Message Encryption Algorithm for UniPool

Author - Rohan Datta

Summary

This is the theory behind the algorithm that is used to encrypt messages between users of UniPool, in an end-to-end fashion. The implementation uses Diffie-Hellman algorithm [1] *to generate a key and encrypts the message sent by one user to another, using that key. In this way, messages are obscured from external users* ***and*** *database administrator i.e. the app development team.*

Implementation

Suppose user Alice has to send a message to user Bob.

Step 1: Alice picks two prime numbers **g** and **p**, and sends them to Bob.
Step 2: Alice picks a random *secret* number, a, and computes $A = g^a \, mod \, p$.
Step 3: Bob comes up with b and computes $B = g^b \, mod \, p$.
Step 4: Alice sends A to Bob, and Bob sends B to Alice.
Step 5: Alice computes $B^a \, mod \, p$, and Bob computes $A^b \, mod \, p$.
Step 6: Now, $B^a \, mod \, p = A^b \, mod \, p = g^{ab} \, mod \, p = g^{ba} \, mod \, p$; lets call this value K.
Step 7: The message is encrypted by using K as its key, known *only* to the two users.
Step 8: The message is then sent to Bob, who decrypts it using K.

So, in this way, end-to-end encryption is maintained and the users' privacy is preserved.

[1] Diffie, Hellman, 1976