Praneeth Manubolu
MSCS 630
Android Jabber Application
Project Milestone

Abstract –

This paper explains the implementation of Android based chat application, methodology and technologies. To secure the information, AES has been implemented where the AES keys are initially encrypted using RSA. The server used for connecting users only shares the encrypted information there by making it impossible to decrypt even by the provider. XML is used for exchanging information where only sensitive data like actual message is encrypted. For verifying or validation SHA-2(256 bits) has been used.

Although this project is aimed to implement a secure chat application, the entire system (client & server) is designed to look and work like commercial chat application like WhatsApp, messenger etc.

Introduction –

In the present world, people communicate with others through messages more, than by voice or video. The challenge lies in not only securely transmitting them but to implement an efficient encryption system as millions of messages fly around the globe at any given second. It comes to no surprise that we already live in the most secured era where an attack on a well encrypted data takes more time than the formation of universe. The project takes inspiration from this, implementing latest encryption standards and techniques.

Background –

One of the highly used crypto system for instant messaging is OTR (off-the-record messaging) which is a combination of AES symmetric key algorithm of 128 bits key length, the Diffie–Hellman key exchange with 1536 bits group size, and the SHA-1 hash function. This project uses a similar approach to OTR. E2EE (End-to-end encryption) is another kind of crypto system where the even the provider cannot decrypt the data. This is a true implementation of security where even the provider cannot decrypt messages which makes government meddling or cyber threats impossible. E2EE has been implemented in WhatsApp and has been gaining popularity extensively.

Methodology –

To create a user-friendly application, where the user can also see statistics of the encryption and decryption. Using OTR as reference, the application is programmed to generate RSA and AES keys on initializing, on the client app. The RSA public key and AES encrypted keys are shared during the handshake process between users. It was required to understand how AES works as encrypting messages was causing issues due to varied lengths. Padding was required to overcome this problem.

Experiments –

Initially two applications were created one for just sending/receiving message and other to check encryption and decryption. It could encrypt and decrypt text file successfully using AES keys of size 128, 196 and 256. Also, used only RSA for encrypting/decrypting messages to understand why RSA alone cannot be used. From experimenting RSA showed sever performance issues where it took 17 seconds to encrypt a piece of data, the same data took 294 milliseconds if AES was used.

Discussion –

Experiments proved that AES is much faster than RSA. An important observation is that, while comparing encryption/decryption with AES and RSA it was seen that AES takes more time to encrypt than to decrypt with a ratio of 6:1(E:D - where E is time taken for encryption and D is the time taken for decryption). In case of RSA, Decrypting took more time than encrypting the same data with a ratio of 1:8(E:D).

Conclusion –

It is proven that AES is much faster than RSA and can be used for securing any type of data. A major problem still existing is the sharing of the encryption keys. RSA still plays a dominant role in this field due its unique implementation of using public and private keys. Where private key is kept secret, and is used for decryption. Using these two encryption systems together increases security to the highest level. SHA-2 is used in aid of the above system to verify the data which acts as a digital signature.

References –

1. Jiang Bian, Remzi Seker and Umit Topaloglu “Off-the-Record Instant Messaging for Group Conversation”. *Information Reuse and Integration*. IEEE International Conference on Information Reuse and Integration 2007.

2. Alessandro Barenghi, Michele Beretta, Alessandro Di Federico, Gerardo Pelosi “An End-to-End Encrypted Online Social Network”. 2014 IEEE Intl Conf on High *Performance Computing and Communications*, 2014 IEEE 6th Intl Symp on *Cyberspace Safety and Security*, 2014 IEEE 11th Intl Conf on *Embedded Software and Syst* (HPCC, CSS, ICESS).