International Journal of Computing Algorithm
Volume: 04 Issue: 01 June 2015 Pages:49-53
ISSN: 2278-2397

Format Preserving Encryption for Small Domain

S.Vidhya[1],K.Chitra[2]
[1]Research Scholar, Department of Computer Science and Applications, SCSVMV University, Kancheepuram, Tamilnadu, India
[2]Assistant Professor, Department of Computer science, Govt. Arts college, Melur, Madurai, Tamilnadu, India.
Email: vidhyarajmca@gmail.com, manikandan.chitra@gmail.com

Abstract-Cryptography is important in communicating secured information that is vulnerable to distortion. The main goal of this paper is encrypting the small arbitrary length data without any changes in the length and data type. We propose a flexible arbitrary length small domain block cipher (FPESD). FPESD is based on AES Algorithm. The resulting cipher text is the same as input plaintext. Here, we use Galois finite field GF (28) and format preserving key to implement FPE. For decryption format preserving key is used along with cipher text and secret key.

Keywords -Format Preserving Encryption, Small domain, AES CTR, Credit Card Number Encryption

I. INTRODUCTION

There is a need for privacy of sensitive data before data are shared with any cloud provider, client server etc. Sensitive data fields having well defined data formats[1]. While designing privacy for sensitive fields, it may be desirable to maintain the length and format of the inputs, in order to avoid any re-constructions of packet formats or database columns of existing system. We propose a traditional AES cipher with CTR mode along with format preserving key[2].

II. NEED FOR FPE

- In normal encryption, the length and format of thePlaintext was entirely changed.
- The database structure was also changed by the encrypted column.
- The reconstruction of database is a very difficult process.
- If randomization was used in the encryption algorithm then referential integrity of the data base was also affected.
- Special problems arise when encrypted data is indexed. The index of the column contains encrypted values and then the index is unusable for encrypted data.
- An encryption algorithm not only alters the structure of the table it also alters the queries passed to the database. Changing of database and queries are complex tasks and also cost prohibitive.

III. IMPACT OF NORMAL ENCRYPTION ON DATABSES AND QUERIES

Most of the block ciphers support length preserving encryption. N bit block is mapped into another N bit block. But the data type of cipher text is not same as plain text. Format preserving encryption is a combination of length and data type preserving encryption.We give an example to encrypt credit card number in stored in the data base. Create a Customer_detail a table which contains credit card details of the customers. Our task is to protect this data by encrypting the column, which contains the credit Custid, Custname and Credit card number.

```
/* Create a table Customer_detail */
Create table Customer_detail (Custidint,
Custtnamevarchar(25), Credicardnuminteger(16);
/* Display the content of Customer_detail */
Select * from Customer_detail;
/* Create index to locate and retrieve data faster and more
efficient */
Create index idindex on Customer_detail(Credicardnum);
```

Table1. Database Before Encryption

Custid	Custname	Creditcardnum
1	N.S.Velu	1234567878901245
2	S.Ushadevi	7661234567199887
3	S.Dhivya	3456789897890124

Electronic Code Book (ECB) mode is used to encrypt our database column. This mode does not require any initialization vector. This mode requires our input should be multiple of 8 bytes [5].The following queries are required to update Customer_detail table after encryption.

```
/* Add columns which will hold the encrypted data in binary */
AltertableCustomer_detail add
EncryptedCreditCardNumvarbinary(128);
/* Update new column with encrypted data */
Update Customer_detail set EncryptedCreditCardNum =
encryptbykey('Symmetrickey', Creditcardnum);
/* Drop the original column */

Alter table Customer_detail drop column Creditcardnum;
```

The original column and decrypted column have different type. The conversion function is required to get the original data type.

```
/* Drop the index */
Drop index idindex;
```

An index is useless for encrypted column, so it is removed.

```
Select Custid, Custname, convert (varchar(20), decryptbykey
(Encrypted CreditCardNum ) ) fromCustomer_detail;
```

After completing all the encryptions, our database schema will be changed to store the encrypted value. The corresponding queries that handle the credit card number will also be changed[11]. Cryptographic operations, required to store sensitive data securely, need more amount of arithmetic calculations. Coupled with bad block cipher choice, expensive cryptographic operations needed for querying and processing encrypted data could decrease the system performance dramatically. To avoid such impacts we introduce format preserving encryption to produce cipher text as like as plaintext in length and format[6].

International Journal of Computing Algorithm
Volume: 04 Issue: 01 June 2015 Pages:49-53
ISSN: 2278-2397

Table 2. Database after Encryption

Cus tid	Custna me	Encrypted CreditCardNum
1	N.S.Ve lu	0BDC16E6A797C535C49F67688C6D4E21 D3F36088C206C85A
2	S.Usha devi	0BDC16E6A777C534264AF5FD1E8BD57 0DDD44E842A72C00B
3	S.Dhiv ya	5408551E9C4A0F8FC49F67688C6D4E21 D3F36088C206C95A

IV. FPE USING AES BLOCK CIPHER

All the existing format preserving algorithms used arbitrary length cipher mostly Feistelcipher[7]. In our proposed FPESD we introduce most secured AES cipher with CTR mode. CTR mode can handle messages of arbitrary length. Unlike other common modes of operation, handling messages of arbitrary bit-length is made trivial. No bits are wasted in doing this—the cipher text C is of the same length as the plaintext M. An arbitrary length feature is especially useful for small domain encryption[3].

A. *Advantages of CTR Mode*

1. CTR-mode encryption enables effective utilization of the ofthearchitectural features such as parallel processing, pipelining, multiple instruction dispatch per, many registers, and SIMD instructions[5].
2. CTR model is fully parallelizable: one can be computing blocks C_1rC_2, . . rall at the same time.
3. With CTR mode, both encryption and decryption depend only on E—neither depends on the inverse map, $D = E^{-1}$. So D need not be implemented.
4. No plaintext expansion is needed. Cipher text can be truncated to plaintext length.
5. The CTR mode of operations treats the blocks independently, so there is no propagation of error from one block to another.

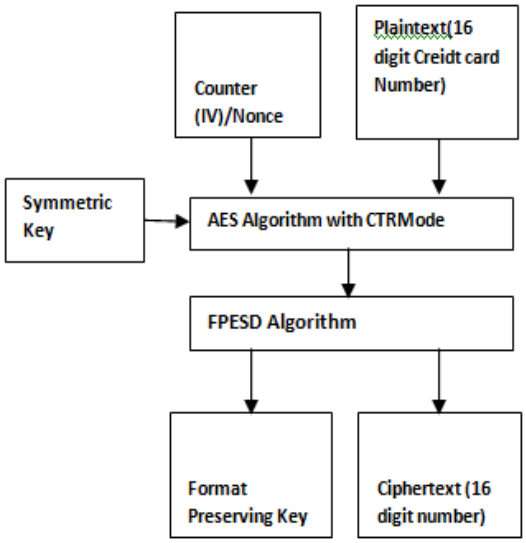


Fig1 : FPESD Encryption Algorithm

V. FPESD ALGORITHM

The proposed FPESD algorithm is based on AES algorithm. At the end of the AES algorithm some calculations are added to produce the cipher text which is same as plaintext[8]. Along with cipher text the format preserving key is also produced. It is stored along with the initial vector IV.In this paper we illustrate how to apply our proposed system to encrypt credit card numbers.

A. *Generating 128 – bit Symmetric key*

An AES key is nothing more than a random bit string of the right length. For a 128-bit AES key we need 16 bytes. The strength of the key depends on the unpredictability of the random.For example AES Key in hex 000102030405060708090a0b0c0d0e0f.

Table 3 : Aes Key

K_0	K_4	K_8	K_{12}
K_1	K_5	K_9	K_{13}
K_2	K_6	K_{10}	K_{14}
K_3	K_7	K_{11}	K_{15}

Table 4: Aes Key Example

00	04	08	0c
01	05	09	0d
02	06	0a	0e
03	07	0b	0f

B. *Generating 128 – bit Initial Vector*

An initialization vector (IV) is arandom number and used along with a symmetric key for data encryption. This number, also called a nonce, is used only one time in any session. The use of an IV prevents duplication in data encryption, making it more difficult for a hacker to find patterns and break a cipher text.

Table5 :Initial Vector Iv

3E	E5	D3	EE
0F	A6	BF	A6
57	63	F1	3E
EE	F1	D3	0F

C. *Plaintext*

The input for encryption algorithm is a single 128 bit block. The input is represented as a square matrix of bytes. The block is copied to state array. The ordering of bytes in a matrix is by column. The first four bytes occupied the first column; the second four bytes occupied the second column and so on. The same procedure is followed for key also. The state array is represented as follows.

Table 6 :Aes Plaintext

S0	S1	S2	S3
S4	S5	S6	S7
S8	S9	S10	S11
S12	S13	S14	S15

International Journal of Computing Algorithm
Volume: 04 Issue: 01 June 2015 Pages:49-53
ISSN: 2278-2397

For example using any padding technique with 16 digit credit card number to represent 32 digit hex is 3EE5D3EE0FA6BFA65763F13EEEF1D30F. Each hex is converted to 4 bit binary number. Totally we have 128 bit.

Table 7: Aes Plaintext Example

3E	E5	D3	EE
0F	A6	BF	A6
57	63	F1	3E
EE	F1	D3	0F

D. AES Encryption

The IV/nonce and the counter can be combined by XOR to produce the actual unique counter block for encryption.

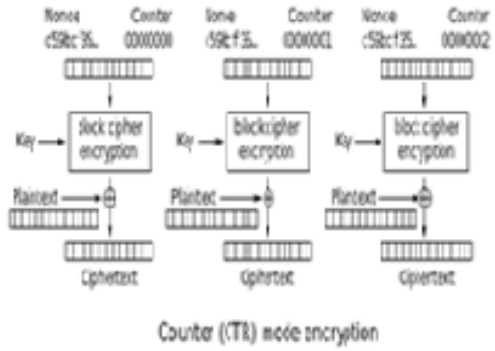


Figure:1

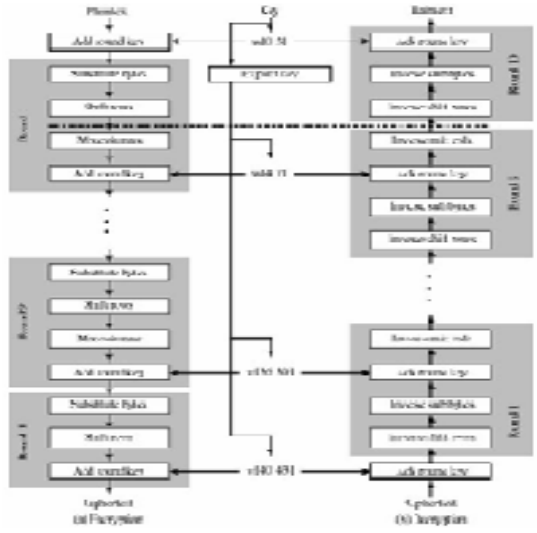


Fig 2.AES Data Structures

Table:8

3E	E5	D3	EE
0F	A6	BF	A6
57	63	F1	3E
EE	F1	D3	0F

Table 9 :SubByte Transformation

b2	d9	66	28
76	24	08	24
5b	2e	a1	b2
28	a1	66	76

→

b2	d9	66	28
24	08	24	76
a1	b2	5b	2e
76	28	a1	66

CTR supports parallelization because you can split the message into blocks, each chunk having a range of counter values associated with it, and encrypt (or decrypt) each block independently

The cipher begins with an AddRoundKey step and nine rounds. Each round includes all four steps, and a final round includes only three steps. Four different steps are used, one for permutation and three for substitution:

- Substitute bytes
- ShiftRows
- MixColumns
- AddRoundKey

A. Substitute bytes

AES defines a 16 x 16 matrix called S-Box. Each element in a matrix represents byte values. Each individual byte of State is converted into a new byte in the following way: The leftmost 4 bits of the byte indicate row index and the rightmost 4 bits indicate column index. These row and column values provide indexes into the S-box to select a byte as output value. S-box is constructed using multiplicative inverse of *Galois* finite field GF(28).SubByte transformation for the given example is as follows

B. ShiftRows

The first row of State is not shifted. From the second row to fourth row circular left shift is performed. For the second row 1 byte, third row 2 bytes, fourth row 3 bytes are shifted. The following is an example of ShiftRows:

C. MixColumns

MixColumns, operates on each column separately. Each byte of a column is converted into a new byte that is a function of all four bytes in that column. The result can be defined by the following matrix multiplication in GF(28) on State.

Table:10 mix columns

$$\begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \begin{pmatrix} S0 & S1 & S2 & S3 \\ S4 & S5 & S6 & S7 \\ S8 & S9 & S10 & S11 \\ S12 & S13 & S14 & S15 \end{pmatrix} = \begin{pmatrix} S'0 & S'1 & S'2 & S'3 \\ S'4 & S'5 & S'6 & S'7 \\ S'8 & S'9 & S'10 & S'11 \\ S'12 & S'13 & S'14 & S'15 \end{pmatrix}$$

D. AddRoundKey Transformation

The 128 bits of State are XORed with the 128 bits of the key. The operation is viewed as a column wise operation between the 32 bits of a State column and one word of the round key.The following is an example of Addroundkey Transformation.

International Journal of Computing Algorithm
Volume: 04 Issue: 01 June 2015 Pages:49-53
ISSN: 2278-2397

Table:11

C4	2a	5a	82
54	2c	0c	D0
55	ab	cc	A8
ec	9f	8c	ec

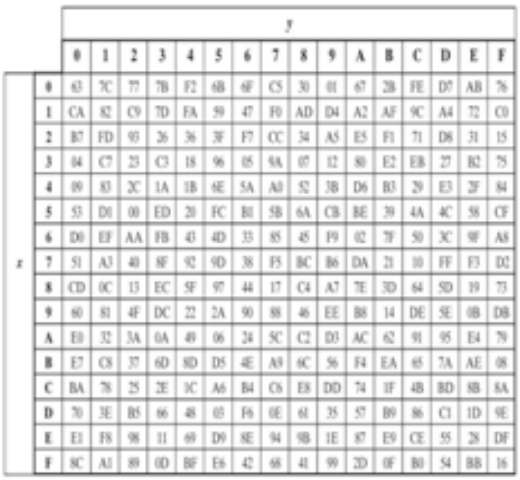
⊕

00	04	08	0c
01	05	09	0d
02	06	0a	0e
03	07	0b	0f

c4	2e	52	8e
55	29	05	dd
57	ad	c6	a6
ef	98	87	e3

E. FPESD Encryption

At the end of tenth round of AES cipher we get 128 bit (16 bytes) cipher text. The following steps are used to implement format preserved cipher text that converts 128 bit into 16 decimal digits.



Input :

128 bit cipher text S & 128 bit key K
Output : 16 digits and Format preserving Key F.

1. Set i and sum value to 0.
2. Select i th byte from state array(s_i) and i th byte from expanded key(k_i).
3. Calculate ei = $s_i k$⊕
4. The resultant ei is applied to s-box and we get the new cipher text nci
5. Calculate the decimal equivalent of nc_i.
6. D_i = decimal(nc_i)mod 10 and F[i]=tochar(decimal(nc_i)/10)
7. Calculate sum = sumx10+d_i
8. Increment i value with 1 .
9. Repeat from step 2 to step 7 until I value becomes 16.
10. The sum variable contains format preserved cipher text.(16 decimal digits) and array F contains Format preserving key.

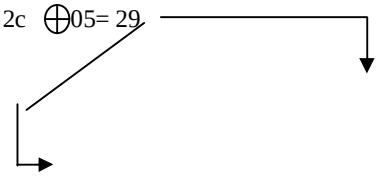
The IV must be random and unpredictable. The IV can be made public. The format preserving Key F is stored along with the key. Even cryptanalysis find out IV and F, they cannot break the cipher without the secret key K.

VI. TEST VECTORS

The FPESD encryption Algorithm contains 16 iterations. For simplicity,iteration for each algorithm is explained below. At the end of the 16th iteration the character array F contains Format Preserving Key. It is stored along with the initial vector IV.

For example

K_5 = 2c
S_5 = 05
e_5 = 2c ⊕05= 29



nc_5 = a5
d_5 = decimal (a5) mod 10
= 165 mod 10 = 5
F[5]= 165 / 10 = 16 (ordinal position of an alphabet) = p

VII. PERFORMANCE ANALYSIS OF FPESD

1. Existing block cipher AES is used only once. It requires only small amount of additional work to implement FPE.
2. The input length is not restricted by the block length.
3. The duration of ciphering is deterministic.
4. Instead of original key if we are using expanded key in FPE Algorithm then it is more complex to break.
5. Indexing Technique can be applied for encrypted data.

The following chart represents number of block cipher encryption to encrypt credit card number[12].

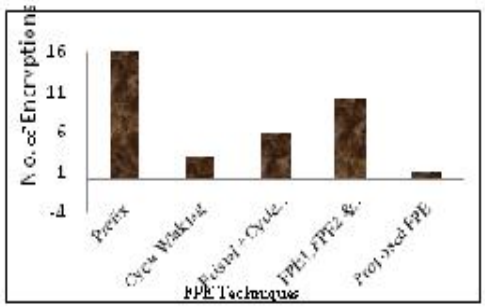


Fig 4.Performance Analysis Graph

Cycle walking and Feistel network requires block cipher with 54 bit block. The number of bits required to represent 16 digits decimal number is minimum 54 bits. If we are using more secured AES-128 block cipher then the numbers of iterations are also increased. For cycle walking the cycle length for aes-128 is 274.(128 – 54 = 74)[13].

Our proposed system requires only minimum number of encryptions compared with other existing techniques.

International Journal of Computing Algorithm
Volume: 04 Issue: 01 June 2015 Pages:49-53
ISSN: 2278-2397

VIII. CONCLUSION

FPE is an emerging technique in cryptographic field. Especially it is useful for encrypting credit card numbers and social security numbers. Using Format preserving Encryption the data base schema and applications related to the database will never changed. The cost and time for changing the data base is minimized.

REFERENCES

[1] Gary C.Kessler, "An overview of Cryptography ",21 January 2015 .
[2] Thomas Ristenpart, Scott Yilek, Advances in Cryptology – CRYPTO 2013, Lecture Notes in Computer Science Volume 8042, 2013, pp 392-409 The Mix-and-Cut Shuffle: Small-Domain Encryption Secure against N Queries
[3] Helger Lipmaa ,Phillip Rogaway,Comments to NIST concerning AES Modes of Operations: CTR-Mode Encryption
[4] Sashank Dara and Scott Fluhrer , Advanced Encryption Standard (AES) New InstructionsSet FNR : Arbitrary length small domain block cipher proposal.
[5] White Paper Shay Gueron Mobility Group, Israel Development Center Intel Corporation
[6] M. Bellare, T. Ristenpart, P. Rogaway, and T. Stegers. Format preserving encryption. SAC 2009. LNCS 5867,Springer, 2009.
[7] V. Hoang and P. Rogaway. On generalized Feistel networks. Conference version of this paper. CRYPTO 2010,Springer, 2010.
[8] Format Preserving Encryption Terence SpiesVoltage Security, Inc.
[9] Phillip Rogaway , A Synopsis of Format-Preserving Encryption
[10] Vidhya.S and Dr.K.Chitra "Format Preserving Encryption using Feistel Cipher" in International Conference on Research Trends in Computer Technologies (ICRTCT - 2013) Proceedings published in International Journal of Computer Applications® (IJCA) (0975 –8887).
[11] Stephen Tu M. Frans Kaashoek Samuel Madden Nickolai Zeldovich MITCSAIL, Processing Analytical Queries over Encrypted Data
[12] S.Vidhya and Dr.K.Chitra , "Securing Data at Rest using Format Preserving Encryption using Pass phrase",International review on Computers and Softwares (IRECOS) V9.N.5 ISSN 1828 – 6003. May 2014.
[13] Morris Dworkin, Recommendation for Block Cipher Modes of Operation: Methods for Format-Preserving Encryption