Report

Group Information

> Assignment name, group number and names/IDs for all group members.

Assignment name: DIMY

Group number: 28

z5243835, He Yue | z5200638, Wang Wei

Summary of implementation

> Executive summary that provides a brief introduction to the salient features in the assignment implementation.

We used `git` to help us collaborate. So, in order to reduce conflicts, we made a framework for this assignment before we get started. This framework was not modified largely since created, most interfaces reserved are being used later on (you can use git to check initial commits). Though this cannot be shown directly to you, but this really helps us collaborating since we are not living in the same city.

In order to understand each other's code, we made a directory to store demonstration code to read. Also a readme file was created for us to understand how the classes and functions work. This is also a efficient way improving efficiency.

In addition, if you tries to run our code, please check `readme.md` if you meet problems.

And you can checkout to branch `heyue` to do run the code. This branch now was configured as a "faster" version of main branch

Implement detail

> A brief discussion of how you have implemented the DIMY protocol. Provide a list of features that you have successfully implemented. In case you have not been able to get certain features of DIMY working, you should also mention that in your report.

We implemented task 1 to task 10. First of all, I'd like to give to a overview showing which task corresponds to which feature:

- Elliptic curve Diffie-Hellman key exchange
  - Task 1: generate ephid
  - Task 5: generate encid
- Shamir's secret sharing
  - Task 2: use k-out-of-n Shamir secret sharing
  - Task 3: broadcast n shares @ i unique share per 10 seconds

    - Task 4: reconstruct ephid
- Bloom filter
    - Task 6: store encid
    - Task 7: construct DBF
    - Task 8: construct QBF (combine DBFs)
- Communicate with backend server
    - Task 9: upload QBF
    - Task 10: Upload if test positive

So we used python to implement this assignment, because it's easy to use and also a lot of libraries available. Team members collaborate with each other with `git`.

Some of the features are using third party library like ECDH because it's too difficult to implement in short time. There are still some features we accomplished ourselves like communicating with backend server and bloom filter.

Trade-offs and improvement

> Discuss any design trade-offs considered and made. List what you consider is special about your implementation. Describe possible improvements and extensions to your program and indicate how you could realize them.

In this assignment, we used a library called `pymmh3`. This library provides murmur hash 32, but it's implemented in pure python, which may leads to performance issue. The reason using this library is because it provided us a good portability. In real situation, we may need a faster implementation using C++ or C.

And for this assignment, we can improve our code by improving multithread programming. At the beginning, we think two threads are pretty enough, but later on we realized more threads are needed. So the management of the threads are kind of messy. We can use some models like `producer and consumer` model to improve multithread performance.

Code from the Web / Other books

> Indicate any segments of code that you have borrowed from the Web or other books

We didn't borrow code from others. But there are some references we used. And here are the list:

- UDP broadcasting: socket
- Murmur hash: pymmh3
- ECDH: ecdsa
- bitarray: bitarray
- Shamir's secret sharing: pycryptodome
- Multithreading: threading
- Requests: Requests

Diary

09/04/2021 Read DIMY and assignment specification and plan how to accomplish this work.

10/04/2021 A group meeting was hold, discussing the details about the implementation

12/04/2021 Assign jobs for group members.

He Yue tries was assigned with UDP broadcasting.

Wang Wei was assigned with generating EphID and EncID.

14/04/2021

He Yue was assigned with Shamir's secret sharing

16/04/2021 More detailed job assignment come out.

He Yue majorly deal with details - how client interacts with other devices. (Shamir's secret sharing, broadcasting, multithreading)

Wang Wei plans the framework of the code and implementing features may used in this assignment (Bloom filter, communicate with backend server, ECDH)

17/04/2021 Framework created, merge both group members' code and create git repository.

Branch `wangwei-dev` and `heyue` created

18/04/2021 Merged branch `wangwei-dev` with `main`

He Yue: Tried to use library for Shamir's secret sharing

Wang Wei: Used murmur hash to generate identifier. Now we can generate 16-bytes EphID and EncID.

Added readme file and demo directory to learn each others' code

Implemented bloom filter

19/04/2021 Debugging, fixing conflicts

20/04/2021 Debugging

He Yue: Tries to put Shamir's secret sharing, bloom filter, ECDH together.

Reported bugs in both He Yue's code and Wang Wei's code.

Wang Wei: Added a static function `combine_filters()` to class `bloom_filter`

Added functions to communicate with backend server

Fixed bugs in function `get_shared()` in class `enc_mgr` and bug in `put()` in class `bloom_filter`

21/04/2021 Debugging.

He Yue: Changed another Shamir's secret sharing implementation

Added code printing logs

Wang Wei: Start writing readme.md

23/04/2021 Report.md finished