Received: 30 July 2018 | Revised: 11 June 2019 | Accepted: 28 August 2019

DOI: 10.4218/etrij.2018-0410

ORIGINAL ARTICLE

ETRI Journal WILEY

MKIPS: MKI-based protocol steganography method in SRTP

Amir Mahmoud Alishavandi | Mohammad Fakhredanesh

Faculty of Electrical and Computer Engineering, Malek Ashtar University, Tehran, Iran

Correspondence
Mohammad Fakhredanesh, Faculty of Electrical and Computer Engineering, Malek Ashtar University, Tehran, Iran.
Email: m-fakhredanesh@aut.ac.ir

This paper presents master key identifier based protocol steganography (MKIPS), a new approach toward creating a covert channel within the Secure Real-time Transfer Protocol, also known as SRTP. This can be achieved using the ability of the sender of Voice-over-Internet Protocol packets to select a master key from a pre-shared list of available cryptographic keys. This list is handed to the SRTP sender and receiver by an external key management protocol during session initiation. In this work, by intelligent utilization of the master key identifier field in the SRTP packet creation process, a covert channel is created. The proposed covert channel can reach a relatively high transfer rate, and its capacity may vary based on the underlying SRTP channel properties. In comparison to existing data embedding methods in SRTP, MKIPS can convey a secret message without adding to the traffic overhead of the channel and packet loss in the destination. Additionally, the proposed covert channel is as robust as its underlying user datagram protocol channel.

KEYWORDS

Covert channel, master key identifier, protocol steganography, Secure Real-time Transfer Protocol, Voice-over-Internet Protocol

1 | INTRODUCTION

Steganography is the science of hiding secret data in seemingly legitimate mediums. Scientists in this field are constantly trying to discover new methods and media to embed secret data or create covert channels. Since the growth of communication networks changes the life of humankind each day, the importance of communication technologies such as Voice-over-Internet Protocol (VoIP) becomes more significant. The constant traffic volume increase, real-time properties, and existing misplaced trust make VoIP technology a notable target for steganographic studies.

There have been numerous studies regarding steganography in VoIP networks, which can be classified into two main categories. The first category consists of studies that aim to use voice and audio codecs along with compression as carriers of the hidden data [1]. The second category consists of studies that use VoIP protocol structures to create covert channels [2]. A comprehensive study on the categorization of steganographic channels was carried out by Mazurczyk [3]. The main focus of the present paper is on techniques that utilize VoIP protocol structures to convey secret messages.

Generally, a VoIP call consists of two phases: the signaling phase and the communication phase. Each of these phases has its own protocols and properties. The dominant protocol in the signaling phase is Session Initiation Protocol, whereas it is Real-time Transport Protocol (RTP) in the communication phase. Despite the short duration and lack of capacity in comparison with the communication phase, there have been numerous efforts toward steganography in the VoIP signaling phase. Focused studies have been conducted on signaling-based steganography [4–6]. This family of steganographic techniques is beyond the scope of this research. The present paper mainly considers steganography in the secured version of RTP, also known as Secure RTP (SRTP).

In what follows, first, the terminology of steganography and SRTP technologies is reviewed. Then, a brief review of currently available methods of real-time steganography in VoIP streams is presented. After that, the present paper concerns the introduction of master key identifier based protocol steganography (MKIPS), a new approach toward creating covert communication within a secure real-time transport protocol. Next, the experimental results of a proof of concept implementation for this method, along with a comparison between the only available SRTP steganography method and MKIPS, have been presented. Finally, the paper is concluded by a discussion of the proposed method along with its strengths and weaknesses.

2 | TERMINOLOGY

2.1 | Steganography terminology

Steganography is the art and science of hiding secret information in a legitimate medium in a way such that the presence of this information is not detectable to an unaware or unsuspicious warden [7]. A warden is a person or an entity responsible for detecting the presence of steganographic activities in the network. His/her abilities and accesses are determined based on practical environments. The steganogram is the result of embedding the secret data in the cover data. Steganalysis is the science of detecting the presence of steganographic activities in a medium. VoIP steganography is the science of transferring hidden information across a VoIP network. VoIP steganography methods utilize different aspects of the VoIP network for steganography, namely encoded voice with different codecs and various protocols contents and behaviors. Protocol steganography is the science of utilizing different properties of a protocol to hide information or create a covert channel.

A covert channel is an entity developed by the unexpected, unforeseen, or unconventional use of channels to transfer information across a legitimate network. The properties of a covert channel can be measured by its transfer rate (bits per second; bps), error propagation (bits), and overhead (percent). These measures are typically used to compare two different methods of creating a covert channel. Applications of a covert channel can include but are not limited to the accomplishment of increased quality of service [8], security [9,10], and capacity [11] in networks on one hand, or to create information leakage as an attacking mechanism for unauthorized data exfiltration on the other hand [12]. The amount of information leakage to be considered as a threat depends on the network policy. According to a United States Department of Defense standard [13], more than 1 bps of information leakage in a secure network must be considered as a threat, whereas in normal networks, it is 100 bps.

It should be mentioned that in the creation of a protocol-based covert channel, the sender of the steganogram is free to change the sending mechanism, but he/she is not allowed to destroy the functionality of the protocol. The goal is to make an unaware warden unable to differentiate between a normal channel and a covert channel.

2.2 | SRTP terminology

Voice-over-Internet Protocol is a technology that was initially designed with the intention of transferring voice over the Internet. Later, other applications were introduced, including but not limited to video streaming, online gaming, and video/audio conference calls. As the usage, traffic, and coverage of this technology grew, the need for security in VoIP streams was more strongly felt. One solution that has been proposed to address this need is SRTP.

Secure Real-time Transfer Protocol, as an application layer security protocol, tries to secure the RTP streams by employing encryption, authentication tags, and replay lists to provide confidentiality, authentication, and replay protection [14]. To perform this job, SRTP relies on master keys and salt values that are negotiated or distributed through an external key management protocol. A guide on key management protocols to secure SRTP is available in RFC 2701 [15]. Session keys are then derived from the negotiated master keys so that the encryption/decryption procedures can be performed with them. Similar to other encryption systems, to decrease the size of the analyzable ciphertext for a potential attacker or to limit unauthorized access to the encrypted content, SRTP can periodically refresh its master keys. This is especially effective in multicast or conference communications when the number of users sharing the same master keys and consequently the amount of available and analyzable ciphertext for the attacker increases. As stated in RFC 3711 [14], one approach to decrease the ciphertext available for an attacker is to employ the master key identifier (MKI) key refreshment schedule. Figure 1 shows the structure of an SRTP frame with its MKI field at the end.

In this approach, both sides of the communication are equipped with a list of main keys and their indicators. As stated in RFC 4568 [16]: “One or more master key(s) with its/their associated MKI can be initially defined, and then later updated or deleted and new ones get defined.” All of these keys and their indicators have the same length during a session. At the end of each packet, a field containing the

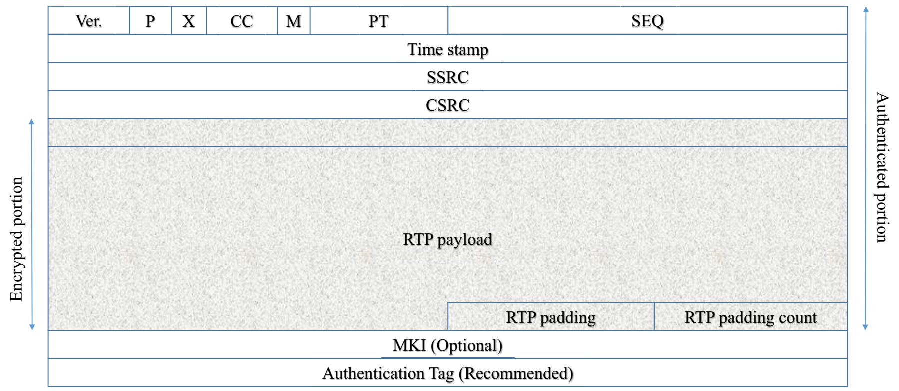


FIGURE 1 Secure Real-time Transfer Protocol packet with its master key identifier field

indicator of the master key is attached by processing the packet that is encrypted.

3 | RELATED WORK

From a general perspective, steganography in VoIP traffic or VoIP covert channels can be divided into three major categories [2]. Protocol storage covert channels, protocol behavioral covert channels, and hybrid covert channels.

Protocol headers and payload fields have always been a major candidate for data embedding operations. Since scientists recognized that in a practical conversation, not all of the header fields in a packet change, they tried to manipulate these fields using their proposed steganographic algorithms to create covert channels. As an example of storage-based covert channels in research conducted by Forbes [17], timestamp fields of RTP packets are used to embed secret data. As another example, Bai and others [18] suggested the use of jitter fields in Real-time Transport Control Protocol (RTCP) reports to create a covert channel. Compared with two other categories, storage-based methods covert channels are relatively easier to implement.

Behavioral protocol steganography techniques include methods that utilize protocol behavioral properties such as packet delay, jitter tolerance and buffers, packet loss concealment, packet order, and packet time relations for steganography. As an example of the behavioral protocol covert channel creation method, a study conducted by Huang and others [19] can be mentioned in which the relative behavior of RTP and RTCP packets was used to convey secret information. Some other examples of these types of methods can be found in two articles written by Shah and others [20,21], and two studies that were carried out by Chen and others [22,23].

Hybrid methods are procedures that attempt to simulate intentional protocol misbehavior by manipulating protocol header fields. Examples of these methods can be found in research conducted by Schmidt and others [24] or research conducted by Hamdaqa and Tahvildari [25].

From a different perspective, different protocol steganography methods can be jointly used to increase the bandwidth of a covert channel. Vertical or multi-layered protocol steganography methods try to use steganographic techniques in multiple layers of the protocol stack to embed the secret data. For example, IP headers from the network layer, TCP header from the transport layer [26], and the RTP header can all be simultaneously used to embed different parts of the secret data [2]. Horizontal or single-layer protocol steganography methods, on the other hand, attempt to use different steganographic methods to embed data in a single network layer. In these methods, protocol headers can be individually manipulated, and the relative behavior of data units can also be used as a steganographic medium. The study carried out by Huang and others is an example of this type of covert channel creation that exploits the relative behavior of RTP and RTCP packets to transfer data [19]. Other methods of VoIP steganography that are not based on protocol properties are considered out of the scope of this paper.

To the author's current knowledge, there is only one major work on steganography in SRTP streams using its security features [2]. The mentioned effort tries to embed the secret data in the authentication tag of each SRTP packet. As clearly stated in RFC 3711 [14], any failure in the authentication process of the packets should be logged and the packet must be discarded. According to this, methods that use an authentication tag as a cover medium not only can discard the altered packet but also generate a log that would be noticeable to an unaware and unsuspicious warden. If the receiving end is in the first scenario, that is, it is under full control of the VoIP packet sender machine, he/she might try to remove the generated log or process the unauthenticated packets, which are both against the protocol standards and would greatly degrade the security of the whole system. On the other hand, discarding the packet because of its authentication failure at the destination substantially increases network inefficiency. As stated in RFC 3711 [14], the length of an authentication tag must be 32 bits to 80 bits, so to transfer 1 kbits of secret data, at least 13 frames should be discarded.

4 | SCENARIOS AND COMMUNICATION MODEL

Based on the access level of the agent trying to create a covert channel, four variations of covert channels in a VoIP stream are possible. Understanding the differences between these scenarios can be helpful in practical implementations.

In the first scenario, the steganographer has full control over both endpoints of the communication. To create a covert channel, he/she can modify the packet crafting entity, as long as he/she does not disrupt the normal functionality of the protocols. In the second and third scenarios, the steganographer is only in full control of one of the VoIP call endpoints and has no control over the other. This can either be the VoIP packet crafting entity (the second scenario) or the packet receiving entity (the third scenario). The fourth scenario occurs when the steganographer has no control at all over the VoIP endpoints; thus, he/she can only act as a "man in the middle."

Most VoIP steganographic techniques use the first scenario in which the sender and receiver of a steganogram are in full control of their VoIP endpoints. Figure 2 illustrates these different scenarios.

Because the security features of SRTP limit the network steganographer in many ways, nearly all of the methods mentioned above used RTP as the application layer protocol. However, the fact is that because of the transparency of the SRTP header, many of these steganographic methods that are suggested for RTP are also applicable to SRTP. Additionally, the security features of SRTP can be intelligently utilized to embed secret data.

The only technique for steganography using SRTP security features was proposed by Mazurczyk and Szczypiorski [2]. In their effort, they used SRTP authentication tags to transfer secret data. As mentioned earlier, packets that are received and contain secret data in their authentication tag cannot be authenticated, so they must be discarded in the receiver agent and a security incident log created.

5 | PROPOSED METHOD

As stated in RFC 3711 [14], in the MKI key refreshment schedule, a list of master keys along with their indicators should be made available to each client by an external key management protocol. After encryption and adding authentication tags to the packets in the sending agent, a field named MKI should be added to each packet, indicating the master key that was used to secure the SRTP packet. In the receiving agent, based on the MKI field, the correct master key is selected from the pre-shared list of master keys. The selected master key is then used to feed the session key derivation mechanism. According to RFC 3711 [14], the length of the MKI fields can vary from one to 128 bytes. However, the length of this field must remain constant during each SRTP session. In a steganographic approach, the sender agent can intentionally select master key indicators in such a way that the master key selection conveys secret bits of data, that is, mapping between secret data code words and his/her available MKI values. Then, the resulting covert channel data rate can be calculated as follows:

$$R_C = \mathrm{PR}_{\mathrm{SRTP}} \times \left\lfloor \log_2 L \right\rfloor, \tag{1}$$

where R_c is the data rate of the covert channel (bps), $\mathrm{PR}_{\mathrm{SRTP}}$ is the packet rate of the SRTP protocol in the network (packets per second), and L is the length of the list of the master keys (a non-negative integer). Figure 3 presents a general diagram that demonstrates how the MKI-based steganography works.

As an example, in the case where a list of two master keys is made available to both clients, each time the sending agent wants to embed a "0" bit, it selects the first master key to secure the packet, and each time it wants to embed a "1" bit, it selects the second master key.

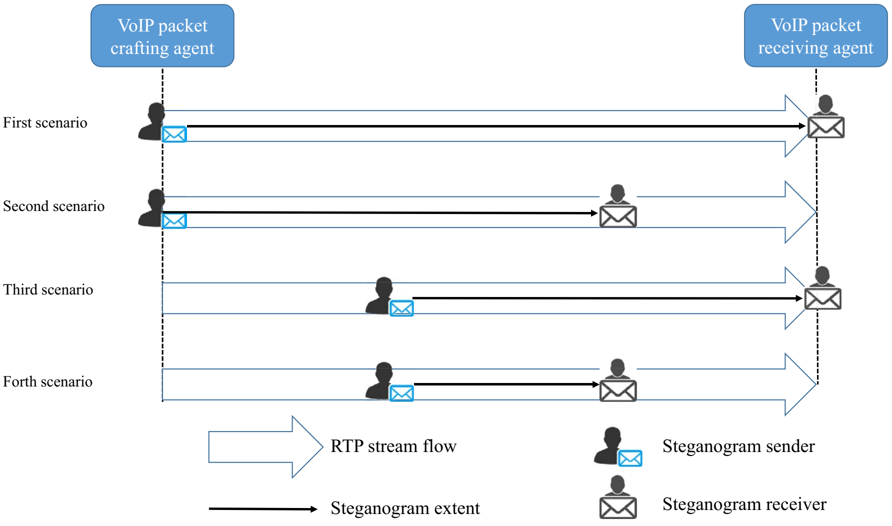


FIGURE 2 Voice-over-Internet Protocol covert channel scenarios

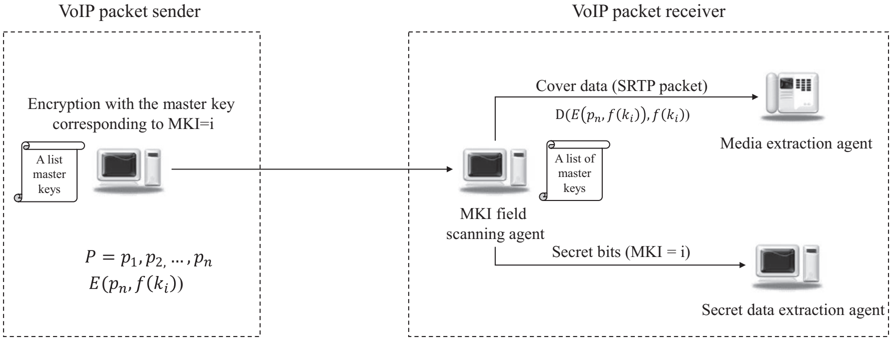


FIGURE 3 Steganography using master key identifier

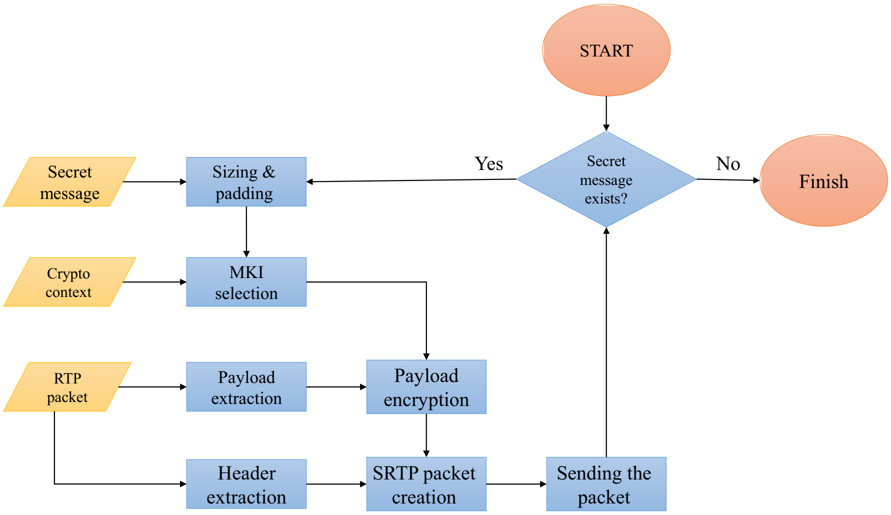
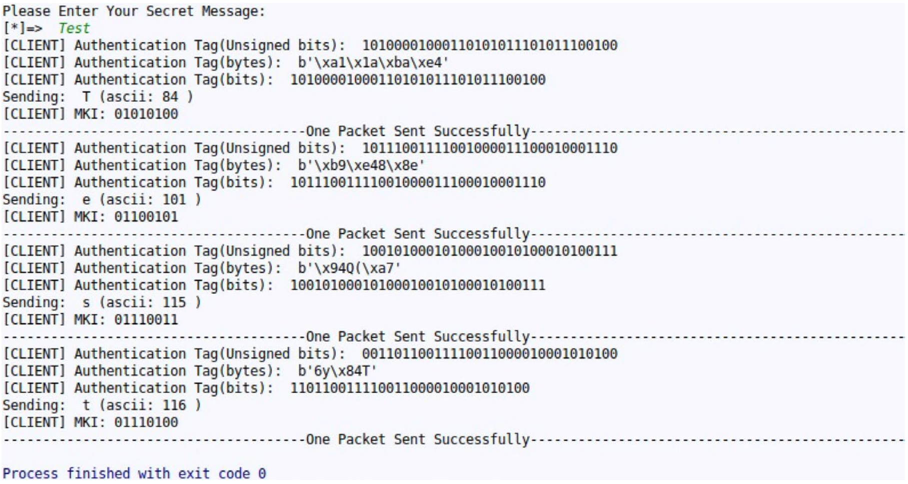


FIGURE 4 Master key identifier based protocol steganography steganographer agent flowchart

The capacity provided by this covert channel is based on the length of the MKI field or the number of master keys that are available to both sides of a conversation.

Now, imagine a more practical case in which the MKI field length is eight bits or the length of the master keys list is 256 = 28. Based on the formula, given the packet rate of 50 packets per second, the transfer rate of the created covert channel is 400 bps.

To implement this technique, the steganogram-sending agent should consider the secret data bit(s), and then select the MKI value and the master key accordingly. On the receiving end, a scanning agent should be implemented to scan the MKI values of the delivered packets prior to any processing. Then, based on these scanned values, the embedded secret message can be extracted. It is notable that not only does this technique



```
Please Enter Your Secret Message:
[*]=>  Test
[CLIENT] Authentication Tag(Unsigned bits):  10100001000110101011101011100100
[CLIENT] Authentication Tag(bytes):  b'\xa1\x1a\xba\xe4'
[CLIENT] Authentication Tag(bits):  10100001000110101011101011100100
Sending:  T (ascii: 84 )
[CLIENT] MKI: 01010100
--------------------------------------One Packet Sent Successfully-----------------------------------------------
[CLIENT] Authentication Tag(Unsigned bits):  10111001111001000011100010001110
[CLIENT] Authentication Tag(bytes):  b'\xb9\xe48\x8e'
[CLIENT] Authentication Tag(bits):  10111001111001000011100010001110
Sending:  e (ascii: 101 )
[CLIENT] MKI: 01100101
--------------------------------------One Packet Sent Successfully-----------------------------------------------
[CLIENT] Authentication Tag(Unsigned bits):  10010100010100010010100010100111
[CLIENT] Authentication Tag(bytes):  b'\x94Q(\xa7'
[CLIENT] Authentication Tag(bits):  10010100010100010010100010100111
Sending:  s (ascii: 115 )
[CLIENT] MKI: 01110011
--------------------------------------One Packet Sent Successfully-----------------------------------------------
[CLIENT] Authentication Tag(Unsigned bits):  00110110011110011000010001010100
[CLIENT] Authentication Tag(bytes):  b'6y\x84T'
[CLIENT] Authentication Tag(bits):  110110011110011000010001010100
Sending:  t (ascii: 116 )
[CLIENT] MKI: 01110100
--------------------------------------One Packet Sent Successfully-----------------------------------------------

Process finished with exit code 0
```

FIGURE 5 Master key identifier based protocol steganography steganographer agent sending a “Test” message in four packets

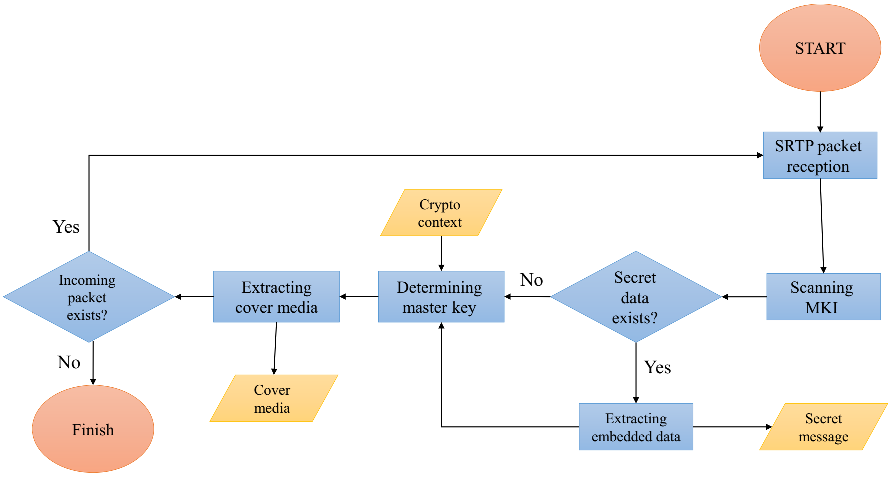
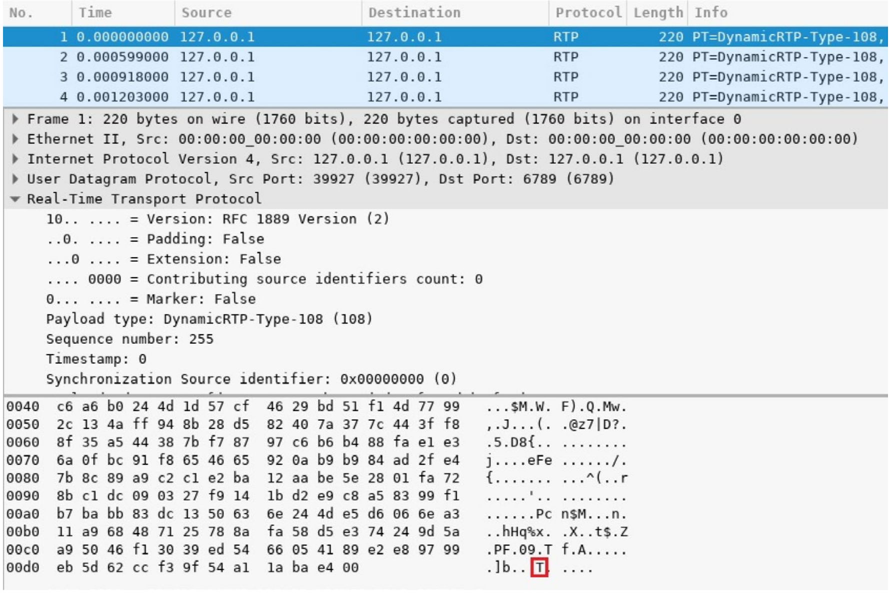
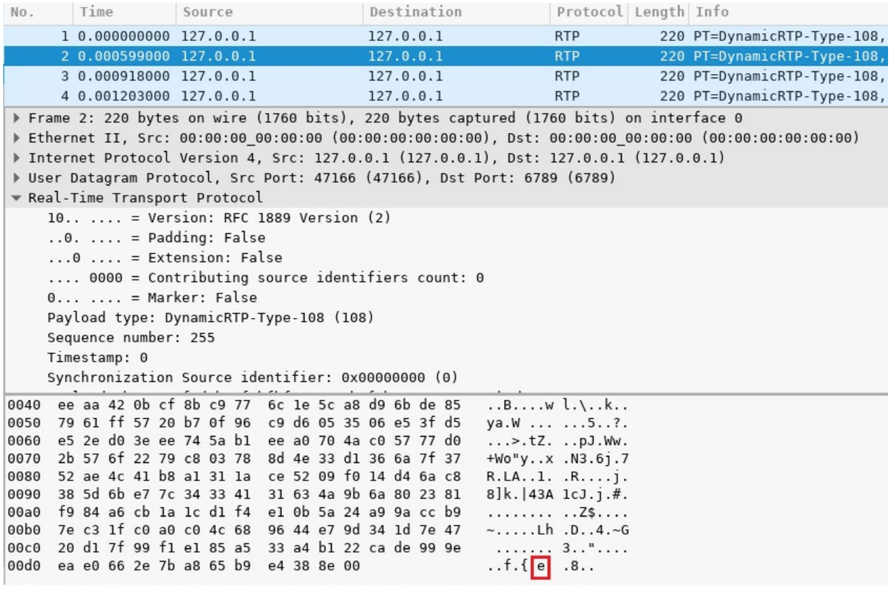


FIGURE 6 Master key identifier based protocol steganography receiving agent flowchart



```
No.   Time         Source          Destination     Protocol Length Info
    1 0.000000000  127.0.0.1       127.0.0.1       RTP       220 PT=DynamicRTP-Type-108,
    2 0.000599000  127.0.0.1       127.0.0.1       RTP       220 PT=DynamicRTP-Type-108,
    3 0.000918000  127.0.0.1       127.0.0.1       RTP       220 PT=DynamicRTP-Type-108,
    4 0.001203000  127.0.0.1       127.0.0.1       RTP       220 PT=DynamicRTP-Type-108,
▸ Frame 1: 220 bytes on wire (1760 bits), 220 bytes captured (1760 bits) on interface 0
▸ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
▸ Internet Protocol Version 4, Src: 127.0.0.1 (127.0.0.1), Dst: 127.0.0.1 (127.0.0.1)
▸ User Datagram Protocol, Src Port: 39927 (39927), Dst Port: 6789 (6789)
▾ Real-Time Transport Protocol
    10.. .... = Version: RFC 1889 Version (2)
    ..0. .... = Padding: False
    ...0 .... = Extension: False
    .... 0000 = Contributing source identifiers count: 0
    0... .... = Marker: False
    Payload type: DynamicRTP-Type-108 (108)
    Sequence number: 255
    Timestamp: 0
    Synchronization Source identifier: 0x00000000 (0)
0040  c6 a6 b0 24 4d 1d 57 cf  46 29 bd 51 f1 4d 77 99   ...$M.W. F).Q.Mw.
0050  2c 13 4a ff 94 8b 28 d5  82 40 7a 37 7c 44 3f f8   ,.J...(. .@z7|D?.
0060  8f 35 a5 44 38 7b f7 87  97 c6 b6 b4 88 fa e1 e3   .5.D8{.. ........
0070  6a 0f bc 91 f8 65 46 65  92 0a b9 b9 84 ad 2f e4   j....eFe ....../.
0080  7b 8c 89 a9 c2 c1 e2 ba  12 aa be 5e 28 01 fa 72   {....... ...^(..r
0090  8b c1 dc 09 03 27 f9 14  1b d2 e9 c8 a5 83 99 f1   .....'.. ........
00a0  b7 ba bb 83 dc 13 50 63  6e 24 4d e5 d6 06 6e a3   ......Pc n$M...n.
00b0  11 a9 68 48 71 25 78 8a  fa 58 d5 e3 74 24 9d 5a   ..hHq%x. .X..t$.Z
00c0  a9 50 46 f1 30 39 ed 54  66 05 41 89 e2 e8 97 99   .PF.09.T f.A.....
00d0  eb 5d 62 cc f3 9f 54 a1  1a ba e4 00               .]b..T. ....
```

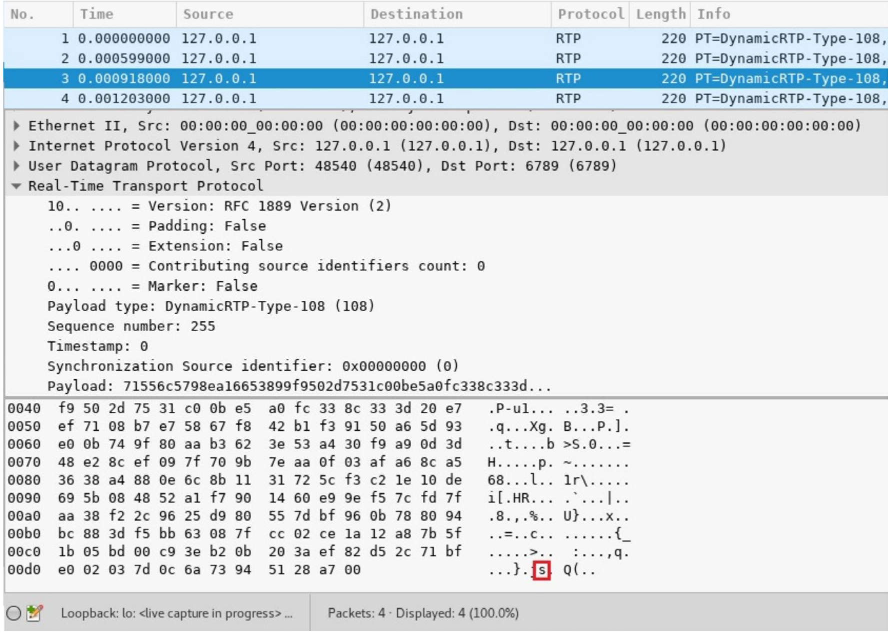
FIGURE 7 First packet with the letter “T” embedded in its master key identifier field



```
No.   Time         Source          Destination     Protocol Length Info
    1 0.000000000  127.0.0.1       127.0.0.1       RTP       220 PT=DynamicRTP-Type-108,
    2 0.000599000  127.0.0.1       127.0.0.1       RTP       220 PT=DynamicRTP-Type-108,
    3 0.000918000  127.0.0.1       127.0.0.1       RTP       220 PT=DynamicRTP-Type-108,
    4 0.001203000  127.0.0.1       127.0.0.1       RTP       220 PT=DynamicRTP-Type-108,
▸ Frame 2: 220 bytes on wire (1760 bits), 220 bytes captured (1760 bits) on interface 0
▸ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
▸ Internet Protocol Version 4, Src: 127.0.0.1 (127.0.0.1), Dst: 127.0.0.1 (127.0.0.1)
▸ User Datagram Protocol, Src Port: 47166 (47166), Dst Port: 6789 (6789)
▾ Real-Time Transport Protocol
    10.. .... = Version: RFC 1889 Version (2)
    ..0. .... = Padding: False
    ...0 .... = Extension: False
    .... 0000 = Contributing source identifiers count: 0
    0... .... = Marker: False
    Payload type: DynamicRTP-Type-108 (108)
    Sequence number: 255
    Timestamp: 0
    Synchronization Source identifier: 0x00000000 (0)
0040  ee aa 42 0b cf 8b c9 77  6c 1e 5c a8 d9 6b de 85   ..B....w l.\..k..
0050  79 61 ff 57 20 b7 0f 96  c9 d6 05 35 06 e5 3f d5   ya.W ... ...5..?.
0060  e5 2e d0 3e ee 74 5a b1  ee a0 70 4a c0 57 77 d0   ...>.tZ. ..pJ.Ww.
0070  2b 57 6f 22 79 c8 03 78  8d 4e 33 d1 36 6a 7f 37   +Wo"y..x .N3.6j.7
0080  52 ae 4c 41 b8 a1 31 1a  ce 52 09 f0 14 d4 6a c8   R.LA..1. .R....j.
0090  38 5d 6b e7 7c 34 33 41  31 63 4a 9b 6a 80 23 81   8]k.|43A 1cJ.j.#.
00a0  f9 84 a6 cb 1a 1c d1 f4  e1 0b 5a 24 a9 9a cc b9   ........ ..Z$....
00b0  7e c3 1f c0 a0 c0 4c 68  96 44 e7 9d 34 1d 7e 47   ~.....Lh .D..4.~G
00c0  20 d1 7f 99 f1 e1 85 a5  33 a4 b1 22 ca de 99 9e    ....... 3..".....
00d0  ea e0 66 2e 7b a8 65 b9  e4 38 8e 00               ..f.{e .8..
```

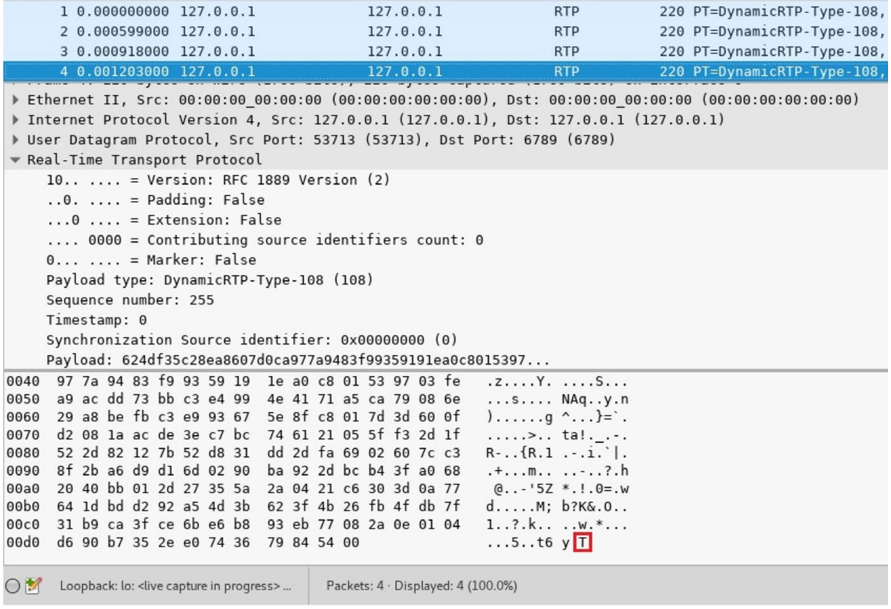
FIGURE 8 Second packet with the letter “e” embedded in its master key identifier field

FIGURE 9 Third packet with the letter "s" embedded in its master key identifier field



```
No.   Time         Source            Destination       Protocol Length Info
    1 0.000000000  127.0.0.1         127.0.0.1         RTP         220 PT=DynamicRTP-Type-108,
    2 0.000599000  127.0.0.1         127.0.0.1         RTP         220 PT=DynamicRTP-Type-108,
    3 0.000918000  127.0.0.1         127.0.0.1         RTP         220 PT=DynamicRTP-Type-108,
    4 0.001203000  127.0.0.1         127.0.0.1         RTP         220 PT=DynamicRTP-Type-108,
▸ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
▸ Internet Protocol Version 4, Src: 127.0.0.1 (127.0.0.1), Dst: 127.0.0.1 (127.0.0.1)
▸ User Datagram Protocol, Src Port: 48540 (48540), Dst Port: 6789 (6789)
▾ Real-Time Transport Protocol
    10.. .... = Version: RFC 1889 Version (2)
    ..0. .... = Padding: False
    ...0 .... = Extension: False
    .... 0000 = Contributing source identifiers count: 0
    0... .... = Marker: False
    Payload type: DynamicRTP-Type-108 (108)
    Sequence number: 255
    Timestamp: 0
    Synchronization Source identifier: 0x00000000 (0)
    Payload: 71556c5798ea16653899f9502d7531c00be5a0fc338c333d...
0040  f9 50 2d 75 31 c0 0b e5  a0 fc 33 8c 33 3d 20 e7   .P-u1... ..3.3= .
0050  ef 71 08 b7 e7 58 67 f8  42 b1 f3 91 50 a6 5d 93   .q...Xg. B...P.].
0060  e0 0b 74 9f 80 aa b3 62  3e 53 a4 30 f9 a9 0d 3d   ..t....b >S.0...=
0070  48 e2 8c ef 09 7f 70 9b  7e aa 0f 03 af a6 8c a5   H.....p. ~.......
0080  36 38 a4 88 0e 6c 8b 11  31 72 5c f3 c2 1e 10 de   68...l.. 1r\.....
0090  69 5b 08 48 52 a1 f7 90  14 60 e9 9e f5 7c fd 7f   i[.HR... .`...|..
00a0  aa 38 f2 2c 96 25 d9 80  55 7d bf 96 0b 78 80 94   .8.,.%.. U}...x..
00b0  bc 88 3d f5 bb 63 08 7f  cc 02 ce 1a 12 a8 7b 5f   ..=..c.. ......{_
00c0  1b 05 bd 00 c9 3e b2 0b  20 3a ef 82 d5 2c 71 bf   .....>..  :...,q.
00d0  e0 02 03 7d 0c 6a 73 94  51 28 a7 00               ...}.s. Q(..
○ Loopback: lo: <live capture in progress> ...   Packets: 4 · Displayed: 4 (100.0%)
```

FIGURE 10 Fourth packet with the letter "t" embedded in its master key identifier field



```
    1 0.000000000  127.0.0.1         127.0.0.1         RTP         220 PT=DynamicRTP-Type-108,
    2 0.000599000  127.0.0.1         127.0.0.1         RTP         220 PT=DynamicRTP-Type-108,
    3 0.000918000  127.0.0.1         127.0.0.1         RTP         220 PT=DynamicRTP-Type-108,
    4 0.001203000  127.0.0.1         127.0.0.1         RTP         220 PT=DynamicRTP-Type-108,
▸ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
▸ Internet Protocol Version 4, Src: 127.0.0.1 (127.0.0.1), Dst: 127.0.0.1 (127.0.0.1)
▸ User Datagram Protocol, Src Port: 53713 (53713), Dst Port: 6789 (6789)
▾ Real-Time Transport Protocol
    10.. .... = Version: RFC 1889 Version (2)
    ..0. .... = Padding: False
    ...0 .... = Extension: False
    .... 0000 = Contributing source identifiers count: 0
    0... .... = Marker: False
    Payload type: DynamicRTP-Type-108 (108)
    Sequence number: 255
    Timestamp: 0
    Synchronization Source identifier: 0x00000000 (0)
    Payload: 624df35c28ea8607d0ca977a9483f99359191ea0c8015397...
0040  97 7a 94 83 f9 93 59 19  1e a0 c8 01 53 97 03 fe   .z....Y. ....S...
0050  a9 ac dd 73 bb c3 e4 99  4e 41 71 a5 ca 79 08 6e   ...s.... NAq..y.n
0060  29 a8 be fb c3 e9 93 67  5e 8f c8 01 7d 3d 60 0f   )......g ^...}=`.
0070  d2 08 1a ac de 3e c7 bc  74 61 21 05 5f f3 2d 1f   .....>.. ta!._.-.
0080  52 2d 82 12 7b 52 d8 31  dd 2d fa 69 02 60 7c c3   R-..{R.1 .-.i.`|.
0090  8f 2b a6 d9 d1 6d 02 90  ba 92 2d bc b4 3f a0 68   .+...m.. ..-..?.h
00a0  20 40 bb 01 2d 27 35 5a  2a 04 21 c6 30 3d 0a 77    @..-'5Z *.!.0=.w
00b0  64 1d bd d2 92 a5 4d 3b  62 3f 4b 26 fb 4f db 7f   d.....M; b?K&.O..
00c0  31 b9 ca 3f ce 6b e6 b8  93 eb 77 08 2a 0e 01 04   1..?.k.. ..w.*...
00d0  d6 90 b7 35 2e e0 74 36  79 84 54 00               ...5..t6 yT
○ Loopback: lo: <live capture in progress> ...   Packets: 4 · Displayed: 4 (100.0%)
```

add no overhead to the network, but it does not decrease the security of the VoIP network. The proposed method does not introduce any additional packet loss to the network because all the delivered SRTP packets will be successfully authenticated and decrypted at the destination. Because the proposed method does not add any traffic overhead or packet loss to the SRTP communication, the quality of service remains intact. The strength of the proposed method is closely related to its underlying user datagram protocol (UDP). Owing to the unreliable nature of UDP transport, for any UDP packet that is lost, its embedded secret data would also be lost.

In cases where a steganographer wants to encrypt the secret data prior to sending it to the network, it is advisable to use encryption algorithms and modes that do not relate the decryption process of secret message parts to each other. This is due to the unreliable nature of the covert channel, which does not guarantee the delivery of every single packet. In this way, if a packet is lost on the way, other parts of the secret message may still be recoverable.

To detect the secret message in this technique of data embedding, a warden must have access to the list of master keys and the pre-shared mapping of master keys and code words in each session, which in most practical cases is against the end-to-end privacy and security policies of the network. In cases where the lawful interception of conversations is restricted to voice recording, there would be no way for the warden to detect the existence

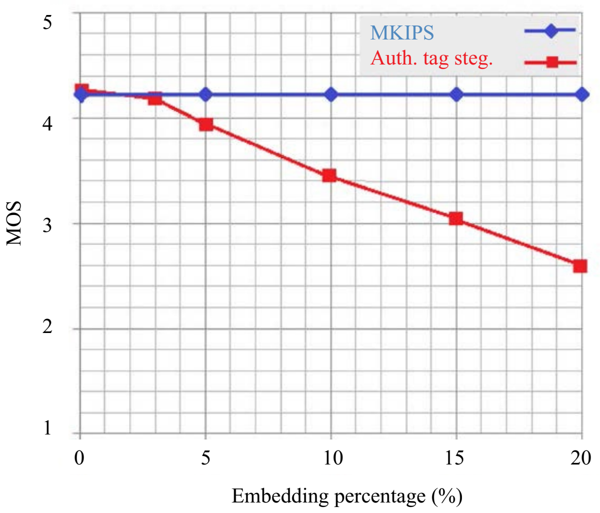


FIGURE 11 Subjective comparison of the voice quality of both methods shows the behavior of mean opinion score as the number of embedded packets increases

of the covert channel. In networks with a high rate of MKI field alteration, even by analyzing each packet individually, the warden will not be able to differentiate steganographic traffic with a legitimate one. In order to detect the existence of a proposed covert channel in a VoIP stream, the warden should suspect that the MKI value changes. Solutions based on intrusion detection systems can help to discover traffic abnormality within the MKI field of packets and mitigate risks of data leakage due to the existence of covert communication.

To increase the invisibility or strength of the proposed covert channel, the data transmission rate can be partially sacrificed. Since the only reasonable approach to detect the existence of the MKIPS covert channel is to analyze the fluctuation rate of the MKI field value, to avoid detection, the steganographer agent may intentionally increase the embedding cycle, that is, the steganographer alters fewer packets per time unit. In addition, to increase the strength of the covert channel, various channel coding algorithms might be used.

6 | EXPERIMENTAL RESULTS

To test and evaluate the possibility and effectiveness of the MKIPS technique, a proof of concept experiment was developed. In this process, SRTP packets were intelligently crafted to convey a secret message across the network. To simplify the experiment, all other factors with no effect on the final result were removed. In addition, to remove network complexities, a loopback connection with the same destination port was used. The goal of the experiment is to successfully embed the secret message in the MKI fields of SRTP packets while preserving the integrity of the covering media. At the receiving end, not only the secret message must be extracted, but the master key should also be correctly determined, and the covering media should be correctly decrypted. The correct decryption of the payload in the destination means that the integrity of the payload was preserved during the embedding process. The proof of concept implementation was developed in Python 3 programming language, and to analyze the results, a Wireshark protocol analyzer program was used.

Figure 4 shows the flowchart of a simple proof of concept implementation for the MKIPS steganographer agent. In this flowchart, a secret message along with the RTP packet and Crypto-Context are considered as inputs of the packet crafting mechanism. The secret message is the value that must be successfully delivered to the receiving agent across the network under the cover of the SRTP protocol. Crypto-Context is a collection of variables that are determined by system configuration or by other external protocols. The RTP packet contains the media that must be secured with encryption and authentication mechanisms of SRTP.

As an example, suppose that the minimal amount of 8 bits in each SRTP packet is reserved for the MKI value. Further, suppose that eight different master keys (recognizable by three bits) are available to both clients. These master keys have been negotiated by an external key exchange protocol prior to the SRTP session initiation. For simplicity, in this proof of concept, the ASCII coding scheme was chosen as a simple mapping between the characters of the secret message and the list of available master keys. Figure 5 shows a packet crafting client that is secretly sending the "Test" message through MKI fields.

At the receiving end, the only addition to a normal SRTP receiver is a scanning agent that scans the MKI portion of each packet and then determines the secret character according to the scanned value of the MKI. Figure 6 displays the flowchart of the MKIPS receiving agent, which is responsible for extracting secret data from the SRTP traffic.

TABLE 1 Comparison of MKIPS and authentication tag-based steganography

Security services available	Min. overhead	Max. covert channel capacity	Technique
Confidentiality, authentication, replay protection	0 bits	128 bytes/packet	MKIPS
Confidentiality, replay protection	32 bits	80 bits/packet	Auth, tag. steganography

In Figures 7–10, the protocol analyzer scan results of the network are presented. As can be seen, the secret characters are embedded in the MKI fields of four subsequent packets.

As discussed in RFC 3711 [14], the maximum allowed space for the MKI field in each packet is 128 bytes, whereas it is 80 bits for the authentication tag. In addition, as discussed earlier, MKIPS uses an intelligent choice of value to convey secret messages, while the authentication tag-based method overwrites the value in the trailer of the packet. Because the real value of the authentication tag is overwritten, the authentication security service will no longer be available.

The intelligent technique used to choose the MKI value used in MKIPS allows it to have no overhead. On the other hand, the overwriting technique used in authentication tag-based steganography leads to a loss of 32 bits to 80 bits of authentication data per packet. Table 1 compares several aspects of MKIPS and authentication tag-based steganography.

To compare the effect of the proposed MKIPS method with existing authentication tag-based steganography, an experiment was designed. In this experiment, in a fixed-length audio clip, the embedded percentage increased with uniform distribution. Therefore, at each step, more SRTP packets were used to convey the secret message. To quantify the quality of the sound, the mean opinion score (MOS) was measured at each step.

In addition to qualitative descriptions, such as "quite good" or "very bad," MOS is a numerical method of expressing voice and video quality. Note that there are some other quality measures such as peak signal to noise ratio and structural similarity index measure that compare the cover and the embedded voice or image [27,28]. However, the MOS is a subjective quality evaluation measure that is calculated as the arithmetic mean over single ratings performed by human subjects for a given content or system.

The results of this comparison are summarized in Figure 11. To conduct this experiment, it was assumed that the authentication, confidentiality, and replay protection security services of SRTP were enabled at both endpoints of a VoIP call.

As can be seen in Figure 11, in the authentication tag-based steganography, a uniform increase in the number of embedded packets in an SRTP stream decreases the voice quality at the receiving end. However, the MKIPS method keeps the voice quality intact.

7 | CONCLUSION

As presented in this paper, owing to the existing transparency in the SRTP header, many of the steganographic techniques that are applicable to RTP streams and are based on the alteration of values in the header fields are also applicable to SRTP streams. Protocol behavior-based steganographic techniques are also applicable to SRTP streams, provided that the sender and receiver of the steganogram are in full control of their VoIP machines (first scenario). On the other hand, because the payload of the SRTP is encrypted, steganographic methods that are based on the alteration of payload values, such as the least significant bit technique, are not applicable to SRTP. Even the slightest change in the payload of an SRTP packet would result in a false decryption of the media at the destination. Thus, any technique that is based on the alteration of the payload after its encryption is not applicable to SRTP streams.

The current study introduced MKIPS, a new steganographic technique that attempts to use SRTP packets to create covert channels within VoIP networks. As proposed in this paper, the MKIPS covert channel creation technique uses the MKI fields of SRTP packets to convey a secret message across the network. The resulting steganographic covert channel can reach a relatively high capacity while retaining every single packet and not adding to the packet loss of the VoIP network. The transfer rate of the MKIPS technique is heavily dependent on the size of the MKI field, which would have been negotiated prior to SRTP session initiation. The size of this field may change between 1 byte and 128 bytes, but it is fixed for each session. Moreover, the MKIPS method does not introduce any overhead to the existing media channel traffic, that is, it does not need to send extra packets or add new fields or bits to the existing packets. It also has a light implementation, which means that it requires heavy processing at neither the sender agent nor the receiving agent.

ORCID

Mohammad Fakhredanesh https://orcid.org/0000-0002-7442-176X

REFERENCES

1. M. Fakhredanesh and N. Sheikholeslami, *Improvement of transteg over VoIP*, J. Electron. Ind. (2019).
2. W. Mazurczyk and K. Szczypiorski, *Steganography of VoIP streams*, in On the Move to Meaningful Internet Systems: OTM 2008, vol. 5332, Springer, Berlin, Germany, 2008.
3. W. Mazurczyk, *VoIP steganography and its detection—A survey*, ACM Comput. Surv. **46** (2013), no. 2, 1–21, Article no. 20.
4. W. Mazurczyk and K. Szczypiorski, *Covert channels in SIP for VoIP signaling*, in Proc. Int. Conf. Glob. e-Secur. (ICGeS), (London, UK), June 2008, pp. 65–72.
5. P. Lloyd, *An exploration of covert channels within voice over IP*, M.S. Thesis, Rochester Institute of Technology, May 2010.
6. M. Mehić, J. Šlachta, and M. Voznak, *Hiding data in SIP session*, in Proc. Conf. Telecommun. Signal Process. (TSP), (Prague, Czech Republic), July 2015.
7. M. Fakhredanesh, R. Safabakhsh, and M. Rahmati, *A model-based image steganography method using Watson's visual model*, ETRI J. **36** (2014), 479–489.
8. N. Aoki, *A packet loss concealment technique for VoIP using steganography*, in Proc. Int. Symp. Intell. Signal Process. Commun. Syst. (ISPACS'03), (Awaji Island, Japan), Dec. 2003, pp. 470–473.
9. W. Mazurczyk and Z. Kotulski, *New VoIP traffic security scheme with digital watermarking*, in Computer Safety, Reliability, and Security, vol. 4166, Springer, Berlin, Germany, 2006, pp. 170–181.

10. N. Aoki, *Potential of value-added speech communications by using steganography*, in Proc. Intell. Inform. Hiding Multimedia Signal Process. (IIHMSP'07), (Kaohsiung, Taiwan), Nov. 2007, pp. 251–254.
11. N. Aoki, *VoIP packet loss concealment based on two-side pitch waveform replication technique using steganography*, in Proc. IEEE Region 10 Conf. (TENCON'04), (Chiang Mai, Thailand), Nov. 2004, pp. 52–55.
12. A. Giani, V. H. Berk, and G. V. Cybenko, Data exfiltration and covert channels, Dartmouth College, Hanover, NH, USA, 2006.
13. US Department of Defense, DOD 5200.28-STD, *Department of Defense Trusted Computer System Evaluation Criteria*, Dec. 1985.
14. IETF | RFC 3711, *Secure Real-Time Protocol (SRTP)*, 2004.
15. IETF | RFC 7201, *Options for Securing RTP Sessions*, 2014.
16. IETF | RFC 4568, *Security Descriptions for Media Streams: Session Description Protocol (SDP)*, 2006.
17. C. R. Forbes, A new covert channel over RTP, M.S. Thesis, Rochester Institute of Technology, Aug. 2009.
18. L. Bai et al., *Covert channels based on jitter field of the RTCP header*, in Proc. Int. Conf. Intell. Inform. Hiding Multimed. Signal, Process. (IIHMSP'08), (Harbin, China), Aug. 2008, pp. 1388–1391.
19. L. Yinga et al., *Novel covert timing channel based on RTP/RTCP*, Chin. J. Electron., **21** (2012), no. 4, 711–714.
20. G Shah, A Molina, and M Blaze, *Keyboards and covert channels*, in Proc. USENIX Secur. Symp. (Berkeley, CA, USA), July 2006, pp. 59–75.
21. G. Shah and M. Blaze, *Covert channels through external interference*, in Proc. USENIX Conf. Offensive Technol. (Montreal, Canada), Aug. 2009, p. 3.
22. S. Chen, X. Wang, and S. Jajodia, *On the anonymity and traceability of peer-to-peer VoIP calls*, IEEE Netw. **20** (2006), 32–37.
23. X. Wang, S. Chen, S. Jajodia, *Tracking anonymous peer-to-peer VoIP calls on the internet*, in Proc. ACM Conf. Comput. Commun. Secur. (CCS'05), (New York, NY, USA), Nov. 2005, pp. 81–91.
24. S. S. Schmidt et al., *A new data-hiding approach for IP telephony applications with silence suppression*, in Proc. Availability, Reliab. Secur. (ARES '17), (Reggio Calabria, Italy), Aug. 2017.
25. M. Hamdaqa and L. Tahvildari, *ReLACK: A reliable VoIP steganography approach*, in Proc. Int. Conf. Secur. Softw. Integration Reliab. Improv. (SSIRI'11), (Jeju, Rep. of Korea), Aug. 2011, pp. 189–197.
26. K. Ahsan and D. Kundur, *Practical data hiding in TCP/IP*, in Proc. Workshop Multimed. Secur. Nov. 2002.
27. H. A. Moghadasi and M. Fakhredanesh, *Speech steganography in wavelet domain using continuous genetic algorithm*, J. Math. Comput. Sci. **11** (2014), 218–230.
28. M. Fakhredanesh, M. Rahmati, and R. Safabakhsh, *Steganography in the discrete wavelet transform based on the human visual system and cover model*, Multimed. Tools Appl. **78** (2019), 118475–18502.

AUTHOR BIOGRAPHIES

Amir Mahmoud Alishavandi received his BS degree from the Islamic Azad University of Shahre Rey, Tehran, Islamic Rep. of Iran, in telecommunication engineering in 2015. He then received his MS degree in secure telecommunication and cryptography from Malek Ashtar University, Tehran, Islamic Rep. of Iran, in 2017. He has been working as a security enthusiast and researcher since 2014. His research interests include steganography, steganalysis, computer networks, and secure communication.

Mohammad Fakhredanesh received his BS, MS, and PhD degrees in computer science and engineering from the Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran, in 2005, 2007, and 2014, respectively. Since 2015, he has been an assistant professor at the Malek Ashtar University of Technology, Tehran, Islamic Rep. of Iran. His research interests are the fields of steganography, steganalysis, image processing, and artificial intelligence.