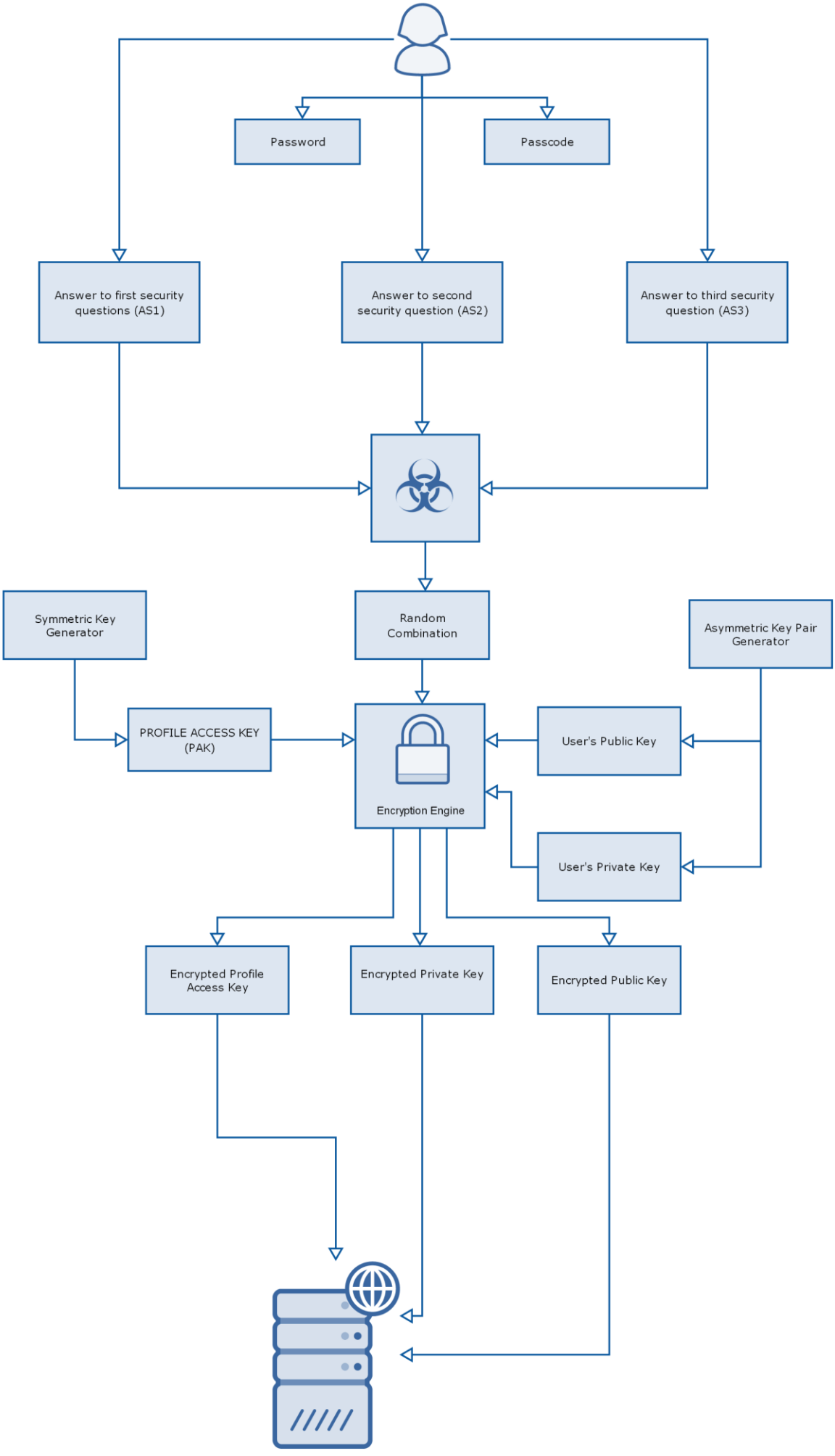
GENZO SECURITY WHITE PAPER

Table of Contents

1. Registration and client's/User's key pair generation
2. Profile encryption for storage on secured servers
3. Login with multi layer authentication
4. Session handling with mortal cookies
5. Transport Layer Security using TLS 1.1
6. Application layer security using end to end encryption : Server & Client, Client to Client
7. Encrypted user profile storage and token based sharing
8. Backend server key pair generation and storage

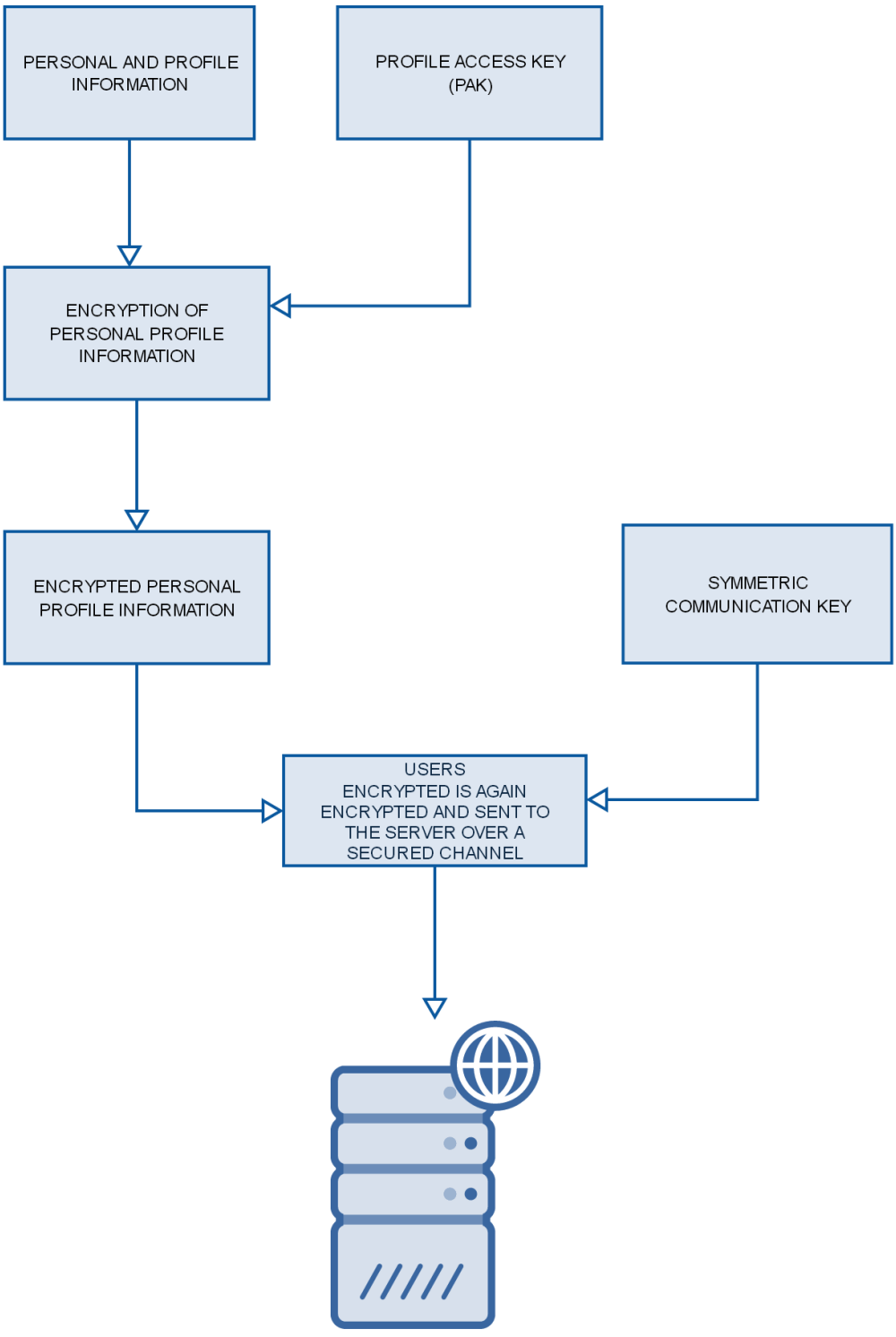
1. Registration and client's/User's key pair generation

- You can register by providing a unique username.
- You provide a password that must be atleast 8 characters long and must contain a upper case letter, a special character, a lower case letter and a number from 0-9.
- You also provide a 5 digit passcode that allows you to be able to make a fast login on a already used device. Passcode must not have all the same digits or a single digit cannot be repeated more than 2 times and the digits must not be in sequence.
- You have to answer 3 security questions. The answers to these questions are NOT saved anywhere.
- A private/public key pair is generated for the user. Generated public key is stored on the backend server and your private key is then encrypted and stored in the secured SL Cipher database on the device.
- Answers to the 3 questions are randomly used for encrypting your public/private key pair and encrypted key pair is stored on the server. This is for the purpose of regeneration of private/public key pair when you login from another device.
- Profile access key is also generated which is encrypted with your public key and saved on the server.


Password
Passcode
Answer to first security questions (AS1)
Answer to second security question (AS2)
Answer to third security question (AS3)
Symmetric Key Generator
Random Combination
Asymmetric Key Pair Generator
PROFILE ACCESS KEY (PAK)
User's Public Key
Encryption Engine
User's Private Key
Encrypted Profile Access Key
Encrypted Private Key
Encrypted Public Key

2. Profile encryption for storage on secured servers

Your personal information in the profile is encrypted with your profile access key (PAK) and saved in the backend server. This profile information can only be decrypted with your profile access key. This ensures that no one else except you and users authorized by you can access your personal information.



3. Login with multi layer authentication

You can login to the app in two scenarios as detailed below:

a) Logging in the app for the first time on a device :

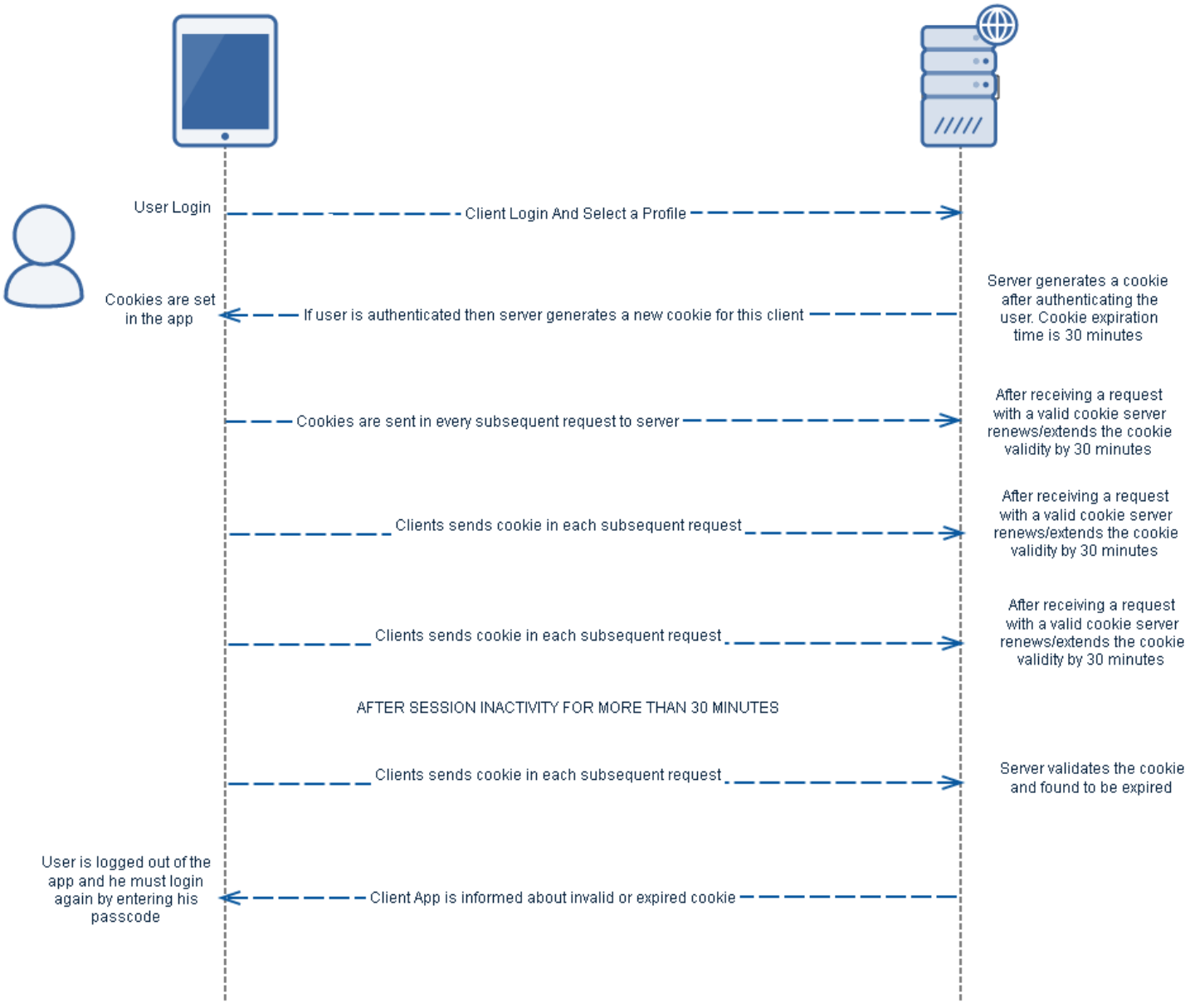
- You have to provide your password for authentication and once your password is validated then you have to provide answers to 3 security questions.
- These answers are used as for re-generating your key pair for this device. Random combinations of the answers are made to decrypt the saved public key and find a generated public key that matches your public key.
- If app is not able to generate any matching public key using the answers provided the you are not allowed to login on a new device.

b) Logged in earlier on the device and already added to the list of users:

- You can perform a fast login after providing your passcode however if you enter the wrong passcode for 3 times in a sequence then you will be removed from the device and keys are also removed from this device.
- You must login again following the process defined in 3a.

4. Session handling with mortal cookies

- Your successful login in the app is succeeded by creation of a cookie based session between client app and server.
- This cookie is generated on the server and passed in response as a result of successful authentication.
- Cookie helps server to identify and authorize the client app in multiple requests and client app need not send logged in users identity.
- Cookie is initially created for a time period of 30 minutes and its expiry is increased by 30 minutes with every successful server request.
- In case there is no activity from the client for 30 minutes then the cookie will be expired and needs to be regenerated.
- Each time app goes in background the cookies are invalidated and will be re created when app comes in the foreground and user enters his passcode.



5. Transport Layer Security using TLS 1.1

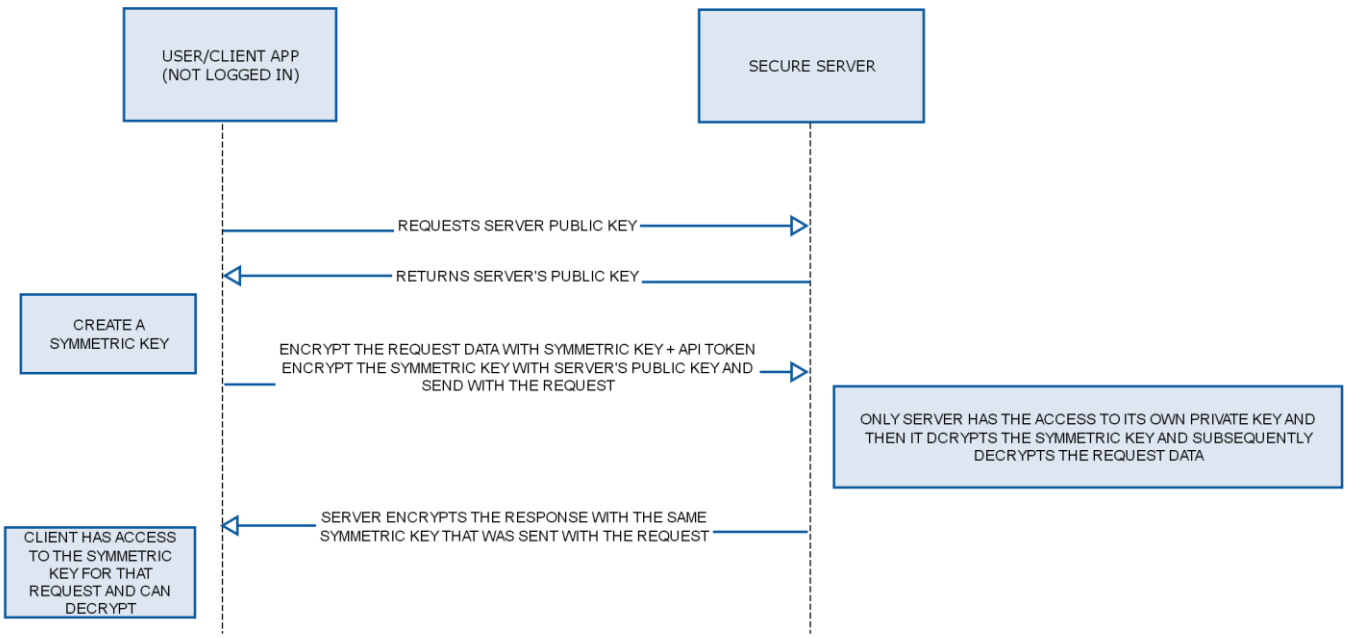
All the data exchange between the client device and the backed cloud server machine happens over SSL connection using TLS 1.1 protocol. This ensures protection of application data from unauthorized disclosure and modification while it is transmitted. We cannot use TLS1.2 because it is not supported for Android version less than 5.0 and Google glass does have Android version 4.4.4. However as soon as we have an update on Google glass we will update the system to TLS1.2.

6. Application layer security using end to end encryption

a) Communication between Server & Client:

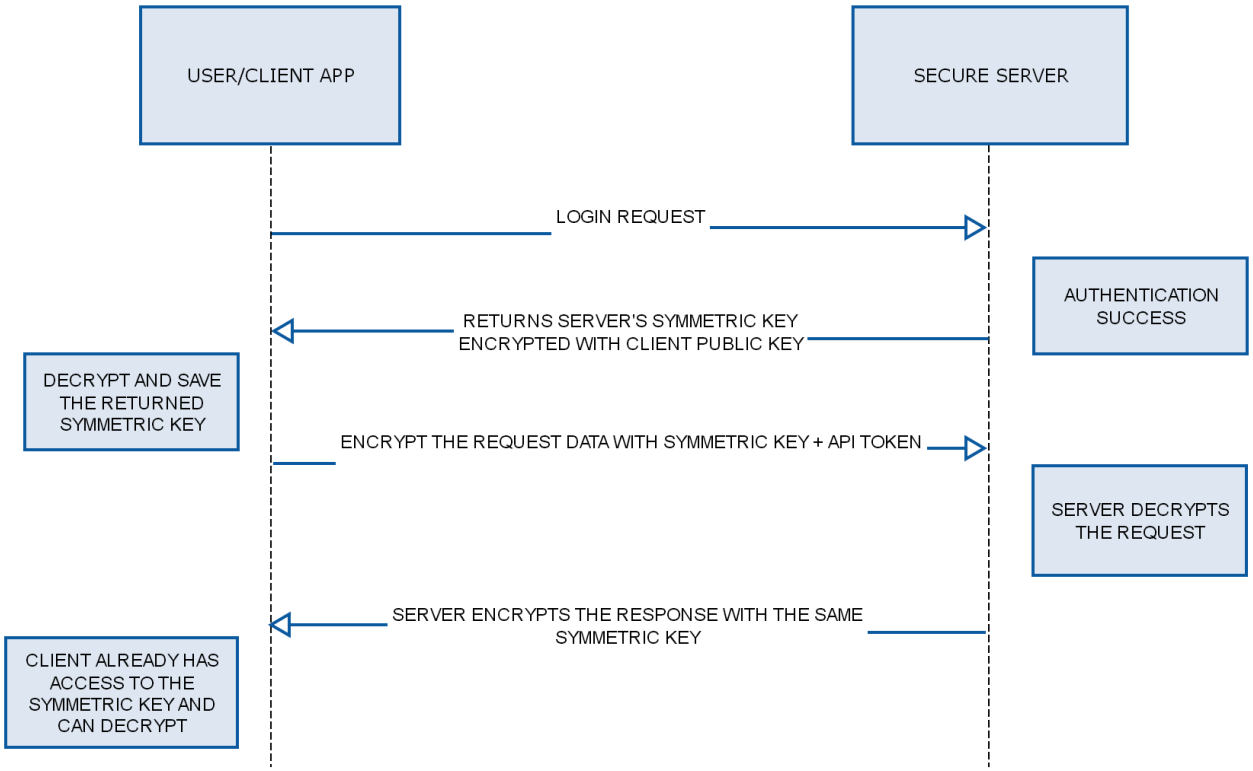
- **Non Logged in use from Mobile App to Server :**
  At start Client app fetches the server's public key and creates a onetime symmetric key for each request.
  Encrypt all the data that is sent from client app to server using the key that is created using a symmetric key + api token.
  Encrypt the one time symmetric key with server public key and send along with the request.
  Only server has the access to its private key and is able to decrypt the symmetric key and hence the request.

- **Non logged in use from Server to Mobile App :**
  In the previous type of communication server has received a symmetric key for that request and all the data that is sent from server to the client app is encrypted using that symmetric key. As only the sender app has the access to that symmetric key and therefore is able to decrypt the response.



- **Logged in use from Mobile App to Server :**
  At the time of login, server will have a symmetric key and it will encrypt it with client public key and send it back to app.
  Now the client can decrypt this encrypted symmetric key using his own private key.
  Encrypt all the data that is sent from client app to server using the key that is created using this symmetric key + API token.

- **Logged in use from Server to Mobile App:**
  At the time of login, server will have a symmetric key and it will encrypt it with client's public key and send it back to app.
  Now the client can decrypt this encrypted symmetric key using his own private key.
  As the server and client both have access to the same symmetric key therefore they encrypt all the data with this symmetric key before exchange.



b) Communication between client app & other client app

In many scenarios there is direct communication between 2 mobile apps. In these cases the client apps use middleware websocket server.

Web Socket Security :

- Websocket are implemented as secure websockets with TLS 1.1 which adds a transport layer data encryption.

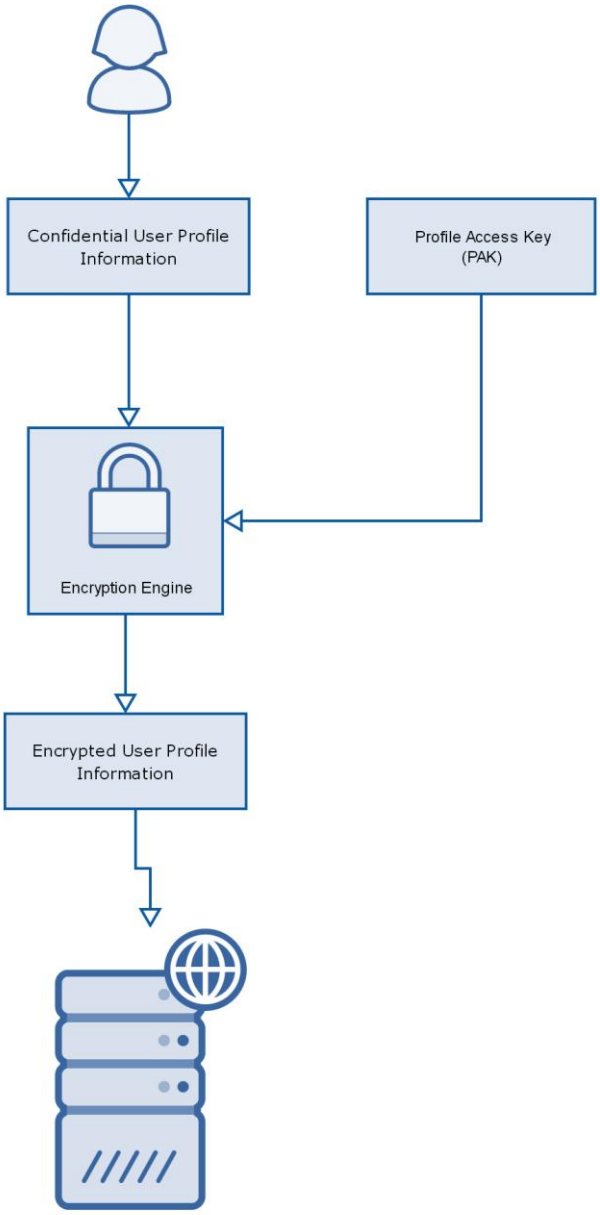
Web Sockets Authentication mechanism:

- There is a authentication mechanism that the client app use when it first creates a socket connection.
- On login, the server sends back an websocket ticket and the client app sends back this ticket when initiating a websocket connection.

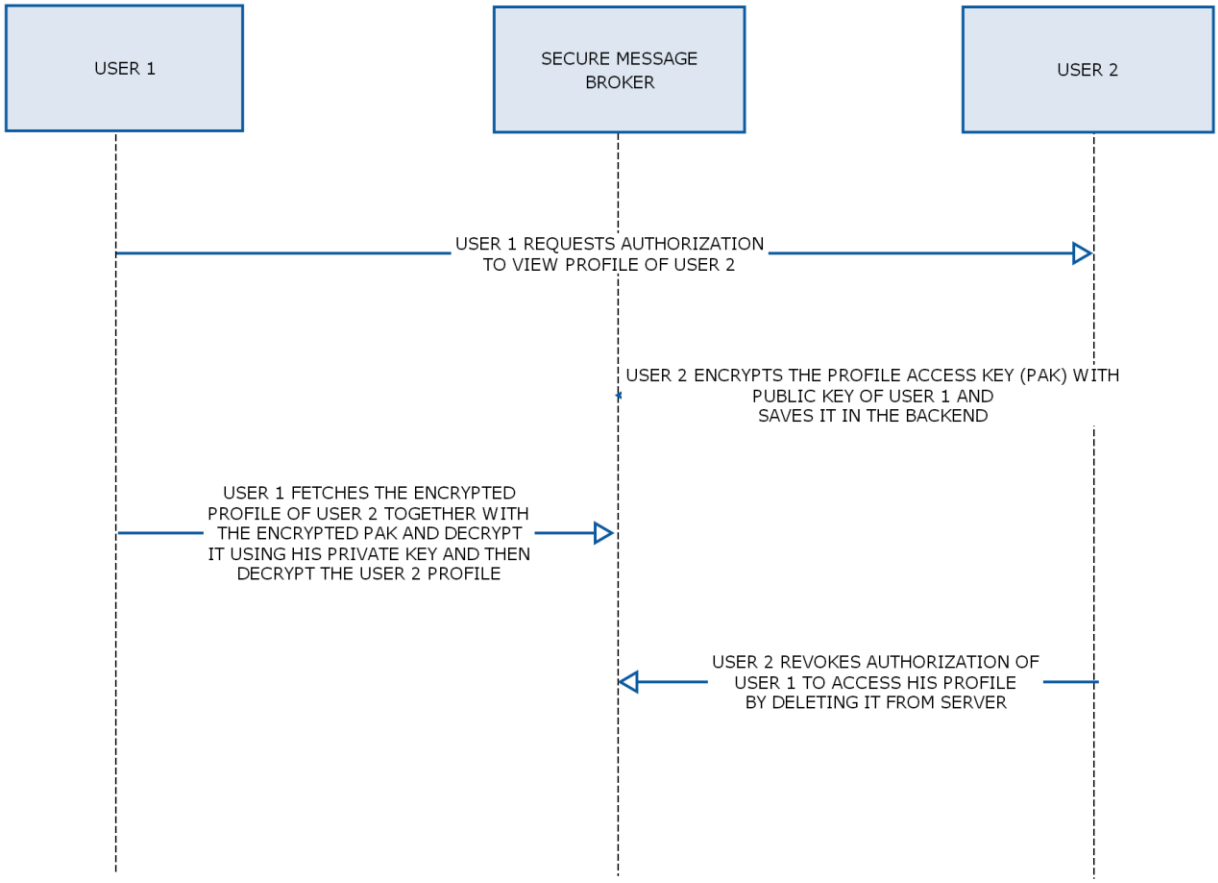
7. Encrypted user profile storage and time bound token based sharing

- As mentioned earlier section of the document, your personal details are encrypted using your profile access key (PAK) and can only be decrypted to plain text by using the PAK and PAK is encrypted with your public key and can be decrypted with your private key which is accessible only to the device you are logged in therefore no other user or even server can access your personal information without your consent.

- However, In certain scenarios you would want other users to be able to review your personal or medical data and these scenarios will be handled by sharing your profile access key (PAK) with that user. PAK will not be shared in plain text but it will be first encrypted with the users public key and only he can decrypt it using his private key. This encrypted PAK is stored in the storage on the secured server.



- When the other user wants to access your profile he can fetch the encrypted PAK and then decrypt it and use it to decrypt your profile.



8. Backend server key pair generation and storage

- The server key pair is generated every time server instance starts and therefore there is no need to save it anywhere on the file system.
- Server private key is thus only available to the running instance of the server and not available to any human.
- At the same time a symmetric key for web services communication is generated on the server.

9. Encryption Methods

a) RSA : We have used RSA encryption algorithm wherever asymmetric encryption is used. RSA is an algorithm used by modern computers to encrypt and decrypt messages. It is an asymmetric cryptographic algorithm. Asymmetric means that there are two different keys. This is also called public key cryptography, because one of them can be given to everyone. The other key must be kept private.
https://simple.wikipedia.org/wiki/RSA_(algorithm)

b) AES : We have used AES encryption algorithm wherever symmetric encryption is used.
https://en.wikipedia.org/wiki/Advanced_Encryption_Standard

References :

https://devcenter.heroku.com/articles/websocket-security
https://www.cs.rit.edu/~spr/PUBL/ehr11.pdf
http://nacl.cr.yp.to/
https://en.wikipedia.org/wiki/Diffie%E2%80%93Hellman_key_exchange