LAF Chat: A Message Encrypting Application Utilizing RSA Algorithm for Android-Based Mobile Device

Rowe-Ann Antenor[1], Robert Bautista[2], Francis Paolo Lesaca[3], Raychelou Valencia[4]

Department of Computer Science, Don Bosco Technical College, Mandaluyong City, Philippines
[1]rfantenor@donbosco.edu.ph
[2]rvbautista@donbosco.edu.ph
[3]fpflesaca@donbosco.edu.ph
[4]rtvalencia@donbosco.edu.ph

***Abstract*— The researchers have decided to create an application for the sole purpose of protecting once private information even as small as a basic text message. Using the Rivest Shamir Adleman algorithm (RSA). It is an algorithm used to encrypt and decrypt messages. It uses an asymmetric cryptography algorithm which means it has 2 different keys. Algorithm involves a public key and a private key. The public key could be shared to anyone; it used to encrypt the message. Message encrypted using the public key can only be decrypted with the private key. The use of the RSA Algorithm is mainly used in banks for security purposes; the researchers have proposed and developed a program that would use the algorithm, combining it in a mobile messaging application in Android and also using Google Cloud Messaging (GCM). Security, especially when it comes to phones is a problem. Private information is something one poses in their everyday lives, and people all over the world have tried to increases security but there are also those numbers of people that have tried stealing information to use for their personal gain. Security is almost slowly becoming impossible since the rapid improvement of technology.**

K*eywords*— Rivest Shamir Adleman algorithm (RSA), Cryptography, Assymetric Encryption, Android, Google Cloud Messaging (GCM), Security

I. INTRODUCTION

A. *Background of the Study*

With the cutting-edge computer technology and the existing capability of Internet telecommunications today, there are many possible ways of relaying a message to someone for something important, confidential or even just a simple conversation. However, security is one of the common problems in forwarding a message online, either through web or a mobile device. A study stated that "Out of the numerous people who are aware and using the World Wide Web, 40% of these people are still using unsafe browsing facility".[1] It can be said that the Internet is one of the most public places to communicate; During this time when the Internet provides essential communication between tens of millions of people and is being increasingly used as a tool for commerce, security becomes a tremendously important issue to deal with[2]; transmitting information to another is dangerous, especially if that information is something that is extremely confidential.

A way to secure ones messages is through encryption. To make sure that they do safe transactions every time, there must be some technology, which assures and safeties of usage. This is known to be Encryption. [1] Encryption is the conversion of electronic data into another form using an algorithm, called ciphertext, which cannot be easily understood by anyone except the authorized parties. A ciphertext is the result of encryption performed on plaintext using an algorithm, called a cipher. Decryption, the inverse of encryption, is the process of turning ciphertext into readable plaintext. The primary purpose of encryption is to protect the confidentiality of digital data stored on computer systems or transmitted via the Internet or other computer networks.

B. *Significance of the Study*

This section will provide description on the various significances of the study given the three categories; users, programmers and future researchers. Nowadays, there are many ways to relay a message to someone for something important, confidential or even just a simple conversation. The proposed study will serve as one of the ways to deliver a message through an android device to another android device but is more secured. Users of android device who are sending messages online will be one of those who will benefit from the study. The application will take their message's confidentiality to the next level. By using the application, the user can now encrypt/decrypt his messages. Another target beneficiary of the study are the programmers and future researchers for the abstraction of the same topics as the study, related to Rivest Shamir Adleman algorithm, message encryption, mobile application, or other related studies, like cryptology, and threats concerns of mobile internet safety.

C. *Scope and Limitations*

1) Scope

The coverage of this study is to implement the Rivest Shamir Adleman (RSA) algorithm in encrypting a personal text message from a user of an android device to a receiver on another android device that is connected to the same network. The study involves developing a message encrypting application that is capable of receiving/sending a message and is able to transform that message from plain text to a cipher text using an asymmetric encryption. The application should be installed on each of the android device to be able to use the program. All the users that is connected to the server is considered online and will be visible to all other users using the application.

2) Limitations

The study is limited to user to user encryption; there will not be a group message encryption available in the study. The user cannot use the application without internet connection and must be connected to the same network. The mobile application is fully reliable to the speed of the internet connection and the server. The encryption can only work with all the letters and symbols that could be found on a regular keyboard but all symbols can be encrypted as well as emoticons.

II. REVIEW OF RELATED LITERATURE

A. Cryptography

Cryptography is basically the science that employs mathematical logic to keep the information secure. It enables someone to securely store sensitive information or transmit information securely through insecure networks to keep it from being hacked, masqueraded, or altered [3]. The purpose of Cryptography is the science of writing in secret code and is an ancient art; the first documented use of cryptography was written to date from circa 1900 B.C. when an Egyptian scribbled non-standard hieroglyphs in an inscription but people argued that Cryptography was invented during the time of war. Cryptography is necessary when communicating with anyone using the internet in case someone would intercept your messages.

When it comes to communication of any application to application, there are specific security requirements that need to be meant. They are Authentication, Privacy, Integrity, and Non-repudiation. Authentication is the process of providing your identity. Privacy is ensuring that no one could read the message except the person intended to receive it. Integrity is assuring the receiver that the message has not been changed or modified and Non-repudiation is a mechanism to prove that the source of message or composer of the message is legit [2]

The program "obfuscation" has been around for at least decades, but no one has ever developed a mathematical framework for the idea. Over the years commercial software companies have created different ways for garbling a computer program so that it will be harder to understand while still processing or doings its purpose but hackers have defeated every single attempt. At best, these obfuscators offer a "speed bump" said by Sahai, now a computer science professor at the University of California. A hacker would now take a few days to unlock the information of the software instead of just taking a few minutes.

Secure program obfuscation would be useful for many possible applications, such as protecting software patches, chips that read encrypted DVDs or even encrypting military equipment's like drones. More futuristically, it would allow us people to create automated virtual agents that we could use to send out into the computing "cloud" to act on our behalf. Your valued information from credit card numbers, pass codes and password would be safe hidden inside that program.

Currently the program doesn't exist. As Sahai pondered program obfuscation, however, he and other of his colleagues quickly realized that its potential far surpassed any specific application. If ever a program as obfuscator would ever be created, it could solve many of the problems that have been driven cryptography for the past years, problems about how to conduct secure interactions with people at, say, the other end of an internet connection, whom you may not know or even trust for that fact [4].

B. Asymmetric Encryption

Asymmetric cryptography or public-key cryptography is cryptography in which a pair of keys is used to encrypt and decrypt a message so that it arrives securely. Initially, a network user receives a public and private key pair from a certificate authority. Any other user who wants to send an encrypted message can get the intended recipient's public key from a public directory. They use this key to encrypt the message, and they send it to the recipient. When the recipient gets the message, they decrypt it with their private key, which no one else should have access to. [5]

An asymmetric algorithm, as outlined in the Diffie-Hellman paper, is a *trap door* or *one-way* function. Such a function is easy to perform in one direction, but difficult or impossible to reverse. For example, it is easy to compute the product of two given numbers, but it is computationally much harder to find the two factors given only their product. Given both the product and one of the factors, it is easy to compute the second factor, which demonstrates the fact that the hard direction of the computation can be made easy when access to some secret key is given. Public-key algorithms are based on mathematical problems which currently admit no efficient solution that are inherent in certain integer factorization, discrete logarithm, and elliptic curve relationships. [6]

This allows for the exchanging of securely signed and one-to-one messages. The sender encrypts the message using the common algorithm and his own secret key. They then sign the result, encrypt it again (with their signature in cleartext) using the recipient's public key, and send it. The recipient decrypts the received message using their own secret key, identifies the sender from their now-cleartext signature, and then decrypt the result using the sender's public key. This ensures the recipient that whoever composed the message had access to the sender's private key, and that nobody tampered with the message or read it along the way. [5]

In symmetric cryptography, the same key is used for both encryption and decryption. This approach is simpler in dealing with each message, but less secure since the key must be communicated to and known at both sender and receiver locations.

That is why the researchers decided to use asymmetric encryption instead of symmetric encryption. Asymetric encryption is much harder to break rather than symtric because it has a private and a public key.

C. *Google Cloud Messaging*

In order to connect the applications we will be using GCM in our Message encryption application. According to developer.android.com, Google Cloud Messaging (GCM) for Android is a service that allows you to send data from your server to your users' Android-powered device, and also to receive messages from devices on the same connection. The GCM service handles all aspects of queuing of messages and delivery to the target Android application running on the target device, and it is completely free.

According to Google, The service provides a simple, lightweight mechanism that servers can use to tell mobile applications to contact the server directly, to fetch updated application or user data. The service handles all aspects of queuing of messages and delivery to the target application running on the target device. The free service has the ability to send a lightweight message informing the Android application of new data to be fetched from the server. Larger messages can be sent with up to 4 KB of payload data. Each notification message size is limited to 1024 bytes. [7]

Applications on an Android device don’t need to be running to receive messages. The system will wake up the application via a mechanism called Intent Broadcast when the message arrives, as long as the application is set up with the proper broadcast receiver and permissions. GCM does not provide any built-in user interface or other handling for message data. Instead, it simply passes raw message data received straight to the application, which has full control of how to handle it. For example, the application might post a notification, display a custom user interface, or silently sync data.

D. *Java Programming Language*

Java Programming Language is a general-purpose computer programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. It is also intended to let application developers "write once, run anywhere", meaning that code that runs on one platform does not need to be recompiled to run on another. Java applications are typicallycompiledtobyte-code that can run on any Java virtual machine regardless of computer architecture. Java is, as of 2014, one of the most popular programming languages in use, particularly for client-server web applications, with a reported 9 million developers

Java was originally developed by James Gosling at Sun Microsystems which has since merged into Oracle Corporation and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++, but it has fewer low-level facilities than either of them. Sun Microsystems released the first public implementation as Java 1.0 in 1995. The language was initially called Oak after an oak tree that stood outside Gosling's office. Later the project went by the name Green and was finally renamed Java, from Java coffee that is said to be consumed in large quantities by the language's creators.Gosling designed Java with a C/C++-style syntax that system and application programmers would find familiar. [8]

E. *Rivest Shamir Adleman Algorithm*

RSA is an algorithm used to encrypt and decrypt messages. It uses an asymmetric cryptography algorithm which means it has 2 different keys. RSA was announce publicly in 1978 and stands for Ron Rives, Adi Shamir and Leonard Adleman. The algorithm involves a public key and a private key. The public key could be shared to anyone; it used to encrypt the message. Message encrypted using the public key can only be decrypted with the private key [9]. The algorithm can be used for both public key encryption and for digital signatures. Its security is based on the difficulty of factoring the large integers.

III. SYSTEM ARCHITECTURE

A. *System Architecture*

The researchers will be using Android Studio and java programming language for making the system. It includes features like Live Layout: WYSIWYG Editor, Live Coding, Real-Time App Rendering, rich layout editor that allows users to drag-and-drop UI components, option to preview layouts on multiple screen configurations and other tools for mobile development.

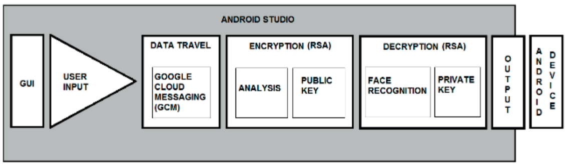

Fig. 1 System Architecture

Generally, the primary part of the Message Encryption Application will be assembled inside the Android Studio integrated development environment including but not limited to; Graphical User Interface, overall design, buttons, text and images that will be used or be implemented in the Application. The RSA algorithm will be used for message encryption of the application, it includes the generation of the public key and private key for the sender and recipient. Most Android apps need to save data, even if only to save information about the app state during onPause() so the user's progress is not lost. Most non-trivial apps also need to save user settings, and some apps must manage large amounts of information in files and databases. For this we will use internal storage for saving data. This introduces the principal data storage options in Android, including; saving key-value pairs of simple data types in a Shared Preferences file.

B. Algorithm

1) RSA

The Rivest-Shamir-Adleman (RSA) algorithm is one of the most popular and secure public-key encryption methods. The algorithm capitalizes on the fact that there is no efficient way to factor very large (100-200 digit) numbers. RSA is a cryptosystem for public-key encryption, and is widely used for securing sensitive data, particularly when being sent over an insecure network such as the Internet.

As stated in the book by Rivest, R.L., Shamir, A., and Adleman, L. (February 1978), entitled “A Method for Obtaining Digital Signatures and Public-Key Cryptosystems", *Communications of the ACM*, Vol 21, No. 2 using an encryption key (e, n), the algorithm is as follows:

1. Represent the message as an integer between 0 and (n-1). Large messages can be broken up into a number of blocks. Each block would then be represented by an integer in the same range.
2. Encrypt the message by raising it to the eth power modulo n. The result is a ciphertext message C.
3. To decrypt ciphertext message C, raise it to another power d modulo n

The encryption key (e, n) is made public. The decryption key (d, n) is kept private by the user. Rivest, Shamir, and Adleman provide efficient algorithms for each required operation.

How to Determine Appropriate Values for e, d, and n

1. Choose two very large (100+ digit) prime numbers. Denote these numbers as p and q.
2. Set n equal to $p * q$.
3. Choose any large integer, d, such that GCD(d, ((p-1) * (q-1))) = 1
4. Find e such that $e * d = 1$ (mod ((p-1) * (q-1)))

The question of how secure is a communication using RSA is secure still remains. In line with the statement on the website courses.cs.vt.edu entitled “Cryptography: RSA algorithm”, it is said that Cryptographic methods cannot be proven secure. Instead, the only test is to see if someone can figure out how to decipher a message without having direct knowledge of the decryption key. The RSA method's security rests on the fact that it is extremely difficult to factor very large numbers. If 100 digit numbers are used for p and q, the resulting n will be approximately 200 digits. The fastest known factoring algorithm would take far too long for an attacker to ever break the code. Other methods for determining d without factoring n are equally as difficult. Any cryptographic technique which can resist a concerted attack is regarded as secure. At this point in time, the RSA algorithm is considered secure.

The encryption using the public key generated by the RSA algorithm is as follows. Let’s say you want to send the message “BAD CHEF” and encrypt it using the public keys 5, 14 and assuming that you still don’t know the private keys that will be used to decrypt it.

2) Encryption Process

Public Key: 5, 14
Private Key: 11, 14

(1)The first step to do this is to convert the letters into numbers (A is to 1, B is to 2, C is to 3….).

B	**A**	**D**	**C**	**H**	**E**	**F**	→ Original Text
2	1	4	3	8	5	6	→ Equivalent number

(2)The second step is to raise the converted text to the power of 5.

B	**A**	**D**	**C**	**H**	**E**	**F**	→ Original Text
2	1	4	3	8	5	6	→ Equivalent number
32	1	1024	243	32768	3125	7776	→ raise to the power of 5

(3)The third step is to get the %14.

B	**A**	**D**	**C**	**H**	**E**	**F**	→ Original Text
2	1	4	3	8	5	6	→ Equivalent number

32 1 1024 243 32768 3125 7776 → raise to the power of 5

4 1 2 5 8 3 6 → %14, and the encrypted Text

3) Decryption Process

Now, for the decryption process we need to use the private key which is 11, 14.

(1)The first step is to get the encrypted message and raise it to the power of 11

Public Key: 5, 14
Private Key: 11, 14

4 1 2 5 8 3 6 → encrypted text
4194304 1 2048 48828125 177147 8589934592 362797056 → raised to 11

(2) The second step is to get the %14.

4 1 2 5 8 3 6 → encrypted text
4194304 1 2048 48828125 177147 8589934592 362797056 → raised to 11
2 1 4 3 8 5 6 → %14

(3) After getting the %14, you can now covert it to its original text.

4 1 2 5 8 3 6 → encrypted text
4194304 1 2048 48828125 177147 8589934592 362797056 → raised to 11
2 1 4 3 8 5 6 → %14

B A D C H E F → Decrypted Text

Note that the private key and public key that is given on the example is for explaining purposes only. The public and private keys on the LAF chat application consists of very large numbers for higher security.

C. Data Collection Methodology

- Survey Questionnaire

A survey would be used by the researchers to give to random individuals for them to gather information on how much big is the population that would play or use their software. It is when a person or a group would have different individuals answer different questions that would provide information to the group about their certain product or project, in this case, a study entitled "LAF CHAT: A Message Encrypting Application Utilizing Rsa Algorithm For Android-Based Mobile Device."

- Software Evaluation

Software evaluation is a type of assessment that looks to determine if the software is the best possible fit for the client. The idea is look at the resources and tools given by the software that it either currently in use or is being tested as a possible addition to programs used by the client.

D. System Development

- Iterative method

It is developed to overcome the weaknesses of the waterfall model. It starts with an initial lanning and ends with deployment with the cyclic interactions in between. The basic idea behind this method is to develop a system through repeated cycles.

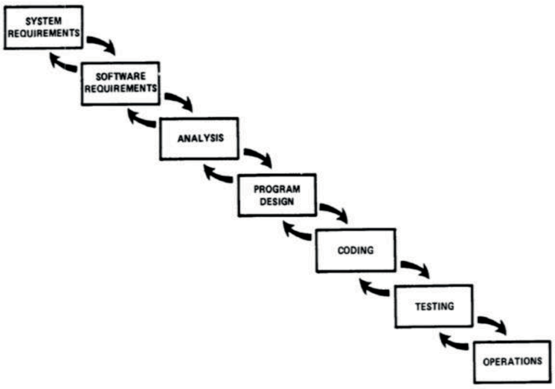

Fig. 2 Iterative Method

On the first phase, the researchers were to identify the system requirements os the application to be developed. Specifically, the following were to be identified: What version of android phones will be able to use the product? What specific algorithm should the researchers use to fully guarantee the security of the system? What methods are to be used? Then follows the software requirements and the analysis of the data gathered. From that the researchers came up with the program design. The researchers will use the Google Cloud Messaging and the Android studio for coding/programming the mobile message encrypting application. The good thing about this method is that once the researchers ended a phase and started another one is that one could always go back and do it twice, unlike the waterfall model.

IV. RESULTS AND DISCUSSIONS

A. *Testing Encryption and Decryption*

TABLE I : Sample Input and Output

Input	Decrypted Output	Match?
A Chip on Your Shoulder.	A Chip on Your Shoulder.	YES
An Arm and a Leg.	An Arm and a Leg.	YES
Fisherman's friends.	Fisherman's friends.	YES
Back to the Future.	Back to the Future.	YES
A Million ways to Die IN the WEST.	A Million ways to Die IN the WEST.	YES
"This is it!"	"This is it!"	YES
Get to the point, leaving out all of the unnecessary details.	Get to the point, leaving out all of the unnecessary details.	YES
Remember KIDS!	Remember KIDS!	YES
!@#$%^&*()	!@#$%^&*()	YES
_+	_+	YES
[ ] \ ; ' , . /	[ ] \ ; ' , . /	YES
{ } | : " < > ?	{ } | : " < > ?	YES
>ghgd	>ghgd	YES
:) :D ;) ;D :3 ;3	:) :D ;) ;D :3 ;3	YES
Emoticons	blank	NO
<3	blank	NO
<>	blank	NO
</3	blank	NO

Table I shows the sample input and output of the application. The researchers did not include samples of encrypted text input for the reason that each and every user of the application have their own unique private and public key which means that each will have different results in the ciphertext.

The application will only be able to encrypt and decrypt successfully if the text is less than or equal to 64 characters, this is because the KeyPairGenerator is only 512 bit long. We can use longer KeyPair and make the system more secure and allow for longer messages but will sacrifice on longer time for decrypting the message.

As you can see in the bottom part of the table, that some output after decryption is blank. What happens here is that it can encrypt the data and send it to the recipient but when decryption is applied on the part of the recipient's device, it is read to be blank and displays nothing. Emoticons built into the phone is not supported also.

B. *Time Consumed*

TABLE II : Time for the message to be delivered

Time to Complete (ms)	Internet Speed
3242ms	2.95Mbps down | 1.04Mbps up
3676ms	2.95Mbps down | 1.04Mbps up
1025ms	2.95Mbps down | 1.04Mbps up
1054ms	2.95Mbps down | 1.04Mbps up
484ms	2.95Mbps down | 1.04Mbps up
710ms	2.95Mbps down | 1.04Mbps up
632ms	2.95Mbps down | 1.04Mbps up
1060ms	2.95Mbps down | 1.04Mbps up
571ms	2.95Mbps down | 1.04Mbps up
1229ms	2.95Mbps down | 1.04Mbps up
1349ms	2.95Mbps down | 1.04Mbps up
609ms	2.95Mbps down | 1.04Mbps up
1060ms	2.95Mbps down | 1.04Mbps up
6501ms	2.95Mbps down | 1.04Mbps up
N/A	N/A
1273ms	2.95Mbps down | 1.04Mbps up
3791ms	2.95Mbps down | 1.04Mbps up
1242ms	2.95Mbps down | 1.04Mbps up

Table II shows the time consumed for the message from the sender to be delivered to the recipient. We included the speed of the internet that we are using to serve as a baseline. Because having different speeds affect the time needed for the data to be sent. We can see from the table that our fastest message is delivered in 0.484sec and the slowest was 6.501sec.

The researchers included the internet speed because it affects the time of completion process in addition to how many characters are used. But as you can see the slowest was still running the same as others internet speed. This can be due to the phone's Wi-Fi capability and obstructions that results in weak signal or just plain signal loss.

V. CONCLUSIONS AND RECOMMENDATIONS

A. *Conclusions*

The research paper tackles about the possibility of using RSA algorithm as a security measure for a messaging application for Android OS. The researchers used Google cloud messaging for a path way for users to interact with their messages, programmed using java; the researchers have created a message encryption application utilizing RSA algorithm. The applications would allow its users to message different people while having RSA algorithm as their number 1 source of security since it will encrypt the user's message and only the receiver would have the capability to decrypt the encrypted message.

As a conclusion the researchers have achieved the following

1) The researchers have successfully implemented the RSA algorithm in a mobile application and use it as

the main tool for security for communication via messaging. RSA is usually used for pins usually found in ATMs and the researchers successfully made use of RSA as encryption tool for longer number of characters then its purpose for shorten number of characters such as pins.

2) The researchers created an application that could send and receive encrypted messages and decrypted by the receiver. The message would be encrypted with use of algorithm and passed through the Google cloud, then would be delivered to the designated receiver.
3) The researchers were able to add more security to the application. The researchers were able to add security questions and pin number security for added security of the application itself since the message is still accessible via the phone but to add more security the researchers placed more security measure to assure the user that even if the phone is stolen, their information would be secured.

B. Recommendations

Based from the different errors and limitations that the researchers have encountered, the researchers recommend that:

1) For future use of the creating a messaging application that the future developers would consider using a faster possible internet since the transaction of messages depends on how fast the internet connection is.
2) The future developers would also consider using a different path way other then Google's Cloud Messaging since the researchers have this during its pre release and was given permission to test Google's product. Since it was pre release it is possible that it might not be available to use o could require paying for the use of Google cloud messaging.
3) Improve the interface for possible new users since people have different understandings and it might not be easy for new users to understand the interface of the application.
4) Adding face recognition for a possible security measure since it's a more reliable when it comes to security. It would require the user's exact facial features and it would be impossible for anyone to gain access or by pass the security
5) Add also Finger print scanning as another source of security since face recognition now a day aren't that advance yet and don't deliver amazing results. Finger print scanning is currently advancing and getting good result. It is used as one of the main security of current smartphones.

REFERENCES

[1] K. I. Lakhtaria, "Protecting Computer Network with Encryption Technique: A Study," in *Ubiquitous Computing and Multimedia Applications*, 2011, pp. 381–390.

[2] K. Gary C., "An Overview of Cryptography." [Online]. Available: https://www.garykessler.net/library/crypto.html. [Accessed: 4-Jan-2016].

[3] *Practical Cryptography: Algorithms and Implementations Using C++*, Boca Raton: CRC Press, 2015.

[4] "Cryptography Breakthrough Could Make Software Unhackable," *WIRED*. [Online]. Available: https://www.wired.com/2014/02/cryptography-breakthrough/. [Accessed: 8-Feb-2016].

[5] "What is asymmetric cryptography (public key cryptography)? - Definition from WhatIs.com," *SearchSecurity*. [Online]. Available: https://searchsecurity.techtarget.com/definition/asymmetric-cryptography. [Accessed: 18-Jul-2016].

[6] W. Diffie and M. Hellman, "New Directions in Cryptography," *IEEE Trans. Inf. Theor.*, vol. 22, no. 6, pp. 644–654, Sep. 2006.

[7] "Google Cloud Messaging: Overview | Cloud Messaging," *Google Developers*. [Online]. Available: https://developers.google.com/cloud-messaging/gcm. [Accessed: 24-Jun-2016].

[8] "History of Java Technology," *Oracle*. [Online]. Available: http://www.oracle.com/technetwork/java/javase/overview/javahistory-index-198355.html. [Accessed: 25-Jan-2016].

[9] "What is RSA algorithm (Rivest-Shamir-Adleman)? - Definition from WhatIs.com," *SearchSecurity*. [Online]. Available: https://searchsecurity.techtarget.com/definition/RSA. [Accessed: 16-Aug-2016].