RESEARCH ARTICLE OPEN ACCESS

Secure Routing and Identifying Hacking In P2p System

Gowsiga Subramaniam*[1], Senthilkumar Ponnusamy [2]
[*1]Assistant Professor, Saveetha Engineering College, Chennai.
[2]Professor&Head, Information Technology, SKR Engineering College, Chennai.

ABSTRACT
Packet forwarding is the procedure to route the packet from source to destination via routers. Getting the systems ip address in a LAN and decides the source, destination and the intermediate systems for our routing process. Using any database form the routing table with columns like source address, destination address and router addresses. After creating the routing table enters the ip address in the respective columns. As per this procedure by comparing ip addresses in the routing table the packet will be forward to next ip address. The ip addresses in the routing table, are the path selected by the user according to the respective path the packet will be forward and finally reach destination. Source generate a packet and forward to ip address which is in router ip as per in routing table. Source need to know whether the packet received by authenticated router or not thus each router send an acknowledgement packet to the source. Thus each and every router follows the same procedure to avoid unauthenticated router access in a routing process. As per acknowledgement received from each router source finds out the unauthenticated router access. As per security concern added the signature scheme in a routing process. Before send the packet, source generates the signature with peer ID of destination then forwards the packet to the ip address as per routing table. The packet will be decrypted only the system matched with the signature otherwise just forwards the packet to the next system. Tracer routing is to find out the unauthorized router access in a routing process. This is verifying by getting acknowledgment from each and every router in the routing process by a source. Then source compare the acknowledgement with the predefined routing table if any ip mismatch then it will be consider as unauthorized ip.

Keywords: P2P, Route Selection, Packet Forwarding, Hacking

I. INTRODUCTION

Client-server computing or networking is a distributed application architecture that partitions tasks or workloads between service providers (servers) and service requesters, called clients. Often clients and servers operate over a computer network on separate hardware. A server machine is a high-performance host that is running one or more server programs which share its resources with clients. A client does not share any of its resources, but requests a server's content or service function. Clients therefore initiate communication sessions with servers which await (listen to) incoming requests.

The client-server model has become one of the central ideas of network computing. Many business applications being written today use the client-server model. The most basic type of client-server architecture employs only two types of hosts: clients and servers. This type of architecture is sometimes referred to as two-tier. It allows devices to share files and resources. The two tier architecture means that the client acts as one tier and application in combination with server acts as another tier.

Database debugging is a server-based report generation software system. It can be used to prepare and deliver a variety of interactive reports. It is administered via a LAN interface. Reporting services features a LAN services interface to support the development of custom reporting applications.

The reports tool is a way to provide commonly used database queries in a regularly updated, easy-to-use form. It is a companion to the SQL Query service. Computer program that is used to create statistical reports based upon the information in a computer file. Most report generator programs are written for use by people other than programmers. The simplest report generators create reports that illustrate the relationship between only two variables in the database.

II. SYSTEM ANALYSIS

Peer-to-Peer (P2P) overlay systems offer a substrate for the construction of large scale and distributed applications. However, they pose new challenge for security. Secure assignment node identifiers and secure routing maintenance can be achieved by minimizing the probability that nodes are controlled by attackers. However, an adversary can prevent correct message delivery throughout the overlay. When one or more peers between initiator and target are malicious, a message might be dropped, polluted or forwarded to a wrong place. Previous works on secure message routing concentrated mainly on employing redundancy to increase the probability of a message being delivered successfully the target. The goal of message routing is to maximize the probability of at least one copy of the correct message reach the correct target. Presents a multiple redundant router algorithm for Pastry. Initiator sends a query to all of its neighbors in Pastry overlay. Then each neighbor forwards the query toward the target. If at least one copy of query arrive target, the query is considered successfully delivered. However, it will inevitably cause congestion and burden system, especially in a bandwidth intensive system. Disrupt the message routing, or take advantage of locality to control some routes. Pretend to be the target

Disadvantage:

- Destination once receives the packet makes key match if accepted then conclude their no more hacking in the routing.
- No approaches followed to detect the route hacking.

III. METHODOLOGY

To make the routing strategy perform best, we present an efficient routing strategy, called tracer routing. Tracer routing enables the initiator to trace the whole routing process. In each step, the intermediate peer *x* not only forwards the query to the next hop, but also returns the IP address of the next hop to initiator. With the additional information, the initiator has the knowledge about the whole routing process. Each intermediate peer directly forwards the query to the next hop, thus the query can be routed quickly. Combined with the peer-ID based signature scheme, tracer routing offers a good tradeoff between routing efficiency and security. We propose to address routing message attack by combined tracer routing with Peer-ID based signature scheme. Note that Peer-ID based signature scheme is not necessary. Any techniques of verifying the Peer-ID of remote peer can work with tracer routing.

Advantage:

- Source side detection and identification of hacking gives more security of data transfer.
- With the additional information, the initiator has the knowledge about the whole routing process.
- Each intermediate peer directly forwards the data to the next peer, thus the data can be routed quickly.

IV. STRUCTURED P2P NETWORK SETUP

A peer-to-peer (abbreviated to P2P) computer network is one in which each computer in the network can act as a client or server for the other computers in the network, allowing shared access to various resources such as files, peripherals, and sensors without the need for a central server. P2P is a distributed application architecture that partitions tasks or workloads among peers. Structured P2P network contains self organizing peer's network. Each peer should be initialized with respective IP address and port number for communication like files sharing or messaging.

V. ROUTING SELECTION AND PACKET FORWARDING

Routing tables are generally not used directly for packet forwarding in modern router architectures; instead, they are used to generate the information for a smaller forwarding table which contains only the routes which are chosen by the routing method as preferred routes for packet forwarding. Whenever a node or system needs to send data to another node on a network, it must know where to send it, first. If the node cannot directly connect to the destination node, it has to send it via other nodes along a proper route to the destination node. Path selection between source and destination is the procedure to follow on this module. Getting the systems ip address in a structured network and decides the source, destination and the trusted intermediate systems for our routing process.

Packet forwarding is the procedure to route the packet from source to destination via routers. As per this procedure by comparing ip addresses in the routing table the packet will be forward to next ip address. The ip addresses in the routing table, are the path selected by the user according to the respective path the packet will be forward and finally reach destination.

VI. ACKNOWLEDGEMENT PROCESS

Source generate a packet and forward to ip address which is in router ip as per in routing table. Source need to know whether the packet received by authenticated router or not, thus each data received router's peer ID must be automatically send as acknowledgement to the source. Thus each and every router follows the same procedure to avoid unauthenticated router access in a routing process. As per acknowledgement received from each router source finds out the unauthenticated router access.

VII. PEER ID BASED SIGNATURE SCHEME

As per security concern added the signature scheme in a routing process. Before send the packet, source generates the signature with peer ID of destination then forwards the packet to the ip address as per routing table. The packet will be decrypted only the system matched with the signature otherwise just forwards the packet to the next system. We are using DES data encryption standard cryptography algorithm for secure data transfer.

VIII. TRACER ROUTING APPROACH

To make the routing strategy perform best, we present an efficient routing strategy, called tracer routing (below diagram). Tracer routing enables the initiator to trace the whole routing process. In each step, the intermediate peer *x* not only forwards the query to the next hop, but also returns the IP address of the next hop to initiator. With the additional information, the initiator has the knowledge about the whole routing process. Each intermediator peer directly forwards the query to the next hop, thus the query can be routed quickly. Combined with the peer-ID based signature scheme, tracer routing offers a good tradeoff between routing efficiency and security. Tracer routing is to find out the unauthorized router access in a routing process. This is verifying by getting acknowledgment from each and every router in the routing process by a source. Then source compare the acknowledgement with the predefined routing table if any ip mismatch then it will be consider as unauthorized ip.

IX. IMPLEMENTATION

When the initial design was done for the system, the client was consulted for the acceptance of the design so that further proceedings of the system development can be carried on. After the development of the system a demonstration was given to them about the working of the system. The aim of the system illustration was to identify any malfunction of the system.

After the management of the system was approved the system implemented in the concern, initially the system was run parallel with existing manual system. The system has been tested with live data and has proved to be error free and user friendly. Implementation is the process of converting a new or revised system design into an operational one when the initial design was done by the system; a demonstration was given to the end user about the working system. This process is uses to verify and identify any logical mess working of the system by feeding various combinations of test data.

After the approval of the system by both end user and management the system was implemented. A product software implementation method is a blueprint to get users and/or organizations running with a specific software product. The method is a set of rules and views to cope with the most common issues that occur when implementing a software product: business alignment from the organizational view and acceptance from the human view.

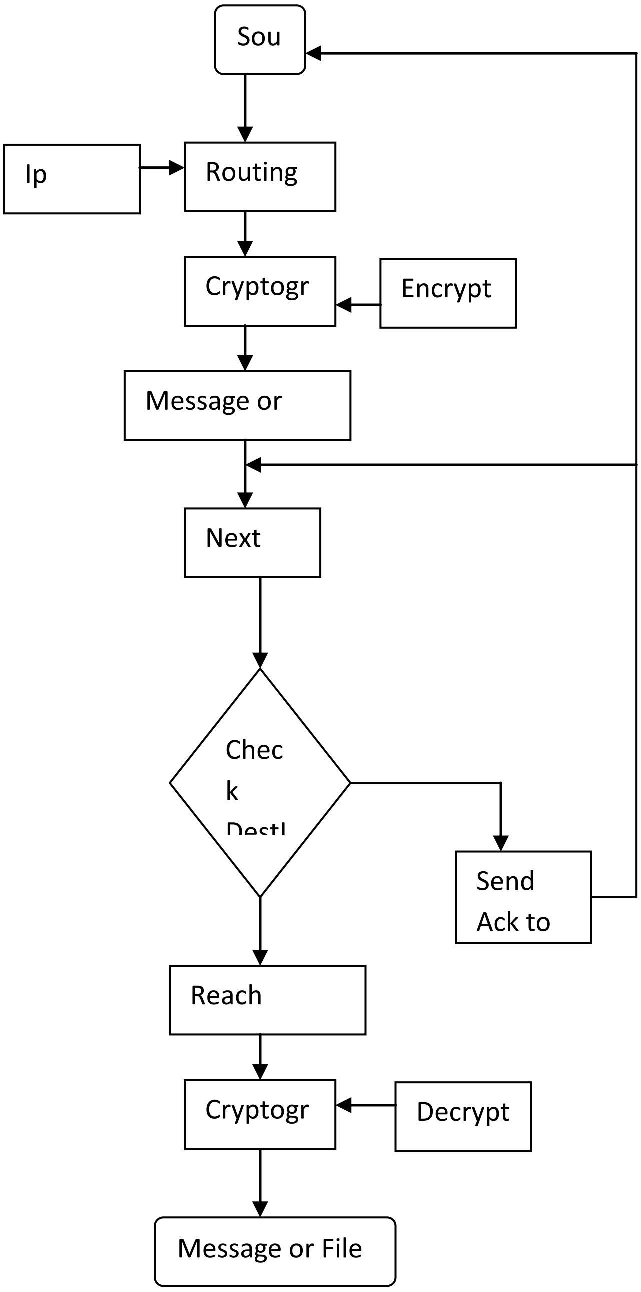
The implementation of product software, as the final link in the deployment chain of software production, is in a financial perspective of a major issue. The Implementation methodology includes four phases - Discovery, System Development, User Acceptance Testing and Production Rollout. It's easy to be overwhelmed by slick marketing presentations, particularly when the sales force is talking about things that most people don't completely understand. Showmanship gets in the way of real capabilities. Unless the review team is judging each vendor against the same list of needs, with the same understanding of the significance of each rating, "likeability" can win over capability.

These implementation phases are designed to provide clients with a seamless transition from an existing electronic or paper-based system to Sigmund while ensuring all aspects of a client's operations are accounted for by the software. The Sigmund project team, comprised of individuals with clinical, billing and operations experience, is equipped with the skills and tools to manage the entire process from system requirements gathering to deployment. Sigmund provides various levels of support, depending on client needs, including client-side Project Management.

The Discovery Phase is preceded by a Project Kick-off Work-session that includes application demonstrations, completion and review of requirements and configuration questionnaires, identification of key client documentation, as well as consultation on possible process re-engineering needs. The Project Kick-off provides your organization with the opportunity to not only introduce the Sigmund Project Team to your organization, but also to define and structure your organization's Project Team.

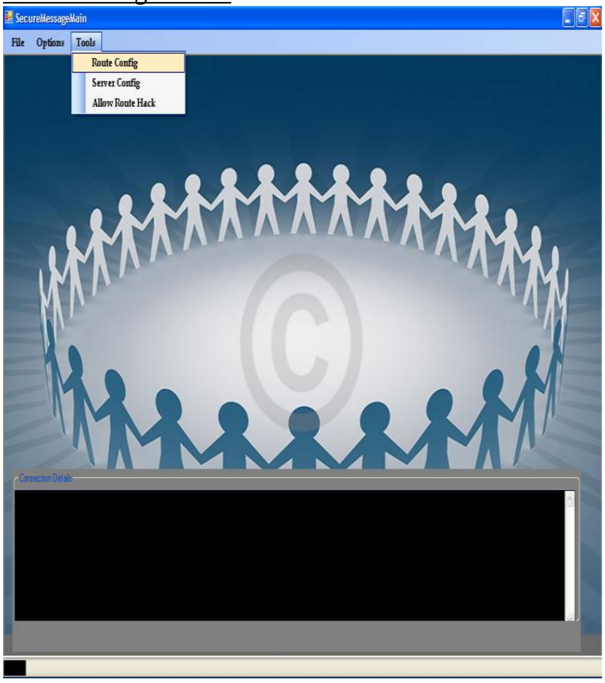
System implementation is made up of many activities. The six major activities used in project development are as follows.

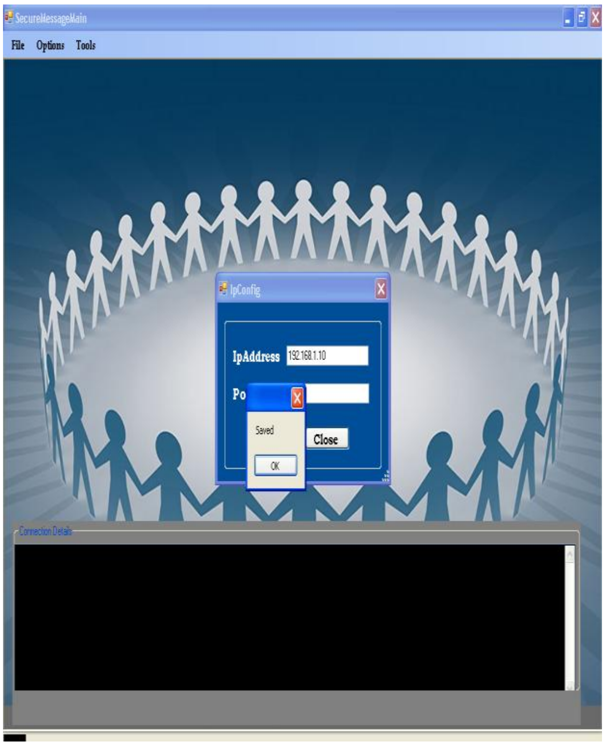
X. FLOW DIAGRAM



XI. SAMPLE SNAPSHOTS

Route Configuration:

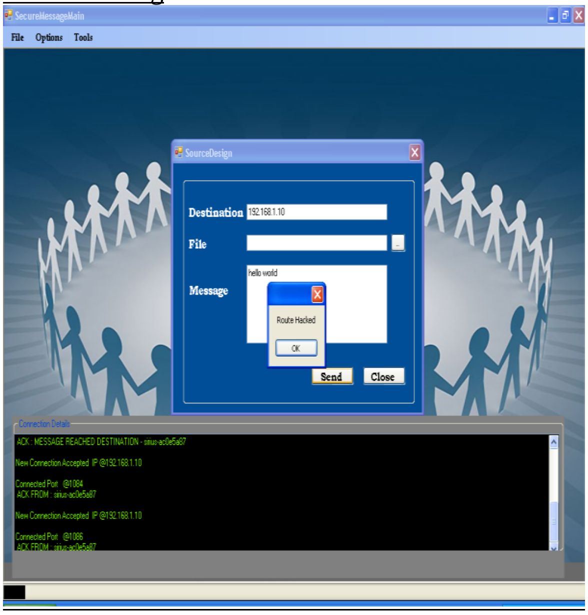




IpAddress 192.168.1.10

Saved

Route Hacking



Destination 192.168.1.10

File

Message hello world

Route Hacked

```
ACK : MESSAGE REACHED DESTINATION - siius-ac0e5a87

New Connection Accepted  IP @192.168.1.10

Connected Port  @1084
 ACK FROM : sirius-ac0e5a87

New Connection Accepted  IP @192.168.1.10

Connected Port  @1086
 ACK FROM : sirius-ac0e5a87
```

SAMPLE CODING

Source Form:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.Net.Sockets;
using System.IO;
using System.Threading;
using System.Net;
using System.Xml;
using System.Security.Cryptography;
using System.Data.OleDb;
using MySql.Data.MySqlClient;
using MySql.Data.Types;
namespace MessageRouting
{
    public partial class SourceDesign : Form
    {
System.Net.Sockets.TcpClient clientSocket = new 
System.Net.Sockets.TcpClient();
        NetworkStream serverStream;
        private StreamWriter clientStreamWriter;
        private static byte[] key = { };
        private static byte[] IV = { 18, 52, 86, 120, 144, 
171, 205, 239 };
        private const string EncryptionKey = 
"abcdefgh";
        private MySqlConnection dataConnection;
        public SourceDesign()
        {
            InitializeComponent();
        }
        private void SourceDesign_Load(object sender, 
EventArgs e)
        {
        }
        private void setDBConnection()
        {
            try
            {
                string connectionStr = 
"server=localhost;database=secure;uid=root;passwor
d=root";
                dataConnection = new 
MySqlConnection(connectionStr);
                dataConnection.Open();
            }
            catch (Exception e)
            {
                
MessageBox.Show(e.StackTrace.ToString());
            }
        }
      public static string toEncrypt(string 
stringToEncrypt)
        {
            try
            {
                key = 
System.Text.Encoding.UTF8.GetBytes(EncryptionK
ey);
                DESCryptoServiceProvider des = new 
DESCryptoServiceProvider();
                byte[] inputByteArray = 
Encoding.UTF8.GetBytes(stringToEncrypt);
                MemoryStream ms = new 
MemoryStream();
                CryptoStream cs = new CryptoStream(ms, 
des.CreateEncryptor(key, IV), 
CryptoStreamMode.Write);
                cs.Write(inputByteArray, 0, 
inputByteArray.Length);
                cs.FlushFinalBlock();
                return 
Convert.ToBase64String(ms.ToArray());
            }
            catch (Exception ex)
            {

            }
            return "";
        }
        private void btnSend_Click(object sender, 
EventArgs e)
        {
            try
            {
                MySqlDataReader dataReader;
                string routeString = "";
                string cmdStr = "Select * FROM routetable
where destination= '" + txtIpaddress.Text + "'";
                setDBConnection();
                MySqlCommand oleCmd;
                oleCmd = new MySqlCommand(cmdStr,
dataConnection);
                dataReader = oleCmd.ExecuteReader();
                while ((dataReader.Read()))
                {
                    routeString =
dataReader.GetValue(2).ToString();
                }
                  string[] nextRoute = routeString.Split(new
char[] { ';' });
                string[] nextIP = nextRoute[0].Split(new
char[] { ',' });
               string hostName =
System.Net.Dns.GetHostName();
                string myIP =
System.Net.Dns.GetHostByName(hostName).Addres
sList.ToString();
                clientSocket = new
System.Net.Sockets.TcpClient();
                clientSocket.Connect(nextIP[0],
Int32.Parse(nextIP[1]));
                serverStream = clientSocket.GetStream();
                clientStreamWriter = new
StreamWriter(serverStream);
                clientStreamWriter.AutoFlush = true;
                 if (txtPath.Text.Length > 0)
                {
                    FileStream stream = new
FileStream(txtPath.Text, FileMode.Open,
FileAccess.Read);
                    StreamReader reader = new
StreamReader(stream);
                    String ipAddress = reader.ReadLine();
                    reader.Close();
                    stream.Close();
                    txtMessage.Text = ipAddress;
                }
                string encStr =
toEncrypt(txtMessage.Text);
                System.Net.IPAddress oAddr;
                string sAddr;
                string dnshostName = Dns.GetHostName();
                oAddr = new
System.Net.IPAddress(System.Net.Dns.GetHostByN
ame(dnshostName).AddressList[0].Address);
                sAddr = oAddr.ToString();
                String ipValue = sAddr + "," +
PeerProcess.localPort;
                clientStreamWriter.WriteLine("" + ipValue
+ "?" + routeString +"#"+ encStr.Length + "&" +
encStr + ":" + routeString);
                clientSocket.Close();
            }
            catch (Exception ex)
            {
                Console.WriteLine(ex.StackTrace);

MessageBox.Show(ex.StackTrace.ToString());
            }
             }
        private void btnClose_Click(object sender,
EventArgs e)
        {
            Visible = false;
        }
        private void btnPath_Click(object sender,
EventArgs e)
        {
            DialogResult digResult =
openFile.ShowDialog();
            txtPath.Text = openFile.FileName.ToString();
        }
    }
}
```

XII. CONCLUSION AND SCOPE FOR FURTHER WORK

Peer-to-peer overlay networks, both at the network layer and at the application layer. We have shown how techniques ranging from cryptography through redundant routing to economic methods can be applied to increase the security, fairness, and trust for applications on the p2p network. Because of the diversity of how p2p systems are used, there will be a corresponding diversity of security solutions applied to the problems. has presented the design and analysis of techniques for secure node joining, routing table maintenance, and message forwarding in structured p2p overlays. These techniques provide secure routing, which can be combined with existing techniques to construct applications that are robust in the presence of malicious participants.

In peer to peer system, **load balancing** is a technique to distribute workload evenly across two or more computers, network links, CPUs, hard drives, or other resources, in order to get optimal resource utilization, maximize throughput, minimize response time, and avoid overload. In our system we implement the system to check the overload and retransmit the files to sub nodes in the network. As future enhancement for client server communication we will add symmetric key sharing as introduce security. We don’t know the client is trusted one so request and response between client and server must flow under key sharing.

REFERENCES

[1] Elias M awas, “System Analysis and Design ” Galgotia publication, Second Edition 1996

[2] Harvey M. Deitel , Paul J. Deitel, Tem R. Nieto “C# .NET How to Program” 2nd Edition

[3] Richard Fairley, ”SOFTWARE ENGINEERING CONCEPTS”, Tata Mc Graw Hill Publication, Second Edition,1997

[4] "SQL Server "Katmai" to Deliver Entity Data Platform and Support LINQ". http://oakleafblog.blogspot.com/2007/05/sql-server-to-deliver-entity-data.html. Retrieved 2007-05-12.

[5] MSDN Library: Reporting Services Render Method - See Device Information Settings

[6] Microsoft. "Microsoft .NET Framework 3.5 Administrator Deployment Guide". http://msdn.microsoft.com/library/cc160717.aspx. Retrieved 26 June 2008. S. Somasegar. "Visual Studio 2010 and .NET FX 4 Beta 1 ships!". Archived from the original on 27 May 2009. http://www.webcitation.org/5h5lV7362. Retrieved 25 May 2009.

[7] Scott Guthrie: Silverlight and the Cross-Platform CLR". Channel 9. 30 April 2007. http://channel9.msdn.com/shows/Going+Deep/Scott-Guthrie-Silverlight-and-the-Cross-Platform-CLR.

[8] ECMA 335 - Standard ECMA-335 Common Language Infrastructure (CLI)". ECMA. 1 June 2006. http://www.ecma international.org/publications/standards/Ecma-335.htm. Retrieved 1 June 2008.

[9] "Technical Report TR/84 Common Language Infrastructure (CLI) - Information Derived from Partition IV XML File". ECMA. 1 June 2006. http://www.ecma-international.org/publications/techreports/E-TR-084.htm.

[10] "ECMA-334 C# Language Specification". ECMA. 1 June 2006. http://www.ecma-international.org/publications/standards/Ecma-334.htm.

[11] M. Castro, P. Duschel, A. Rowstron, and D. S. Wallach, Secure routing for structured peer-to-peer overlay network. In Proceedings of the fifth Symposium on Operatating System Design and Implementation, Dec 2002.

[12] D. Wallach, A survey of peer-to-peer security isssues. In Proceedings of International Symposium of Software Security - Theories and Systems, Nov 2003.

[13] M. S. Artigas, P. G. Lopez, A. F. G. Skarmeta, A Novel methodology for constructing secure multipath overlays. IEEE Internet Computing, 9(6):50-57,2005.

[14] E. K. Lua, T. Griffn, M. Pias, H. Zheng, and J. Crowcroft, On-line secret sharing. In ACM SIGCOMM-Usenix Internet Measurement Conference 2005 (IMC2005), Oct 2005.

[15] E. K. Lua and T. G. Griffn, Embeddable overlay networks. In Proceedings of IEEE Symposium on Computers and Communications 2007, July 2007