Assignment COMP9337: DIMY (Did I Meet You)

Group T17- 7
Christopher Flinn z3291100
Hardique Dasore z5456086

Executive Summary

The COVID-19 pandemic prompted the need for efficient and innovative solutions for contact tracing to prevent the spread of the virus. The DIMY protocol is a digital tracing protocol that will allow for infected individuals to register their diagnosis and create a chain-of-contact that can be further analyzed to monitor the spread of COVID-19. The Digital contact tracing application is used to break the chain of infection by identifying the infected person's close contact. This prototype framework, when deployed to mobile devices, will allow for real-time feedback of viral transmission through the population. The application leverages cryptographic techniques to generate a secure identification number that can be used for communication and tracing. This scalable solution will assist medical authorities to efficiently prevent widespread transmission of the virus.

Salient Features

1. Bloom Filter Manager:
    - Uses a circular buffer to rotate Daily Bloom Filters
    - Manages the creation and rotation of Query Bloom Filters
2. Did I Meet You Node:
    - Node generates an Ephemeral ID using the Elliptical Curve Diffie-Hellman Key agreement algorithm
    - Splits the Ephemeral ID into shares, distributes the (shares + hash) of Ephemeral ID over UDP using Shamirs Secret sharing
    - Receives shares from other nodes
    - Reconstructs shares if num_shares >= req_shares to reconstruct a secret
    - Hashes the Ephemeral ID to confirm the received ID hashes to the same hash that was received
    - Generates a private key from the Ephemeral ID using an HMAC Key Derivation Function
    - Generates a shared Encounter ID using the private key + Ephemeral ID
    - Encodes the Encounter ID into a Daily Bloom Filter (DBF)
    - Stores a set of Daily Bloom Filters as a Query Bloom Filter (QBF)
    - Uploads Query Bloom Filters to the DimyServer
    - Can trigger a Contact Bloom Filter (CBF)
3. DIMY Server:
    - CBFs are triggered by a signal interrupt
    - CBFs and QBFs are preceded by a Type Designator packet
    - The back end uses this packet-type designator to distinguish a CBF from a QBF
    - The server pads a CBF to be the same length as the QBF it compares to. Regards only 1-bits, with a 10% match indicating a 'close contact' with COVID-19

4. Thread Safe Socket:
    - Enables sequential processing of data with no loss due to competition on ports
    - Enables safe data transfers with timeout handling
5. Attacker:
    - Uses a Denial of Service Attack to disrupt the DimyServer
    - Binds to the same IP/Port as the Backend
    - Intercepts all CBFs/QBFs and returns a positive match in all cases
    - Forces nodes into isolation
    - Can be hardened against using a checksum to validate Server/Node interactions

Dependencies

```
tar -xvf assign.tar
pip install bitarray cryptography shamirs
```

DIMY Protocol Implementation

1. Ephemeral IDs are derived using the X22519 elliptic curve cryptography library. Shares are derived using the shamirs library, and broadcast.
2. Shared Encounter IDs are derived from a HMAC Key derivation function private key, with the Ephemeral ID serving as public key. The Encounter ID is then encoded into the Daily Bloom Filter, represented as a bit array.
3. Bloom Filters are either consolidated into CBFs or QBFs and sent using a 2-stage protocol. The first packet sent to the server identifies the incoming bloom filter as either a CBF or a QBF and instructs the server to treat the following bitarray accordingly.
4. The attacker capitalises on the fact that the communication protocol is neither encrypted, nor is the information validated through means such as checksum.

Screen Recording

https://youtu.be/1dXr_x9Zsiw?si=LUMN3MdIcgmD5ab9

Features Successfully Implemented

We believe all features have been successfully implemented

Unique Features

- HMAC Key Derivation Function to establish a shared Enc ID
- Communications protocol between node/server to manage Bloom Filters
- MiTM Attack

Design Trade-Offs

1. There is a possibility of a hash mismatch, less than 1%, using the shamirs library. Sporadically appears, is handled.
2. Using bitarray means using an “unterminated string”. It appears as “`001…010′`. This prevents it being received as part of a JSON object, which necessitates the 2-part communication between client and server.

Possible Improvements

1. Write a JSON encoder that works or successfully convert the bit array to a list of integers

Assignment Diary

Tasks	**Week 4**	**Week 5**	**Week 6**	**Week 7**	**Week 8**	**Week 9**	**Week 10**
Assignment Walkthrough and Research							
Task 1-3							
Task 4							
Task 5-6							
Task 7-11							
Report							

Team members	
Christopher Flinn	
Hardique Dasore	