**CSCE 4013 Applied Cryptography – Spring 2016**

**Lab 2: : Privacy‐Preserving Data Aggregation (Report)**

**Syndall Murray**

**Problem Statement**

The purpose of this assignment was to implement a privacy-preserving sum aggregation protocol using a scheme provided. In the scheme, the goal is to guarantee the privacy of each user’s data against the untrusted aggregator, i.e., the aggregator obtains the Sum aggregate without knowing any individual user’s data. Also, any party without the appropriate aggregator key obtains nothing.

**Design and Implementation**

I used java programming to create 4 classes, a trusted authority class, a aggregator class, and 2 user classes.

In the trusted authority class (TA), we have a set of secret S, those secret are divided into a disjoint subsets Si and Sj. Each subset contains 2 secrets, so S contains 4 secrets. The trusted authority sends Si to user 1, Sj to user 2 and S to the aggregator.

In the user programs, each user first receives the secret subset. Then, they derive the two secret from the subset and compute the key k

C:\Users\Kimberly\Downloads\mm.PNG

*t is and interval starting from 1 to n for each aggregation.*

*M is a large integer*

To perform this computation I first compute the hmac of t using the first secret in the received subset using java api. Then, I computed the hash value of the resulted hmac digest and converted the hash value to a BigInteger (JAVA api). We simply repeated the process for the second secret in the received subset, and add both values mod M to compute k.

The cipher text is computed using

C = k + x (user data) mod M

Each user sends C to the aggregator. After each user sends their cipher text they add +5 to their data. The user’s data is just an integer value.

In the aggregator class, we first receive the all the secrets S from the trusted authority. The aggregator computes

C:\Users\Kimberly\Downloads\k0.PNG

This process is done similarly to the user programs but for **all** 4 secrets in S. So during each interval when the aggregator received the cipher text for both users, they will simply add both cipher text and subtract k0 to get the sum.

S = C1 + C2 – K0

**Testing**

|  |  |  |  |
| --- | --- | --- | --- |
| Interval | User\_1 data | User\_2 data | Aggregator Sum |
| 1 | 35 | 15 | 50 |
| 2 | 40 | 20 | 60 |
| 3 | 45 | 25 | 70 |

**Conclusion**

The program provides the necessary functionalities to insure that problem statement requirement are met. The aggregator obtains the Sum aggregate without knowing any individual user’s data. Also, any party without the appropriate aggregator key obtains nothing. The program work successfully with no known errors.