International Journal of Scientific Research and Management (IJSRM)
||Volume||12||Issue||01||Pages||1016-1017||2024||
Website: https://ijsrm.net ISSN (e): 2321-3418
DOI: 10.18535/ijsrm/v12i01.ec03

Implementation of Anonymous social network

Nripesh Trivedi

Department of Mathematical Sciences, Indian Institute of Technology, Varanasi

Abstract

This paper describes a implementation of a server-client model that could be used as anonymous social network. It could be used for n clients. The anonymity of client is ensured on server side. The scheme implemented in this paper was proposed by Dr. Hamza Harkous and Dr. Rameez Rahman. The scheme was proposed for trusted users where users trust each other.

Scheme and Implementation

The scheme [1] is as follows -

1. Users in a group have a single group key.
2. The server side has generated a pair of public and private keys.
3. The public key is with the clients while the private key is with the server.
4. Users encrypt messages first using public key and then the group key. The server gets the encrypted message. It waits till it has received n number of messages before broadcasting them to the client.
5. After receiving n messages, the server broadcasts these messages to the clients. The clients decrypt these messages using the group key and randomly shuffle these messages.
6. The server keeps on waiting till it has received a particular set of messages. Then the server decrypts these messages using the private key.
7. At the last step, the messages are broadcasted to the group.

In this scheme, the assumption is that users trust each other (that's why a common group key). It could be seen that anonymity is ensured on the server side by using encryption and decryption schemes.

The implementation of this scheme is described below [2]-

The above scheme was implemented as broadcast server client application. The language used for implementation was java. Client could connect to the server once they know the name of the server. To connect server to the client, NIO library [3] in java was used. RSA keys [4] were used for public and private key generation and PBEWithMD5AndDES [5] for group key generation. The public key is provided to the client as a constant as the generation of public and private keys takes place on server side. The messages were first encoded using a group key and then the public key. All the basic message exchange between the server and client happens as byte code. For conversion from string to bytes and bytes to string, ISO-8859-1 [6] was used. The encrypted messages were sent to server. The server waits till it has received a certain number of messages. (Here 3 as in the code). After receiving n messages (here 3), the messages were sent to all the clients within the group. The clients receive the messages and shuffle them and send them back to server after decrypting using group key. The clients send the decrypted messages (by group key) to the server. The server decrypts the messages using the private key and send them back to the clients. Since the messages were shuffled by clients, anonymity is ensured on server side.

References

1. Trivedi, N. (2014). *Scheme for Anonymous social networks*. EPFL.
2. Trivedi, N. Report on 12 week internship/training undertaken at EPFL (Ecole polytechnique fédérale de Lausanne
3. Source - https://docs.oracle.com/javase/8/docs/api/java/nio/package-summary.html
4. Source - https://docs.oracle.com/cd/E21764_01/apirefs.1111/e10668/oracle/security/crypto/core/RSA.html
5. Source – https://www.tabnine.com/code/java/classes/org.jasypt.encryption.pbe.StandardPBEStringEncryptor
6. Source -https://docs.oracle.com/javase/8/docs/technotes/guides/intl/encoding.doc.html