

Simple Encryption/Decryption Application

Majdi Al-qdah

*Faculty of Information Technology
Multimedia University
Cyberjaya, 63100, Malaysia*

majdi.qdah@mmu.edu.my

Lin Yi Hui

*Faculty of Information Technology
Multimedia University
Cyberjaya, 63100, Malaysia*

Abstract

This paper presents an Encryption/Decryption application that is able to work with any type of file; for example: image files, data files, documentation files...etc. The method of encryption is simple enough yet powerful enough to fit the needs of students and staff in a small institution. The application uses simple key generation method of random number generation and combination. The final encryption is a binary one performed through rotation of bits and XOR operation applied on each block of data in any file using a symmetric decimal key. The key generation and Encryption are all done by the system itself after clicking the encryption button with transparency to the user. The same encryption key is also used to decrypt the encrypted binary file.

Keywords: Encryption, Decryption, Key, Rotation, XOR.

1. 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 [1]. The purpose of Encryption is to prevent unauthorized parties from viewing or modifying the data [2]. 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 ciphertext, which is representation of the original data in a difference form [2].

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 [4, 5].

One simple and good way to encrypt data is through rotation of bits or sometimes called bit shifting. But, rotation of bits is more advanced than simple bit shifting. In rotation of bits operation, the bits are moved, or shifted, to the left or to the right. The different kinds of shifts typically differ in what they do with the bits [6]. Another way is to perform logical operation on the bits of the file such as XOR operation. The idea behind XOR Encryption is that it is impossible to reverse the operation without knowing the initial value of one of the two arguments [7].

2. PRELIMINARIES

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 [8]. One of the first types of Encryption was made by Julius Caesar. [9]. in his system, Caesar wrote B instead of A and C instead of B – so to a sentence “ABC” will be written in “BCD” [6].

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 security[10].

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 Executables programs [11].

Neekprotect is a 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 a files and the key will be used again when some one else trying to open the files been decrypted through decryption on the certain files [12].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 on a similar encryption technique that uses binary rotation of bits with XOR logical operation using a custom made encryption key that operates on any type of a file.

3. METHODOLOGY

The main feature of the encryption/decryption program implementation is the generation of the encryption key. Other features are related to the design of the GUI, progress of encryption details, and user notification of the status of encryption.

3.1 Key Generation

A symmetric Encryption key is used for this application, which means the same key is shared for both Encryption and decryption. A copy of the generated key is saved in a file named .ekf during the Encryption process and the same key is used as the decryption key to retrieve the encrypted file. The technique of generating the key uses two methods: random number generation and combination. First, a long number with only digit values called A is generated. Then another long number with character values called B is generated. The size of B is twice the size of A. Then an insertion operation is performed such that each digit of A is inserted after two characters of B. The result of the insertion is called C. Then another only digit number called D is randomly generated. C is combined with D by placing alternately one character or digit from C after a character or a digit from D. The result of the combination is a relatively strong key. Then, an odd and even partitioning is performed on the key. The position of each character in the key decides

it to be an even or an odd character. For example, the character at position 0 is an even one while the character at position 1 is an odd character. Similarly, position 2 is an even position while position 3 is odd one. The even part of the key is combined together and the odd part of the key is combined together. Finally, the two parts of the key are joined as an even part followed by an odd part to produce one final encryption key. Since the final key is a key that consists of all characters, another key with only ASCII values of each character is obtained. The result is a very long decimal key. Figure 1 shows the complete key generation process.

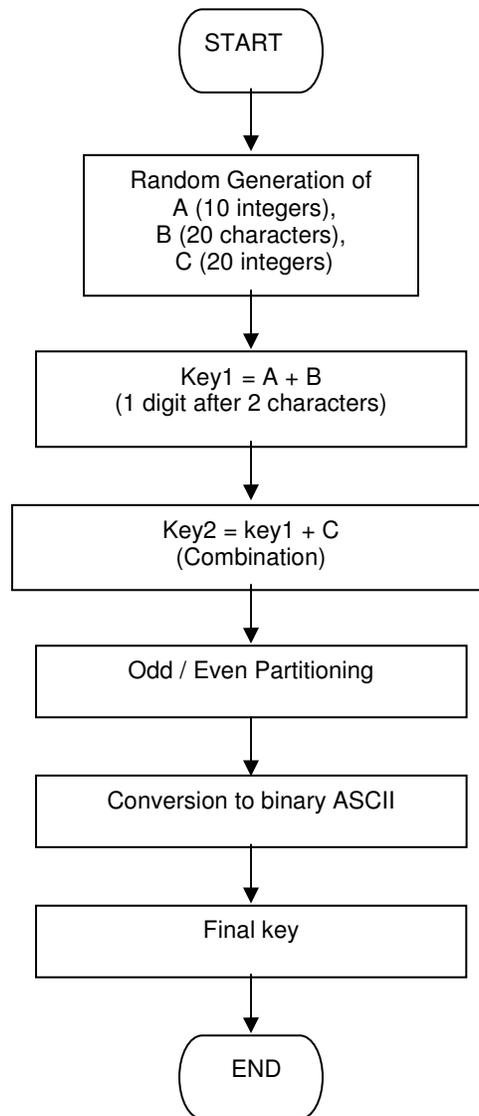


FIGURE 1: Key Generation Flow Chart.

3.2 Encryption Rules

For the Encryption method, a single digit in the decimal ASCII representation will decide which Encryption method is to be applied to the single a binary block in a file.

- I. 0 in the key means a rotation of bit to the left is performed and the next integer to 0 in the ASCII code decides how many bits the block will be moved to the left.
- II. 1 in the key means a rotation of bit to the right is performed and the next integer to 1 in the ASCII code decides how many bits the block will be moved to the right.
- III. 2 means the block will be passed to an XOR Encryption to be performed with a binary block from the file.
- IV. Else, all other numbers in the key like 3,4,5,6,7,8,9 are ignored.

The above rules are summarized in the encryption chart shown in figure 2.

3.3 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.

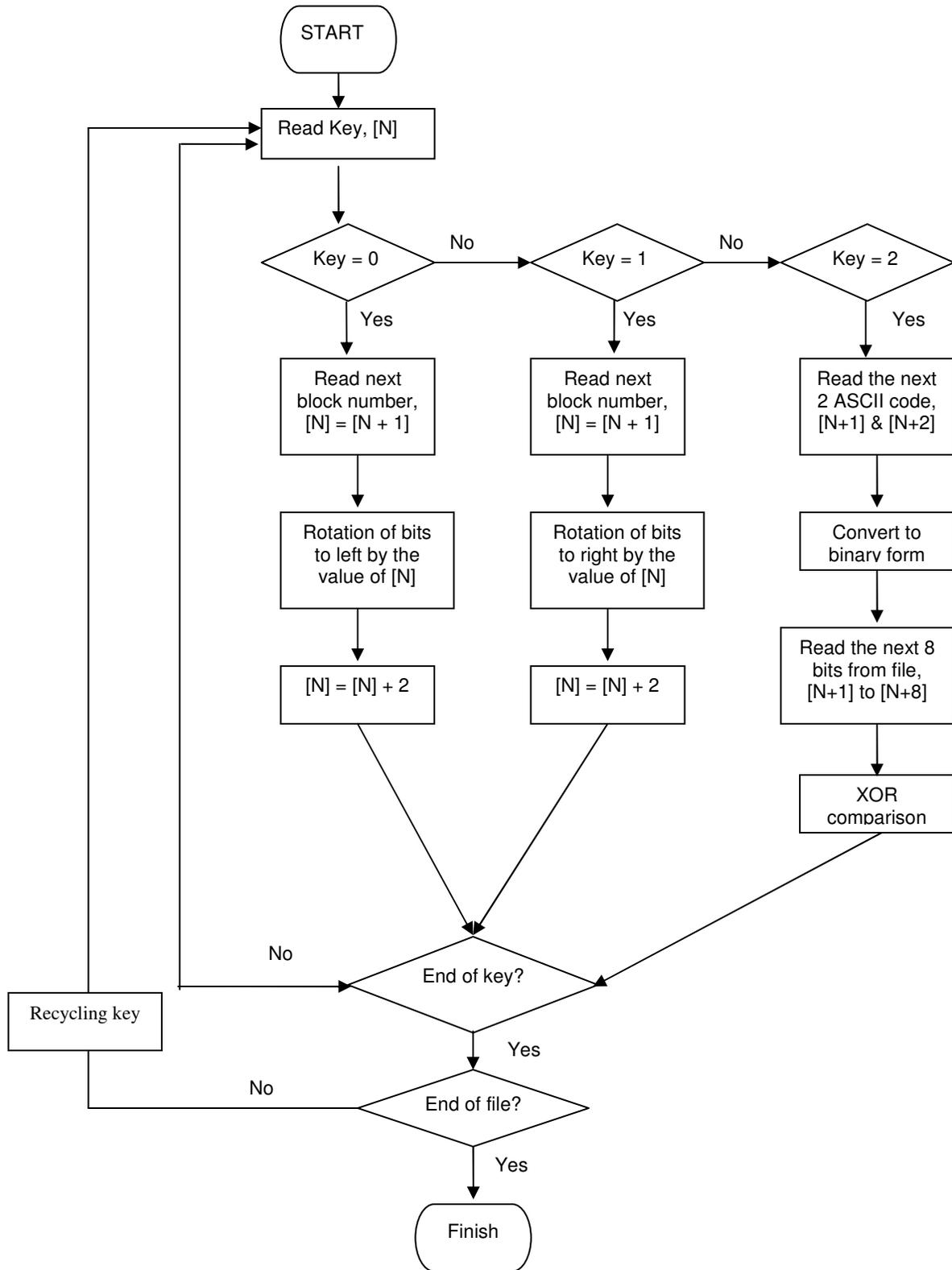


FIGURE 2: Encryption Flow Chart.

4. RESULTS

The interface of the application is simple enough to be used by any user. Figure 3 shows the interface with the encryption and a decryption buttons. The encryption is performed simply by choosing any file while decryption is executed by choosing an encrypted file with an appropriate key. Figure 4 and 5 shows a successful encryption and decryption, respectively.



FIGURE 3: shows the Encryption/Decryption interface.



FIGURE 4: shows a successful encryption and the encryption progress bar.



FIGURE 5: successful decryption and the decryption progress bar.

5. APPLICATION TESTING & PERFORMANCE CHECKING

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 and an encrypted file. Table 1 shows the testing of different types and sizes of files. Figure 6 shows the encryption time for six different types of files that are of the same size of 5 mb. 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 university 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	9/27/45	100%
Image	1/ 3/ 5	10/26/44	100%
Audio/ Video	1/ 3/ 5	18/28/45	100%
Zipped	1/ 3/ 5	10/25/44	100%
Exe.	1/ 3/ 5	12/27/48	100%

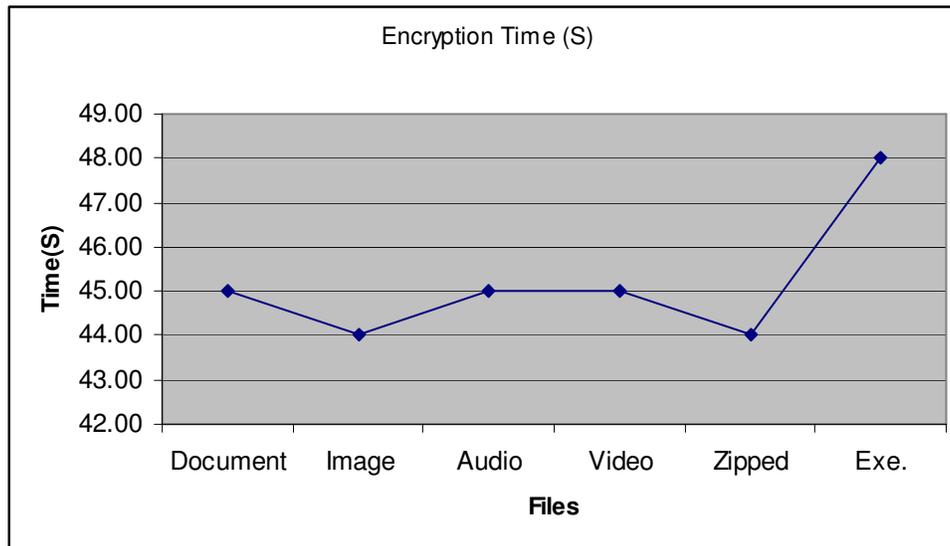


FIGURE 6: Graphical view of the encryption time for different types of files.

6. CONSLUION & FUTURE WORK

A new simple tool has been created, which is targeted for use inside of a small institution such as a small university for lecturers' daily use of sending exam files and sensitive material such that the material can be encrypted and the file is sent in one e-mail while the encryption key is sent in another e-mail or via any secure communication channel. The encryption application developed and described in this paper might not be comparable to well-known encryption algorithms but its

simplicity and availability proves that tools can be developed that fit the needs of an institution without resorting to purchasing expensive software from the market. For future enhancement to this application public key encryption can be applied where two keys can be generated: one to encrypt a file using the public key and another private key to decrypt it. Also, other more advanced encryption operations can be included to enhance the security of the application so that it can be used to encrypt more sensitive administrative material in an institution.

7. REFERENCES

- [1] Wikipedia, "*Encryption*", <http://en.wikipedia.org/wiki/Encryption>, modified on 13 December 2006.
- [2] 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
- [3] 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.
- [4] 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
- [5] IBM. "*The Data Encryption Standard (DES) and its strength against attacks*". IBM Journal of Research and Development, Vol. 38, PP. 243-250. 1994
- [6] Wikipedia, "*Bitwise operation*", http://en.wikipedia.org/wiki/Bitwise_operation, last modified on 10 December 2006.
- [7] Andy Wilson, "*Tips and Tricks: XOR Encryption*" <http://www.andyw.com/director/xor.asp>, 1998.
- [8] 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
- [9] Microsoft, "*Encrypting File System for Windows 2000*", <http://www.microsoft.com/windows2000/techinfo/howitworks/security/encrypt.asp>, 1998.
- [10] Dariusz Stanislawek, "*Free Software copyright 1997 - 2006*" <http://members.ozemail.com.au/~nulifetv/freezip/freeware>
- [11] NeoCrypt, "*NeoCrypt File Protection Utility*", <http://sourceforge.net/projects/neocrypt>
- [12] Vivek Thakur, "*NeekProtect*", <http://neekprotect.sourceforge.net>, 2006.
- [13] Artur Ekert, Carolina Moura Alves, Ajay Gopinathan, "History of Cryptography".
- [14] Cryptomathic, "E-SECURITY DICTIONARY", <http://www.cryptomathic.com/labs/techdict.html>, 2003.