ULTIMATE MESSAGE SECRECY WITH PROPOSED NETWORK ARCHITECTURE
We hereby certify that this dissertation, submitted by Amogh Vijay Pandit, satisfies the dissertation requirement of the Masters degree of Computer Science has been approved.

Dr. Narayan Murthy, Chair, CS Dept
Supervisor and Dissertation Committee Member

Member of Dissertation Committee

Dissertation Committee Member

Dr. Susan M. Merritt
Dean School of Computer Science and Information Systems
Revision History

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Description</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>01/18/2005</td>
<td>1.0</td>
<td>Stegnographic Solution</td>
<td>Amogh</td>
</tr>
<tr>
<td>03/01/2005</td>
<td>1.1</td>
<td>CryptoGraphic Solution and Combination</td>
<td>Amogh</td>
</tr>
<tr>
<td>04/25/2005</td>
<td>1.2</td>
<td>SMTP Proxy Engine and Email Arch</td>
<td>Amogh,</td>
</tr>
</tbody>
</table>
TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>4</td>
</tr>
<tr>
<td>1.0 INTRODUCTION</td>
<td>5</td>
</tr>
<tr>
<td>2.0 PROPOSED STEGNOCRYPTOGRAPHIC PICTURE MESSENGER</td>
<td>6</td>
</tr>
<tr>
<td>2.0.1 PROPOSED STEGNOGRAPHIC SOLUTION</td>
<td>8</td>
</tr>
<tr>
<td>2.0.2 PROPOSED CRYPTOGRAPHIC SOLUTION</td>
<td>10</td>
</tr>
<tr>
<td>2.0.3 Final Picture Messenger Proposal</td>
<td>11</td>
</tr>
<tr>
<td>3.0 RESULTS</td>
<td>13</td>
</tr>
<tr>
<td>4.0 PROPOSED SMTP ENGINE-NG MAIL ARCHITECTURE</td>
<td>15</td>
</tr>
<tr>
<td>4.1 DEMILITARIZED ZONE</td>
<td>16</td>
</tr>
<tr>
<td>4.1.1 FILE ARCHITECT</td>
<td>16</td>
</tr>
<tr>
<td>4.2 SWITCHOVER PROCESS METHODS</td>
<td>18</td>
</tr>
<tr>
<td>5.0 PROPOSED PROXY ENGINE</td>
<td>20</td>
</tr>
<tr>
<td>5.1 FLOWCHART</td>
<td>21</td>
</tr>
<tr>
<td>6.0 RESULTS</td>
<td>22</td>
</tr>
<tr>
<td>7.0 FUTURE WORK</td>
<td>24</td>
</tr>
<tr>
<td>8.0 CONCLUSION</td>
<td>24</td>
</tr>
<tr>
<td>9.0 REFERENCE</td>
<td>25</td>
</tr>
</tbody>
</table>

CS694, Spring 2005                  Pace University                  Page- 3
ABSTRACT:

E-Messaging is an essential application of the Internet. Effective October 2000 email signatures have same weight age as pen-paper to close multibillion business deals. Also, there is urgent need to hide the data while transferring over the digital media. Internet messaging is the backbone of information flow in today’s world. Computer viruses, worms, sniffing are major threats to the society by disrupting the normal information flow and collecting or destroying information without authorization. Increase of these threats has been significant in current period. With the given “Stegnecryptographic Technique” user will also be able to send mission-critical information secretly using digitally sealed documents. The email-client architecture proposed would be self sufficient to detect & protect itself from unsolicited d-messages & concealed messages which may be a threat to national security. We propose a new impediment of deterrence & control to the administrators for protecting their network assets. Thus we want to present a new way to flow these messages over the media which will be a new frontier in digital era and parallely equipped with necessary engine for secure and smooth transmission and reception.
1.0 INTRODUCTION:

Message transmission over the internet still has all kinds of security problems. Current approaches in encryption, including cryptography, can leave obviously noticeable marks to the message, suspicious enough for eavesdroppers to get attention to it.

Steganography is the art and science of hiding communication which is Greek for ‘covered writing’. Thus it embeds hidden content in unremarkable cover media without any suspicion. Classical steganography system security relies on the encoding system secrecy. It attempts to be detectable only if secret information is known. Essentially, steganographic communication senders and receivers agree on a steganographic system and a shared secret key that determines how a message is encoded in the cover medium. Steganography and Cryptography are considered to be cousins in cyber craft family. Crypto scrambles the message, making it difficult to understand, while Steganography hides the message so that it is not seen.

There are several techniques proposed by researchers and they all depend on properties of images. An image is an array of numbers that represent light intensities at various points (pixel). Digital images are normally stored in 24-bit or 8 bit files. A 24-bit image provides most space for hiding information. However it can be quite large(exception – JPEG file). Each primary color is represented by 1-byte. 24 bit images use 3 byte per pixel to represent the color value. For example, White is combination of 100% red, blue and green and its decimal value (255,255,255) and its binary value will be (11111111,11111111,11111111). There are compression methods termed “Lossless Compression” typically saved as 8-bit BMP or GIF and lossy for JPEG but doesn’t keep
the message integrity. In 8-bit image each pixel is represented by single byte which points to color index table.

Some of the insertion techniques proposed were **Quantization modulation technique**: use different Quantization index, **LSB Modification**: this technique based on modifying LSBs. Masking approach where pixel value in masked area raised or lowered by some percentage, **Transform domain technique** depend on Fourier Transforms and **Space Spectrum Technique** where hidden data is spread throughout the cover image. But the purpose of Stegnographic methods can be defeated by using statistical methods to reveal the message. That’s a reason we also proposed public and private cryptographic codes for each bunch of users so that picture contents are also password protected. If also hidden message received by stegnanalyst which will be useless and ultimate purpose of message secrecy can be achieved.

In near future looking at increasing needs of sending mission critical messages over digital media we have to shield with this technology. Both parties need to take care as image appearance can deceive the users and that emerge the need for some engine which will run on client side and make sure that transmitted and received stegnography messages are not harmful which we tried to achieve in our messenger by using Private and Public Crypto code proposed in Ultra Cryptographic method. Also we developed file code to register and rename the file extensions. As we increase the encryption size of image file get increase. So proposed engine should also make sure that client email box should not get piled up with messages because of slow response from Email SMTP server. Also email architecture is constantly changing looking at the threats on email.
communication. We visualize the possible email architecture will look like in near future due to these threats.

In this paper we present a sample VB application -**Picture Messenger** to hide information in digital image and design of **Multi Modular SMTP- engine** which will front the media for the client. We also tried to propose how the email architecture should look like considering all threats and required uptimes for message flows.

We demonstrate new applications to understand the new era of digital transmission traffic and its analysis.
2.0 PROPOSED STEGNOCRYPTOGRAPHIC PICTURE MESSENGER:

As shown in the following flow chart below -

The message will get eloped within a normal image by our proposed Stenographic Algorithm Additionally, before hiding the same within an image file the message can be encrypted using available Ultra Cryptographic algorithm.
2.0.1 Proposed Stenographic Solution:

We tried to propose an available masking approach in steganography. This program uses three pixels for a character. It uses eight components of the three pixels (last remains unused) to form a binary representation of the ASCII code of the character, but instead of using zeroes and ones, it uses odd and even numbers. To be clearer let me give you a practical example. Let's say that we need to store the character "a" in the following three pixels: (154, 73, 211),(98, 110,39) and (16,255,85). The code for "a" is 97(in decimal) and in binary is 11000001.

This program transforms the first eight components in even number so the three pixels became: (154,72,210),(98,110,38),(16,254,85). Then, it makes a correspondence between the components and the binary representation of the character. If a digit in the binary number is 1, program adds 1 to the correspondent component. The final values for the components are: (155,73,210),(98,110,38),(16,255,85).You can notice that there is not much difference between the initial and the final values. This is done to distinguish between a Numeric & Character.

Thus our approach represent each character by its ascii value and use some kind of masking approach but making sure that there is very less difference between final values which makes Stegnoanalysis very difficult and can be recovered at the other end with same type of Algorithm. As follows when you will ask for the following window will pop up and then you can load the .bmp image and put or get message from other user.
And we get following output after putting correct secret code:
2.0.2: CRYPTOGRAPHIC SOLUTION:

In our next phase we decided to use strong code which was proposed by D. Rijmenants also called (ULTRA v1.0.3 Cryptographic Algorithm (c) 2004). Features proposed for this Algorithm was:

1) Variable-length triple transposition key

2) Fast symmetric stream cipher in output feedback mode

3) Optional exclusive users groups by PCC (Private Crypto Code)

4) Each encryption with the same key and data is different due random header

5) Full file handling support, including extension encryption

6) Text encryption to text crypto format

7) Data compression before ciphering to strengthen encryption

8) Public functions return error codes and descriptions

This code will give your project a strong encryption for text and files. The data is encrypted and decrypted with a variable length key and, if desired, a second PCC-key (Private Crypto Code) is used to create an exclusive user group. To encrypt data the program uses a random dummy header that is encrypted to ensure that the key settings in the beginning of the actual data is impossible to retrieve, unless the random byte sequence is
properly decrypted during decoding. The program will use the **Randomize** and **RND** function to generate a random sequence. We tried to represent the same with following flowchart:

![Flowchart](image)

In this output of encryption function that is feedback to the shift register. There is also additional security for user group by using PCC and which can be integrate with Stegnographic solution proposed.
2.0.3. FINAL PICTURE MESSENGER PROPOSAL:

Thus we integrated both modules as follows. Each file created by our picture messenger can be registered and represented by .pmn type files. While hiding the data steganographic module interact with the Ultra cryptographic module. Proposed Messenger can interact with user group by using public key and decrypt the message using private key.

As shown Modules interaction happen in our messenger. Also we tried to represent how each form interact with each other in next diagram.
3.0 RESULTS:

Following is the picture messenger functions and representation:

**ENCRYPTION:**

![Encryption Image]

**Decryption:**

![Decryption Image]
Public Crypto Code Settings:

![Image of Public Crypto Code Settings](image-url)
4.0 PROPOSED SMTP ENGINE-NG MAIL ARCHITECTURE

We continued our study to figure out how best email architecture should be for flow of messages. We tried to check all the zones of Email Architecture and ideal network setup should be done as follows:

As shown it forwards incoming SMTP requests to a pre-configured Mail Server IP. Before forwarding it scans the messages based on predefined Anti Virus, Spam filter rules. Application can be optionally configured to check for images modified using Steganography techniques.
This application is Mail Server independent providing a next level of defense. External users/Subscribers are unaware of the actual SMTP server. Other then the SMTP proxy plug-in the rest are optional. System administrators can install/uninstall plug-in without complete application reinstallation. This application can help security agencies to eave drop on mails irrespective of the mail server type thus making it easier for service providers to cooperate.

Following are subsection of Architecture:

**4.1 De-Militarized Zone**

In Above Architecture we have separated SMTP/DNS and POP3 as De-Militarized Zone. In computer networks, a DMZ (demilitarized zone) is a computer host or small network inserted as a "neutral zone" between a company's private network and the outside public network. It prevents outside users from getting direct access to a server that has company data. A DMZ is an optional and more secure approach to a firewall and effectively acts as a proxy server as well. The DMZ host then initiates sessions for these requests on the public network. However, the DMZ host is not able to initiate a session back into the private network. It can only forward packets that have already been requested.

To save on cost & providing the users on flexibility of implementation we will try to use as much as open source product for further development and cost effectiveness.
4.1.1 File Architecture:

We will like to discuss the file architecture by following Diagram:
As shown in the diagram smtpd (can be used from obtuse system) divert the incoming and outgoing traffic into different folder. Afterwards Procmail service will scan these folders for dangerous attachments. We have to build SpamAssasin filters separately and can be used as plug-in for procmail to identify spams and hoax.

All virus logs will be forwarded to the repository server for further analysis. Also the whole setup will get monitor by that server. Thus Spam forwarder forwards the email message through Firewall to the Send mail or other email services. We can use Antivirus to detect any additional viruses get ignored by above setup.

If you follow the architecture we have proposed throttle base Smtp service in case of virus attack. Complete information and implementation is described here [http://www.hpl.hp.com/techreports/2002/HPL-2002-172.pdf](http://www.hpl.hp.com/techreports/2002/HPL-2002-172.pdf). This service will best suite our redundant setup as its mainly build to stop flow of viruses by restricting/delaying the clients to send emails according to there frequency of emails which determines slack (that is number of period over which no email has been sent). It makes complete sense here as Email services are running on standby servers. We have built some sample Expect scripts to continuously scan alarms and NIC card status which is explained in the Switch over Process.
4.2 Switchover Process Methods:

This is very critical part of our architecture. We proposed two possible methods for the same.

1. Script based for which we use EXPECT http://expect.nist.gov/ (planning to switch to Perl)
2. Virtual Linux Networking.

4.2.1 Script Base Switchover:

We implemented three scripts which are perfectly working with two Linux Servers.

Alarm’s script can be built once repository server gets configured.

Checpri.sh : Main script which needs to run all the time in the Backup SMTP

Switch.sh : Switch over script which needs to be run on the pri smtp server

As going forward we will implement this scripts such way that major alarms if exceed over limit backup script will trigger to switch to another server.

4.2.2 Virtual Linux Server Cluster Based Redundancy:

In this method we don’t need script to check NIC card as two parallel server can run on different virtual ip but we still needing Alarm.sh to shutdown the NIC of primary servers if needed.Clusters of servers, interconnected by a fast network, are emerging as a viable architecture for building a high-performance and highly-available service. This type of loosely-coupled architecture is more scalable, more cost-effective, and more reliable than
a single processor system or a tightly-coupled multiprocessor system. However, there are challenges, including transparency and efficiency.

The Linux Virtual Server is one solution that meets the requirements and challenges of providing an always-on service. In LVS, a cluster of Linux servers appear as a single (virtual) server on a single IP address. Client applications interact with the cluster as if it were a single, high-performance, and highly-available server. Inside the virtual server, LVS directs incoming network connections to the different servers according to scheduling algorithms. Scalability is achieved by transparently adding or removing nodes in the cluster. High availability is provided by detecting node or daemon failures and reconfiguring the system accordingly, on-the-fly.

For transparency, scalability, availability and manageability, LVS is designed around three-tier architecture, as illustrated in Fig 4.1. The three-tier architecture consists of:

A load balancer, which serves as the front-end of the whole cluster system. It distributes requests from clients among a set of servers, and monitors the backend servers and the other, backup load balancer. A set of servers, running actual network services, such as Web, email, FTP and DNS. Some of the important configuration steps needed for building virtual IP base virtual service
5.0 PROPOSED SMTPPROXY ENGINE

We proposed Next Generation Digital Mail SMTP Architecture once we introduce stegnography for message flow. There is three particular reasons for the same

1) Intruder can spoof the information flow and add virus code with image.

2) Because of Email increased sizes all messages can piled up at it will be difficult to control that message flow.

3) SMTP Throttling can get implemented at each client side instead of server-side to control the spread of virus in network if any. Complete information and server implementation is described here [http://www.hpl.hp.com/techreports/2002/HPL-2002-172.pdf](http://www.hpl.hp.com/techreports/2002/HPL-2002-172.pdf). Also we have added some references for the same. This service will best suite our redundant setup as it’s mainly build to stop flow of viruses by restricting/delaying the clients to send emails according to there frequency of emails which determines slack (that is number of period over which no email has been sent).

This application serves as SMTP Forwarder. Find below its architecture details:

1. Application Engine platform

2. SMTP Forwarder Module (Default)

3. Anti Virus Module

4. Content Inspection Module
**Application Engine platform**

This is a management console where the administrator can install various available modules & all the modules work in seamless integration. This is a single point of administrator management console which can be used by the System Administrators to control & manage all the modules.

**SMTP Forwarder Module**

Offers SMTP services to the subscribers & forward them to the actual SMTP server.

**Anti Virus Module**

Offers automatic scanning & removal of viruses arriving in the mails, Can also be integrated with third party Anti-Virus solutions.

**Content Inspection Module**

Offers contention inspection of mails & attachments based on a predefine rules.
5.1 FLOWCHART:

Mail Message -> SMTP Proxy

Connection Reset

Log Details -> User is Valid

Virus Check Pass -> Mail Server

SPAM Check Pass

Yes

No

Yes

No
6.0 RESULTS:

1. The Blue circle shows the current address of your laptop learnt by the SMTP Proxy

2. Enter the SMTP address of your actual SMTP Server & Listening Port. Then Press Button “Set Port And Address”
3. Remote Address Field will now contain the IP Address of the SMTP Server

4. Press “New” button in case you want to configure a different listening port for your clients. This is the port address which is used to listen to all the smtp requests.

Testing:

1. Telnet into the SMTP Proxy IP Address on Port 25.

   For example:

   - `telnet 169.254.156.44 25`
   - Give command “HELO” (Without the quotes)
2. Change to “Client History Tab” to view connection history for the session.
7.0 FUTURE WORK:

This is vast topic and we can keep continue research in some of the following area as we go forward. We understand that modules developed are prototypes and there is to be lot work should be done on the same. Following are the area we will look forward to develop:

1) Introduction of Content inspection module in Smtp Engine.
2) Develop the Stegnography code for JPEG images w/o conversions.
3) AES Cryptographic Solution for Picture messenger.
4) Spam Inspection Module for the Smtp Engine.

8.0 CONCLUSION:

We understand that Stegnography will play a major role in near future for shielding the critical data. Our proposed solution can be easily integrated with any existing solution. We will be continuously working on it to sharpen it more with cryptographic techniques so that it will be more difficult to reveal it without applying proper Stegnographic and Cryptographic Algorithm.

We understand the risk in media and attacks from hackers so we also provided a source code for SMTP proxy which will make it easier for user to scan and prevent their system from getting harmed by some additional plug-in. Also users can restrictively pass this information only through their pre approved IP address with this engine.

Thus we will be continuously analyzing these methods and make sure we get perfect digital veil for these messages.
9.0 REFERENCES:


7. N. Provos, “Defending Against Statistical Steganalysis,” *Proc. 10th Usenix Security*


12. Cryptographic Solution by D. Rijmenants