Military Technical College
Kobry El-Kobbah,
Cairo, Egypt

8th International Conference
on Electrical Engineering
ICEENG 2012

New Steganographic Method for Data Hiding in the IP ID Field

By

Manal A. Shehab Noha O. Korany *

Abstract:

Many covert channels could be generated by using different IP header fields to introduce data hiding schemes. This paper presents the scheme and the description of a new suggested steganographic method for data hiding in the IP ID field.

In the suggested method; an appropriate encryption algorithm and key could be used to encrypt the plaintext character so the ciphertext value could be presented in the form of the 8-bit binary representation and then could be embedded within the IP ID field using a new suggested embedding algorithm that uses pre-agreed key and direction.

The method may be applied with either IPv4 or IPv6 packets. With IPv6; IP packet fragmentation is required by the packet source. In this method; the packet source should be the steganogram sender. The method has the advantages of being resistant to packet filtering and stateful inspection firewalls and it could be applied through different network scales and characteristics. To provide more confidentiality to the hidden data in the IP ID field; the paper suggests implementing the IPsec encryption in the tunnel mode between the steganogram sending and receiving gateways.

Keywords:

Steganographic, data hiding and IP ID field

---

* Electrical Engineering Department, Faculty of Engineering, Alexandria University- Egypt

1. Introduction:

There are some necessary considerations for the scheme which hides data in the IP ID field of the IPv4 header or the IPv6 fragment header extension. The data hiding scheme must be robust in the meaning of making the resulted stego datagrams appear as normal datagrams and the scheme must accept any datagram that can be interpreted. For the design aspect of the value of the IP ID field; it should be unique and immutable (unchangeable) value for the specified packet source/ destination pair and protocol combination for the time at which the packet (or any fragment of it) could be alive in the network. Also the followed IP ID field values should reflect random impression without appearing as if they are related by certain function or relation.

In this paper; we will assume that Alice is the Steganogram Sender (SS) and Bob is the Steganogram Receiver (SR), and they would use the basic covert channel framework model which is shown in figure 1. In this framework; the network packet Np is the cover object that is used to conceal the covert information. Alice uses the stego algorithm with the secret key, the covert plaintext P and the original packet Np to generate the stego-network packet S_k, then transfers S_k to Bob. For reasons of security; the used secret key and the stego algorithm must be known only to Alice and Bob. Bob needs to apply the extraction/detection algorithm and key to the stego packet to extract and decode the covert information [1].

The network transmission channel is modeled as non-ideal channel that could present an incidental processing on the stego network packet that could affect the covert information flow by introducing position error(s) in the packets sequence. The extracted covert information which may be possibly affected by the non-ideal channel is denoted as P*. To save the covert channel from detection; intermediate nodes shouldn't find any difference between the original network packet and the stego-network packet in their processing.

In the case of embedding hidden data within the IP ID field; the covert plaintext P equals the retrieved covert information that could possibly be affected by the non-ideal network channel P* (i.e. P = P*), because the non ideal network channel characteristic doesn’t affect the value of the IP ID field.

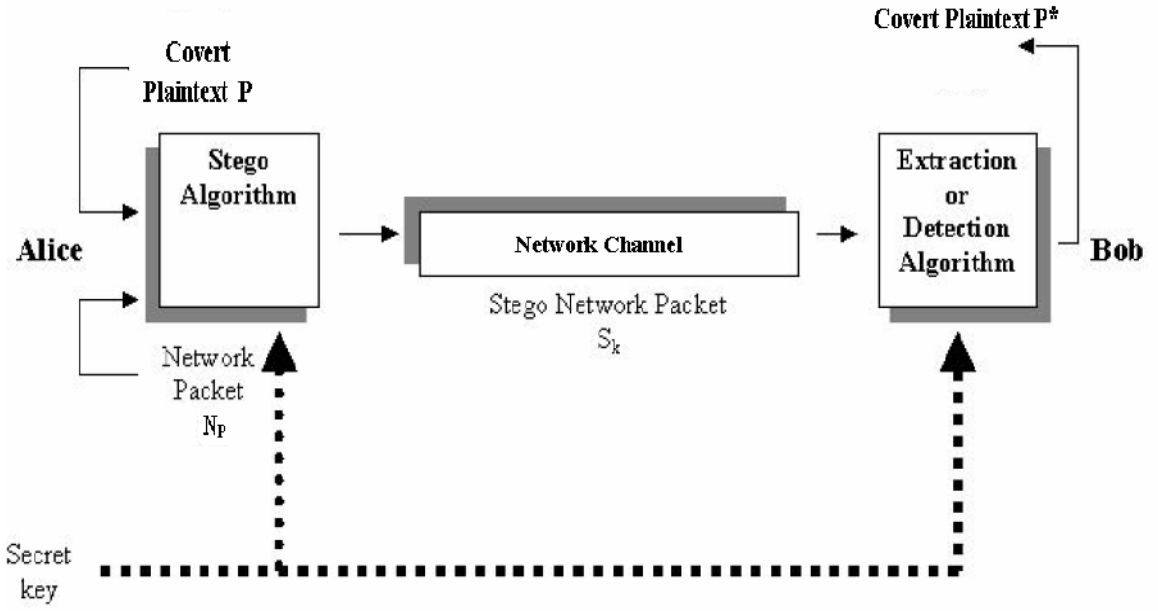


Figure (1): General covert channel framework

Ashan [1] suggested using the IP ID field of the IPv4 header to conceal an (8-bit) covert data after encoding it by a pre-defined manner. He suggested embedding the resulted (8-bit) encoded data in the IP ID field in either the first or the last eight bits of the IP ID field according to a previous agreement between the SS and the SR, then generating the rest of the 8 bits of the IP ID field randomly to form the 16-bit IP ID field as $[i_{15},: : ,i_1 ,i_0]$ which would carry the covert encoded message as RRRRRRRR$c_7c_6c_5c_4c_3c_2c_1c_0$ or $c_7c_6c_5c_4c_3c_2c_1c_0$ RRRRRRRR, where $c_7c_6c_5c_4c_3c_2c_1c_0$ is the 8-bit ciphertext and R is a random generated bit. Generating the remaining 8 bits of the IP ID field of the IPv4 randomly serves the uniqueness of the 16-bit IP ID value for the specified source/ destination pair and protocol combination through the packet lifetime in the network.

The following sections describe and evaluate a new suggested steganographic method for data hiding in the IP ID that is denoted by M1 for simplicity [2].

2. New Suggested Steganographic Method (M1) for Data Hiding in the IP ID Field:

2.1 The Scheme of the M1 Steganographic Method:

The new steganographic method M1 uses two keys, the encryption key 'e' and the embedding reference pointer key 'k' [2]. Both encryption and embedding algorithms as well as both keys e and k should be known for Alice and Bob only.

The M1 steganographic method may be applied to IPv4 or IPv6 packets. But with IPv6; IP fragmentation is required in order to add the IPv6 fragment header extension (which includes the IP ID field) to the IPv6 packet. The IP ID field in the IPv6 fragment header extension is a 32-bit field. So embedding 8 bits hidden data in the 32-bit IP ID field involves generating the remaining 24 bits of the IP ID field randomly to provide more uniqueness for the resulted IP ID than in the case of using IPv4. Also the 32-bit IP ID in the IPv6 provides more available hidden size for M1 than that in IPv4, because we could embed two 8-bit hidden characters (i.e. 16 hidden bits) in the 32-bit IP ID and generate the remaining 16 bits randomly.

The M1 steganographic method uses the framework which is shown in figure 2 to hide data in the IP ID field. At the SS side, the steganographic algorithm includes three operations as follows:

1- Encrypting the ASCII code of the plaintext character P using an appropriate encryption algorithm and key 'e'.

2- Converting the resulted ciphertext from decimal form $(C)_{10}$ to its 8-bit binary representation $(C)_2$. [This is an optional operation depends on the used encryption algorithm].

3- Embedding the resulted 8-bit ciphertext code in the IP ID field using the pre-agreed embedding algorithm and reference pointer key 'k'.

At the SR side; the detection algorithm includes three operations on the received stego-packet S_k to retrieve the covert plaintext P as follows:

1- Extracting the bits of the ciphertext $(C)_2$ from the IP ID field of each stego packet using the pre-agreed extraction algorithm and reference pointer key 'k'.

2- Converting each 8 bit ciphertext code $(C)_2$ to its decimal equivalent $(C)_{10}$. [This operation is optional depending on the used decryption algorithm].

3- Decrypting the ciphertext $(C)_{10}$ using the appropriate decryption algorithm and key 'd'.

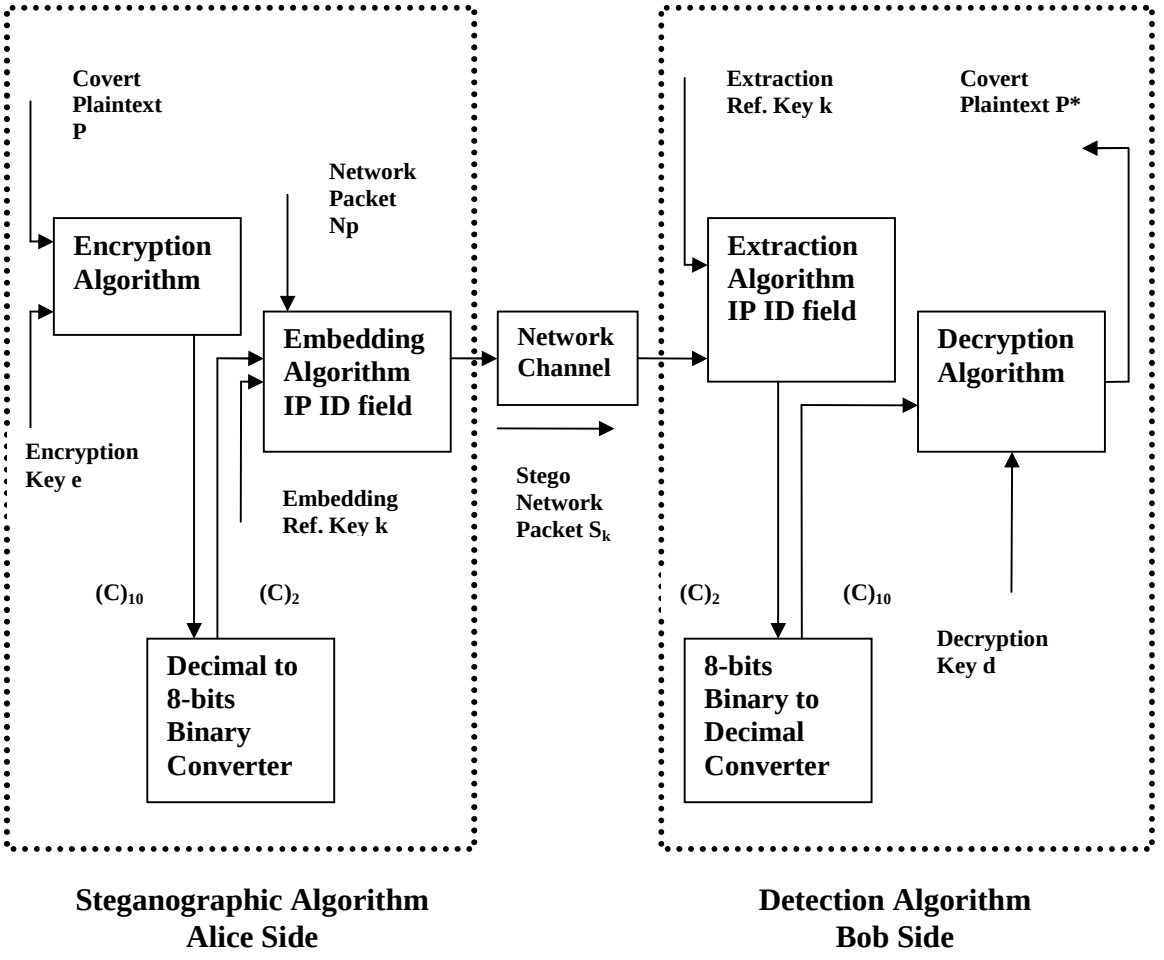


Figure (2): The framework of the M1 steganographic method for data hiding in the IP ID field

2.2 The Encryption/ Decryption Algorithms of the M1 Steganographic Method:

For the M1 steganographic method; many different encryption algorithms could be used regarding that the resulted ciphertext should not exceed 255 to be able to convert to the 8-bit binary representation. It is suggested to use the modular exponential cipher with modulus 257 [2] or the new encryption method which was suggested in [2] & [3].

2.3 The Embedding/ Extracting Algorithms of the M1 Steganographic Method:

Any suitable algorithm could be agreed by Alice and Bob. The following section describes the embedding/ extracting algorithms using pre-agreed reference key k and direction to embed/ extract the ciphertext bits in/ from the IP ID field.

2.3.1 The Embedding Algorithm:

For IPv4:

1- Alice would use the reference pointer key k to calculate the index "x" which is the number of the bit within the IP ID field at which the 8-bit ciphertext would start.

$$x \equiv k \pmod{16} + 1 \, , \, 1 \leq x \leq 16 \qquad (1)$$

2- Assume that the 16-bit of the IP ID field are denoted as $[i_{15},: :,i_1 ,i_0]$, then Alice would embed the 8 bits of the ciphertext code $[c_7, : :,c_1,c_0]$ in the IP ID field starting from the bit number x which is i_{x-1} with its seven followed bits, regarding that the embedding direction would be in the clock wise or the anti-clock wise direction according to her pre-agreement with Bob.

3- The remaining 8 bits of the IP ID field would be generated randomly.

For IPv6:

The same embedding concept as in IPv4 packet except the follows:

1- The IP ID field in the IPv6 is a 32-bit field. This requires replacing modulus 16 in the embedding and extracting algorithms by modulus 32.

$$x \equiv k \pmod{32} + 1 \, , \, 1 \leq x \leq 32 \qquad (2)$$

2- Embedding 8-bit hidden data in the 32-bit IP ID field involves generating the remaining 24 bits of the IP ID field randomly which provides more uniqueness for the resulted IP ID than that which is resulted in the IPv4.

The 32-bit IP ID in the IPv6 provides more available hidden size for the M1 method than that for the IPv4 as Alice could embed till 16 covert bits in it and generate the remaining bits randomly. This provides multiple permutations in which Alice and Bob could agree to choose how to embed the two 8-bit hidden characters as they could be separated or concatenated (as a followed 16-bit string). The availability of having different permutations may increase the hardness for a warden to determine the positions of the hidden bits. The 32-bit IP ID field consists of four sequenced 8-bit blocks which are ordered from the right to the left as the first, the second, the third and the last. Table 1 shows some available permutations for embedding two 8-bit characters in the 32-bit IP ID field as follows:

Case No. #	The position of one of the 8-bit hidden character in the IP ID	The position of the other 8-bit hidden character in the IP ID
1	First eight bits	Second eight bits
2	First eight bits	Third eight bits
3	First eight bits	Last eight bits
4	Second eight bits	Third eight bits
5	Second eight bits	Last eight bits
6	Third eight bits	Last eight bits
7	Embed the two followed 8-bit characters starting from bit number x of the IP ID field (where x is calculated using equation 2) in either the clockwise or the anti-clock direction according to the pre-agreement.	

Table (1): Some available permutations for embedding two 8-bit characters in the 32-bit IP ID field of the IPv6 fragment header extension

It's not recommended to embed the two hidden characters as a followed 16 bit string within the 32-bit IP ID field as in cases 1, 4, 6 and 7 of table 1, because some two characters words like (as, is, to, in, go,…,etc) may be repeated in the message and detected by a warden who knew the embedding algorithm and performed frequency analysis to characters. It's recommended to embed one 8-bit character only in the 32-bit IP ID field to save its uniqueness and decrease the frequency detection.

2.3.2 The Extraction Algorithm:

For IPv4:

1- Bob would use the reference pointer key k to calculate the index "x" from equation 1.

2- Bob would point to the bit number x in the IP ID field of the stego packet and select this bit to be the first bit in the ciphertext code.

3- Bob would count 7 bits after the first selected bit in the clock or the anti-clock wise direction according to his pre-agreement with Alice, then he would write them in their followed order after the first bit to get all the 8 bits of the ciphertext**.**

For IPv6:

The concept of the used extraction algorithm depends on the used embedding algorithm.

2.4 Example on Using the M1 Steganographic Method with the Modular Exponential Cipher of Modulus 257:

Assume that Alice and Bob used the M1 steganographic method to hide data in the IP ID field of IPv4 packets, and they agreed to use the modular exponential cipher with modulus 257 for encryption [2] by the key e=7, and they used the reference pointer key k= 26 for embedding the ciphertext character in the IP ID field in the clock wise direction. If Alice wants to hide the capital letter A, she needs to check its ASCII code that has a decimal equivalent = 65. So; the plaintext P = 65.

Alice would do the following at her side:

1- Calculate the equivalent ciphertext C using the modular exponential cipher of modulus 257: For P = 65 and e =7; C= 10.

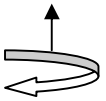
2- Convert the ciphertext from decimal to its 8-bit binary equivalent;
$C = (10)_{10} = (0000\ 1010)_2 = (c_7, : :, c_1, c_0)_2$

3- Calculate the index x using the embedding reference pointer key k as following:
$x \equiv k \pmod{16} + 1 \equiv 26 \pmod{16} + 1 \equiv 11 \pmod{16}$.

4- Point to the 11th bit within the IP ID field which is $i_{10,}$ then embed the 8 bits of $(C)_2$ in the clock wise direction starting from i_{10} and ending by i_1, and generate the remaining 8 bits of the IP ID field randomly as follows:

The IP ID field **[i_{15} i_{14} i_{13} i_{12} i_{11} i_{10} i_9 i_8 i_7 i_6 i_5 i_4 i_3 i_2 i_1 i_0]**

Would be



[c_5 c_4 c_3 c_2 c_1 c_0 R R R R R R R R c_7 c_6]

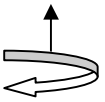
The final form of the IP ID in our example would appear as [001010RRRRRRRR00], where R is a random generated value.

Bob would do the following at his side:

1- Calculate the index x from the reference pointer key as;
$x \equiv k \pmod{16} + 1 \equiv 26 \pmod{16} + 1 \equiv 11 \pmod{16}$.

2- Point to the 11th bit at the IP ID field of the IPv4 header of the stego packet; which is here i_{10}, then count 7 bits after it in the clock wise direction and extract these 8 bits of the ciphertext C as;

[i_{15} i_{14} i_{13} i_{12} i_{11} i_{10} i_9 i_8 i_7 i_6 i_5 i_4 i_3 i_2 i_1 i_0]



So the extracted ciphertext code C would be;

$(i_1\ i_0\ i_{15}\ i_{14}\ i_{13}\ i_{12}\ i_{11}\ i_{10})_2 = (c_7\ c_6\ c_5\ c_4\ c_3\ c_2\ c_1\ c_0)_2$

Bob would find that $C= (0000\ 1010)_2$.

3- Convert the ciphertext from its 8-bit binary form to the decimal equivalent as;

$C = (0000\ 1010)_2 = (10)_{10}$

4- Use the modular exponential deciphering algorithm of modulus 257 to retrieve the plaintext using the deciphering key d = 183 for the corresponding encryption key e =7, when the ciphertext C = 10, then the retrieved plaintext P = 65.

2.5 Advantages of the M1 Steganographic Method:

1- The M1 steganographic method might be applied to hosts which communicate covertly across local area networks (LANs) or wide area networks (WANs) [like the Internet].

2- The embedded hidden data in the IP ID field isn't affected by the characteristics of the non-ideal network channel, because the IP ID field is an immutable field whose value doesn't change during the journey of the packet or its fragments to the destination.

3- Due to the manipulation of the IP ID field; the M1 data hiding method is resistant to packet filtering firewalls. Stateful inspection firewalls can't detect the M1 data hiding method due to the introduced randomness in half (or more) of the bits of the IP ID field. [Transparency to network filters will be briefly discussed in section 3.1].

4- The M1 method may be used with IPv4 or IPv6 packets. But with IPv6; fragmentation is required by the packet source.

5- The probability of having the same IP ID field for the similar covert data is low due to the random generated bits. For IPv4; it is 1/ 256. For IPv6 with one 8-bit hidden character; it is $1/ 2^{24}$. While with two 8-bit hidden characters; it is $1/ 2^{16}$.

6- The steganographic bandwidth of the M1 method is high. For one 8-bit hidden character in the IP ID field of IPv4 or IPv6 packets; it equals 8 [bits/ packet]. For two 8-bit hidden characters in the IP ID field of IPv6 packets; it equals 16 [bits/ packet].

7- The embedding scheme of the M1 method is strong due to using a reference key to point to the first embedded bit of the ciphertext. The attacker needs to know the reference pointer key in addition to the embedding direction (clock or anti-clockwise) in order to extract the ciphertext assuming that he knows that this scheme is used. For two 8-bit hidden characters in the 32-bit IP ID field of IPv6; many embedding permutations could be used to embed them, this may increase the hardness for a warden to determine the position of the hidden bits.

8- Many encryption algorithms could be used with the M1 steganographic method.

3. Evaluation of the M1 Steganographic Method for Data Hiding in the IP ID Field:

3.1 Transparency to Network Filters:

Static filter examines the packet based on the information in its header. Stateful inspection according to dynamic packet filtering; is a firewall architecture that works at the network layer. Unlike static packet filtering; stateful inspection has the following functions:
A. Tracking connections which traverse the firewall interfaces to check their validation.
B. Examining the content of the packet up through the application layer in order to determine more about its source and destination.
C. Monitoring the connection state and compiling its information in a state table.

The stego network packet S_k would appear transparent to network filters based on the selection of the specified fields of the IP header which would carry the hidden data, the employed steganographic methods and the hidden data embedding/ extracting schemes.

In the framework of figure 1; the covert communication cannot be detected because the network packet N_P and the steganogram packet S_k seem similar for the network administrator, warden and any intermediate nodes. The non ideal channel possesses the properties which were explained in section 1. Due to immutability, uniqueness and randomness nature of the IP ID field; the M1 steganographic method is resistant to packet filtering firewalls. Stateful inspection firewalls also can't detect M1 due to the introduced randomness in half or more of the bits of the IP ID field.

3.2 General Covert Channel Parameters:

3.2.1 Steganographic Bandwidth:

The steganographic bandwidth could be expressed by the meaning of RBR (Raw Bit Rate) which is the total number of the steganogram bits which are transmitted during one time unit [bit/s], or equivalently by PRBR (Packet Raw Bit Rate) which is the total number of steganogram bits which are transmitted within a single packet during the hidden communication process [bit/packet] [4].

When the M1 method aims to hide 8 bits in the IP ID of the IPv4 header of a network packet, its steganographic bandwidth (PRBR) = 8 [bits/packet]. If M1 with IPv6 is used to hide 8 bits in the IP ID, then the PRBR = 8 [bits/ packet], while if it used to hide 16 bits, then the PRBR = 16 [bits/ packet].

3.2.2 Covert Channel Capacity Estimation:

The steganographic capacity is one of the most significant performance measurements of covert channels. Its estimation relates the cost of data hiding in terms of the time taken to process the stego-network packet and the protocol header overheads with the total time to transmit the stego-network packet from the source. The capacity of the covert channel at the network layer could be a function of the quantity of the transmitted hidden information, the time required to execute the used data hiding scenario and the effect on the system performance (per execution of the used data hiding scenario). The M1 method is as Ashan method [1] in using IP packets to embed hidden bits in their IP ID but by different ways. So for IPv4; we could refer the covert channel capacity of the M1 method to Ashan's considerations for his data hiding method in the IP ID field.

Ashan considered that the covert channel capacity depends on the total time T taken to transmit a stego network packet S_k from the network layer. The time of transmitting one data block T (in seconds) could be a function of the following elements:

1- The time used by the software (in seconds). This time doesn't dependent on the size of the data block.
2- The size of the transmitted data block or packet (in bytes).
3- The size of the network protocol overhead (in bytes).
4- The network speed (in bits per second).

While T increases with the increase of the first three elements, it decreases with the increase of the network speed. The covert channel capacity decreases with the increase of the packet size. If more than one bit is embedded, the capacity of the data hiding scheme can be expressed in (bits per second) as;

$C = B_{DH} / T$, where B_{DH} is the number of data hiding bits per packet.

3.2.3 Covert Channel Detection:

With respect to nodes other than the intended recipient of the covert communication; detection of the covert channel increases if the capacity of the covert channel increases. But if the used encoding/ decoding schemes and the embedding/ extracting algorithms are strong, they make the hidden information more imperceptible. Therefore; the M1 steganographic method provides an excellent way of having secret communications. The probability of covert channel detection could be controlled by decreasing its “usage frequency” especially if it has a high data hiding capacity.

While the overt channel exhibits similar behavior for both normal packets and stego packets which used the M1 steganographic method, then detecting the stego packets of the M1 method is extremely hard.

3.3 Hidden Communication Scenarios:

The steganogram sender (SS) must be the source of sending the packets and the one who apply the encryption and embedding algorithms in the M1 steganographic method. For IPv6 packets; to apply a steganographic method that hides data in the IP ID field, fragmentation is required by the packet sender. Noting that; IPv6 packet fragmentation/ reassembly occurs only by the packet sender/ receiver respectively. For IPv4 packets; packet fragmentation is not a mandatory to apply the M1 steganographic method, so the steganogram receiver (SR) could be the packet receiver or any intermediate node.

4. Using the M1 Steganographic Method with IPsec:

IPsec transport mode security associations have been defined to not carry IPv4 or IPv6 fragments [5]. So, The M1 method can't be used with IPsec transport mode for IPv6 packets because fragmentation is required.

If Alice and Bob didn't implement IPsec or they implemented IPsec in transport mode between them through the network channel which communicates them, then the IP ID of the IPv4 header packets would be clear for checking by any man in the middle.

To provide more confidentiality to the hidden data in the IP ID field by the M1 steganographic method; we suggest for SS and SR to implement IPsec encryption in the tunnel mode through the network channel which communicates their gateways to encrypt the IP ID field by IPsec to counter the traffic analysis.

5. Conclusions:

This paper suggests a new steganographic method M1 for data hiding in the IP ID field. In the M1 steganographic method; an appropriate algorithm and key are used to encrypt the plaintext character, then the resulted ciphertext is converted to its 8-bit binary representation and embedded in the IP ID field using pre-agreed reference pointer key and embedding direction. In the M1 steganographic method; the steganogram sender must be the packet source and it should maintain the randomness and the uniqueness natures of the IP ID field for the specified source/ destination pair and protocol combination for the time in which the IP packet (or any fragment of it) could be alive in the network.

For IPv6 packets; to apply a steganographic method that hides data in the IP ID field, fragmentation is required by the packet source. While fragmentation/ reassembly of IPv6 packets occurs only by the packet sender/ receiver respectively, then SS must be the packet source and SR must be the packet receiver to apply the M1 steganographic method with IPv6 packets.

Due to the specifications of the IP ID field, the M1 steganographic method is resistant to packet filtering firewalls. Stateful inspection firewalls also can't detect the M1 method due to the introduced randomness in the remaining bits of the IP ID field. The M1 method could be used with IPv4 to hide 8 bits per packet, or with IPv6 to hide till 16 bits per packet. The covert message in the IP ID field isn't affected by the non-ideal network channel characteristics because the overt channel exhibits similar behavior for both normal and stego packets that use the M1 steganographic method. Thus, the detection of the M1 stego-packet is extremely hard.

To provide more confidentiality to the hidden data in the IP ID field by M1; IPsec tunnel encryption could be implemented within the network channel which communicates the SS and the SR gateways to encrypt the IP ID by the IPsec to make the hidden data unclear for man in the middle traffic warden.

References:

[1] Kamran Ahsan, "Covert Channel Analysis and Data Hiding in TCP/IP", Master of Science, Department of Electrical and Computer Engineering, University of Toronto 2002. http://www.springerlink.com/content/20xn5j76r25002t5/
[2] Manal A. Shehab, "New Encryption and Steganographic Methods for Data Hiding in the IP Packets or Their Fragments", Master of Science in Electrical Engineering, Electrical Engineering Department, Faculty of Engineering, Alexandria University, Egypt, October 2011.
[3] Manal A. Shehab, Noha O. Korany, "New Encryption Method Based on Using The Kharaghani Array of Order 8", Submitted to: The 8th International Conference On Electrical Engineering ICEENG-8, May 2012.
[4] Murdoch S.J., Lewis S., "Embedding Covert Channels into TCP/IP", The 7th Information Hiding Workshop, pp.247-26, June 2005.
http://www.cl.cam.ac.uk/~sjm217/papers/ih05coverttcp.pdf
[5] S. Kent & K. Seo," Security Architecture for the Internet Protocol ", IETF RFC 4301 December 2005. http://tools.ietf.org/pdf/rfc4301.pdf

Last access to the web sides was in14 March 2012.