Internet Protocol (IP) Steganography Using Modified Particle Swarm Optimization (MPSO) Algorithm

Sana Adnan Abbas

Directorate General of Education in Diyala Governorate

sanaalasadi2000@yahoo.com

Received: 16 November 2017 Accepted: 14 January 2018

Abstract

All users of the Internet are dealing with Internet protocol (IP). Therefore, it is a kind of smart when IP packet is used as a cover for hiding secret messages. Hence, this paper presents network security by using IP packets steganography. The proposed method is developed by an intelligent technique called modified particle swarm optimization (MPSO). In the proposed system, the secret characters are merged in IP identification field using a proposed embedding algorithm so as the system be more robust in face of attackers. The use of MPSO algorithm is considered as a development to enhance the proposed system, by selecting most convenient packets for hiding inside. The total time for hiding (53) characters in (100) raw sample, if MPSO is employed, takes approximately (128 μs for transmitting and 109 μs for receiving). while the delay time for the same sample without using MPSO is approximately (121μs for transmitting and 98 μs for receiving).

Keywords: Internet protocol, Packet steganography, Transmission control protocol (TCP\IP), Particle swarm optimization(PSO), Modified Particle swarm optimization (MPSO).

DIYALA JOURNAL FOR PURE SCIENCES

Internet Protocol (IP) Steganography Using Modified Particle Swarm Optimization (MPSO) Algorithm

Sana Adnan Abbas

أخفاء أتفاقية الأنترنيت بأستخدام خوارزمية أمثلية السرب الجزيئية المطورة

سناء عدنان عباس

مديرية تربية ديالى

ألخلاصة

يتعامل جميع مستخدمي الأنترنيت مع أتفاقية الأنترنيت لهذا فأن أستخدام حزمة أتفاقية الأنترنيت كغطاء لأخفاء الرسائل السرية يعتبر نوع من الذكاء. بالتالي فأن هذا البحث يقدم أمنية الشبكة بأستخدام إخفاء (ستيكانوكرافي) حزم أتفاقية الأنترنيت. لقد تم تطوير الطريقة المقترحة من خلال تقنية ذكية تسمى أمثلية السرب الجزيئية المطورة. في الطريقة المقترحة تم دمج الحروف السرية في حقل التعريف لحزمة أتفاقية الأنترنيت بأستخدام خوارزمية وضع مقترحة لكي يكون النظام أكثر قوة في وجه المهاجمين. يعتبر أستخدام خوارزمية أمثلية السرب الجزيئية المطورة تطوير لتحسين النظام المقترح من خلال أختيار الحزم الأكثر ملائمة للأخفاء في داخلها. الوقت الكلي لأخفاء (53) حرف داخل (100) عينة خام بأستخدام خوارزمية أمثلية السرب الجزيئية المطورة هو (128) مايكرو ثانية للأرسال و (109) مايكرو ثانية للأستلام. بينما وقت التأخير لنفس العينة بدون أستخدام خوارزمية أمثلية السرب الجزيئية المطورة هو (121) مايكرو ثانية للأرسال و (98) مايكرو ثانية للأستلام.

الكلمات المفتاحية: أتفاقية الأنترنيت، ستيكانوكرافي الحزمة، أتفاقية سيطرة الأرسال / أتفاقية الأنترنيت ، أمثلية السرب الجزيئية . أمثلية السرب الجزيئية المطورة.

Introduction

Most of the Internet users are utilized the TCP and IP protocols. This makes these two protocols as a part of everyday life. The security of computer networks has incremented partially through the last years, while the security of computers has a high degree of attention in this field. Therefore, the issues of network security yet exist. Hiding in network protocol is not the most recognized origin of blusters, and actually neglected by the people, but it comprises an actual threat [1]. Because of protocol existence in every communication, TCP/IP steganography has been a motivating research point. Embedding data in header fields within TCP/IP it is generally supposed that undetectable steganography. Header fields can be filled with “random” data, such as (ISN) initial sequence number, an identifier of IP, or the TCP least significant bits. These

fields in natural manner show enough structure and randomness to be effective and reliable distinguish from unchanged message [2].

Headers of protocols can be used as a carrier for the hidden channel, if a set of values can be taken by header field, each of which seems reasonable to the positive observer. The observer should not be able to differentiate whether the header was created by the protocol or by a steganographic technique [3]. This work is about steganography in TCP/IP packet header (since Internet packet is encapsulated under TCP segment) using swarm intelligence specifically (MPSO) exploiting TCP/IP network traffic to formulate a covert channel transporting secret message in the TCP/IP protocol header field.

Related Work

Rowland [3], in 1997 described a prototype implementation of steganography using TCP initial sequence number, or Acknowledgment Number or IP Identification. The chosen field is replaced with the data to be sent, so can be detected by observing that fields, or by comparing the data observed with statistical patterns of suspected plaintext. An idea of using Internet protocol checksums for covert communication is discussed by Christopher Abad [4] in 2001, but this idea faced a lot of problems since hiding data in IP checksum can be easily discovered. Also, techniques for detecting covert channels as well as possible places to hide data in the TCP stream are presented and analyzed by the author.

K. Ahsan [5], in 2002, a proposed steganographic method that uses IP fragmentation fields, and DF (Don't Fragment) flag as a covert data carrier. If the sender knows the correct MTU (Maximum Transfer Unit) for the end-to-end path to the receiver and packets size is less than MTU, then DF can be set to arbitrary values.

The options field in TCP header can be used to store data by creating nonexistent options, the timestamp option is common. The Time Stamp Value (TSV) encodes the timestamp clock value of the sender. In most case, the least significant bits of this field appears random. Steganography system was proposed by Giffin et.al. [6] in 2002, formulating network covert channel based on modulating the least significant bit of the TCP timestamps in TCP/IP packets transmitted by a host.

The Initial Sequence Number (ISN) of TCP header was also used. This has been successfully applied by Joanna [7] in 2004, the work is known as "NUSHU". It was a passive covert channel proof of concept for Linux, that modify the ISN of existing segments to code information. Information is coded in ISNs and encrypted with Data Encryption Standard (DES), thus ISNs appear to be random.

Network steganography method is presented by Jones et al [8] in 2004, they proposed a covert channel in the IP Time to Live (TTL) field to trace back IP packets without using the source address field.

Cauich et. al. [9], in 2005, described how to use fragment offset fields to carry hidden data between intermediate nodes but under the condition that the packet is not fragmented. Additionally, in selected packet reserved flag is used to mark packet so that the receiver can distinguish between real and covert fragments.

Lewis and Murdoch [10], in 2005, proposed transmitting hidden information by modulating the size of the fragments to match the hidden data inserted into fragment offset field.

Sebastian et. al. [11], in 2006, proposed a novel covert channel in the IP header's Time to Live (TTL) field. Although this header field was never intended to be used for communication, they demonstrated that a covert sender can encode information in the TTL fields of subsequent packets, which can be later decoded by the receiver.

M. Hussain [12], in 2011, a proposed technique based on packet length and payload to achieve high capacity for data embedding. To consider the normal traffic distribution, utilizing the real network packet length for covert communication. A retransmission of packet filled with (covert data) stego-data. This proposed scheme filled covert data into payload of the packet to increase the covert data capacity.

R.M. Goudar [13], in 2012, presents a work focusing on Identification field of the IP header to hide secret encrypted data. The proposed method is to use the entire (16) bit field to hide the secret encrypted message.

IP and TCP

The Internet Protocol (IP) uses the network layer of TCP/IP protocol suite, it is an unreliable, connectionless protocol. The function of IP is to transfer packets from a computer to another [14]. A packet is a chunk of data, related to self-referential data (metadata). Metadata point out where the data have to be transmitted, the source of coming, the metadata part of a data chunk is called a header. The fields of the IP header are represented in Figure (1) [15].

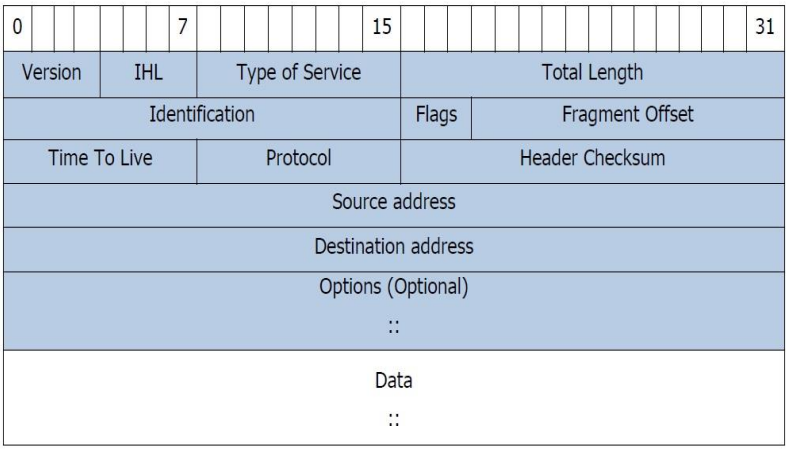

Figure 1: The IP header fields [5]

The largest length of an IP packet is (64 KB) but usually, it is smaller than this value on most transfer connections, this limitation for the specified connection is called Maximum Transmission Unit (MTU). MTU relays on the kind of the transmission link.

Communication between two computers can be handled by TCP. This protocol is a reliable protocol, and data corruptions are detected and the packets are sent out again if necessary. Figure (2) describes the TCP header fields [15].

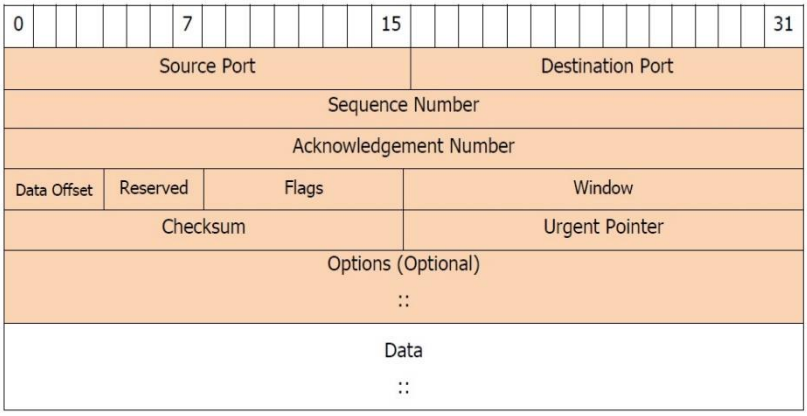

Figure 2: The TCP header fields after [15]

TCP data are encapsulated over IP as shown in Figure (3) [15].

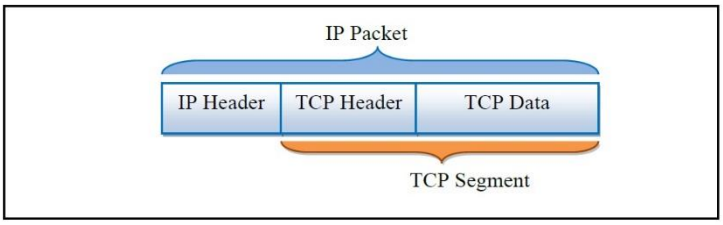

Figure 3: TCP segment encapsulation over IP [15]

PSO and MPSO

The standard PSO scheme includes a swarm of particles, which are started with a population of random solutions. The population particles have positions symbolized by a position-vector ***xi***, and also have a velocity symbolized by a velocity-vector ***vi***. These particles have the ability to remember their own best positions so far in a vector *i*-th, and their d-dimensional values are pbest (pid). The best positions in the swarm are then stored in the vector *i*-th, and their d^{th} dimensional value is gbest (pgd). The velocity (vid) from prior velocity to the new velocity during the iteration time *t*, is updated using formula (1). However, the new position (xid) is updated by formula (2) [16].

$$V(id+1) = w *vid + c1 *r1* (pgd -xid) +c2 * r2 * (pid -xid)\ldots\ldots\ldots\ldots (1)$$

$$X(id+1) = xid + v(id+1)\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

Where (r1 and r2) are the random numbers, which are utilized to preserve the variance of the population, and they are uniformly distributed in the interval [0,1]. However, i =1,2,…..,n, c1 is a positive constant, called coefficient of the self recognition component; c2 is a positive constant, called coefficient of the social component. The basic algorithm for the PSO can be shown in the algorithm (1) [16]:

Algorithm 1: PSO Algorithm [16]

Input: Feed algorithm parameters (c1,c2,r1, r2,Vmax, W, Max-Iteration, swarm size.

Output: Get the highest fitness as found by PSO.

Step1: Create initial particles and velocities randomly to form a swarm.

Step2: Determine fitness of each particle.

Step3: Update the particle when the current position is better than the previous one.

Step4: Use equations (1) and (2) to find the best particle of the swarm,

Step5: If the highest fitness is found or max, the number of iterations has exceeded, go to step (6), else go to step (2).

Step6: Record the best value and exit.

Modified Particle Swarm Optimization (MPSO) was introduced by Eberhart and Shi in 1997 and 1998. This algorithm proposed that birds own a memory about the previous best and worst positions. Hence, particles own two experiences, a bad experience supports that each bird can remember its previous worst position. In order to determine the new velocity, the worst experience of each bird is considered. Therefore, the new velocity update equation is given in Eq. (3) [17].

$$V_{i+1} = w \times V_i + C1g \times r1 \times (Pbest_i - S_i) + C1b \times r2 \times (S_i - Pworst_i) + C2 \times r3 \times (gbest1 - S_i) \quad \text{..... (3)}$$

where (r1, r2, r3) are uniformly distributed random numbers in the range [0 to 1]. C1g is acceleration coefficient which speeds up the bird to its best position, C1b is acceleration coefficient which speeds up the bird away from its worst position. P_{Worst} i is also an acceleration

coefficient that speeds up the bird away from its worst position of the particle i. MPSO algorithm is executed by the same way of the algorithm (1) but with exchange Eq. (1) by Eq. (3) [17].

The Proposed System

The general block diagram of the transmitter and receiver sides of the proposed system is shown in figure (4) and figure (5).

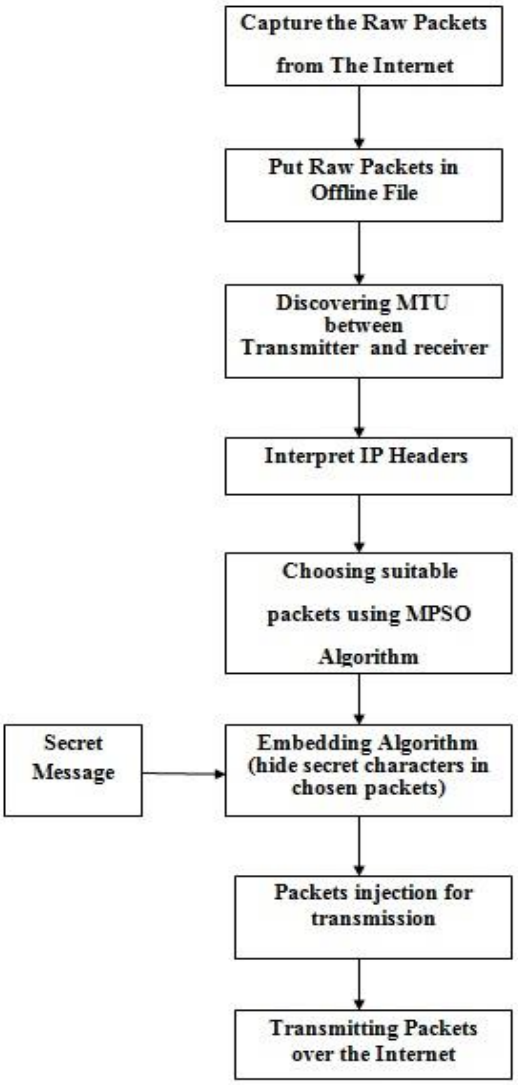

Figure 4: Block diagram of the transmitter side for the proposed system

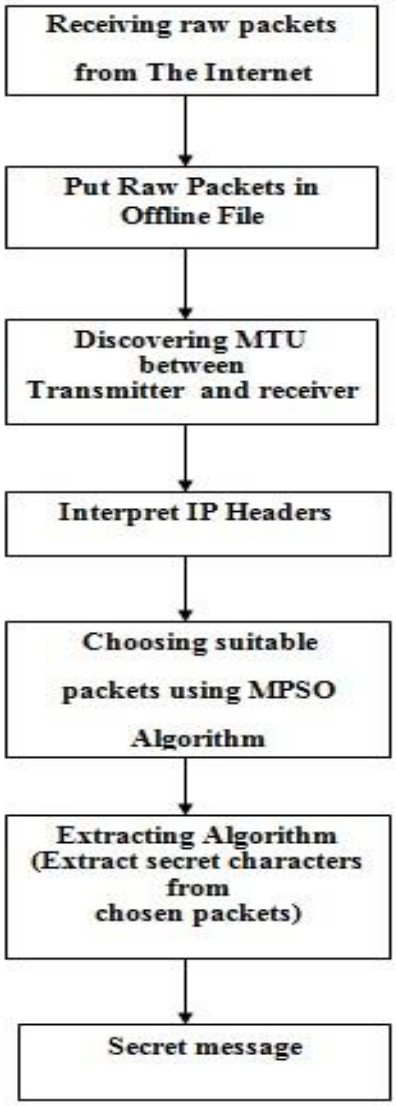

Figure 5: Block diagram of the receiver side for the proposed system

Transmitter Side

At transmitter side in figure (4), raw packets of normal traffic are captured by transmitter host of address (192.168.1.101). The offline file is generated by Wireshark application. This file contains information of raw packets with details about headers, data, and protocol type. The transmitter discovers MTU in the path between sender and receiver using Linux commands.

MTU will be used in MPSO algorithm as value to determinate the suitable maximum length for identification field in IP header of the chosen packet. The proposed system hosts (transmitter and receiver) are designed to interpret the raw packets concentrating on particular features in the packet header. These features are represented by the protocol type field, the packet total length field and the identification field in the IP header. Algorithm (2) represents the applied MPSO algorithm that is used to choose packets depending on features of packets such as the protocol type field in the header of each packet as it should be TCP. The fitness value of the algorithm depends on the range of packet total length as in formula (4).

$$500 \text{ Byte} \leq \text{chosen packet length} \leq 1452 \text{ Byte} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (4)$$

Algorithm 2: Applied MPSO Algorithm

Input: N of population packets.
Output: M fittest particles.
Step1: Initialize a population of N particles.
Step2: Calculate fitness of first raw.
(N=1; MTU according to formula (4) and TCP protocol.
Step3: While (! EoF N raw Packets)
For (i=1 to N)
For (J=1 to M)
If (protocol type = TCP)
If (500 < Total current packet length(L) ≤ MTU
If (features of the current packet is better than previous one)
Update according to equations (3) and (2).
Choose the j^{th} packet.
End of If.
End of If.
End of If.
End of For.
End of For.
Step4: Sort chosen packet according to total lengths.
End While.
Step5: Return chosen packets.

An embedding algorithm is designed to embed one secret character per chosen packet. The identification field is (16-bit) and the designed embedding algorithm uses the least significant (8-bit) only to make a relatively small change to the value of original identification field as shown in the algorithm (3).

Algorithm 3: Proposed Embedding Algorithm

Input: Chosen packets and secret data.
Output: Stego packets.
While (Not End of Secret Characters)
Step1: Read character of secret data.
Step2: Convert The read character into ASCII value.
Step3: Read raw IP ID.
Step4: Zeroing least significant bits (Temp=raw IP ID and 65280).
Step5: New IP ID = Temp + Read Character.
Step6: Return packet for injection with new IP ID field.
END of While

The offline packets are reconstructed but with embedding secret data in the identification field of the chosen packet header, then the proposed system injects these packets for transmission to receiver IP address.

Receiver Side

At receiver side in figure (5), the incoming raw packets are received normally. The receiver of the proposed system at particular time, will receive and save only the traffic of offline file that contains stego packets which have been injected by the sender, since the receiver stops receiving from other servers. All the processes in receiver side are done by the same way in transmitter side except the embedding algorithm is changed by the extracting algorithm as shown in the algorithm (4).

Algorithm 4: Proposed Extracting Algorithm

Input: Stego packets.

Output: Secret Data.

For (i=1 to number of secret characters)

Step1: Read received identification field of the received packet.

Step2: Zeroing most significant eight bits (RE char=RE IP ID and 255).

Step3: Convert RE char into equivalent character.

Step4: Put an extracted character in secret data file.

END of For.

Results

The analysis of protocol type field in IP packets proved that most of IP packets use TCP protocol type as shown in the table (1) using 100 raw packets.

Table 1: Analysis of protocol type field in IP packets

Protocol Name	Usage %
TCP	75
UDP	19
Others(ICMP)	6

The total length analysis of the TCP/IP packets gives that the length of these packets is between (40 bytes - 1452 bytes). MPSO algorithm is applied with the proposed system on a sample of (100 raw packets) as cover to embed same secret message of (40 Bytes) MTU value range (from 500 to 1452 byte) in the algorithm. MPSO algorithm would choose packets that fit the proposed criteria as shown in figure (6).

Internet Protocol (IP) Steganography Using Modified Particle Swarm Optimization (MPSO) Algorithm

Sana Adnan Abbas

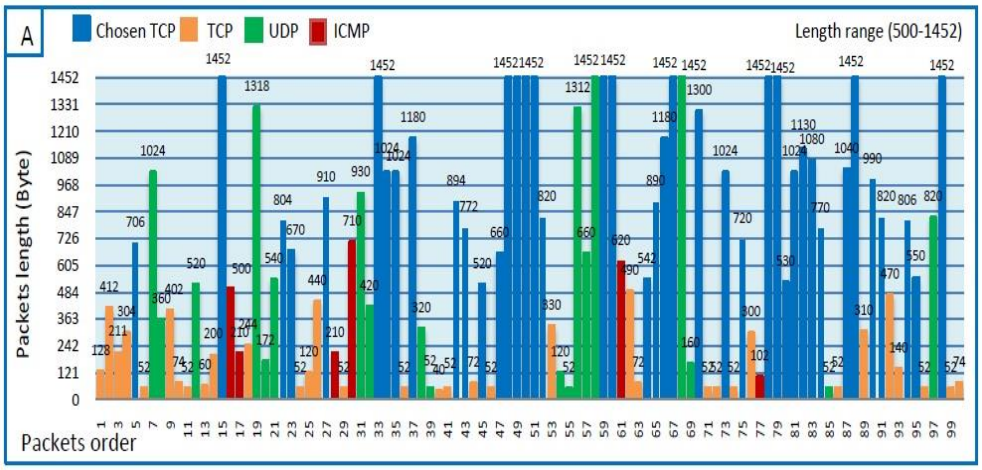

Figure 6: Histogram of Selected Packets by MPSO

The relation between a number of iteration for MPSO and number of secret characters that are to be embeded is shown in figure (7).

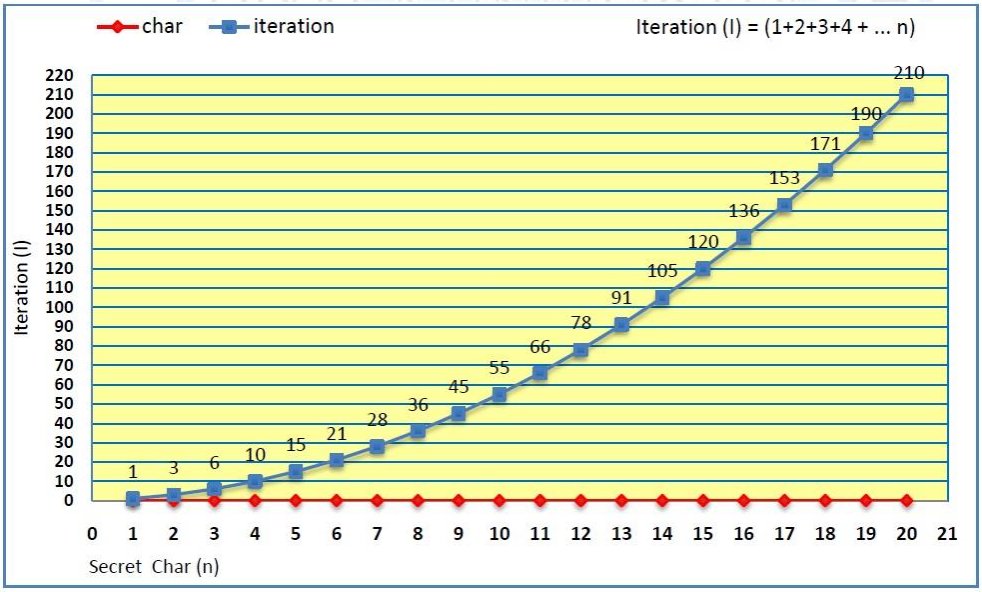

Figure 7: Relation between number of MPSO iterations and number of secret characters

MPSO algorithm selects the most convenient packets from (100) packets sample as shown in the table (2).

Table 2: Analysis of Sample Raw Packets

Order of Packets	Type of Protocol	Length	IP ID	Order of Packets	Type of Protocol	Length	IP ID
1	TCP	128	28646	51	TCP	1452	1452
2	TCP	412	28647	52	TCP	820	820
3	TCP	211	28648	53	TCP	330	330
4	TCP	304	12558	54	UDP	120	120
5	TCP	706	12559	55	UDP	52	52
6	TCP	52	0	56	UDP	1312	1312
7	UDP	1024	1570	57	UDP	660	660
8	UDP	360	1571	58	UDP	1452	1452
9	TCP	402	52313	59	TCP	1452	1452
10	TCP	74	52314	60	TCP	1452	1452
11	TCP	52	0	61	ICMP	620	620
12	UDP	520	11546	62	TCP	490	490
13	TCP	60	0	63	TCP	72	72
14	TCP	200	2831	64	TCP	542	542
15	TCP	1452	2832	65	TCP	890	890
16	ICMP	500	45312	66	TCP	1180	1180
17	ICMP	210	45313	67	TCP	1452	1452
18	TCP	244	33023	68	UDP	1452	1452
19	UDP	1318	58923	69	UDP	160	160
20	UDP	172	58924	70	TCP	1300	1300
21	UDP	540	58925	71	TCP	52	52
22	TCP	804	58215	72	TCP	52	52
23	TCP	670	58216	73	TCP	1024	1024
24	TCP	52	0	74	TCP	52	52
25	TCP	120	61589	75	TCP	720	720
26	TCP	440	61590	76	TCP	300	300
27	TCP	910	61591	77	ICMP	102	102
28	ICMP	210	37029	78	TCP	1452	1452
29	TCP	52	0	79	TCP	1452	1452
30	ICMP	710	22842	80	TCP	530	530
31	UDP	930	48819	81	TCP	1024	1024
32	UDP	420	48820	82	TCP	1130	1130
33	TCP	1452	64420	83	TCP	1080	1080
34	TCP	1024	64421	84	TCP	770	770
35	TCP	1024	64422	85	UDP	52	52
36	TCP	52	0	86	TCP	52	52
37	TCP	1180	27034	87	TCP	1040	1040
38	UDP	320	61030	88	TCP	1452	1452
39	UDP	52	61031	89	TCP	310	310
40	TCP	52	0	90	TCP	990	990
41	TCP	52	0	91	TCP	820	820
42	TCP	894	44964	92	TCP	470	470
43	TCP	772	44965	93	TCP	140	140
44	TCP	72	44966	94	TCP	806	806
45	TCP	520	44967	95	TCP	550	550

46	TCP	52	0	96	TCP	52	52
47	TCP	660	11032	97	UDP	820	820
48	TCP	1452	9371	98	TCP	1452	1452
49	TCP	1452	9372	99	TCP	52	52
50	TCP	1452	9373	100	TCP	74	74

The total time for hiding (53) characters in (100) raw sample, if MPSO is employed, takes approximately (128 μs for transmitting and 109 μs for receiving). while the delay time for the same sample without using MPSO is approximately (121μs for transmitting and 98 μs for receiving).

Discussion and Conclusion

The proposed system is successfully implemented however, there are some worthy points need to be discussed or inferred:

1. The proposed system is the steganography system that exploited only (8-bit) from the IP ID field while the field status is used, making an unobservable change to the cover field value. This feature supports high security for the proposed system.
2. The proposed system employs swarm intelligence, represented by MPSO algorithm. This can be considered as a major enhancement in the protocol steganography systems because MPSO algorithm performs as an optimization technique to choose best-suited packets.
3. The results of section (5) shows that small delay time (measured in microseconds) is added to both sides of the proposed system because of using MPSO algorithm. The delay time is affected by many factors that increase it. These factors that affect the delay time in the proposed system are:

   - Using MPSO algorithm to choose suitable packets.
   - The size of the secret message (in bytes).
   - The assigned length range of the chosen packets.
   - The number and size of the raw packets.

References

1. T. Handel and M. Sandford, “Hiding Data in the OSI Network Model”, Information Hiding Workshop (IH 1996), Springer Press, vol. (1174) of LNCS, pp. (23–38), Cambridge, UK, May/June, 1996.
2. Artz, D., “Digital Steganography: Hiding Data within Data”, IEEE Internet Computing Journal, vol. (3), No. (1), pp. (75 – 80), May- June, 2001.
3. Craig H. Rowland, “Covert Channels in the TCP/IP Protocol Suite”, (First Monday Jornal on the Internet), Vol. 2(5), No. (5)., May, 1997.
4. Christopher Abad, "IP Checksum Covert Channels and Selected Hash Collision" Internet Survey, 2001, Available at http://www.citeulike.org/user/grrtrr/article/1413458 , April, 2011
5. K. Ahsan, “Covert Channel Analysis and Data Hiding in TCP/IP “, MS.c Thesis, Dept. of Electrical and Computer Engineering, University of Toronto, August, 2002.
6. J. Giffin, R. Greenstadt, P. Litwack, and R. Tibbetts, “Covert Messaging in TCP ”, Sprin Press, vol. (2482) of LNCS, pp. (194–208), San Francisco, CA, US, April, 2002.
7. Rutkowska Joanna, “The Implementation of Passive Covert Channels in the Linux Kernel”, 21st Chaos Communication Conference, Chaos Computer Club e.V., Berlin, pp. (44-52), Germany, 2004.
8. E. Jones, O. Le Moigne, J.-M. Robert, “IP Traceback Solutions Based on Time to Live Covert Channel”, Proceedings of 12th IEEE International Conference on Networks (ICON), pp. (451–457), November, 2004.
9. Cauich E., Gomez Cardenas R., Watanabe, R., “Data Hiding in Identification and Offset IP Fields”, Proc. 5th Int’l. School and Symp. Advanced Distributed Systems (ISSADS), January, 2005.
10. [10] Stephen Lewis and Steven J. Murdoch, “Embedding Covert Channels into TCP/IP”, Information Hiding Workshop 2005 proceeding, Cambridge, pp. (1-7), 2005.
11. Sebastian Zander, Grenville Armitage, Philip Branch, “Covert Channels in the IP Time to Live Field”, Australian Telecommunication Networks & Applications Conference (ATNAC), Australia, December, 2006.

12. Mehdi Hussain and M. Hussain, " A High Bandwidth Covert Channel in Network Protocol", International Journal of Advanced Science and Technology (SZABIST), Islamabad, Pakistan Vol. 30, May, 2011.

13. R.M. Goudar, Prashant N. Patil, Aniket G., "Secure Data Transmission by using Steganography", Information and Knowledge Management, Pune University, MAE, India, Vol (2), No. (1), 2012.

14. J. Millen, "20 Years of Covert Channel Modeling and Analysis", Proceedings of IEEE Symposium Security and Privacy, pp. (113–114), May, 1999.

15. N. Lucena et al., "Syntax and Semantics Preserving Application Layer Protocol Steganography", Proc. 6th Information Hiding Wksp., pp. (82-91),May, 2004.

16. Ismail Khalil Ali, " Intelligent Cryptanalysis Tool Using Particle Swarm Optimization", Ph.D.Thesis, Univercity of Technology, Department of Computer Science, 2009.

17. Iman Soltani, Mohammad Sarvi, and Fatemeh Salahian, " Various Types of Particle Swarm Optimization-based Methods for Harmonic Reduction of Cascade Multilevel Inverters for renewable energy sources ", International Journal of Innovation and Applied Studies, ISSN 2028-9324 Vol. 2 No. 4 Apr. 2013, pp. 671-681.