ENHANCED DYNAMIC ENCRYPTION ALGORITHM

RANBIR SINHA [1], NISHANT BEHAR [2], DEVENDRA SINGH [3] AND RANDHIR SINHA [4]

Abstract : In this paper we introduce a cipher where the mapping from the plain text to the cipher text consists of a computational process that will generate a new encryption system in response to every combination of input messages and cipher keys. In this paper we present an encryption/decryption technique where dynamism is introduced. The application uses a dynamic table where the mapping of the original text is done with the rows of the dynamic table. Each character in the plain text is mapped to each row of the dynamic table. The dynamic identifiers from each row are collected together. The collected characters form the cipher text. The cipher text formed will have dynamic cryptanalytic properties, which we show is an obstacle that prevents the cryptanalyst from breaking the cipher. Ciphers of the introduced type may be adapted to implementation constraints and application specific issues, thereby substantially increasing the technical efficiency of implemented ciphers.

Keywords: encryption, decryption, cryptography, key, cryptanalysis, cryptanalyst, cipher construction, dynamic encryption, security.

Introduction : Encryption is the most effective way to achieve data security. The process of Encryption hides the contents of a message in a way that the original information is recovered only through a decryption process 2]. The purpose of Encryption is to prevent unauthorized parties from viewing or modifying the data 3]. Encryption occurs when the data is passed through some substitute technique, shifting technique, table references or mathematical operations. All those processes generate a different form of that data. The unencrypted data is referred to as the plaintext and the encrypted data as the cipher text, which is representation of the original data in a difference form 3].

Key-based algorithms use an Encryption key to encrypt the message. There are two general categories for key-based Encryption: Symmetric Encryption which uses a single key to encrypt and decrypt the message and Asymmetric Encryption which uses two different keys – a public key to encrypt the message, and a private key to decrypt it. Currently, there are several types of key based Encryption algorithms such as: DES, RSA, PGP, Elliptic curve, and others but all of these algorithms depend on high mathematical manipulations 5, 6].

One best way to encrypt data is through the mapping of plain text to the corresponding rows of the table to yield cipher text. Also, the use of dynamic table gives the algorithm - a dynamic nature - where nothing is known in advance. This dynamic mapping automatically inherits the concept of dynamic keys for each character in the plain text as introduced by "Ranbir Sinha" 1].

A message may be protected in transit or in storage by encryption 19, 21]. The input message M is called plain text. The cipher text C = f (K, M), an unintelligible form of the plain text, is computed as a function of the plain text and a finite secret cipher key K. The legitimate receiver may recover the plain text from the cipher text by applying an inverse transformation M = f-1(K, C). Both the sender and the receiver share a secret key K that must be distributed between the parties using secure means.

The scientist who constructs ciphers is known as a cryptographer. We assume that a cryptanalyst (or enemy) is one who will try to find the secret cipher key K, or convert the cipher text C back into intelligible form.

The distribution of message lengths, transmission frequencies, transmission errors, frequencies of repeated messages, context switching frequencies, etc., is likely to be different for every new application area. Ciphers may be used for many purposes, like link encryption, storage encryption, or authentication. It may be required that the cipher prevents a certain attack, or that it can easily be adapted to a certain plaintext format. Some ciphers will be implemented in software. Other ciphers will be implemented on Smart-Cards, FPGAs, or on ASICs. Different implementation strategies will perform differently. A mismatch between the selected cipher and the target technology and application area could decrease the technical efficiency obtained.

It is evident that no single cipher 24], no matter how flexible and efficient, will be optimal in meeting this challenge in respect to the above mentioned constraints. There will be a need for new cipher systems that meet the evolving demands of various application areas.

When a cryptographer is designing a new cipher, its security level may be difficult to establish. The security is an estimation of how difficult it would be to break the cipher without knowing the secret cipher key. Conventionally, it is assumed that the analysis made by the cryptographer and the cryptanalyst is based upon identical information, the

cryptanalyst knows the system being used 16] (Ch. 11.2.2). A key point, which we show in this paper, is that this condition is necessary. A cryptanalytic break implies that the cryptanalyst has obtained a part of the secrets of the cipher corresponding to the degree of success. This opens the possibility to challenge this fundamental assumption by introducing a construction that will prevent the cryptanalyst from learning the details of the cipher being used.

Basics : There are several kinds of Encryption software in the market categorized by their functions and target groups. For example, some are single Encryption applications for files and database security; some are for messenger security or email Encryption applications that hide the actual text in the medium between the sender and the receiver 9]. One of the first types of Encryption was made by Julius Caesar. 10]. in his system, Caesar wrote B instead of A and C instead of B – so to a sentence "ABC" will be written in "BCD" 7].

dsCrypt is AES/Rijndael file Encryption software with simple, multi-file, drag-and-drop operations. It features optimal implementation, performance and safety measures. dsCrypt uses an advanced Encryption algorithm and offers unique options for enhanced security11].

NeoCrypt is a free, open-source File Protection Utility for Windows. It helps to protect sensitive information easily by encrypting it with password or key. It yields fast, reliable and unbreakable Encryption and supports many popular Encryption algorithms. All types of files can be encrypted like Audio, Video, Documents and Executable programs 12].

NeekProtect is software in the market right now with the ability to make Encryption on any files in the window platform; a key is set when one try to encrypt the files and the key will be used again when someone else trying to open the files that have been decrypted through decryption on the certain files 13]. NeekProtect is a good software operated under Microsoft window because of the flexibility of this program's advanced features integration such as double click, file icons, .npt file extension etc.

This paper reports an encryption technique using dynamic table and mapping the plain text onto the dynamic table to yield the cipher text.

Methodology : The main Feature of the encryption/decryption program implementation is the mapping of the plain text to produce the encrypted text. Other features are related to the design of the GUI (Figure 1), progress of the encryption details, and user notification of the status of encryption.

Plain Text Mapping

Suppose the sender wants to send the confidential data to the receiver. So, he/she writes the data or attaches the file containing the data in the GUI of the algorithm.

The Plain text is divided into blocks of 128 characters each. If the last block contains less than 128 characters then some specific numbers of zeros are appended at the end of the last character of that block. The GUI maps the block onto the dynamic table. Then the characters resulting from the mapping in each row are taken and collected together in the form of a string. So, this string forms the encrypted text.

Encryption Rules

The rules to be followed in this encryption algorithm are as follows: -

- The plain text mapping procedure is executed for each block for procuring the encrypted text from the mapping of original text and the dynamic table.
- The difference is taken out by subtracting the mapped "character" and the "character identifying the row". This difference is taken out from each row of the dynamic table. The differences taken out are collected and written together. So, this difference forms the private key for the algorithm.
- The encrypted text can be sent to the receiver.
- The receiver has to enter the private key and the encrypted text/encrypted file received in the application and the application performs the decryption process to yield the original text/file which the sender wants to convey.

File Types

There are no limitations of the type of files accepted for encryption in this application, which means any type of a file such as data files, audio files, video files or image files can be encrypted by the application. This is because all the files are encrypted at the binary level. There is also no limitation of the size of the file that can be encrypted using this application, which provides flexibility to the user. The encrypted file can only be opened and viewed after it has been decrypted to its original file using the symmetric encryption key array.

Analysis of the proposed encryption algorithm : Any successful break of a cipher, equivalent to the complete recovery of the keys of the cipher, will reveal to the cryptanalyst all details of the cipher, except parts that cannot influence the cipher text. This is obvious as the cryptanalyst may simulate the encryption of a message using the recovered key. If the cryptanalyst has not obtained a sufficiently detailed description of the cipher to facilitate cryptanalysis, the cryptanalyst must first investigate the structure of the system prior to searching for the secret plaintext. We conclude that if the cryptanalyst cannot obtain a sufficiently detailed description of the cipher, cryptanalysis will not be possible.

If we, as another example, assume that the

cryptanalyst has been able to break a particular instance of the secret input stream Pn = {Mn,Kn}, the cryptanalysis method will apply to another stream only if the cryptanalysis method can be applied in general. If the break, on the other hand, exploits some specific weakness of the input stream Pn, the cryptanalysis method will only apply to this stream, and cannot work against any other combination of key and message. We conclude that, if specific instances of cipher text are vulnerable, the cryptanalysis effort must be restarted from the beginning for every transmitted message.

We note that the general problem of investigating the properties of software has been extensively studied in the software industry 19]. We may compare a cryptanalyst, attempting cryptanalysis on Dynamic Encryption, and a software engineer, struggling with debugging problematic software. The cryptanalyst would be investigating the properties of a universal machine that reads a string {M, K} and outputs a string {C}. We assume that the string {M} is known by the cryptanalyst, who attempts to find a key {K} such that {C} = {Co}.

The software engineer will be investigating the behaviour of a general purpose computer executing software {P} with input {X}. The engineer observes the output {Y}, and tries to find an input {X} that makes the computation behave in some specified way {Y} = {Yo}. The comparison {C} = computation ({M, K}); {Y} = computation ({P, X}) shows that breaking the proposed cipher will be at least as hard as debugging software. Clearly, the cryptanalyst will not be allowed to inspect the software, single-step using a debugger, or inspect the internal state of the memory, tools that are essential for the success of the software engineer.

Results: The interface of the application is simple enough to be used by any user. Figure 2 shows the user interface with the encryption and decryption buttons along with the plot. The encryption is performed simply by choosing any file while decryption is executed by choosing an encrypted file with an appropriate key. The software has been written in MATLAB 7.6 (R2008a).

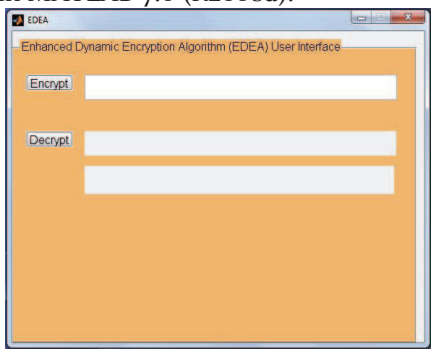

Fig. 1: Enhanced Dynamic Encryption Algorithm (EDEA) User Interface.

Application Testing and Performance Check

Application testing is applied to the entire application with multiple application features to make sure the application can encrypt all type and all sizes of files. Successful testing means the application is user-friendly and comfortable to be used by all range of target users. For performance checking, the application was tested with different types and sizes of files and the performance of the application was rated by computing the time required for encryption of the files. Also, the reliability of the application was examined by the success rate of encryption. A successful execution means an encrypted file is not visible by others; also successful execution means a decrypted file was obtained using a key array and an encrypted file. Table I shows the testing of different types and sizes of files. It can be seen that the encryption time is similar for all the files especially when the file size is small. The small size of files is a typical example of the use of this application as it is mainly targeted for small campus. Most of the documents used in this environment are of text type with some figures inside of the text; therefore, the sizes of the files may not go over few mega bites.

TABLE 1: Testing of the Application with Different Types and Sized Files.

File Types	File Size(Mb)	Encryption Time(s)	Success Rate (%)
Document	1/3/5	10/30/45	100
Image	1/3/5	11/28/47	100
Audio	1/3/5	19/29/45	100
Video	1/3/5	13/29/47	100

Conclusion : Most of the available encryption/decryption techniques are not perfect for applications over the Internet since they were originally built for text data, and due to their extensive computations which result in an unacceptable delay and processing time. The work in this paper attempts to develop a new encryption/decryption approach which adds a minimum delay time that makes it appropriate for cryptography. A new simple tool has been created, which is targeted for use inside of a small institution such as a small university for professors' daily use of sending exam files and sensitive material such that the material can be encrypted and the file is sent in one email while the encryption key is sent in another e-mail or via any secure communication channel.

References

1. Ranbir Sinha, "RS Dynamic Encryption (A Dynamic Encryption Technique)" in the Proceedings of the 2012 International Conference on Communication and Electronics Information (ICCEI 2012), Mumbai, INDIA, 14th – 15th January, 2012.
2. Wikipedia, "Encryption", http://en.wikipedia.org/wiki/Encryption , modified on 13 December 2006.
3. Freeman J., Neely R., and Megalo L. "Developing Secure Systems: Issues and Solutions". IEEE Journal of Computer and Communication, Vol. 89, PP. 36-45. 1998
4. Agnew G. B., Mullin R. C., Onyszchuk I. M., and Vqanstone S. A. "An Implementation for a Fast Public-Key Cryptosystems". Journal of Cryptology, Vol.3, No 2, PP. 63-79. 1995.
5. Beth T. and Gollmann D. "Algorithm Engineering for Public Key Algorithms". IEEE Journal on Selected Areas in communications; Vol. 7, No 4, PP. 458-466. 1989
6. IBM. "The Data Encryption Standard (DES) and its strength against attacks". IBM Journal of Research and Development, Vol. 38, PP. 243- 250. 1994
7. Wikipedia, "Bitwise operation", http://en.wikipedia.org/wiki/Bitwise_operation , last modified on10 December 2006.
8. Andy Wilson, "Tips and Tricks: XOR Encryption" http://www.andyw.com/director/xor.asp , 1998.
9. Baraka H., El-Manawy H. A., and Attiya A. "An Integrated Model for Internet Security Using Prevention and Detection Techniques". IEEE Journal of Computer and Communication Vol. 99, PP. 25-33. 1998
10. Microsoft, "Encrypting File System for Windows 2000", http://www.microsoft.com/windows2000/techinfo/howitworks/security/encrypt.asp , 1998
11. Dariusz Stanislawek , "Free Software copyright 1997 - 2006 " http://members.ozemail.com.au/~nulifetv/freezip/freeware
12. NeoCrypt, "NeoCrypt File Protection Utility", http://sourceforge.net/projects/neocrypt
13. Vivek Thakur, "NeekProtect", http://neekprotect.sourceforge.net , 2006.
14. Artur Ekert, Carolina Moura Alves, Ajay Gopinathan, "History of Cryptography".
15. Cryptomathic, "E-SECURITY DICTIONARY", http://www.cryptomathic.com/labs/techdict.html , 2003.
16. Bauer, Friedrich L. "Decrypted Secrets - Methods and Maxims of Cryptology", Springer-Verlag Berlin Heidelberg 1997, ISBN 3-540- 60418-9.
17. Biham, Eli and Shamir, Adi. "Differential Cryptanalysis of the Data Encryption Standard", Springer-Verlag, 1993, ISBN 0-387-97930-1, 3-540-97930-1.
18. Dömstedt, Bo and Stenfeldt, Mats. "Processing method and apparatus for converting information from a first format into a second format", Patent Applications PCT/SE99/01740; EP 98118910.3, Lateca Computer Inc N.V.
19. Fåk, Viiveke (ed.); Ekhall, Stig-Arne; Cristo_ersson, Per; Widman, Kjell-Ove; et al. "Crypto Users' Handbook", North-Holland 1988, ISBN 0-444-70484-1.
20. Hellman, M.; Merkle, R.; Schroeppel, R.; Washington, L.; Di_e, W.; Pohlig S.; Schweitzer, P. "Results of an Initial Attempt to Cryptanalyze the NBS Data Encryption Standard", Technical Report SEL-76-042, Sept 9, 1976, Information Systems Laboratory, Department of Electrical Engineering, Stanford University.
21. Herlestam, Tore. "Kryptering - för säkerhets skull", Tidningen Elteknikmed aktuell elektronik, pages 24-26 in #14, 1979.
22. Hopcroft, John E. and Ullman, Jeffrey D. "Introduction to Automata Theory, Languages and Computation", Addison-Wesley Publishing Company, 1979, ISBN 0-201-02988-X.
23. Kaner, Cem; Falk, Jack; Nguyen, Hung Quoc. "Testing Computer Software", Second Edition, Thomson Computer Press, 1993, ISBN 1-85032-847-1.
24. National Institute of Standards and Technology. "Advanced Encryption Standard (AES) Development Effort", http://csrc.nist.gov/encryption/aes/
25. Papadimitriou, Christos H. "Computational Complexity", Addison- Wesley Publishing Company, 1994, ISBN 0-201-53082-1.
26. Schneier, Bruce. "Applied Cryptography", Second edition, John Wiley & Sons, 1996, ISBN 0-471-11709-9.
27. Shannon, Claude Elmwood. "Communication Theory of Secrecy Systems", Bell System Technical Journal, Vol 28, 1949, pp 656-715.
28. Turing, Alan Mathison. "On computable numbers, with an application to the Entscheidungsproblem", Proc. Lond. Math. Soc. 42, 230-265 1936; received May 25,1936, Appendix added August 28; A correction, ibid., 43 pp 544-546, 1937.
29. Wolfram, Stephen. "Computer Software in Science and Mathematics", Scientific American, Vol 251 pp 188-203, Sept 1984.
30. Wolfram, Stephen. "Origins of Randomness in Physical Systems", Physical Review Letters, Vol

55, pp 449-452, 29 July 1985.
31. E. Cole, R. Krutz and J. W. Conley, Network Security Bible, Wiley Publishing Inc, 2005.
32. Menezes, V. Oosrschot and A. Vanstone, Handbook on Applied Cryptography, CRC Press Inc., NY, USA, 2000.
33. D. Stinson, Cryptography Theory and Practice, CRC Press Inc., NY, USA, 1995.
34. G. Blelloch, Introduction to Cryptography, online: http://www.cs.cmu.edu/afs/cs/project/pscicoguyb/realworld/crypto.ps , 2000, accessed on Sept. 2011.
35. G. Carter, E. Dawsony and L. Nielseny, Key Schedule Classification of the AES Candidates, in the Proceedings of the end AES Conference, Rome, Italy, 1999.
36. J. Dray, Report on the NIST Java AES Candidate Algorithm Analysis, online: http://csrc.nist.gov/encryption/aes/round/r1-java.pdf , 1999, accessed on Sept. 2011.
37. J. Dray, NIST Performance Analysis of the Field Round Java AES Candidates, online: http://csrc.nist.gov/encryption/aes/roubd2/conf2/papers/8-jdray.pdf , 2000, accessed on Sept. 2011.
38. J. Nakahara, B. Preneel and J. Vandewalle, Square Attack on Extended Rijndael Block Copher, COSIC Technology Report, 2002.
39. D. Baudran, H. Gilbert, L. Granboulan, H. Handschun, A. Joux, P. Nguyae, F. Noilhan, O. Poincheva, T. Pornin, G. Poupard, J. Stern and S. Vaudenay, Report on the AES Candidates, in Proceedings of the 2nd ASE Conference, Rome, Italy, 1999.
40. T. Verhoeff, Encryptography, online: http://www.pa.win.tue.nl/wstomv/software/AES Rijndael/rijndaeltest.pas , 2001, accessed on Sept. 2011.
41. P. Pfleeger and S. L. Pfleeger, Security in Computing, 3rd edition, Prentice-Hall, 2003.
42. William Stallings, Cryptography and Network Security, 4th edition, Prentice- Hall, 2005.
43. Behrouz. A. Forouzan, Data Communications and Networking, 4th edition, McGraw- Hill, 2007.
44. Wikipedia Website, online: http://en.wikipedia.org , accessed on Sept., 2011.
A. H. Omari, B. M. Al-Kasasbeh, R. E. Al-Qutaish and M. I. Al- Muhairat, A New Cryptographic Algorithm for the Real-Time Applications, in the Proceedings of the 7th International Conference on Information Security and Privacy - (ISP'08), Cairo, Egypt, from Dec. 29 - Dec. 31, 2008.

***

[1] B.Tech Student, Department of CSE, Institute of Technology, Guru Ghasidas University, Bilaspur, C.G.
[2] Assistant Professor, Department of CSE, Institute of Technology, Guru Ghasidas University, Bilaspur, C.G.
[3] Assistant Professor, Department of CSE, Institute of Technology, Guru Ghasidas University, Bilaspur, C.G.
[4] B.Tech Student, Department of CSE, Doon College of Engineering & Technology, Dehradun, Uttarakhand
E-mail address: rsinha.u64@gmail.com