



The CryptoMailer Using Keyword Cipher Algorithm

Abhishek Bhalotia¹, Aditya Rajneesh Singh², Supriya HS³

B.E Student, Department of Computer Science and Engineering, Sir M Visvesvaraya Institute of Technology, Bengaluru, India¹

B.E Student, Department of Computer Science and Engineering, Sir M Visvesvaraya Institute of Technology, Bengaluru, India²

Assistant Professor, Department of Computer Science and Engineering, Sir M Visvesvaraya Institute of Technology, Bengaluru, India³

ABSTRACT: It is known that email is one of the most suitable and efficient form of communication amongst the companies and organizations for internal use and also for communication with their clients or customers. Most of us often overlook how unsafe communication through email actually is – in reality email message can be as easily read by any outsider as a postal card. Hence, communication through email is not self-evident. So, emails should be secured through various techniques such as encryption of messages that can ensure its confidentiality. Many have a vague notion that emails travel from one system to other system only. In fact, the route traversed by the email to reach its destination is considerably more complicated. In simple terms, email works so that end users deploy mail servers that can store, send and transfer messages with different client programs. Mail servers in turn transmit messages through another mail servers and different proxy servers to recipient's mail server. Receivers read the messages from their mail server with the client program in use. Consequently, the email moves through various servers and not directly from computer to computer. Thus, the intent of "The CryptoMailer" is to shield email messages sent over unreliable or untrusted networks, so that snoopers can't gain access to the confidential messages or alter their contents. This may strive to be invisible to senders and highly usable to recipients. Therefore, when we encrypt all email messages as a standard practice, hackers wishing to access our personal information have a more worthwhile task in front of them. Decryption of email messages one-by-one in search of only few messages containing sensitive information may be a daunting and tedious task that even the foremost dedicated hackers may feel is not worth the effort.

KEYWORDS: Keyword Cipher Algorithm, Encryption, Decryption, SMTP, Python

I. INTRODUCTION

In view of the fact that email communication is regularly used by organizations and public bodies and in these scenarios one must be able to send and receive confidential details swiftly, so encryption of email should be an essential requisite. Nevertheless, this is often not the case, which likely has somewhat to do with ignorance. Companies do not inevitably know how risky communication through email in reality is. Confidential information leaking outside the companies will result in substantial drawback to the organizations. In worst case, the leak of information may lead to loss of revenue, reputation, penalties or even criminal proceedings. That is why it is necessary for the companies and public entities to protect personal data. Many countries have made email encryption mandatory for the companies. Therefore, investing in email encryption will soon be reimbursed. Email's appositeness for sending and receiving confidential information can be made more specific with the following questions. Do we observe whether: (a) the email we send has not been viewed? (b) someone is reading or copying our messages? (c) the messages intended to us does not reach? (d) the details of the message changes on its way to the receiver? Most of us using email communications cannot answer to the above questions based upon security of email information. This manifests how non-confidential communication through email is. So, it is the duty of the organizations to make sure that its privacy is not hampered. From the user's point of view, solution to email encryption should be easy to use along with the safety of their organization's communication.

Between the increasing fear of emails getting hacked and inspections by government, people are growing more concerned about the privacy of their inboxes, and the use of email encryption is also on the rise. Still, only 50- 60% of

emails are encrypted in transit, and client-side email encryption is infrequent still. We know that most of the email providers provide security to every user through a unique email-id and password. The problem arises if someone has illegally taken access to the email and password, then no mail server can provide any security. Security measures such as Two-factor authentication (2FA) have come up in recent times but most of the users don't implement them. So, here our main focus is to provide a platform to send and receive encrypted emails through Gmail by using a secret keyword which is an addition to the security to every user. One of the most important features that we look for when it comes to encryption of email is a high-level of security use. Encryption is either required or recommended for email compliance in all major data regulatory advice, and so it is very important to find a solution that adheres to data regulatory security standards. Another important factor is considering how easy the service is to use. Encryption can be complex, but it's important that when using an encryption solution, users can easily send encrypted email and crucially, the recipient can easily open the encrypted email. Implementing an encryption solution is an important security need, but if users avoid using it because it's complex, there isn't much point. Thus, the biggest factor informing the security of the encryption made us use the very well known "Keyword Cipher Algorithm" and how easy the service is to use made us write the entire code using Python.

A keyword cipher is a type of monoalphabetic substitution cipher. A keyword is used as a key which will determine the matching letter of the cipher alphabet to the plain alphabet. Repeated letters in the word are eliminated, then the cipher alphabet is produced with the matching keyword to A, B, C etc. until the keyword has been used up, consequently the remaining cipher text letters will be used in alphabetical order, excluding those that has been already used in the key.

Python is an object-oriented, integrated and a high-level programming language with dynamic semantics and has high level built-in data structures, combined with dynamic typing and binding, making it very attractive for rapid Application Development and for using as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax intensifying readability and hence reducing the cost of maintaining the program. Python supports modules and packages that encourages program modularity and reusability of the code. The Python interpreter along with its extensive standard libraries are available in source or binary form with zero charge for all major platforms and is freely distributed.

Here we have also used Simple Mail Transfer Protocol (SMTP) as a method to transmit mail amongst various users. SMTP is a push protocol which is used to send the emails whereas POP (post office protocol) or IMAP (internet message access protocol) are the mail access protocols at the receiver's end.

II. RELATEDWORK

As a major application field of Cryptography and network security, Email security has been an issue. According to a survey, 74% of the businesses say that email-borne cyber-attacks are having a major impact and the cost of email breaches is increasing. In this paper, a python based program known as "The CryptoMailer" is proposed to keep the confidentiality of the messages sent via Gmail. In the CryptoMailer the sender encrypts the message using the Keyword Cipher Algorithm and the receiver decrypts the message after he/she successfully receives it [1].

One of the related papers explains the working of emails and also discusses various threats in communication via emails and several security solutions related to them. This paper also explains several models and techniques used to fix and increase the security of email systems [2].

Another paper proposes how Simple Mail Transfer Protocol (SMTP) which is most widely used for email delivery has less security features like authentication, privacy and integrity of email message to make communication via emails highly secured and private. Email servers have adopted various security features along with several limitations. This paper also discussed the constraints of with respect to email security protocols and evaluates its effectiveness in email servers along with various attack methods and defense mechanisms against them [3].

To improve security and efficiency, one of the papers inferred that most email systems adopt Public Key Infrastructure (PKI) as the technique to implement security, but PKI based systems face the problem of expensive certificate management and difficulties in scalability. It also tells about Identity Based Cryptography (IBC), an another method which is based on certificate less cryptography, that uses Domain Name System (DNS) for public key exchange and a fingerprint authentication system for user's authentication. The message is encrypted by a symmetric key generated from a secret value, the public and private keys of both the sender and the receiver. This mailing system is secure against standard security model. [4].

One of the papers explains about the unawareness of the companies about the risk they are going through by communicating private messages through emails and it explains a theory on secure Emails. It also builds a theory of cryptography in which encryption is a process by which messages are encoded in such a way that only authorized user can access it. Encoded message is known as the Cipher text and the decoded message is known as the Plain text and this

process is called Decryption. For technical reasons, encryption uses a random key, which can be generated by various algorithms. This paper uses an algorithm called RSA algorithm for security related issues. [5].

III.METHODOLOGY

Under this section we include the methodology of the proposed system as shown in Fig 1 which is the use case diagram of the proposed system. Each use case shows various interactions that a user or actor experiences when using the system. The use case diagram of the proposed system consists of two major parts: first, the senders end where the message is encrypted and second, the receivers end where the encrypted message is decrypted. Encryption can be of two types: symmetric and asymmetric encryption, which means whether or not the same key is used for encryption and decryption. Here, we have used symmetric encryption that is we have used same key for both encryption and decryption. During encryption, the user has to give two inputs: First, the keyword which he/ she wishes to enter and second, the string which he/ she has to encrypt.

Plain text: A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Encrypted: L E G O A B C D F H I J K M N P Q R S T U V W X Y Z

With LEGO as the keyword, following conversion is done, all A's to L's, all B's to E's and so on. Now, encrypting the message "Health is wealth" using "lego" as the keyword. Encoded message: "Daljtdfswaljtd".

Few points to be noted during encryption are: (a) encoding of messages are done in uppercase. (b) Special characters, whitespaces and numeric values are not considered in selecting the keyword although it can be placed there. (c) During encryption of the message all special characters, numeric values, whitespaces is not affected in keyword. In order to decode the message we look for the position of the given message in encrypted text with the plain text.

Plain text: A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Encrypted: L E G O A B C D F H I J K M N P Q R S T U V W X Y Z

Message: OAGFPDARAO

Deciphered Text: DECIPHERED

Now, let us see that how we have deciphered the message? Firstly we look for 'O' in the encrypted Text and compare its position with the letter in plain text and then generate that letter. Likewise 'O' became 'D', 'A' became 'E', 'G' became 'C' and so on. In order to improve the Classical Cipher's Keyword also we can also take uppercase, lowercase and numeric values also into consideration.

Now, this encrypted message is sent by the user and a receiver receives this message using the Simple Mail Transfer Protocol (SMTP) which is an application layer protocol. Usually mail servers are listening at port 25. The sending server initiates a TCP connection to the receiving mail server. If the receiver's server is down, the sending mail will try later. If the connection is established then the client and server perform application layer handshaking. The client then indicates the email address of the sender and the receiver. Finally the client sends the message to the server over the same TCP

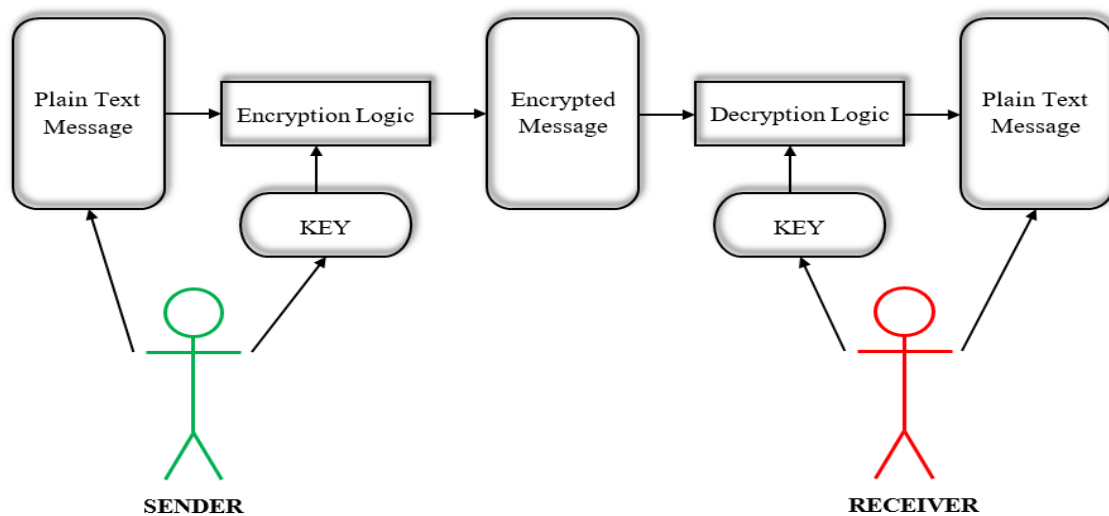


Fig 1: Use case Diagram of the Proposed

connection. The client- SMTP starts the session and the receiver-SMTP responds to the request.

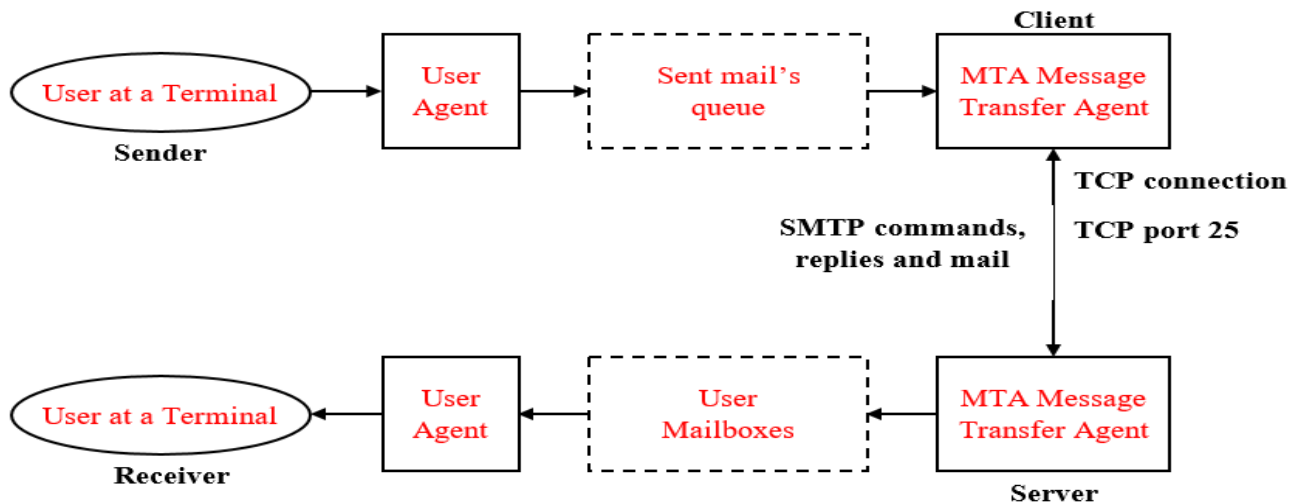


Fig 2: Block diagram of SMTP System

Fig 2 shows the block diagram of the SMTP System. The sender along with the user agent prepares the message and transfers it to the MTA whose function is to transmit the mail across the network to the recipient's MTA. The sender's system should have the client's MTA to send the mail, and the receiver's systems should have the server's MTA to receive the mail. Mails are sent by a sequence of request - response messages between the client and the server. The message that is sent across consists of a header and the body and a null line is to terminate the mail header. Everything after the null line is the body of the message consisting of sequence of ASCII characters. The actual information that is read by the recipient is contained in the body of the message. The mailboxes are checked at particular time intervals by the user agent at the server side and it informs the user about any information that is received in the mail. A short description is displayed along with each mail when the user tries to read the list of mails. In order to view the contents of the mail on the terminal, the user can select that particular mail.

IV. SOFTWARE REQUIREMENTS

OPERATING SYSTEMS: (a) Windows OS, is a series of operating system developed by Microsoft. Each version of Windows consists of a graphical user interface and a desktop to allow users to view files and folders in windows. In the past decades, Windows has become the most widely used operating system for PCs. Windows is developed to run on standard x86 hardware, such as AMD and Intel processors. Various versions of Windows that can be used to execute this program are Windows 7,8,10.

(b) We can also run this program on Linux OS which is one of the best known and most used open source operating system. Linux operating system sits underneath all of the other software on a computer by receiving requests from the programs and transferring these requests to the computer's hardware.

(c) Another OS that can be used is the Macintosh (MAC) Operating System for Apple Macintosh series of computer systems. Mac OS provides almost similar functionality and services same as Windows or Linux OS.

TEXT EDITORS: (a) Visual studio code (VS Code) is a powerful source code editor which can run on desktops having Windows, macOS and Linux operating system. It has various built in support for TypeScript, JavaScript and Node.js and has extremely nice extensions for various programming languages (such as C#, C++, Java, PHP, Python, etc.) and runtimes (such as dot net and Unity).

(b) PyCharm is an integrated development environment, specifically developed for programming in Python language and can run on computer systems having Windows, Linux or MacOS operating systems. It is developed by the Czech



company JetBrains. It provides functions like graphical debugging, code analysis, unit testing and integration with version control systems. It also supports web development with Django and Data Science with Anaconda.

INTERNET BROWSERS: In order to view the mail that has been sent or received we need internet browsers that support Gmail like: (a) Google Chrome is a cross-platform web browser developed by Google. Initially it was first released for Microsoft Windows and was later ported to Linux, macOS, iOS, and Android as the default browser built into the OS.

(b) Mozilla Firefox, or simply Firefox, developed by the Mozilla Foundation and the Mozilla Corporation, is a free open source web browser and it implements current and anticipated web standards.

(c) Safari is also a graphical web browser developed by Apple, for Desktop/Laptop computer systems having macOS and for Devices like iPads, iPods and iPhones, which run iOS. This web browser is well known for its minimalistic and simplistic interface. The users of Safari enjoy speedy internet browsing without any interruption which makes it a suitable competition to other well-known browsers like Google Chrome and Mozilla Firefox.

V. EXECUTION RESULTS AND DISCUSSIONS

The following steps are performed by the sender in order to send an encrypted message via Gmail and also by the receiver to decrypt the message that will be received from the sender:

Step 1: User sending the email via terminal.

Firstly, the user has to log in to his/her Gmail account by entering the email id and password. Once the user has successfully logged in then he/she has to enter the subject and required message that has to be sent. After this, the user is asked to enter the Keyword for encrypting the message. Once the message gets encrypted then the user enters the receiver's email id after which the email gets generated. And finally, the email message is successfully sent to the receiver's account.

```
Enter your Email ID: [REDACTED]@gmail.com
Enter your Password:
Logging in: [REDACTED]@gmail.com ...
Successfully logged in!
Enter the subject: First message
Enter the message: hello from cryptomailer
Enter the keyword:
Encrypting message ...
Message encrypted!
Enter the receiver email id: [REDACTED].testing@gmail.com
Generating mail ...
Mail sent successfully to receiver!
Closing the SMTP server.
```

Fig 3: Sending of Email



[NOTE:(a) Here, we have used “getpass” library for not displaying the password and keyword in the terminal for security purpose. (b) The email id of sending and receiving account has been hidden due to privacy reasons.]

Step 2: Inbox of the receiver

Fig 4 shows the inbox of the receivers account where we can see that the message is successfully received from the intended sender and the message is also in the encrypted form.



Fig 4: Inbox of the Receiver

[NOTE: The email id of the sender is hidden due to privacy reasons.]

Step 3:User receiving the mail via terminal

Firstly, the user has to log in to his/her Gmail account by entering the email id and password. Once the user has successfully logged in then he/she has to enter the email id of the intended sender whose message he/she wants to read. After this the count of the emails received from that particular sender is displayed followed by the encrypted message received from the sender is displayed on the terminal. In order to perform decryption of the message, the user then has to enter the keyword with which the encryption was done. On entering the keyword correctly, the receiver can see the decoded message on the terminal.

```
Enter the receiver email id: [redacted].testing@gmail.com
Enter your Password:
Logging in the receiver: [redacted].testing@gmail.com
Successfully logged in!
Enter the intended sender whose email you want to read: [redacted]@gmail.com

Email number: 1
Subject: First message
Received message: GCLLO EROM IRYPTOMAHLCR
Enter the keyword:
Decoded message: HELLO FROM CRYPTOMAILER
```

Fig 5: Receiving of Email

[NOTE:(a) Here, we have used “getpass” library for not displaying the password and keyword in the terminal for security purpose. (b) The email id of sending and receiving account has been hidden due to privacy reasons.]

VI. CONCLUSION

The CryptoMailer – sends Encrypted email which can be described as email messages which have been encoded to prevent unauthorized access to their contents. Given the frequent number of targeted attacks and impersonations through email, forward-thinking enterprises should embrace encrypted email to stop the leaking of their company’s IP. In order to describe email as a truly ‘encrypted email’, only the sender and intended recipients should have the ability to read the message. The message shouldn’t be readable by the e-mail provider or any intermediate party. With end-to-end encryption that is used in this package, encryption of the email messages is done on the sender’s device with the recipient’s public key and the decryption of the messages is done on the recipient’s device by using the private key. No server or provider in between can ever decrypt the e-mail and understand it. By using end-to-end encryption, Gmail and Microsoft OMS would be unable to read an encrypted message because these providers would not have the recipient’s private key.

The CryptoMailer:-

- Sets up in a few clicks
- Sends encrypted email to anyone
- Receives encrypted email directly in the terminal.

VII. FUTURE SCOPE

In the upcoming years also communication through emails will remain the core of communication in businesses as it is simply too fixed into the psychology and business procedures of many companies for anything else to replace it. However the way to use email will be quite different. The focus will move towards restraining the data and providing assurance to the business partners that their information is secured, and that the communication with them are unchanged and certain. Further few more modifications can be made to this package in the upcoming phases such as as- (a) the UI can be made more appreciative as per the companies need. (b) Developers can develop their own algorithms which may be easier and with more security and can implement it in place of the Keyword Cipher Algorithm. (c) Currently, the package can send and receive the emails through Gmail only, hence we can extend to all email providers. (d) Once an email account is hacked, all email attachments in the inbox are compromised and snooped, hence functions can be added to the current package which can encrypt the email attachments and secure it from unauthorized users.

REFERENCES

- [1] Haider M. Al-Mashhadi University of Basrah, College of Computer Science and Information Technology, Basrah, Iraq, “A Survey of Email Service; Attacks, Security Methods and Protocols”, International Journal of Computer Applications (0975 – 8887) Volume 162 – No 11, March 2017.
- [2] K.Sivaraman, P.Arumugam, Assistant Professor, Department of CSE, BIST, BIHER, Bharath University, Chennai, “The Security Related to Electronic Mail”, International Journal of Pure and Applied Mathematics Volume 119 No.12 2018, 9665-9671
- [3] Suresh Kumar Balakrishnan and V. P. Jagathy Raj, School of Computer and Information Sciences, Indira Gandhi National Open University (IGNOU) 1 New Delhi 110068, India, “Practical Implementation of a Secure Email System Using Certificate less Cryptography and Domain Name System”, International Journal of Network Security, Vol.18, No.1, PP.99-107, Jan. 2016.
- [4] Niti Sharma, Student, Computer Science and Engineering, Delhi Institute of Technology Management & Research, Faridabad, India, “Secure Mailing System”, International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE), Vol. 6, Issue 4, April 2017.
- [5] Fatima Aziz Rawdhan, MahmoodKhalel Ibrahim, “Enhancement of Email Security Services”, International Journal of Scientific & Engineering Research, Volume 8, Issue 1, January-2017
- [6] Munish Mehta, Vijay Goyar, Vishnu Bairwa, “Security and Authentication through Text Encryption and Decryption based on Substitution Method”, International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-8-, Issue-9S, July 2019.