Assignment 4:Secure Two Party Computation

Privacy Enhancing Technologies (201500042)

(Total number of achievable points: 20)

Issue date: 06 May 2016; **Due date: 26 June 2016, 23:59 CET** *(hand in via BB)*

1 Introduction

In this assignment you are going to design a privacy-preserving data processing scheme using secure two party computation. Initially you are going to generate an encrypted dataset, then perform some queries on encrypted data using secure comparison protocol. Furthermore, you are expected to provide computational analysis and real-time results for your design.

This is an individual assignment. For this assignment you can implement the scheme using either Java or Python programming languages. You have to hand in your source code (plain text) and a document (PDF) with your answers on Blackboard. Please mention your name and student number in submission. You can submit an archive (ZIP, GZip, etc.) containing both the report and the source code, but please do not submit a RAR file.

2 Privacy-Preserving Data Processing

Privacy-preserving data processing aims to process encrypted data using a homomorphic cryptosystem and secure two party computation. In this assignment you are going to implement a simple privacy-preserving data processing scheme to find the number of people who are older than a certain age and furthermore, you are going to compute the total income for that group of people.

Figure 1 overviews the system design. The scheme has two parties as Alice and Bob. Alice is a semi-trusted party who has access to an encrypted database and is authorised to perform some queries on encrypted data. She holds public key (pk) of database. To be able to see the result of queries Alice needs to collaborate with Bob who holds the private key (sk) of database. However, since she does not want to leak information about her queries, they have to use a secure two-party computation scheme to collaborate.

2.1 Homomorphic Encryption

Homomorphic cryptosystems enable users to perform arithmetic operations on encrypted data. In this assignment, you are going to use Paillier cryptosystem which is an additively homomorphic encryption scheme. Below the setting and homomorphic properties of the cryptosystem is explained. For this assignment you do not have to implement Paillier cryptosystem, instead we are providing you an implementation in Java and Python languages.

Additive Homomorphic Properties of Paillier:

- $E(m_1) \cdot E(m_2) = E(m_1 + m_2)$

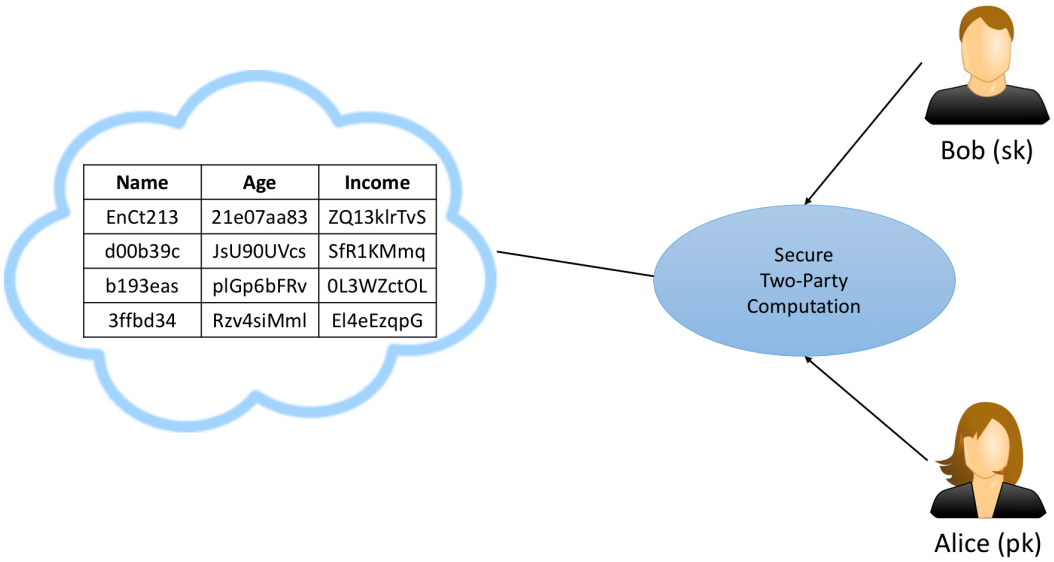

Figure 1: Overview of Privacy-Preserving Data Processing Scheme

- $E(m)^c = E(m \cdot c)$

2.2 Secure Comparison Protocol

The protocol you need to implement is given in Figure 2 together with an example in Figure 3.

[a], [b] are the inputs of comparison protocol. ℓ is the bit length of the inputs in plaintext. r is a random number which is $80 + \ell + 1$ bits. The result is [1] if $a > b$, [0] otherwise. [.] represents encryption.

When you have to decrypt the final result of your computation (for example the total number of people whose ages are greater than x), to prevent information leakage to Bob, you have to perform a secure decryption mechanism as follows:

Alice: $[C'] = [C] * [r] = [C' + r]$ where r is a $32 + 80$ bits random number.

Bob: decrypt $[C']$

Alice: $C = C' - r$

3 Assignment

1. **Data Generation & Encryption** *(4 points)*

   In this assignment, we do not provide you a database, instead you are going to generate it. The generated database should consist of 3 attributes; Name, Age and Income. For each row of database, you are going to select an Age value in range [18,90] and an Income value in range [10000, 100000]. Since you are not going to perform any operation on Name attribute, you can assign any value for it. The size of database is retrieved from user as an input.

(a) *(1 point)* Take database size as an input of your program and generate a random database as specified above.

(b) *(2 points)* Using Paillier cryptosystem, encrypt the database. Select the key size for encryption as 2048 bits. Remember when encrypting one row of database, you have to encrypt each attribute one by one.

(c) *(1 point)* Measure the total time spent for the encryption of database.

2. **Secure Comparison Protocol Implementation** *(8 points)*

(a) *(4 points)* Using instructions in Section 2.2, implement secure comparison protocol.

(b) *(3 points)* Provide a computational analysis of comparison protocol in terms of number of exponentiations and multiplications for one run of protocol.

(c) *(1 point)* Measure the time spent for one run of comparison protocol.

3. **Privacy-Preserving Data Processing** *(8 points)*

In this part, you are going to use the secure comparison protocol in Question 2 for data processing.

(a) *(3 points)* Take an age value x as input and perform secure comparison protocol on encrypted database to find the number of people whose age are greater than x.

(b) *(1 point)* Measure the total time spent to find the number of people.

(c) *(3 points)* Compute the total income for people whose age are greater than x.

(d) *(1 point)* Measure the total time spent to compute total income.

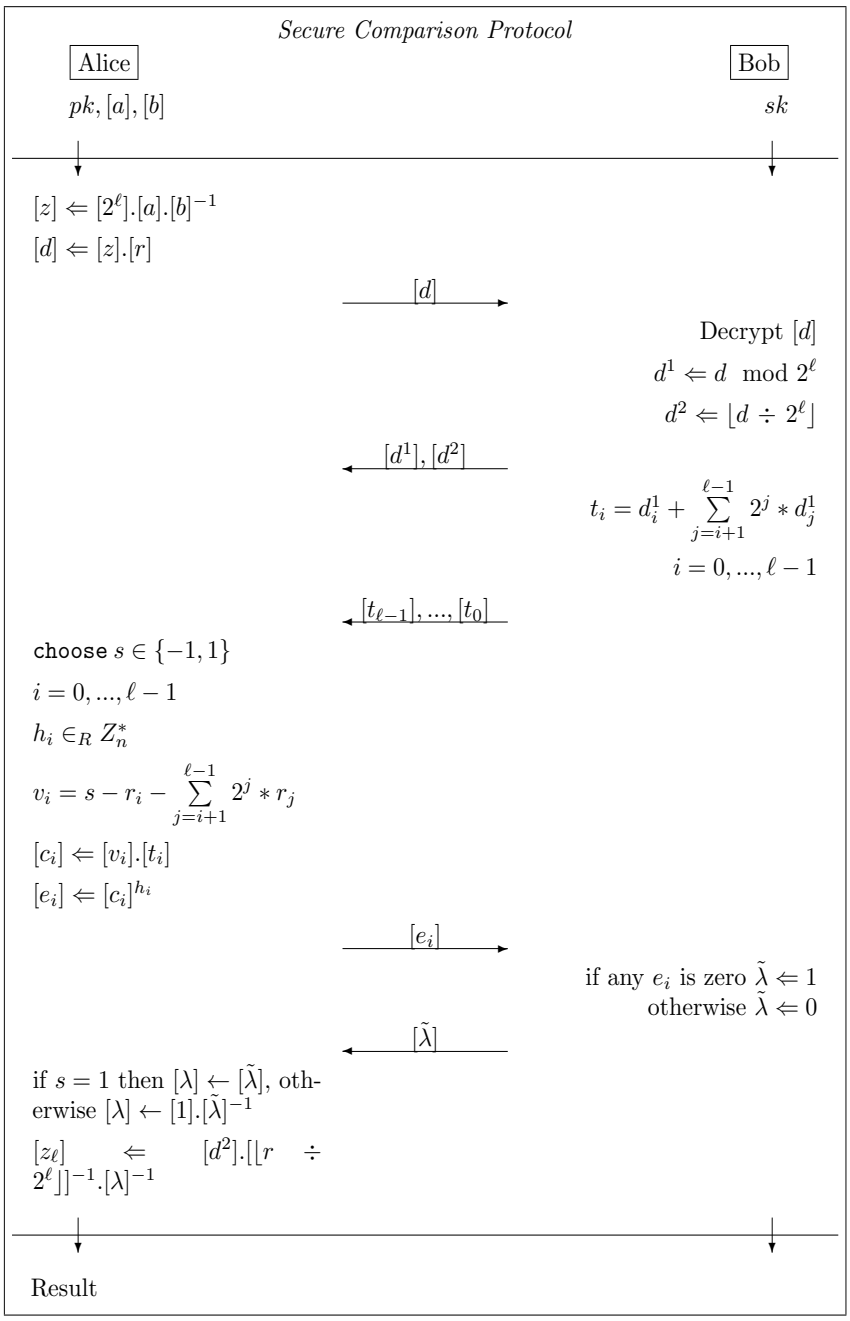

Figure 2: Secure Comparison Protocol

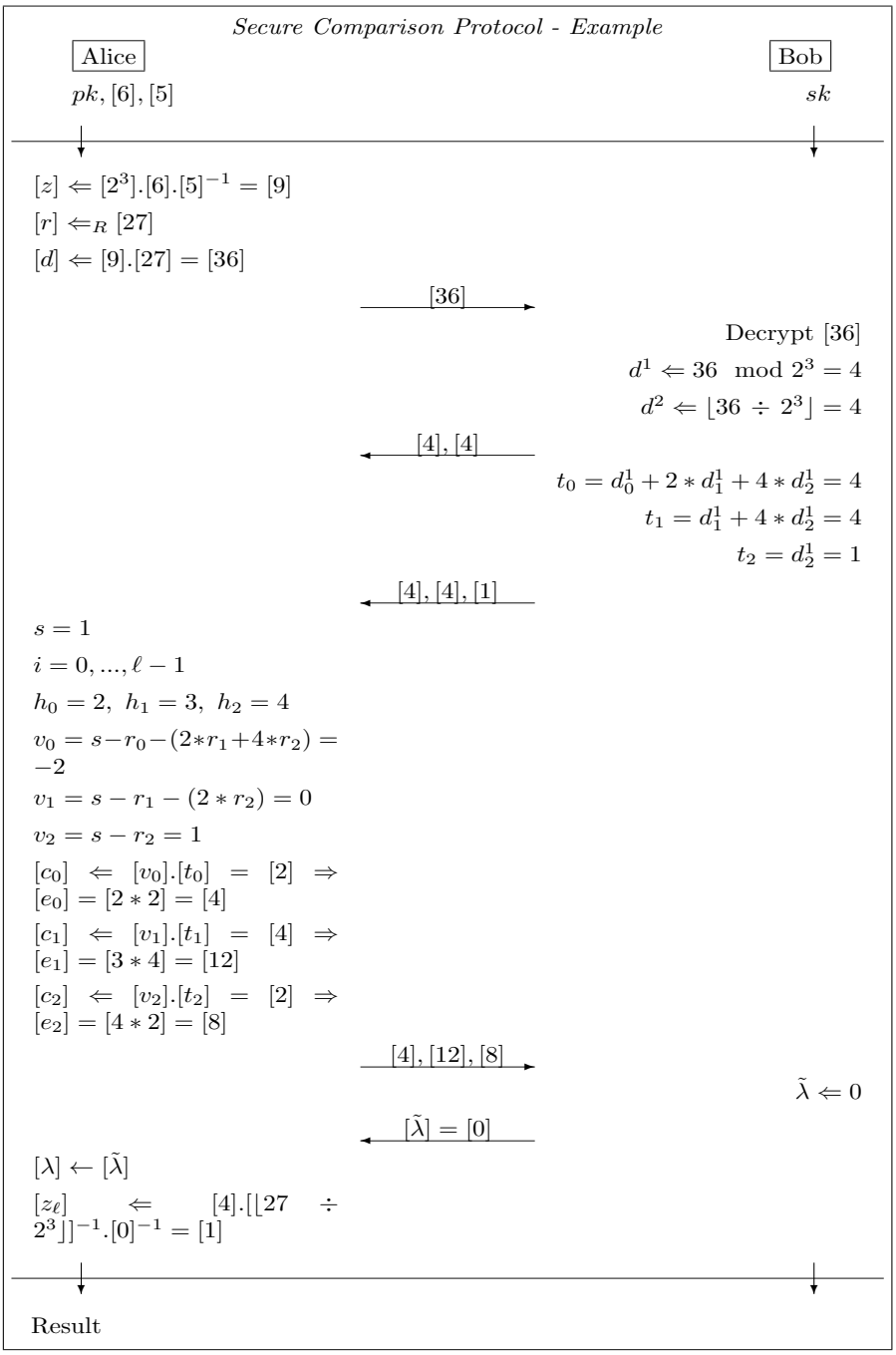

Figure 3: An Example Run for Secure Comparison Protocol