International Journal of u- and e- Service, Science and Technology
Vol.10, No.9 (2017), pp.115-124
http://dx.doi.org/10.14257/ijunesst.2017.10.9.12

An Imperceptible Security Method for Web Services

Ch. Sudhakar[1], Nakka Thirupathi Rao[1], Debnath Bhattacharyya[1] and Hye-jin Kim[2]

[1]*Department of Computer Science & Engineering*
Vignan's Institute of Information Technology
Visakhapatnam-530049, India
[2]*Business Administration Research Institute, Sungshin W. University,*
2, Bomun-ro 34da-gil, Seongbuk-gu, Seoul
nakkathiru@gmail.com, debnathb@gmail.com
hyejinaa@daum.net
(Corresponding Author)

Abstract

XML might be an all inclusive, stage independent dialect that portrays the way, information are frequently hang tight, exchanged and changed electronically. Security is that the prime worry in XML records. Steganography is that the workmanship and investigation of disguise information by implanting messages at interims diverse messages so unapproved clients can't get to the information. We tend to propose a Steganography strategy to be utilized for SOAP messages at interims web administrations conditions. The system is predicated on modifying the request of particular XML segments in accordance with a mystery message. This system envelops a high physical property, it leaves about no way because of misuse the correspondence convention as a shelter medium, and since it keeps the structure and size of the SOAP message in place. The technique is by experimentation substantial utilizing a conceivable circumstance accordingly on show its utility and worth.

Keywords*: Steganography, SOAP, XML, Web-Services*

1. Introduction

Steganography, recurring from the Greek words "stegos", which means "roof or covered" and "graphia" which suggests "writing", is that the art and science of concealment the actual fact that communication is going down. A secret message will be concealed within a trustful place where the data is stay put and might be sent while not understanding the presence of the key message that is termed as Steganography. It joins with on hand communication ways, and is employed to hold out concealed exchanges. Secrets will be hidden within all types of cowl source: audio, video or transmission, text, images. Steganography utilities currently days hide info within pictures, as this area unit comparatively simple to figure on. There are tools out there to stock up secrets within nearly any sort of concealment supply. It's conjointly doable to cover info within sounds, texts and video films. The foremost vital property of a canopy supply is that the quantity of knowledge that may be keep within it, while not dynamical the noticeable properties of the quilt.

Received (May 15, 2017), Review Result (August 21, 2017), Accepted (September 11, 2017)

ISSN: 2005-4246 IJUNESST

Web Services

Web administrations give a stage supporting programming dialect which is free of innovation that can trade information of machine-to-machine connection over a system. Additionally, customers and different frameworks interface with the Web benefit utilizing an institutionalized XML informing framework, for example, SOAP. In this way, organized and wrote data can be traded between associates of appropriated condition utilizing the messages that were being generating from the SOAP. In Web benefits, the cooperation between specialist organizations and requesters happens regularly by means of SOAP messages. Subsequently, the messages that were being generated or observed from the sender to be intended to receiver offers a sort of Steganography based applications and the documents containing it. Subsequently, mystery data can be secured in SOAP messages and sent over the system to a planed goal. Fundamentally, a SOAP message is a XML report that contains content. Accordingly, Steganography techniques utilized for content records and XML archives can hypothetically be utilized for SOAP messages. Basically, a few or these techniques may be finished. Subsequently, we will outline and propose another Steganography technique to incorporate mystery data in SOAP messages. The present considered technique modifies the request being generated from the XML components as indicated by the mystery message to be secured.

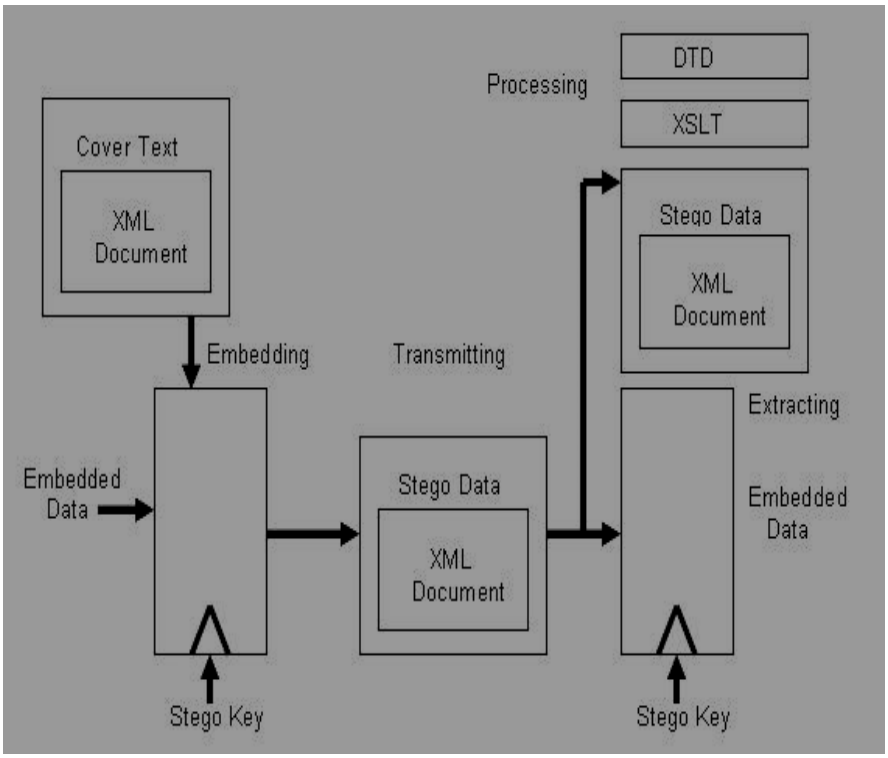

Figure 1. XML Steganography Model

2. Related Work

To create Steganography technique to be utilized to shroud cleanser messages inside web-administrations situations. The technique depends on reworking the request of a particular XML components as per given mystery message. The accompanying outlines us the way how the correspondence amongst sender and recipient goes ahead in our Steganography show. Considering the past idea, we have planned and executed an information concealing technique that screens a SOAP message shortly after its transform in the dispatcher endpoint prior to it is sent, breaks down its components and incorporates a mystery message in like manner. Figure 2 delineates the common model of information stowing away in SOAP messages. At the point while the stego message generated from the SOAP touches base at the recipient endpoint, the mystery message is sorting out a stego key so as to be distributed among the sender and collector.

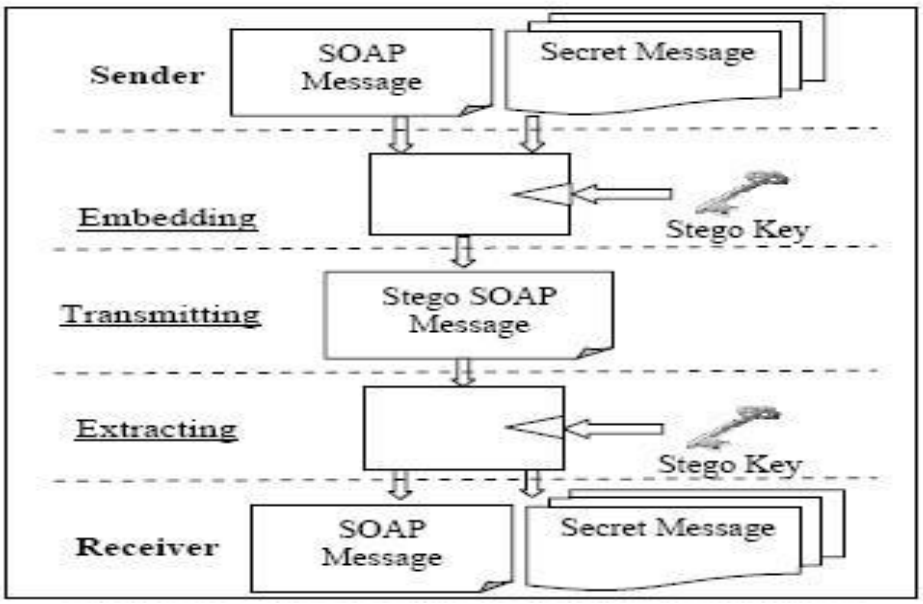

Figure 2. SOAP Steganography Model

Technique by hiding in Soap message:

Their implementation technique is as follows

- Consider a single character 'A' be a message to be Embed in the message
- Compute the ASCII value of A
- Find the ASCII value Sequence of permutation
- Shuffle the tags in that particular order.
- Using more number of tags more than 25 just to transmit a simple 5-letter message.
- They used ASCII value of letters which means always a particular value will have same value.
- Sending a long message is not reliable and which is always not use full.

Drawback:

- Using more number of tags more than 25 just to transmit a simple 5-letter message.
- They used ASCII value of letters which means always a particular value will have same value.

- Sending a long message is not reliable and which is always not use full.

Proposed Steganography:

In the earlier proposed systems for a job to accomplish we are in need of huge amount of data for communication. But in our system we just need sample amount of data for the job to get done. For eg.; In the previous technique we are making use of 25tags for sending a 5letter message. We in our system used just 5tags for sending the same 5letter message. For this we used hash table and a mod matrix.

1. **Recording SOAP message:**

   The Client requests the server for a web service. That request is sent in the form of a SOAP message. This SOAP message is captured or recorded using SOAP Handler.

2. **Embed Application:**

   This is taken as input by our embed application which takes secret message of the client and embed into the soap message by rearranging the tags.

3. **Extract Application:**

   Receiver uses the extract application to uncover the secret message.

3. Steganography on WEB-Services

Major Steps

1. Creating a database
2. Creating a Web Service
3. Creating a Client.
4. Creating a SOAP handler at Client Side
5. Creating a SOAP handler at Server Side
6. Writing Application to modify SOAP request and Convert to STEGO SOAP message
7. Writing Application to modify STEGO Soap message to Normal Message.
8. Displaying the SECRET Message.

Creating a Database:

- Go to services tab and start the database server.
- And connect to the database and then set APP as default schema.
- Create the table name as computers and specify the attributes and then insert the values in to table.

Creating a Web Service:

Creating web service

- Move to File – Select original File - Web services - Next
- state name of service being used from the web, CPwebservice
- state name of package being used from the web, cpproject
- Click Finish

Developing the Client…

- Right-click the CPclients.
- Choose New > Other > Web > Servlet. Click Next.
- Name the servlet NewServlet and type cpproject as package name.
- Drag and drop the operation node directly from the Projects window into the editor.

Process for creating a SOAP message:

Generate a message manager in an application being used from a Web in the IDE:

- To choose the project we need, first right click on the mode of nodes in the Applications of the Web
- Creating the original message manager.
- In order to work with the handler, the requirement of adding the handler with the name of the client for handler is required and after creating the event of handler clicks or applies the finish option.
- After the completion of creating the message handler for handling the messages, the user or the operator need to work combined with the service that was being operated from the web.

The process of involving or making involvement of handler of messages to services being offered by the web:

- In order to configure the handlers in the system, the users or the operators need to right click on the services that were being offered by the web in the projects window of the user manual.

4. Stego Soap Message

Algorithm:

- Read the input secret message to be embedded **(say "hello").**
- Develop mod matrix

Table 1. Mod Matrix

	0	1	2	3	4
0	E	A	B	C	D
1	J	F	G	H	I
2	O	K	L	M	N
3	T	P	Q	R	S
4	Y	U	V	W	X

Table 1 shows a Mod Matrix in which assign all alphabets based on Mod 5. For example alphabet **A** is stored in a place at 1st (1% 5) position. Alphabet **B** is stored at 2nd (2%5) position. Alphabet **E** is stored at 0th (5%5=0) position. Like that algorithm placed all alphabets as per modular formula.

Table 2. Hash Table

0	**E**
1	**L**
2	**L**
3	**H**
4	**O**

Table 2 shows the message "**HELLO"** is stored a in a place as based on **Table 1** and apply linear probing for hashing. For example based on table 1 the alphabet **H** is placed at 3rd position. The alphabet **L** is placed at 2rd position. Again the alphabet **L** is there and then by the method of linear probing it is placed at 1st position.

Develop the hash:

It uses the linear probing to fill its entries.
Order of tags shuffled is 30214.

Key generation technique:

- Consider the mod entry for letter **h it's entry is matrix[1][3].**
- Hash entry for h is hash[3] //no collision
- Key entry for h is 53
- Consider the mod entry for letter **o it's entry is matrix[2][0].**
- Hash entry for o is hash[4] //collision
- Key entry for h is 20
- **So key for "hello" is 5150522220.**

Extracting Techniques……

- Key received is 5150522220.
- From this 51 50 52 22 20
- Generate the order in which Tags are shuffled, which is 30214 in this case.
- Message is

Table 3. Mod Matrix

	0	1	2	3	4
0	e	a	b	c	d
1	j	f	g	h	i
2	o	k	l	m	n
3	t	p	q	r	s
4	y	u	v	w	x

matrix[1][3] ->h
matrix[0][0] ->e
matrix[2][2] ->l
matrix[2][2] ->l
matrix[2][0] ->0

Table 3 shows an Mod matrix in which explain place of an alphabet. For example the alphabet **H** placed at matrix [1][3] means 1st row 3rd element.

Displaying secret message….
The secret message "hello" is entered and applies key exchange algorithm as shown in figure 4.

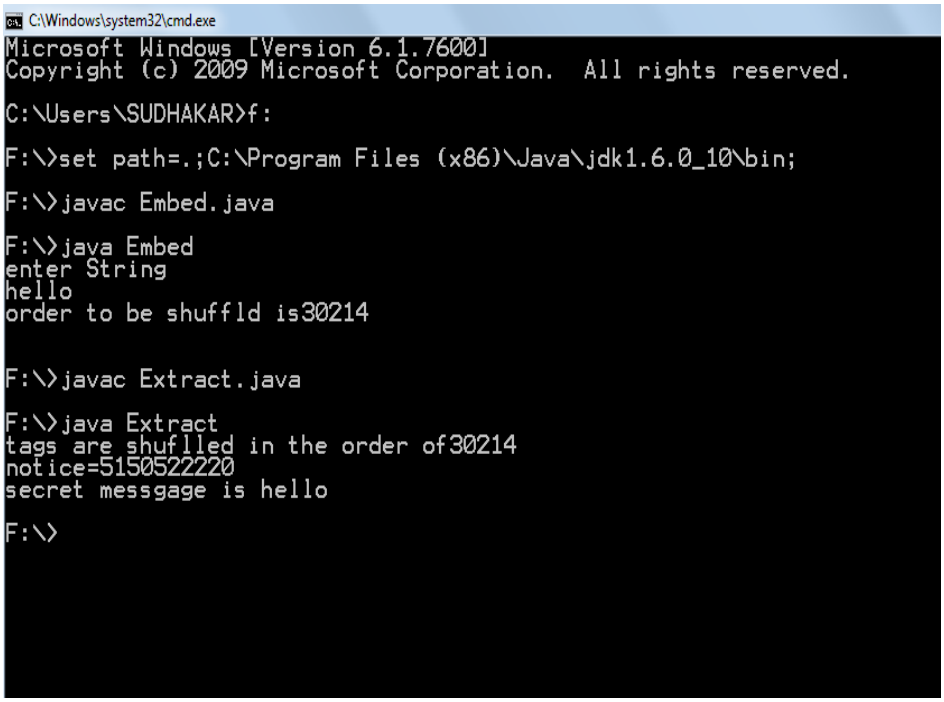

Figure 4. Apply Key Exchange

Figure 5 Shows, Generating more complex secret keys by exchange techniques and extending for longer message length greater than 5. Ultimately generates more Dynamic keys.

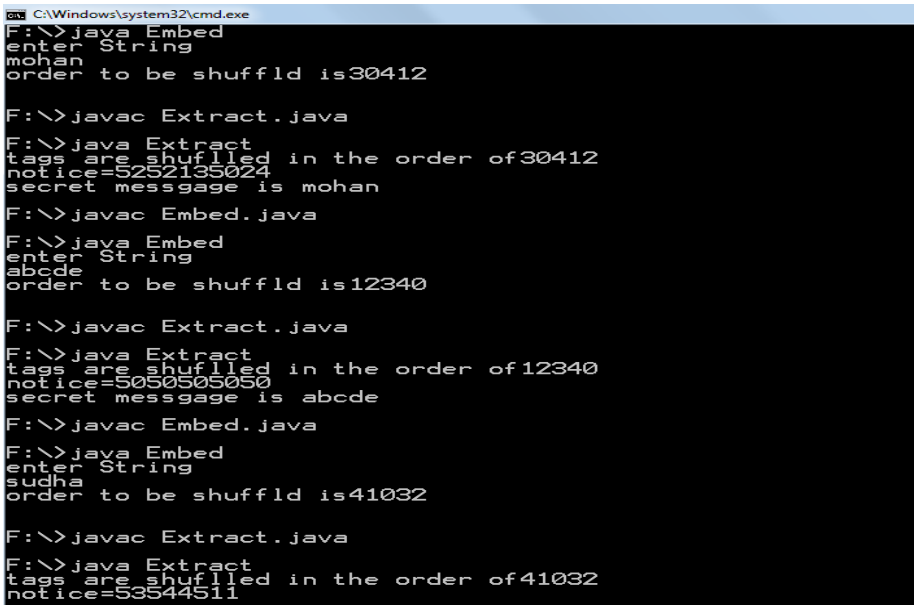

Figure 5. Extracting Dynamic Keys

Figure 6. Shows an entire algorithm works generating keys for secret message and transfer to receiver. The receiver was extracted the dynamic keys for the secret message which is sent by the user.

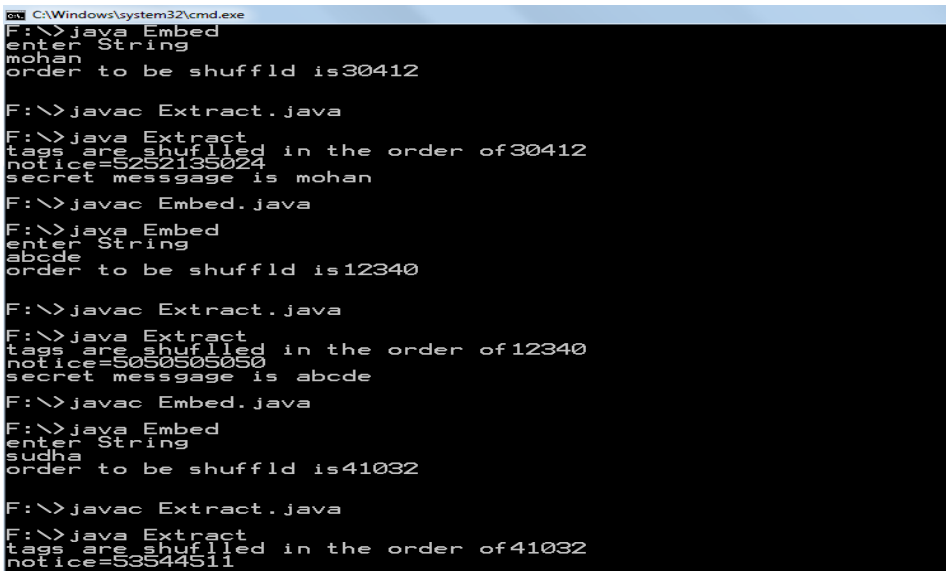

Figure 6. Display a Secret Message by Extracting Keys

5. Conclusion

In the current paper, we provide a soap protocol which is manipulated by communication based steganography with same size of original messages through a soap message. Steganography screens soon after registering in the sender end point and furthermore before it is sent by examining the cleanser components and conceal mystery message by adjusting the request of substance and properties of cleanser components. Here the mystery key is shared between the sender and recipient where each stage speaks to a particular image as it utilizes correspondence convention as cover medium this strategy gives high resistance which is not given by customary advanced documents. High

security levels are needed by distributed applications rather than internal applications still a ordinary encryption method can function on few numeral of platforms so because of it steg soap method can be a reasonable way for transmitting data where secret communication can be done using different kind of networks irrespective of application's used at the distributed endpoints to add another layer of system security. Generally encryption algorithm gives the normal data safety but the encrypted information is still visible to everyone to overcome the limitations and challenges to offer two times cover of safety this steg soap algorithm was used. It has a wide use of applications such as authentication, identity proof, message hash, water marketing and digital signature.

References

[1] K. Bailey, "Assessment of Pixel-Based Steganography and Stego detection Methods", The Imaging Science Journal, vol. 52, no. 3, pp. 131-150.
[2] S. Engle, "Current State of Steganography: Uses, Limits, and Implications", University of California, Davis Website, **(2003)**.
[3] D. Artz, "Advanced Steganography: Hiding Data inside Data", IEEE Internet Computing, vol. 5, no. 3, **(2001)**, pp. 75-80.
[4] M. Paprzycki and A. Abraham, "Importance of Steganography on Data Security", International Conference on Information Technology: Coding and Computing, **(2004)**.
[5] A. Lu, "Methods for Data Hiding", IBM Systems Journal, vol. 35, no. 3, **(1996)**, pp. 313-336.
[6] E. Newcomer, "Understanding Web Services: XML, WSDL, SOAP and UDDI", Addison Wesley, **(2002)**.
[7] J. Sun, "An Information Hiding Method in View of SOAP", Third International Conference on IIHMSP, **(2014)**.
[8] S. Inoue, K. Makino, I. Murase, O. Takizawa, T. Matsumoto and H. Nakagawa, "A Proposal on Information Hiding Methods Utilizing XML", First Workshop of NLP and XML, **(2001)**.
[9] L. M. Patnaik, "A Secure Image Steganography using LSB, DCT and Compression Techniques on Raw Images", IEEE Conference on Image Processing, **(2005)**; Bangalore.
[10] P. C. Reddy, "Implementation of LSB Steganography and Its Evaluation for Various File Formats", International Journal of Advanced Networking and Applications, vol. 02, no. 05, **(2011)**, pp. 868-872.
[11] M. S. Olivier, "An Overview of Image Steganography", Fifth Annual Information Security South Africa Conference, **(2005)**; Sand to South Africa.

