An Accelerator to Additive Homomorphism to Handle Encrypted Data

Angelin Gladston, Anna University, Chennai, India*
S. Naveenkumar, Anna University, Chennai, India
K. Sanjeev, Anna University, Chennai, India
A. Gowthamraj, Anna University, Chennai, India

ABSTRACT

Homomorphic encryption provides a way to operate on the encrypted data so that the users can be given with the maximum feasible privacy. Homomorphic encryption is a special kind of encryption mechanism that can resolve security and privacy issues with rich text. Research gap is the performance overhead associated with this which poses a barrier to the real time implementation of this scheme. The objective of this work is to implement an algorithm to achieve increased performance and faster execution when compared with a classical cryptographical algorithm, the Paillier Cryptographical Algorithm, which is predominantly used to achieve additive homomorphism and analyse the performance gain obtained by this algorithm. The same algorithm is also integrated into an encrypted database application, CryptDB, developed by the MIT, as a replacement to the Paillier algorithm used in the application. The derived algorithms are 2600 time faster in key generation, 5 lakh times faster in encryption, and 3500 times faster in decryption, when compared with the Paillier algorithm.

KEYWORDS

Additive Homomorphism, CryptDB, Cryptography, Encrypted Data Computation, Encryption, Pailler

1. INTRODUCTION

In this modern world, people use a wide range of applications and they rely on these applications for their data security. There are numerous encryption algorithms in the world and every algorithm serves a specific purpose. But as encryption methods evolve, the art of breaking encryption algorithms also evolved. Simple encryption methods are used till late 19th centuries. The problem with those were, they can be easily broken by a human mind without any additional resources. But after late 19th century encryption algorithms are made with super protection, so that a human could not break it without any resources. But all it took was days or weeks to break them. And then the invention of computers led to public key, private key encryption methods. An advanced hybrid scheme of public key encryption is discussed by Jung et. al., (2015) and elaborated a new method called homomorphic encryption which is explained below.

DOI: 10.4018/IJBDCN.341589

*Corresponding Author

In Cryptography, encryption is a process of encoding certain sensitive information and store it in a manner that the encrypted data cannot be read without a secret key. With different security attacks being developed every day, there is a dire need for these algorithms to hold the data in an encrypted manner. Some of the common encryption algorithms known to the world are AES, DES, Blowfish, Twofish, RSA algorithms, etc. One such well used algorithm (Rajesh et. al., (2023) is Homomorphic encryption. It was proposed by Rivest, Adleman, and Dertouzos in 1978. The early stage of homomorphic encryption was really a tough process (Ryu et. al., (2023). Yang et. al., (2012) did an impressive job on finding and proving the homomorphic properties of integers, which provides a vivid view of homomorphic encryption. Almost 90 percent of banks either rent cloud storage or hire a group of technicians from an organization to help them store client's data. They will encrypt and decrypt client's data on sending and receiving. But while operating them they can't do that. This particular vulnerability leads to a situation where everything that has been done before had gone wasted.

These kinds of problems can be solved using methods like Homomorphic encryption. A new fully homomorphic encryption method is discussed by Mahmood et. al., (2018). Besides that, how multiple and multistage partial homomorphic encryption is used to develop a fully homomorphic encryption method is described and these two technologies are used in cloud applications. Homomorphic encryption allows us to perform calculations on encrypted data without decrypting it in the first place, and when decrypted the output is the same as if these operations are performed on the unencrypted data. For example, there is an algorithm called Paillier, which can perform addition and multiplication on encrypted data. Other than these operations Homomorphic addition and paillier are used in many other departments. As in Li et. al., (2021) the privacy concern in IoT is a big challenge in IoT applications and services, so this problem is encountered with Homomorphic encryption. In addition to this, paillier is used in homomorphic volume rendering as discussed by Mazza et. al., (2021).

However, the classical paillier based Homomorphic encryption does provide results for few operations, the performance of paillier algorithm is not up to mark and it consumes more cycles during runtime. Paillier encryption method's homomorphic properties are thoroughly analysed in Nassar et. al., (2015). This work theoretically stated an algorithm that can beat the traditional Paillier algorithm in addition operation. They named it Fast Additive Homomorphic Encryption, for sort FAHE. The basic idea of this paper is based on fast additive homomorphic encryption. ACD is also the same method that has been used in Eduardo et. al., (2020).

Zhang et. al., (2016) provides enough information about fully homomorphic encryption method over integers. Other than that it is also specifies how to accelerate the already exist homomorphic mechanism but only for integers. The proposed algorithm defeated Paillier in every single efficient test. But algorithms aren't of use without an application to employ them. One such application where the FAHE algorithm can be effectively applied to yield better results is, Cryptdb. Cryptdb is an MIT licensed product. This Crypt- dDB executes encrypted queries over the encrypted data stored. It provides basic database operated on the encrypted data using encrypted queries. Cryptdb already employs Paillier to perform addition and multiplication. This algorithm consumes a lot of cycles during runtime and it results in increased runtime of the encrypted queries run by cryptdb. The FAHE algorithm can be used as an effective replacement in this case, considering it has better performance over the paillier system without the loss of functionalities.

There are numerous encryption algorithms applied in real world application. Specific algorithms provide specific use cases. One such algorithm is Homomorphic Encrpytion. Homomorphic encryption allows us to perform operations on encrypted data and the results would as similar as though the operations are performed on original data.

However, the classical paillier based Homomorphic encryption does provide results for few operations, the performance of paillier algorithm is not upto mark and it consumes more cycles during runtime. This makes the paillier algorithm flawed and there is a dire need to replace the existing algorithm with a much faster one and with stronger encryption as the former. Also the paillier system uses asymmetric key system which uses different keys for the processes of encryption and decryption.