Crytographic Instant Messaging

by Nelso Malgro

Submission date: 28-Feb-2019 08:37AM (UTC+0800)
Submission ID: 1085040072
File name: ographic_Instant_Messaging_System_for_Secured_Data_Exchange.docx (1,012.06K)
Word count: 13522
Character count: 75108

Chapter 1

`INTRODUCTION

Background of the Study

Data exchange or data communication is a key process of providing accurate and timely information to senior leaders, top management, and decision makers. It involves the use of email, instant messaging, video conference, chat, online forums, and other applications that transmit data from sender to receiver. Many organizations and modern companies prefer to use instant messaging rather than email for simpler and more user-friendly collaboration among staff and management. It also exchanges data almost at an instant, allowing two-way communication in near real-time.

Instant messaging applications like Facebook messenger, SKYPE, and Yahoo Messenger provide easy, fast, and cost-effective means of text, voice, and video communication. It has become the most preferred application to communicate with peers, families, and among professionals from various industries and sectors. Businesses are now beginning to embrace IM for their office communication. Although it is less formal than a face to face meeting, it can bridge the gap of needing to collaborate and communicate with each other. Private companies, schools, organizations, and government agencies use IM to support or provide redundancy with their other existing mode of communication.

The military recognizes the necessity to develop or acquire a secure, reliable, and near real-time data communication tool for effective exchange of information and to provide the high-command a means to collate information needed for decision-making. In

today's network-centric operation, Instant Messaging (IM) is an essential information communication tool for rapid delivery of messages and reports, and for situational awareness.

The US military has long been using IM for its military operations. In fact, Cummings (2004, p. 654) stated that "it was the primary means of communication between navy ships during Operation Iraqi Freedom in 2003". The Philippine Navy (PN), a branch of service in the Armed Forces of the Philippines (AFP), sought to acquire similar tool but is constrained with limited resources and lack of communication infrastructure to securely extend its network to mobile units thereby utilizing the un-secure Internet for its data communication.

The PN, with its prevailing need for a secure data communication like an instant messaging, started to adopt several free and open-source IM but unsatisfied with the minimal encryption it provides. Text messages can be intercepted, decoded, and read in clear text as it travels across the network from sender to receiver. The pervasiveness of these applications makes it vulnerable from various attacks especially sniffing and hijacking. Some vendors, however, promise end-to-end or asynchronous encryption with their expensive commercial IM. They may either use open-standard advanced encryption protocols or they are using closed and proprietary protocols. The benefit it provides may surpass that of the free versions, however, the vendor obscures any vulnerabilities of this application from the end-users and it will not be long until a sophisticated hacker can crack its encryption protocol through a purchase of similar application.

Among the popular IMs available online are unencrypted by default such as Facebook Messenger, Pidgin, Google Allo, and Viber. IMs are designed with usability

rather than security in mind. Almost all freeware IM software does not have encryption capabilities.

Few news articles can be found as an evidence of this vulnerability, many IM vendors bargain this flaw as leverage for the users to purchase the licensed version. Eavesdropping and sniffing IM conversation is common to most hackers and cybersecurity professionals especially if they are within the Local Area Network (LAN).

Data security is of utmost importance in any organization whether private or government sector. The effect of confidential messages being sniffed, leaked, or compromised can be devastating to any individuals, businesses or institutions such as the Armed Forces. Quality of software should be measured not just by the functionality and performance but also the security it guarantees. It has to assure the confidentiality, integrity, authenticity, and availability of the data it processes. Thus, adoption of an encryption mechanism is imperative to secure the data at-rest or in-transit.

The growing importance of security for an IM or for any type of application cannot be overemphasized. All communication application (i.e Instant Messenger) must be integrated with an encryption algorithm that is proven unbreakable and resistant to any mathematical cracking techniques. With an encryption component present in an IM, users are assured of an optimum level of security the software provides.

Data hiding or Steganography has been used to hide secret messages into an image. While it is commonly used as a standalone application, it has never been implemented into an IM. Integrating both encryption and steganography is a research worth exploring.

Secure IM is sought to benefit primarily the PN, and the AFP. Other sectors such as business offices, government agencies, schools, industries, banks, and other institutions can also adopt this software for office communication, report collection, and information dissemination.

Objectives of the Study

This study aims to develop an Instant Messaging desktop application called Cryptographic Instant Messaging (CIM) system.

Specifically, the study aims to:

1. Design CIM system with the following features and characteristics:
    a. Multi-layer encryption by combining Advance Encryption Standard (AES) algorithm and Hidden In Plain Sight (HIPS) image hiding technique;
    b. Secure One-to-one and room chat using AES algorithm;
    c. Secure File transmission using the combination of AES 256-bit encryption and HIPS;
    d. Secure login authentication through password encryption;
    e. Transmit recorded voice message;
    f. Self-delete or automatic deletion of secret message;
    g. Use of cipher key exchange methods:
        1) System Generated Key (SGK); and
        2) Manual Key Input (MKI).
    h. Encrypted username and password on the database
    i. The users decrypt the file at the time of their choosing.

2. Develop the desktop application using Visual C# .net, as designed
3. Test the functionality and security of the system across different Penetration Testing software.
4. Test the portability of the system across different multiple versions of MS Windows operating system.
5. Evaluate the quality of the software using ISO 25010 with criteria for Functional Suitability, Performance Efficiency, Usability, Reliability, Security, Maintainability, Portability, and Compatibility.

Scope and Limitations

The development of Cryptographic IM (CIM) system is inspired by the PN's desire to develop a data communication tool that provides secure data exchange and rapid delivery of reports and messages.

The system has the essential features of a typical IM such as one-to-one chat, room chat, and file transfer. It is designed and developed in modules or components and provides the platform that allows other library/API/modules to be integrated into.

The CIM integrates AES 256-bit algorithm and HIPS hiding algorithm by Engr. Mardonio M Agustin Jr. to provide multi-layered encryption. It has two (2) methods of key exchange: SGK and MKI – the former is the default setting whereby randomly generated keys are pushed on every IM Client every week while the latter allows user to type in the keys manually for a guaranteed privacy among peers.

It is developed using the client-server architecture whereby the server relays all messages among connected clients. It uses Microsoft Visual C# for coding. It runs on any computers with MS Windows operating system.

Chapter 2

CONCEPTUAL FRAMEWORK

This chapter presents the review of previous literatures and studies related to Instant Messaging applications. It explores the earlier developments, previous research, and existing technologies leading to the development of a prototype cryptographic IM (CIM). It includes the operational definition of terms used, and the conceptual framework to visually synthesize all materials and techniques in achieving the desired objective.

Review of Related Literature and Studies

Historical Development of Instant Messaging (IM)

The advent of Instant Messenger like Facebook Messenger, Skype, WhatsApp, Telegram, and other popular IM revolutionizes the way people communicate with each other. But its impressive features would not be possible without the earlier breakthrough of their predecessors.

Desjardin (2016), in his article "The Evolution of Instant Messaging", narrates the written accounts that brought the IM and how it grows prominence in the new era. Instant Messaging became known in the 1990's but decades ago Massachusetts Institute of Technology (MIT) already demonstrated an information sharing program called Compatible Time-Sharing System (CTSS). In 1988, Jarkko Oikarinen developed the first chat system that allows sending messages to users through their computers (Oikarinen, 1988). His Internet Relay Chat (IRC) paved the way for other IM players like AOL IM, ICQ, and Microsoft IRC to battle for share in the IM market. Yahoo messenger

dominates in 1998 with its intuitive designs and feature-rich IM environment. Engel (2014) recounts how IM grow prominence in the 2000s. This was the golden age of instant messenger. MSN Messenger introduced the concept of sharing photos and included the PC-to-PC and PC-to-phone audio capabilities. Skype Messenger peaked in popularity in 2003 as it showcased Audio and Video chat capabilities. Facebook chat was first introduced in 2008 and accelerated its popularity in 2014 when it decouples from the Facebook App.

Application of IM

Military Institution

The US military has long been using IM for its military operations. In fact, Cummings (2005, p. 654) stated:

"It was the primary means of communication between navy ships during Operation Iraqi Freedom in 2003".

In his journal "The Need for Command and Control Instant Message Adaptive Interfaces: Lessons Learned from Tactical Tomahawk Human-in-the-Loop Simulations" Cummings affirms the obvious advantage in real-time communication embodied in IM. It allows rapid response to inquiries, multiple conversations with various units, and access to previous conversations and orders.

In a thesis presentation entitled "Applicability of Instant Messaging in the Military Command and Control Systems"; Vermaja (n.d.) explains that IM can offer many advantages but security issues in implementing it is a problem. Furthermore, commercial off-the-shelf (COTS) IM does not satisfy wholly the needs of a Command and Control (C2) communication. Customized applications are needed to make the

communication easy and practical. Tailor-fit applications with specific platforms are needed in order to build automated systems.

Businesses

Various studies were already conducted on the maturity of IM to be used for business communication. Results clarify how businessmen and their clients choose this technology for daily transactions and communication. It was learned that people, generally, uses IM for a variety of substantive business purposes, and their peers include a diversity of co-workers, clients, managers, and other business personalities (Muller, Raven, Kogan, Millen, & Carey, 2003).

Gotrunk (2017), a US company that offers secure and reliable VOIP services, shares their experience in IM. According to them, businesses are now beginning to embrace IM for their office communication. IM is virtually real-time communication like a telephone call. It is a text-based conversation like email but in an instant. It proves advantageous for collaboration among employees from different locations. In fact, some employees even use it for private communication within and outside the office.

Schools

People of college age constitute a significant and considerable number or population among IM users. Flanagin (2005), in his study, found out that IM has become a central communication tool within the college population. IM is being used for a variety of personal needs and school requirements and utilized it heavily relative to other forms of communication like email and telephone. Moreover, evidence suggests that college students are satisfied with simultaneously processing multiple conversations and that IM might displace email in many consequential ways.

Security Issues of IM

The efficiency and usability of this application come with a major drawback on security. Hackers and cybersecurity professionals found several vulnerabilities with this system. These problems make most organizations to re-think implementing and adapting IM. Moreover, the pervasiveness of this application makes it vulnerable to various attacks especially sniffing and hijacking. Symantec (n.d.), a leader in cyber security applications, exposes some of the threats and vulnerabilities found in popular IM. IMs are designed with usability rather than security in mind. Almost all freeware IM software do not have encryption capabilities. They enumerated some of the threats and vulnerabilities found in many IM. These are eavesdropping, sniffing, man-in-the-middle attack, session hijacking, and malicious software injection such as Worms and Trojans.

Sweigart, C. (n.d.) explains that most IM System configures the server to act as a mediator and relay all message traffic; some, on the other hand, use peer-to-peer communication. The latter offers better security than the former. In peer-to-peer, the client sends the message directly to the recipient allowing data traffic to flow without the IM server's intervention.

SANS (2003) Institute, a popular training center for information security published a whitepaper to make people understand the risk associated with every IM. According to them, the lack of encryption protocols on free IM means that IM session conducted in a public network is like an open book to the entire Internet community. Classified information and trade secrets if communicated over free IM on a public network can become public knowledge within seconds.

Few news articles can be found as an evidence of this vulnerability, many IM vendors bargain this flaw as leverage for the users to purchase the licensed version. However, major system flaw can cause a commotion among IT professionals and result can be devastating if not fixed. On January 5, 2002, *The Washington Post* reported that a hacking group known as w00w00 found a hole in a software code of a popular IM program. Matthew Conover, the founder of the group, reported the software flaw to AOL. Failing to recognize his effort, the impatient hacker published the flaw to the community and an explosion of national media attention wreaks AOL (Oldenburg, 2002).

A recent news article from ZDNet news, Condon (2017) reported a hacking incident of a popular IM, HipChat. The incident affected a server in the HipChat where messages and content in chat rooms may have been accessed. In response, HipChat posted a security notice to the public about a vulnerability of the third-party library used by their IM.

On an Internet post, Marcel Ackerman explains the need to encrypt the Instant Messenger. Ackerman provides recommendations on existing IM technology that are being used today and posted list of popular IM that is equipped with security protocols. According to his blog, people are fooled by companies that use security as a marketing buzzword. They will show the fancy user interface but do not provide details of the encryption or unwilling to disclose the algorithm for scrutiny (Ackermann, 2018).

IM Architecture

Client-Server Architecture

The development of this prototype IM software largely depends on previous literary studies on client-server architecture and socket programming techniques. A thesis

worth reviewing is the work of Tim Van Lokven (2011) entitled Review and Comparison of Instant Messaging Protocols. Lokven examines the architecture of the three (3) commonly used protocols by popular IM. These are Microsoft Notification (MSN), Open Source for Communication in Real-time (OSCAR), and the Extensible Messaging and Presence Protocol (XMPP) protocol. The efficiency of these protocols is evaluated based on a predefined set of criteria including its security. In his findings, Lokven stated that steps should be taken to encrypt messages sent to the servers.

IM can also be developed using a web-based or browser-based architecture. I3M is an instant messaging and chat system developed by Hans Schmid for web-based reporting and collaboration. The I3M application server stores centrally all user accounts and provides communication to clients in the form of Web service. The server is written in Visual C# and runs on the Windows 2000 platform (Schmid, n.d.).

Yulianto (2015), in his proposed IM architecture model, outlines the use of XMPP protocol and OpenFire web server as a cost-effective means to develop an Android-based IM application. The web server is used to implement a web application and can communicate via XMPP with chat server and MySQL as its database. Yulianto concluded that his proposed architect is running well and ninety-three percent (93%) of the respondents are willing to install his IM to their terminals.

In a presentation during Hackathon – a convention for ethical hackers and cybersecurity professionals, Eugene Letuchy reveals the chat architecture of Facebook Messenger. Facebook uses Erlang Programming Language to develop their chat application. Client browsers send messages using Ajax scripting; web server receives this message and creates a channel for every client. Erlang powers the backend of the web

servers aggregating and distributing all messages. Eventually, Facebook rewrites its chat application codes to C++ for stability and scalability (Letuchy, 2007).

Linan Zheng (2005) explains the possibility of developing an integrated communication platform through the combination of "Presence" technology and instant messaging. The "Presence" service allows users to know the status of his peers and can decide what communication channel to use whether SMS, Video Call, or leave a voice message. Zheng also examines the use of the Jabber protocol, a fork from XMPP, and outlines the flow of messages using this protocol.

Socket Programming

A socket whether Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) is one of the fundamental and significant protocols in developing a client-server application. TCP or UDP Sockets enable software applications to communicate with each other in a network. Singh (2014) elucidated the process of TCP connections and how it creates sockets. Application X connects to Application Y using the command "tcp.connect" while Application Y waits for any application to connect using the "tcp.accept" command. Once a connection is established a socket is created for that connection. Application X then uses the socket to stream or transmit data through the command "socket.send" while on the other end; application Y receives the packet using "socket.receive". Sockets act as an endpoint for a two-way communication link between two applications running on a network.

Socket programming is the backbone of almost all client-server applications like web services, email services, telnet services, IM, and other applications who implement similar TCP and UDP protocols. By learning socket programming, programmers can

develop network application that allows data transmission from one host to another. Similarly, Microsoft developed MSN or Microsoft Network for its proprietary IM and America Online (AOL) developed Open System for Communication in Realtime (OSCAR) for AIM application. Abba (2013) developed a LAN Chat Messenger (LCM) application using socket programming in JAVA language. His LCM created a solution to the organization's internal communication problem.

In a journal from sciencedirect.com, Eddie Law and Roy Leung presented a unified layered architecture for any TCP/IP networks. The model, known as Active Network Socket Programming (ANSP), allows an active application to be written once and run on multiple environments. This unified programming interface layer is light-weighted and can be easily deployed (Eddie Law & Leung, 2003) .

Encryption Algorithm

Advanced Encryption Standard (AES)

Data security is of utmost importance in any organization whether private or government sector as stated in previous section of this study. Adoption of an encryption mechanism is an effective, if not the most effective, means to achieve data security. Encryption converts plain messages to a format unreadable by any person. It mathematically scrambles the data as it travels from the sender to the receiver.

Padate (2005), in his journal, emphasized the importance of data security and how Advanced Encryption Standard (AES) improves the security of any data communication application. AES which was originally called Rijndael is an open-standard or open-book specification of an encryption algorithm published by the U.S. National Institute of

Standards and Technology (NIST) in 2001. AES was selected among fifteen (15) encryption algorithms that were subjected to rigid test and evaluation. AES was designed to be resistant against all known brute-force attacks. It is a symmetric block cipher that comes in three key lengths: 128, 192, and 256 bit. The basic implementation of AES is simple mathematical and logical table lookup operations. This symmetric key algorithm is fully open to the public for evaluation and scrutiny, and to ensure transparent analysis and validation of the design.

This was further proved by Mahajan (2013) in his journal entitled Study of Encryption Algorithms AES, DES, and RSA for Security. Based on his test result, Mahajan concluded that the AES algorithm is faster than RSA which consumes longest encryption time. Further, decryption of AES is better than other algorithms.

Steganography or Hiding Technique

Steganography is a type of cryptography which conceals or hides valuable information inside a benign object. Previous research on cryptography at Technological University of the Philippines (TUP) offers a viable option for encryption to complement the AES. Agustin (2005), with his Hidden in Plain Sight thesis, discusses the method of hiding plain files into an image file. Contents or bytes of the file fill in the white spaces of the image file using Least Significant Bit (LSB) insertion technique. These blank spaces are almost unnoticeable to the human eye. Such method of encryption is commonly called steganography. While there are several methods in steganography Mardonio provided a unique method in securing the password. The plain password is translated into a numerical value using shift, modulo and XOR bitwise operations.

Steganography technique is further implemented in the thesis of Torres (2012) entitled Development of FileGuard: Steganography Software. Torres simplified steganography technique by creating modules or algorithms for extraction, encryption, and compression. FileGuard software passed the test and evaluation of quality software based on its functionality, Usability, Reliability, Efficiency, Portability, and Maintainability.

Integration of Cryptography into Software Applications

Desktop Applications (Standalone)

Establishing a secure data transmission requires implementation of cryptographic in the communication channel. Both sender and receiver must agree upon a set of security protocols for them to communicate and understand each other (Del Pozo & Iturralde, 2015).

Remijan (2014) briefly discussed the procedure in integrating encryption to a Visual C# application using the readily available Class component for AES. According to him, the code for AES encryption and decryption is not very long since this is an open standard published by NIST. However, figuring out the right method, variable, and configuration is the tricky part. Remijan uploaded his AES project in the Github.com for anyone to evaluate and use it.

Email Application (Client-Server)

Recent developments in email communication introduce the adoption of asynchronous or certificate-based encryption into the email. A study from Jakarta, Indonesia proposes a secure method of e-mail communication using a hybrid encryption.

It combines hash function, symmetric encryption and asymmetric encryption (Mantoro & Zakariya, 2012). The study is further enhanced by Criseldo Calinawan in his research "Hybrid Encryption Algorithm Implementation on Electronic Mail Service". The researcher implemented the Rivest–Shamir–Adleman (RSA), a certificate-based encryption, and his own encryption algorithm written in PHP language into a webmail client. The result shows faster execution time, reliability, and improved security (Calinawan, 2015).

Communication Application (Client-Server)

Few studies are conducted to demonstrate the integration of an open standard encryption algorithm like the AES into a communication application (i.e. Instant Messenger). Moh Heng Huong successfully integrates the AES 128-bit encryption into his communication application. According to him, AES 128 encryption provides a more secure communication than other encryption algorithms. The encrypted data is unbreakable until today using this algorithm (Huong, 2014) .

IM Application (Client-Server)

Integration of the chosen encryption algorithm into the IM presents a huge challenge. By studying and adopting previous development of network application incorporating encryption algorithm significantly reduces the amount of time hard-coding these algorithms. Krishna (2011) published a journal of his method of integrating AES encryption into file transfer application using secure shell protocol. It uses a "Toolkit" to handle the process of encryption and password authentication. Toolkits are a readily available software component that can be integrated into the main application.

Google patented a secure instant messaging called the AOL Instant Messaging. The secure IM adopts a certificate-based encryption. A certificate authority (CA) issues security certificates to users and use its public key to encrypt messages and files for the user. The user then sends his certificate to an IM server which distributes the user's certificate to other users. Users encrypt messages and files using a symmetric encryption algorithm and the recipient's certificate (US 9,509,681 B2, 2016) .

Software

Programming Software

Visual C sharp (#) dotNET (.NET) is one of the programming languages bundled in Visual Studio.NET. It is designed to develop a wide range of business-driven and enterprise applications. It provides simple but high-performance object-oriented language in a .NET environment. It also enables developers to build applications for an array of needs and requirements including deployable form-based applications, centralized transaction, operational and logistical automation, and network socket communication. (Price & Gunderloy, n.d.).

C++ and C# are the ideal programming languages for socket application development. The C# language aims to leverage the power of C++ with programmer-friendly environment and ease of Visual Basic. The C# language is suitable for a firm and soft real-time applications (Lutz & Laplante, 2003). Samia Tapur introduced online laboratory courses that allow students to learn the basic and advanced functions of network sockets using C language. The students can better understand the behavior,

properties, and functions of these sockets during the connection of the client to the server through simulated exercises (Talpur, 2016) .

Operating System Software

Microsoft Windows Operating Systems (MS Windows OS) is a family of computer operating systems developed by Microsoft since 1985. Almost eighty percent (80%) of computers nowadays runs with Windows OS. Popular versions of MS Windows OS are Window XP, Vista, 7, and 10 (Carpenter, 2012). MS Windows is highly modular. Each function is managed by components of the operating system. The protected subsystems and its applications are structured using the client-server model, which is a common model for distributed computing like the computer networks. A client application requests a service by sending a message. The message is then routed to the appropriate server. The server performs the requested operation and returns the results back to the client through another message (Stallings, 2004) .

Vulnerability Assessment and Penetration Testing (VAPT) Software

Kali Linux is a variant of Linux Operating System. It is an opensource operating system specifically designed to provide a vulnerability assessment (VA) and penetration testing (Pentest) platform for ethical hackers and cybersecurity professionals in order to test the security of a particular network or information system. (Pérez & Binders, 2003).

Penetration testing is the method of finding vulnerabilities of an application. Global statistics show that more than 70% of the applications have vulnerabilities which can be exploited by any hacker. It needs to be secured and the best way to secure the application is to try hacking into it using a specifically designed tool like the Kali Linux (Sarmah & Hachan, 2018).

Hackers are categorized as a White hat, Black Hat, or Gray Hat. White Hat hackers are ethical hackers with some certifications. This type of hacker uses tools like Kali Linux for implementing hacking and breaking into systems purposely to identify vulnerabilities and repair them (Gawhale, 2016).

Database Software

Microsoft Structured Query Language Server (MS SQL) is an object-oriented and relational database developed by Microsoft. It basically stores and retrieves data required by any connecting applications. It can be accessed through standalone applications or online connectivity using TCP/IP network. SQL is the standard language used to communicate and manipulate the data stored in this database (Sarka, Radevojivec, & Durkin, 2018).

Microsoft Access (MS Access) is a standalone database management system developed by Microsoft to provide the easy, fast, and deployable database. It is the database of choice for programmers and non-programmers to deploy a short-to-medium size application and does not require storage of bulk data. (Hennig, Bradly, Linson, Purvis, & Spaulding, 2010).

Network Infrastructure

Computer Network is a series of nodes and terminals interconnected by communication links in order to exchange data, voice and video packets. Popular network devices are routers, hubs, switches, modems, and any terminals equipped with Network Interface Cards (NIC). These devices are joined together by network links like network cables (e.g. UTP, Serial, Fiber Optics) and wireless connectivity (e.g. Wireless Access

Points, 3G, LTE). TCP/IP is the commonly used protocol or algorithm for network devices to communicate and interact with each other (Peterson & Davie, 2012). The interconnectivity of vast computer networks across the globe is called the Internet.

The Internet protocol is the networking model and set of communications protocols used for computer networks or the internet. The Transmission Control Protocol (TCP) and the Internet Protocol (IP) are the first networking protocols defined in a public standard. It was originally known as the DoD model because the development was funded by DARPA, an agency of the United States Department of Defense. TCP/IP provides end-to-end connection specifying how data are transmitted, broken (packetized), addressed, and routed at the destination (Nath & Uddin, 2015) .

Evaluation Standards

The International Organization for Standardization (ISO) 25010 standard is a framework that evaluates the quality characteristic of software. It is divided into eight (8) criteria or characteristic. These are Functional Suitability, Performance Efficiency, Usability, Reliability, Security, Maintainability, Portability, and Compatibility (Galin, 2018).

Functional Suitability defines the completeness, correctness, and appropriateness of various features and functionalities of the software. This test determines whether components of the system meet and satisfy the requirements of the stakeholders.

Performance Efficiency gauges the response time, utilization of resources, and the capacity of the software during runtime under normal and stress condition.

Usability determines the accessibility, operability, user interface friendliness, and appropriateness to the needs of the stakeholder. The degree to which software is easy to operate is evaluated.

Reliability measures the degree of availability, fault tolerance, and recoverability of the software under normal and stress operation.

Security evaluates the ability of software to protect the data processed in terms of confidentiality, integrity, authenticity, accountability, and non-repudiation. Applied encryption and steganography are evaluated for this purpose.

Maintainability determines the effectiveness and efficiency of the software during modification on any of its components.

Portability refers to the degree of effectiveness of the software to adapt various hardware and software environment after installation or during runtime. It is tested on most common Windows platforms such as Windows 7, Vista, and 10

Compatibility measures the interoperability of the software when integrated with various systems, toolkits, Software Development Kits (SDK), Application Programming Interfaces (API), and libraries. Components of the software are evaluated to determine if it performs effectively with no detrimental impact on other identified software while sharing a common environment.

CONCEPTUAL MODEL OF THE STUDY

Based on the objectives of this research and the review of previous studies, proven designs, applied theories, and other dependable concepts and literatures, the following model was outlined to conceptualize the development of the CIM

.

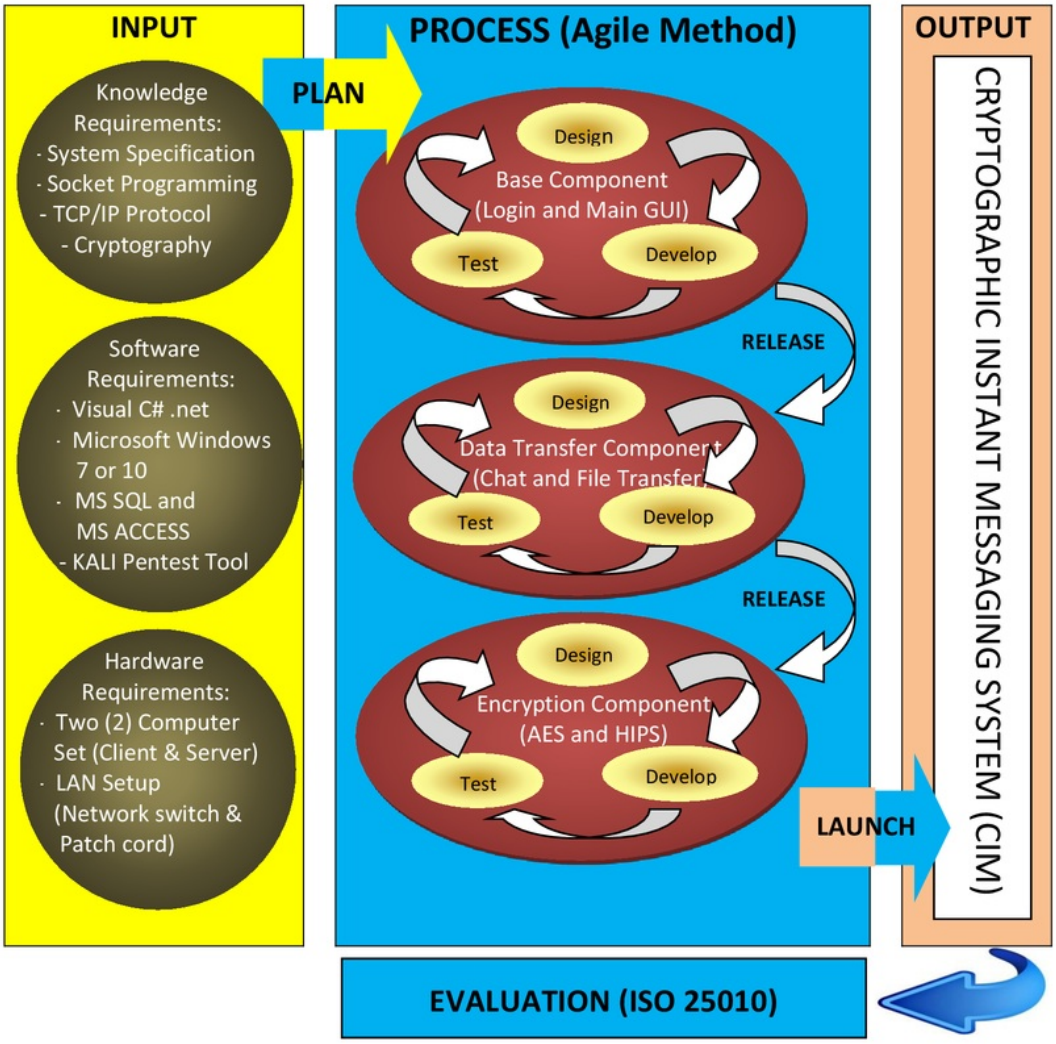

Figure 1. Conceptual Model of the CIM System

Figure 1 from the previous page shows the conceptual model that visualizes the processes involved in order to develop the CIM System. Hardware, software, and knowledge requirement, including the system specifications, shall be the input that guides and fuels the entire processes.

Planning should be an interrelated activity between input phase and process phase to identify the necessary requirements needed for the successful completion of every module. This is done to address the issue of needing to acquire additional software or libraries that emerged only during the actual coding.

In the Process phase, Agile methodology shall be adopted for the coding of the software so that one component can be released or launched even without the other unfinished components making it ideal for evolutionary or continuously improved software. **Agile** refers to an effective software development technique that is time-boxed and iterative. It works by breaking the project into components of functionalities, prioritizing them, and then continuously delivering them in cycles called iterations. The entire system shall be divided into three (3) components – the base, data transfer, and encryption. Data transfer component begins its development process upon release of the base component.

Similar with the planning activity, launching or deployment of the entire system shall be an interrelated activity between process phase and output phase. This is to allow immediate debugging on the errors that triggered during the deployment. Evaluation is done after the operational testing of the CIM. Modification of the system can still be done after the evaluation to satisfy the requirements of the end users.

Operational Definition of Terms

This section provides the definition of terms which are observable, identifiable and repeatable.

The following terms are defined operationally to better understand the project study:

API or Application Programming interface is a set of methods or subroutine definitions that allow communication among various components.

ARP Spoofing or Address Resolution Protocol spoofing is a technique used by an attacker to send a spoofed address to the ARP table of every device in a local area network. It reroutes all transmission from the victim computer to the attacker.

Asymmetric Encryption is also known as Public Key Cryptography which uses two (2) keys to encrypt a plain text. A public key is made available to anyone who wants to use it while the private key is kept a secret to the owner.

Closed Source refers to proprietary software whose source code is not shared with the public for anyone to look at.

Command and Control refer to the exercise of the authority of a designated officer over a group of personnel under his/her command in the accomplishment of task or mission.

Cryptography refers to the practice or study of techniques which mathematically scrambles the plain message into a form unreadable to any person except for the intended recipient.

Internet Protocol (IP) refers to the dominant communication protocol used by a Local Area Network, Wide Area Network, or the Internet.

Libraries or Dynamic Link Libraries (DLL) are Microsoft implementation of modules or subroutines (similar to API) that can be shared or reuse across multiple applications.

MAC Flooding or Media Access Control flooding is a technique employed to compromise the security of network switches by flooding the MAC table causing it to broadcast the data out to all ports.

MD5 Hash Value or Message Digest Fifth version is a one-way cryptographic hash function with 512 bit block size. It is an encryption algorithm that maps data to a bit string of a fixed size called a hash and is designed to be irreversible.

Man-in-the-middle attack is an attack wherein the hacker resides in between two communicating parties in an attempt to sniff, relay, or alter the transmission of data.

NET Framework is a Microsoft model or structure of interrelated class libraries which provides interoperability across several programming languages. It allows reuse or sharing of libraries among different windows applications.

Open Source refers to a type of software in which the source code is released under a general public license. This license grants other users to study, modify, and redistribute the software to anyone and for any purpose.

Packet refers to the chunk or unit of data that is routed over a TCP/IP network or the Internet.

Session Hijacking a.k.a. cookie hijacking is a technique used by exploiting a valid network session in order to gain unauthorized access to network services or computer systems.

Socket refers to the network stream or channel which allows transmission of packets between hosts in a network.

Socket Programming refers to computer programming practices which deal with socket communication between nodes or hosts in a TCP/IP based network.

Software Developers Kit (SDK) is a set of software development tools, API, libraries, and documentation to enrich an application with specific functionalities and for a specific platform that is deemed complex to develop. It reduces the amount of time solving programming problems.

Symmetric Encryption is the simplest method of encryption which involves only one key to encipher and decipher a message.

Transmission Control Protocol (TCP) is one of the main protocols which complement the Internet Protocol (IP) in handling the packets over the internet. It provides reliable, connection-oriented, error-checked, and ordered delivery of data between hosts.

User Datagram Protocol (UDP) is one of the core member protocols which complement the Internet Protocol (IP) in handling the packets over the internet. UDP is connectionless and unreliable but transmits faster than TCP.

VOIP or Voice over Internet Protocol is a hardware or software technology that allows voice communication over a TCP/IP Network or the Internet.

Vulnerability Assessment and Penetration Testing (VAPT) is a systematic way of finding vulnerabilities of an application, computers, or network. It involves the employment of intrusive techniques in order to bring the flaws to the surface with the intent to repair those.

Chapter 3

RESEARCH METHODOLOGY

This chapter presents the project design and project development to provide blueprint of the entire system architecture. Operational procedure and method of testing and evaluation are explained to measure its effectiveness based on predefined parameters.

Project Design

The Cryptographic Instant Messaging (CIM) system is Instant Messaging (IM) application software that provides encrypted chat communication and file transfer for secure and reliable exchange of information.

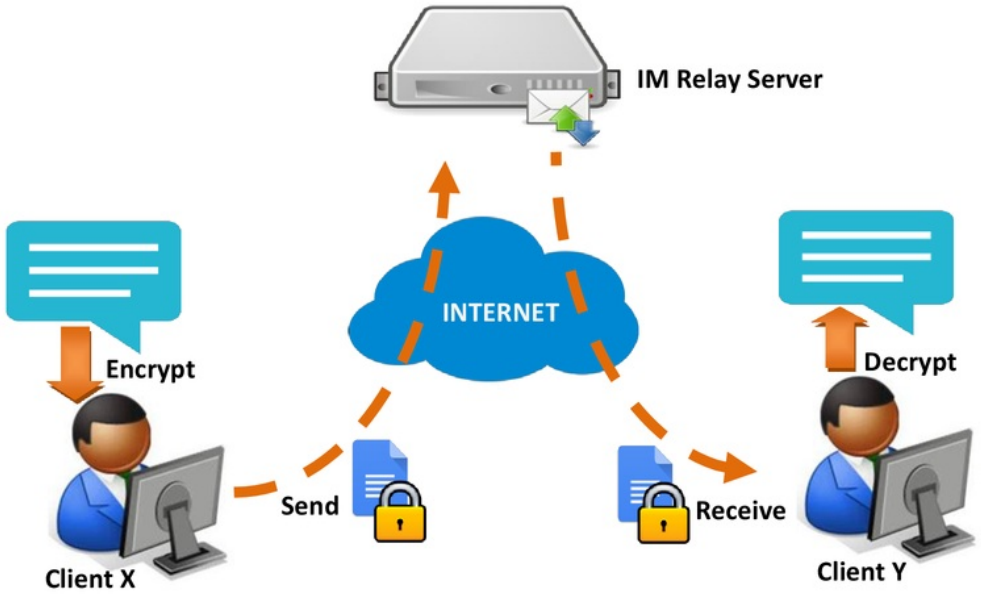

Figure 2. Interface Diagram of CIM

Operational Concept and Dataflow

Figure 2 from the previous page illustrates the fundamental structure of the system and flow of data from an end-user's perspective. Client X initiates a chat conversation with Client Y using the CIM client software. During the chat conversation, the system automatically encrypts the text message prior sending it to the IM Relay Server. The server now searches for the intended recipient from its list of client sockets. When the recipient's socket is available, the server transmits the encrypted text message to Client Y (recipient) through its socket. The server manages all clients' sockets. It handles authentication of all incoming clients prior acceptance of the socket. IM Relay Server maintains a minimal database using MS SQL Server, SQL Express, or MS Access for accounts and activity logs. As soon as the text message arrived at Client Y (recipient), the software then decrypts the text using the system-generated key as the default. Unreadable message or randomized characters indicates unmatched passkey. Both Client X (sender) and Client Y (recipient) must encode manually the correct or agreed passkey to decode their conversation. The manual encoding of a passkey is done to deny the system administrator or any authority from decoding the message during the conduct of any special investigation. System generated keys are stored in the server specifically for this purpose. It is strongly suggested that this manual key should be transmitted through another mode of transmission like SMS, Email, or voice call.

A similar procedure can be done during sending and receiving files. Client X starts up the File Transfer window and browse for the file. By default, CIM uses Advanced Encryption Standard (AES) 256-bit to encrypt and decrypt text messages and files. The sender can choose to double encrypt the file by hiding it in an image file using

steganography or Hidden in Plain Sight (HIPS) algorithm. A checkbox or option button is provided for this feature. By checking both AES and HIPS, the system first encrypts the file using AES and hides the resulting file into an image. The CIM then transmits the image file to the IM relay server. The server searches for the intended recipient from its list of client sockets. When the recipient's socket is available, the server relays the file or message to Client Y (recipient). As soon as the encrypted file or image file arrived, the user can now browse for the file and choose to open it with the same application. The encrypted file is extracted from the image file using the HIPS function for file extraction. After the successful extraction, the CIM now tries to decrypt the extracted file using the system-generated key. If unsuccessful, it asks for the passkey which must be encoded by the receiving user to convert the encrypted file back to its original form. Manual encoding of passkey happens when the sender chooses to encode a different key other than the default system-generated key.

Network Architecture

To visualize its deployment from a technical perspective, the network diagram is drawn for this purpose. In figure 3 of the next page, the CIM primarily utilizes the Local Area Network (LAN) for inter-office instant messaging and collaboration. Client X (desktop computer) transmits messages to Client Y (laptop computer) through the adjacent relay server within the same LAN.

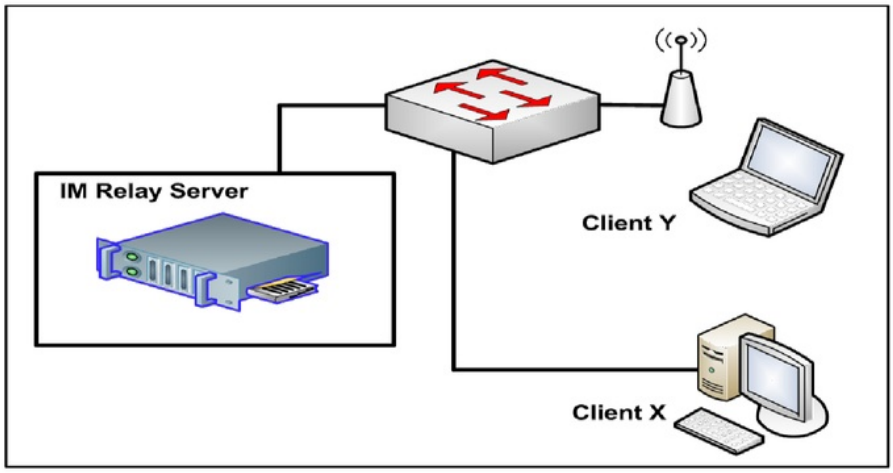

Figure 3. Local Area Network (LAN) Setup of CIM

The CIM is also applicable to clients outside the office and anywhere in the world. For a Wide Area Network (WAN) setup, the Internet is utilized. The IM Relay Server is assigned with a global or public IP address to allow connection from clients outside of its network. This allows interconnectivity among regional offices, company branches, or widely dispersed units through this system.

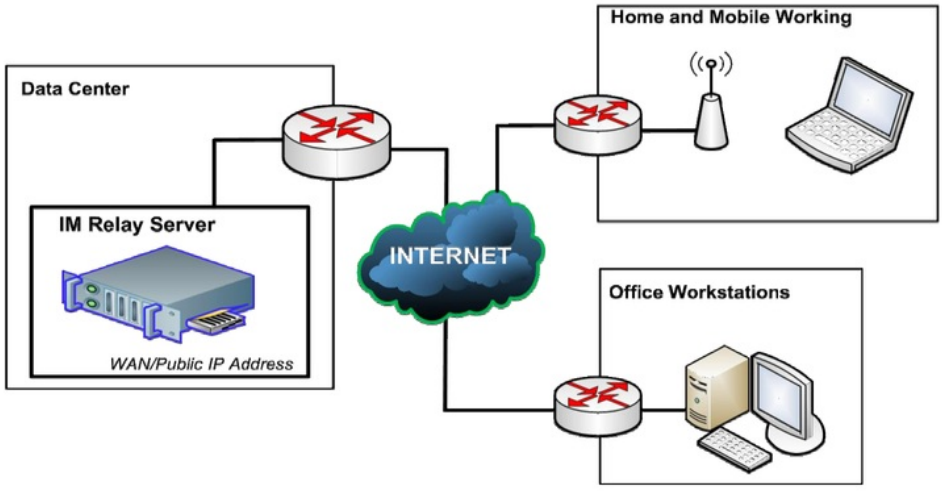

Figure 4. Wide Area Network (WAN) Setup of CIM

Users who bring their work at home can now securely communicate and send files to the main office through the CIM. Firewalls and Intrusion Prevention System (IPS) provide optional security reinforcement at the network layer between the CIM relay server and CIM client. Servers are ideally housed in a Data Center for a large-scale deployment.

System Analysis and Design

The interaction between the system and users is explained below through the Use Case Diagram where the administrator and the end users are the actors.

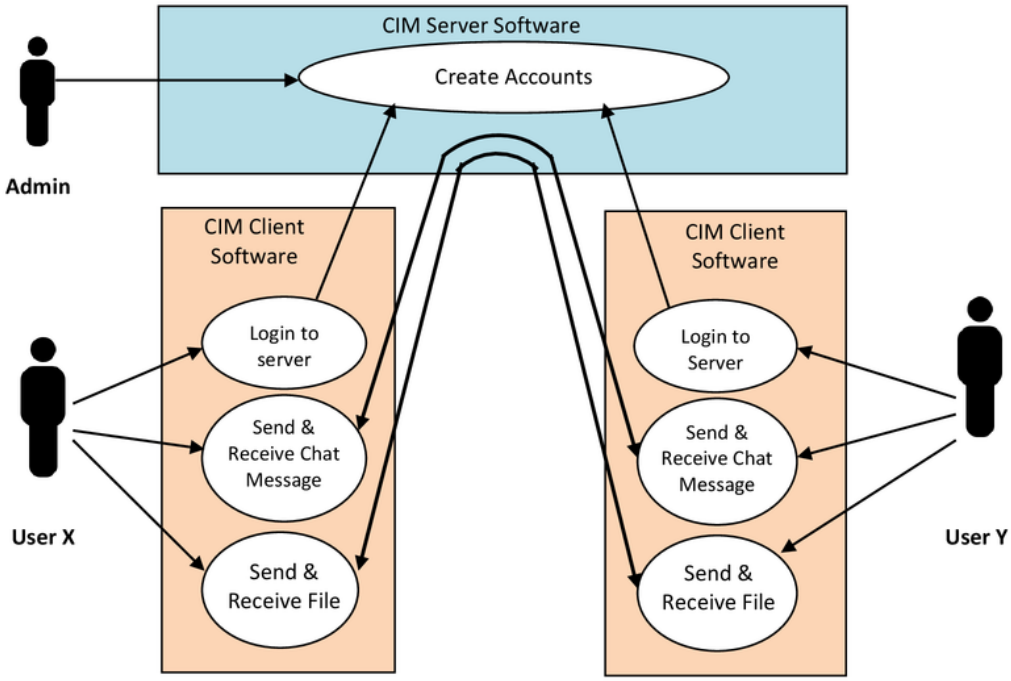

Figure 5. Use Case Diagram of the CIM

Above use case diagram depicts the procedure in utilizing the CIM. The system administrator creates the accounts needed for end users to log in or authenticate. After a

successful login, users can now send and receive chat messages as well as transmit files with each other via the relay server.

Encryption and decryption of chat messages and files are done automatically by the system. The CIM adapts AES 256-bit by default as its first layer encryption. Users may choose not to use the system-generated key and encode a different manual key known to both the sender and recipient. This is done to deny the system administrator or any authority from decoding the files during any special investigation.

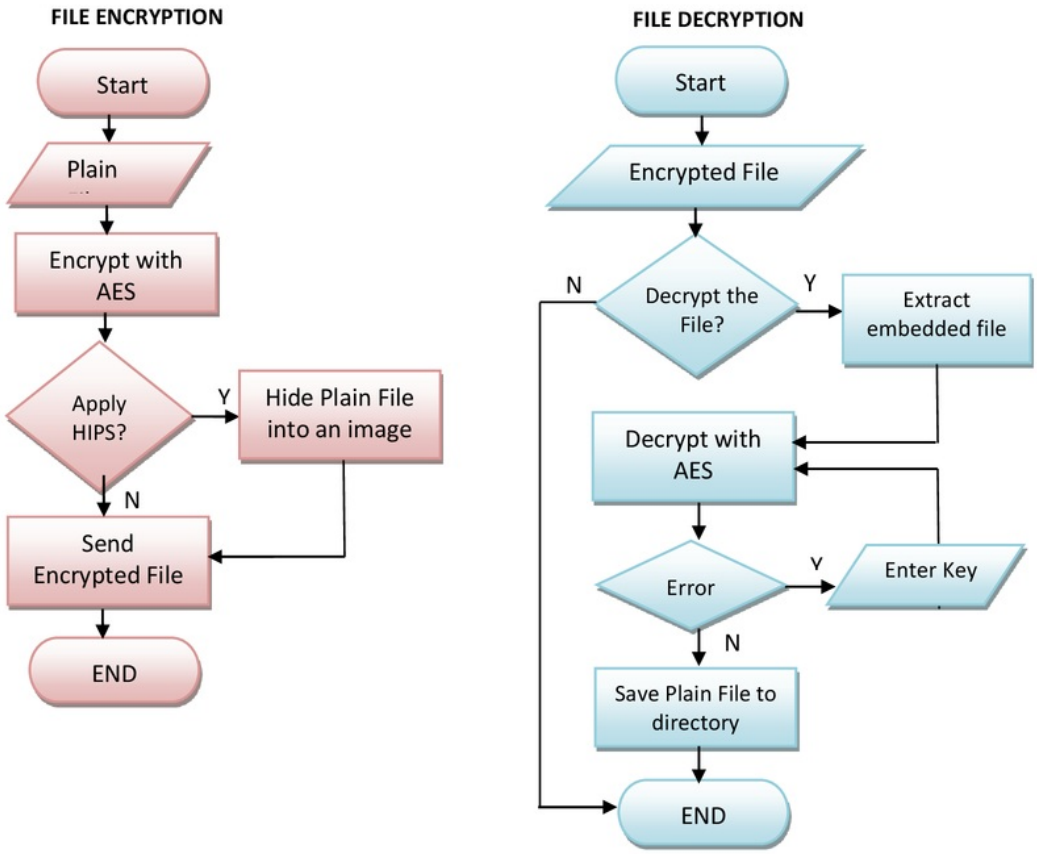

Figure 6. Flowchart for File Encryption and Decryption

Figure 6 of the previous page explains the process of encrypting and decrypting files. Users may choose to double encrypt the file using steganography hiding technique. To do this, end users clicks the HIPS radio button to hide the data file in an image prior to the transmission. When the recipient received the encrypted file, the user can now browse for the file and open it with same CIM application. The system extracts the embedded file using HIPS algorithm and then decode it using the AES decrypt function.

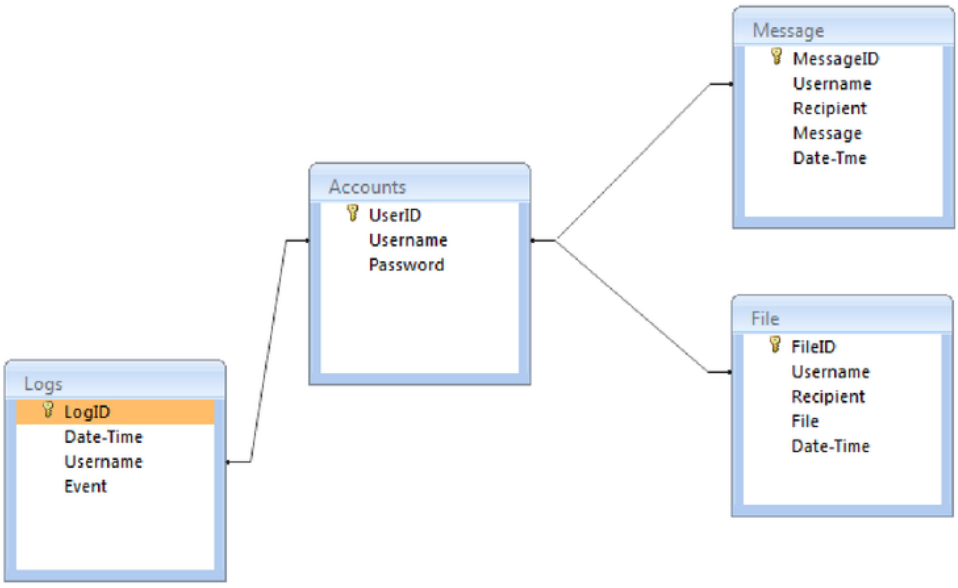

Figure 7. Table Relationship Diagram of the CIM Database

Above figure shows the different tables to be used for handling client authentication, chat, and file transmission. Tables are connected or joined by Username so that a query and updating can be made through this field. Account table is used for authentication, Message table is for storing chat conversation, and File table is used to record files being transmitted. Apart from these primary tables is the Logs table which is

used to monitor and review all activities by the client IM. This is also important for the security of the CIM server and its connecting clients.

Project Development

The development of CIM adopted agile methodology so that various modules or components including other third-party modules (e.g. DLL, API, and SDK) can be integrated into the system; vis-a-vis one component can be released or launched even without the other unfinished components making it ideal for evolutionary or continuously improved software.

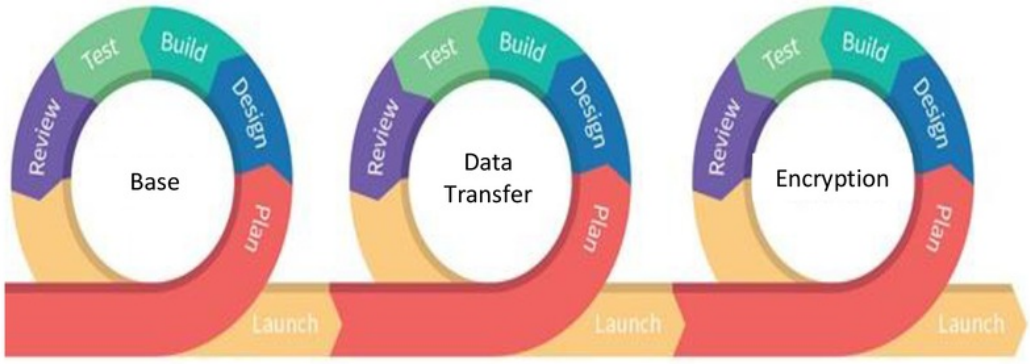

Figure 8. Software Development Life Cycle using Agile Methodology
https://medium.com/

Above figure elucidates the process of SDLC as earlier presented in the conceptual model of this study. The CIM was divided into three (3) components - the base, data or file transfer, and encryption component. These components served as sprints or iterative work activities needed to complete the entire system.

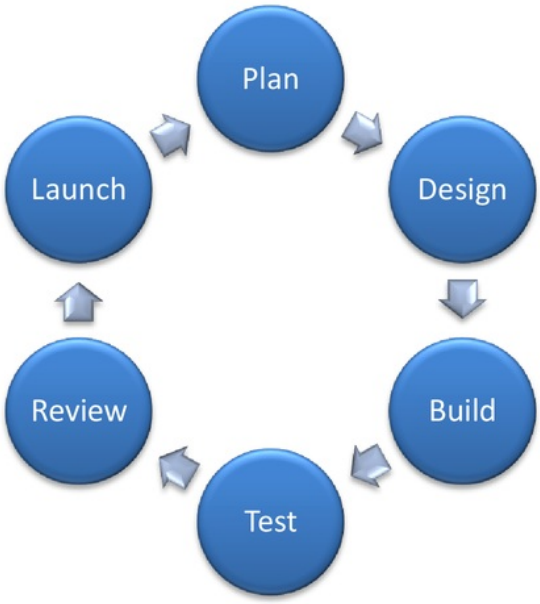

Figure 9. Phases of Software Development

Planning Phase

Planning involved identifying requirements, determining system specifications, and establishing work breakdown schedules. Every component of the system was born out from the requirements and specifications provided by the stakeholders or end users. Hardware, software, and skill requirements were identified to ensure completion of the project. The base component of the CIM adopted open architecture technique to allow various components to be integrated.

Designing Phase

CIM system architecture was mapped out to identify technical specifications of every component. These specifications include forms, screens, reports, databases, modules, and routines. Use case diagram, network diagram, and other related figures

previously drawn in this chapter were examined to serve as blueprints for the coding of the system.

Building Phase

This is the coding or development phase of the project. The software was written using Visual C# computer programming language. The practice of secure coding was also observed during this phase in order to address any application-specific vulnerability thereby reducing the possibility of getting exploited by a determined hacker.

Testing Phase

Each component underwent functional and portability testing prior to its release. All forms, functions, classes, and procedures, were tested to ensure an error-free application. It uses standard testing methodology, test case forms, and test logs. Results of these test cases were documented and formed part of this research.

Reviewing Phase

The system or components of this system was reviewed to ensure conformity with the project specification. In this phase, the CIM was demonstrated to fifteen (15) respondents for evaluation. The respondents were composed of five (5) IT Experts, five (5) cybersecurity experts, and five (5) end users. They assessed the CIM based on ISO 25010 criteria for software quality and security.

Launching Phase

In this phase, the component is now ready for deployment. The completion of the base component marked the start of the next development cycle for the other components. The CIM, with its base component, was a working IM and can already be deployed in a production environment even without the other components. The completed data transfer

and encryption components were then integrated into the base component for a complete CIM system as required in this research.

Operation and Testing Procedure

Operation Procedure

For CIM System Administrator, the following procedure was conducted:

1. Configured the IP Address of the server.
2. Started the CIM Server software.
3. Added users through the "Add User" button.
4. Informed the end user for their username and password as well as the IP Address of the server they will use to connect.

For the CIM user X (sender), the following procedure was conducted:

1. Checked for internet or network connectivity.
2. Opened CIM Client software.
3. Entered the username and password provided by the administrator as well as the IP address of the server.
4. After a successful login, the main window was displayed showing the hierarchical list of users.
5. Started conversation by selecting the user and clicked the "Chat" button.
6. Sent files by browsing and selecting the desired encryption algorithm or a combination of both the AES 256-bit and HIPS hiding technique.

For CIM user Y (recipient), the following procedure was conducted:

1. Checked for internet or network connectivity.

2. Opened CIM Client software.
3. Entered the username and password provided by the administrator as well as the IP address of the server.
4. After a successful login, the main window was displayed showing the hierarchical list of users; offline messages were also displayed.
5. A file was received; the user browsed for the file and extracted the embedded file from the image.
6. The system decrypted the file using the system-generated key.
7. Started conversation by selecting the user and clicked the "Chat" button.

Testing Procedure:

The CIM was subjected to functional and portability testing to ensure an error-free application across all versions of MS Windows operating system. The testing was conducted among computer programmers and system administrators to immediately resolve or debug any fault of the system. Test cases were documented and form part as supplementary material or evidence of this research. The table on the next page enumerates the procedure or steps conducted during the testing of CIM through various scenarios.

Table 1

Testing Procedure of the CIM

Scenario	Steps to be undertaken
Client Login	1. Attempted to login with erroneous entries 2. Login with correct entries 3. Verified the password was encrypted using the Wireshark tool of KALI Linux
Send and Receive Chat Message	1. Sent a long and random chat message to determine a possibility of a crash. 2. Sent readable chat message and verified the accuracy of the message as it reached the recipient. 3. Measured the time it took for the message to reached the recipient 4. Verified the text was encrypted during transmission using Wireshark tool of KALI Linux
Send and Receive File	1. Sent a file and measured the time it took for the file to reach the recipient. 2. Sent large files and checked the possibility of crashes. 3. Verified HIPS hiding technique by locating the image file containing the embedded data file. 4. Verified AES encryption is applied by opening the encrypted file if it is unreadable.

Test case forms and test incident logs were accomplished or filled up to document the alpha testing of the CIM. These forms included description which explains the purpose of the test scenarios; while the pre-requisite establishes the required state of the system prior to the execution. The previously enumerated steps were then reflected in the test execution steps of this form. The table on the next page is the test case form that was used during the conduct of alpha testing.

Table 2

Test Case Form

Test Scenario ID		**Test Case ID**	
Test Case Description		**Test Priority**	
Pre-Requisite		**Issue Severity**	
Test Execution Steps:			

Step Nr	Action	Inputs	Expected Output	Actual Output	Test OS	Test Result	Test Comments
1							
2							

Ten (10) test cases or scenarios were executed for this research. Results of the test cases were recorded, including whether the condition was "pass" or "fail". A pass condition means that the actual results meet the expected outcome. On the other hand, a fail means that the system is executed with errors or the actual results do not meet the expected outcome. Test incident log was provided to document un-forecasted errors that occurred during the exploration of the system including the previously documented test scenarios. The table below is the test incident log format that was used during the alpha testing.

Table 3

Test Incident Log

No.	Test Scenarios	Error Description	Case Ref	Severity	Priority	Screenshot
1						
2						
3						

The classification table below was used to determine the severity of error resulted during the test execution. Its corresponding description served as bases of classifying such severity. The severity and priority level was then reflected into the test case form.

Table 4

Severity Classification and Priority Level of Errors

Severity	Description	Priority
Critical (Severity 1)	These issues are showstoppers and productivity is severely hindered. No workarounds exist and require immediate resolution.	High
Major (Severity 2)	These are issues that have an impact on productivity but can have a workaround.	Medium
Minor (Severity 3)	These are issues that do not prevent productivity to be carried over. All issues have a workaround.	Low

Test results were summarized to determine if the system meets the required percentage score for a system worthy to be deployed. The following criteria were used to determine if the system or components of the system can be released:

1. All of the test cases are executed.
2. "Pass" result should be no less than eighty percent (80%) of the executed test cases.
3. There must be no critical or Severity 1 issue unresolved.

The table on the next page was used to document the summary of all test cases.

Table 5

Test Execution Summary

Test Execution	Expected Results
Total no. of test cases	
No. of test cases executed	
% executed	
No. of test cases passed	
% passed	
No. of test cases failed	
% failed	
No. of test cases not executed	
% not executed	

Portability Testing

The system ideally runs on any Microsoft Windows platform except for the mobile version of Windows OS since this is a desktop application. Portability testing was conducted to check the CIM for varying errors across all versions of Windows operating systems. This test was also necessary to determine other required libraries or software in order to successfully execute the CIM, The system was developed using .NET framework in which previous versions of Windows OS are not installed by default. The table on the next page was used to document the conduct of portability testing of CIM. Errors that occurred during this testing were also recorded in the test incident log.

Table 6

Portability Testing Summary

Operating System (Version)	Prerequisites	Expected Results
MS Windows OS		

Vital to the conduct of alpha testing is the security test. It determines the validity of encryption implemented. The table below was used to test the encryption of the CIM against the three (3) commonly used Sniffing or Penetration Testing software – Wireshark, Cain&Abel, and Ettercap as part of Kali Linux.

Table 7

Security Testing Summary

Instances	Original Message (Source)	Result Message (Destination)	Penetration Testing Software		
			Wireshark	Cain&Abel	Ettercap (Kali)
Send Chat Message					

Evaluation Procedure

Software evaluation was conducted to ensure conformity to the required specification and that the system exhibits characteristics that meet the criteria of a quality and secure application software. This study adopted the ISO 25010 standard for evaluating the software quality and validates the security of CIM. The following are the essential criteria of this standard which were used as the evaluation metrics for this study: Functional Suitability, Performance Efficiency, Usability, Reliability, Security, Maintainability, Portability, and Compatibility.

The following activities were conducted to deduce an accurate and justified evaluation of CIM.

1. Five (5) IT experts, Five (5) cybersecurity professionals, and Five (5) end users were invited as respondents/evaluators for this research.
2. The evaluation metrics and the Likert Scale as shown in the table below were presented and discussed to the evaluators; the Likert scale was necessary to measure the respondent's opinion for the system during the conduct of evaluation.

Table 8

Likert Scale

Numerical Rating	Descriptive Rating
5	Excellent
4	Very Good
3	Good
2	Fair
1	Poor

3. Walkthrough of the system were conducted and evaluators were encouraged to explore the CIM through hands-on experience.
4. The interception of CIM communication between the client and the server were demonstrated through the use of KALI Linux in order to validate the security of the system.
5. The evaluators were given sufficient time to rate and evaluate the system.
6. The evaluation instrument accomplished were collected and tabulated.
7. The score mean was computed to deduce a numerical result of the evaluation.
8. The result was then interpreted using the equivalent descriptive rating as shown in the table below.

Table 9

Rating Scale for Interpreting the Evaluation Result

Mean Rating Scale	Descriptive Evaluation
4.51- 5.00	Excellent
3.51-4.50	Very Good
2.51-3.50	Good
1.51-2.50	Fair
1.00-1.50	Poor

Every criterion or characteristic specified in the ISO 25010 were computed with a mean score and its equivalent descriptive rating. A poor rating means unacceptability of the system to be released in the production environment and should be subjected for re-

development. A fair rating only equates to ordinary software which contains only the basic feature and does not offer any advantage over other IM. A good rating indicates a satisfying performance of the software and exhibit features less found in popular IMs. Very good rating demonstrates outstanding performance of the software and showcased features advantageous over other IM. Excellent rating bespeaks superb performance and harbors unparalleled features deemed to be the IM of choice if released in the market.

Chapter 4

RESULTS AND DISCUSSION

This chapter presents the final output of the system and the result of the evaluation. Project description, project structure, and the capabilities and limitations are briefly discussed to elucidate the finished project. The result of the evaluation and its interpretation was also presented in this chapter.

Project Description

The Cryptographic IM (CIM) system is an instant messaging application that allows data communication among peers. It provides essential features for office collaboration such as one-to-one chat, group chat, file transfer, and others. Multi-layered encryption is integrated into the CIM to provide greater security. Advanced Encryption Standard (AES) and the Hidden in Plain Sight (HIPS) algorithm encrypts the data as it traverses across the network. The system was developed in modules or components making it agile in any type of deployment. Meaning, the CIM can be deployed even without the other modules if it is not required for a particular deployment. There are three (3) modules developed, the base, file transfer, and the encryption module. The base component provides the platform for the two (2) components and other future components to be integrated into. These future components can also be a third-party DLL, API, or SDK.

Project Structure

The CIM was developed using the Microsoft Visual C# 2017. It runs on both MS Windows 7 and Windows 10 platform and requires dot.Net 4.6 framework. Microsoft Access database engine was included during the installation to allow local storage of user lists and other settings. The figure below shows the final structure of the system.

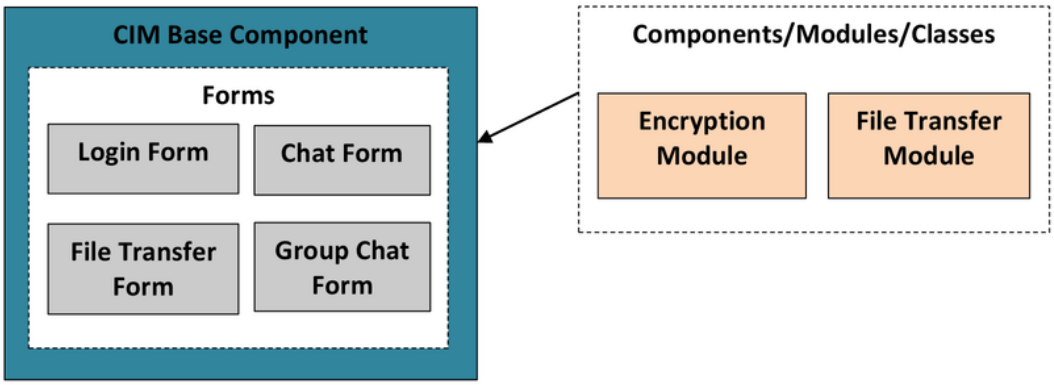

Figure 10. Final Structure of the CIM

The CIM is composed of Three (3) modules or components, namely:

1. Base Component – This is the most important component of the CIM. It provides the platform in which other components are integrated into. It comprised the forms, references, properties, libraries, and the main program needed to run the system. The following are the forms created for the base component:
    a. Login Form – it provides the interface needed to authenticate the user to the CIM server. The password entered by the user is automatically encrypted upon clicking the login button.

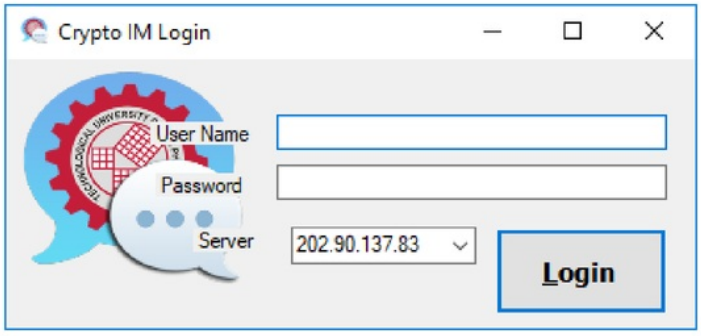

Figure 11. Login Form of the CIM

b. Main Window Form – this is the main window that displays after a successful login. Groups and users are populated in the tree view. The green user icon indicates an online user while the gray icon indicates an offline user.

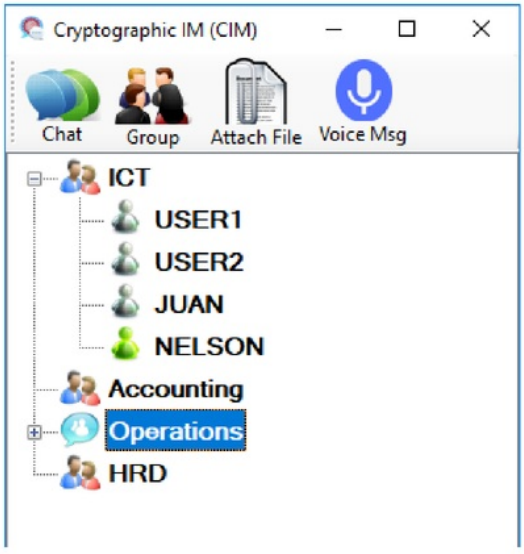

Figure 12. Main Window of the CIM after a Successful Login

c. Chat Form – this form allows the user to interact with other peers in the CIM through chat or text-based conversation. The chat communication is encrypted by default using AES 256-bit encryption with passkey

generated by the system. The user has the option to manually assign a key for a more private conversation. All system generated keys are changed weekly and stored on the server. This allows the organization to decode the previous conversation during an internal investigation as the need arises. Having keys manually entered denies the possibility of having someone to decode the conversation other than the two (2) communicating peers.

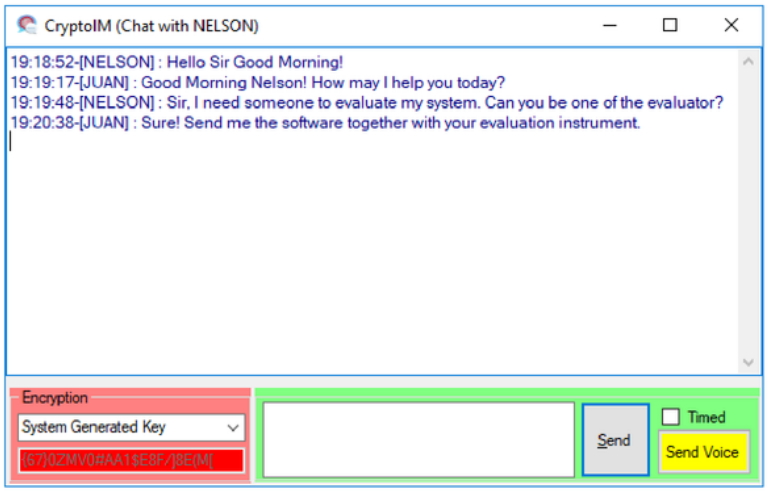

Figure 13. One-to-One Chat

d. Group Chat Form – this allows users to broadcast text conversation among members in the room. Like any similar room chat, any users can join the room and the moderator has the privilege to kick any members he or she chooses. Similar encryption is applied in this window.

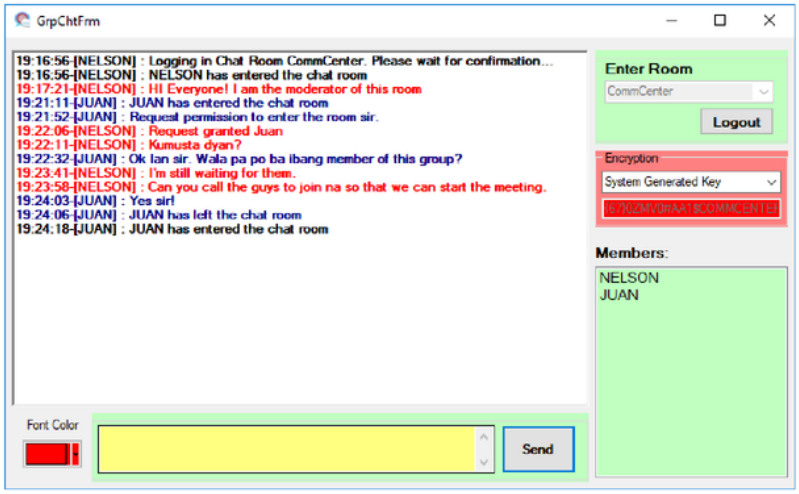

Figure 14. Group Chat Form

e. File Transfer Form – this form requires File Transfer module/class to function. It allows transmission of documents, images, audio, video, applications, and other files from one user to another. When applying encryption to the file, encryption module/class is called. The user has the option to use the HIPS for multi-layered encryption of the file.

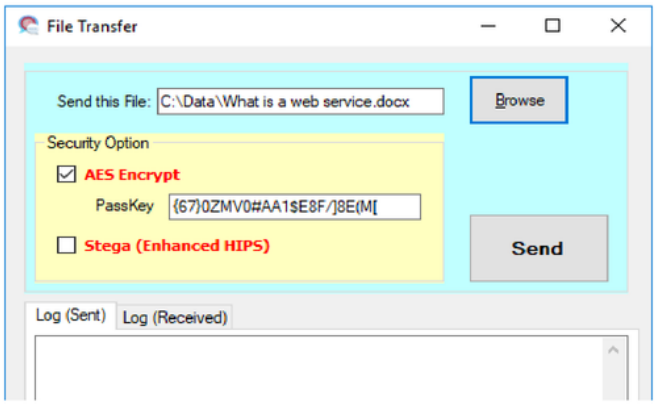

Figure 15. File Transfer Form

2. Encryption Component/ Module – this is the component that provides AES encryption and HIPS hiding feature to the CIM. This module can be deployed separately through Dynamic Link Library (DLL) and can be integrated to any applications. The figure below shows the resulting files after encrypted with AES and embedded in an image file (.cim file extension).

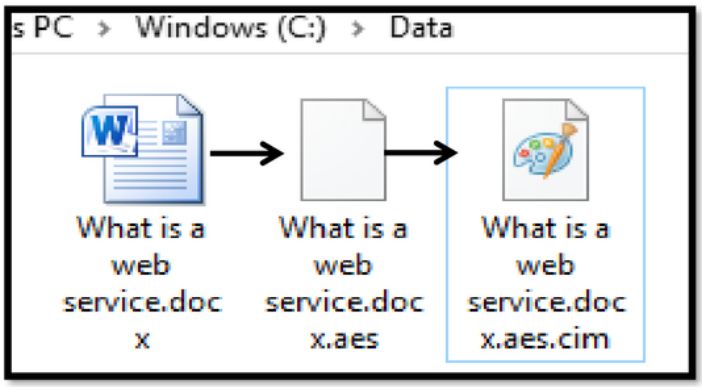

Figure 16. CIM Automatic Encryption and Decryption

a. AES Encryption – AES as explained in chapter two is an open standard algorithm which can be freely acquired and integrated into an application for strong and robust encryption. The CIM imports the AES library provided in the Microsoft .NET API and wrote specific functions applicable to the system. These four (4) functions are Encrypt String, Decrypt String, Encrypt File, and Decrypt File. The figure on the next page shows the class library for encryption.

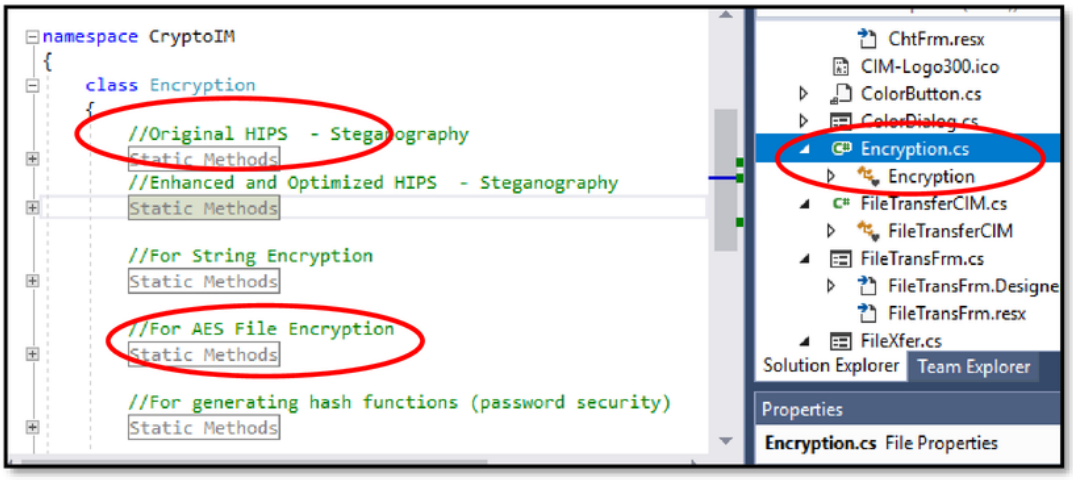

Figure 17. Encryption Class of the CIM

b. Hidden In Plain Sight (HIPS) Encryption or Hiding Technique- CIM integrates the HIPS source code from Engr. Agustin. Unfortunately, some functions of the code are already obsolete and some controls are no longer supported in the new Visual Studio. Few revisions are necessary to properly integrate the code into the CIM. These modifications improved the HIPS while retaining vital algorithms as formulated in Engr. Agustin's thesis especially the least significant bit (LSB) replacement and the conversion of the password into a numerical value to be used as a random number (seed value) for picking the pixel positions. The following are some of the enhancement of the HIPS:

1) Auto-resize of the image file – the original HIPS requires the user to import an image file that must be larger than the embedded file or else an error will return. This method was enhanced with a function that auto-resize an image file to a length twice the size of the embedded file.

2) Direct image file insertion –the original HIPS loads the image file into the Image Control of the Form before hiding the embedded file and then saving the image from the control into another filename. This technique is a slow process; thus, it is replaced with a function that allows direct insertion of the data byte into an object that encapsulates the image file.

3) Simplified Formula – insertion of the data byte is done by dividing the image file into several bytes and recursively replacing its 4-bits LSB with every 4-bit data byte of the embedded file.

3. File Transfer Component/ Module – this component allows file transmission from one user to another. File transmission is done by establishing TCP socket connection from an online user to the server and then the data byte is transmitted through this socket. The detailed explanation of socket communication is presented in the previous chapters. The Base component calls file transfer module/class when the File Transfer form is opened. Upon clicking the send button, the CIM calls the encryption module and encrypts the file; then the encrypted file is divided into several bytes for transmission. These bytes are sent into streams so that when it reaches the server it is re-assembled back to its original form.

The CIM is further improved to satisfy some of the evaluators' requirements. These added competitive features are:

1. It allows sending of voice messages;
2. It provides option for self-delete or self-destruct messages;

3. Users are organized into groups or departments;
4. Multi-colored font for room chat;
5. Server logs for auditing;
6. System-generated key is changed weekly;
7. Encrypted username and password on the database; and
8. Users can decrypt the file at the time of their choosing.

Project Test Results

A total of ten (10) test cases of the three (3) scenarios were conducted for the operational and functional testing of the CIM. This activity was documented in the Test Incident Log (refer to the appendix). The table below provides the summary of the test results based on the consolidated test cases that was executed during the alpha testing of the system.

Table 10

Summary of Test Execution

Test Execution	Results
Total no. of test cases	**10**
No. of test cases executed	**10**
% executed	**100%**
No. of test cases passed	**8**
% passed	**80%**
No. of test cases failed	**2**
% failed	**20%**
No. of test cases not executed	**0**
% not executed	**0%**

The result of the functional testing yields eighty percent (80%) or eight (8) test cases successfully run; however, two (2) of the 10 test cases functionally run but generate an error. These failed test cases are minor in severity and are already resolved as of this writing. Documents of these test cases are included in this research for perusal (refer to the appendix).

Portability testing is conducted to check the compatibility of the CIM across different versions of Microsoft Windows operating system. The table below provides the result of this test.

Table 11

Result of the Portability Testing

Microsoft Windows	Prerequisites	Results
Windows 10		• CIM client runs and login window appears • Successfully login to the CIM Server
Windows 8.1	.net 4.0	• CIM client runs and login window appears • Successfully login to the CIM Server
Windows 7	.net 4.0	• CIM client runs and login window appears • Successfully login to the CIM Server
Windows Vista	.net 4.0	• CIM client runs and login window appears • Successfully login to the CIM Server
Windows XP		• CIM Client not running
Windows 2012	.net 4.0	• CIM Server runs and the main window appears
Windows 2016		• CIM Server runs and the main window appears

Among the different versions of the Microsoft Windows operating system, only Windows XP is not supported. This is an obsolete operating system and known to have a number of vulnerabilities; thus, it is not recommended to modify the CIM and pursue installing in this platform.

Vital to the conduct of alpha testing is the security test. It determines the validity of encryption in the CIM. The table below reveals consistent result of a secured data after the system is tested across different Penetration Testing software.

Table 12

Result of the Security Testing

Instances	Original Message (Source)	Result Message (Destination)	Penetration Testing Software		
			Wireshark	Cain&Abel	Ettercap (Kali)
User Login	User: user1 Pass:USER@123	24C9E15E52AFC47C225B757E7BEE1F9D E2B31C4CF92DD40E079B9B8BA414F9BD	Encrypted with MD5 Hash	Encrypted with MD5 Hash	Encrypted with MD5 Hash
Send Chat Message	Hello I am user1	cMYEhI3nugUR4+f97OL1+CjAFe8M7/1vPKrKvBiDvR0=	Encrypted with AES-256 bit	Encrypted with AES-256 bit	Encrypted with AES-256 bit
Transmit Document (AES+HIPS)	Smarthouse.docx	Smarthouse.jpg	Image w/ hidden file	Image w/ hidden file	Image w/ hidden file
Decrypt Image File	Smarthouse.jpg	Smarthouse.aes Smarthouse.docx	Encrypted with AES-256 bit	Encrypted with AES-256 bit	Encrypted with AES-256 bit

There were four (4) instances conducted to test the resulting data or message as it traverses across the network. The penetration testing software sniffed the transmission from clientX (source) to clientY (destination) using ARP Spoofing technique. Sniffing can also be done using a port-mirrored network switches. During login the resulting message appears to be scrambled characters when sniffed. All penetration testing software detected the resulting message as MD5 hash values. Another instance is when clientX sends a chat message to clientY, the message appears to be scrambled characters in the chat window of clientY (the key is changed to prevent decrypting the message). All penetration testing software detected the resulting message as AES-256 bit encryption. During file transmission, clientX sends a plain document to clientY and browsed a JPEG file to hide the document. The jpeg image containing the encrypted file was received by clientY. All penetration testing softwares detected the data as an image file. When clientY manually decrypts the image file, it extracted sequentially the AES file and the plain document (docx) file.

Project Capabilities and Limitations

The CIM provides the essential features of a typical instant messaging application; however, if continually improved, this application can compete with any popular IM today.

Capabilities

The following are the capabilities of the CIM:

1. Secure login through password encryption.
2. Secure one-to-one text communication or chat.

3. Secure Group or Room Chat.
4. File transfer with default AES 256-bit encryption.
5. File Transfer with reinforced encryption using HIPS hiding algorithm.
6. Send recorded voice message.
7. Self-delete of the text message or recorded voice message after several seconds.
8. Encrypted username and password on the database
9. Users can decrypt the file at the time of their choosing

Limitations

The following are the limitations of the CIM:

1. It only runs on Microsoft Windows 10, 7, 8, and Vista.
2. It requires a Microsoft Net 4.6 framework and Access Database Engine.
3. It does not implement Peer-to-Peer (P2P) technology which means data communication is through the server.
4. Only the CIM can decode the image file that was earlier encoded using the enhanced HIPS. The old HIPS application cannot decode it even if it is the same algorithm that was used. As explained in the above section, there is a need to modify the original codes of HIPS due to obsolescence. For example, the function PSet (*Picture.PSet (r, c), RGB(clrR, clrG, clrB)*) in the legacy HIPS is no longer supported in Visual Studio.Net 2017 thereby replacing it with WriteByte function (*fw.WriteByte((byte)*).

Project Evaluation

Five (5) IT professionals, five (5) cybersecurity experts, and five (5) end users evaluated the CIM. The system is evaluated using the eight characteristics/criteria of ISO 25010. These criteria are further broken down into sub-characteristics as discussed in the previous chapter.

The table below shows the average mean score of the CIM in terms of Functional Suitability:

Table 13

Average Mean Score in terms of Functional Suitability

Criteria	Mean Score			Average Mean	Descriptive Ratings
Functional Suitability	IT Experts	Cyber Security Experts	End Users		
Functional completeness - functions cover all the specific tasks and objectives.	3.80	4.00	3.60	**3.80**	Very Good
Functional correctness – the system provides the correct results with a certain degree of precision.	4.00	3.80	3.80	**3.87**	Very Good
Functional appropriateness - functions of the system allow the accomplishment of specific tasks and objectives.	4.00	4.20	3.60	**3.93**	Very Good
Average Mean	**3.93**	**4.00**	**3.67**	**3.87**	Very Good

As presented in the above table, the CIM got an average mean of 3.87 with "Very Good" descriptive rating. The system definitely provides the needed functionality for data communication and office collaboration.

The table below shows the average Mean score of the CIM in terms of Security:

Table 14

Average Mean Score in terms of Security

Criteria	Mean Score			Average Mean	Descriptive Ratings
Security	IT Experts	Cyber Security Experts	End Users		
Confidentiality – system ensures that data are accessible only to authorized users.	4.20	4.60	4.20	**4.33**	Very Good
Integrity - system prevents unauthorized modification of data.	4.00	4.40	4.00	**4.13**	Very Good
Non-repudiation – events and actions can be proven to have taken place.	4.00	4.20	4.00	**4.07**	Very Good
Accountability - actions of a user can be traced uniquely to the entity.	4.00	4.60	4.20	**4.27**	Very Good
Average Mean	**4.05**	**4.45**	**4.10**	**4.20**	Very Good

As presented in the above table, the CIM got an average mean of 4.20 with "Very Good" descriptive rating. The system is tested and evaluated using the Kali Linux – a popular tool for conducting a vulnerability assessment and penetration testing. The validated encryption of the CIM transcends other IMs in terms of security.

The table on the next page shows the average Mean score of the CIM in terms of Performance Efficiency:

Table 15

Average Mean Score in terms of Performance Efficiency

Criteria	Mean Score			Average Mean	Descriptive Ratings
Performance Efficiency	IT Experts	Cyber Security Experts	End Users		
Time behavior - running time and rates of the system meet the requirements.	4.00	3.80	4.00	**3.93**	Very Good
Resource utilization - amount and types of resources used by the system meet the requirements.	4.00	4.00	4.20	**4.07**	Very Good
Capacity – maximum limit of the system meets the requirements.	4.20	3.80	4.00	**4.00**	Very Good
Average Mean	**4.07**	**3.87**	**4.07**	**4.00**	Very Good

As presented in the above table, the CIM got an average mean of 4.0 with "Very Good" descriptive rating. The system runs in optimum performance under varying conditions.

The table on the next page shows the average Mean score of the CIM in terms of Usability:

Table 16

Average Mean Score in terms of Usability

Criteria	Mean Score			Average Mean	Descriptive Ratings
Usability	IT Experts	Cyber Security Experts	End Users		
Appropriateness recognisability – The system is appropriate to the users' needs.	4.00	4.00	3.80	**3.93**	Very Good
Learnability – the system can be used by specific users to achieve certain goals of learning.	4.20	3.60	4.00	**3.93**	Very Good
Operability - a system has characteristics that make it easy to control and operate.	4.20	4.20	3.80	**4.07**	Very Good
User error protection – the system provides notifications and protects the user during any fault.	3.80	3.80	3.80	**3.80**	Very Good
User interface aesthetics - user interface enables satisfying interaction for the user.	4.00	3.80	3.80	**3.87**	Very Good
Accessibility - system can be used in a wide range of characteristics to achieve a certain goal in a specific context of use.	4.00	3.80	4.00	**3.93**	Very Good
Average Mean	**4.03**	**3.87**	**3.87**	**3.92**	Very Good

As presented in the above table, the CIM got an average mean of 3.92 with "Very Good" descriptive rating. The system is easy to use and analogous with other popular IM in terms of user interface (UI).

The following table is the average Mean score of the CIM in terms of Reliability:

Table 17

Average Mean Score in terms of Reliability

Criteria	Mean Score			Average Mean	Descriptive Ratings
Reliability	IT Experts	Cyber Security Experts	End Users		
Maturity – the system meets the needs for reliability under normal operation.	4.20	3.80	3.80	**3.93**	Very Good
Availability – the system is accessible and fully functional when required for use.	3.80	4.00	4.00	**3.93**	Very Good
Fault Tolerance – the system operates despite the presence of software or hardware faults.	3.60	4.60	3.80	**4.00**	Very Good
Recoverability – the system can recover and re-establish the desired state after an interruption or failure.	3.40	3.80	4.00	**3.73**	Very Good
Average Mean	**3.75**	**4.05**	**3.90**	**3.90**	Very Good

As presented in the above table, the CIM got an average mean of 3.90 with "Very Good" descriptive rating. The system is stable and resilient when tested in an undesirable condition.

The table on the next page shows the average Mean score of the CIM in terms of Maintainability:

Table 18

Average Mean Score in terms of Maintainability

Criteria	Mean Score			Average Mean	Descriptive Ratings
Maintainability	IT Experts	Cyber Security Experts	End Users		
Modularity – the system is developed into components (so that a change of one component has minimal impact on another).	4.00	4.00	4.00	**4.00**	Very Good
Reusability – the system can be used in any mode of application.	4.00	4.00	4.20	**4.07**	Very Good
Analysability - the impact of any intended change to a system can be assessed or system deficiencies can be diagnosed.	4.00	3.80	4.00	**3.93**	Very Good
Modifiability – the system can be modified without introducing defects to the system.	3.80	3.80	3.80	**3.80**	Very Good
Testability – the tests can be performed to determine whether certain criteria have been met.	4.20	4.20	3.80	**4.07**	Very Good
Average Mean	**4.00**	**3.96**	**3.96**	**3.97**	Very Good

As presented in the above table, the CIM got an average mean of 3.97 with "Very Good" descriptive rating. The system is developed in modules or components making it agile in any environment.

The table on the next page shows the average Mean score of the CIM in terms of Portability:

Table 19

Average Mean Score in terms of Portability

Criteria	Mean Score			Average Mean	Descriptive Ratings
Portability	IT Experts	Cyber Security Experts	End Users		
Adaptability - the system can adapt to different or evolving hardware, software or other operating environments.	4.00	4.20	4.40	**4.20**	Very Good
Installability – the system can be successfully installed and/or uninstalled in any types of operating environment.	4.00	3.60	4.00	**3.87**	Very Good
Replaceability - the system can replace other software for similar purpose and environment.	3.80	4.20	4.20	**4.07**	Very Good
Average Mean	**3.93**	**4.00**	**4.20**	**4.04**	Very Good

As presented in the above table, the CIM got an average mean of 4.04 with "Very Good" descriptive rating. The system can only be ported or installed in a MS Windows environment.

The table on the next page shows the average Mean score of the CIM in terms of Compatibility:

Table 20

Average Mean Score in terms of Compatibility

Criteria	Mean Score			Average Mean	Descriptive Ratings
Compatibility	IT Experts	Cyber Security Experts	End Users		
Co-existence – the system functions efficiently while sharing common resources with other systems and without detrimental impact with each other.	4.20	4.00	4.40	**4.20**	Very Good
Interoperability - two or more systems can exchange information and interoperate seamlessly	3.60	3.80	4.60	**4.00**	Very Good
Average Mean	**3.90**	**3.90**	**4.50**	**4.10**	Very Good

As presented in the above table, the CIM got an average mean of 4.10 with "Very Good" descriptive rating. The system performs efficiently while sharing a similar .NET library with other applications. It can co-exist with other windows applications.

The table on the next page depicts the overall evaluation rating of the CIM based on ISO 25010 standards for software quality.

Table 21

Overall Average Mean Score of the CIM

Criteria	Mean Score			Average Mean	Descriptive Ratings
ISO 25010 Characteristics	IT Experts	Cyber Security Experts	End Users		
Functional Suitability	3.93	4.00	3.67	3.87	Very Good
Security	4.05	4.45	4.10	4.20	Very Good
Performance Efficiency	4.07	3.87	4.07	4.00	Very Good
Usability	4.03	3.87	3.87	3.92	Very Good
Reliability	3.75	4.05	3.90	3.90	Very Good
Maintainability	4.00	3.96	3.96	3.97	Very Good
Portability	3.93	4.00	4.20	4.04	Very Good
Compatibility	3.90	3.90	4.50	4.10	Very Good
Total	**3.96**	**4.01**	**4.03**	**4.00**	**Very Good**

It was rated high in terms of "security" and garnered an average rating of 4.20. The validity of its encryption gives the evaluator an assurance of its security. The CIM may not have achieved an excellent rating due to its limited features as compared to other popular IM; however, continues improvement of the system and integration of additional modules will enable it to compete in the IM arena.

Chapter 5

SUMMARY OF FINDINGS, CONCLUSIONS AND RECOMMENDATION

This chapter presents the summary of findings based from the evaluation conducted. Conclusion and recommendations are drawn based on the result of the study.

Summary of Findings

The Cryptographic IM (CIM) system is an instant messaging application that allows data exchange and office collaboration such as one-to-one chat, group chat, file transfer, and others. The completion of the three (3) modules/components – Base, Encryption, and File Transfer – allowed the whole system to deliver the desired functionalities as enumerated in the objectives and scope of this study. The CIM was able to provide the basic features found in any typical IM today such as chat and file transfer. Its prowess was seen in the implementation of multi-layered encryption mechanism - the integration of AES 256-bit encryption and HIPS hiding algorithm.

The result of the evaluation revealed that the CIM has little or no advantageous over other typical IM in terms functionalities. It scored low in portability since the application can only be installed in Microsoft Windows platform and not on an android or IOS mobile devices.

However, the system proved advantageous in terms of security as it garnered an average mean of 4.20. The validity of encryption gave the evaluator an assurance of its security.

From the three (3) groups of respondents or evaluators, IT experts gave 3.96 average mean, Cybersecurity experts gave 4.01, and end users gave 4.03. The end users examined the system from a usability perspective and found little or no difference with that of a typical IM, while the cybersecurity experts understand very well the importance of security in an instant messaging application.

Conclusion

Based on the result of the study, the following conclusions were drawn:

6. The CIM was successfully designed and developed in accordance with standards and required features. The following are the completed features of CIM:
   j. Multi-layered encryption through the combination of AES 256-bit and HIPS;
   k. Secured One-to-one and room chat;
   l. Secured File transmission;
   m. Secured login authentication;
   n. Able to transmit recorded voice message;
   o. Self-delete or automatic deletion of secret message; and
   p. Use of cipher key exchange methods:
      g.1 System Generated Key (SGK); and
      g.2 Manual Key Input (MKI).
   q. Encrypted username and password on the database
   r. The users can decrypt the file at the time of their choosing.
7. The CIM was successfully created using Visual C# .net.

8. The test results proved that the developed application is functional and secure when tested across different Penetration Testing software
9. The test results proved that the CIM will run on any MS Windows operating system except Windows XP.
10. The performance of CIM was evaluated using ISO 25010 software quality criteria with a grand mean of 4.00 and "Very Good" descriptive rating.

Recommendation

Based on the foregoing findings and conclusions, the following recommendations are drawn:

1. Further research to develop modules for voice and video communication must be done. Some open-source software libraries, API, and SDK are already available to enable this feature. These libraries can be integrated into the CIM.

2. Improvement of HIPS hiding technique should be considered. Since the programming language that was used to develop the HIPS is already obsolete, there is a need to conduct further study to learn new techniques in steganography. Tools for steganography nowadays can hide data in images, sound clips, videos, and office documents like MS Word, MS Excel, and MS Powerpoint.

3. Further research to develop modules for Short Messaging System or SMS may be conducted. This will allow messaging even if the user is offline and the only available mode of communication is the Global System for Mobile communication (GSM) commonly known as cellphone signal.

4. Development of an android-based and IOS-based CIM client can be initiated. These versions of CIM will provide easier access and mobility to many users.

5. Further research to develop modules for Voice Over Internet Protocol or VOIP may also be considered. This will allow voice communication on any VOIP-enabled phone.

6. Establishment of a standard procedure for the utilization and maintenance of the CIM can be made. This will allow segregation of duties between the administrator and the users.

Crytographic Instant Messaging

ORIGINALITY REPORT

6%	4%	2%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Technological University Of The Philippines Student Paper	1%
2	en.wikipedia.org Internet Source	<1%
3	Submitted to TAR University College Student Paper	<1%
4	Submitted to Murdoch University Student Paper	<1%
5	www.ee.ryerson.ca Internet Source	<1%
6	H.A. Schmid. "Performance problems of large operational systems based on web services and a solution", IEEE International Conference onServices Computing, 2004. (SCC 2004). Proceedings. 2004, 2004 Publication	<1%
7	db.s2.chalmers.se Internet Source	<1%

8	Temilade Adefioye Aina, Louise Cooke, Derek Stephens. "Methodology for evaluating CI software packages", Business Information Review, 2016 Publication	<1%
9	Submitted to American Sentinel University Student Paper	<1%
10	Submitted to University of Mindanao Student Paper	<1%
11	cdn.softwaretestinghelp.com Internet Source	<1%
12	Yulianto, Budi, Eileen Heriyanni, Lusiana Citra Dewi, and Timothy Yudi Adinugroho. "Architecture and Implementation of Instant Messaging in Educational Institution", Procedia Computer Science, 2015. Publication	<1%
13	www.sanbeda-alabang.edu.ph Internet Source	<1%
14	www.inass.org Internet Source	<1%
15	Submitted to Segi University College Student Paper	<1%
16	Cai, Lizhi, Xiaoyan Xie, and Shidong Huang. "Software Quality Model Development-An	<1%

Introduction", Energy Procedia, 2011.
Publication

17	www.slideshare.net Internet Source	<1%
18	letran-calamba.edu.ph Internet Source	<1%
19	Submitted to CSU, San Jose State University Student Paper	<1%
20	Submitted to Western International University Student Paper	<1%
21	Chinnakandukuri Paul Pramod, Manjit Jaiswal. "An advanced AES algorithm using swap and 400 bit data block with flexible S-Box in Cloud Computing", 2017 International Conference on Computing, Communication and Automation (ICCCA), 2017 Publication	<1%
22	Submitted to Cavite State University Student Paper	<1%
23	Submitted to Keller Graduate School of Management Student Paper	<1%
24	pnrsolution.org Internet Source	<1%
25	www.issr-journals.org	

	Internet Source	<1%
26	Submitted to Wawasan Open University Student Paper	<1%
27	Submitted to Technological Institute of the Philippines Student Paper	<1%
28	Submitted to Mahidol University Student Paper	<1%
29	Submitted to University of Edinburgh Student Paper	<1%
30	Submitted to Deakin University Student Paper	<1%
31	searchsecurity.techtarget.com Internet Source	<1%
32	docplayer.net Internet Source	<1%
33	ir.library.oregonstate.edu Internet Source	<1%
34	gurukpo.com Internet Source	<1%
35	Andrew Flanagin. "IM Online: Instant Messaging Use Among College Students", Communication Research Reports, 8/1/2005	<1%

	Publication	
36	www.vyapin.com Internet Source	<1%
37	kenmchale.com Internet Source	<1%
38	www.faqs.org Internet Source	<1%
39	www.babelway.com Internet Source	<1%
40	bombshellauthors.com Internet Source	<1%
41	journal.student.uny.ac.id Internet Source	<1%
42	A. Aldris, A. Nugroho, P. Lago, J. Visser. "Measuring the Degree of Service Orientation in Proprietary SOA Systems", 2013 IEEE Seventh International Symposium on Service-Oriented System Engineering, 2013 Publication	<1%
43	anyfh.strefa.pl Internet Source	<1%
44	voicepoint.ch Internet Source	<1%
45	Zubrow, David. "Software Metrics and	<1%

Measurements", Computing Handbook Third Edition, 2014.

Publication

Exclude quotes	On	Exclude matches	< 5 words
Exclude bibliography	On		

Crytographic Instant Messaging

GRADEMARK REPORT

FINAL GRADE

/0

GENERAL COMMENTS

Instructor

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

PAGE 19

PAGE 20

PAGE 21

PAGE 22

PAGE 23

PAGE 24

PAGE 25

PAGE 26

PAGE 27

PAGE 28

PAGE 29

PAGE 30

PAGE 31

PAGE 32

PAGE 33

PAGE 34

PAGE 35

PAGE 36

PAGE 37

PAGE 38

PAGE 39

PAGE 40

PAGE 41

PAGE 42

PAGE 43

PAGE 44

PAGE 45

PAGE 46

PAGE 47

PAGE 48

PAGE 49

PAGE 50

PAGE 51

PAGE 52

PAGE 53

PAGE 54

PAGE 55

PAGE 56

PAGE 57

PAGE 58

PAGE 59

PAGE 60

PAGE 61

PAGE 62

PAGE 63

PAGE 64

PAGE 65

PAGE 66

PAGE 67

PAGE 68

PAGE 69

PAGE 70