ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 21, August 2017

Light Weight Cryptographic Tool for Enhancing the Security of Personal Communication

P.LakshmiPrasanna

Lecturer in Computer Science Department

CH.S D.S.T.Theresa's Autonomous College For Women,

Eluru.Andhra Pradesh.

prasanna_mnsk@yahoo.co.in

A.Mamatha

Lecturer in Computer Science Department

CH.S D.S.T.Theresa's Autonomous College For Women,

Eluru.Andhra Pradesh.

mamathasree15@gmail.com

Abstract: Recent days the data security in very important in any aspect. This paper presents a new approach for providing limited information only that is necessary for data transfer there by safeguarding the data and increasing the data security. Unfortunately, many of these techniques are difficultor expensive for protecting personal communication. In thispaper, we introduce a novel light weight cryptographic tool we developed for enhancing the security of personal communication.The new technique encrypts information with equal characterfrequencies and with a totally new character set. By building thecipher text with soagainst a common characterfrequency analysis. The self-created character set is not availableanywhere else for obfuscating information in a unique way. Thetool we developed can build a completely new character setwithin only a few minutes and vast amounts of data to obfuscatemessages can also be generated rather quickly. Since the newdesign of the cryptographic technique and the character set areunknown to attackers, the technique provides a reasonablesecurity enhancement for personal communication. The casestudy discussed in this paper has shown the effectiveness andefficiency of the technique.

Keywords: *Cryptography, randomgenerator, character set,characterfrequency*

Introduction

Two main goals of cryptography are protecting information confidentiality and integrity. Confidentiality seeks to keepinformation secret while integrity seeks to ensure thatinformation is not corrupted. Weaknesses in the confidentialityand protection of the integrity of a cipher may lead toundesired consequences. Vast amounts of valuable data, suchas personal emails and passwords, financial information andcontacts, are being communicated in today's society. Some ofthese data have been targeted and used in malicious ways dueto poor security protection. As a consequence of hackersknowing the data, people's privacy has been invaded,businesses have collapsed, and lives have been put at risk.Many cryptographic techniques have been developed, but fewof them are easy to use for general users. In addition, manycommunication systems such as some widely used emailsystems do not provide reasonable encryption of personalinformation purposely or because of the technical or businesschallenges. Therefore, an easy to use self-controlled securityenhancement tool for protecting the security of personalcommunication is needed.

In this research, we developed a novel and light weight cryptographic technique that can easily be integrated with communication systems to encipher information and provide reasonable security. The method creates new challenges to attackers with self-developed cryptographic technique and unknown character sets and put users a step ahead of malicious crackers. The new technique ensures certain previous cryptanalysis technologies will no longer be useful and cryptanalysts will have to search for new approaches and technologies to ty decipher an encoded message. In addition, a user can recreate a new character set for future communication as soon as he or she does not like to continue the communication with certain parties. Then future communication will be secure to the previous ones. The newly created cryptographic technique can be used to obfuscate many types of data. Possible applications of this cryptography could be used for enhancing the security of transmitting emails and instant messages. Currently, the crypt graphic technique only supports characters from the English alphabet, the numbers 0-9 and a space, but other characters can be easily added. Of course, the tool can be easily used for protecting sensitive data such as passwords and credit card numbers in personal computing devices.

The core of the technique is the self-created characters and the even distribution of each character in the cipher text. The encryption of plain text is further protected with large amounts of confusion stings. The created character set is located on a file and anyone without the file cannot see what the characters look like. We used a plain text alphabet consisting of 37 characters (i.e. 0 - 9, A - Z and a space) and each character is randomly mapped to a cipher character from the new set at the start of enciphering. A plain text is simply hidden by converting each character from the plain text into its mapped cipher character, generating stings of random length of random characters and placing them before each encoded character and then making all characters have equal frequencies. The stings, known as confusion stings, are easily generated by selecting a random number 1,000 through 30,000 as the length and then selecting random cipher characters for each index of the sting. After the confusion stings are inserted, a character count is done and characters are inserted so that each character appears equal amounts of times. The indexes of the characters from the original plain text and the correct characters that have mappings are appended at the end of the cipher text in encoded format. Deciphering the messagecanbe done by decoding the character mappings and the indexes and then locating each character at each of the decoded indexes and printing them.

Approach

Our solution focuses on keeping information secure by the use of newly created characters, randomness, large amounts of data, and equal character frequencies. The technique takes in a user's plain text message as input and outputs a much larger cipher text message. A communication initiator creates 91 new cipher text characters that are not available in any other computer. Each plain text character is randomly mapped to one of the cipher characters at the start of encoding a message. Each character from the plain text message is separated by random cipher stings which range from one thousand to tens of thousands of characters. All characters have the same frequencies to prevent characters from frequency analysis.

Enciphering the Message

In the technique, we use the 37 characters as plain text alphabet including A-Z, 0-9 and a space. The cipher text alphabet is consisted of 91 characters and each plain text character is randomly mapped to one of the cipher characters. Some cipher characters will not have a plain text character mapped to it. We first created a cipher alphabet by using Private Character Editor. Using Private Character Editor, we created the 91 different characters and each was assigned a unique Unicode code point. Unicode is a character encoding standard used on many computers. Every character has its own unique code point. The cipher text characters were assigned to a certain code point area called the Private Use Characters area. [8] discussed about Reconstruction of Objects with VSN. By this object reconstruction with feature distribution scheme, efficient processing has to be done on the images received from nodes to reconstruct the image and respond to user query. Object matching methods form the foundation of many state- of-the-art algorithms. Therefore, this feature distribution scheme can be directly applied to several state-of- the-art matching methods with little or no adaptation. The future challenge lies in mapping state-of-the-art matching and reconstruction methods to such a distributed framework. The reconstructed scenes can be converted into a video file format to be displayed as a video, when the user submits the query. This work can be brought into real time by implementing the code on the server side/mobile phone and communicate with several nodes to collect images/objects. This work can be tested in real time with user query results.

After the cipher text alphabet is created, we randomly assign each plain text alphabet character with one of the cipher text characters in the newly

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 21, August 2017

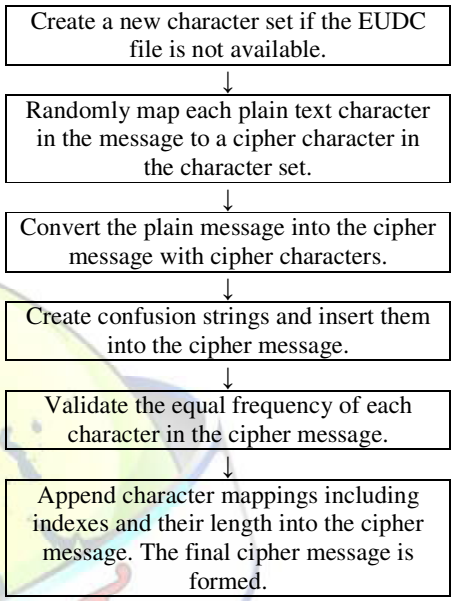
created character set. Each character from the user's message is then converted into its corresponding cipher character. A confusion sting S, or a string containing large amounts of random data of random length, is placed before each character to increase complexity.The strings are filled with randomly selected cipher characters at each index and their lengths randomly range from 1,000 to 30,000 characters. In order for the cipher text to be deciphered, the recipient must know the indexes of the characters from the sender's message M in the cipher text C. An integer array named charIndex is created and it holds the indexes needed to find every character in C that is converted from M. Each index holds the length of the confusion sting L added to the index of the previous character P which had its S inserted before it adding the number of confusion stings with addition of N.

$$Charindex[I\ ]=L+P+N$$

The pseudo code in Fig. 1 shows how to create the strings and insert them into the encrypted message and how indexes of each character from the sender's original message M are stored in charindex. For every while loop, a confusion string is created and inserted before an encoded character from M. timesAdded is a variable which keeps count of how many confusion stings have been inserted. For the first character index, we only insert the length of the confusion string plus I, because there is no previous confusion sting length. P is then set to the length of S. timesAdded is then incremented and for all characters afterwards, another S is created and added before the character. Now, the length of the confusion sting plus the length of the previous confusion sting plus timesAdded is inserted for all the character indexes. M is equal to the result after adding S plus every other character which has not had an S inserted before it at the end of a loop.

Once the confusion stings have been inserted before each character from M, the indexes 1 from charindex are appended to the end of M. Each number from each index is first encoded into its corresponding cipher text character and added. Every index is separated by a non-numerical cipher character to indicate the end of an index. In order for the deciphering program to know the correct mappings, each cipher character which is mapped to a plain text character Z, is appended to the end of 1 Each Z is separated by 512 randomly selected characters. So the decryption program will know where to begin reading the indexes, the length of the indexes T and the amount of non-numerical characters that separate them A are added. This number is encoded and appended to the start of M A non-numerical character is added after to mark the end of the number.

A flow diagram of the entire enciphering process:



Finally, the frequencies F of each cipher text character need to be equal and are counted. The highest frequency H is found and a new string E is created by appending cipher characters until each F is equal to H Each character in E is randomly ordered. By adding E to M, every character has equal frequencies. The final message is shown below

$$Cipher\ Text= (T + A) + M + E + 1 + Z$$

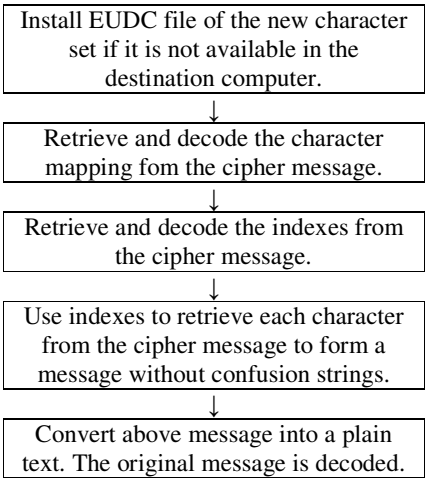
Deciphering the Message

Deciphering the cipher text can easily be done by locating the indexes of the characters from the sender's original message and identifying the correct character mapping. First, the index B of the beginning Z is calculated by using the formula shown below equation. Because there are 37 different Z's and each Z is separated by 512 characters and they are the last part of the cipher text, the index where the program needs to start reading characters is:

$$B=M.length-((512x37)+37)$$

Once B is found, the program reads the character at B and then every 51^{th} character to find which cipher characters are mapped to which plain text character.

Next, the starts of the indexes needed to find the characters from M are found. Each character from the start of C is read and converted into its mapped character until the non-numerical character is found. Each character before the non-numerical character is stored and now the length of the string of the indexes (T + A) is found. The indexes beginning at B minus the length of the indexes are read and stored. Finally, the character at each stored index is decoded and printed for the recipient. A flow diagram of the complete deciphering process is shown below:



Analysis

We believe this cryptographic technique provides a new, more complex method of securing data from malicious users without knowledge of our system. Because this is a newtechnique, certain previous technologies and cryptanalysis methods will not be able to decode our cipher text. Our technique adds complexity to decoding, because malicious users will most likely have to rely on researching for new ways to decipher it.

Due to Private Use Characters in our cipher text, a malicious user without the EUDC file will only see squares, literal question marks or possibly nothing at all depending on what software was used to view the message. An example of what a malicious user could see if a plain text was converted into its new Private Use Characters without the insertion of confusion strings is shown:

Plain text: GO TO POSITION X

Cipher text: □□□□□□□□□□□□□□□□□□□

Without understanding Unicode and the Private Use Area, malicious users may be confused and have to search for new methods of decoding. The cipher text may appear very complex and confusing, but is not impossible to decode. Obtaining the EUDC file or finding a way to determine where certain characters are located may aid the user in cracking the text.

Case Study

To implement our cryptography in a personal communication mechanism, such as email, we would first need to create a new cipher character set, and then it needs to be programmed to swap the randomly mapped cipher characters with the plain text characters, generate and insert the confusion strings, and format the message for deciphering. The character set stored in the EUDC file and the cryptographic tool for encoding and decoding messages are given to any person who joins in the communication. It would be more secure for users to not share or disclose to anyone besides the intended recipients of the existence or the newly created character set and the tool. Then, a sender inputs a message such as the body of an email to the tool, and the tool automatically sends the encrypted message to the destination user. As soon as the recipient side receives the message, the tool in the side decrypts the message with the character set located in the EUDC file, and finally the recipient is able to read the message.

Fig – 1: Sample cypher text

In order to demonstrate the effectiveness of our new technique, we compare our technique to the well-known symmetric AES encryption side by side in this section. The protection of information confidentiality and integrity of each technique will be observed. We start with examining how an actual message is enciphered using our technique. The plain text message that we hide with our technique is "THE PASSWORD IS HY1U25D2J7MD". A very small portion of the sample cipher text (it could be different with a different character set) after using our technique is shown below fig.

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 21, August 2017

Fig – 2: Sample display of cipher text in an unauthorized computer

Each of the 91 cipher character had frequencies of 1,975. In total, there were 179,725 cipher characters and only 28 (the length of the plain text) are meaningful. An attacker would have to locate the correct 0.02% of the cipher characters and determine the character mapping to read the message. Remember, if the attacker did not have the EUDC file, he or she would not see the appearance of our cipher characters. Examples of what he or she could literally see are a set of messed code (which could be different in different systems, but none of them should be readable). The attacker may see absolutely nothing depending on the program used to view the cipher text. He or she must obtain the EUDC file to understand the cipher characters or find a way to determine the actual character values from the squares or question marks. Now we examine the same plain text being encrypted using AES. We used 256 bit AES encryption with the password:

IK}P$_Y{'=+ac7qvsSII!%(gI6!uAn\f@!hGHRtn /B+1!@NvUKQSt)ylcEID

The final cipher text is:

U2FsdGVkXISGqjyOyT/2hJa+jtdlbr6azo4gkcS vHYzJwKQSLqnFmKlijTBPgQX4

Many widely used encryption techniques such as AES have been researched. If a cryptanalyst look at the AES cipher, he or she may recognize that this was encrypted message using AES due to its popularity. Successful side-channel attacks and related key attacks have been discovered and implemented against certain types of AES. Examples of side-channel attacks include time attacks and power consumption attacks. On the other hand, if a cryptanalyst were to examine our cipher, he or she would probably not recognize much because of its newness and novelty. Because no previous research or knowledge is available to a cryptanalyst, time and resources may have to be used to find how to decipher our cipher. Each of the techniques is still vulnerable to a brute force attack though. Both cryptographic techniques show many different combinations in keys or character mappings. AES has256 different keys and our technique has 2.8 x 1025 different character mappings. Our new 91 cipher character set can easily add more cipher characters due to thousands of available Private Use Area code points. This allows for the number of possible character mappings to highly increase and therefore make guessing the correct character mapping more difficult. Both AES and our technique must try to keep passwords, keys or files hidden. The integrity of both of the cipher texts may be vulnerable. In our cipher text, a malicious user may have difficulty tampering with the cipher text to create a meaningful plain text when decoded because of the ciphers newness and novelty, but could delete or change the text. AES is able to be used in OCB mode to help increase its integrity. Digital signatures and timestamps can be used to help improve the integrity of both cipher texts.

Our technique hides a message by inputting large, random length stings with random cipher characters before each character. We introduced a new way to encode data for cryptography using Private Character Editor and the Private Character Area in Unicode. Because the technique is new, previous cracking technologies may not be helpful to crack the message. Crackers have to do substantial research effort to try to decode the cipher text. A malicious user would be unable to view what our cipher text characters look like if he or she does not have access to the EUDC file. Our technique is resistant to a character frequency analysis and a brute force attack would be highly inefficient. If a cryptanalyst were to decode one cipher text, he or she may still have trouble decoding another cipher text because of different character mappings and different locations of the original plain text characters. Overall, we believe this technique pose new challenges for cryptanalysts and help cryptographers find new methods of enciphering data.

Results

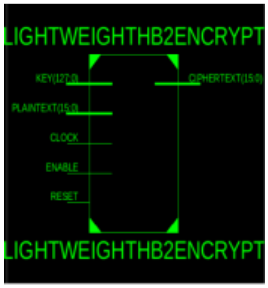


Fig – 3: Encryption Block

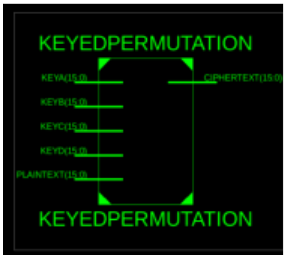


Fig – 4: Decryption Block

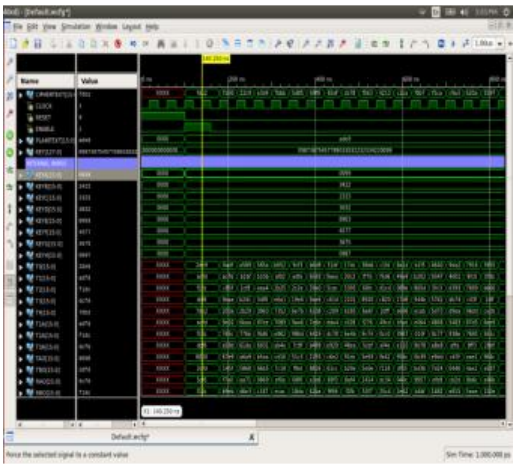


Fig – 5: Encryption Simulation

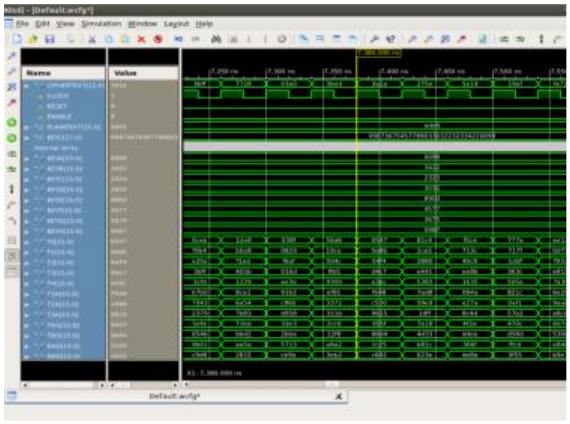


Fig – 6: Decryption Simulation

Conclusion

Cryptography is used to keep important information from malicious use. Electronic data, such as credit card numbers, intelligence data, and important emails, help our society function properly and may be vulnerable. We introduced a new technique which obfuscates data with a self-createdcharacter set, equal character frequencies, and large amounts of confusion data to enhance the security for personal communication. The technique is implemented as a software tool to automate the protection process. The tool will be integrated with other communication systems to offer additional security protection within closed communication parties. We have studied many cases using the new techniques, and the tool is available by request as well. In the future, we will extend our technique to prevent malicious cracking from people who even understand how our cryptographic technique works.

References

[1] A. Dhavare, R. Low and M. Stamp, "Efcient Cryptanalysis of Homophonic Substitution Ciphers," Master thesis report, San Jose State University, January 2012.

[2] Spezeski, William 1. "A Keyless Polyalphabetic Cipher." Technological Developments in Networking, Education and Automation (2010), pp.529-532. Springer. July 2013.

[3] Al Hasib, A.; Haque, A.A.M.M., "A Comparative Study of the Performance ad Security Issues of AES and RSA Cryptography," 3'd interational conference on convergence and hybrid information technolog, vol.2, no., pp.505 - 510, Nov., 2008.

[4] Jorg Roth, ComplexitTheor and Crptolog, Texts in Theoretical Computer Science, A EATCS Series, Springer, 2005.

[5] Oracle, Class SecureRandom, http://docs.oracle.com Ijavase/6/docs/api/javasecurity, Last accessed July 2013

[6] Biryukov, Dunkelman, Keller, Khovratovich, & Shamir "Key Recovery Attacks of Practical Complexity on AES Variants With Up To 10 Rounds" University of Luxembourg, The Weizmann institute CryptologeFrint Archive, Report 2009/374, 2009.

[7] AbuTaha, Farajallah, Taboub&Odeh "Survey Paper: Cryptography Is The Science Of Information Security," international Journal of Computer Science and Securit (JCSS), Volume (5) : Issue (3) : 2011.

[8] Christo Ananth, M.Priscilla, B.Nandhini, S.Manju, S.Shafiqa Shalaysha, “Reconstruction of Objects with VSN”, International Journal of Advanced Research in Biology, Ecology, Science and Technology (IJARBEST), Vol. 1, Issue 1, April 2015, pp:17-20

[9] Elaine B. Barker. "Recommendation for Digital Signature Timeliness". Technical

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 21, August 2017

Report 2009. SF 800-102. NIST, Gaithersburg, MD, 2009.

[10] Phillip Rogaway, MihirBellare, and John Black. 2003. "OCB: A blockcipher mode of operation for efcient authenticated encryption." ACM Transaction on Information System Security. vol. 6(3), pp. 365-403, August 2003.

[11] Kumar, G.P.; Murmu, A.K.; Parajuli, B.; Choudhury, P., "MULET: A Mu\tilanguage Encryption Technique," Seventh interationalCotiference on ltiformationTechnolog: New Generations (TNG), 2010, vol., no., pp.779 --782, April 2010.

[12] Unicode Character form http://www.unicode.or

ACKNOWLEDGEMENTS

1) Mrs.P.LakshmiPrasanna M.C.A is the Lecturer of Computer Science from CH. S.D. St Theresa's Autonomous College for Women Eluru, Andhra Pradesh.She received B.Ed from Nagarjuna University,Guntur in 2002,and received M.C.A from Andhra University Visakhapatnam in 2007.

2) Mrs.A.MamathaM.Sc,M.Tech is the Lecturer of Computer Science from CH. S.D. St Theresa's Autonomous College for Women Eluru, Andhra Pradesh. she receivedM.Sc Computers from Andhra University Visakhapatnam in 2007,and received M.Tech from Andhra University Visakhapatnam in 2014.

Mrs.B.Annapurna M.C.A is the department head of Computer Science from CH. S.D. St Theresa's Autonomous College for Women Eluru, Andhra Pradesh. She is an eminent researcher at the University Of Ontario Institute Of Technology (UOIT). She is currently persuing a Master's of Science in Computer Science from UOIT and received a Bachelor of Information Technology (2013)