http://ejournal.seaninstitute.or.id/index.php/InfoSains
Jurnal Info Sains : Informatikan dan Sains, Volume 10, No 01 Maret 2020
E-ISSN.2797-7889, P-ISSN.2089-3329

Use of Electronic Code Book (Ecb) Algorithm in File Security

Pilipus Tarigan
Teknik Informatika, STIKOM Medan, Indonesia
Email: philipus97@gmail.com

Keywords

Cryptography
ECB Method

Abstract. Electronic communication such as sms, e-mail, chat, web, e-banking is a commonly used communication tool today. To prevent the misuse of such data by other parties, a good data security system is required. Cryptography is a way of securing data that aims to maintain the confidentiality of the information contained in the data, so that the data information can not be known by unauthorized parties. In maintaining the confidentiality of information, cryptography encodes plaintext data into an unrecognizable form of password data (chipertext), and although others later obtain the data, it cannot understand its contents

1. INTRODUCTION

Electronic communication such as sms, e-mail, chat, web, e-banking is a commonly used communication tool today. Data flows through communication networks and from some personal data that should not be known by others. Meanwhile, data interception on communication networks is a frequent occurrence [1].

This will ultimately result in misuse of data by unauthorized parties. To prevent the misuse of such data by other parties, a good data security system is required. Cryptography is a way of securing data that aims to maintain the confidentiality of the information contained in the data, so that the information cannot be known by unauthorized parties [2]–[6]. In maintaining the confidentiality of information, cryptography encodesplaintextdata into a form ofciphertextthat cannot be recognized, and although later others obtain the data, it cannot understand its contents [7].

Electronic Code Book is the latest fast hash function, designed to run quickly on modern computers, and specifically for computers based on 64 bits (such as DEC-Alpha),and also this algorithm is still no slower than other hash functions suggested in 32-bit machines (although now no longer, since MD5 and SHA-1 have been found to be weaknesses) [8], [9].

2. METHOD

The stages in this research are as follows;

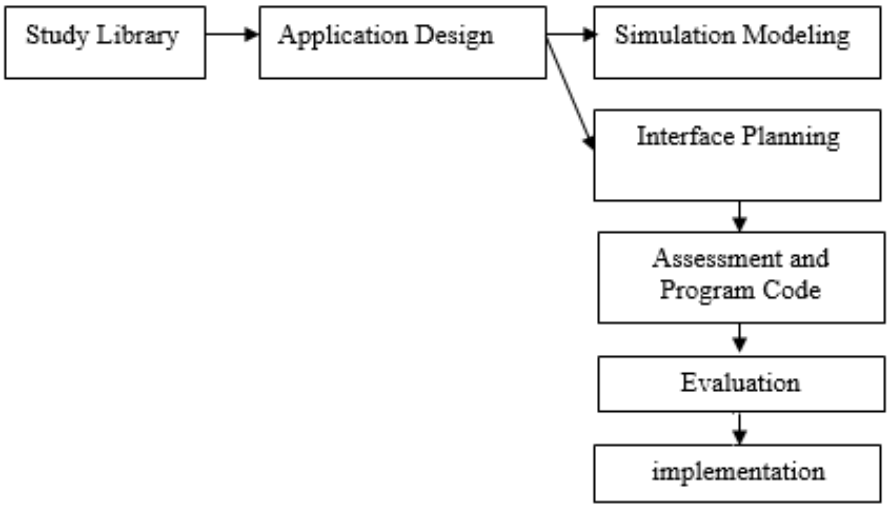

Figure 1. Research Work Steps

Encryption and decryption of a random nature is very suitable to be implemented with *the ECB (Electronic Code Book) mode* block chiper algorithm, provided that each *record* consists of the same number of discrete blocks [10], [11]. ECB mode is suitable for encrypting randomly accessed *files* because each *plaintext* block is independently encrypted. Even if *the ECB* mode is done with a parallel processor, then each processor can encrypt or decrypt different ecb plaintext blocks that will be used to encrypt or decrypt data is the *ECB* that has been modified so that the resulting chipertext block is not the same even though it encrypts the same plaintext [8], [10]–[12]. This is to avoid the often repetitive plaintext part, which is one of the weaknesses of *ECB fashion.* In this mode, each block of plaintext is encrypted individually and independently [2], [3]. Mathematically, encryption with *ECB* mode is expressed as

$$C_i = E_K(P_i)$$

and decryption as

$$Pi = DK(Ci)$$

Which in this case, Pi and Ci block plaintext and ciphertext respectively i. Figure 2 shows the encryption of two plaintext blocks, P1 and P2 in *ECB mode,* in which case E states an encryption function that encrypts plaintext blocks using the Kkey.

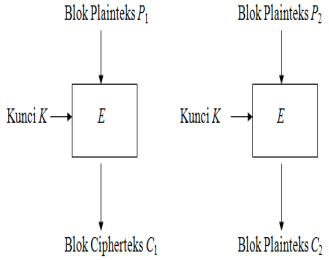

Figure 2 : ECB Operation Mode

Supposing plaintext (in binary) is 1010001000111010101001For plaintext becomes blocks that are 4 bits in size:
1010 0010 0011 1010 1001 or in *HEX* notation is A23A9. Suppose the key (K) used is (the length is also 4 bits) 1011 or in HEX notation is B. Suppose the simple (but weak) E encryption function is to*XOR*-kan block plaintext Pi with K, then slide wrapping bits of Pi $\oplus$ K one position to the left. The encryption process for each block is described as follows:

1010 0010 0011 1010 1001
1011 1011 1011 1011 1011

XOR : 0001 1001 1000 0001 0010
Slide : 0010 0011 0001 0010 0100
In notation HEX : 23124

So, the result of plaintext encryption
1010001000111010101001 (A23A9 in *HEX*notation)
be
0010001100010010100 (23124 in *HEX*notation)
Note that the same plaintext block is always encrypted into the same (or identical) ciphertext block. In example 1 above, block 1010 appears twice and is always encrypted to 0010. The word*"code book"* within the *ECB* arises from the fact that because the same plaintext blocks are always encrypted into the same ciphertext block, it is theoretically possible to create a plaintext code book and

corresponding ciphertext, however, the larger the block size, the larger the code book size. Suppose if the block is 64 bits in size, then the code book consists of 264 – 1 piece of code *(entry)*, which means it is too large to store. After all, each key has a different code book.

3. RESULTS AND DISCUSSION

The following is the Encryption Process conducted by the ECB method.

Plaintext = Keyso
Kunci = B
Kardo ➔ 01101011 01100001 01110010 01100100 01101111
B ➔ 1011
Plaintext Encryption :

0110 1011 0110 0001 0111 0010 0110 0100 0110 1111
1011 1011 1011 1011 1011 1011 1011 1011 1011 1011 ⊕

1101 0000 1101 1001 1100 1001 1101 1111 1101 0100
1010 0001 1011 0011 1001 0011 1011 1111 1010 1001

Cipherteks : A1B393BFA9

The decryption process using ecb method is as follows:

Cipherteks = A1B393BFA9
Binary = 1010 0001 1011 0011 1001 0011 1011 1111 1010 1001
key = B
Plaintext ➔ 1010 0001 1011 0011 1001 0011 1011 1111 1010 1001
B ➔ 1011

Plaintext Decryption :

1010 0001 1011 0011 1001 0011 1011 1111 1010 1001
1011 1011 1011 1011 1011 1011 1011 1011 1011 1011 ⊕

1101 0000 1101 1001 1100 1001 1101 1111 1101 0100
0110 1011 0110 0001 0111 0010 0110 0100 0110 1111

Plainteks = Keyso

Testing the program is used to try the application that has been designed whether it is in accordance with the desired or not, note the image below:

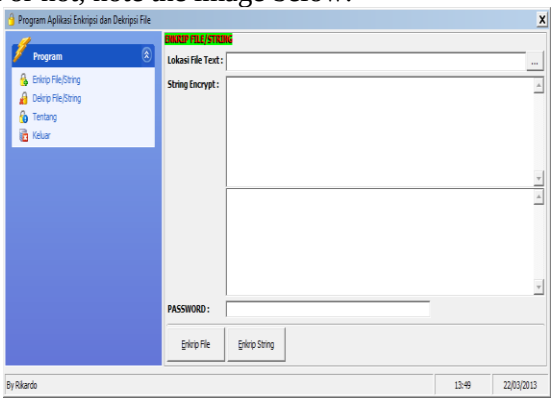

Figure 3 : Main *Program*

The picture above is the main program that is done for the process of encryption and decryption of *files*, to perform the encryption process must certainly take the *file* to be encrypted, the *file* can be a

Microsoft *Word file* or *text file*,to retrieve the *file*simply by pressing the button in the locationof the *file* and looking for the *file*so that the result is as below:

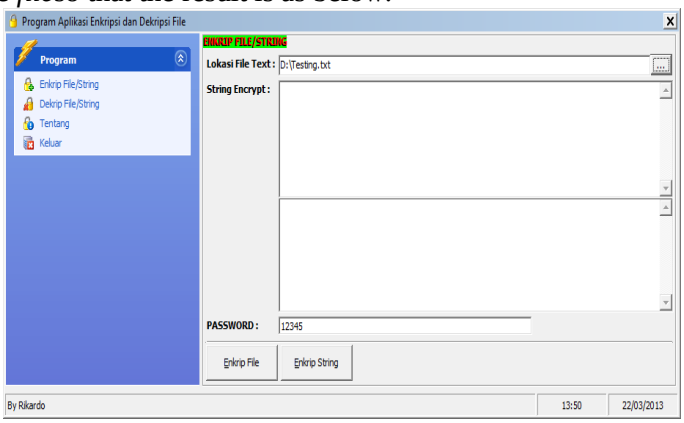

Figure 4. Encrypted File Location

After the encryption process is complete then the next step is to perform the decryption process, the decryption process is done to restore the encrypted *file* to its original *file* form, for the decryption process pay attention to the image below:

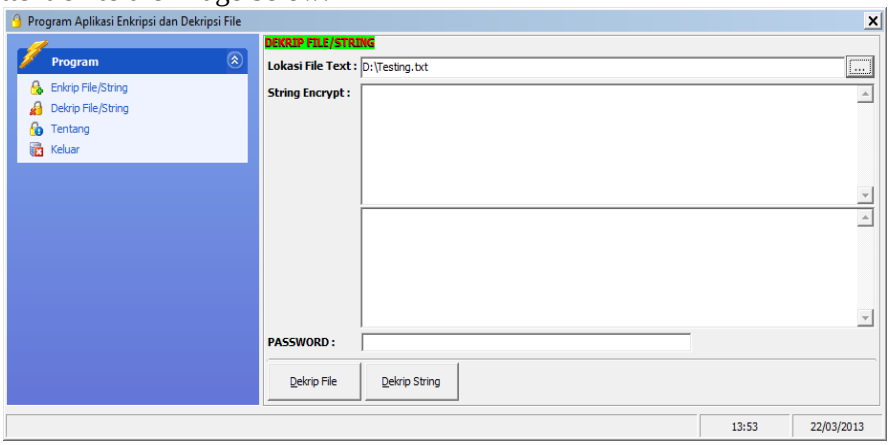

Figure 5 Decryption Process

4. CONCLUSION

After completing the design of encryption and decryption software, it is concluded that the Program can run well in accordance with encryption and decryption procedures using *ECB algorithms and* cryptographic programs designed with an attractive AND easy *GUI* in operation. ECB mode is also suitable for encrypting randomly accessed archives*(files)*,such as database archives. If the database is encrypted in *ECB*mode, then any *record* can be encrypted or decrypted independently of other *records* (assuming each *record* consists of the same number of discrete blocks). The weakness of *ECB* mode is that the plaintext section often repeats (so there are the same blocks of plaintext), so the encryption results in the same chipertext block. Plaintext parts are often repeated e.g. words such as (in Indonesian) *and, which, this, it* and so on.

REFERENCE

[1] A. Zieleniewska, S. R. Harper, D. P. Arnold, and D. M. Guldi, “Ground State versus Excited State: Discrepancy in Electronic Communication in a Series of meso–meso Two-Atom-Bridged Diporphyrins,” *Chem. - A Eur. J.*, vol. 24, no. 12, 2018, doi: 10.1002/chem.201705938.

[2] D. C. Prakoso and Y. Prayudi, “Model Enkripsi XML Pada Output DFXML untuk Pengamanan Metadata Bukti Digital,” *JUMANJI (Jurnal Masy. Inform. Unjani)*, vol. 1, no. 1, 2018, doi:

10.26874/jumanji.v1i1.8.
[3] M. A. A and A. Suprianto, "Penggunaan Algorithma AES-RIJNDAEL Pada Sistem Enkripsi Dan Dekripsi Untuk Komunikasi Data," *Sainstech J. Penelit. dan Pengkaj. Sains dan Teknol.*, vol. 25, no. 2, 2018, doi: 10.37277/stch.v25i2.94.
[4] W. M. Rahmawati and F. Liantoni, "Penggunaan Arnold Cat Map Dan Beta Chaotic Map Pada Enkripsi Data Citra," *J. ELTIKOM*, vol. 2, no. 2, 2018, doi: 10.31961/eltikom.v2i2.85.
[5] S. Retno and N. Hasdyna, "ANALISIS KINERJA ALGORITMA HONEY ENCRYPTION DAN ALGORITMA BLOWFISH PADA PROSES ENKRIPSI DAN DEKRIPSI," *TECHSI - J. Tek. Inform.*, vol. 10, no. 2, 2018, doi: 10.29103/techsi.v10i2.858.
[6] A. Prameshwari and N. P. Sastra, "Implementasi Algoritma Advanced Encryption Standard (AES) 128 Untuk Enkripsi dan Dekripsi File Dokumen," *Eksplora Inform.*, vol. 8, no. 1, 2018, doi: 10.30864/eksplora.v8i1.139.
[7] W. Zhai, "Design and application of a remote electronic communication teaching system in a network environment," *Int. J. Emerg. Technol. Learn.*, vol. 13, no. 4, 2018, doi: 10.3991/ijet.v13i04.8480.
[8] M. Asti, A. Kamsyakawuni, and K. A. Santoso, "PENGAMANAN IMAGE DENGAN MODIFIKASI ALGORITMA ELECTRONIC CODE BOOK (ECB)," *Maj. Ilm. Mat. dan Stat.*, vol. 18, no. 2, 2018, doi: 10.19184/mims.v18i2.17252.
[9] J. C. Das and D. De, "Qca based secure nanocommunication block cipher design based on electronic code book," *Malaysian J. Comput. Sci.*, vol. 31, no. 2, 2018, doi: 10.22452/mjcs.vol31no2.3.
[10] Y. S. Fatmala, A. Kusyanti, and M. Data, "Implementasi Algoritme Speck untuk Enkripsi dan Dekripsi pada QR Code," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 2, no. 12, 2018.
[11] D. A. Meko, "Perbandingan Algoritma DES, AES, IDEA Dan Blowfish dalam Enkripsi dan Dekripsi Data," *J. Teknol. Terpadu*, vol. 4, no. 1, 2018.
[12] K. R. Ilaga and C. A. Sari, "Analysis of Secure Image Crypto-Stegano Based on Electronic Code Book and Least Significant Bit," *J. Appl. Intell. Syst.*, vol. 3, no. 1, 2018.