*SAI Intelligent Systems Conference 2016*
*September 21-22, 2016 | London, UK*

# SIMSSP: Secure Instant Messaging System for Smart Phones

Kahtan Aziz
College of Engineering Computing.
Al Ghurair University
Dubai, United Arab Emirates
Email: kahtan@agu.ac.ae

Saed Tarapiah
Telecommunication Engineering Dept.
An-Najah National University
Nablus, Palestine
Email: s.tarapiah@najah.edu

Shadi Atalla
College of Information Technology (CIT)
University of Dubai
Dubai, United Arab Emirates
Email: satalla@ud.ac.ae

***Abstract*—Hand held smart devices such as smart phones, Personal Digital Assistant (PDAs) and tablets are ubiquitous and touch almost all people' life functions. In most cases, Hand held devices can be categorized as messaging-centric that conveying information in many form of electronic messages i.e. e-mail, Short Message Service (SMS) and instance messaging (IM). IM software exchange large amount of messages (e.g. plain text, images and files) over insecure communications networks for example wireless and internet communication networks. IM data are transferred through communication networks in different locations. Thus, the potential for unauthorized access, abuse, or fraud is not limited to a single location but can occur at any access point in the communication network. The objective of this work is to provide a secure effective platform for encrypting and decrypting IM data. The proposed platform is based on pre-shared secure key mechanism and Vernam algorithm in order to guarantee secure transmission and storage functions for IM applications through a well-established Application Programming Interface (API) that provides seamless integration with existing IM software.**

***Keywords*—Smartphone; Security; Instance Messaging**

## I. Introduction

Digital data generated by IM software are vulnerable to destruction, misuse, error, fraud, and hardware or software failures. Mostly IM software use the Internet which is an open system and makes internal and private users' networks more vulnerable to actions from outsiders. Hackers can take control of a user PC or penetrate a corporate networks, causing serious system misuses and disruptions. Furthermore, Wi-Fi networks can be easily penetrated by intruders using sniffer programs to obtain IM data while travelling on the air collecting the users confidential and private data. Unsecured IM Software presents problems because it leaves the doors open for intruders to gain access on large amount of plain text and unencrypted files e.g. photos while being exchanged between the IM software two ends. In order to alleviate this issue system developers use encryption, the coding and scrambling techniques of messages for guarantee a secure and encrypted digital transmissions over such unprotected networks [1] [2] [3] [4].

Data encryption is the foundation step for realizing secure data storage and communication methods. Encryption algorithms transform any plain text and file contents into a secret code called cipher. The security appears when an entity tries to transform back the secret cipher into the original plain text or the file contents such translation process of data from cipher to plain data is called decryption. To successfully decryption and access the original contents from the cipher a secret key or password is required. Generally, data encryption is categorised to be asymmetric encryption called public key encryption or to be symmetric key encryption called pre-shared key encryption. In computer science, cryptography is the art and science of using mathematics to encrypt and decrypt information to keep messages secure, as long as there is communication, people practice the cryptography to make their transaction more secure. Figures 1 and 2 illustrate a cryptography system that enables its users to store sensitive information or transmit them across secure networks so the information can't be read by anyone except the intended recipient. Nowadays the use of cryptography is almost everywhere, with particular focus on in cell phones, emails, computers, banks. Along with cryptography in computer science, cryptanalysis is the mechanism of reversing the unreadable cipher back into readable plain message without knowing how the original message was initially converted into the corresponding cipher [3].

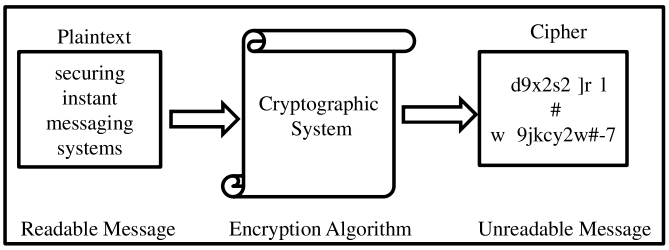


Fig. 1: *Cryptography system*, here is an example of encryption system, that can be viewed as a secrecy enabler tool that lock data when they are transmitted. The sender will use his assigned key to encrypt the message.

Instant Messaging applications offers near real time communication among their users; where in few steps such application enabled information sharing such as text messages and files. IM information sharing is classified based on its function into two cases. the first case involves interaction between two users, while in the second case involves exchanges information among a set of users at the same time. Such features e.g. easy of use and multi-functions have made IM applications to stay at top on widely used and installed smart-phone and tables. Generally, IM are based on the well-known client server distributed computing paradigm, although in few cases IM

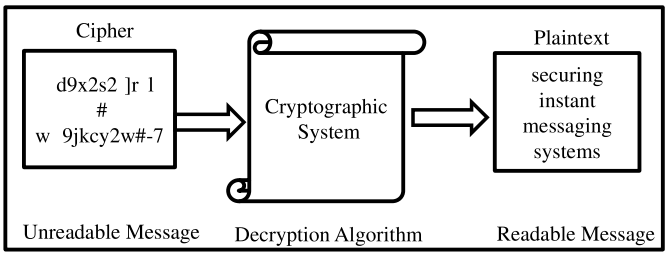


Fig. 2: *Cryptography system*, here is an example of decryption system, that can be viewed as a secrecy enabler tool that unlock data when they are received. The receiver locates the sender's key and uses it to decrypt the message.

application cloud allow a direct communication between its users, .i.e. in case of large file sharing, this is important to offload the server from a large computation efforts [5].

Notwithstanding, IM applications pose several ethical and security issues such as users' privacy [6], [4]. The privacy issues exist due to fact that IM applications requires from its users to create personal profiles as foundation step to be able to use the IM application as well as such application uses the satellite signals Global Positioning System (GPS) to infer the user location changes as well as IM applications provide their end-users presence information of [7]. Most importantly, IM applications raise a serious security issues; for instance, sensitive personal information are exchanged over insecure networks (internet and wireless). This paper focuses on the above security issue; more in details, the safety of the message transmission is analyzed to find out whether One Time Pad (OTP) [8] encryption algorithm can be used to ensure that IM information are the exchanged and/or stored with high level standard of security while maintaining near real time IM communication without affecting the performance of IM application.

## II. Related Work

Since almost 100 years ago, Gilbert Vernam developed Vernam cipher algorithm which is proven to be a secure cryptography system. In Vernam algorithm the plain text message is composed by a binary sequence of 0s and 1s; and the secret key is a truly random a binary string of 0s and 1s such that the key length equals to message length. Then the encryption process is done by adding the key to the message modulo 2, bit by bit. This process is often called exclusive or, which is denoted as XoR. In literature Vernam cipher also known as one time pad. In one time pad cipher ensures full secure cipher which involves and requires two conditions to be satisfied; the first condition, holds that the key must not be repeated. while, the second condition, to use a purely random key that is as long as the plain message [8]. The essence of the proposed work is based on developing a smart phone application based on One Time Pad (OTP) [8] encryption algorithm.

One Time Pad (OTP) algorithm has been applied by Arifin, Anthony, and Hazinah Kutty Mammi to secure IM interactions among its end-users in [5]. Moreover, authors presented an end-to-end cryptography system leveraging on OTP, which is capable to effectively encrypt/decrypt all the IM messages transmissions -without burden the real time communication efficiency which is the main theme of having IM applications. Furthermore, the paper illustrates that less than 0.05 seconds is required by OTP to process a plain text message of length 100 characters from end-to-end including encryption, sending, receiving and decryption. This small amount of time (0.05 seconds) is not easily noticeable by the end-users; hence, our work somehow can be viewed as an extension to this one [5] although, our work implements Vernam algorithm as described by its authors and is focalized on smart phone and hand help devices in general.

Generally, several entities such as standardization and academia organizations have been studying IM protocol as well as its security threats and counter attack measures. For example, Transport Layer Security (TLS) [9] is used by Extensible Messaging and Presence Protocol (XMPP) in order guard XML from tampering and eavesdropping [10]. SIMPLE [11] leverage on TLS and Secure/Multipurpose Internet Mail Extensions (S/MIME) protocols for realizing mechanisms for user authentication, end-to-end encryption, prevention of replay and denial of service attack. SIMPP [12] presents secure mechanism for both type of communication used by IM software: IM client-client and IM client-server.

Pidgin [13] is a software platform combines the access to several widely-used IM applications into single user interface. Pidgin end-users are able to use their IM accounts provided from multiple IM software simultaneously on single screen. For example a user can communicate with friends on MSN, Google Talk, AOL Instant Messenger(AIM), ICQ, Jabber/XMPP, Yahoo chat room and many others all together at the same time. More importantly, Pidgin employee advanced encryption algorithm based on the concept Off-The Record (OTR); Pidgin uses NSS crypto library from Mozilla which is based on 4096 bit RSA cryptosystem. Pidgin is resistant to man-in-the-middle attacks.

## III. Design and Algorithms

This section introduces Secure Instant Messaging System for Smart Phone (SIMSSP) which is the main output of this work. SIMSSP is an open source software library implementing One Time Pad (OTP) for securing IM communications. In Our vision, SIMSSP users are smart phone developers aiming to build a secure applications. Whereas, SIMSSP provides its users with encryption and decryption functionalities for storing locally and conveying to a remote users sensitive information. Mainly, SIMSSP is designed to suit for IM software packages security requirements.

SIMSSP main design requirements and features are:

1) Effectively encrypt IM data with a high security standards without burden IM software functions mainly the near time transmission among IM end-users in any collaboration session.
2) Light weight in terms of computation resources minimizing the needed cup cycles and memory units needed to perform encryption or decryption tasks due to resource limitation in some types of the targeted platforms.
3) Establish well organized Application Programming Interface(API) to simplify the integration efforts

while combining this software library to new IM software. The library must be written using portable programming language to ensure it is running in almost all hand held and smart phone devices.

SIMSSP is composed of two parts; the first one is providing both encryption and decryption functions to handle text messages and text files;and the second part is able to encrypt and decrypt binary files e.g. pictures, videos and audio files. Text related part is mostly used by the IM sender to encrypt IM text messages and decrypt the cipher on the receiver side. While the binary handler is used to process file attachments and picture sharing using IM application.

Text encryption/decryption algorithm is shown in the below box. When the algorithm takes an input as a plain text the output will be the cipher resulted of encrypting input message; while if the input was the cipher then the output is plaintext and the algorithm performs decryption.

**The Text Encryption Algorithms**

- **Step 1.** Encode using the ASCII code each character of plaintext and encode it into binary bits.
- **Step 2.** Repeat the key to make it as the length of plaintext.
- **Step 3.** Take the ASCII code for each character of key and encode it into binary bits.
- **Step 4.** Perform the XoR operation between the bits of plaintext and key.
- **Step 5.** Block the result bits in group of 7 bits.
- **Step 6.** Decode the ASCII code and write its character in cipher text.

Binary decryption algorithm is shown in the below box.

**The Binary Decryption Algorithms**

- **Step 1.** Take the color points of the picture.
- **Step 2.** Repeat the key on the picture box.
- **Step 3.** Take the ASCII code for each character of key and encode it into binary bits.
- **Step 4.** Perform the XoR operation between the bits of color points and key.
- **Step 5.** Block the result bits in group of 8 bits.

As depicted in Figure 3, which illustrates the overall text encryption/decryption algorithm, when the plaintext message is “GP”, and the encryption key is “hi”. In order to decrypt the cipher “/&” the reader need to replace the letter “P” by “/” and “G” by “&” and then perform the same steps show in figure 3.

## IV. SIMPLE PROTOTYPE IMPLEMENTATION AND VERIFICATION

A well functioning system prototype was built composed of the following components: Online Web application acting as IM server, two Apple iphone devices each of them is running a custom smart phone IM client leveraging on SIMSSP library. The main prototype setup is depicted in Figure 4.

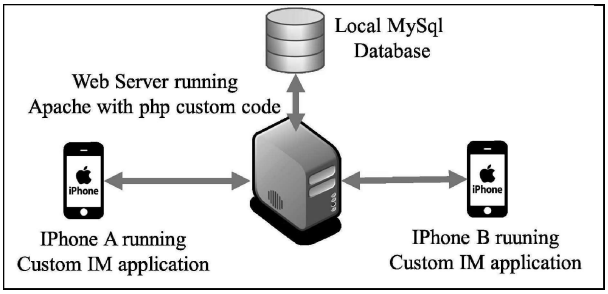


Fig. 4: Illustrates Simple Prototype for Verifying SIMSSP Library functions.

The SIMSSP library is developed using the C# programming language [14]. The resulted library code was complied using the Xamarin compiler [15]. SIMSSP chooses Xamarin compiler in order to ensure that it is compatible with both iOS and android which are the widely used Operating Systems (OSs) for smart phones and hand held devices in general.

The custom IM client software has been tested for both iOS and Android as it is also developed using Xamarin compiler and C# programming language. This software mainly composed of three modules: The first one is the SIMSSP library used to perform encryption and decryption functions for both binaries and text contents. The second module is the communication part that is involved with interaction with the web server using HTTP request and JSON data representation for sending and receiving the IM messages and attachment such as pictures sharing. The last module is the Graphical User Interface (GUI) which is very basic one which contains the main menu of the program with only four buttons in the main screen as depicted in Figure 5.

When the user click the *encrypt text* button the IM client opens a new screen as depicted 6 to allow the user enters a text message then the IM client encrypts the message and sends the cipher to the server.

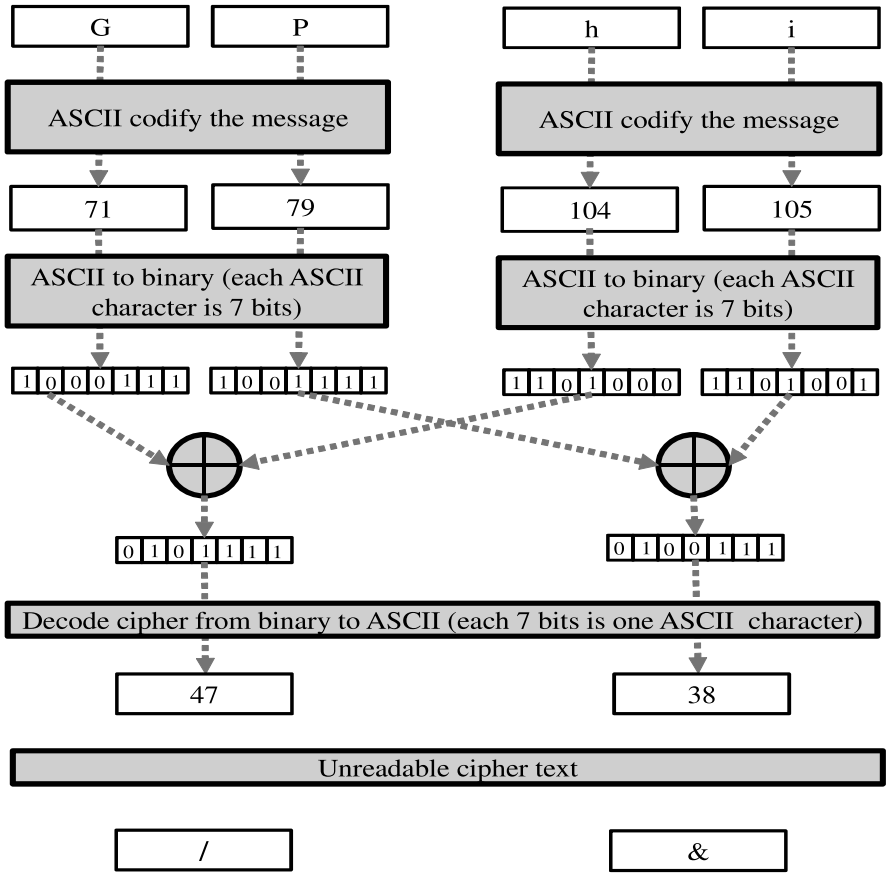


Fig. 3: illustrates the overall text encryption algorithm, when the plaintext message is “GP”, and the encryption key is “hi”

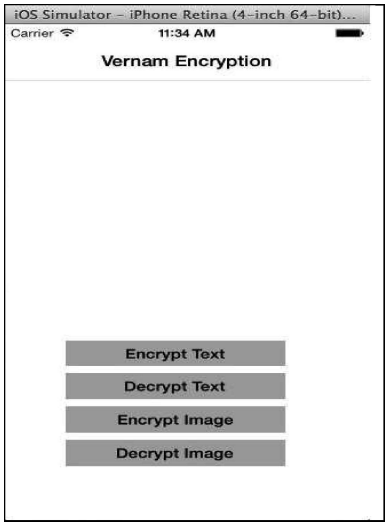


Fig. 5: Illustrates the Main Menu of the Customised IM Client Software Developed in C# and Running in IPhone Device.

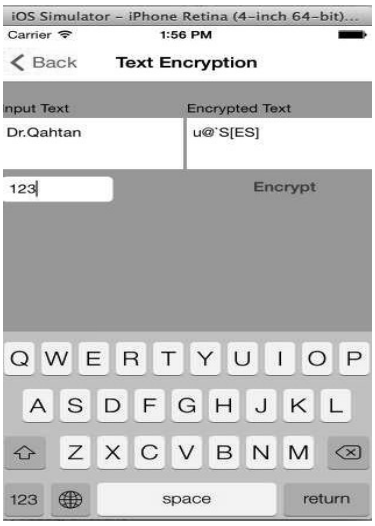


Fig. 6: Illustrates the Message and its Cipher after Successful Encryption.

When the user clicks the *decrypt tex* button the IM client opens new screen as depicted 7 to which pull the cipher from the web server then it decrypts to its original readable form.

If the user clicks the *encrypt picture* button then the IM client will open a new screen as depicted 8 to allow the user to enter a text message then the IM client encrypts the message and it

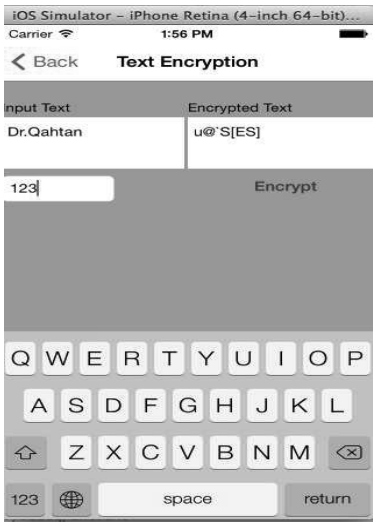


Fig. 7: Illustrates the Cipher and its Original Message after Successful Decryption.


Fig. 8: Illustrates the Picture and and the Corresponding Cipher after Successful Encryption.

sends the cipher to the server.

When the user clicks the *decrypt picture* button the IM client opens a new screen as depicted 9 to which pull the cipher from the web server then it decrypts cipher to its original image form.

The web-server component of the prototype is provided with a simple web-based application acting as intermediate IM server that receives and stores into the database the text and binary ciphers from the IM clients. The implemented web application is accessible through a standard smart phone and tablets devices. The Representational State Transfer (REST) or (RESTful) software architectural style has been adapted to simplify the interfacing between the server side and the IM clients. Moreover, Server client communication uses JavaScript Object Notation (JSON) data representation. The IM server


Fig. 9: Illustrates the Cipher and its Original Picture after Successful Decryption.

consists of the following:

1) The main server logic has been developed using PHP programming language and this logic has been deployed on an Apache web server. This tire uses RESTful style to expose its internal functionality towards the client side application as well as this software tire leverage on the MySQL native driver for PHP in order to store and retrieve data.
2) The MySQL database server and this database is used to store all the IM data in cipher format. This design of this module is based on a relational database structure.

In a RESTful software architectural style every thing is a resource and for each resource there is a Universal Resource Identifier (URI) that represents the corresponding resource unique address. Moreover, there are four verbs that are usable to transfer and manipulate any resource representation. Finally, the word CRUD refers to these four verbs and the C letter is coming from create, R from read, U from update and D from delete.

## V. Conclusion

The this work introduces SIMSSP library developed using the C# programming language. The resulted library code is an open sources platform that is compatible with iOS and Android Operating System. SIMSSP main objective is to provide a secure effective platform for encrypting and decrypting IM software data able to prevent eavesdropping attacks. The proposed platform is tested and verified using a simple pre-shared secure key mechanism and Vernam algorithm in order to guarantee secure transmission and storage functions for IM applications through a well-established Application Programming Interface (API) that provides seamless integration with existing IM software.

A full end-to-end prototype leveraging on the proposed SIMSSP library was described in details. The presented proto-

type illustrates that SIMSSP to process a plain text message of length 100 characters from end-to-end including encryption, sending, storing and retrieving from the database ,receiving and decryption does not the IM end-users perception of near real time communication offered by IM applications.

The merit of this project relies on two factors; first its open source SIMSSP library that is reusable and integrable by smart phone developer application specially IM software; secondly, advanced key management and distributed techniques cloud be used to avoid insecure key exchange between the two ends of IM applications which somehow may limit the usability of the proposed software library.

## REFERENCES

[1] Leavitt, Neal. *Instant Messaging: A new target for hackers*, Computer 38.7 (2005): 20-23.

[2] Mannan, Mohammad, and Paul C. van Oorschot. *Secure public instant messaging: A survey*. Proceedings of Privacy, Security and Trust (2004).

[3] Kahate, Atul. *Cryptography and network security*. Tata McGraw-Hill Education, 2013

[4] Laudon, Jane P. & Laudon, Kenneth C. *Essentials of Management Information Systems*. (10th Ed.) Prentice Hall, 2013

[5] Arifin, Anthony, and Hazinah Kutty Mammi. *Study on the Effectiveness of Securing Instant Messaging Using One Time Pad*. International Journal of Advances in Soft Computing & Its Applications 6.1 (2014).

[6] Gunter Ollmann, *Securing Against the "Threat" of Instant Messaging*, Network Security Journal, vol. 2004, no. 3, pp. 8-11, March 2004.

[7] Carey, Charles A., and Bruce A. Robinson. *Method and system for instant messaging across cellular networks and a public data network*, U.S. Patent No. 6,714,793. 30 Mar. 2004.

[8] Vernam, Gilbert S. *Cipher printing telegraph systems: For secret wire and radio telegraphic communications*. AIEE, Journal of the 45.2 (1926): 109-115.

[9] E. Rescorla, *SSL and TLS: Designing and Building Secure Systems*, Addison Wesley, 2001.

[10] Saint-Andre, Peter. *Extensible messaging and presence protocol (XMPP): Core*, (2011).

[11] Rosenberg, Jonathan. *SIMPLE made simple: An overview of the IETF specifications for instant messaging and presence using the session initiation protocol (SIP).* (2013).

[12] Yang, Chung-Huang, et al. *Design and implementation of a secure instant messaging service based on elliptic-curve cryptography,* Journal of Computers 18.4 (2008): 31-38.

[13] S. Egan and others. *Pidgin.* http://www.pidgin.im/.Accessed January 2016.

[14] Microsoft Corp. *C# Language.* http://www.microsoft.com/.Accessed March 2016.

[15] Xamarin Corp. *Xamarin: Mobile App Development & App Creation Software.* http://www.Xamarin.com/.Accessed March 2016.