Desing and Implementation of Distributed Security using Onion Routing

Ashish T. Bhole
Assistant Professor, Computer Engg.dept.
SSBT'S COET
Jalgoan (M.S), India

Savita H. Lambole
M.E (CSE, Sem- IV),
Computer Engg.dept.
SSBT'S COET
Jalgoan (M.S), India

ABSTRACT

Although methods for reaching security goals such as secrecy, integrity and authentication are widely used in the Internet, there is no widely-used solution providing anonymity. Onion Routing is a flexible communication infrastructure and one such application which enables users to have anonymous communication and yet is so reliable from eavesdroppers and traffic analyzers. First it securely establishes the connection. To ensure the security well known networking and Public key Cryptographic techniques are utilized. Here the identities of the sender and the receiver are hidden by an onion structure, which is cryptographically layered data structure that defines the route through the onion routing network. After the route is established by making the entries into the routing table, the data is transmitted over the channel, which is also repeatedly encrypted. Once the data is transferred the connection is destroyed. Onion routing provides the efficient way of protection, which we have implemented.

Keywords

Anonymous communication, Onion routing, Anonymity, TOR, Security, Cryptography, RSA, OR.

1. INTRODUCTION

Onion routing was conceived in 1996 by David.M.Goldschlag, Michael.G.Reed and Paul.F.Syverson for the Naval Research Laboratory's research group in high assurance system [11]. It lives just beneath the application layer and is designed to interface with a wide variety of unmodified internet services by means of proxies. Onion routing is the mechanism in which the sender (initiator) and the receiver (responder) nodes communicate with each other anonymously by means of some anonymous intermediate nodes called as onion routers. It protects against traffic analysis and makes it very hard for an eavesdropper to determine who is talking to whom over the network. It concentrates on encrypting the packet header in such a way that only the intended destination understands that the packet is meant for him Onion routing is a type of anonymous communication that creates cryptographic circuits along an unpredictable route through a network of nodes called onion routers and passes traffic bidirectionally along those circuits with minimal latency [2]. Onion Routing is a distributed overlay network designed to anonymize TCP-based applications like web browsing, secure, shell, and instant messaging. Clients choose a path through the network and build a *circuit*, in which each node (or "onion router" or "OR") in the path knows its predecessor and successor, but no other nodes in the circuit. Traffic flows down the circuit in fixed-size *cells*, which are unwrapped by a symmetric key at each node (like the layers of an onion) and relayed downstream [3]. Onion routing is widely used, and there are several similar protocols in use as well. The anonymity of the protocol is affected by the timing properties of the network, including timing delays on communication links, congestion at routers, and inaccuracies in local clocks. It also depends on user behaviour, such as how users choose destinations and the pattern of their traffic. The nature of the adversary is a factor as well. He may control small parts of the network, may be able to make observations at many points in the network, or may be able to move around the network during the operation of the protocol. We construct a model of the protocol that makes some concrete choices on these questions[5][6].

Existing System:

In the current scenario in order to connect to the internet the user must use a network. In this way the user can be traced back. So as we can see there is no privacy in this type of communication. In this way a third party can use this property to hack the information that the client wishes to keep the secret. In other ways the current network are prone to eavesdropping. In order to avoid these disadvantages the TOR network was developed for people who want to maintain their identity a secret. The idea of onion routing is to protect the privacy of the sender and recipient of a message, while also providing protection for message content as it traverses a network. Here we implement onion routing network. [5][6]

- ***Proposed System :***

In the proposed system we implement a TOR network that provides the user with the advantage of privacy and security. The goal of Onion Routing (OR) is to protect the privacy of the sender and recipient of a message, while also providing protection for message content as it traverses a network of Onion Routers. The advantage of Onion Routing is that it is not necessary to trust each cooperating Router; if one or more router is compromised, anonymous communication can still be achieved. This is due to the fact that each Router in an OR network accepts messages, re-encrypts them, and transmits to another Onion Router. An attacker with the ability to monitor every Onion Router in a network might be able to trace the path of a message through the network, but an attacker with more limited capabilities will have difficulty even if he or she controls one or more Onion Routers on the message's path.

Onion routing is a technique for anonymous communication over a computer network. Messages are repeatedly encrypted and then sent through several network nodes called onion routers. Each onion router removes a layer of encryption to uncover routing instructions, and sends the message to the next router where this is repeated. This prevents these intermediary nodes from knowing the origin, destination, and contents of the message. The Tor Network is a low-latency anonymity, privacy, and censorship resistance network whose servers are run by volunteers around the Internet. This distribution of trust creates resilience in the face of compromise and censorship; but it also creates performance, security, and usability issues. The Tor Flow suite attempts to address this by providing a library and associated tools for measuring Tor nodes for reliability, capacity and integrity, with the ultimate goal of feeding these measurements back into the Tor directory authorities [12].

Our objective in this paper is to analyze the concept of onion routing and to implement it for LAN. Onion Routing is a general-purpose infrastructure to support Private and anonymous communication over a public network. Preserving privacy not only means hiding messages sent, but also who is talking to whom (Traffic analysis) [11]

2. ONION ROUTING OVERVIEW

Onion Routing is flexible, communication infrastructure that is resistant to both eavesdropping and traffic analysis. Onion Routing accomplishes this goal by separating identification from routing. It is a bi-directional, near real-time, and can be used for both setup and data movement. In the phase of connection setup an onion is formed. An onion is a cryptographically layered data structure that defines the route through the onion routing connection based and connectionless traffic. To use a network of onion routers, users randomly choose a path through the network and construct a circuit—a sequence of nodes which will route traffic.[4] Although onion routing may be used for anonymous communication, it differs from anonymous remailers in two ways: Communication is real time and bidirectional, and the anonymous connections are application independent. Onion routing's anonymous connections can support anonymous mail as well as other applications.[6]

Onion routing has two main phases: Connection network. This onion is used as the destination address by onion routers in setting up the connection Onions themselves appear differently to each onion router as well as to network observers. After the connection is established, data is sent, which is repeatedly pre-encrypted with the keys that were carried in the onion as data moves through the anonymous connection, each onion router removes one layer of encryption, so it finally arrives as plaintext. This layering occurs in the reverse order for data moving backward. In this manner, data appears differently to both onion routers and network observers as it traverses the connection.[1]

The primary innovation in onion routing is the concept of the routing onion. Routing onions are data structures used to create paths through which many messages can be transmitted. To create an onion, the router at the head of a transmission selects a number of onion routers at random and generates a message for each one, providing it with symmetric keys for decrypting messages, and instructing it which router will be next in the path. Each of these messages, and the messages intended for subsequent routers, is encrypted with the corresponding router's public key. This provides a layered structure, in which it is necessary to decrypt all outer layers of the onion in order to reach an inner layer. The onion metaphor describes the concept of such a data structure. [13].

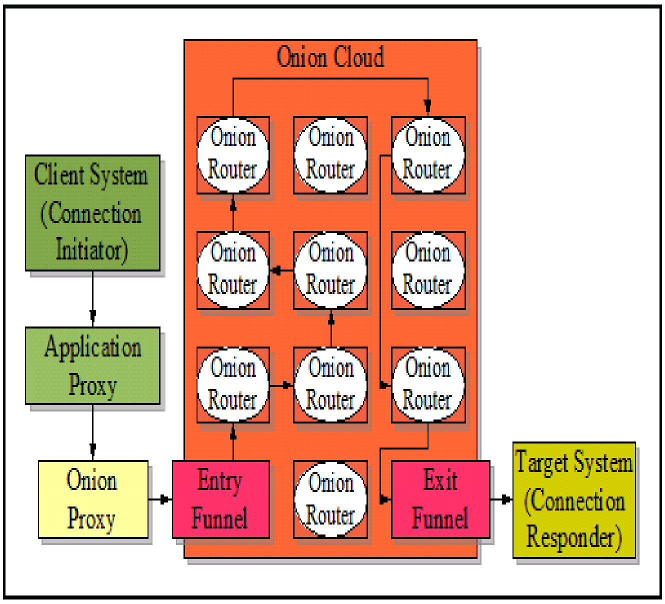


Figure 1: Onion routing path

2.1 Onion Routing Topology

In onion routing, instead of making socket connections directly to a responding machine, initiating applications make connections through a sequence of machines called onion routers.

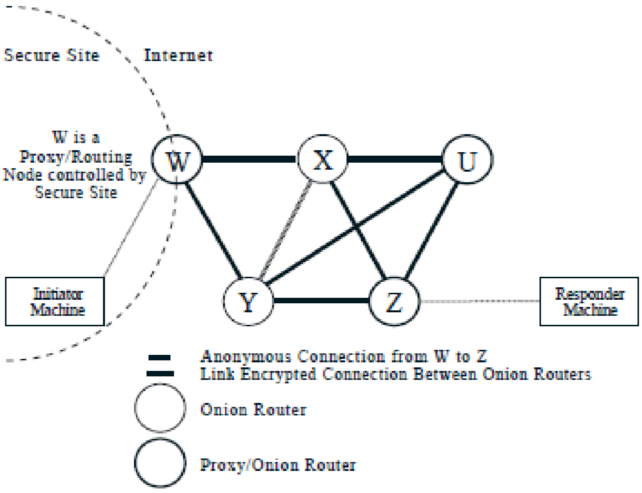

Secure Site
Internet
W is a Proxy/Routing Node controlled by Secure Site
W
X
U
Y
Z
Initiator Machine
Responder Machine
Anonymous Connection from W to Z
Link Encrypted Connection Between Onion Routers
Onion Router
Proxy/Onion Router

Figure 2: Routing Topology

The onion routing network allows the connection between the initiator and responder to remain anonymous. We call this an anonymous socket connection or anonymous connection. Anonymous connections hide who is connected to whom, and for what purpose, from both outside eavesdroppers and compromised onion routers. If anonymity is also desired then all identifying information must be removed from the data stream before being sent over the anonymous connection. We call the onion routing network topology that we use in this paper the basic configuration. This is illustrated in figure. 2 [5][6].

3. ONION ROUTING SPECIFICS:

Onion routing consists of the following steps:

- Defining a route.
- Constructing an anonymous connection.
- Moving data through an anonymous connection.
- Destroying the anonymous connection.

When an onion proxy receives a message it first randomly selects a set of onion routers up to the destination by checking in its existing list of onion routers. It then uses public key cryptography to construct the onions in such a way that only the intended onion routers can peel off the outer layer. The following example illustrates the process.

Consider the case where there are n onion routers numbered from 1 to n. The public and the private key of a particular router say i is denoted by Ipu and Ipr respectively. The onion proxy knows the public keys of all the onion routers in its list. The private keys are known only to that particular router. An encryption and decryption function is used to encrypt and decrypt the data. The encryption function is Ekey (data) and the decryption function is Dkey (data). Data encrypted by a public key can be decrypted by a private key and vice versa. Hence we have DI_public_key (EI_private_key (data)) = data. DI_private_key (EI_public_key (data)) = data [8][9]

On receiving a packet the onion proxy selects a random sequence of routers from its list say 4, 3 and 5. So it constructs the onion in the following manner. It first encrypts the data packet with public key of 5 followed by public key of 3 and finally 4. So the encrypted data now looks like E4pu (3’s IP address, E3pu ((5’s IP address, (E5pu (recipient’s IP address, data))))). This is then sent to onion router 4. Onion router 4 uses its private key to peel the outermost encryption layer. It finds the IP address of the next hop i.e. router 3. So it passes to router 3 the onion which now looks like E3pu ((5’s IP address, (E5pu (recipient’s IP address, data)))). Again router 5 uses its private key to peel the outermost encryption layer. It finds the data and the recipient’s IP address and concludes that it is the final anonymous hop to the destination. It simply forwards the packet to the destination.

Thus sending an onion over a chosen path creates a virtual circuit. This circuit is bidirectional i.e. the destination can also send a message to the source along the same path. In the given example it simply encrypts the data with its private key and forwards it to onion router 5. Erecipient_private (IP address, data). Onion router 5 then encrypts it with its private key and forwards it to 3 as E5pr (Erecipient_private (IP address, data)). Similarly router 3 and 4 also encrypt it step by step with their private key and outer 4 sends it to the onion proxy that initiated connection with it. The data that is received by the onion proxy looks like E4pr (E3pr (E5pr (Erecipient_private (IP address, data)))). The onion proxy now uses the public keys of these routers and decrypts each layer of the onion, using the outermost layers key first. It retrieves the data and simply routes it to the sender. Since the size of the onion reduces as it nears the destination an attacker can infer details about the destination. To avoid this onions are padded at each onion router to maintain the size of the onion. Padding is simply adding redundancy.

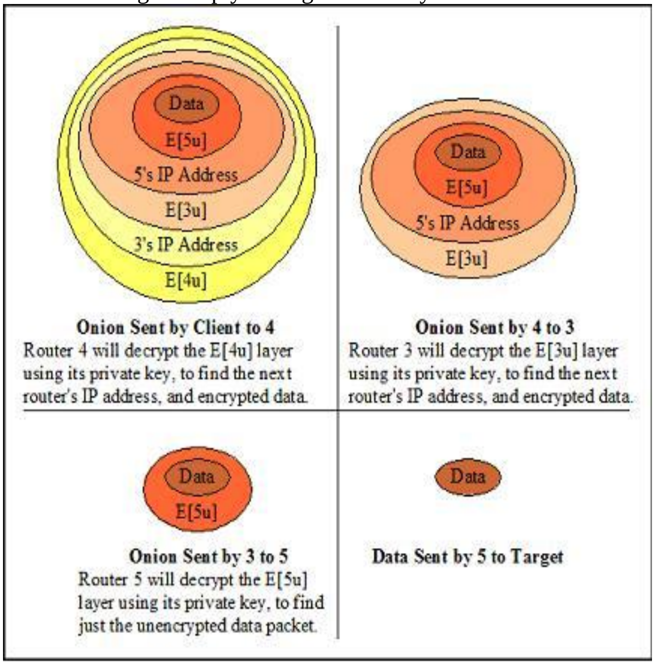

Data
E[5u]
5's IP Address
E[3u]
3's IP Address
E[4u]
Onion Sent by Client to 4
Router 4 will decrypt the E[4u] layer using its private key, to find the next router's IP address, and encrypted data.
Data
E[5u]
5's IP Address
E[3u]
Onion Sent by 4 to 3
Router 3 will decrypt the E[3u] layer using its private key, to find the next router's IP address, and encrypted data.
Data
E[5u]
Onion Sent by 3 to 5
Router 5 will decrypt the E[5u] layer using its private key, to find just the unencrypted data packet.
Data
Data Sent by 5 to Target

Figure 3: Structure of onion at intermediate nodes

This is a really big advantage because it complicates traffic analysis, as an attacker cannot infer location or other details of the destination by getting hold of an onion. Every onion router has details of only its previous and next hop. So even if an onion router has been compromised the attacker can only get the encrypted onion with the next hop. He will not be able to decrypt the onion without the private keys and hence will not infer any valuable information from it[5][6][7]

Since the size of the onion reduces as it nears the destination an attacker can infer details about the destination. To avoid this onions are padded at each onion router to maintain the size of the onion. Padding is simply adding redundancy. This is a really big advantage because it complicates traffic analysis, as an attacker cannot infer location or other details of the destination by getting hold of an onion. Every onion router has details of only its previous and next hop. So even if an onion router has been compromised the attacker can only get the encrypted onion with the next hop. He will not be able to decrypt the onion without the private keys and hence will not infer any valuable information from it.[5][6][7].

Each layer of onion also contains an expiration time. An onion router is to ignore expired and replayed onions. Further if the connection breaks during the routing process then all the onion routers are informed via a destroy message. Ensuring that all onion are of the same size, timing information of the circuit is obfuscated and adding noise makes traffic analysis very difficult.

3.1 Attacks

There are two types of attacks [7][8]: passive and active ones. Passive here means that the attacker has no own nodes in the network and cannot modify messages, but can only listen to traffic. Assuming that an attacker can listen to all the traffic might be paranoid if you consider for instance the Internet. But still a government or intelligence organizations might record a huge amount of Internet traffic or get access to particular interesting traffic, for example the communication of one person. For active attacks the attacker modifies messages and controls a fraction of the nodes. One of the ideas of onion routing is that the user does not trust a single server but uses a route of nodes and assumes that at least one of them is not compromised. It has to be assumed that an attacker can only control a fraction of nodes such that the chance that all nodes of one route are controlled by her is small enough. But even with only a small fraction of nodes there are some possible attacks. The content a user sends could also possibly tell something about her identity. Examples for this are information a browser sends to a website like referrers or cookies

It is susceptible to denial of service attacks. This can be done by forcing onion routers to do a large number of cryptographic operations by many sending packets to it. Eventually the router simply ends up doing cryptographic operations and is not able to forward packets. This can be mitigated using client puzzles. Here the onion proxy (i.e. the server) forces a requesting client to complete a puzzle before it allocates resources. This forces an attacker to find additional resources. But puzzle solving has an impact on the latency although it reduces DOS vulnerability. [9]An attacker can record data going on between routers and can compromise a router at a later stage, to acquire private key and decrypt data. This can be avoided by using a session key between communicating parties. The session key is used to encrypt data and is valid only for the duration of the communication. The benefits of onion routing are protection against corrupt nodes, and protection of origin, destination, and message content along the path. Its drawbacks include the necessity of end-to-end encryption for full protection and performance.

4. GENERAL DESIGN

4.1 RSA Algorithm

Public-key cryptography, also known as asymmetric cryptography, uses different keys for encryption and decryption. In order to encrypt data, a pair of cryptographic keys has to be generated once, a public and a private key. The private key is kept secret and is used for decryption. The public key is published and can be used by anyone who wants to send an encrypted message to the person owning the private key. Another mode of operation is used for message signing. If a message is signed with a private key, anyone in the possession of the corresponding public key can verify its origin and authenticity. OR uses RSA cryptography for the tunnel setup

RSA involves a **public key** and a **private key.** The public key can be known to everyone and is used for encrypting messages. Messages encrypted with the public key can only be decrypted using the private key. The RSA algorithm used to generating keys for encryption and decryption in the implementation of secure distributed onion routing protocol[10] [12]

4.2 Key Generation Algorithm

1. Generate two large random primes, *p* and *q*, of approximately equal size such that their product n = pq is of the required bit length, e.g. 1024 bits.
2. Compute n = pq and (φ) phi = (p-1) (q-1).
3. Choose an integer *e*, 1 < e < phi, such that gcd (e, phi) = 1.
4. Compute the secret exponent *d*, 1 < d < phi, such that ed ≡ 1 (mod phi).
5. The public key is (n, e) and the private key is (n, d). Keep all the values d, p, q and phi secret.

- n is known as the *modulus*.
- e is known as the *public exponent* or *encryption exponent* or just the *exponent*.
- d is known as the *secret exponent* or *decryption exponent*.

4.2.1 Encryption:

Sender A does the following:-

1. Obtains the recipient B's public key (n, e).

Represents the plaintext message as a positive integer *m*.

2. Computes the cipher text c = m^e mod n.
3. Sends the cipher text *c* to

4.2.2 Decryption

Recipient B does the following:-

1. Uses his private key (n, d) to compute m = c^d mod n.
2. Extracts the plaintext from the message representative *m*.

4.3 Key length

When we talk about the *key length* of an RSA key, we are referring to the length of the modulus, *n*, in bits. The minimum recommended key length for a secure RSA transmission is currently 1024 bits. A key length of 512 bits is now no longer considered secure, although cracking it is still not a trivial task for the likes of you and me. The longer your information is needed to be kept secure, the longer the key you should use.

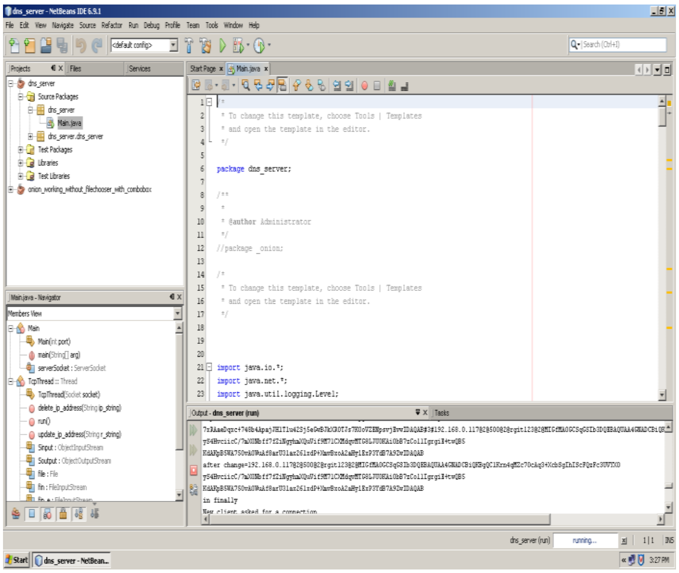
4.4 Construction of Onion

Once the route has been defined, corresponding number of key seed materials are generated corresponding to every node in the route. First the standard structure is encrypted with the onion Key of the last node in the route obtained by hashing its key seed material once with SHA1. In the current implementation only DES encryption is used for encryption with the keys generated. The onion layer corresponding to the last node which is of 28 bytes is appended with a random

string of 100 bytes to bring it to a total length of 128 bytes = 1024 bits and then encrypted with the public key of that particular router using RSA algorithm with a block size of 1024. Note that these public keys are assumed to be known beforehand. The encrypted message is made up to size of 1024 bytes and appended before the encrypted standard structure to form the onion at first step. This onion is encrypted with onion key of last but one node in the route and then appended at the end of RSA encrypted onion layer corresponding to last but one node. This procedure continues till the onion is formed by adding onion layers for all the nodes in the route. This onion is then broken into cells and transmitted to the node. A router on receiving the corresponding
cells of the onion, combines them to reconstruct the onion, decrypts the onion layer with its private key using RSA algorithm. Then decrypts the remaining onion from the hash of key seed material obtained in the onion layer. Adds a padding of arbitrary string to make up for the decreased size of onion due to stripping of onion layer and then sends this padded onion to the next node in the route.

5. RESULTS

An implementation of the project in java tested on Netbeans6.9.1. The different modules which we have implemented for the secure distributed onion routing are OnionApp, KeyGenrator, IpAddress, dns-server, clientApp, severApp, KeyReader. Expected results of the distributed security using onion routing are:



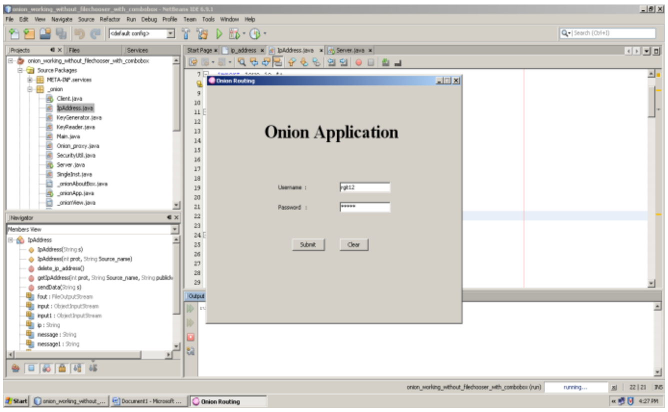
```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package dns_server;

/**
 *
 * @author Administrator
 */
//package _onion;

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */



import java.io.*;
import java.net.*;
import java.util.logging.Level;
```

```
7xRAaeDqxc+748b4ApajJHlTlu42Sj5eGwBJkXR0TJs7R0oVZENpsvjBvwIDAQAB#3#192.168.0.117@2@500@2@rgit123@2@MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQK
yS4HvciicC/7mX0Mbff7f2iNgyhmXQuVif9M71CXMdgvMTG8LJU0KAi0bB7zCollIgrgiH+twQB5
KdAKpB5MA750vA0WuAf8arU31ax261xdP+XmvBxoA2aHy1ExP3YdB7A92wIDAQAB
after change=192.168.0.117@2@500@2@rgit123@2@MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC1Krn4qMZc70cAq3+XcbSgIhIScFQxFc3UVYX0
yS4HvciicC/7mX0Mbff7f2iNgyhmXQuVif9M71CXMdgvMTG8LJU0KAi0bB7zCollIgrgiH+twQB5
KdAKpB5MA750vA0WuAf8arU31ax261xdP+XmvBxoA2aHy1ExP3YdB7A92wIDAQAB
in finally
New client asked for a connection
```

Figure 4: Running DNS Server with encryption results

Step 1: First run DNS Server on one machine using Netbeans

Step 2: Run Onion Application on different machine in network

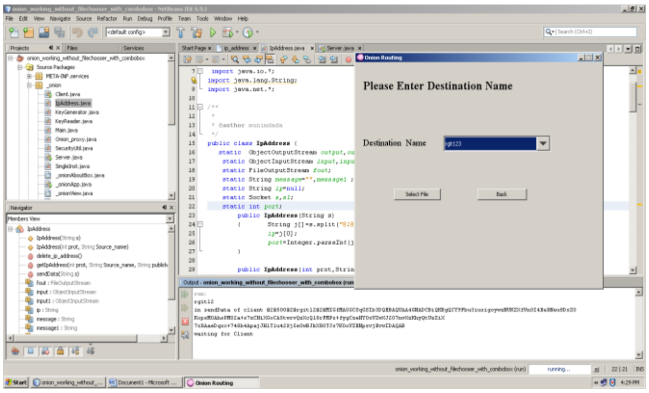


Onion Application

Username : rgit12

Password : *****

Figure 5: Running Onion Routers to share files



Please Enter Destination Name

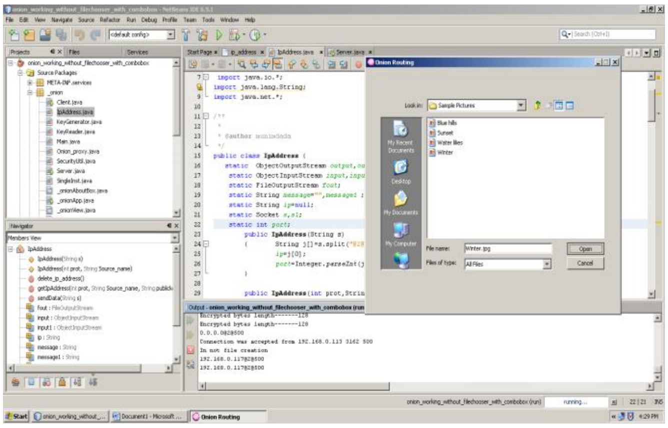
Destination Name rgit123

```
import java.io.*;
import java.lang.String;
import java.net.*;

/**
 *
 * @author [illegible]
 */
public class IpAddress {
    static ObjectOutputStream output,ou
    static ObjectInputStream input,inpu
    static FileOutputStream fout;
    static String message="",message1 ;
    static String ip=null;
    static Socket s,s1;
    static int port;
        public IpAddress(String s)
        {       String j[]=s.split("@2@
                ip=j[0];
                port=Integer.parseInt(j
        }

        public IpAddress(int prot,Strin
```

```
run:
rgit12
in sendData of client @2@500@2@rgit12@2@MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCY9[illegible]
[illegible]
7xRAaeDqxc+748b4ApajJHlTlu42Sj5eGwBJkXR0TJs7R0oVZENpsvjBvwIDAQAB
waiting for Client
```

Figure 6: Share files with destination

Step 3: Share the files between the choosen terminal which are encrypted



Look in: Sample Pictures

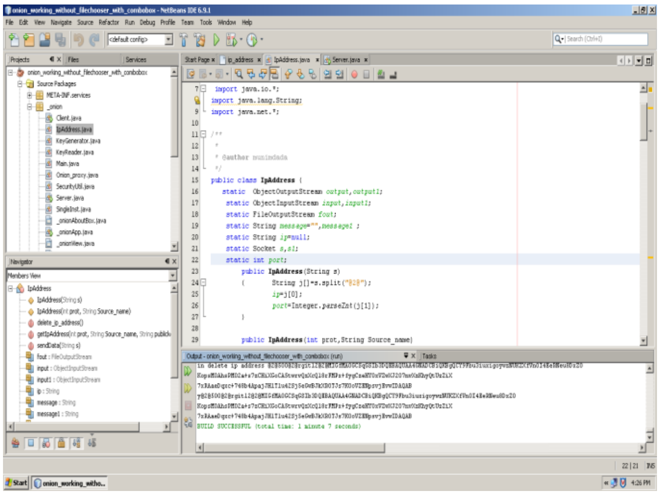
File name: Winter.jpg

Files of type: All Files

```
Encrypted bytes length-------128
Encrypted bytes length-------128
0.0.0.0@2@500
Connection was accepted from 192.168.0.113 3162 500
In not file creation
192.168.0.117@2@500
192.168.0.117@2@500
```

Figure 7: Select files on onion routers

Step 4: The shared data or files are shows on destinations which are decrypted



```
 import java.io.*;
import java.lang.String;
import java.net.*;

/**
 *
 * @author munindada
 */
public class IpAddress {
   static  ObjectOutputStream output,output1;
    static ObjectInputStream input,input1;
    static FileOutputStream fout;
    static String message="",message1 ;
    static String ip=null;
    static Socket s,s1;
    static int port;
        public IpAddress(String s)
        {       String j[]=s.split("@2@");
                ip=j[0];
                port=Integer.parseInt(j[1]);
        }

        public IpAddress(int prot,String Source_name)
```

```
in delete ip address @2@500@2@rgit12@2@MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCT9Fbu3iuzigoywzMUK2XfVn0I4HeRNeu8Dx20
KopsMDAhsPMD2a+s7zCHiXGoCA5twrvQzXcQl8rFMPx+fygCzeHY0xV2wXJ207mvXnKhyQtUzZiX
7xRAaeDqrc+748b4ApajJH1T1u42SjSeGwBJkXR0TJs7KOoVZENpsvjBvwIDAQAB
y@2@500@2@rgit12@2@MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCT9Fbu3iuzigoywzMUK2XfVn0I4HeRNeu8Dx20
KopsMDAhsPMD2a+s7zCHiXGoCA5twrvQzXcQl8rFMPx+fygCzeHY0xV2wXJ207mvXnKhyQtUzZiX
7xRAaeDqrc+748b4ApajJH1T1u42SjSeGwBJkXR0TJs7KOoVZENpsvjBvwIDAQAB
BUILD SUCCESSFUL (total time: 1 minute 7 seconds)
```

Figure 8: Results after file sharing on OR.

Step 5: The results of Encryption and Decryption of data shared between different end systems are shows on DNS Server Because DNS Server act as an Onion Router between the different terminals and route the encrypted and decrypted data among the terminals.

These are the results shown the actual routing of data. Where the data is encrypted and decrypted with onion layers when data will pass through the different terminals which are in the networks. So here the results getting with the Onion Routing features

6. CONCLUSION

Here we presented a protocol called Onion Routing. The purpose of Onion Routing is to protect the anonymity of a user who wants to communicate over a network. To judge the level of anonymity it is important to take the intention of the attacker into account because it has to be evaluated if the effort is worth the benefit for an attacker. Onion routing provides protection against adversaries who what to gather a large amount of information by doing traffic analysis for example for commercial reasons. For such adversaries there is no benefit of getting only communication data of a small number of persons and attacking a large number of parties costs too much effort. But if the goal is to really hide with whom one is communicating it is always possible to reveal hat. Anonymous connections may be used as a new primitive that enables novel applications in addition to facilitating secure versions of existing services. Anonymous socket connections provide protection against both eavesdropping and traffic analysis. Although our focus is on anonymous connections, and not anonymous communication, anonymous communication is also possible by removing identifying information from the data stream. Onion routing's anonymous connections are application independent and can interface with unmodified Internet applications by means of proxies. Our implementation of onion routing includes
proxies for Web browsing, e-mail, and remote login.

REFERENCES

[1] K. Kaviya, "Network Security Implementation by Onion Routing" Sri Ramakrishna Engineering College, 2009

International Conference on Information and Multimedia Technology.

[2] Aaron Johnson,Paul Syverson "More Anonymous Onion Routing Through Trust" International Conference,2008,Center for High Assurance Computer Systems, U.S. Naval Research Laboratory,

Washington, DC 20375 USA

[3] Roger Dingledine,Nick Mathewson,Paul Syverson "Tor: The Second-Generation Onion Router" Naval Research Lab, LNCS 2009

[4] Aniket Kate, Greg Zaverucha, and Ian Goldberg "Pairing-Based Onion Routing" David R. Cheriton School of Computer Science University of Waterloo, Research project 2008, Waterloo, ON, Canada N2L 3G1

[5] Paul Syverson , Gene Tsudik y,Michael Reed ,Carl Landwehr z "Towards an Analysis of Onion Routing Security" Proceedings of the DARPA Information Survivability Conference and Exposition, Hilton Head, SC, IEEE CS Press, January 2000.

[6] Paul F. Syverson, David M. Goldschlag, and Michael G. Reed "Anonymous Connections and Onion Routing", Naval Research Laboratory , 12th Annual Computer Security Applications Conference,San Diego, CA, IEEE CS Press, December, 1998

[7] Can Tang,Ian Goldberg "An Improved Algorithm for Tor Circuit Scheduling", CCS'10, October 4–8, 2010, Chicago, Illinois, USA.

Copyright 2010 ACM 978-1-4503-0244-9/10/10 ...$10.00.

[8] Stefen Michels "Analysing Onion Routing Bachelor-Thesis" ,June 22, 2009

[9] Torreleasenotes https://git.torproject.org/checkout/tor/master/ReleaseNotes.

[10] D. Eastlake 3rd and P. Jones. US Secure HashAlgorithm 1 (SHA1). RFC 3174 (Informa- tional), September 2001. Updated by RFC4634

[11] Michael G. Reed, Paul F. Syverson, David M. Goldschlag. "Proxies for anonymous Routing". Naval Research Laboratory, Washington DC.

[12] RSA Algorithm.on internet.

[13] The Anonymizer. http://www.anonymizer.com