IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Encryption And Decryption End-To-End Encryption, Android Application

Koppala Yamuna
Dept. of computer science and Engineering
Lovely Professional University, Punjab, India

Bangla Sriman
Dept. of computer science and Engineering
Lovely Professional University, Punjab, India

Chintha Lakshminarayana
Dept. of computer science and Engineering
Lovely Professional University, Punjab, India

Maddukuri Manideep
Dept. of computer science and Engineering
Lovely Professional University, Punjab, India.

Introduction:

In this project we will build an encrypted chat messaging app for Android. In this we will be using end-to-end encryption. End-to-end encrypted messaging guarantees that only messages required by two parties can be read by the user inside that particular talk. To achieve this messages that are sent are encrypted and can only decrypted by the intended recipient (end user) before learning user device.

Security is essential to the end-to-end encryption provides. We have seen many cases of malicious hackers secretly accessing vast some of private data and abusing the technologies to harm or hurt individuals with their stolen information. Data security has been much more critical since the introduction of end-to-end encryption, since 2016.

The primary advantage of end-to-end encryption is it limitations of anyone but the recipients transmitted data. It is as is you headed it in a hack eye that was physically difficult to open when you sent a host, resistant to any sledge hammer, saw, lock pitch, and to un expect by the addressee. The confidentiality and integrity of your correspondence is ensured by end-to-end encryption.

In the physical world, building an invisible box is not even possible, but it is in the world of knowledge. New encryption algorithms are actively being and enhancing the reliability of old ones.

Another gain is that end-to-end encrypted communications are undecrypted by someone other than the recipient, the code will not be modified by anyone. Modern encryption techniques function in such a manner that the message gets you distorted on decryption is anyone updates the encrypted data, rendering the issue immediately there is no way to make predictable improvements to as encrypted message that is replacing the text is difficult.

That guarantees the credibility of your introduction. When you receive a decrypted message successfully, you should be confident that it is the same message sent to you and that is has not been tampered with in transit.

Relative Framework:

In our project in order to build our End to End Encryption Chat box, we decided to use some of the most popular and trending frameworks which guarantees the stability and scalability of the product.

We started splitting the frameworks we use based on user interactivity and core functionality. In order to satisfy the

user interactivity we decided to build an amazing mobile app which is a more handy and best way of

interacting with the project we are doing. So, in order to build the mobile, we decided to use Flutter. Flutter is an amazing and powerful tool to build native apps for both Android & iOS. Flutter comes up with a whole of tools, APIs and services which makes us the best choice in picking up for our project and to move forward.

On our way to the next step we started thinking about our core model of Encryption and decided to embed in the mobile app itself make it to render on client side representing Client-Side Rendering

Database:

In order to store all our chat data, signature keys and all we decided to use Firebase as our backend database. Firebase is a cloud-based database service which comes up with a free tier for small scale products and is easily scalable whenever there is a necessity. Firebase gives us an up and running time of 99.95% with zero database maintenance.

Working on architecture:

Our core idea is to provide a high level of security to our users who chat on our app.

We decided to make the product with such privacy where the sender and the receiver can see/read or be able to understand the message. Even the chat info on the database is readable and understandable.

We decided to use RSA providing the secured public key and the private key to both users. Public keys can be accessed by anyone in the database, even the database admin but the private key can be Accessed by one the authenticated sender &receives only.

Public Key (Visible for everyone)

Private Key

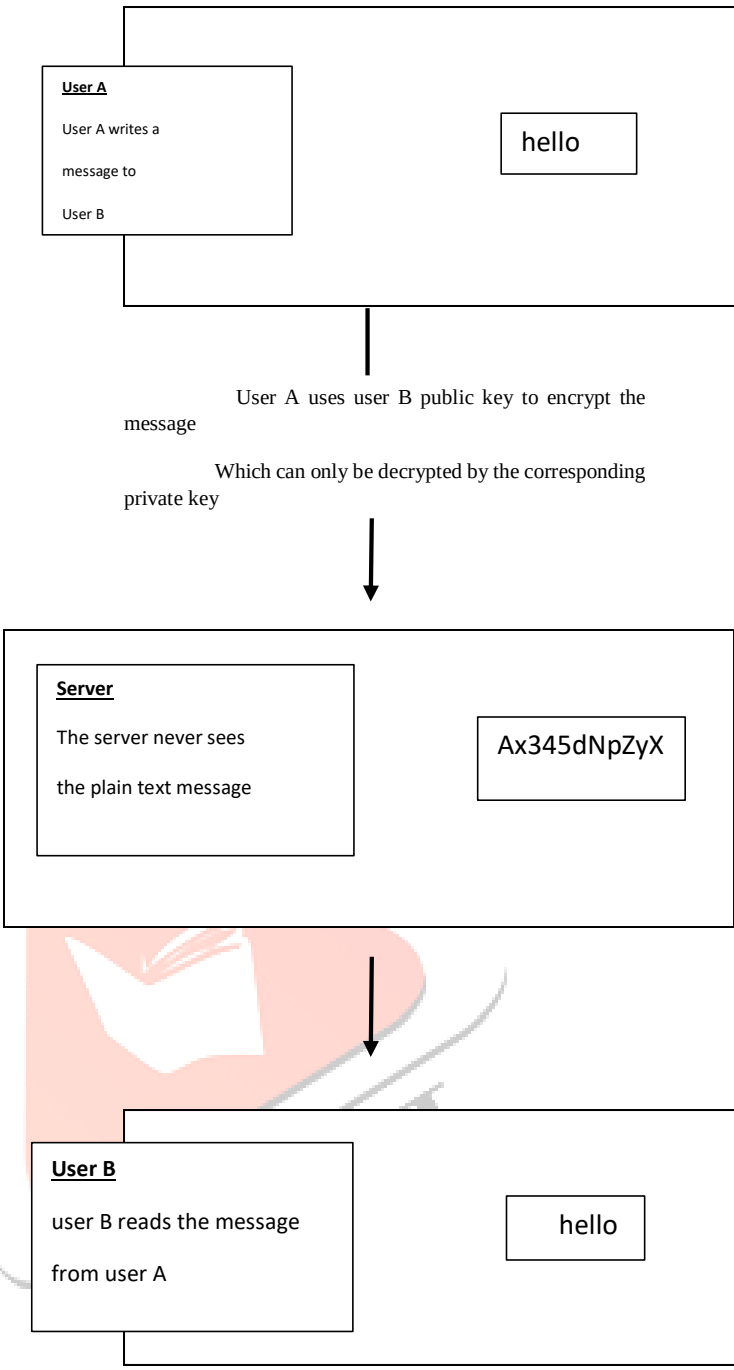
(Have access for only sender & receiver) - Securely stored in Database

(Sender) (Receiver)

Authenticated-->RSA Encrypt with --> Database --> RSA Decrypt with-->Authenticate

Sender User Private & Public key (Encrypted Private & Public Key Receiver message will be stored in database)

Result:



The encrypted message is being transmitted over the public internet. It will pass through many servers along the way, including those belonging to the email server which they are using and to their providers of internet services. Although these companies can attempt to read the message or they may exchange with third parties. It is difficult for them to translate the text which is ciphered into plain text which is readable.

Observation:

In this end-to-end encryption we are using RSA (Rivest- shamiy -Adleman) Algorithm is used which is also known as asymmetric encryption. Messages are encrypted under RSA encryption using a Secret key called a public key which can be freely exchanged Because of certain distinct mathematical Properties of the RSA Algorithm ,it can only be decrypted with another key known as the private key, after a message has been encrypted using the public key each RSA user

has a key pair consisting of their public and Private keys. The private key needs to be kept hidden, as the as the name Suggests.

public key encryption mechanism vary from symmetric key encryption where the some private key is used for both the encryption and decryption process. These Variations make encrption of public key such as RSA valuable for interacting in cases where there was little possibility to distribute keys Securely beforehand.

Symmetric key algorithms have their own uses, such as encrypting date for personal use of exchanging private keys where there are protected networks.

Conclusion:

To make things more efficient, your file will usually be encrypted with a Symmetric key algorithm, and then the Symmetric key will be encrypted with RSA encryption. Under This method, the symmetric key Can only be decryptable by a person that has access to the RSA private key.

The initial file can't be decrypted without being able to access the Symmetric key without taking too long or using too much processing resources, this approach Can be used to key messages and files secure, which keeps your data Safe from hacks, date private and its good for democracy.

End-to-end encryption is the technological backbone of our vision for a more private and Secure internet.

References:

https://medium.com/swlh/encrypted-messaging-app-android-c57f14a180cb#:~:text=End%2Dto%2Dend%20encrypted%20messaging%20means%20that%20the%20users%20within,recipient%20(end%2Duser).
https://www.comparitech.com/blog/information-security/rsa-encryption/
https://simple.wikipedia.org/wiki/RSA_algorithm#:~:text=RSA%20(Rivest%E2%80%93Shamir%E2%80%93Adleman,can%20be%20given%20to%20anyone.
https://www.geeksforgeeks.org/rsa-algorithm-cryptography/
https://getstream.io/blog/most-secure-messaging-apps/