ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 5, Issue 3, March 2018

Onion-Line Complaint Bot

[1] Divya Vadhyar, [2] Parnasi Meher, [3] Sakshi Mhatre, [4] Vijaya Sagvekar
[1] Padmabhushan Vasantdada Patil Pratishthan's College of Engineering, Mumbai University.
[2] Padmabhushan Vasantdada Patil Pratishthan's College of Engineering, Mumbai University.
[3] Padmabhushan Vasantdada Patil Pratishthan's College of Engineering, Mumbai University. [4] Professor

***Abstract:* - Complaint Registrations becomes chaotic when done manually. There is a great possibility that the complainant's identity is revealed to the accused and his/her security is compromised. It is possible that the complaint has just been registered and no action is taken to improve the condition. In such cases, there is no way to keep constant track of the same. Hence arises a great need to implement a digital way to implement the same. Onion-Line Complaint Bot allows user to register their complains without revealing their identity and also allows higher officials to check the status of the complaint.**

***Keywords*- Onion routing algorithm, Complain Portal, Tor Browser, Tor Websites, Anonymity of user;**

I. INTRODUCTION

In the traditional methods, the complaint box is installed manually at various locations where the complainants are supposed to travel upto and put in their written complains. The existing system is completely paper based. Traditionally file system was used to maintain the details of the citizen and the complaints they registered. The existing system requires personal visit to the office and registering complaints on paper, which is very time consuming and requires a lot of man-power. The government provides online platform for all its activities, but unfortunately the interface is not user-friendly. The proposed system creates a user-friendly interface using web technology. The people need not go to the higher authorities always when they face problems. They can use the service of this software and can register their complaint and the complaint is taken up by the employee of specified department and he solves the problem.

The main objective was to create a user-friendly online interface for citizens to communicate with administrative body and, reduce the distance and time barrier between citizens and administration as well as to encourage the citizens to actively participate in city administration, in order to bring transparency and flexibility in system. This method has many drawbacks as the complainant may have to travel and it can be time consuming so he/she may decide not to complain or delay their complaint. Secondly there is a high possibility that the complainant can be identified and threatened. The complaint box is a physical object unlike a virtual network or a database so it can be damaged, tampered or harmed. Moreover there is no evidence of the given complain, as nothing is registered and there is no assurance that help will be provided or actions would be taken. No liability is maintained.

In the proposed system, the citizens as per their wish can register in the portal or skip the registration. Following which she/he can select the intended department or complain in general. Accordingly the citizen can lodge their complain maintaining their identity or by maintaining complete anonymity. The complain is thereby registered and can be viewed in the pool of complains by the officials. The complains are prioritized according to keywords and tags. Considering as per vocabulary, complains containing keywords such as blasts, terrorism and other threats will be ranked and displayed at the top and accordingly lower rankings would be given to other complain. Further we plan to differentiate complains on the basis of national and individual interest.

Further, the officials can send the generated report of the complaints and cases solved to their higher official. The higher officials can also personally view the status of the complaints. To maintain anonymity, we will implement onion routing. Onion routing promises to protect the integrity and confidentiality of data from the theft, eaves dropping over the network and internet, onion routing proceed a devised a technique to limit the knowledge of information as possible while high level of anonymity is achievable. It initiates a communication with an application specific router called onion routing proxy that was enough to manage TCP and Sock request of the client. Routing onion is a data structure designed by wrapping a plain text message with the successive layer of encryption such that each layer can be unwrapped by an one intermediary and no other can decrypt it. Onion routing is implemented with the help of encryption in the application layer of network in the communication stack like the layer of an onion.

Onion routers keep track of received onions until they expire. Replayed or expired onions are not for- warded, so they cannot be used to uncover route information, either by outsiders or compromised onion .The verb to encrypt means the application of a cryptographic operation, be it encryption or decryption. Note that clock skew between onion routers can only cause an onion router to reject a fresh onion or to keep track of processed onions longer than necessary. Also, since data is encrypted using stream ciphers, replayed data will look different each time it passes through a properly operating onion router. Although we call this system onion routing, the routing that occurs here does so at the application layer of the protocol stack and not at the IP layer. More specifically, we rely upon IP routing to route data passed through the longstanding socket connections.

An anonymous connection is comprised of portions of several linked longstanding multiplexed socket connections. Therefore, although the series of onion routers in an anonymous connection is fixed for the lifetime of that anonymous connection, the route that data actually travels between individual onion routers is determined by the underlying IP network. Thus, onion routing may be compared to loose source routing.

The most suitable example prevailing today is of the Tor web browser that provides a variety of services. Tor is free software for enabling anonymous communication. The name is derived from an acronym for the original software project name "The Onion Router".[8][9] Tor directs Internet traffic through a free, worldwide, volunteer overlay network consisting of more than seven thousand relays[10] to conceal a user's location and usage from anyone conducting network surveillance or traffic analysis. Using Tor makes it more difficult for Internet activity to be traced back to the user: this includes "visits to Web sites, online posts, instant messages, and other communication forms". [11] Tor's use is intended to protect the personal privacy of users, as well as their freedom and ability to conduct confidential communication by keeping their Internet activities from being monitored. Tor does not prevent an online service from determining when it is being accessed through Tor. Tor protects a user's privacy, but does not hide the fact that Tor is being used. Some websites restrict what is allowed when using Tor. For example, Wikipedia limits the edits that can be made through Tor.[12] Tor help in encryption of original data including the IP address and send to the destination through a virtual circuit comprising successive, randomly selected. Each relay decrypts the layer of encryption to obtain only the successive relay in order to transmit the data. The final relay decrypts the innermost layer of encryption and sends the original data to its final destination without revealing and hiding the information of sender

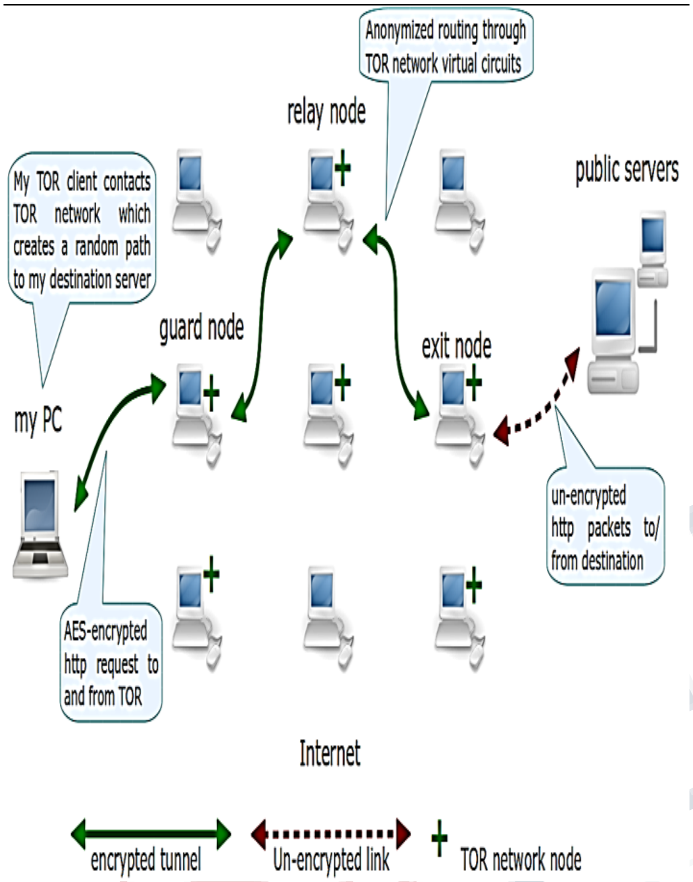
TOR is the descendant of the onion routing project work by the project had many concept in it. TOR is a collection of onion routers which may have different functionality and roles in the network and during the network communication they perform their roles. Each router send an information in a secure way to next hop in the TOR network connection whereby if any single nodes is compromised then this been will be not affected anonymity as well as data communication send to and from the sender and receiver is work properly. TOR main aim is to hide the communication between the initiator and the target host fir which the initiator needs to communicate with the nodes [3].

The Tor network is an network in which each onion router runs as a normal and perform their usual duties without having any special type of privileges. A TLS connection is maintain for every other onion router. Each user can fetch their directories and establish circuit across the network and handle difficulties in handling connection from user application. Each router in the Tor maintain a long term identity key and short term identity key that is use to sign as TLS certification Salt is not only a single hash function, it is all about using more than one hash function among more the one hash function. Salt is the process of selecting a unique hash function from many hash function that are also known to server. Salt is also be added to make it more difficult from an attacker to break in a system by using password hash matching strategies because adding salt to a password hash prevent an attacker from testing known dictionary words across the entire system. Salt can also be added to make it more difficult for an attacker to break into a system by using password hash-matching strategies because adding salt to a password hash prevents an attacker from testing known dictionary words across the entire system.

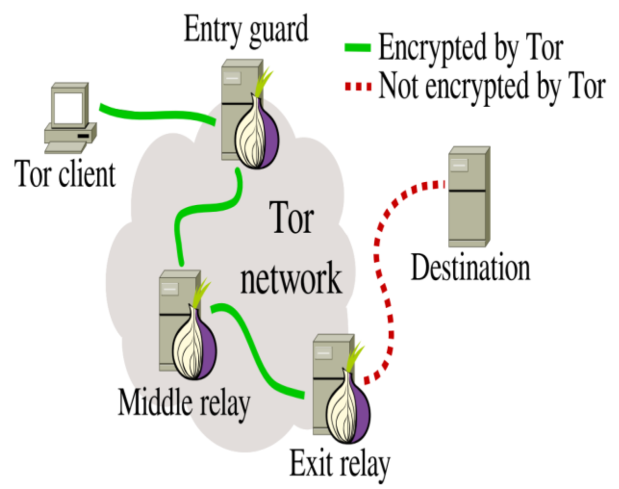
Hash = (salt + password)
Verifier = salt + hash (salt + password).

a. Figures

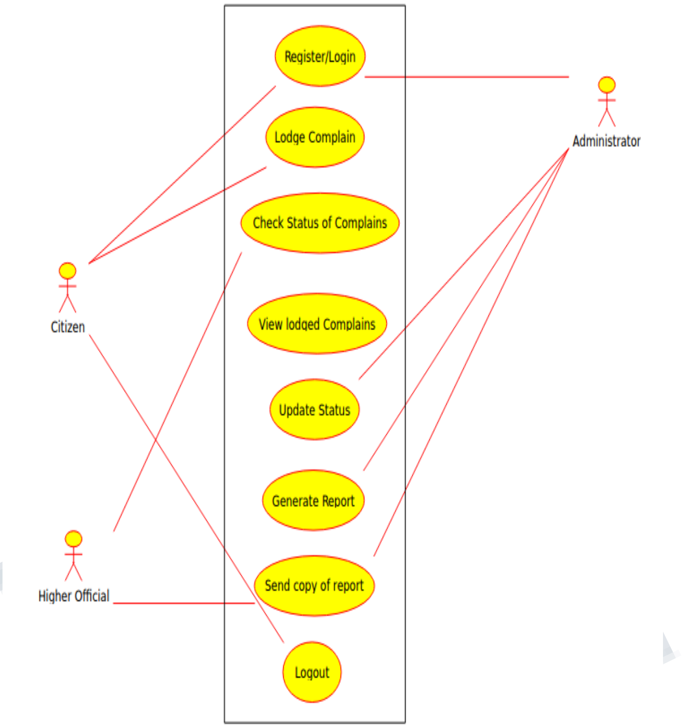
2.1 Architecture of interaction between Tor clients and the Tor Network


Anonymized routing through TOR network virtual circuits
relay node
public servers
My TOR client contacts TOR network which creates a random path to my destination server
guard node
exit node
my PC
un-encrypted http packets to/ from destination
AES-encrypted http request to and from TOR
Internet
encrypted tunnel
Un-encrypted link
TOR network node

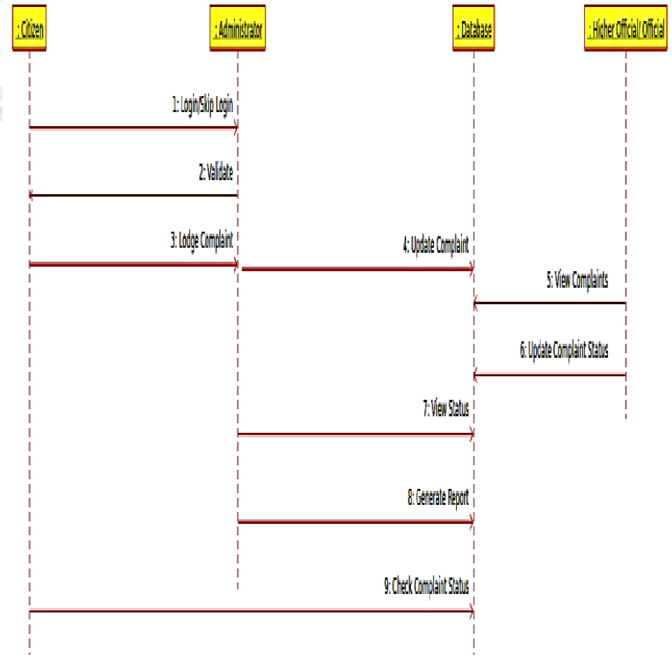
2.2 Software architecture of Tor browser



2.3 Figure depicting actors involved and their functionalities in the system:



2.4 The sequential working of the complaint system



III. IMPLEMENTATION

The implementation of the proposed system involves the following steps:

1. Building a web browser like Tor
2. Developing a complaint registration website
3. Deploying the website through the built browser.
4. Implementing anonymity

1. Developing a web browser

Tor is initialized by calling the InitializeTor function which is an exact copy from the Tor.NET sample application.

Handlers are used to make CefSharp to "behave". Here is a list of the CefSharp handlers and what they do:

DownloadHandler: Tells the application when a new download starts and about the progress of the ongoing downloads.

KeyboardHandler: When the focus is on the CefSharp browser, it "eats" all the keys pressed by the user. The application uses Ctrl-F4 to close the active browser tab so this handler helps to inform the application that Ctrl-F4 was pressed in the browser.

LifeSpanHandler: CefSharp calls the OnBeforePopup function of this handler before it opens a pop-up window and it tells CefSharp to open it in a new tab and not in a seperate window.

MenuHandler: I added the items from the Chrome Browser which I use most to the browser context menu. This handler does that stuff. An interesting extra feature of this context menu is the "Save as Pdf" option which does not exist in Google Chrome.

SchemeHandler: This handler helps to load the web pages that start with "chrome://".

2. Developing a tor like website

Though the widely used Tor anonymity network is designed to enable low-latency anonymous communication, interactive communications on Tor incur latencies over 5 × greater than on the direct Internet path, and in many cases, autonomous systems (ASs) can compromise anonymity via correlations of network traffic. In this paper, we develop LASTor, a new Tor client that addresses these shortcomings in Tor with only client-side modifications. First, LASTor improves communication latencies by accounting for the inferred locations of Tor relays while choosing paths. Since the preference for shorter paths reduces the entropy of path selection, we design LASTor so that a user can choose an appropriate tradeoff between latency and anonymity. Second, we develop an efficient and accurate algorithm to identify paths on which an AS can compromise anonymity by traffic correlation. LASTor avoids such paths to improve a user's anonymity, and the low runtime of the algorithm ensures that the impact on end-to-end communication latencies is low. Our results show that, in comparison to the default Tor client, LASTor reduces median latencies by 25% while also reducing the false negative rate of not detecting a potential snooping AS from 57% to 11%.

3. Hosting Website

4. Implementing anonymity

All this process is implemented to provide security in avoiding data modification at the end of server side. For the same purpose two different servers are made and maintained one for storage server for storing user data file and second is for rehashing user password.

I: Implementing the encryption algorithms with salt.
II: Connection Establishment.
III: Data transfer

The first step starts with implementation onion routing encryption algorithms adding salt in it. There are many different algorithms are used for connection establishment and data transferring. RSA algorithm is used for establishing connection. It is the standard public key cryptography algorithm and cipher text are not easily decrypted, because the process of decryption is not an inverse process of encryption.. Random prime numbers are generated for encryption and decryption. Using system time onion key is generated. While sending the onion keys are generated that made difficult to predict the keys.

TCP socket connection is used for connection. For anonymous communication and private is to be performing first. So the path that is to be followed by the sender and receiver and the address of the proxies through they pass during connection.

The first layer of onion decrypted at its intermediated proxy and appropriate details such keys, IP address and the function for decrypting the data that will be build into routing table.

As connection is established over the network and data start passing through it in encrypted form of the onion[5]. finally the data is sent to receiver send by the sender by encrypting at each level of intermediate proxies and finally it decrypted at the initiating proxy and serves as plain text the sender.

A. Execution Steps:
1. Start Connection
2. Data Encryption with salt
3. Key Exchange – Diffie Hellman
4. Network communication using ToR
5. Files Transferred to router
6. Connection End.

IV. CONCLUSION

The onion-line registration bot is an advanced way of registering complaints by causing minimal interference in the complainant's routine.

It ensures complete anonymity of the complainant.

Tracking the complaint status is also feasible for the officials.

The officials can send the generated report to their higher subordinates.

REFERENCES

[1]http://www.ijcta.com/ documents/volumes /vol3issue2 /ijcta2012030232.pdf

[2]http://www.ijesit.com /Volume%205 /Issue %202 /IJESIT201602_15.pdf

[3] International Journal of Scientific and Research Publications, Volume 5, Issue 7, July 2015 1 ISSN 2250-3153

[4] https://www.onion-router.net /Publications /JSAC-1998.pdf