Packet Payload for Network Steganography

R. Bagchi[1]
[1]*Techno India College of Technology, West Bengal*

Abstract - Cryptography, Steganography and Digital Watermarking are the three major techniques to protect sensitive data from unauthorized users. Cryptography and steganography are cousin brother and sister and are data hiding techniques. Digital watermarking is used for unique identification of a particular data. This thesis concentrates on Network Steganography. In this technique the data of network packets are used as a carrier of the steganogram. Tremendous data of various protocols like TCP, IP, UDP, SCTP, etc. could be used as a carrier of the secret message. The capacity of different protocols to hide secret message depends on the packet data size. Steganalysis should be performed to avoid this covert channel.

Index Terms – Covert, Packet Payload, Network, Protocols, Steganography.

I. INTRODUCTION

Steganography is a Greek word in which "Steganos" means "Covered" and "Graphos" means "writing" [1]. Steganography simply means "covered writing". It is a very old technique which is used to send secret messages or to carry out secret communication between two parties. The third person is not able to know that an exchange of messages is taking place between the sender and the receiver. If someone other than the sender and receiver comes to know about the secret communication than the motive of steganography is not completed and hence it could no longer be called steganography. A proper and innocent carrier is required to accomplish it.

II. BASIC MODEL

As mentioned earlier, Steganography requires a carrier to carry the secret message. Figure 2.2 gives the idea behind hiding secret messages. At the sender side, the secret message is hidden in the carrier using embedding mechanism. Then the secret message sent. The receiver gets the message and decodes it to get the secret information. Here the two operations viz embedding and extraction are important.

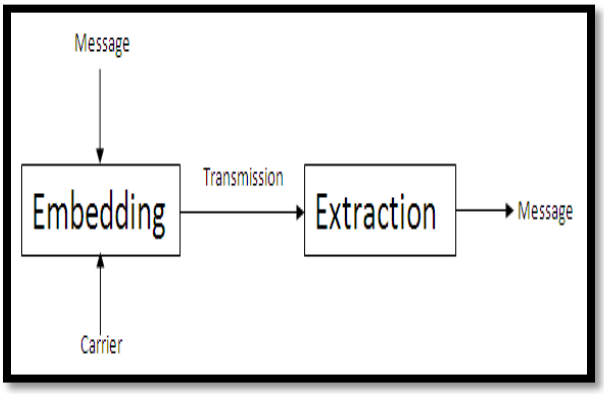

Figure 1 Basic Model

III. PROPOSED TECHNIQUE

IV.

The workflow of the proposed solution is divided into sender side and receiver side. The necessary workflows are represented below.

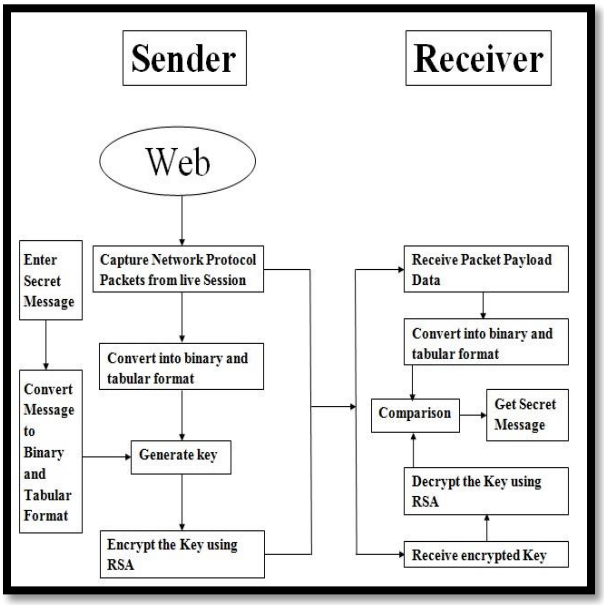

Figure 2 Flow of Proposed Technique

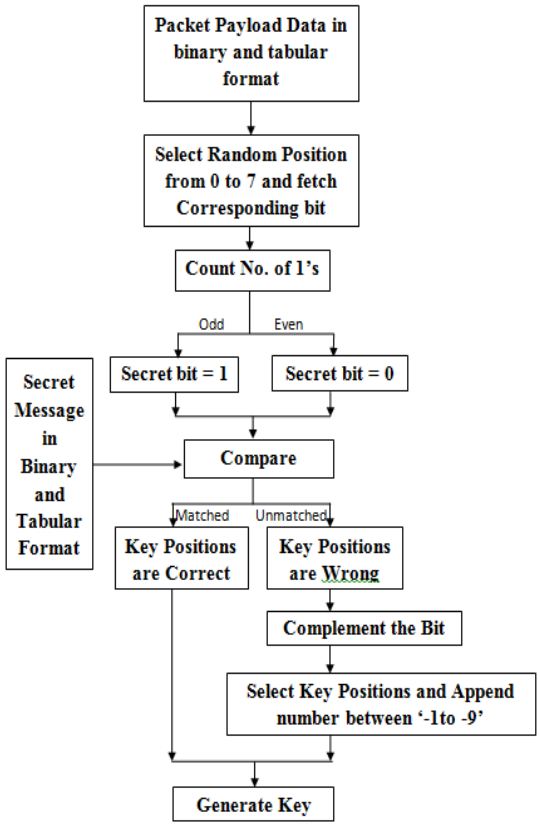

Figure 3 Flow of Key Generation

Sender Side Workflow

To send the secret message using packet payload data, algorithm is as follows:

Step 1: Capture network protocol packets.

Step 2: Convert the packet data into binary format and in tabular arrangement.

Step 3: Take secret message from the sender.

Step 4: Convert the secret message to binary format and in tabular arrangement.

Step 5: Generate key using following methodology:

Step a: Select random bit from 0 to 7 including both, from the packet payload data.

Step b: Count the number of 1's from the selected bits.

Step c: If number of 1's is even, secret bit is 0.

Step d: If number of 1's is odd, secret bit is 1.

Step e: Match the generated bit with the message secret bit.

Step f: If matches, key generated successfully.

Step g: If not matches then complement the bit and append any of '-1 to -9' with the key.

Step 6: Encrypt the generated key using RSA cipher.

Step 7: Send the original data packet and encrypted key to the receiver.

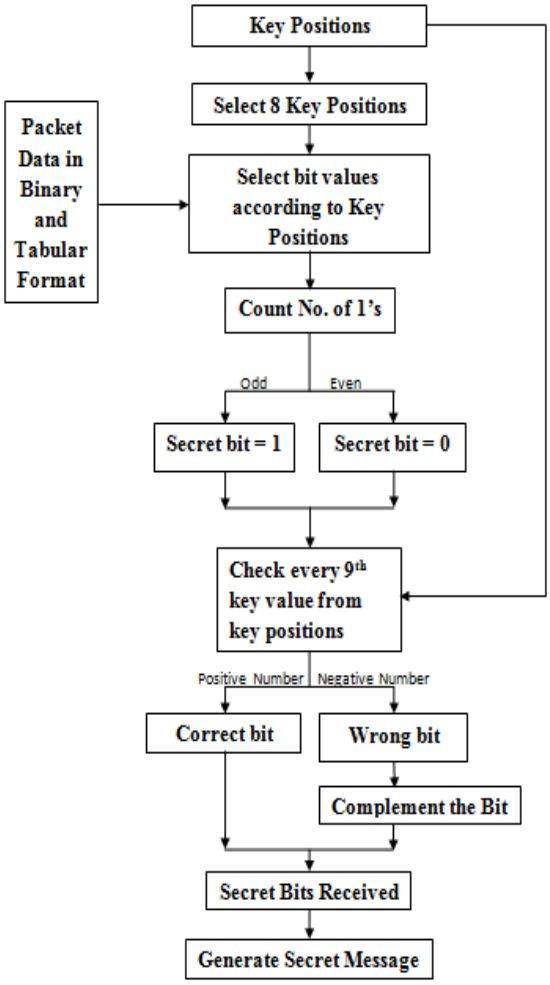

Figure 4 Flow of Comparison

Receiver Side Workflow

To receive the secret message using packet payload data, algorithm is as follows:

Step 1: Receive the packet payload data and encrypted key.

Step 2: Convert the packet payload data into binary format and in tabular arrangement.

Step 3: Decrypt the key using RSA cipher.

Step 4: From the key positions, select the corresponding bits from the packet payload data.

Step 5: Comparison is done using following methodology:

Step a: Count the number of 1's from the selected bits.

Step b: If number of 1's is even, secret bit is 0.

Step c: If number of 1's is odd, secret bit is 1.

Step d: Check every 9th key value from key positions.
Step e: If its 'positive number', then secret bit is correct.
Step f: If its 'negative number', then secret bit is incorrect. Complement the secret bit.

Step 6: Get the secret bits.
Step 7: Convert the secret bits to get secret message.

V. EXPERIMENTAL RESULTS

The experimental results are calculated by downloading number of packets of TCP protocol using Wireshark. The packet data is used to generate key positions. The bits of key generated are then encrypted using RSA encryption algorithm.

The snapshot of the packet capture using wire shark is as below:

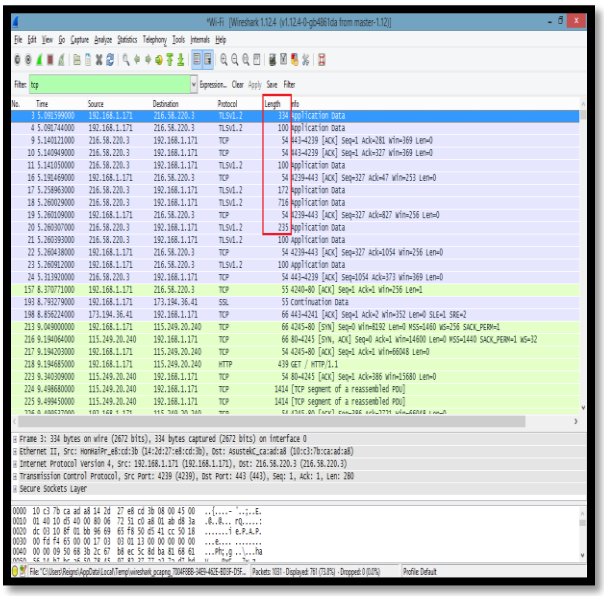

Figure 5 Packet Capture

The length of the captured packet data is in bytes. It resembles the length of data in the payload. The equivalent conversion of the length i.e. bytes to bits is shown. From the bits we can calculate the number of secret bits that could be transferred easily. Dividing the number of bits by 64 will give us the length of secret bits that could be transferred. So for a particular packet the maximum number of secret bits that could be transferred is shown effectively.

Table 1 Experimental Results

No.	Length (Bytes)	Bits	Secret bits
1.	334	2672	41
2.	100	800	12
3.	54	432	6
4.	54	432	6
5.	100	800	12
6.	54	432	6
7.	172	1376	21
8.	716	5728	89
9	54	432	6
10	235	1880	29

VI. PERFORMANCE EVALUATION

From the comparison it could be shown that the stego-capacity is more as compared to existing ones. Also the proposed technique is having high undetectablility. Following table shows the performance with other methods:

Table 2 Performance Evaluation

No.	Title	Protocol	Capacity	Detection
1.	Network Packet Payload Parity Based Steganography [2]	UDP	1 bit/packet	Difficult
2.	Practical Internet Steganography: Data Hiding in Ip[3]	IP	Depends on selected field	-
3.	Retransmission Steganography and its detection [4]	TCP	180 byte/s	Not easy
4.	Length Based Network Steganography using UDP Protocol [5]	UDP	456 bit/s	Not easy
5.	StegTorrent: a Steganographic Method for the P2P File Sharing Service [6]	UDP	270 bit/s	Difficult
6.	PadSteg: Introducing inter-protocol Steganography [7]	TCP and ARP	32 bit/s	Difficult
7.	Stream Control Transmission Protocol Steganography [8]	TCP and UDP	Depends on selected field	-
8.	Proposed Method	TCP	Depends on size of Packet Data	Difficult

Above table shows that comparatively, proposed method is more anonymous and has larger message hiding capacity.

VII. CONCLUSION

Network protocols are the modern carriers for implementing steganography. Everyday a large amount of data is transferred from one node to the other node. Numerous numbers of packets are exchanged across the globe. In the present thesis, how

a packet data can be used to send secret message using reference of positions is shown and it's more covert as compared to other approaches. The proposed technique is difficult to trace and the overall stego capacity is increased, making it a creamy approach to use.

REFERENCES

[1] Sebastian Zander, Grenville Armitage and Philip Branch, "A Survey of Covert Channels and Countermeasures in Computer Network Protocols", IEEE Communications Surveys & Tutorials, Volume 9, No. 3, 3rd Quarter 2007, page no. 44-57.

[2] Osamah Ibrahiem Abdullaziz, Vik Tor Goh, Huo-Chong Ling and KokShiek Wong, "Network Packet Payload Parity Based Steganography", Conference on Sustainable Utilization and Development in Engineering and Technology, IEEE 2013, page no. 56-59.

[3] Deepa Kundur and Kamran Ahsan."Practical Internet Steganography: Data Hiding in IP", In Proceedings of Texas Workshop on Security of Information Systems, April 2003.

[4] Wojciech Mazurczyk, Milosz Smolarczyk and Krzysztof Szczypiorski. "Retransmission Steganography and its detection", Warsaw University of Technology, Institute of Telecommunications, Warsaw, Poland, Springer 05 November 2009, DOI 10.100709-0530-1.

[5] Anand S Nair, Abhishek Kumar, Arijit Sur and Sukumar Nandi. "Length Based Network Steganography using UDP Protocol", Department of Computer Science and Engineering, Indian Institute of Technology, Guwahati, IEEE 2013, page no. 726-730.

[6] Pawel Kopiczko, Wojciech Mazurczyk and Krzysztof Szczypiorski. "StegTorrent: a Steganographic Method for the P2P File Sharing Service", Security and Privacy Workshops,IEEE 2013, page no 151-157.

[7] Bartosz Jankowski, Wojciech Mazurczyk and Krzysztof Szczypiorski. "PadSteg: introducing inter-protocol steganography", Institute of Telecommunications, Warsaw University of Technology, Warsaw, Poland, ISSN 1018-4864, Volume 52, Number 2, Springer 2013, page no. 1101-1111.

[8] Wojciech Fraczek, Wojciech Mazurczyk and Krzysztof Szczypiorski. "Stream Control Transmission Protocol Steganography", International Conference on Multimedia Information Networking and Security, IEEE Computer Society, 978-0-7695-4258-4/10, 2010, page no. 829-834.