SECURE E-MESSAGING SCHEME USING SYMMETRIC KEY ENCRYPTION - EHDES

Awakash Mishra¹ and Deo Brat Ojha*²,

¹(Research Scholar Singania University, Jhunjhunu, Rajasthan)
Department of M.C.A, Raj Kumar Goel Engineering College, Ghaziabad, U.P., INDIA
awakashmishra@gmail.com
*² Deptt. Of mathematics, R. K. G. Institute of Technology, Gzb., U.P.(India),
deobratojha@rediffmail.com

Abstract: In this article, we present a secure scheme of e-messaging system for communication. It is the model of a high level secure mailing scheme for any organization. In this model, anyone can send a secret message even to any strange person in an unidentified way. The users of this model are assumed to be may or may not be the members of a closed organization. In the current era of communication, message security is too much important due to more usability of users and increases the stealing rapidly over internet.

Keywords: Message, EHDES, Steganography, Covert Mailing System, Random Number.

INTRODUCTION

Steganography has a relatively short history; even today ordinary dictionaries do not contain the word “steganography”. Books on steganography are still very few [1], [2].

The most important feature of this steganography is that it has a very large data hiding capacity [3], [4]. Steganography can be applied to variety of information systems. Some key is used in these systems when it embeds/extracts secret data. One natural application is a secret mailing system [5], [6] that uses a symmetric key. Another application pays attention to the nature of steganography whereby the external data (e.g., visible image data) and the internal data (any hidden information) cannot be separated by any means. We will term this nature as an “inseparability” of the two forms of data.

In this current paper, we will show an example of a mixed scheme of stegnography and cryptography are Secure E-Messaging Scheme Using Symmetric Key Encryption – EHDES, which are an anonymous and covert e-mailing system with complete security.

Present paper is as follows. In Section 2 describes the scheme of enhanced data encryption standard (EHDES) Section 3 we will show a secure messaging scheme using symmetric key. How we can make it a safe system in Section 4. Finally, section 5 is conclusion.

PRELIMINERIES

The amount of transfer messaging has increased rapidly on the Internet. Cryptography is a branch of applied mathematics that aims to add security in the ciphers of any kind of messages. Cryptography algorithms use encryption keys, which are the elements that turn a general encryption algorithm into a specific method of encryption. The data integrity aims to verify the validity of data contained in a given document. [7]

Enhanced Data Encryption Standard (EHDES)

In Enhanced Data Encryption Standard (EHDES) [8], [9], [14], we use the block ciphering of data and a symmetric key. As traditional Data Encryption Standard (DES), we also break our data into 64-Bit blocks and use a symmetric key of 56-Bit.

EHDES having three phases:

1. Key Generation.
2. Encryption on Input Data.
3. Decryption on Input Cipher.

Figure 1: Encryption and Decryption process of EHDES.

Key Generation: In this phase of EHDES, We moderate the initial 56 Bit key using Random Number Generator (RNG) [10], [11], [12], [13] for every block of message (M₁, M₂, M₃ ...Mₙ). The new generated 56 Bit keys (Knew1, Knew2, Knew3........ Knewₙ) from initial key K is used for encryption and decryption for each block of data. For new keys, we generate a random number and implement a function F on generated random number (N_RNG) and the initial key K.

© JGRCS 2010, All Rights Reserved
Figure 2: Process of new generated key (Knew) of EHDES.

**Encryption on Input Data:** As we know Data Encryption Standard (DES) is based on block cipher scheme. Message breaks in 64 Bit n blocks of plain text.

\[ M = \{ M_1, M_2, M_3, \ldots, M_n \} \]

Now, we encrypt our message \( \{ M_1, M_2, M_3, \ldots, M_n \} \) blocks by each new generated key
\[ K_{new1}, K_{new2}, K_{new3}, \ldots, K_{new n} \].

**Decryption on Input Cipher:** Decryption is the reverse process of encryption. For decryption, we also used the same key which is used in encryption. On the receiver side, the user also generate the same new key \( K_{new i} \) for each block of cipher and generate plain text through decryption process of data encryption standard.

---

**A MODEL OF AN E-MAILING SYSTEM**

Secure E-Messaging Scheme Using Symmetric Key Encryption –EHDES (SEMSUSK-E) is a steganography application program with cryptography. In the following description, \( M_{\text{ses Envelope I}} \) denotes a member \( \text{SES Envelope I} \), and \( M_{\text{ses Envelope II}} \) denotes a member \( \text{SES Envelope II} \). An SEMSUSK-E consists of the three following components.

3. Envelope Opener (EO).

We denote \( M_{\text{ses Envelope I}} \)’s SEMSUSK-E as \( \text{SEMSUSK – E}\) (i.e., customized SEMSUSK by \( M_{\text{ses Envelope I}} \)). So, it is described as
\[ M_{\text{ses Envelope I}} = (E_{\text{ses Envelope}}, M_{\text{ses Envelope I}}, E_{\text{ses Envelope I}}) \] is a component that produces \( M_{\text{ses Envelope I}} \)’s envelope \( (E_{\text{ses Envelope I}}) \) and a \( f = \sum_{i=1}^{E_{\text{ses Envelope I}}} \). \( E_{\text{ses Envelope I}} \) is the envelope (actually, an image file) which is used by all other members in the organization when they send a secret message to \( M_{\text{ses Envelope I}} \). \( (E_{\text{ses Envelope II}}) \) is produced from an original image \( (EO) \). \( M_{\text{ses Envelope I}} \) can select it according to his preference. \( (E_{\text{ses Envelope II}}) \) has both the name and e-mail address of \( M_{\text{ses Envelope I}} \) on the envelope surface (actually, the name and address are “printed” on image \( (E_{\text{ses Envelope II}}) ). It will be placed with function \( f \) at an open site in the organization so that anyone can get it freely and use it any time. Or someone may ask \( M_{\text{ses Envelope I}} \) to send it directly to him/her. \( (MI_{\text{ses Envelope I}}) \) is the component to insert (i.e., embed according to the steganographic scheme) \( M_{\text{ses Envelope I}} \)’s message into another member’s (e.g., \( M_{\text{ses Envelope II}} \))’s envelope \( (E_{\text{ses Envelope II}}) \) when is sending a secret message \( (M_{\text{ses Envelope II}}) \) to \( (M_{\text{ses Envelope I}}) \). One important function of \( (M_{\text{ses Envelope I}}) \) is that it detects a key \( (Key_{\text{ses Envelope I}}) \) that has been hidden in the envelope \( (E_{\text{ses Envelope I}}) \), and uses it when inserting a message \( (M_{\text{ses Envelope II}}) \) in \( (E_{\text{ses Envelope II}}) \). \( (EO_{\text{ses Envelope II}}) \) is a component that opens (extracts) \( (E_{\text{ses Envelope II}}) \)’s “message inserted” envelope \( (E_{\text{ses Envelope II}} (M_{\text{ses Envelope II}})) \) which \( M_{\text{ses Envelope II}} \) received from someone as an e-mail attachment. The sender \( (M_{\text{ses Envelope II}}) \) of the secret message \( (M_{\text{ses Envelope II}}) \) is not known until \( (E_{\text{ses Envelope II}}) \) opens the envelope by using \( (EO_{\text{ses Envelope II}}) \).

---

**CUSTOMIZATION OF A SEMSUSK-E**

Customization of an SEMSUSK-E for member \( (M_{\text{ses Envelope I}}) \) takes place in the following way. \( (M_{\text{ses Envelope I}}) \) first decides a key \( (Key_{\text{ses Envelope I}}) \) with \( f = \sum_{i=1}^{E_{\text{ses Envelope I}}} \) where \( i \) is a positive integer, when he/she installs the SEMSUSK-E onto his computer. Let us suppose \( (Key_{\text{ses Envelope I}}) \) try to communicate at any time \( t \), then he/she picks up a number randomly form \( i \). Now, SEMSUSK-E generates \( f_i = \sum_{i=1}^{E_{\text{ses Envelope I}}} \) . Let \( R = f_i - f_I \). SEMSUSK-E generate a key \( (Key_{\text{ses Envelope I}}) \) with the help of R using EHDES key generation process. Then he types in his name \( (\text{Name}_{\text{ses Envelope I}}) \) and e-mail address \( (\text{Email addr}_{\text{ses Envelope I}}) \), \( (\text{Key}_{\text{ses Envelope I}}) \) is secretly hidden (according to a steganographic procedure in his envelope \( (E_{\text{ses Envelope I}}) \). This \( (Key_{\text{ses Envelope I}}) \) is eventually transferred to a message sender’s \( (MI_{\text{ses Envelope II}}) \) in an invisible way. \( (\text{Name}_{\text{ses Envelope II}}) \) and \( (\text{Email addr}_{\text{ses Envelope II}}) \) are printed out on the envelope surface when \( (MI_{\text{ses Envelope II}}) \) produces \( (E_{\text{ses Envelope II}}) \) by using \( (EO_{\text{ses Envelope II}}) \). \( (Key_{\text{ses Envelope II}}) \) is also set to \( (EO_{\text{ses Envelope II}}) \), when communicators wish to start the communication. \( (\text{Name}_{\text{ses Envelope II}}) \) and \( (\text{Email addr}_{\text{ses Envelope II}}) \) are also inserted (actually, embedded) automatically by \( (MI_{\text{ses Envelope II}}) \) any time \( (MI_{\text{ses Envelope II}}) \) inserts his message \( (M_{\text{ses Envelope II}}) \) in another member’s envelope \( (E_{\text{ses Envelope II}}) \). The embedded \( (\text{Name}_{\text{ses Envelope II}}) \) and \( (\text{Email addr}_{\text{ses Envelope II}}) \) are extracted by a message receiver.
HOW IT WORKS

When some member (M_secret) wants to send a secret message (M_secret) to another member (M_secret), whether they are acquainted or not, (M_secret) gets (e.g., downloads) the (M_secret)’s envelope (E_secret) and uses it to insert his message (M_secret) by using (M_secret). When (M_secret) tries to insert a message, (M_secret)’s key (Key_secret) is transferred to (M_secret) automatically in an invisible manner, and is actually used. (M_secret) can send (E_secret)(M_secret) directly, or ask someone else to send it to (M_secret) as an e-mail attachment. (M_secret) can be anonymous because no sender’s information is seen on E_secret. (M_secret) is hidden, and only (M_secret) can see it by opening the envelope. It is not a problem for (M_secret) and (M_secret) to be acquainted or not because (M_secret) can get anyone’s envelope from an open site.

CONCLUSION

SEMSUSK-E is a very easy-to-use system because users are not bothered by any key handling, as the key is always operated automatically. As SEMSUSK-E doesn’t need any authorization bureau, this system can be very low cost. All these features overcome the drawbacks of an encrypted mailing system.

REFERENCES

[7] Diego F. de Carvalho, Rafael Chies, Andre P. Freire, Luciana A. F. Martimiano and Rudinei Goularte, “Video Steganography for Confidential Documents: Integrity, Privacy and Version Control”, University of Sao Paulo–ICMC, Sao Carlos, SP, Brazil, State University of Maringa, Computing Department, Maringa, PR, Brazil.