INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH & DEVELOPMENT

ISSN 2278 – 0211 (Online)

Mail Security Using Modified RSA Algorithm

M. Praveen
Student, Fourth Year Computer Science and Engineering
V.S.B Engineering College, Karur, Tamil Nadu, India
R. Sarath Kumar
Student, Fourth Year Computer Science and Engineering
V.S.B Engineering College, Karur, Tamil Nadu, India
V. Muhamadhu Bilal
Student, Fourth Year Computer Science and Engineering
V.S.B Engineering College, Karur, Tamil Nadu, India
A. P. V. Raghavendra
Associate Professor, Computer Science and Engineering
V.S.B Engineering College, Karur, Tamil Nadu, India

Abstract:
In recent years, there is a great thread for security in our communication. With new technologies, it became easy for the hackers to steal our information or message in a communication medium like Gmail, Yahoo, etc,. To protect our information that we want to communicate with authorized persons, this application is developed. It is a desktop application (like Google Talk), that performs all the activities in mail (sending and receiving mail with or without attachments) that can be performed in a web browser. The application performs encryption while sending and decryption while receiving. The encryption process is done by the use of modified RSA algorithm. The encryption process will be made in the application and then the encrypted data will sent through the communication medium. Similarly, the decryption process will also be made only in the application. Mail(s) that are sent using this application won't be present in a readable format in your Inbox. Those mail(s) can be viewed in the application only. This application will completely protect our message from reading by others, even if a third party person knows our mail password, the message won't be leaked.

Key words: *Encryption, decryption, RSA algorithm, communication, security*

1. Introduction

- A window application (like Google Talk) that performs all the activities in mail (sending and receiving mail with or without attachments) that can be performed in a web browser.
- The application performs encryption while sending and decryption while receiving. The encryption process is done by the use of modified RSA algorithm.
- Mail(s) that are sent using this application won't be present in a readable format in your Inbox. Those mail(s) can be viewed in the application only.
- This application will completely protect our message from reading by others.

2. Base Concept

- In this application, RSA algorithm is used. The most common public-key algorithm is the RSA cryptosystem, named for its inventors (Rivest, Shamir and Adleman)
- RSA uses two exponents, e and d, where e is public and d is private. P is the plain text and C is the cipher text. Sender uses $C = P^e \bmod n$ to create cipher text C from plain text P
- Receiver uses $P = C^d \bmod n$ to recover back the plain text from the cipher text. This encryption and decryption process use modular exponentiation.

- n is a large number which is produced by the multiple of two prime numbers p and q, i.e. n = p * q.

3. Base Concept Disadvantage

- Since Gmail has declared that they will read the mails that are in their database, there is no privacy for the sender.
- If a person hacks into our mail, then he can easily read the mails in our Inbox and Sent box.

4. Proposed Concept

This proposed concept will overcome the disadvantages of the existing concept by the following process,

4.1. Encryption

- The key and the message are taken in two separate arrays.
- The number at the even position of the Key will get added to the character (or symbol) at the even position of the message.
- Similarly, the number at the odd position of the Key will get subtracted from the character (or symbol) at odd position of the text.
- Then the key values will get added and the total sum of the key is taken to get a prime number for using it as p in RSA algorithm.
- A smallest value from the key is taken to get a prime number q, which is used in RSA process.
- Then the RSA process will be performed for further encrypting the message.

4.2. Decryption

- Decryption will be done in a exact opposite way as encryption.
- First RSA process will take place and then key at odd position will get added to the character at odd position and key at even position will get subtracted from character at even position to get back the original message.

5. Algorithm

- **RSA_Key_Generation**

{
Select two large primes p and q such that $p \neq q$
$n \leftarrow p * q$
$\Phi(n) \leftarrow (p-1) * (q-1)$
Select k such that $1 <= k <= \Phi(n)$
$d * e = k * \Phi(n)$, where d and e should be prime
Apply the values for d and e, that satisfies the above condition
}
Then it uses $C = P^e \bmod n$ to encrypt and $P = C^d \bmod n$ to decrypt.

6. Design of the Application

- This application will get the email id and password from the user in the initial tab.
- When the user clicks the Login button, it will get authentication from the mail server and make us to login into our mail.
- After getting authentication, three tabs will be available in the application, one for sending mails, called Send, one for receiving mails, called Receive and the last can be used for sample encryption and decryption purpose.

6.1. Login Tab

- A text field to enter the user name.
- A password field to enter the password.
- A Login button to login into mail.

6.2. Send Tab

- A Key field will be available for providing the key for the encryption process.
- A To field for entering the mail address (es) and a Subject field to enter the subject of the mail.
- A Message field to enter our message and a Attachment field to attach files.
- A Send button, which encrypt and send our message, when it is clicked.
- A Logout button to logout from Mail.

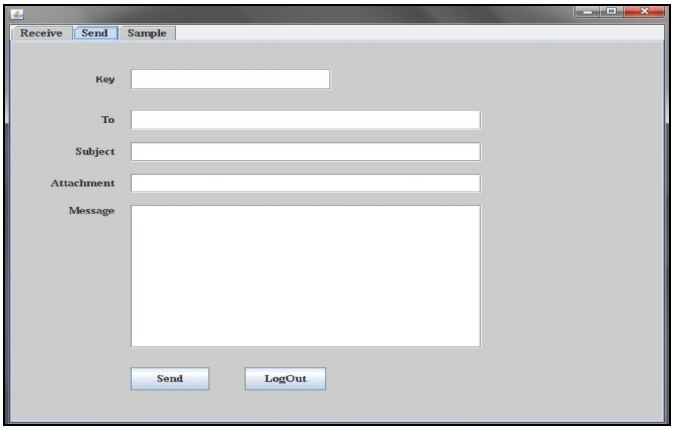

Figure 1: Send Tab

6.3. Receiver Tab

- A Key field for entering the Key which the sender provides.
- A Message field to display the mail(s) in the Inbox.
- A text field that asks the user to enter the identity number of the mail that he wants to read.
- A text field that asks the user to enter the name of the label that he wants to read.
- A Read button to read the selected mail.
- A View Mail button to view the mail(s) in the Inbox.
- A Delete button to delete the selected mail.
- A Download button to download the attachment(s), if any.

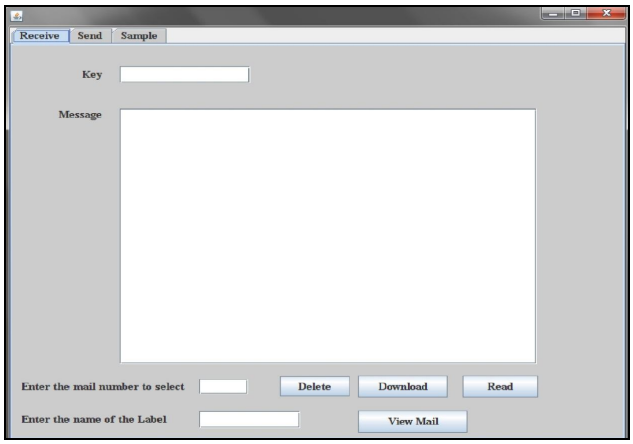

Figure 2: Receive Tab

7. Working

- After getting the user id and password from the user, the application will make authentication with the mail server and get into the mail.
- Then, if we want to send mail, go to the Send tab and fill the required fields and send the mail, else if we want to read the mails in the inbox, go to the Receive tab and click View Mail button, then the mails in your Inbox will be listed with an id representing it nearby.
- Enter the id of the mail that you want to read in the respective field and the key for the mail in the key field and click the Read button.
- Then the decrypted mail will be displayed in the message field. If there is any attachment available, it can be downloaded by clicking the Download button.
- The original form of the mail can be viewed only in this application by providing the Key. It cannot be viewed in the web browser.
- Even if anybody gets our password, he/she cannot view the contents of our mail. The content either cannot be viewed in Gmail database.
- Hence the message security will be maintained by using this application.

8. Software Requirements

- Windows XP or later
- JDk 1.6.0 or later

9. Technical Background

- Front End – Java
- Back End – Gmail Server
- IDE – Netbeans 7.0.1

10. References

1. Rivest, R.; A. Shamir; L. Adleman (1978). "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems". Communications of the ACM 21 (2): 120–126.Doi: 10.1145/359340.359342.
2. SIAM News, Volume 36, Number 5, June 2003, "Still Guarding Secrets after Years of Attacks, RSA Earns Accolades for its Founders", by Sara Robinson
3. http://www.rsa.com/press_release.aspx?id=261
4. Boneh, Dan (1999). "Twenty Years of attacks on the RSA Cryptosystem". Notices of the American Mathematical Society46 (2): 203–213.
5. Johan Håstad, "On using RSA with Low Exponent in a Public Key Network", Crypto 85
6. Don Coppersmith, "Small Solutions to Polynomial Equations, and Low Exponent RSA Vulnerabilities", Journal of Cryptology, v. 10, n. 4, Dec. 1997