(k – n) Oblivious Transfer Using Fully Homomorphic Encryption System

Mohammed Kaosar¹, Quazi Mamun¹, Rafiqul Islam¹, and Xun Yi²

¹ School of Computing and Mathematics, Charles Sturt University, Australia
² School of Engineering and Science, Victoria University, Australia
{mkaosar,qmamun,mislam}@csu.edu.au, xun.yi@vu.edu.au

Abstract. Oblivious Transfer (OT) protocol allows a client retrieving one or multiple records from a server without letting the server know about the choice of the client. OT has been one of the emerging research areas for last several years. There exist many practical applications of OT, especially in digital media subscription. In this paper, we propose a fully homomorphic encryption based secure k out of n oblivious transfer protocol. This novel protocol, first ever to use fully homomorphic encryption mechanism for integers numbers, allows the client choosing its desired records by sending encrypted indexes to the server, server works on encrypted indexes and sends back encrypted result without knowing which records the client was interested in. From the encrypted response of the server, the client only can decrypt its desired records. The security analysis demonstrates that, the desired security and privacy requirement of OT is ensured by the proposed protocol. Some optimizations are also introduced in the proposed solution to reduce transmission overhead.

Keywords: Oblivious Transfer, Homomorphic Encryption, Private Information Retrieval, Data Outsourcing.

1 Introduction

In the current world, the use of information technology has increased tremendously. Consequently, secure storage, transmission and retrieval of information become one of the top concerns in the IT era. The diversity of devices, applications and infrastructures have increased this concern by another fold. The privacy of information in any transaction is no more a small issue. Private Information Retrieval (PIR) and Oblivious Transfer (OT) are some of the cryptographic protocols that ensures the privacy of the user in retrieving information from a storage or a server. Unlike PIR, OT ensures the server security too by not allowing the user retrieving unauthorised record(s). OT has been used in many applications including certifying email and coin flipping [1], simultaneous contract signing [2], digital right management [3], e-subscription to sell digital goods [4], privacy preserving data mining in distributed environment [5] etc.

To understand the basic principle of OT protocol, let us consider an example: let us say, a server stores n number of digital contents or records of information...
$x_1, x_2, \ldots, x_n$. Clients or users need to subscribe with the server to access an item. In such e-subscription, there will be two requirements to be fulfilled from the server’s and the client’s point of view respectively: (i) the client should not be able to retrieve any item(s) which it did not subscribe for and (ii) the server is not allowed to know which item(s) the client retrieved. That is, if the client wants to retrieve or access item $x_i$, OT protocol ensures that server cannot learn the value of $i$ and the client cannot learn any $x_j$ for all $j \neq i$.

In this paper our proposed solution uses a secure cryptographic protocol, particularly the fully homomorphic encryption over integer numbers proposed by Dijk and Gentry [6] in 2010, to ensure data privacy of the client. The server’s security is ensured by encrypting all of its records using a symmetric key encryption system such as, AES [7] or DES [8]. $k$ out-of $n$ OT can be achieved by repeating 1 out-of $n$ OT protocol $k$ times. This approach incurs extremely high overhead. In this paper, we have proposed some optimizations in the $k - n$ OT protocol. It transmits the encrypted database only once at the beginning of the protocol. The server uses separate keys to encrypt each record using a symmetric key encryption technique. The protocol only allows the desired keys to be decrypted by the client. On the other hand, the client encrypts its choices using the homomorphic encryption technique and transmits to the server. The server encrypts and manipulates keys and indexes using the same technique without being able to decrypt any of the choices of the client. The fully homomorphic encryption of Dijk and Gentry is as strong as the approximate Greatest Common Divisor (GCD) problem (more detail of approximate GCD can be found in [9]). The security analysis shows that the proposed protocol ensures both the server’s and the client’s requirements.

The rest of the paper is organized as follows: Section 2 describes some background knowledge on the topic of the paper including the fully homomorphic encryption system which is used in the proposed protocol, Section 3 and 4 discuss our proposed model and the protocol, Section 5 discusses the security and performance analysis and finally, Section 6 concludes the paper with some hints towards the future research directions.

2 Background and Related Work

This section discusses about various OT protocols and existing solutions and the definition of homomorphic and fully homomorphic encryption system. This section also discusses how the fully homomorphic encryption for binary digits is extended to work for integer numbers.

2.1 Types of Oblivious Transfer Protocol

OT can be of three basic types:

- $1$-out-of-$2$ $(1 - 2 \text{ OT})$:
  
  1 out-of 2 oblivious transfer protocol allows the client retrieving one item