ABSTRACT

Storage and management in cloud services are of growing importance due to their cost-effective approaches in using large shared resources. However, with public access to the information in clouds, security is a very important issue. Confidentiality of data can be preserved by encrypting critical data before storing it in the cloud. However, bringing the data back and processing after decryption also leads to an overhead and outweighs the advantage of cloud computing. Homomorphic encryption which allows operations directly on the encrypted data is a solution to reduce this overhead. Fully homomorphic encryption (FHE) goes a step further and provides an effective primitive to perform arbitrary operations on encrypted data. With the support of FHE, cloud can evaluate any functions on encrypted data without having access to the secret key and without knowing the result. Significant research is being performed to make the original scheme more efficient and practical for real-life applications in recent few years. Motivation of our work is from the fact that for developing suitable tools to execute algorithms operating on FHE data on general purpose computers, one also needs to architect suitable translations of algorithms operating on unencrypted data to those which operate on FHE encrypted data. Further, FHE schemes are by design *circuit-based* and are not amenable to a non-circuit computation. However, classic algorithms are mostly non-circuit based, implying that they are not described in terms of logical gate level operators, like AND-OR multiplexers. This inspires us to develop suitable synthesis techniques to handle algorithms which operate on FHE data and find methodologies to improve their performances. In this dissertation, we start with the age old problem of sorting, and propose the first circuit based implementation of FHE encrypted sort. We extend our work to identify the basic components of an algorithm and then try to realize them in encrypted domain. Subsequently, we target algorithms in both non-recursive and recursive versions and discuss their realizations while operating in the FHE domain. We identify that detecting termination while implementing encrypted algorithms on existing unencrypted processors is a major challenge and we propose a possible solution to this problem with the use of encrypted processor. While existing encrypted processors are chosen plaintext attack (CPA) insecure due to absence of randomization in underlying encryption scheme, this motivates us to design FHE encrypted URISC (FURISC) architecture which works with single opcode. Finally, we analyze the challenges of realizing such FHE encrypted processor with very large parameters in reconfigurable hardware and propose design of addition and multiplication building blocks for such processor with suitable compression techniques.

Keywords: Fully homomorphic encryption (FHE), Sorting, Algorithm Translation, URISC, Reconfigurable hardware, Hardware implementation.