

**CHENNAI – PONDICHERRY**

**Translating Algorithms to handle Fully Homomorphic Encrypted Data on the Cloud**

**Abstract:**

Cloud provides large shared resources where users (or foundations) can enjoy the facility of storing data or executing applications. In spite of gaining convenience of large resources, storing critical data in cloud is not secured. Hence, cloud security is an important issue to make cloud useful at the enterprise level. Data encryption is a primary solution for providing confidentiality to sensitive data. However, processing of encrypted data requires extra overhead, since repeated encryption-decryption need to be performed for every simple processing on encrypted data. Hence, direct processing on encrypted cloud data is advantageous, which is supported by homomorphic encryption schemes. Fully Homomorphic Encryption (FHE) provides a method of performing arbitrary operations directly on encrypted data. This seemingly magical idea is a welcome to cloud computing. However, there are several challenges to overcome for making the technology viable in practical applications. In this paper, we make an initial effort to highlight the problem of translating algorithms that can run on unencrypted or normal data to those which operate on encrypted data. Here, we show that although FHE provides the ability to perform arbitrary computations, its complete benefit can only be obtained if they also allow to execute arbitrary algorithms on encrypted data. In this pursuit, we provide techniques to translate basic operators (like bitwise, arithmetic and relational operators), which are used for implementation of algorithms in any high level language like C. Subsequently, we address decision making and loop handling and related data structures which are vital to realize when the controlling variables are encrypted. Since, termination is a major challenge while handling encrypted data, we propose a method of handling termination by message passing between server and client.

**Existing System:**

Fundamental encrypted additions and multiplications on single bits are defined and implemented using integers. Survey and further efficiency enhancement on FHE has been reported. In some advancements have been proposed to implement faster encryption schemes.

Further, recent developments of FHE have been discussed. In searching and sorting on FHE data have been investigated. In researches have been performed to conceptualize encrypted processor using FHE as underlying scheme. However, it is stated that “it must be emphasized that homomorphy is a theoretical achievement that merely lets us arithmetically add and multiply plaintexts encapsulated inside a ciphertext. In theory, this allows the execution of any algorithm complex manipulations like text replacements or similar, but putting this to practice requires the design (compilation) of a specific circuit representation for the algorithm at hand.

**Proposed System:**

We first identify the basic components of an algorithm and then try to realize them in the encrypted domain. We thus map the unencrypted variables to a data structure storing corresponding encrypted variables or ciphertexts.

We use the multi-precision library to describe the corresponding data structures. Subsequently, we target algorithms in both non-recursive and recursive version and discuss their realizations while operating in the encrypted domain. A program being a sequence of instructions can be converted to the encrypted domain by replacing the instructions by their homomorphic equivalents. The instructions often set values of control variables, which decide the control flow path in the program. The challenge remains that in the homomorphic counterpart, these variables are also encrypted. We present a mechanism of translating these constructs using FHE-Mux (FHEMultiplexers) to operate on the encrypted control variables. Interestingly, these components do not leak to an adversary the outcome of the decision.