encrypted, distributed data.

Introduction

Homomorphic encryption is a technique for encrypted data in a way that it may still be operated on while encrypted. In the case of images, you can still perform image operations, like adding filters, combining images, or other modifying techniques without the need to decrypt the image. Traditionally, images stored on cloud-server networks would need to be downloaded to be modified, then re-uploaded to the cloud -server network. Now, add on encryption before uploading, and decryption after downloading. This becomes incredibly expensive when working with massive sets of images, crippling reach, and capability. Homomorphic encryption offers a solution, such that it is not only asymmetric, meaning that more data can be encrypted with the same public key from anywhere, but that it allows the public key holder to modify the data while it is encrypted. There are far superior techniques developed for Full Homomorphic Encryption (FHE) than I can develop in the span of this project, and I am not trying to re-invent the wheel. I will be using partial homomorphic encryption with Paillier’s algorithm.

Traditionally, encryption of an image is quite slow. Assuming a full-color image is an MxNx3 matrix of 8-bit integers, where the last layer is the RGB value, the complexity of encrypting the image would be O(MxNx3). When working with massive datasets for High Performance Computing (HPC), having to constantly encrypt and decrypt images for computation would be quite slow. We can first accelerate this by the implementation of homomorphic encryption, to limit the amount of decryption required to perform operations on an image. Then, by further breaking up an image into chunks, based on the number of processors in the HPC Cluster, we can perform encryption, decryption, and operations on the image at a much smaller scale. And we can securely store an image in a cloud-server network, as these chunks. By implementing a novel chunking algorithm for High Performance Computers, one that has multiple processors, we can split the image into smaller ‘chunks’ to perform the encryption, and decryption on. These chunks can also be further distributed on a cloud-server network, then recollected, as chunks, for their decryption. The chunking algorithm should also separate them in a way that they may be able to be operated on, using the homomorphic encryption, without the need for recollection.

Problem Being Solved

As described, image encryption, and decryption are expensive. The goal of this project is to limit the expense from every angle. There are two major facets to this solution: Encryption/Decryption, and operations on an encrypted image. By implementing this solution, we should limit the time for encryption/decryption with chunking, limit the number of decryptions, and limit the complexity of each operation with the chunks.

2 Related Work

Encryption is a broad term for morphic some data in a reproducible way such that any viewer would no longer understand it until it has been decrypted. Modern encryption employs several mathematical techniques to complicate the encryption making it more difficult for a bad actor to decrypt and understand the data, all while remaining decryptable to the good actor without any data loss. Homomorphic encryption is the mathematical technique of encrypting data such that any operation performed on the data while encrypted will have the same effect on its decrypted counterpart.

2.1 Paillier’s Algorithm

In 1999, Pascal Paillier proposed an algorithm for homomorphic encryption that would allow for an unlimited number of addition operations to be performed on the encrypted data. Like traditional asymmetric encryption models, this employs dual-prime key generation. This system proposed using a probabilistic model for computing random values to

3 Approach

The primary objective of this research is to