iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 8 **Issue:** VII **Month of publication:** July 2020

DOI: https://doi.org/10.22214/ijraset.2020.30352

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue VII July 2020- Available at www.ijraset.com

A Detailed Study on Android Chat Applications and XMPP Protocol

Suganthi Sivakumar[1], Harshini R[2], Bhargavi P[3]
[1, 2]*Computer Science Department, Visvesvaraya Technological University*

Abstract: ***Today with the vast technological advancements in the various means of communication, instant messaging applications play a very crucial role in the transmission of messages from one user to another who will be residing at different locations at a minimal cost or with the support of the internet facility. With the advancements in technology comes the need to secure applications from malicious attackers who try to steal user credentials or may also cause passive attacks on the network. The chat application uses the XMPP server for the secure transmission of messages between the users. The defense organizations need a more secure chat application for communicating in a highly confidential manner. Therefore we try to propose a smart and secure chat application for the defense organization. Many existing Android chat applications are compared to analyze their features, security principles, and flaws to bring forward an innovative and secure android chat application.***
Keywords: Chat, end-to-end security, attacks, messages, encryption.

I. INTRODUCTION

There has been a massive need to develop free modes of communication since the Short Message Services or SMS are not free of cost. Therefore the instant messaging applications came into existence to make human life much easier. These applications are online chat systems that allow their users to communicate with each other over the internet facility. These instant message applications not only support text transmission but also facilitate the transmission multimedia files such as PDFs, images, audios, videos, contacts, and live location, etc. The instant messengers also have many additional features such as audio/video calls, audio/video conferences, etc. which make them more human-friendly. However, security has been one of the most important concerns that still need to be improved as there are malicious attackers who use these social media to steal user data or disrupt the services of the applications. Some of the attacks are described below as follows:

1) *Untrusted APK's:* An android application package is a file format package used for distributing and installing the mobile applications. Intruders send malicious applications to the users to download. When such APK's are downloaded it allows the intruders to gain more control over the mobile device.
2) *Spying:* This attack allows the application to watch over the mobile device and sends information to the attackers who wish to hack the device.
3) *Phishing (Email):* Attackers often mask themselves as trusted entities and send the users emails, to extract information such as login credentials, account numbers, etc. This may lead the users to get redirected to other unnecessary websites that may trade-off the user details. Spam emails may sneak information from the users.
4) *SMS:* Messages sent to the users stating some lottery or winning a certain amount are malicious and when clicked on the link it redirects to irrelevant websites which result in leaking of subtle information from the users.
5) *Rooting:* Rooting is done to increase the speed, efficiency, and performance of the android operating device. But this may not be suggested by the android officials as it may lead the attackers to easily gain the user's information and access the mobile device.
6) *App Sandboxing Issues:* Sandboxing is the process in which the application is been tested with a constrained resource environment that may be against malicious attacks and threats. Compromising this sandbox may lead to the attack of the device by various malicious software.

Even though instant messengers such as WhatsApp, Viber, Threema, and signal provide end-to-end security still there are some flaws in their security models. Moreover, defense organizations don't rely on third-party applications and their protocols. Hence there is a need for a more secure chat application that can be used by the defense organizations for secure communications. Chat applications can be created on different operating systems like ios, android. We mainly focus on the creation of an android chat application for the defense organization. Android is an operating system that uses a modified version of the Linux kernel which is mainly designed for smartphones. There are four main components in android namely, Activities, services, content providers, and broadcast receivers. Activities make use of an important tool called the intents to move from one activity to another activity.

The main programming languages used for android application development in java and kotlin. The android architecture includes four

layers. They are the system applications, Application framework, (Libraries, android runtime) hardware abstraction layer, and the Linux kernel. Each layer is responsible for its own set of operations which collectively contribute to the successful implementation of an android application.

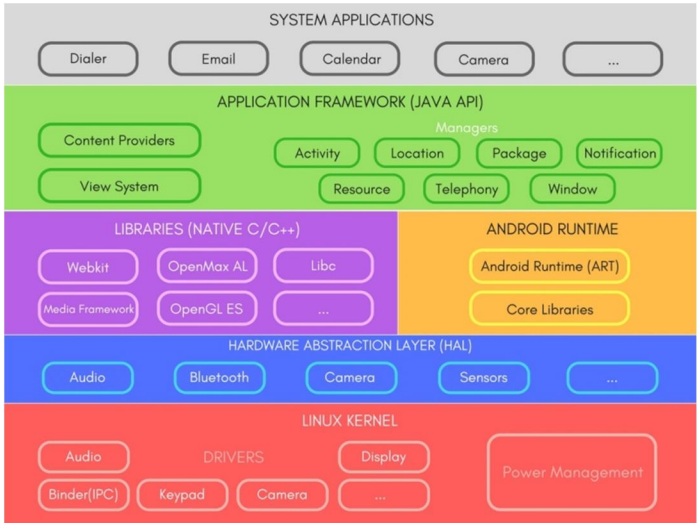


Fig.1 Android Architecture

The applications are susceptible to a large number of malicious attacks. Some of the major security goals include confidentiality, integrity, authenticity, reliability, authorization, and availability. Hence there is a need to secure the transmission of user data and a need to protect the user credentials from the attackers. Hence many chat applications such as WhatsApp provide the end-to-end security to make sure the messages being transmitted are encrypted and can be decrypted only by the intended receiver. The defense application also needs a more secure mode of communication and hence we use encryption techniques such as AES to encrypt the data.

II. LITERATURE SURVEY

A. Forensic Analysis of Encrypted Instant Messaging Applications on Android

The market of smartphones is growing day by day due to the amount of use of smartphones using android as its operating system. Due to the increase in malicious activities, security and privacy are the top concerns when it comes to protecting one's data, thereby resulting in end-to-end encryption. This paper analyses a few of the most common encrypted Instant Messaging (IM) applications such as WeChat, Telegram, Viber, WhatsApp. The main aim of this paper was to analyze the database files, data storage locations, and to check whether they are encrypted. To gain privileges on Android devices, the evidence is collected from applications using the Android Debugging Bridge (ADB) tool such as getting the image of the device using the command 'dd'.

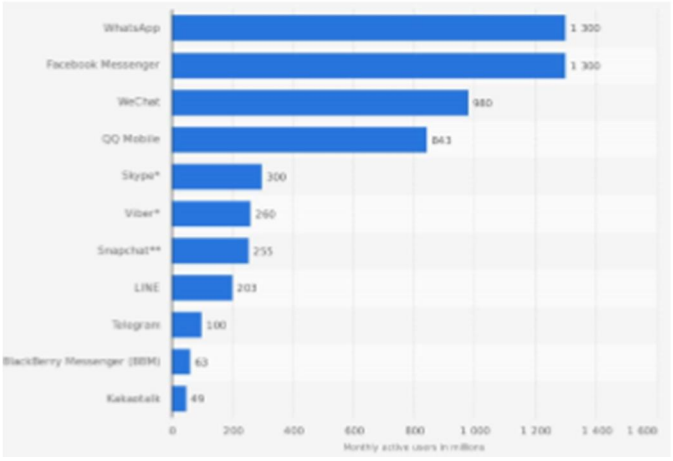


Fig.2 Most popular messenger apps as of Jan 2018, based on monthly active users (in millions) on a global scale.

The steps involved in the study of determining both data storage on the file system as well as categorizing the applications to retrieve the information without super user privileges are:

1) Firstly, install the applications on rooted as well as un-rooted android phones.
2) The test data is been fed and the data is obtained using the ADB tool.
3) Enable the USB debugging device for ADB to recognize, so to get the file location.
4) 'ADB pull' and 'ADB backup' commands are used for both rooted and un-rooted devices.

In WeChat, for the rooted device since the root privilege was accessible, using the 'ADB pull' command the entire directory could be obtained. But for the un-rooted device degradation of the version was required. Later 'adb install' and 'adb backup' commands were used to install and backup the data. A unique ID will be created in WeChat where this ID is stored in the main directory corresponding personal data folder.

In Telegram, for unrooted devices, no information was found available as it requires superuser privilege. For un-rooted devices 'adb pull' command is used to retrieve the database files. Telegram stored it's data in a binary serialized form making it difficult to retrieve the data. There are different types of chats in this application such as one-to-one regular chat, one-to-one secret chat, one-to-many channels, and many-to-many group. In this application, the encrypted algorithm used is the 256-bit symmetric AES encryption, 2048-bit RSA encryption, and Diffie-Hellman secure key exchange. It also has a self-destruction feature available where the message will disappear from both participant's chat history.

In WhatsApp, to access the data, a decryption key is required which only will be obtained using a rooted device. But by using the WhatsApp Key/DB Extractor one can extract a few messages including the timestamps even for an un-rooted device.

In Viber, for un-rooted devices the backup flag was 'on' and using the 'adb backup' the messages can be retrieved. The Android backup extractor converted the .ab file to .tar file format. No access to the database files was located only the manifest file was available. But for rooted devices 'adb pull' command was used to access the database files. However, the plaintext was been obtained even though the Viber messages were said to be encrypted.

Using the rooting device WeChat and Viber database files can be obtained whereas WhatsApp messages can be retrieved by using un-rooting devices. WeChat database file can also be obtained by degrading the version. Telegram can be considered a secure application since the chat database files are not been retrieved. However, for a rooted device, some evidence can still be obtained regarding the database files. Even though the files are been recovered, the encryption key is required to decrypt the files.

B. The Forensic Analysis of WeChat Message

WeChat is an instant messaging application that provides cross-platform services and is used for communication via text and voice. WeChat had many other features such as location sharing, payment, public account, etc. The data is encrypted and stored in the database called SQLite, a local directory of the application. WeChat is one of the top instant messaging, standalone applications which is free of cost and a cross-platform application where the basic operations are text, voice, video, and image messages. It even has other kinds of services like payment, location sharing, and a public account. Since it's an app with millions of users, it is prone to cyber security risks. This paper studies the forensics of the data and information to identify any clue of illegal activities of the criminals. The data which is either in the form of text, voice, video, or image is first encrypted and stored in the local directory of the SQLite database of the app. Previously, studies showed that retrieving the message content from the WeChat database is not an easy task as the messages are encrypted and stored in the database. So, this paper focuses on decrypting the database to extract the chat records of the existing as well as deleted ones. In this paper, first, the location of the data storage and ciphertext format is found. Then the encryption algorithm and its decryption key methods are been analyzed and the decryption method is been put in various circumstances. Finally, this paper also analyzed the recovery of deleted messages.

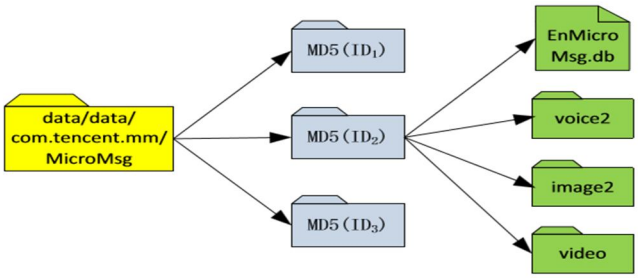


Fig.3 The storage path of WeChat messages.

In WeChat, the messages of all kinds are been stored in various installation folders of the app's storage locations. The app's messages are stored in "EnMicroMsg.db" which is the SQLite database. Every text message is directly stored in the database whereas the other format of messages is stored based on their path data and the original is stored in its corresponding storage locations. The database "EnMicroMsg.db" is encrypted so that the normal database browser cannot retrieve the messages which make it a secure application. Therefore it's necessary to analyze the encryption and decryption methods. Once the database is decrypted to its plain text, its messages are found in their corresponding folders.

This paper ensures that the database is been encrypted by SQL cipher which is an open tool for encryption. SQLite database consists of a fixed file size of 512 bytes and 32768 bytes. The default size for this particular application is 1024 bytes which are otherwise 1KB. SQLite is logically made up of many Btrees. A Btree consists of index and data of the table which is present in the data structure called "B+tree" and its table index as "B-tree"." SQLite master" stores the root page numbers of all data indexes and data tables. For decryption, it is mandatory to have the information on page 1. Other parameters required are the version of SQLite, file format version, etc. Also, the SQLite file header is another important parameter to consider in decryption.

From version 4.5 onwards, WeChat is implementing this encryption mechanism for its database. In encryption algorithm, to encrypt the database file AES algorithm is used and SQL cipher uses the function called "sqlcipher_page_cipher". It selects a CBC i.e. operation mode of the cipher block chaining as well as a key of 256 bits length. By using a derivative function "PBKDF2" is the encryption key derived from the password of length 7 bytes.

Under various circumstances, obtaining all the parameters required to get the decryption key. In Case 1, the phone is rooted. In this case, all the parameters are been acquired directly from the corresponding folders of the database. The parameters required for the decryption key are IMEI, UIN, and salt. IMEI can be acquired from the label behind the mobile. UIN is obtained from the configuration file named "systemInfo.cfg" and salt is the random value of 16 bytes long which is the same 16 bytes of the encrypted file "EnMicroMsg.db".

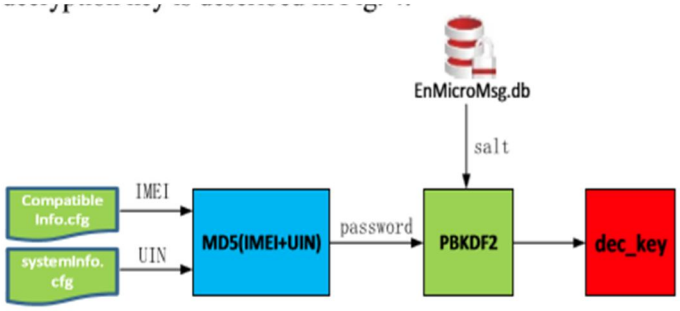


Fig.4 The process of decryption key derivation.

Now coming to Case 2, in this case, the phone is not rooted and the only thing we have is the encrypted database file. Since the IMEI and UIN are not known, the decryption key cannot be obtained by only having the random value i.e. the salt. Also, since the initial page has the header file of the database and the bytes are fixed, every password is been tried along with the corresponding decryption key. Once the ciphertext is been decrypt compare it with the plaintext, if they match then the decryption key of the encrypted database is found.

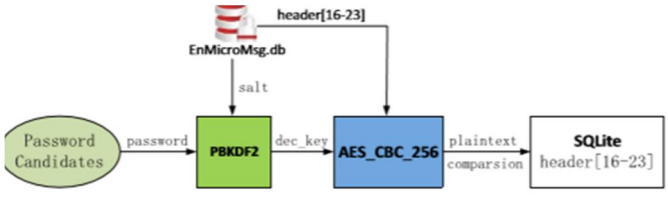


Fig.5 The correct password crack attempting.

Once the database is been decrypted it can be accessed and can retrieve any data like the contacts, messages, qmessages, etc. This paper studied the detailed analysis of the encryption database and methods to decrypt it and retrieve the messages both existing and deleted ones.

C. More is Less: On the End-to-End Security of Group Chats in Signal, Whatsapp and Threema.

The instant messaging application can be used for one-to-one communication or many-to-many communications (Group chats). This paper compares the security guarantees and the cryptographic techniques employed in the protection of privacy in the communications involved in instant messaging applications. There is an enormous amount of instant messaging applications used by people to communicate among themselves. The three applications that are being compared here are Signal, Whatsapp, and Threema. The communications in the dynamic group chats are analyzed and their important properties are extracted. The main aim of this paper is to improve the security and confidentiality of the group chats in various chat applications. The main focus is to preserve the integrity and groups' closeness i.e. the group allows only invited persons to manage the group activities. The presence of more powerful security properties such as Future Secrecy and Perfect forward secrecy in group chats is also analyzed. End-to-End encryption is one of the most essential security features to be met in any instant chat application to ensure secure transmission of messages.

The strength of any security protocol lies in its ability to prevent messages from unauthorized members that are not a part of the group. Also, the protocol must prevent an unauthorized user from adding himself to a group without an invitation. The reliable multicast, the integrity of communication, and the confidentiality of messages must be ensured in the group communications. A realistic and comprehensive security model is used to study the group communication protocols of various instant messaging applications such as WhatsApp, Threema, and Signal. The major security goals to be fulfilled are Confidentiality, Integrity, Authenticity, and Reliability. The traceable delivery, no duplication, no creation, and closeness are some of the other security requirements that are needed to be satisfied. Further advanced security goals such as future secrecy and perfect forward secrecy are also analyzed. The instant messaging applications make use of a central server to transmit messages between the senders and receivers. Therefore protocols are executed in an asynchronous environment in which only the server is online. Every user in a group maintains long-term secrets for the initial contact with other users and a session state for each group in which he is a member. The algorithm involves the following steps:

- snd → c: A vector of ciphertexts that need to be sent to the central server and hence to the network.
- $rcv^{snd, DelivM, ModG, Ack}(c)$: Processes the ciphertext that is received by the central server using some delivery algorithms.
- $SndM^{snd}(gr,m) \rightarrow id$: Sends content message m to group gr.
- $Add^{snd}(gr, V) \rightarrow id$: Adds user V to gr.
- $Leave^{snd}(gr) \rightarrow id$: Enables the process of a user U leaving from gr.
- $Rmv^{snd}(gr, V) \rightarrow id$: Removes a user V from gr.
- DelivM → (id, gr, V,m): Contains m with reference string id from sender V in group gr for showing it to the user U.
- ModG → (id, gr): Updates the content of group gr with IDgr = IDgr to gr after the current modification with reference string id.
- Ack → id: An acknowledgment that is already sent when and action with its associated id was received and processed by all its designated receivers.

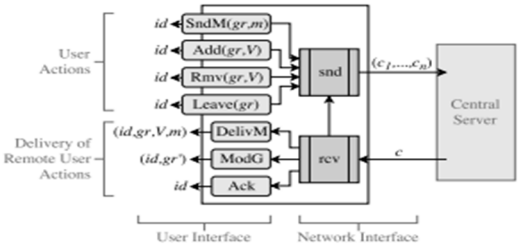


Fig.6 Overview over the syntax of group instant messaging protocols showing the interaction user's interfaces on the left and the interfaces on the application to the network on the right.

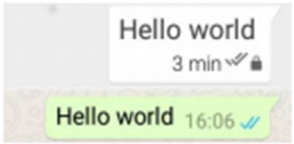


Fig.7 Double-check marks in Signal (upper screenshot) and WhatsApp (lower screenshot) indicating that a group message was successfully delivered to all members' devices.

Three types of attackers/intruders may harm the security of instant messaging protocols. They are as follows:

1) A malicious user may behave differently compared to the protocol specification used in the instant messenger.

2) A network attacker who gains complete control over the communication network and will be able to access and change all unprotected traffic.

3) A malicious server that impersonates itself to be the central server and also attacks the transport layer security.

In addition to the above-mentioned adversaries, there is also a threat of long-term secret compromise and session state compromise. Perfect forward secrecy is an encryption technique that automatically changes keys to encrypt the data repeatedly so that the sensitive data is not discovered easily. Future Secrecy, also called Post-Compromise Security is a technique that renews the session state more frequently to avoid replay attacks. No creation is a security goal that ensures no member other than the ones present in the group can communicate in the group. Traceable delivery means that everyone is a group receives the message(only assures the validity of acknowledgments. The order of the delivered messages must also be preserved by the security protocol. The authenticity of the messages involves ensuring two properties called the additive closeness (for adding a new member to the group) and the subtractive closeness (for removing a user or when a user deliberately leaves the group). The subtractive closeness is a correctness property that mainly focuses on reliability. Any protocol is said to be reliable when it follows either the Weak FIFO order or the Weak casual order.

The comparative analysis of the three instant messaging applications namely, Signal, Whatsapp, and Threema are as follows:

a) *Signal:* This is an open-source application that is well known for its key exchanges that achieve the goal of perfect forward secrecy especially for one to one message transmission. A session is established with the server by providing the user credentials such as the username and a password which is randomly chosen by the Signal server during a device initial usage. Credentials are provided for every request and the channel between the server and the user is protected by the Transport Layer Security. The root key is calculated using the X3DH Key Agreement Protocol that uses static and ephemeral Diffie-Hellman shares of both sides. The Double Ratchet algorithm (DR algorithm) is used for encryption. This algorithm is a combination of symmetric and asymmetric ratcheting. Ratcheting is a process of updating the encryption key regularly or more frequently. Chain keys are nothing but the initialization keys of the symmetric ratcheting. The symmetric ratcheting does not provide the future secrecy but provides perfect forward secrecy of the resulting keys. Unlike some apps that have administered groups, the Signal Messenger app executes non-administered groups where all the members can manipulate the group management. The group is uniquely identified by a random 128-bit vector ID. The Signal server cannot differentiate between the direct message and the group message. The message is state-fully end-to-end encrypted for each member in the group using a timestamp and the receiver ID. However, the acknowledgments are not end-to-end encrypted. Group management contains two protocol flows: an updated flow (creation of group) and a flow that is processed once a user leaves the group. There are three weaknesses in the Signal application. They are as follows: It may allow an attacker to become a member of the group and break the confidentiality of messages. It may also lead to forging acknowledgments where a malicious server breaks the traceable delivery and confidentiality. The third weakness is due to the reordering of messages that can be carried out by a malicious server.

b) *Whatsapp:* This is the most commonly used instant messaging application that uses closed source instant messaging protocol. It utilizes a Signal protocol for key exchange and encryption. The signal protocol also uses the X3DH Key Agreement Protocol and the DR algorithm for creating a private channel between two users.[3]Some of the weaknesses are the same as that of the signal application namely, forging acknowledgments, reordering of messages and also breaks the additive closeness due to the presence of a malicious server.[3]Further there is no future secrecy but preserves the Casual order.

c) *Threema:* Another application that uses centralized server architecture is the Threema which seems to be a proprietary closed source instant messenger. The application uses a proprietary key exchange protocol for the session establishment and the protocol uses three dependent Diffie Hellman key exchanges. [3]The weaknesses of Threema are as follows: It may allow an attacker to exercise the replay attack and hence breaks the No Duplication property. It also does not provide future secrecy or forward secrecy. It breaks the property of traceable delivery and may pave the way to reordering of messages. However, the Casual order is preserved.

D. Design and Implementation of AES and SHA-256 Cryptography for Securing Multimedia File over Android Chat Application

Providing Security and protecting the privacy of users has been one of the major concerns in developing any android chat application such as WhatsApp, Threema, Line, etc. There have been various security measures to protect the transfer of the confidential files among various users utilizing any chat application. A symmetric key encryption algorithm called the AES (Advanced Encryption Standard) is widely used to protect the sensitive data.

Any symmetric key algorithm uses a single private key for both encryption and decryption of the messages. AES is advantageous due to its large key size and the speed associated during the encryption/decryption process in the hardware or software. Many multimedia files such as images, audios, and videos exchanged between users need to be encrypted for secure transmission across the network. The user's key is found to be encrypted using the SHA-256 algorithm to protect the contents of the file. The encrypted files are stored in the UUencoding format to avoid compression of files while they are transferred to the receiver and thus maintain the originality of the file. This is a binary-to-text coding method with Uuencode based ASCII on UNIX.

The application uses Java language and was created on Android Studio. The main menu consists of an input file sub-menu, input key or generate a key, encrypt, and decrypt. AES algorithm is a symmetric algorithm that transforms plain text into a different form called the ciphertext. There are three kinds of AES keys varying by sizes 128,192 and 256. There are four-byte transformation processes involved in the AES algorithm that includes SubBytes, ShiftRows, MixColumns, and AddRoundKey.

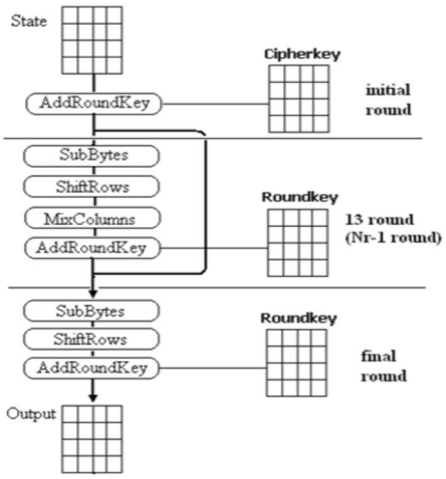


Fig.8 AES Encryption Algorithm.

The Secure Hash Algorithm (SHA-256) is a one-way encryption algorithm that makes it impossible to crack the original text from the ciphertext. This algorithm was designed by the National Institute of Standards and Technology in the year 2002. It generates an output of a hash value with a length of 256 bits.SHA-256 uses several logic functions including AND, OR, XOR, RIGHT SHIFT and RIGHT ROTATE operations. It operates on MD4, MD5 and SHA-1 algorithms.

1) Encryption Process

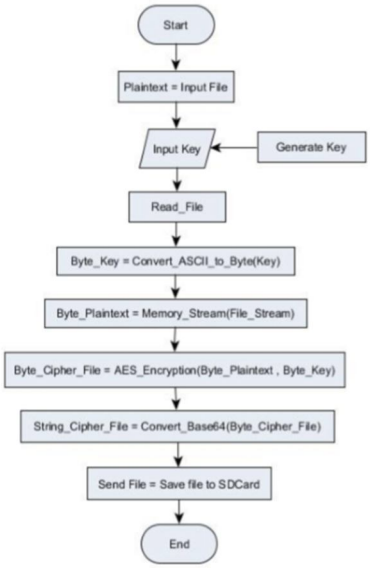


Fig.9 Encryption and Sending file method.

2) Decryption Process

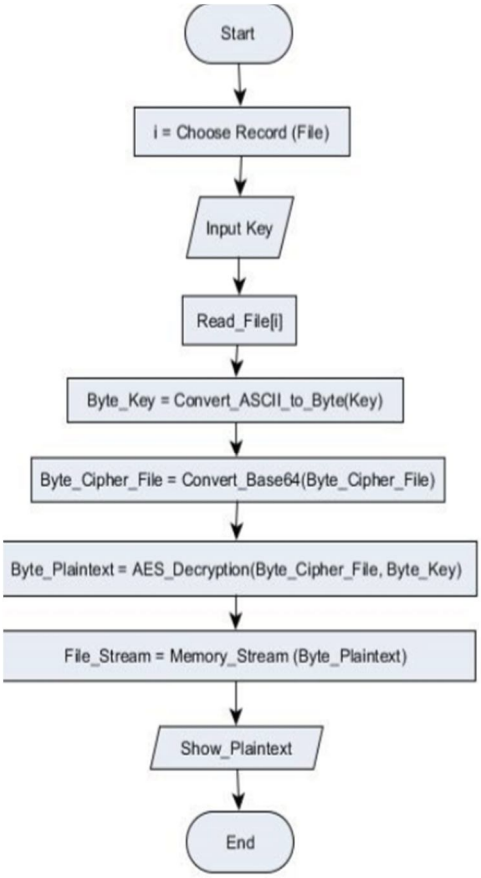


Fig.10 Decryption method.

The main functions of this application are to first prompt the user the upload an input file, enter a key-value, encrypt the file at the sender side, and decrypt the file with the shared key at the receiver end. The application is tested with audio, image, and video files.

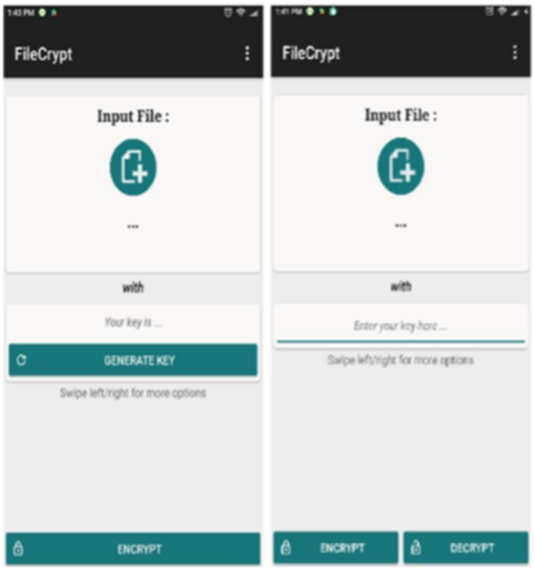


Fig.11 Application Interface.

The encrypted file will look similar to the figure given below:

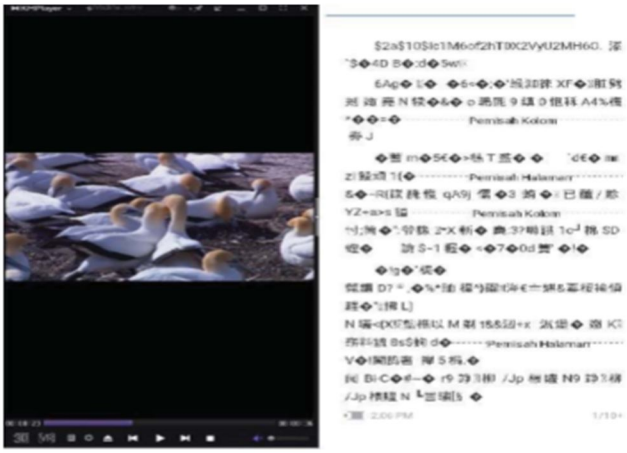


Fig.12 Encryption results of the video file.

The files are stored in a specific format with .enc extension. The size of the encrypted file will be slightly larger than the original file but does not change the value and meaning of the original file contents. The decrypted files also change in size. The PSNR value is generated to compare the contents of the original and encrypted files to ensure that the contents of both the files are the same and have not been altered during the process of encryption or decryption. Changes made occur only in the metadata and will not affect the actual contents of the file. However, the decryption process takes a longer time than the encryption process.

E. A Lightweight XMPP Publish/Subscribe Scheme for Resource-Constrained IoT Devices

Wireless technologies such as IPV6 and 6LoWPAN are employed in resource-constrained networks for facilitating communication in the lower layers but the incompatible application layer protocols create havoc when employed in heterogeneous networks. For this purpose, a protocol called Extensible Messaging and Presence Protocol (XMPP) is employed for better means of interoperability between heterogeneous networks. Thus, the model here employs XMPP in the IoT networks. The protocol is further optimized to provide a lightweight XMPP Publish/Subscribe Scheme for resource constraint IoT network. The XMPP server makes use of the advantages of the XEP0060 to maintain and manage one-to-many and many-to-many relationships between publishers and subscribers. Further, the proposed scheme also increases the battery life of the sleeping client devices.

XMPP (originally named:-Jabber) was mainly introduced for instant messaging and presence awareness in the year 1999. Some of its key advantages include openness, extensibility, and flexibility. To fulfill the non-uniform IoT application layer standards, various alternative protocols such as CoAP, MQTT, and XMPP are used in the resource-constrained networks. However, XMPP proves to be a much better option compared to MQTT and CoAP. Peter Waher put forward many valuable XEPs that provide a wide variety of architectures and use cases for employing XMPP in IoT. The publish/subscribe scheme id defined in XEP0060 which contains two subset protocols, namely XEP-0163 and XEP-0248. The basic functionality of Publish/Subscribe is that it allows the subscribers who subscribe to the content nodes to get data through one or many nodes according to the desire. The XEP-0163 supplies a subset that generates an XMPP user's JID, which acts as a virtual publish/subscribe service that makes the invention of the account owner's syndicated data and event notifications easier. The XEP-0248 organizes the nodes to build simplified relationships between them.

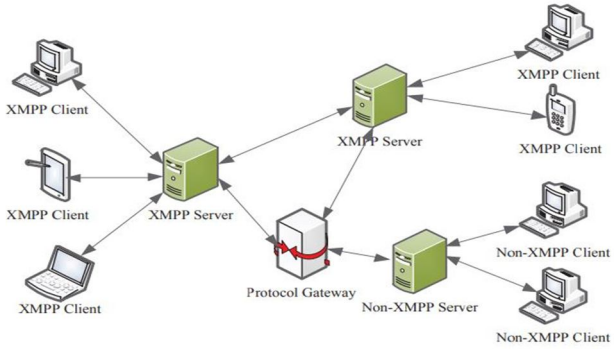


Fig.13 XMPP communication system architecture

The concept of the XMPP publishes/subscribe scheme is explained with the example of a communication network system consisting of three entities, namely client, server, and gateway. Each independent communication function has a separate address which is named Jabber identity (JID). A true JID comprises of valid elements which consist of local parts, domain parts, and resource parts. The server performs tasks like managing the connections, saving the messages, routing the messages between two clients, and so on. Communication between various instant messaging protocols is done by the gateway. XMPP follows a two-way communication. However, XMPP clients and XMPP servers can communicate directly to each other. But an XMPP and non-XMPP client or server cannot communicate directly. This is only possible through a gateway. Also, two XMPP clients cannot communicate with each other directly. They can communicate only through the XMPP server. For better, safe, and secure communication the transfer of messages between XMPP servers uses a Simple Authentication and Security Layer (SASL) and also encrypted with Transport Layer Security (TSL) to encrypt the information as well as secure the account.

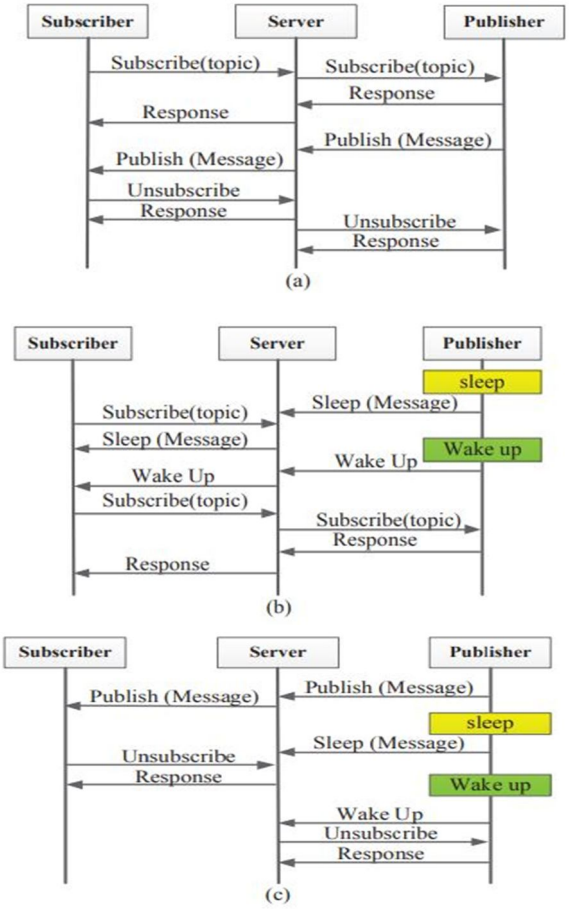


Fig.14 The specific procedure of the proposed scheme. (a) The subscribers subscribe/unsubscribes to the active publisher. (b) The subscriber subscribes to the sleeping publisher. (c) The subscriber unsubscribes from the sleeping publisher.

Generally, the publish/subscribe service gives importance to networks that are heavy to work which limits both resource and energy. The XMPP publish/subscribe system in IoT, the subscribers subscribe to the interesting data, and though the servers the publishers get notified and send the data when the subscription rules meet. Along with the data transfer, it also acts as a proxy sleep service, when the publisher is inactive for a long time. The XMPP client acts both as a publisher as well as subscriber or both at the same time. Firstly the subscriber should find the resource and then subscribe to it with a specific node. The publisher sends a request to the server, where the server acts as a transport channel and sends it to the requested publisher. Once the request is received by the publisher he responds to it by sending the required resources to the subscriber. The scheme mentioned in this paper takes advantage of the XEP-0060, where the server is used for maintaining and managing the connection between the publish/subscribe entities and gathers and stores the authorized subscribers. Subscribers not only send the information which is been requested but also send some additional information which will be useful for other subscribers.

In IoT applications relating to sensor networks, some nodes involve inactivity all the time and some nodes which go into hibernation when not used for a long time. The sensor nodes are further divided into two types of modes that is an active mode and sleep mode. In an active mode, nodes always work continuously whereas in sleep mode the sensors hibernate until it wakes up and is ready for work. The subscription and un-subscription procedure for actives nodes is, first the subscribers send a request through the server. Once the server sends the request to the publisher, the publisher needs to validate its request and sends a subscription request to the subscribers. Then the publishers constantly keep sending messages to the subscriber. The procedure for sleeping nodes is, when the publisher is sleeping it does not get any request from the server, and thus the subscriber gents into a "confused "state. Therefore it would be efficient to maintain a sleep function. In multiple publications, In Publish/Subscribe Object and XML Grammar Design, to attain easy access and utilize it, every device should have a Data Object (DO).

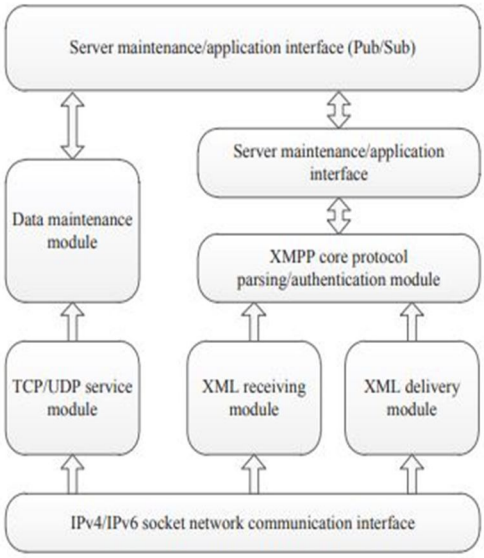


Fig.15 XMPP server software architecture

The XMPP publish/subscribe scheme is established on the 6LoWPAN platform. Nine different sensor nodes were used as publishers and 6LoWPAN gateway along with the XMPP server and a mobile device were used. The sensors and gateway are connected wirelessly and mobile device is connected via wi-fi channel. The sensor nodes were deployed with STM32L152 Microcontroller Unit (MCU) with ARM Cortext-M3 32-bit core operating at 32 Mhz frequency, along with a Linux platform on resource-constrained IoT devices.

The sensor nodes in IoT are usually based on battery and restricted resources. In this paper, the publish/subscribe scheme can allow the publisher to restrict the data based on the subscriber's need so that the energy consumption will be reduced which thereby leads to the decrease in battery consumption. To provide a normal publish/subscribe service, the node's work time is extended to 10 times more on resource-constrained devices. The full publish/subscribe is not globally implemented due to its complexity.

The benefits of this scheme are it is mainly used in the field of IoT applications and is fully lightweight. The data can be adjusted based on the subscriber's needs so that the battery consumption is reduced. Moreover, using the sleeping function this scheme makes it more efficient for the publisher to work more effectively even in the sleeping mode. On the contrary, although security measures have been looked into, no steps are taken forward to look into end-to-end data transmission. Also, it’s not implemented for large-scale yet. There has always been a tradeoff between processing capacity and response time.

This scheme has improved the earlier publish/subscribe scheme based on the XMPP protocol for data publication in IoT. It draws some features from XEP-0060 as the server manages the connections between many subscribers and transfers data to clients which leads to many server-side issues and complexities. It mainly focuses on saving energy and resources which helps in longer battery life by using the sleeping function. Even though this scheme is efficient it does not provide Quality of Service (QoS) which may lead to packet loss. Therefore in the future, the aim would be to provide the reliability of data transfer and providing a better QoS. When the devices are connected on a large-scale it becomes an issue in the implementation of XMPP. Another key factor to look into is the security of the resource-constrained devices.

F. Implementation Application Internal chat Messenger using Android System

Chat messenger applications are utilized worldwide for communication purposes. Communication takes place via the internet and there are many chat applications out there that make the communication a lot more user-friendly and efficient. Despite there being many advantages of the chat messenger, there has still been a language barrier to communication with different kinds of people. Therefore it is necessary to translate the language for better communication between users. With this feature implemented in the application, users can communicate with foreigners without any language barriers. Group or distributed team is another feature used by the android chat messenger which is brought to existence for internal communication, which utilizes an internal server.

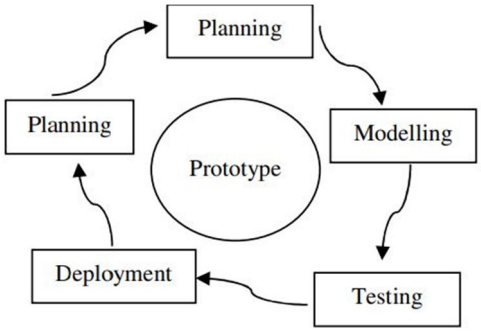


Fig.16 Prototype development model

This feature reduces the usage of memory on the device. Communication between two or more people are done using instant messaging where people are allowed to send text, voice, and video messages. Along with this, there are many features and functions integrated. Chat messenger utilizes a LAN network or area network or Bluetooth to communicate between users.

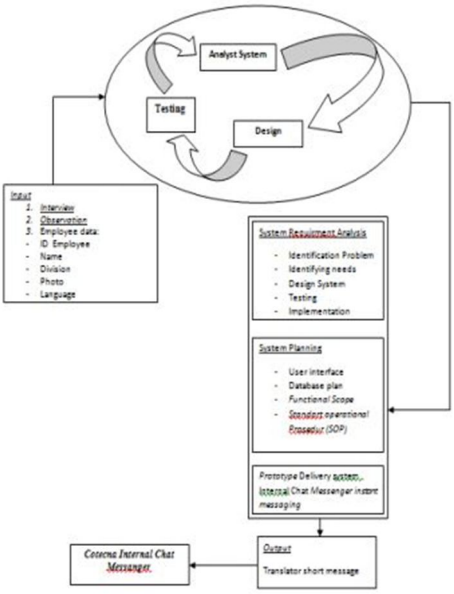


Fig.17 Concept Chat Prototyping

Google API is the simplified code used for web applications that utilize the features and functions provided by Google. Firstly, the process requires gathering and analyzing the system requirements by doing some research on the employees of different branches and collecting their data to discover the problems of the user on different devices. The programming is done with the help of an android studio and MySQL database based on the prototype for the displaying screen.

When Alice sends a message to Bob, first the message is sent to the server in a particular language. Once the message reaches the server which is installed with API Google translate, it begins to translate the message based on Alice settings of the language. Then the message is translated to the language which is preferred on Bob's device setting and vice versa. Since it is based on translation, data must be acceptable in particularly the language used.

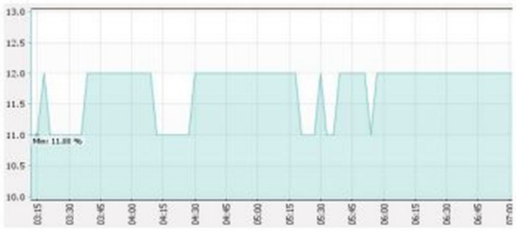


Fig.18 CPU usage

Fig.19 RAM and GPU monitoring

The method showed that it can achieve good sync and misc time performance. This paper helps to minimize space storage and also shows the high performance of CPU, RAM, GPU, and bandwidth. The future of this paper looks into adding some additional features like video call, map, etc. to enhance the application.

III. COMPARITIVE ANALYSIS OF EXISTING CHAT APPLICATIONS

SL. NO	App	Pros	Cons
1.	WhatsApp	• End-to-End Encryptions to secure the messages from third parties • Ads Free&Free of cost • Starred Messages • Delete sent messages	• No Virtual Assistant • Self-Destructing messages • File Size Limit(up to 100 Mb) • Contact Number Required • Internet Access is a must • Sharing of Wrong news and information without any verification or validation
2.	Viber	• Free of Cost • Chat Extensions • End-to-End Encryption • Connects a non-Viber user through mobile or landline also with only low cost • Can transfer calls between devices • Option to set message and video destruction timer	• Do not have a call blocking feature • Low Quality • It does not use GSM architecture and services to channel calls and messages.
3.	Signal	• Extremely familiar, intuitive interface • Allow audio chat with verification • Self-destruction of messages after a set amount of time • Supports encrypted group chats • Note to self-feature • Message mirroring on signal desktop	• Bugs and glitches • Not widely used or known • Unreliable notifications.
4.	Threema	• End-to-End encryption • Threema ID	• Not Free of cost • All features are not applicable for a few devices

		• Multimedia encryption • It uses standard push notification service provided by OS • Contact synchronization is optional • Protects from man-in-the-middle attack by allowing you to verify the ID of whomever you are communicating with.	• The app does not connect when the internet connection is slow.
5.	Google Hangouts	• Conducts Webinars • Broadcast live • Video conferencing • Chats are archived in Gmail so you can always search them back. • Screen Sharing	• Not End-to-End encrypted • Not used by many
6.	Telegram	• High security through secret chats • Can share a huge amount of data without any loss • Ads Free • Self-destruction mode is available • Messages can't be forwarded from secret chats.	• Cannot know the status of the other person. • Binding to the phone number • If the messages are sent via an ordinary, rather than a secret chat, then formally, they are not encrypted. • No voice messages
7.	Hike	• Timeline features • Hidden Mode • Live filters • News, cricket, games, etc. • Share large files up to 100mb • Free of cost	• No video calling feature • Too much of lag • Should improve the user interface • No End-to-end encryption
8.	WeChat	• Multi-Platform • Has its unique QR code • Free of cost • Walkie-Talkie, a live chat that helps people to communicate virtually.	• Less expressive • Must have high internet service • Easy to get personal data • Security issues in business • The device is prone to be attacked by a virus.
9.	Facebook Messenger	• Faster communication • Free of cost • Voice calls • Making money on the internet • Connect to anyone, anywhere.	• Difficult to completely protect your privacy. • It can quickly drain the battery of the device. • Takes up a lot of space • Too much spamming • Your device is prone to viruses and hackers.
10.	Wickr Me&Wickr Pro	• End-to-End encryption • Self-destruction of messages. • User-defined lifespan for media and messages. • Register with only a username and password. • Screenshot detection	• It may screenshot the content. • It has a smaller user base. • Measures of security don't offer sync among multiple phones. • Does not have encrypted vault. • Asks for the email address. This leads to spamming. • You cannot be anonymous anymore.

IV. CONCLUSION

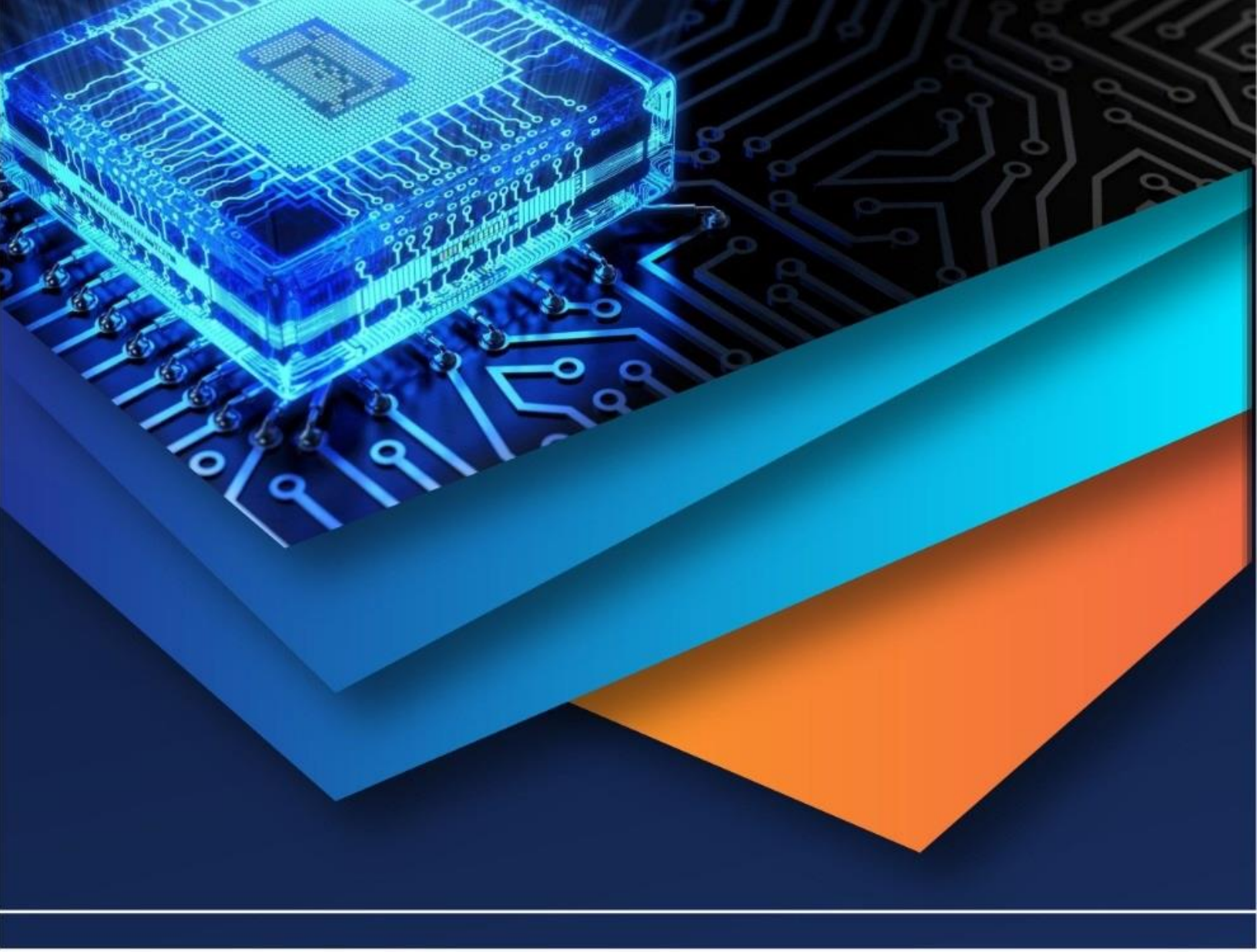
In this period of technological advancements, the numerous instant messaging applications provide an easier and efficient means of communication at a minimal cost. These instant messaging applications work with the help of an internet facility and a centralized server like XMPP that acts as a set of open technologies for instant messaging, presence, generalized routing of XML data, and so on. However, there is a need to improve the security concerns of these instant messengers to make them more reliable. Defense organizations need a highly smart and secure means of communication to protect the secrecy of their messages and data. Hence in this paper, we compare the existing instant messaging applications and their protocols and detect the technologies used in those applications to build an efficient instant messenger for Defense organizations and also including some artificial intelligence techniques to make them more attractive and human-friendly.

V. ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany a successful completion of any task would be incomplete without the mention of people who made it possible, success is the epitome of hard work and perseverance, but steadfast of all is encouraging guidance.
So, with gratitude I acknowledge all those whose guidance and encouragement served as beacon of light and crowned our effort with success.
I am thankful to our Management for being a constant inspiration and providing all the facilities that needed throughout the Internship.
I would like to thank our Principal Dr.P.Mahabaleshwarappa, for his constant guidance, advice and encouragement to complete the Internship.
I consider it a privilege and honor to express my sincere gratitude to our beloved HOD, Mrs. Prameela Devi for her constant encouragement and all the support provided during this Internship.
I convey my sincere thanks to my guide Mrs. Suganthi Sivakumar, Assistant Professor, Dept. of CSE for his valuable guidance throughout the tenure of this Internship, and those support and encouragement made this work possible.
It's also a great pleasure to express my deepest gratitude to all my faculty members of my department for their cooperation and constructive criticism offered, which helped me a lot during my project work.
Finally, I would like to thank all my family members and friends whose encouragement and support was invaluable.

REFERENCES

[1] Heng Wang, Daijin Xiong, Ping Wang, Yuqiang Liu, "A Lightweight XMPP Publish/Subscribe Scheme for Resource-Constrained IoT Devices.", Journal Article, IEEE Access, Volume: 5, 2017.
[2] Noveline Aziz Fauziah, Eko Hari Rachmawanto, De Rosal Ignatius Moses Setiadi, Christy Atika Sari, "Design and Implementation of AES and SHA-256 Cryptography for Securing Multimedia File over Android Chat Application", Conference paper, 2018.
[3] S V V Satya Surya Sravan Kumar Mangi, Saddam Hussian.SK, Leelavathy.N, "An Approach for Sending a Confidential Message to the Restricted Users in Defense Based Organization", Conference paper, 2019.
[4] Jimmy Florez Z, Camilo Logreira R, Mario Munoz, Jesus Francisco Vargas, "Architecture of Instant Messaging Systems for Secure Data Transmission", Conference paper, 2016.
[5] Juha Jarvinen, Aleksi Marttinen, Marko Luoma, Markus Peuhkuri, Jukka Manner, "Architecture of XMPP proxy for Server-To-Server Connections", Conference paper, 2017.
[6] Oral Gurel, Mehmet Ulas Cakir, "Promising XMPP Based Applications for Military and Defence Systems.", Conference paper, 2013.
[7] Tarun Mehrotra, B.M.Mehtre, "Forensic Analysis of Wickr Application on Android Devices", Conference paper, 2013.
[8] Lijun Zhang, Fei Yu Qingbing Ji, "The Forensic Analysis of WeChat Message", Conference paper, 2016.
[9] Fu-Ching TSAI, En-Cih CHANG, Da-Yu-Kao, "WhatsApp Network Forensics: Discovering the Communication Payloads behind Cybercriminals.", Conference paper, 2018.
[10] Khushboo Rathi, Umit Karabiyik, Temilola Aderibigbe, Hongmei Chi, "Forensic Analysis of Encrypted Instant Messaging Applications on Android", Conference paper, 2018.
[11] Paul Rosler, Christian Mainka, Jorg Schwenk, "More is Less: On the End-to-End Security of Group Chats in Signal, WhatsApp, and Threema.", Conference paper, 2018.
[12] Sergej Dechand, Alena Naiakshina, Anastasia Danilova, Matthew Smith, "In Encryption We Don't Trust: The Effect of End-to-End Encryption to the Masses on User Perception", Conference Paper, 2019.
[13] Zhen Wang, Zhaofeng Ma, Shoushan Luo, Hongmin Gao, "Enhanced Instant Message Security and Privacy Protection Scheme for Mobile Social Network Systems", Journal Article, Volume: 6, 2018.
Robi Sanjaya, Abba Suganda Girsang, "Implementation Application Internal Chat Messenger Using Android System", Conference paper, 2017



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY