2009 International Conference on Machine Learning and Computing
IPCSIT vol.3 (2011) © (2011) IACSIT Press, Singapore

Correcting Noisy Ciphers in CBC mode

Nabil Mirza [1], Ziad Osman [1], Rached Zantout [1], Mohamed El-Sayed [2]

[1] Electrical Engineering Department, Beirut Arab University, Debbieh, Lebanon
[2] Electrical Engineering Department, Alexandria University, Egypt

Abstract. Correction of noisy cipher is a challenging task. Previous work has been done on correcting noisy ciphers using AES in ECB mode. In this paper, error detection and correction is done at the receiver end, without any changes to the encryption algorithm that uses AES in CBC mode. A property of CBC is that noise affecting one encrypted block will affect the corresponding decrypted block and its neighbor. This property is exploited to identify the noise vector that contaminated the encrypted message. For a specific corpus the space of all possible messages (Datagram book) is modeled. The noise-free neighbor of the noise contaminated decrypted data is used to generate multiple possible candidates. The candidates are then compared to the Datagram book to identify the noise-free block.

Keywords: CBC, Error Correction.

1. Introduction

The importance of the problem of error correction of encrypted data at the receiver end stems from the wide range of areas in which this problem exists. One of the areas in which error correction is important is secure file storage systems where the user encrypts a file, stores the encrypted version, and destroys the original [1, 2]. In block cipher cryptosystems with good diffusion, flipping an input bit should change each output bit with a probability of one half [3]. This means that a one bit error in the encrypted block will alter nearly half the bits in the corresponding decrypted block. In block ciphers that use the ECB mode, error correction is done by exploiting a property of the encryption algorithm that any encrypted block has a large Hamming distance from any other encrypted block and in some cases relying on information about the noise-free neighbors of the contaminated block [4, 5]. AES operates with 128 bit block sizes. Messages longer than AES's input block size are handled using CBC mode of operation. In CBC each block is first xored with the previously encrypted block. Then the result is encrypted with the symmetric key. This means that each encrypted block is dependent on the previous block. The properties used in [4] and [5] cannot be exploited in block ciphers using the CBC mode. In [4] the error correction relies on the encrypted noisy message. In [5] the correction relies on information from neighboring blocks which are not contaminated by noise in ECB mode.

In this paper, ciphertexts are assumed to have resulted from encrypting plaintexts using AES with a symmetric key of 256 bits. A closed corpus is used to generate a Datagram Book which is the collection of all possible 8 character strings that occur in the corpus. Noise that contaminates one encrypted block generates two decrypted noisy blocks, the block corresponding to the encrypted noisy block (block A) and its immediate neighbor (block B). Block A is contaminated beyond recognition while block B has the same bit positions in error as the contaminated encrypted block. The immediate neighbor of block B is used with the Datagram book in order to generate a list of possible corrections. Each member of the list is used to predict the noise that contaminated the original encrypted block. One of the possible corrections leads to a combination of corrected blocks A and B that exist in the Datagram book.

In part 2 the method used to correct the noisy cipher is outlined. In part 3 simulations are presented proving the validity of the search criteria. Part 4 concludes the paper with a summary of the work and

recommendations for further research.

2. Method Description

The Datagram book used is a listing of all unique blocks in a closed corpus. It contains 33632 unique entries. Figure 1 represents a schematic diagram of the receiver end that describes the approach for error detection and correction of noisy CBC cipher.

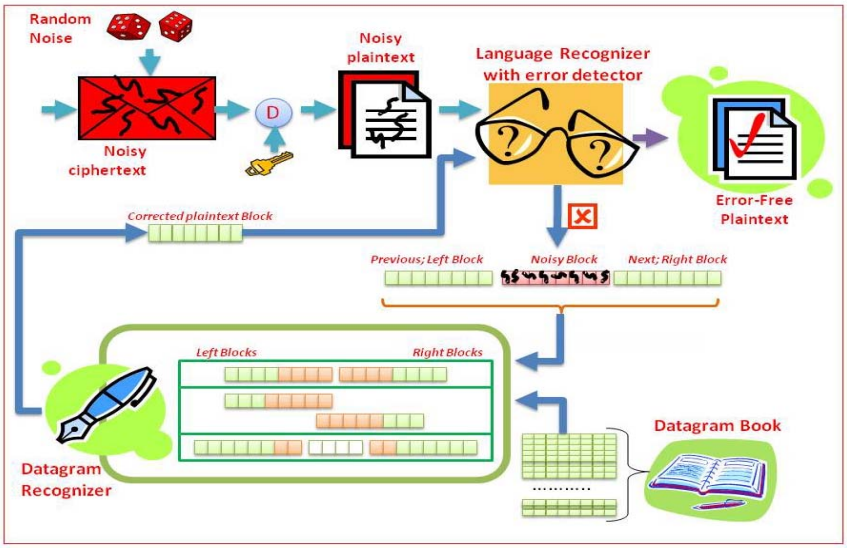

Figure 1. Error detection/correction of noisy cipher

The Language Recognizer with error detector module detects noisy plaintext blocks. It receives a block of 8 characters and checks whether any of those characters is invalid. The block containing invalid characters (Block A) and its successor (Block B) will be contaminated by noise. The successor of Block B (Block C) is used to guide the search for the noise-free block in the datagram book, as shown in Figure 1. Bits 1 through 7 of block C are used with the datagram book to extract nominees for bit 8 in Block B. Bits 1 through 6 of Block C and each of the nominees for bit 8 in Block B are then used with the datagram book to generate nominees for bit 7 of Block B. This is repeated with a decreasing number of bits from Block C and increasing number of nominees for Block B. Each resulting nominee is xored with Block B to generate an error vector. The error vectors are then xored with the noisy cipher of block A to generate a list of possible noise-free ciphers. This list is then decrypted and compared with the datagram book to come up with the only possible solution.

3. Results

Software was written in Visual Basic. Net v9 to simulate the method described in part 2. Figure 2 shows three blocks A, B and C. The first row shows the encrypted blocks that were obtained by using AES in CBC mode on the blocks shown in the subsequent rows. The second row shows the actual characters of each block, while the third row shows their Unicode representation in hex. The fourth row shows the actual text as it appears in the corpus for the whole message.

	Block A	Block B	Block C
Cipher blocks (Hex)	C7A3C58C5E9DD615A0EEF74F83FFF915	4E9D9BB53C3BAECDA40045A53AAD46C:	0EB405C1DB5674B2BC81D1EACB65AA82
Plaintext blocks (Unicode)	اَبٌ أَل	يمٌ يم	اَكانْ
Plaintext blocks (Hex)	064E0627062806 4C00200623064E0644	0650064A0645064C0020062806500645	064E062700200643064E06270646064F
Plaintext (Unicode)	اَبٌ أَلِيمٌ بِمَا كَانْ		

Figure 2. Decryption of noisy free blocks in CBC mode

Figure 3 shows the same three blocks when the cipher of block A was hit by a 1 bit noise (i.e. 1 bit of the cipher of block A was changed). When decrypted, block A was hit by so much error that it was totally

different from the noise-free decrypted version in figure 2. However, the decrypted block B was hit by only a 1-bit error.

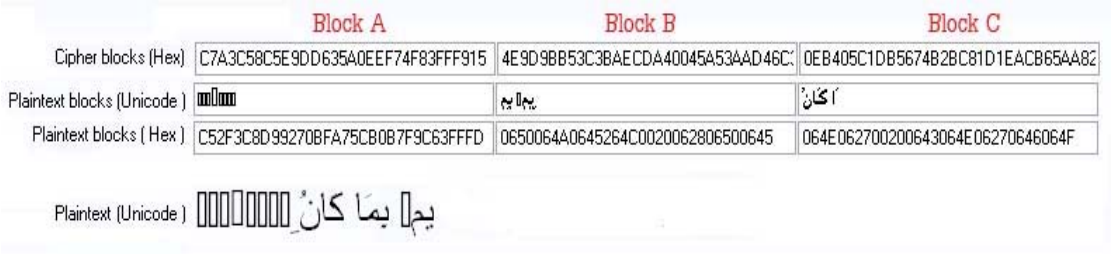

Figure 3. Case of a 1-bit error noise

Figure 4 shows the same three blocks hit with 10 bit noise. The procedure described in section II was applied to generate the blocks shown under "Plaintext Candidates". 158 candidates were produced in the example given in Figure 4. Each block was then xored with the plaintext block B to generate the noise vectors shown under "Noise Vector Candidates". Each noise vector was then xored with the noisy cipher of block A to generate the list under "Cipher Blocks Candidates". Each of those candidates was then decrypted to produce new block A and block B candidates. The results are shown under "Decrypted cipher text (Block A)/Plaintext candidates (Block B)". Only one candidate (id 8804 in Figure 4) was found in the datagram book which was the correct choice.

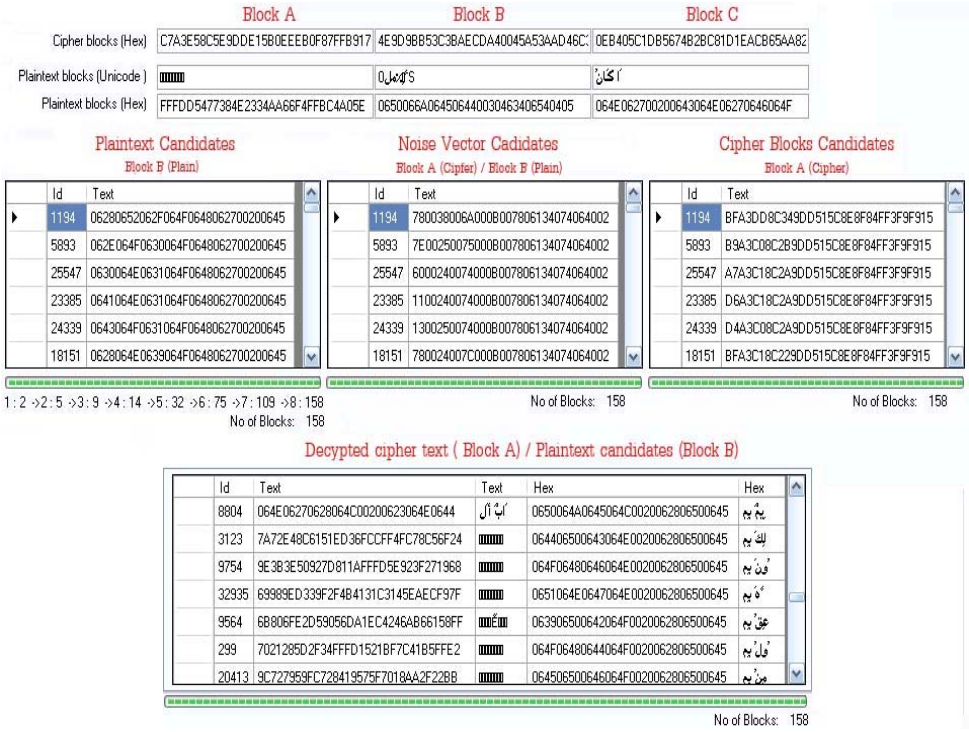

Figure 4. Procedure for a 10-bit error noise

Figures 5 and 6 show the same blocks hit by 70 and 128 noise bit. In all three cases, the number of plaintext candidates was 158 since this number depends on the characters in Block C rather than on the error hitting the cipher of block A. This proves that regardless of the number of error bits, this method will always find the correct answer.

4. Conclusion

In this paper, a new method for correcting errors in CBC block ciphers was presented. It exploits the error propagation property of CBC. This method is applicable only to closed corpora such as secure file storage systems. More research is needed to make the method applicable to open corpora through the use of Natural Language Processing and the Statistical properties of the plaintext language.

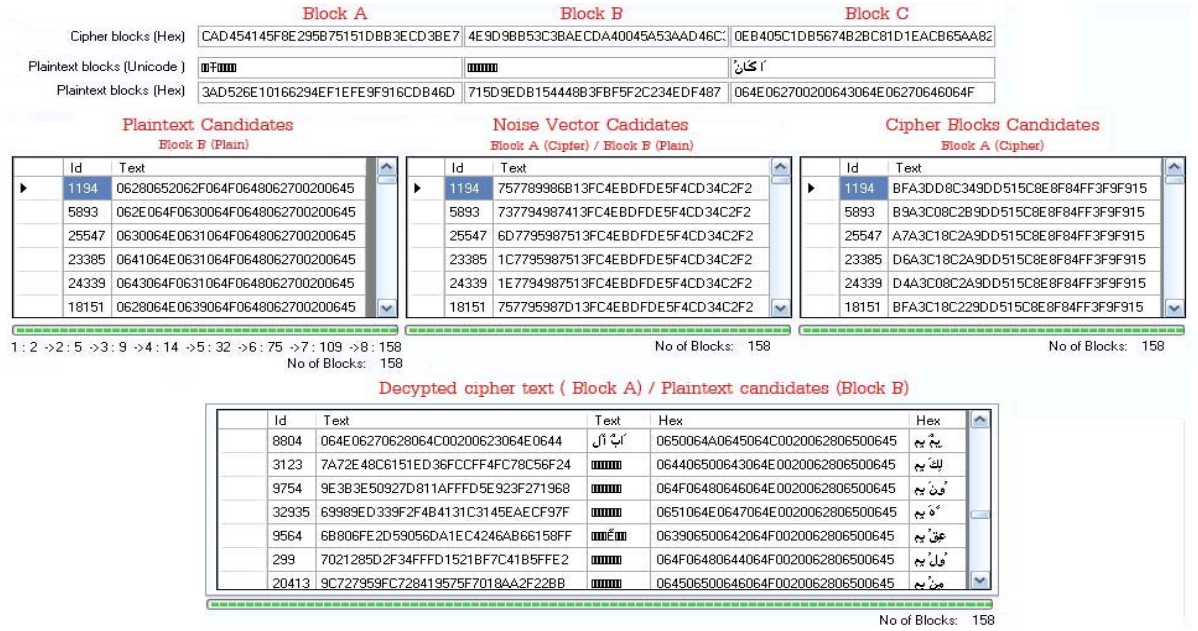

Figure 5. Procedure for a 70-bit error noise

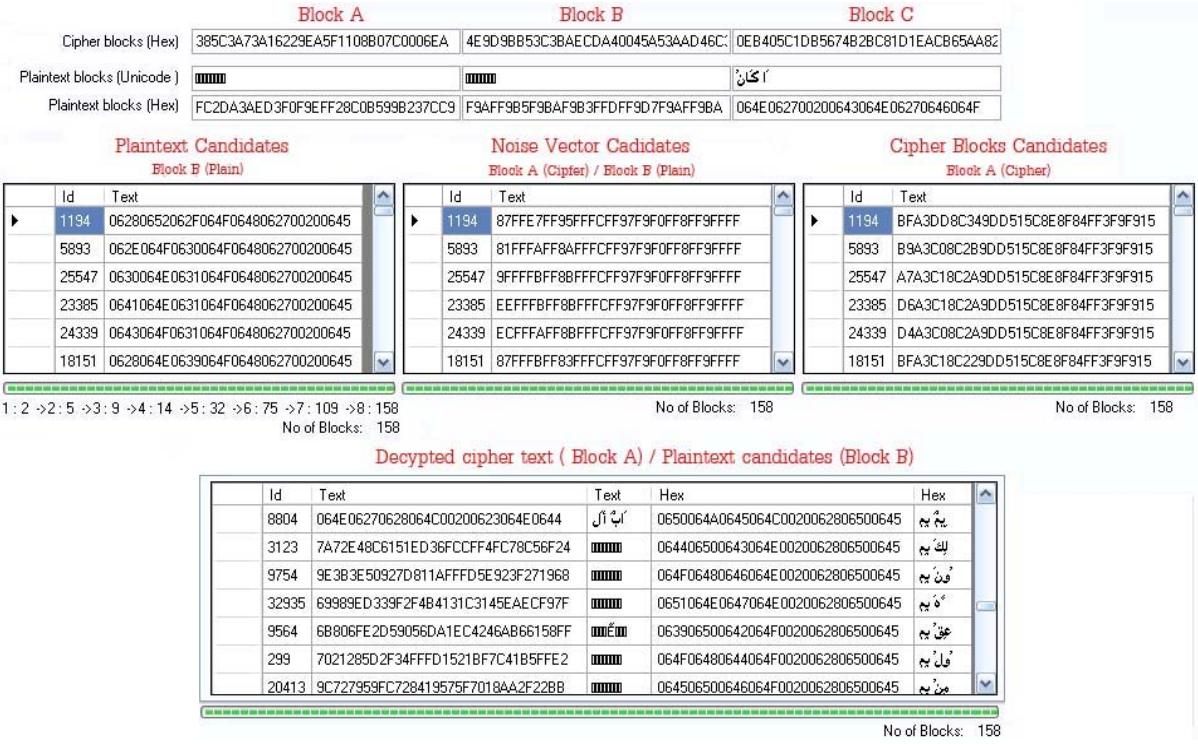

Figure 6. Procedure for a 128-bit error noise

5. References

[1] Agnew, G.B. Cryptography systems using redundancy. *IEEE Transactions on Information theory* 36 (1990) 31—38

[2] U, J.N.P.S.B.M.S.K.R.A. Reliable transmission and storage of medical images with patient information using error control codes. In: *Proceedings of the IEEE INDICON*. (2004) 147—150

[3] Bertoni, G., Breveglieri, L., Koren, I., Maistri, P., Piuri, V. Error analysis and detection procedures for a hardware implementation of the advanced encryption standard. *IEEE Trans.* Comput. 52 (2003) 492—505

[4] Nabil Mirza, Ziad Osman, Rached Zantout. A Novel Approach for Correcting Noisy AES Ciphers, ISP09, *Orlando, Florida*, July 2009.

[5] Nabil Mirza, Ziad Osman, Rached Zantout. Correcting Noise in Block Ciphers, *Submitted for Publication*, 2009.