New Method for Encryption using Mixing Advanced Encryption Standard and Blowfish Algorithms

Saba Abdul-Baqi Salman

Al-Iraqia University/ College of Education/Computer Department

tuqa52004@yahoo.com

Abstract

The Encryption algorithm transform the data in an unintelligible form using an encryption key so that the only possible way to recover the original data from the encrypted data is to decrypt it using the corresponding decryption process with the decryption key. The proposed algorithm used two known algorithms Advanced Encryption Standard and Blowfish by mixing between them. The new algorithm is more secure than AES and Blowfish and the cipher can't be broken by any attacks. The tested of proposed algorithm used variables data length such as 128 bits, 192 bits and 256 bits with variables key length as 128 bits,192 bits and 256 bits. The proposed algorithm has been designed using Visual Basic 6.0 programming.

Keywords: (AES) Advanced Encryption Standard, (MAC) Message Authentication Code, (DES)Data Encryption Standard.

المستخلص

ان خوارزمية التشفير هي تحويل البيانات من شكل واضح باستخدام مفتاح التشفير بحيث أن الوسيلة الوحيدة الممكنة لاستعادة البيانات الأصلية من البيانات المشفرة هو فك تشفيرها باستخدام عملية فك التشفير المقابلة مع مفتاح فك التشفير. الخوارزمية المقترحة تستخدم اثنين من الخوارزميات المعروفة وهما خوارزمية AES وخوارزمية Blowfish بواسطة الدمج بين الاثنين. الخوارزمية الجديدة تكون اكثر امنا من خوارزمية AES و خوارزمية Blowfish والشفرة لهذه الخوارزمية لا يمكن كسرها بواسطة المهاجمين. ان عملية اختبار الخوارزمية المقترحة تمت بواسطة احجام مختلفة لطول البيانات منها (١٢٨ بت و١٩٢ بت و ٢٠٩ بت) و باستخدام اطوال مختلفة المفتاح التشفير وهي (١٢٨ بت و١٩٢ بت و ٢٠٩ بت). تمبرمجة الخوارزمية المقترحة باستخدام لغة Visual Basic 6.0.

Introduction

Securing data in transmission is the most common real-life cryptographic problem. Basic security services require both encryption and authentication. This is (almost) always done using a symmetric cipher public-key systems are only used to set up symmetric keys and a Message Authentication Code (MAC). Cryptography is only a part of security, as it does not provide all the necessary parts of a secure system. Cryptography adds some security to existing systems Cryptography is the science of protecting clear, meaningful information using mathematical algorithms. Using common encryption algorithms, cryptography provides support for multiple security services to protect the information. Security Services such as digital signature and data confidentiality can be provided by Cryptography. Multiple algorithms may be used to protect information with cryptography.

They are two different types: Hashing algorithms: the Hashing algorithms digest data to produce a string of data (the message hash) that is unique to this data, so that If any part of the original data is changed, the resulting hash changes. It is impossible to recover the original data from the hash. Hashing functions are one-way functions. [1]

Encryption Algorithms: the Encryption algorithms transform the data in an unintelligible form using an encryption key so that The only possible way to recover the original data from the encrypted data is to decrypt it using the corresponding decryption process with the decryption key .It is not possible to deduce either of the keys from any forms of the message. It is possible to recover the original data from the enciphered one

1. Advanced Encryption Standard Algorithm

The Advanced Encryption Standard (AES) is a symmetric-key encryption standard adopted byte United States government. The standard comprises three block ciphers, AES-128, AES-192 and AES-256, adopted from a larger collection originally published as Rijndael. Each of these ciphers has a 128-bit block size, with key sizes of 128, 192 and 256 bits, respectively. The AES ciphers have been analyzed extensively and are now used worldwide, as was the case with its predecessor, the Data Encryption Standard (DES). (3)

AES was announced by National Institute of Standards and Technology (NIST) as U.S. FIPS PUB 197 (FIPS 197) on November 26, 2001 after a 5-year standardization process in which fifteen competing designs were presented and evaluated before Rijndael was selected as the most suitable. It became effective as a Federal government standard on May 26, 2002 after approval by the Secretary of Commerce. It is available in many different encryption packages. AES is the first publicly accessible and open cipher approved by the NSA for top secret information (see Security of AES, below). The Rijndael cipher was developed by two Belgian cryptographers, JoanDaemen and Vincent Rijmen, and submitted by them to the AES selection process. (3,4)

AES is based on a design principle known as a Substitution permutation network. It is fast in both software and hardware. Unlike its predecessor, DES, AES does not use a Feistel network. AES has a fixed block size of 128 bits and a key size of 128, 192, or 256 bits, whereas Rijndael can be specified with block and key sizes in any multiple of 32 bits, with a minimum of 128 bits. The block size has a maximum of 256 bits, but the key size has no theoretical maximum. AES

Operates on a 4x4 matrix of bytes, termed the state (versions of Rijndael with a larger block size have additional columns in the state). Most AES calculations are done in a special finite field.

The AES cipher is specified as a number of repetitions of transformation rounds that convert the input plaintext into the final output of cipher text. Each round consists of several processing steps, including one that depends on the encryption key. A set of reverse rounds are applied to transform cipher text back into the original plaintext using the same encryption key. 5

High-level description of the algorithm

1. ***Key Expansion-round keys are derived from the cipher key using Rijndael's key. Schedule***
2. ***Initial Round.***
    a. **Add Round Key-each byte of the state is combined with the round key using bit wisexor**
3. ***Rounds***
    a. **Sub Bytes-a non-linear substitution step where each byte is replaced with another according to a lookup table.**
    b. **Shift Rows-a transposition step where each row of the state is shifted cyclically a certain number of steps.**
    c. **MixColumns-a mixing operation which operates on the columns of the state combining the four bytes in each column.**
    d. **Add Round Key**
4. ***Final Round (no MixColumns)***
    a. **Sub Bytes.**
    b. **Shift Rows.**
    c. **Add Round Key.**

The SubBytes step

In the subBytes step, each byte in the matrix is updated using an 8−bit substitution box, the Rijndael S−box. This operation provides the non−linearity in the cipher. The S−box used is derived from the multiplicative inverse over GF(8), known to have good non−linearity properties. To avoid attacks based on simple algebraic properties, the S−box is constructed by combining the inverse function with an invertible affine transformation. The S−box is also chosen to avoid any fixed points (and so is a derangement), and also any opposite fixed points.

The ShiftRows step

The Shift Rows step operates on the rows of the state; it cyclically shifts the bytes in each row bya certain offset. For AES, the first row is left unchanged. Each byte of the second row is shifted one to the left. Similarly, the third and fourth rows are shifted by offsets of two and three respectively. For the block of size 128 bits and 192 bits the shifting pattern is the same. In this way, each column of the output state of the Shift Rows step is composed of bytes from each column of the input state. (Rijndael variants with a larger block size have slightly different offsets). In the case of the 256−bit block, the first row is unchanged and the shifting for second, third and fourth row is 1 byte, 3 bytes and 4 bytes respectively − this change only applies for the Rijndael cipher when used with a 256−bit block, as AES does not use 256−bit blocks. Here Bij is from cipher text and Bij is from key.(6)

The Mixcolumns step

In the MixColumns step, the four bytes of each column of the state are combined using an invertible linear transformation. The MixColumns function takes four bytes as input and outputs four bytes, where each input byte affects all four output bytes. Together with ShiftRows, MixColumns provides diffusion in the cipher. During this operation, each column is multiplied by the known matrix that for the 128 bit key is the multiplication operation is defined as: multiplication by 1 means leaving unchanged, multiplication by 2 means shifting byte to the left and multiplication by 3 means shifting to the left and then performing xor with the initial unshifted value. After shifting, a conditional xor with 0x11B should be performed if the shifted value is larger than 0xFF.

In more general sense, each column is treated as a polynomial over GF(2) and is then multiplied modulo x'+1 with a fixed polynomial c(x) = 0x03 − x + x + x + 0x02. The coefficients are displayed in their hexadecimal equivalent of the binary representation of bit polynomials from GF(2)(x). The MixColumns step can also be viewed as a multiplication by a particular MDSmatrix in a finite field (This process is described further in the article Rijndael mix columns. Using the polynomial one matrix was created. Using that matrix add with osp came from previous state.

The AddRoundKey step

In the AddRound Key step, the subkey is combined with the state. For each round, a subkeyisderived from the main key using Rijndael's key schedule; each subkey is the same size as the state. The subkey is added by combining each byte of the state with the corresponding byte of the subkey using bitwise XOR. (7)

2. Blowfish Algorithm

Blowfish is a keyed, symmetric cryptographic block cipher designed by Bruce Schneier in 1993and placed in the public domain. Blowfish is included in a large number of cipher suites and encryption products, including Splash ID. Blowfish's security has been extensively tested and proven. As a public domain cipher, Blowfish has been subject to a significant amount of cryptanalysis, encryption has never been broken Blowfish is also one of the fastest block ciphers in public use, making it ideal for a product like SplashID that functions on awide variety of processors found in mobile phones as well as in notebook and desktop computers.(7,8)

Schneier designed Blowfish as a general-purpose algorithm, intended as a replacement for the aging DES and free of the problems associated with other algorithms. Notable features of the design include key-dependent S-boxes and a highly complex key schedule. Blowfish has a 64-bit block size and a key length of anywhere from 32 bits to 448 bits. It is a 16-round Feistel cipher and uses large key-dependent S-boxes. It is similar in structure to CAST-128, which uses fixed S-boxes.(9)

The Blowfish Algorithm Work

The Blowfish structure has two lines. Each line represents 32 bits. The algorithm keeps two subkey arrays: the 18-entry P-array and four 256-entry S-boxes. The S-boxes accept 8-bit input and produce 32-bit output. One entry of the P-array is used every round, and after the finalround, each half of the data block is XORed with one of the two remaining unused P-entries.

The function splits the 32-bit input into four eight-bit quarters, and uses the quarters as input to the S-boxes. The outputs are added modulo 232 and XORed to produce the final 32-bit output. Since Blowfish is a Feistel network, it can be inverted simply by XORing P17 and P18 to the cipher text block, then using the P-entries in reverse order. Blowfish's key schedule starts by initializing the P-array and S-boxes with values derived from the hexadecimal digits of pi, which contain no obvious pattern The secret key is then XORed with the P-entries in order (cycling the key if necessary). A 64-bit all-zero blocks is then encrypted with the algorithm as it stands. Theresultant cipher text replaces P1 and P2. The cipher text is then encrypted again with the new subkeys, and P3 and P4 are replaced by the new cipher text. This continues, replacing the entire P–array and all the S–box entries. In all, the Blowfish encryption algorithm will run 521 times to generate all the subkeys – about 4KB of data is processed.10

3. The Proposed Algorithm

In this work the proposed algorithm have been built by mixing two known algorithms Advanced Encryption Standard and Blowfish and then tested the new algorithm.

There are five cases in encryption Proposed Algorithm:

a. Using plaintext 128 bits at first encrypted by AES and using key length 128 bits then the output is 128 bits as cipher. The cipher is input to the Blowfish as plaintext and encrypted by this algorithm, the final output is the cipher of proposed algorithm. '

b. In the second case used plaintext 192 bits encrypted by AES using 128 bits of key length. The output using as input in Blowfish then the final output is the proposed algorithm cipher.

c. The plaintext is 256 bits and the key is 128 bits using in AES, then output using as input to Blowfish. The final output is new method cipher 1

d. In this case using the key length as variant such as (128،192.256) bits with only 128 bits as plaintext. The cipher of AES used as input in Blowfish then the output is the final cipher.

e. In this case using variant plaintext and variant key length, such as plaintext (128,192,256) bits and key length (128,192,256) bits. The output of AES using as input in Blowfish then the final output is the proposed cipher. The Proposed Algorithm for encryption shows in figure (1).

Decryption the Proposed Algorithm have been five Cases: –

1. Using cipher text 128 bits input to the Blowfish for decryption and the output decryption by AES with key 128 bits the final text is plaintext.
2. Using cipher text 192 bits input to the Blowfish for decryption and the output decryption by AES with key 192 bits the final text is plaintext.
3. Using cipher text 256 bits input to the Blowfish for decryption and the output decryption by AES with key 256 bits the final text is plaintext.
4. In this case using 128 bits as cipher that decryption by Blowfish and input to the AES decryption algorithm by using key length 192 bits, the output is plaintext in 128 bits.
5. In this case using 128 bits as cipher that decryption by Blowfish and input to the AES decryption algorithm by using key length 256, bits the output is plaintext in 128 bits.
6. In this case using 256bits as cipher that decryption by Blowfish and input to the AES decryption algorithm by using key length 192 bits, the output is plaintext in 128 bits. The proposed algorithm for Decryption show in figure (2).

In the Proposed algorithm used two cases for represented cipher text:

1. Using the AESHexadecimal number for representing cipher text. As shows in figures (4, 5).
2. Using the Special characters for representing cipher text. As shows in figure (6, 7).

4. Result and Discussion

In the figure (3) shows the proposed algorithm for encryption and decryption text. In the figure (4) shows the new algorithm for encryption 128 bits as plaintext and using key length 128 bits. The cipher is 128 bits represented aS hexadecimal numbers. In the figure (5) shows the new algorithm for decryption 128 bits as cipher and using key length 128 bits. The plaintext is 128 bits. For, represented by Nb = 4, which reflects the number of 32-bit words (number of Columns) in the State.

For the new algorithm, the length of the Cipher Key, K, is 128, 192, or 256 bits. The key length is represented by Nk = 4, 6, or 8, which reflects the number of 32-bit words (number of columns) in the Cipher Key.

For the new algorithm, the number of rounds to be performed during the execution of the algorithm is dependent on the key size. The number of rounds is represented by Nr, where Nr = 10 when Nk = 4, Nr = 12 when Nk = 6, and Nr = 14 when Nk = 8, as shows in tables (1)

	Key length (NK words)	Block Size (NK words)	Number Of (Nr)
128	4	4	10
192	6	4	12
256	8	4	14

In the figure (6) shows the encryption text for plaintext 192 bits and key length 192 bits and cipher text represent as special characters. In the figure (7) shows the decryption 192 bits as cipher and key length 192 bits and the plaintext is 192 bits.

In the figure (8) shows the encryption and decryption text for plaintext 256 bits and key length

256 bits and cipher text represent by 256 bits.

5. Conclusions

A new method for encryption have been high security for encryption since it is encryption the plaintext in twice that gives more complex for decryption by attacks, the proposed method using the function of key generation that gave more security since the key cannot be found by using any function, when using 256 bit for key length to encrypted 256 bit plaintext any attack needed 1.157*e+77 years so it is very difficult to broken this cipher. Aes

6. References

[1] S. Miyaguchi, "The FEAL Cipher Family," Advances in Cryptology-CRYPTO '90 Proceedings, Springer-Verlag, 1991, pp. 627-638.

[2] M. Matsui, "Linear Cryptanalysis Method for DES Cipher," Advances in CryptologyCRYPTO'93 Proceedings, Springer-Verlag, 1994, in preparation.

[3] J. Daemen and V. Rijmen, AES Proposal: Rijndael, AES Algorithm Submission, September 3, 1999.

[4] J. Daemen and V. Rijmen, The block cipherRijndael, Smart Card research and Applications, LNCS 1820, Springer-Verlag, pp. 288-296.

[5] A. Lee, NIST Special Publication 800–21, Guideline for Implementing Cryptography in the Federal Government, National Institute of Standards and Technology, November 1999.

[6] J. Nechvatal, et al., Report on the Development of the Advanced Encryption Standard (AES), National Institute of Standards and Technology, October 2, 2000.

[7] A. Menezes, P. van Oorschot, and S. Vanstone, Handbook of Applied Cryptography, CRC Press, New York, 199]. p. 81-83.

[8] T.W. Cusick and M.C. Wood, "The REDOC-II Cryptosystem," Advances in CryptologyCRYPTO'90 Proceedings, Springer- Verlag, 1991, pp. 545-563.

[9] GOST 28147-89, "Cryptographic Protection for Data Processing Systems," "Cryptographic Transformation Algorithm," Government Standard of the U.S.S.R., Inv. No. 3583, UDC 681.325.6:006.354.

[10] C. Merkle, "Method and Apparatus for Data Encryption," U.S. Patent 5,003,597, 26 Mar 2006.-

Appendix

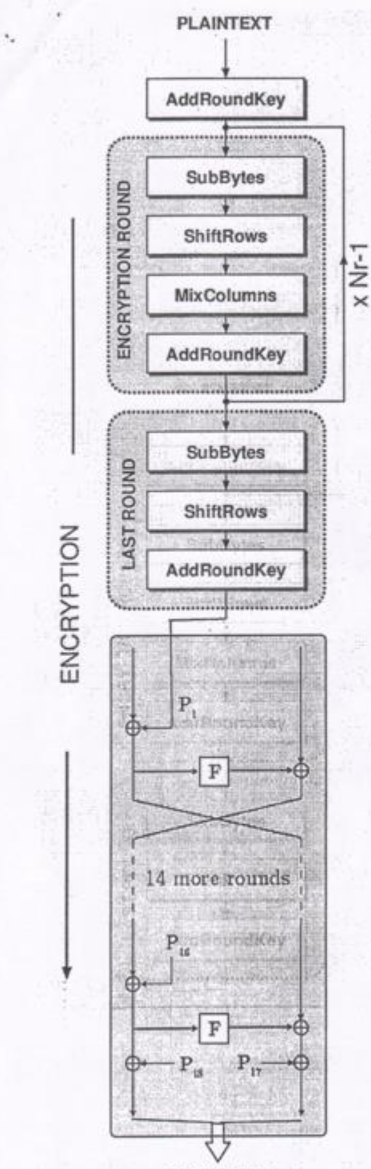


Figure (1)The flowchart for Proposed Algorithm (Encryption process).

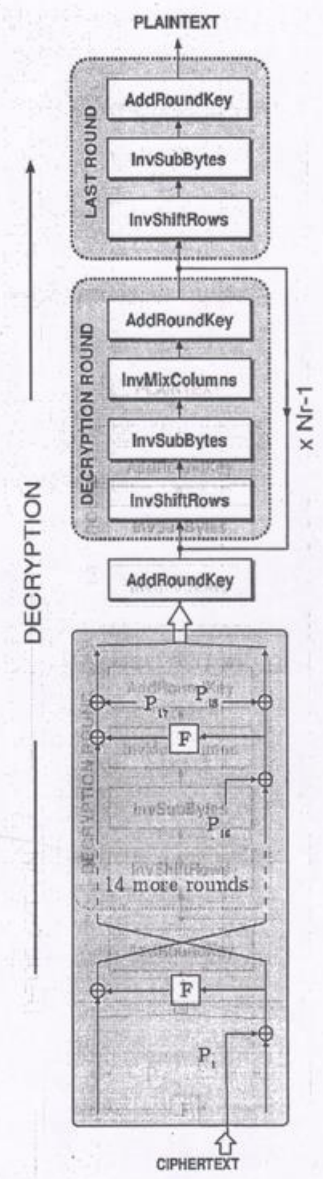


Figure (2)The flowchart for Proposed Algorithm (Decryption process).

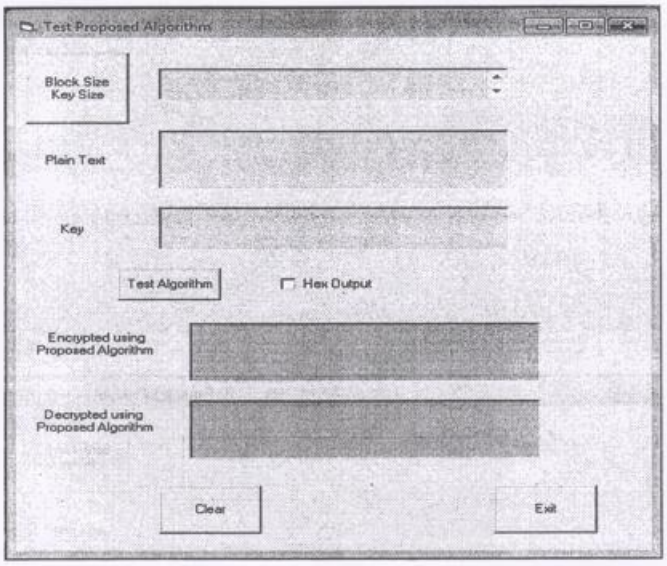
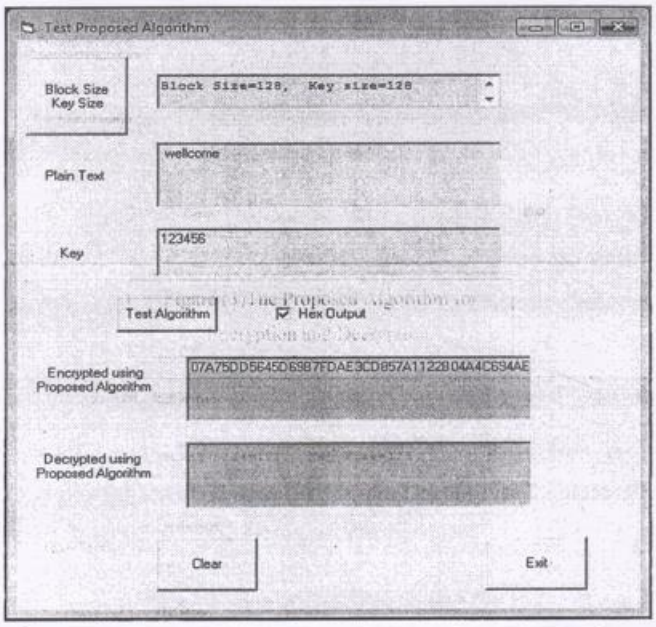


Figure (3)The Proposed Algorithm for Encryption and Decryption.



Block Size Key Size: Block Size=128, Key size=128

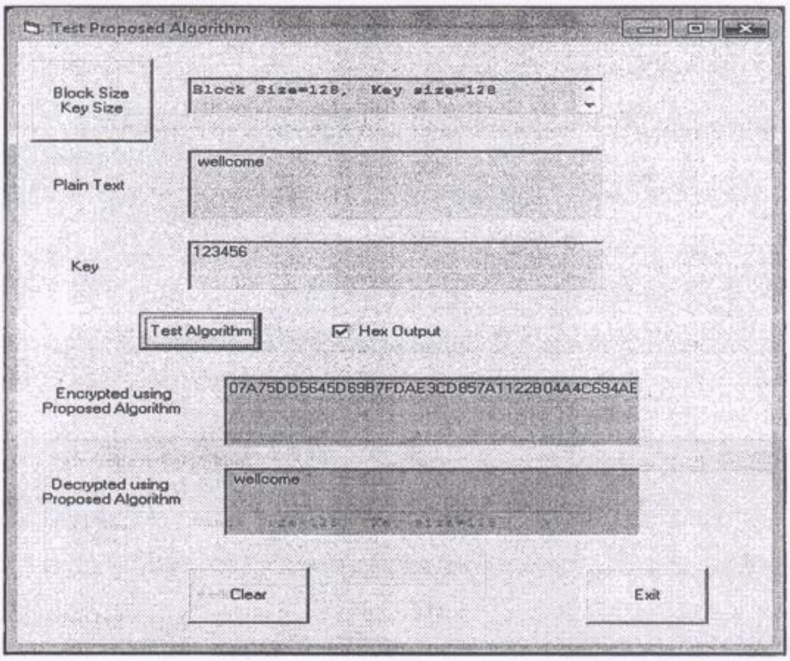
Plain Text: wellcome

Key: 123456

☑ Hex Output

Encrypted using Proposed Algorithm: 07A75DD5645D6987FDAE3CD857A1122804A4C694AE

Figure (4)The Proposed Algorithm for Encryption 128 bit with 128 bit key length represented cipher as hexadecimal numbers.



Block Size Key Size: Block Size=128, Key size=128

Plain Text: wellcome

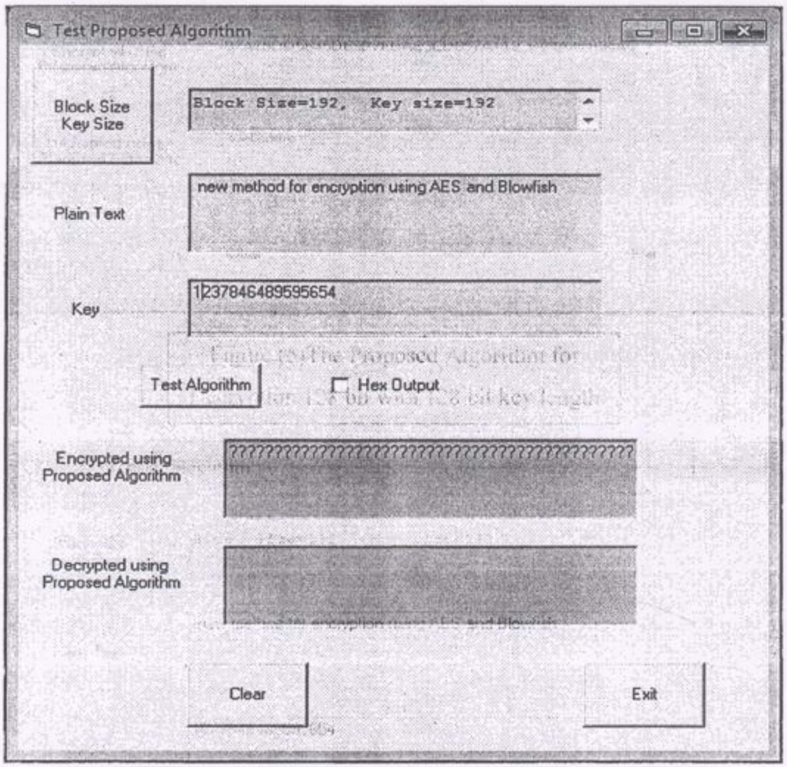
Key: 123456

☑ Hex Output

Encrypted using Proposed Algorithm: 07A75DD5645D69B7FDAE3CD857A1122B04A4C694AE

Decrypted using Proposed Algorithm: wellcome

Figure (5)The Proposed Algorithm for Decryption 128 bit with 128 bit key length



Block Size Key Size: Block Size=192, Key size=192

Plain Text: new method for encryption using AES and Blowfish

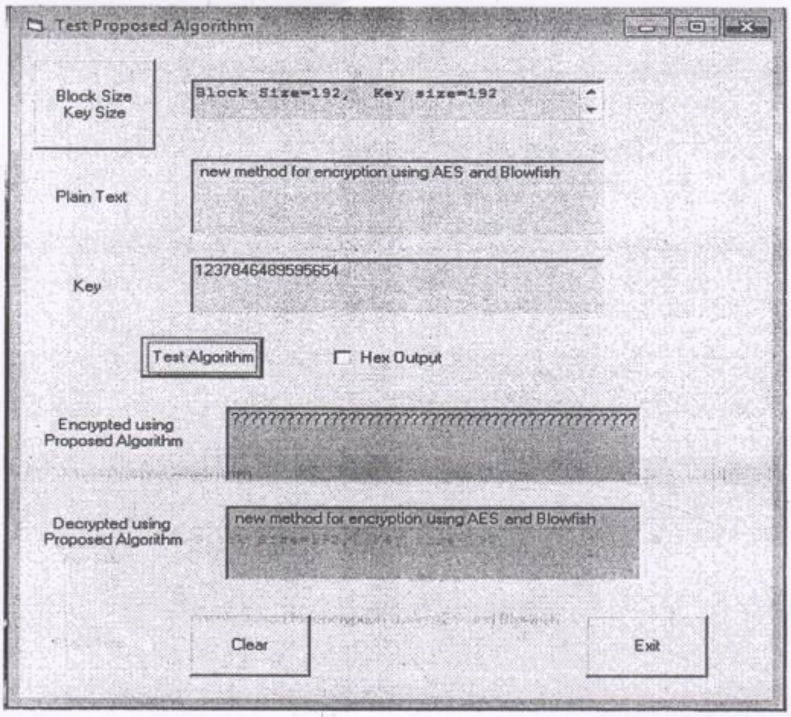
Key: 1237846489595654

☐ Hex Output

Encrypted using Proposed Algorithm: ??

Decrypted using Proposed Algorithm:

Figure (6)The Proposed Algorithm for Encryption 192 bit with 192 bit key length represented cipher as special characters.



Block Size=192, Key size=192

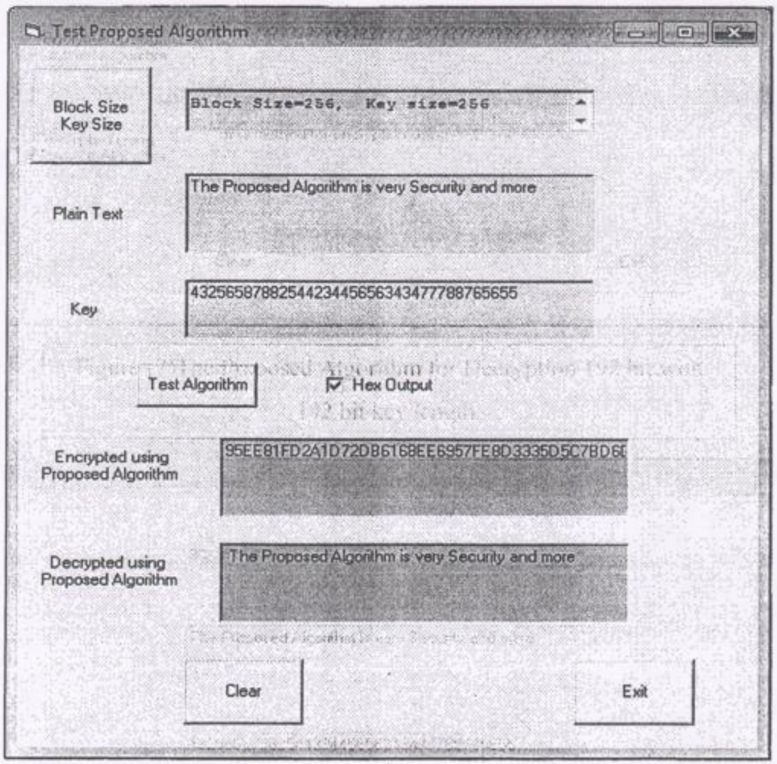
new method for encryption using AES and Blowfish

1237846489595654

??

new method for encryption using AES and Blowfish

Figure (7)The Proposed Algorithm for Decryption 192 bit with 192 bit key length.



Block Size=256, Key size=256

The Proposed Algorithm is very Security and more

43256587882544234456563434777887656555

95EE81FD2A1D72DB6168EE6957FE8D3335D5C7BD6

The Proposed Algorithm is very Security and more

Figure (8)The Proposed Algorithm for Encryption and Decryption 256 bit with 256 bit key length.