DOI: 10.20472/IAC.2016.024.091

AYSE TUNA
Trakya University, Turkey

RESUL DAŞ
Fırat University, Turkey

GURKAN TUNA
Trakya University, Turkey

DESIGN AND IMPLEMENTATION OF A SOFTWARE APPLICATION FOR SECURE WEB-BASED COMMUNICATION FOR PEOPLE WITH SPEECH DISORDERS

Abstract:

All people need to communicate with their families, friends, neighbours, and physicians through various communication means. However, there are common barriers to communication needs of mute or/and deaf people, most importantly in terms of the declining sensory and physical abilities. On the other hand, technological advances have led to the development of a variety of new tools and technologies through various communication medium including the Internet for assistive and augmentative communication for individuals with disabilities. Although communication through the Internet seems highly attractive, it poses privacy and security issues. In this paper a web-based secure message transfer application is proposed to address privacy related information security threats directed to software-based communication of mute and deaf people. The proposed application is practical and easy-to-implement, and is based on the integration of cryptology and steganography. With its easy-to-use graphical user interface suitable for different devices, it can easily be used by its potential users for ensuring the privacy and security of their message transfers. The main limitation of the proposed application is that the application requires a public web server and hence may be vulnerable against attacks directed to web servers if required precautions are not taken.

Keywords:

Security threats; message security; cryptology; steganography.

JEL Classification: L86, C88

Introduction

The ability to communicate is essential for all people. However, it is more essential for mute or/and deaf people since in addition to communicating with their families, friends, and neighbours, they need assistance with daily living activities and enjoy intergenerational contacts with family members, gain access to health and legal information, and fulfil lifelong learning goals. On the other hand, they have major barriers to their communication including sensory and physical abilities.

Deaf people use a variety of methods of communication such as sign language (SL) and lip reading (LR). Therefore, many deaf people communicate quite well with other deaf people and with people that can hear well. However, although many deaf people communicate with SL, many hearing people do not know SL. Also, some deaf people do not know SL and rely on LR. Moreover, all people who know SL do not use the same form of SL. Similarly, some mute people use SL to communicate. Others use alternative methods of communication including written notes, flash cards, Bliss symbols, pictures, words, helper pages, speech recording and replaying, vocalization, and LR lip-reading by the communication partner. In addition, some mute patients have adapted to their disability by using machines which can vibrate their vocal cords and this way allows them to speak.

In the last decade, due to the increasing interest in new technologies for assistive and augmentative communication for individuals with disabilities, new ways of communication has been developed. Especially due to the advances in information and communication technologies, computers, smart phones and the Internet has together offered alternative methods of communication to deaf and mute people. Nowadays, there are many augmentative and alternative communication devices which allow deaf and mute people to communicate such as hardware or software-based text-to-speech solutions. Nowadays, there are some practical and low/no-cost solutions provided by the Internet. Deaf and mute people use can us e-mail programs, chat applications, social media platforms and the Internet to interact with mute and deaf people and with hearing people.

Although some secure options for sending and receiving e-mails and information are available, most of the founding technologies and protocols which transfer data in clear text are still used and therefore the e-mails are insecurely transferred through the Internet. On the other hand, due to the increasing availability of the Internet, this becomes a more severe issue as the data can be captured by eavesdroppers or hackers while it is moving through the Internet (Wong, Gouda and Lam, 2000; Dolev and Yao, 1981; Bloch et al., 2008). Basically, there are two main concerns which can be applied to both message senders and recipients. As well as the security of communication links, the security of messages and attachments must be provided. For these goals, in order to provide confidentiality, integrity and authenticity of messages delivered between Internet users, secure protocols and encryption techniques can be employed to ensure the security of e-mail communications and data transfers (Zhou, Fang and Zhang, 2008; Karlof and Wagner, 2003; Yang et al., 2004; Avancha et al., 2002).

In this paper, a web-based secure message transfer application is proposed to address common information security risks encountered during message transfers, and in this way ensure the integrity and privacy of private user messages.

Proposed Web-Based Communication Framework

The proposed application shown in Figure 1 and Figure 2 has two user types: member and non-member. While members can create and send messages, non-members can receive messages. The proposed application is based on the integration of cryptology (Dzung et al., 2005) and steganography (Provos and Honeyman, 2003), and operates in three main steps. In the first step, the message typed by the user is encrypted using AES (Daemen and Vincent, 2003; Murphy, 1999). Cryptanalysis studies showed that AES is strong against different kinds of attacks (Bogdanov, Khovratovich and Rechberger, 2011; Biryukov and Khovratovich, 2009). Then, the encrypted message is embedded in a cover image. Finally, a code to be used to reach the encrypted message is created by the algorithm and sent to the intended recipient. Using the code, the recipient can get the original message. The member users can also stop receiving hidden messages.

Figure 1: Homepage of the proposed application

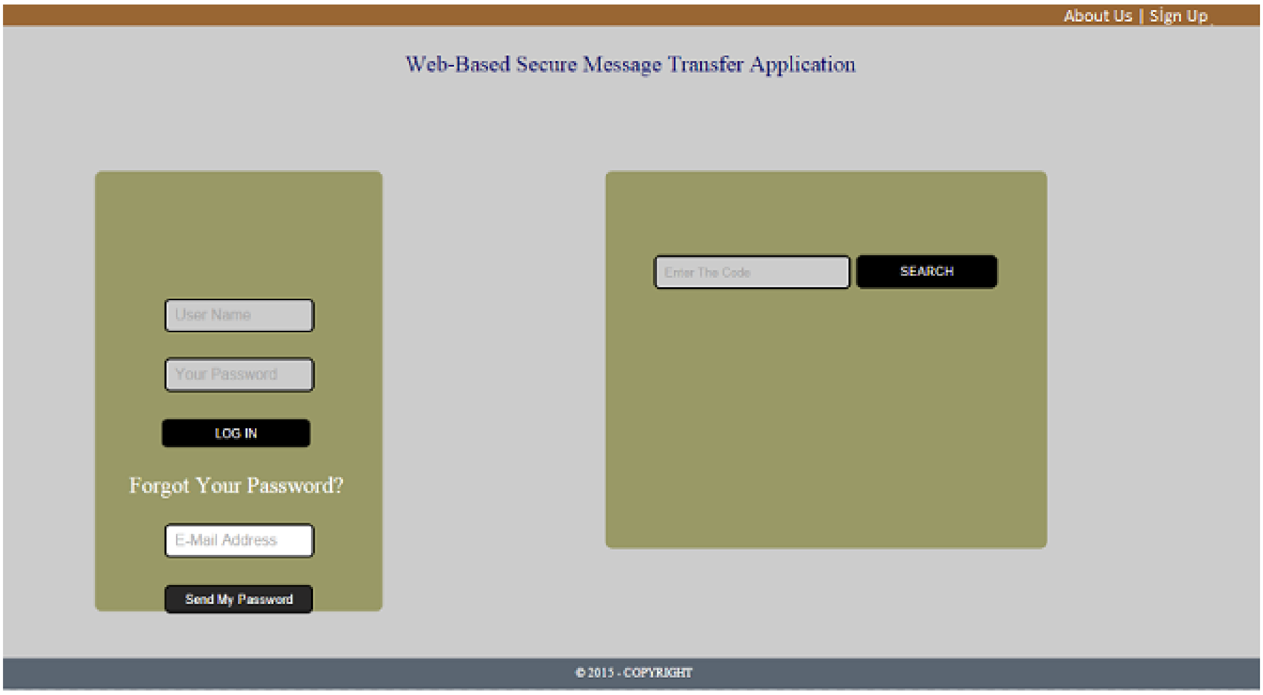


Figure 2: Creating a secure e-mail using the proposed application

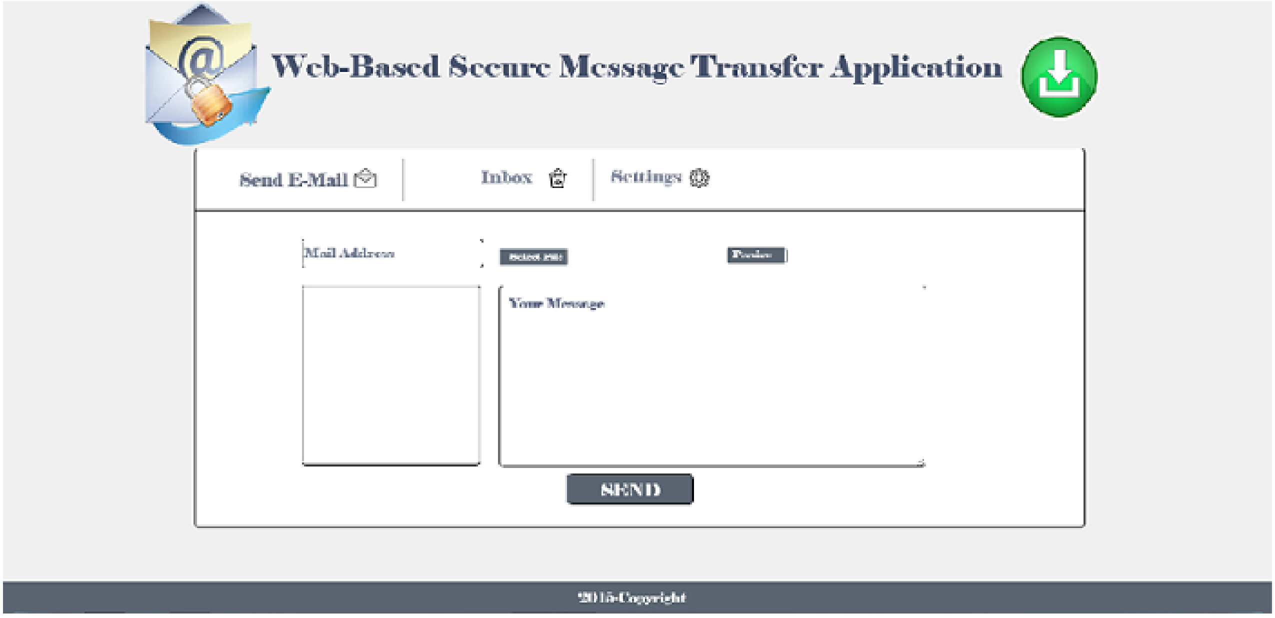


Figure 3: Steps used by the proposed framework to send an e-mail

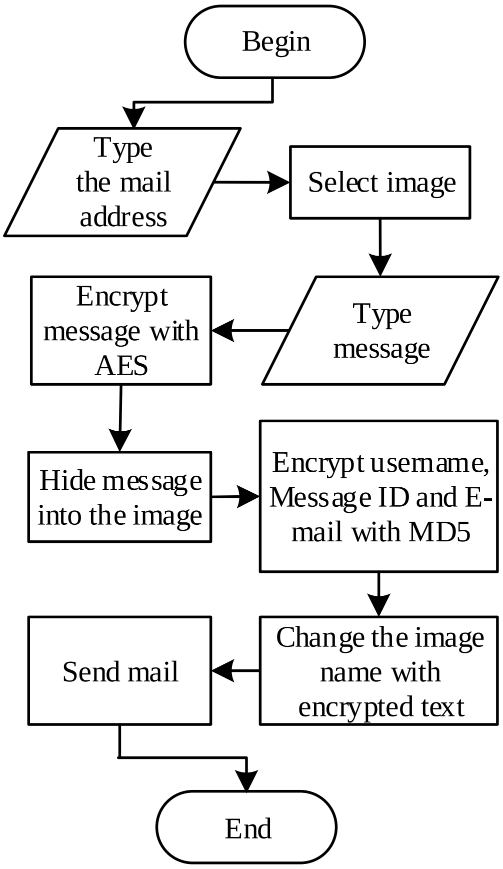
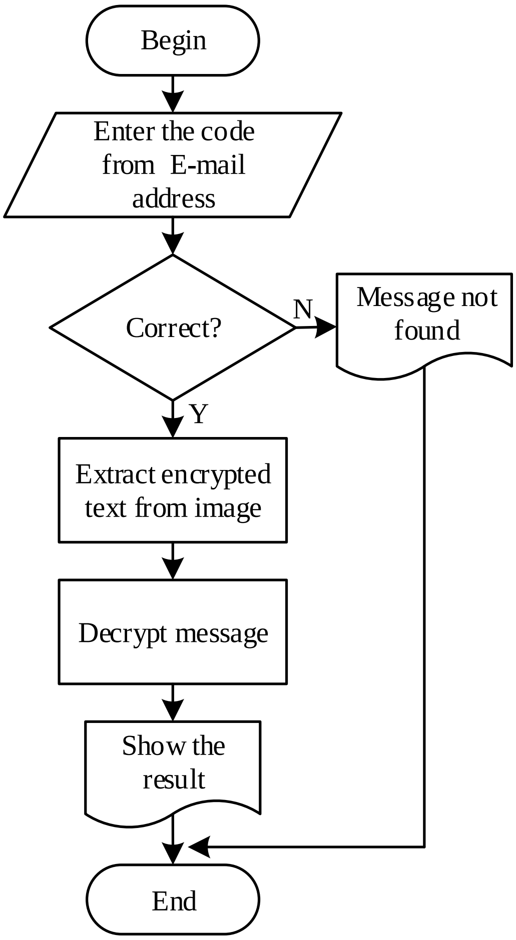

Begin
Type the mail address
Select image
Type message
Encrypt message with AES
Hide message into the image
Encrypt username, Message ID and E-mail with MD5
Change the image name with encrypted text
Send mail
End

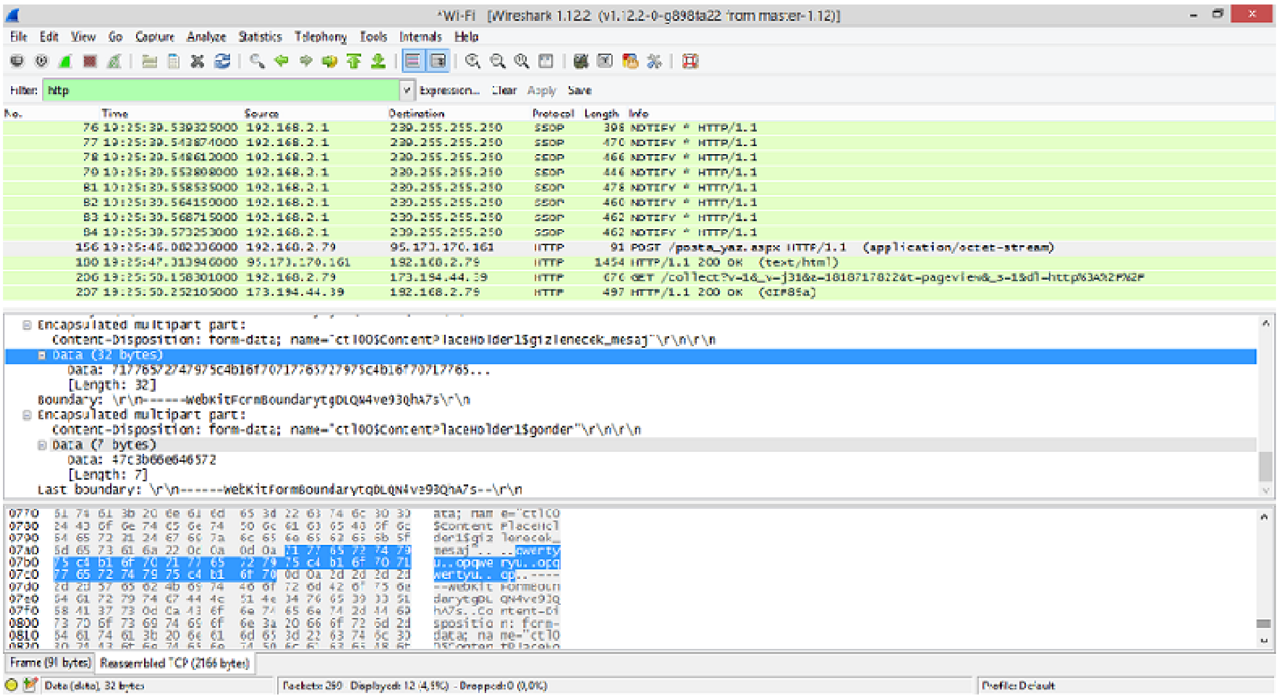
Figure 4: Steps used by the proposed framework to receive an e-mail


Begin
Enter the code from E-mail address
Correct?
N
Message not found
Y
Extract encrypted text from image
Decrypt message
Show the result
End

After the user enters the recipient’s e-mail address, he/she selects an image which will be used in the encryption of the message. Then, he/she types the message and clicks Send button. Flowchart shown in Figure 3 shows all the steps used by the proposed framework to send an e-mail. After the recipient’s enter the code shared between the two parties, if the code is correct, then the server allows extracting the message text from the image. Then, the message is decrypted and displayed to the recipient. Flowchart shown in Figure 4 shows all the steps of the reverse process used by the proposed framework to receive an e-mail.

To analyze and prove the effectiveness of the proposed approach in protecting the privacy and security of message transfers, a sample case study was realized. As shown in Figure 5 and Figure 6, the case study consisting of a packet tracking using Wireshark (Wireshark.org, 2016) and an analysis with Fiddler (Telerik.com, 2016) showed that the proposed application ensured the privacy of the user message.

Figure 5: Packet tracking using Wireshark

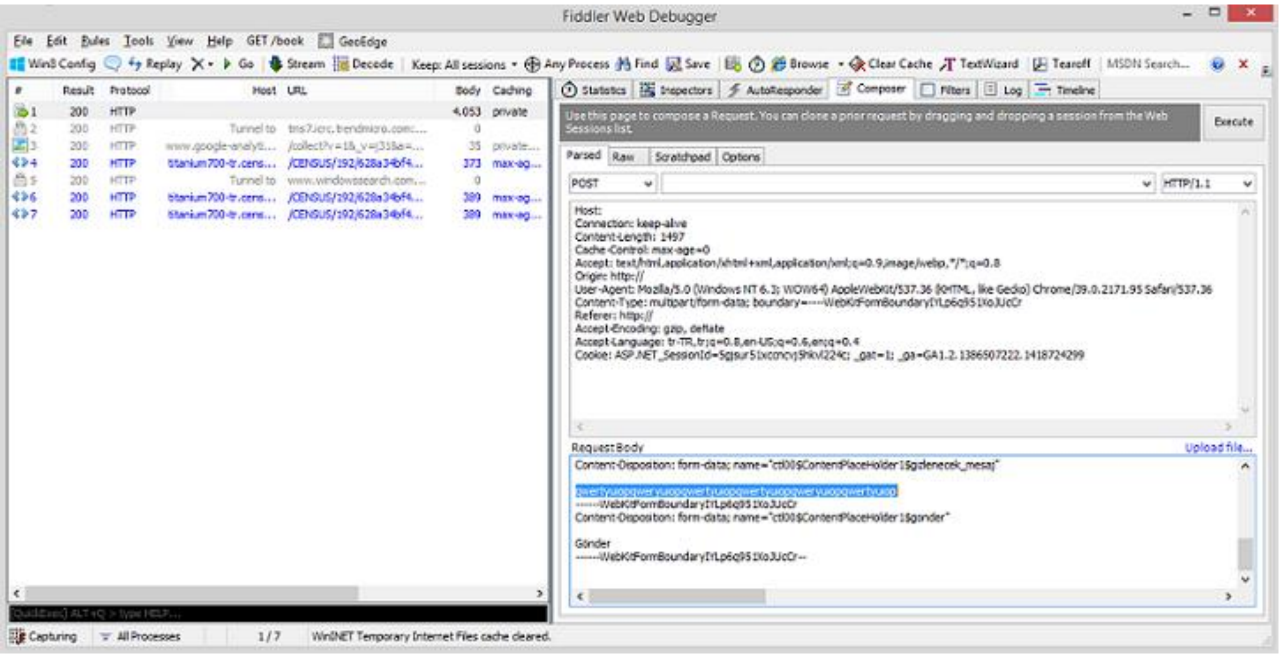

*Wi-Fi [Wireshark 1.12.2 (v1.12.2-0-g898fa22 from master-1.12)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter: http
Expression... Clear Apply Save
Frame (91 bytes) Reassembled TCP (2166 bytes)
Data (data), 32 bytes
Packets: 260 · Displayed: 12 (4,6%) · Dropped: 0 (0,0%)
Profile: Default

```
No.  Time                Source          Destination      Protocol Length Info
  76 19:25:39.539325000 192.168.2.1     239.255.255.250  SSDP      398 NOTIFY * HTTP/1.1
  77 19:25:39.543874000 192.168.2.1     239.255.255.250  SSDP      470 NOTIFY * HTTP/1.1
  78 19:25:39.548612000 192.168.2.1     239.255.255.250  SSDP      466 NOTIFY * HTTP/1.1
  79 19:25:39.553808000 192.168.2.1     239.255.255.250  SSDP      446 NOTIFY * HTTP/1.1
  81 19:25:39.558535000 192.168.2.1     239.255.255.250  SSDP      478 NOTIFY * HTTP/1.1
  82 19:25:39.564150000 192.168.2.1     239.255.255.250  SSDP      460 NOTIFY * HTTP/1.1
  83 19:25:39.568715000 192.168.2.1     239.255.255.250  SSDP      462 NOTIFY * HTTP/1.1
  84 19:25:39.573253000 192.168.2.1     239.255.255.250  SSDP      462 NOTIFY * HTTP/1.1
 156 19:25:46.082336000 192.168.2.79    95.173.170.161   HTTP       91 POST /posta_yaz.aspx HTTP/1.1  (application/octet-stream)
 180 19:25:47.313946000 95.173.170.161  192.168.2.79     HTTP     1454 HTTP/1.1 200 OK  (text/html)
 206 19:25:50.158301000 192.168.2.79    173.194.44.39    HTTP      676 GET /collect?v=1&_v=j31&a=1818717822&t=pageview&_s=1&dl=http%3A%2F%2F
 207 19:25:50.252105000 173.194.44.39   192.168.2.79     HTTP      497 HTTP/1.1 200 OK  (GIF89a)
```

```
⊟ Encapsulated multipart part:
    Content-Disposition: form-data; name="ctl00$ContentPlaceHolder1$gizlenecek_mesaj"\r\n\r\n
  ⊟ Data (32 bytes)
      Data: 71776572747975c4b16f707177657274797 5c4b16f70717765...
      [Length: 32]
  Boundary: \r\n------WebKitFormBoundarytqDLQN4ve93QhA7s\r\n
⊟ Encapsulated multipart part:
    Content-Disposition: form-data; name="ctl00$ContentPlaceHolder1$gonder"\r\n\r\n
  ⊟ Data (7 bytes)
      Data: 47c3b66e646572
      [Length: 7]
  Last boundary: \r\n------WebKitFormBoundarytqDLQN4ve93QhA7s--\r\n
```

```
0770  61 74 61 3b 20 6e 61 6d  65 3d 22 63 74 6c 30 30   ata; nam e="ctl00
0780  24 43 6f 6e 74 65 6e 74  50 6c 61 63 65 48 6f 6c   $Content PlaceHol
0790  64 65 72 31 24 67 69 7a  6c 65 6e 65 63 65 6b 5f   der1$giz lenecek_
07a0  6d 65 73 61 6a 22 0d 0a  0d 0a 71 77 65 72 74 79   mesaj"....qwerty
07b0  75 c4 b1 6f 70 71 77 65  72 74 79 75 c4 b1 6f 70 71   u..opqwe rtyu..opq
07c0  77 65 72 74 79 75 c4 b1  6f 70 0d 0a 2d 2d 2d 2d   wertyu.. op..----
07d0  2d 2d 57 65 62 4b 69 74  46 6f 72 6d 42 6f 75 6e   --WebKit FormBoun
07e0  64 61 72 79 74 67 44 4c  51 4e 34 76 65 39 33 51   darytqDL QN4ve93Q
07f0  68 41 37 73 0d 0a 43 6f  6e 74 65 6e 74 2d 44 69   hA7s..Co ntent-Di
0800  73 70 6f 73 69 74 69 6f  6e 3a 20 66 6f 72 6d 2d   spositio n: form-
0810  64 61 74 61 3b 20 6e 61  6d 65 3d 22 63 74 6c 30   data; na me="ctl0
```

Figure 6: Web debugging proxy - Fiddler


Fiddler Web Debugger
File Edit Rules Tools View Help GET /book GeoEdge
Win8 Config Replay Go Stream Decode Keep: All sessions Any Process Find Save Browse Clear Cache TextWizard Tearoff MSDN Search...
Statistics Inspectors AutoResponder Composer Filters Log Timeline
Execute
Parsed Raw Scratchpad Options
POST
HTTP/1.1
Request Body
Upload file...
[QuickExec] ALT+Q > type HELP...
Capturing
All Processes
1 / 7

#	Result	Protocol	Host	URL	Body	Caching
1	200	HTTP			4,053	private
2	200	HTTP	Tunnel to	tns7.iorc.trendmicro.com:...	0	
3	200	HTTP	www.google-analyti...	/collect?v=1&_v=j31&a=...	35	private...
4	200	HTTP	titanium700-tr.cens...	/CENSUS/192/628a34bf4...	373	max-ag...
5	200	HTTP	Tunnel to	www.windowsearch.com...	0	
6	200	HTTP	titanium700-tr.cens...	/CENSUS/192/628a34bf4...	389	max-ag...
7	200	HTTP	titanium700-tr.cens...	/CENSUS/192/628a34bf4...	389	max-ag...

Use this page to compose a Request. You can clone a prior request by dragging and dropping a session from the Web Sessions list.

```
Host:
Connection: keep-alive
Content-Length: 1497
Cache-Control: max-age=0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Origin: http://
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/39.0.2171.95 Safari/537.36
Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryIYLp6q951XoJUcCr
Referer: http://
Accept-Encoding: gzip, deflate
Accept-Language: tr-TR,tr;q=0.8,en-US;q=0.6,en;q=0.4
Cookie: ASP.NET_SessionId=5gjsur51xccncvj5hkvl224c; _gat=1; _ga=GA1.2.1386507222.1418724299
```

```
Content-Disposition: form-data; name="ctl00$ContentPlaceHolder1$gizlenecek_mesaj"

qwertyuıopqwertyuıopqwertyuıopqwertyuıopqwertyuıopqwertyuıop
------WebKitFormBoundaryIYLp6q951XoJUcCr
Content-Disposition: form-data; name="ctl00$ContentPlaceHolder1$gonder"

Gönder
------WebKitFormBoundaryIYLp6q951XoJUcCr--
```

WinINET Temporary Internet Files cache cleared.

Conclusion

In this paper, to address potential information security risks encountered during message transfers of mute and deaf people, a web-based secure message transfer application is proposed. The proposed application is simple and easy-to-implement. Basically, it is based on the integration of cryptology and steganography, and is an efficient tool for ensuring the communication security of private user messages. The proposed application's member users can encrypt messages and embed them in cover images. They can also send codes to enable the others to receive the altered cover images which contain embedded messages and look at their own inboxes to see whether they have unread messages or not. On the other hand, the non-member users can only receive their messages by means of the codes they have received.

References

Wong, C. K., Gouda, M. and Lam, S. (2000). Secure group communications using key graphs. *IEEE/ACM Transactions on Networking*, 8(1), pp.16-30.

Dolev, D. and Yao, A. C. (1981). *On the Security of Public Key Protocols*. Technical Report. Stanford University, Stanford, CA, USA.

Bloch, M., Barros, J., Rodrigues, M. and McLaughlin, S. (2008). Wireless Information-Theoretic Security. *IEEE Trans. Inform. Theory*, 54(6), pp. 2515-2534.

Zhou, Y., Fang, Y. and Zhang, Y. (2008). Securing wireless sensor networks: a survey. *IEEE Communications Surveys & Tutorials*, 10(3), pp. 6-28.

Karlof, C. and Wagner, D. (2003). Secure routing in wireless sensor networks: attacks and countermeasures. *Ad Hoc Networks*, 1(2-3), pp. 293-315.

Yang, H., Luo, H., Ye, F., Lu, S. and Zhang, L. (2004). Security in mobile ad hoc networks: challenges and solutions. *IEEE Wireless Commun.*, 11(1), pp. 38-47.

Avancha, S., Undercoffer, J., Joshi, A. and Pinkston. J. (2004). Security for wireless sensor networks. In: *Wireless sensor networks*. Norwell, MA: Kluwer Academic Publishers, pp. 253-275.

Dzung, D., Naedele, M., Von Hoff, T. and Crevatin, M. (2005). Security for Industrial Communication Systems. *Proceedings of the IEEE*, 93(6), pp. 1152-1177.

Provos, N. and Honeyman, P. (2003). Hide and seek: An introduction to steganography. *IEEE Security & Privacy Magazine*, 1(3), pp. 32-44.

Daemen, J. and Vincent, V. R. (2003). *AES Proposal: Rijndael*. [online] Available at: http://csrc.nist.gov/archive/aes/rijndael/Rijndael-ammended.pdf [Accessed 11 May 2015].

Murphy, S. (1999). The Advanced Encryption Standard (AES). *Information Security Technical Report*, 4(4), pp. 12-17.

Bogdanov, A., Khovratovich, D. and Rechberger, C. (2011). Biclique Cryptanalysis of the Full AES. *Lecture Notes in Computer Science*, pp. 344-371.

Biryukov, A. and Khovratovich, D. (2009). Related-Key Cryptanalysis of the Full AES-192 and AES-256. *Advances in Cryptology – ASIACRYPT 2009*, pp. 1-18.

Wireshark.org. (2015). *Wireshark · Go Deep.*. [online] Available at: https://www.wireshark.org/ [19 December 2015].

Telerik.com. (2015). *Fiddler free web debugging proxy*. [online] Available at: http://www.telerik.com/fiddler [19 December 2015].