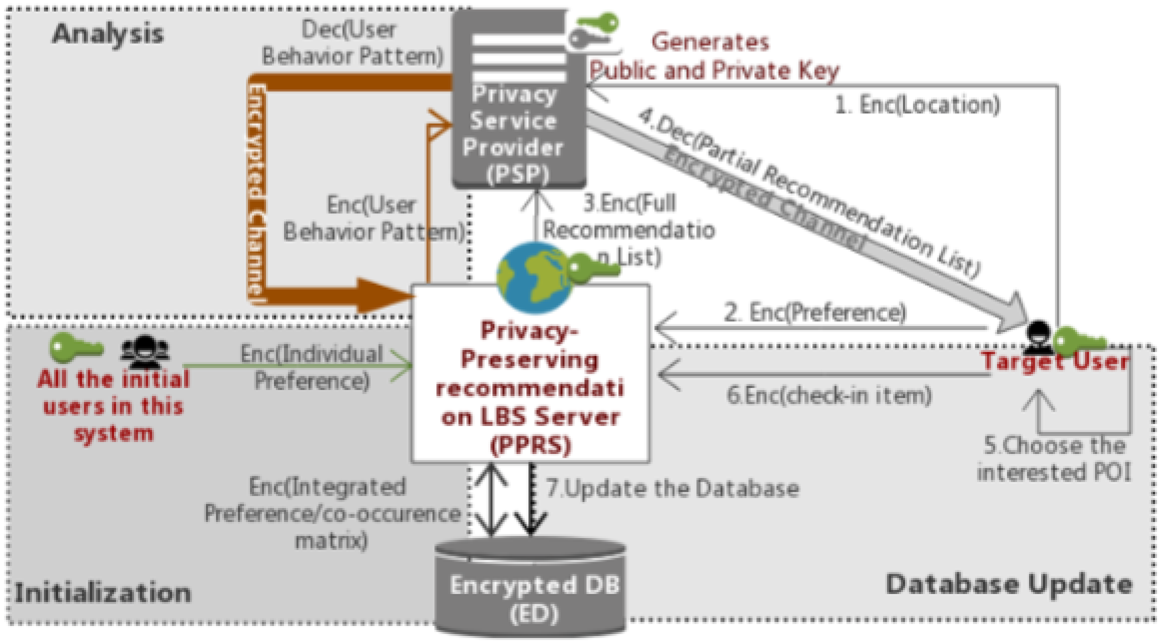
Privacy-Preserving Recommendation for Location-Based Services with Fully Homomorphic Encryption

Implementation of the protocol with the functions:

1. Compute aggregate information concerning user behavior patterns homomorphically while maintaining the privacy of individual users.
2. Encrypted database update.


Analysis
Dec(User Behavior Pattern)
Encrypted Channel
Privacy Service Provider (PSP)
Generates Public and Private Key
1. Enc(Location)
4.Dec(Partial Recommendation List)
Encrypted Channel
Enc(User Behavior Pattern)
3.Enc(Full Recommendation List)
2. Enc(Preference)
Privacy-Preserving recommendation on LBS Server (PPRS)
All the initial users in this system
Enc(Individual Preference)
Target User
6.Enc(check-in item)
5.Choose the interested POI
Enc(Integrated Preference/co-occurence matrix)
7.Update the Database
Encrypted DB (ED)
Initialization
Database Update

Prerequisites

- CentOS 7.3
- Linux Kernel 3.10.0
- glibc 2.17
- g++ 4.8.5
- cmake >=2.8
- HElib
- doxygen (If you generate a document)

Building

1. Build HElib, then you copy the HELib top directory to `/usr/local/src/HElib`.
2. Run following command to build the library.

```
$ mkdir build && cd build
$ cmake ..
$ make
```

- Generated files

File	Content
lbsr/lbsr_psp/liblbsr_psp.so	PSP library

File	Content
lbsr/lbsr_pprs/liblbsr_pprs.so	PPRS library
lbsr/lbsr_client/liblbsr_client.so	Client(Target User) library
stdsc/stdsc/libstdsc.so	stdsc library
demo/psp/psp	PSP demo app
demo/pprs/pprs	PPRS demo app
demo/client/client	Client(Target User) demo app

API Reference

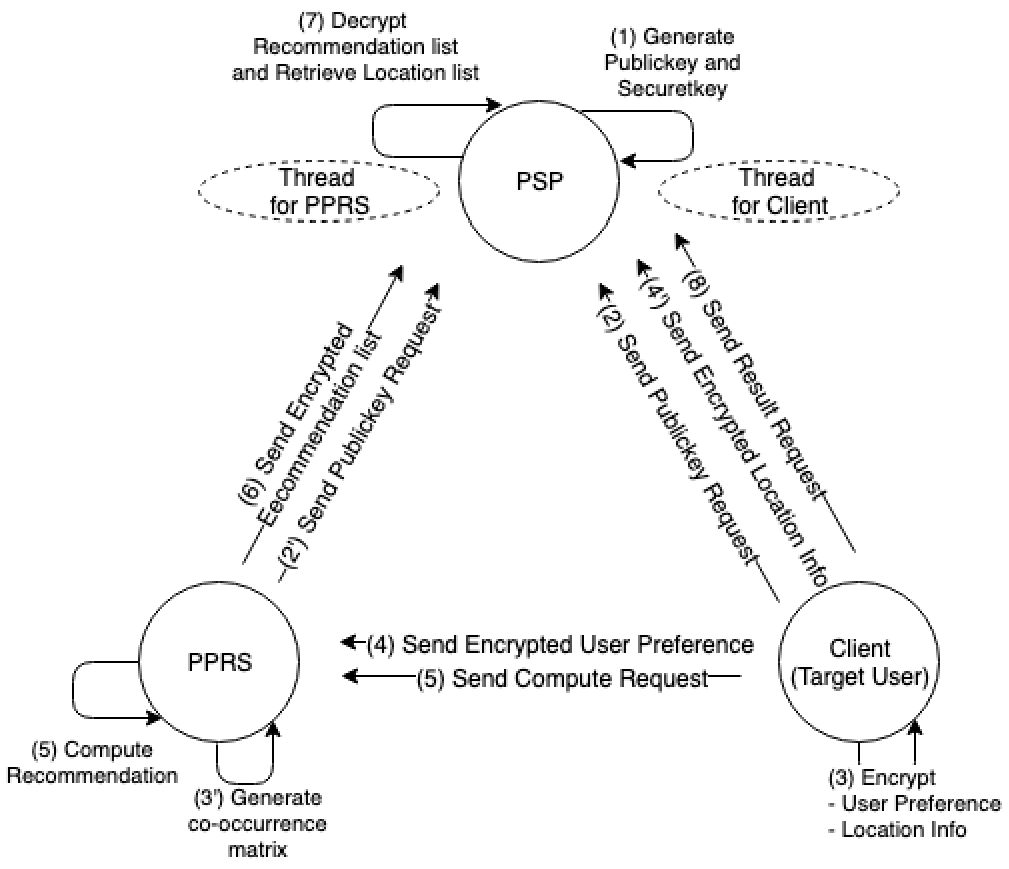
- Run following command to build the documentation.

```
$ cd doc && doxygen
```

- see `doc/html/index.html`

Demo

Demo app consists of three processes: PSP, PPRS and Client(Target User). These processes communicate as shown in the following figure.



(7) Decrypt Recommendation list and Retrieve Location list
(1) Generate Publickey and Securetkey
PSP
Thread for PPRS
Thread for Client
(6) Send Encrypted Eecommendation list
(2') Send Publickey Request
(2) Send Publickey Request
(4') Send Encrypted Location Info
(8) Send Result Request
PPRS
(4) Send Encrypted User Preference
(5) Send Compute Request
Client (Target User)
(5) Compute Recommendation
(3') Generate co-occurrence matrix
(3) Encrypt
- User Preference
- Location Info

PSP demo app

- Behavior
  - If the `-g` option is specified, PSP generates a Public Key file with the file name specified by `pubkey_filename` and a Secret Key file with file name specified by `seckey_filename`. (Fig: (1))

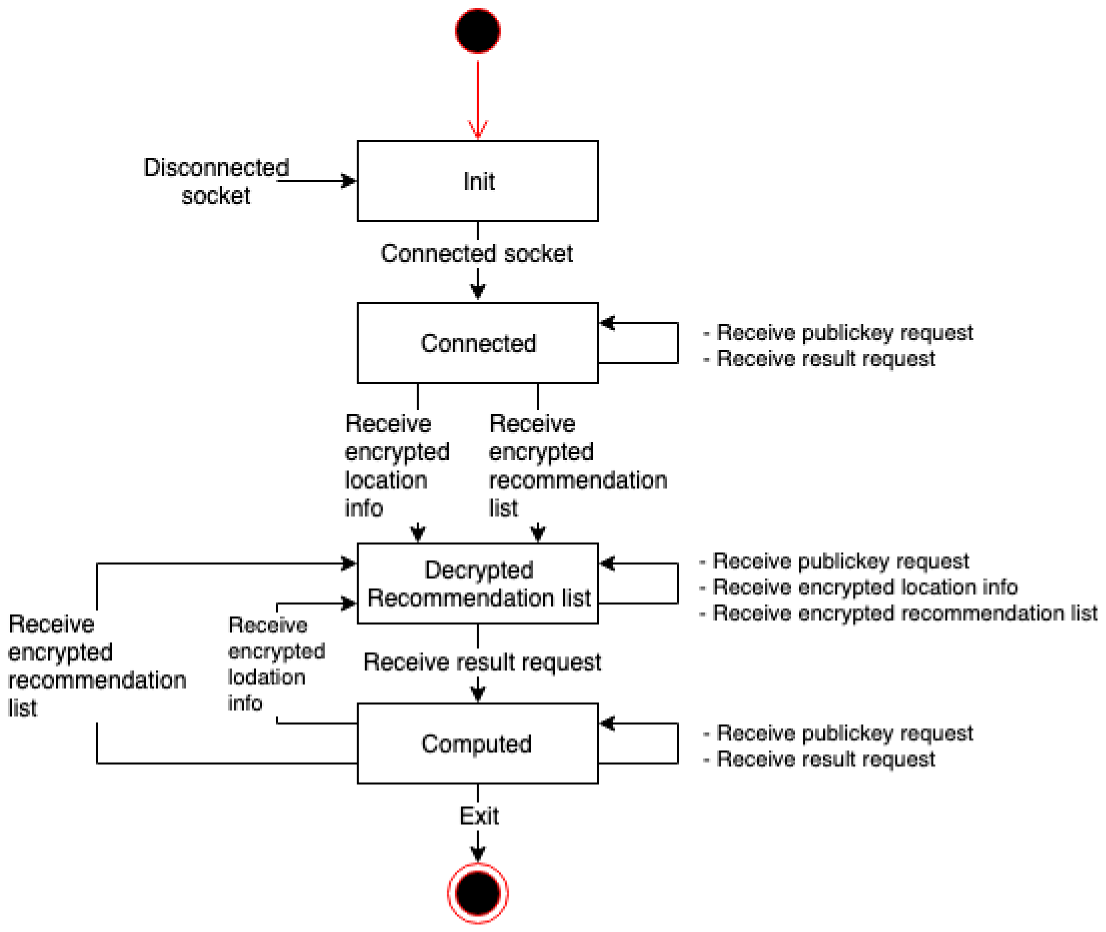
- PSP receives public key request from Client, then return the public key to Client. (Fig: (2))
- PSP receives encryped location info from Client. (Fig: (4'))
- PSP receives result request from Client, then decrypt recommendation list and retrieve location info and return the result to Clinet. (Fig: (8))

- Usage

```
Usage: ./psp [-p pubkey_filename] [-s seckey_filename] [-g]
```

  - -p pubkey_filename : file path of public key file (REQUIRED)
  - -s seckey_filename : file path of secret key file (REQUIRED)
  - -g : if this option is specified, it generates a public Key file and a secret Key file. (OPTINAL)

- State Transition Diagram

  - 

Disconnected socket
Init
Connected socket
Connected
- Receive publickey request
- Receive result request
Receive encrypted location info
Receive encrypted recommendation list
Decrypted Recommendation list
- Receive publickey request
- Receive encrypted location info
- Receive encrypted recommendation list
Receive encrypted recommendation list
Receive encrypted lodation info
Receive result request
Computed
- Receive publickey request
- Receive result request
Exit

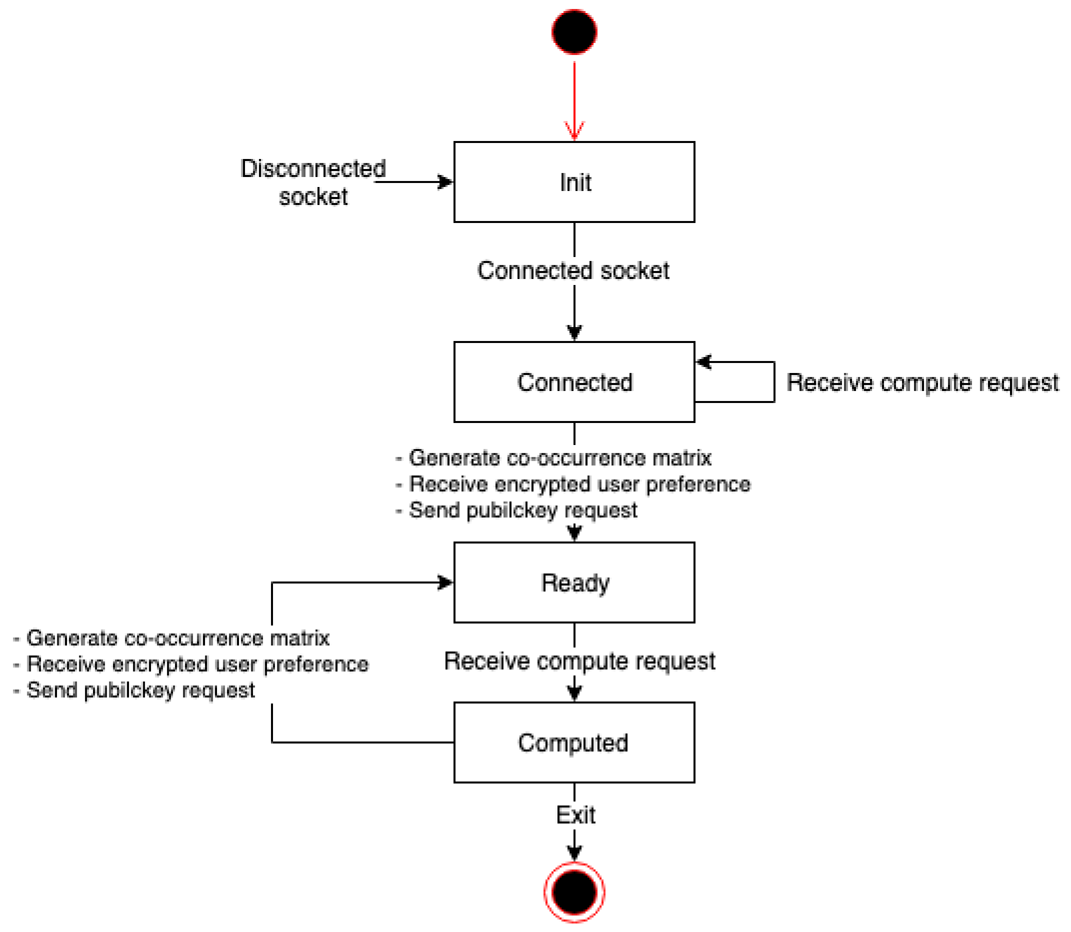
PPRS demo app

- Behavior
  - PPRS sends public key request to PSP, then receives the public key from PSP and save it named `pubkey.txt`. (Fig: (2'))
  - PPRS generates a co-occurrence matrix file named `enc_cooccurrence.txt`. (Fig: (3'))
  - PPRS receives encryped user preference from Client and save it named 'enc_preference.txt`. (Fig: (4))
  - PPRS receives compute request from Client, then compute recommendation and send the result to PSP. (Fig: (5)(6))
- Usage

```
Usage: ./pprs [-i input_filename]
```

- -i input_filename : file path of input data (REQUIRED)

- State Transition Diagram

  - 

Client demo app

- Behavior
  - Client sends public key request to PSP, then receives the public key from PSP and save it named `pubkey.txt`. (Fig: (2))
  - Client encrypts the user preference specified by `input_filename`, then Client sends encrypted data to PPRS. (Fig: (3)(4))
  - Client sends encrypted location info to PSP. (Fig: (4'))
  - Client sends result request to PSP, then receives the result from PSP and write it to stdout. (Fig: (8))
- Usage

```
Usage: ./querier [-i input_filename]
```

  - -i input_filename : file path of input data (REQUIRED)

License