IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 14, Issue 6 (Sep. - Oct. 2013), PP 23-29
www.iosrjournals.org

Efficient Fpe Algorithm For Encrypting Credit Card Numbers

Dr. K. Chitra[1,] S.Vidhya[2]
[1]*(Assistant Professor,Department of Computer Science,Govt. Arts College, Melur,Madurai, Tamilnadu, India.)*
[2]*(Ph.D Scholar,Department of Computer science and Applications,SCSVMC University,Kancheepuram, Tamilnadu, India.)*

Abstract: *The more highly used Internet world contains many sensitive information. Encryption is a process to secure information. An encrypted data requires more storage space for storing. It also needs many changes in queries and application programs. This paper we introduce an efficient algorithm to encrypt the credit card number without changing the format and type.*
Keywords: *Credit card encryption, FPE, Structured preserving encryption, data type preserving encryption.*

I. INTRODUCTION

Encryption is a process of converting the plaintext (clear text) to unreadable cipher text. Decryption is a reverse process in which the cipher text is converted into plaintext. We store and transmit much sensitive information through the internet. Encrypting data at flight means encryption is done in the sender side and decryption is done in the receiver side. Encrypting data at rest is used to protect confidential data stored on host from the privileged users. There are many database security policies like authentication, access rights, digital signature etc. But 45% of the attackers are insiders. The existing security techniques are not enough for protecting the data. So we need powerful encryption algorithm. Database encryption is the process of converting the plaintext in the database to unintelligible cipher text format. Database encryption is implemented using strong encryption such as AES, RSA or SHA256.

II. DRAWBACKS IN DATABASE ENCRYPTION

- Encrypted text requires more storage space than clear text data
- Inserting encrypted text and decrypted queries are slower than the inserting plaintext.
- SQL statements are slow down due to encryption.
- Record searching is more complex in encrypted database.
- Queries will be changed to handle the encrypted data.
- An existing application programs are also changed to handle the encrypted database.
- Cannot maintain intellectual property of the database such as index.

III. FORMAT PRESERVING ENCRYPTION

Format preserving encryption means encrypting the data without changing the format and data type. The format and type of cipher ext is equivalent to plaintext. [1]

FIGURE 1: CREDIT CARD ENCRYPTION USING AES-128

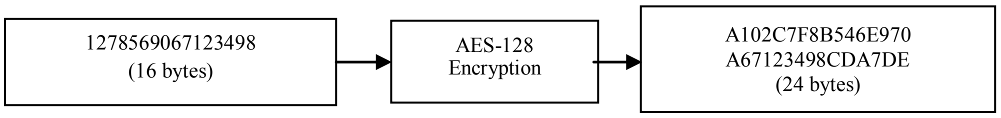
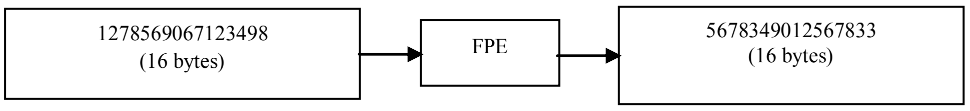


FIGURE 2: CREDIT CARD ENCRYPTION USING FPE



Special care should be taken to implement FPE algorithms. Since the attacker knows the format and data type of the plaintext. The length of plaintext and cipher text is also same.

IV. USES OF FPE

FPE can encrypt all type of data.
Reduce changes to database or application schemas. The data are suitable to the existing data base structure.
It provides referential integrity.
FPE can encrypt primary and foreign keys.

V. FPE AND CLASSICAL ENCRYPTION TECHNIQUES

Most of our earlier classical encryption techniques support FPE. Two basic methods of classical encryption are substitution and transposition.

5.1 Substitution cipher

In this technique each element of the plaintext is replaced by another element. The substitution ciphers such as Caesar cipher, Mono alphabetic cipher, Poly alphabetic cipher etc, were retaining the format and data type of plaintext. In most of the classical cipher the length and format of plaintext and cipher text are same.

5.2 Transposition cipher

This cipher is also called as permutation cipher. In this cipher the plaintext is shuffled without changing the original letters. So that the length and elements of both plaintext and cipher text are same. Only the position of the characters is changed. The historical ciphers could easily break because the computational functions were not complicated. The historical ciphers were only basic to the modern ciphers.

VI. FPE AND MODERN CIPHERS

The modern ciphers are mainly based on Symmetric key encryption or Asymmetric key encryption. In symmetric key encryption the unique key is used for both encryption and decryption. But in Asymmetric two separate keys such as public key and private keys are used. DES, Triple DES, AES, Blowfish and Feistel network are best example for modern ciphers. The computational functions of modern ciphers are more complicated and cannot be easily broken.

The main draw back in modern ciphers is the length of the cipher text. For example encrypting a single digit number using AES and 128 bit key, the cipher text is 32 digits hexadecimal number. To store 32 digits hexadecimal number, 128 bits are required. The cost of altering the database structure is very high. The queries related to the data base will also be changed. The graphical user interface could not support an encrypted text.

VII. EXISTING FPE TECHNIQUES

Black and Rogway suggested three practical methods for FPE such as Prefix method, Cycle walking method and Feistel network. [2]

7.1. Prefix Method

The Prefix method is based on either AES or 3DES algorithm. For example encrypting 16 digits credit card number applying AES algorithm to each digit and store the encrypting values in the table. The table is sorted according to the encrypted text and the corresponding original digits are used as a cipher text. The technique is useful only for small size of plaintext.

One enhancement of Prefix method is storing only half of the encrypted value for each digit. The table contains only first 16 hex numbers instead of storing 32 hex numbers. But still this method requires large amount of memory to store the tables.

7.2 Cycle walking Method

The cycle walking is implemented by encrypting the plaintext repeatedly applying AES or 3DES until the cipher text becomes in acceptable range. The duration for ciphering cannot be calculated. Too much iteration is required for each encryption process.

7.3 Feistel and cyclic method.

The Feistel + Cycle construction is the combination of two main methods. First, the Feistel network that is constructed for the size of the given plaintext. This network used to encrypt the data. The cycle-walking technique is applied to the cipher text to provide the cipher text in appropriate range. The performance is based on the number of rounds used in the network. One more restriction in this method is the Feistel network is nearest to the size of the plaintext.[3]

VIII. EFFICIENT FPE ALGORITHM

The proposed algorithm is based on AES-128 encryption algorithm. The 16 digits credit card number is encrypted using AES-128. At the end of the AES – 128 algorithm two more steps are added to get 16 digits plaintext.

8.1 AES Parameters and Keys

AES-128 uses 128 bit plaintext and 128 bit key to produce128 bit cipher text within 16 rounds. The number of possible keys are 2^{128}=3.4 X 10^{38}. A PC that tries 2^{55} keys per second requires 149 billion years to break the cipher text.[3]

8.2 Structure of each Round

In AES encryption each round contains four main transformations.

i. Sub bytes
ii. Permutation
iii. Column mixing
iv. Adding key

8.2.1 Sub bytes

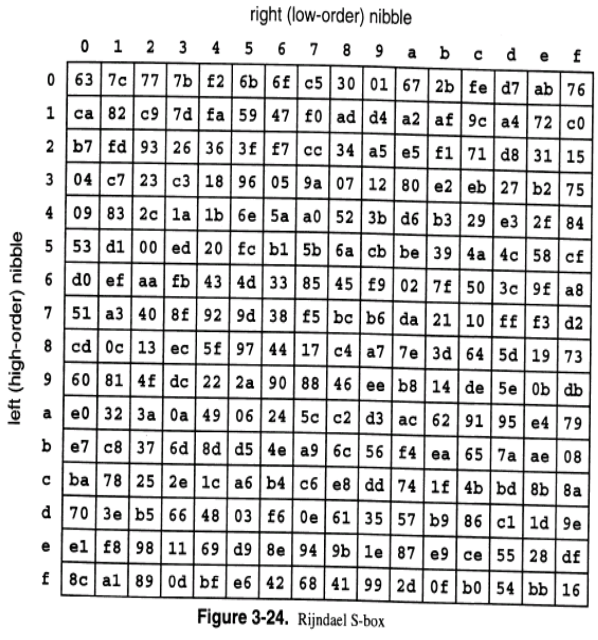
AES operates on a 4×4 column major order matrix of bytes, called as *state.* AES 128 block is represented as 4x4 state array.

FIGURE 3: AES state array

b0	b1	b2	b3	b4	b5	b6	b7	b8	b9	b10	b11	b12	b13	b14	b15

b0	b4	b8	b12
b1	b5	b9	b13
b2	b6	b10	b14
b3	b7	b11	b15

In the first step, each byte $a_{i,j}$ in the *state* matrix is replaced with a SubByte $S(a_{i,j})$ using the Rijndael S-box. The first transformation, SubBytes, is used at the encryption site. In order to substitute a byte, we interpret the byte as two hexadecimal digits. The SubBytes operation involves 16 independent byte-to-byte substitutions.[4]



right (low-order) nibble

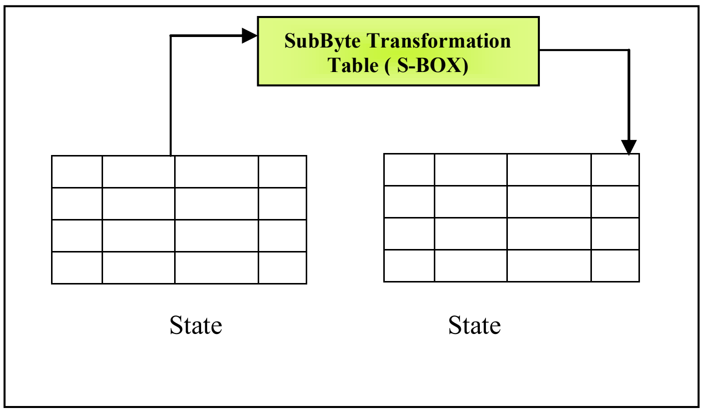
left (high-order) nibble	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Figure 3-24. Rijndael S-box

TABLE 1 : S-BOX

In state array the first four bytes are used to identify the row and lower order four bytes are used to identify the column. For example the number 70 is state array is substituted by 51(7 th row and 0 th column). Similarly all the sixteen elements in the state array are substituted by another sixteen elements using this S-BOX.

FIGURE 4: SubByte Transformation

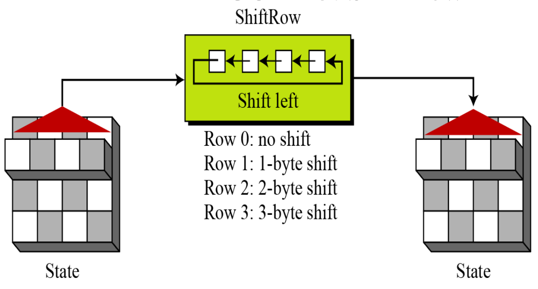


The S-box is generated by determining the multiplicative inverse for a given number in $GF(2^8) = GF(2)[x]/(x^8 + x^4 + x^3 + x + 1)$, Rijndael's finite field. The SubBytes and InvSubBytes transformations are inverses of each other.

8.2.2 Permutation

Shifting rows is used to permutes the bytes. In the encryption, the permutation is called ShiftRows. The first row is never shifted. Each byte of the second row is shifted one to the left. The third and fourth rows are shifted by offsets of two and three respectively. Row n is shifted circular left by n-1 times.

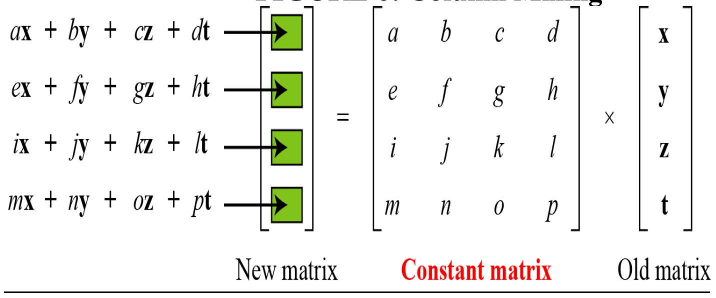
FIGURE 5: ShiftRow



8.2.3 Column Mixing

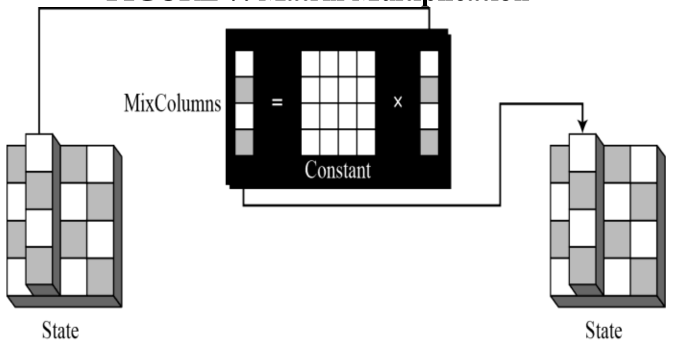
The MixColumns function takes four bytes as input and produce four bytes as output. In column mixingre each input byte affects all four output bytes. Together with ShiftRows, MixColumns provides diffusion process in the cipher. We need an interbyte transformation that modifies the bits inside a byte, based on the bits inside the nearby bytes. To mix bytes to provide diffusion at each bit level.

FIGURE 6: Column Mixing



The MixColumns transformation operates at the column level; it transforms each column of the state to a new column.

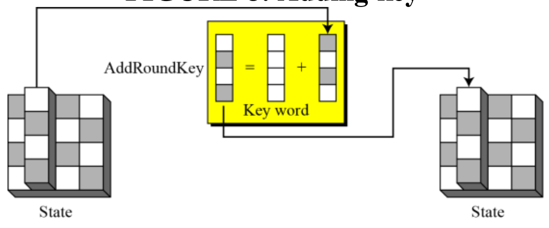
FIGURE 7: Matrix Multiplication



8.2.4 Adding key

AddRoundKey proceeds one column at a time. AddRoundKey adds a round key word with each state column matrix; Matrix addition operation is used in AddRoundKey. The subkey is added by combining each byte of the state with the crelated byte of the subkey using bitwise XOR. The AddRoundKey transformation can also be

FIGURE 8: Adding key



8.2.5 Additional steps for proposed FPE Algorithm

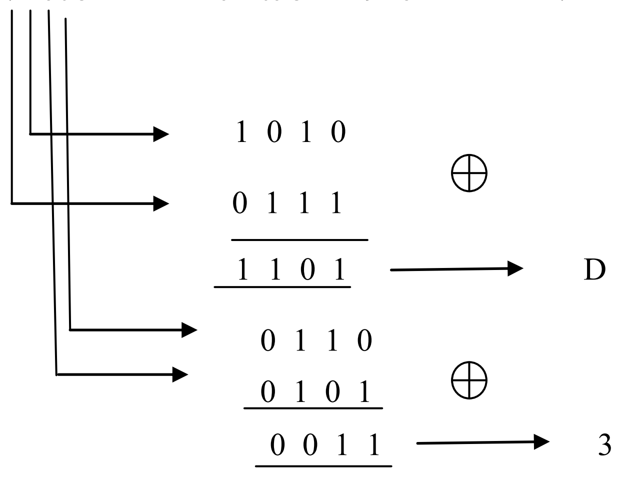
At the end of the sixteenth round two more additional steps are added to retain the format and data type of the plaintext.

8.2.5.1 EX-OR operation

The resultant 128 bit text is divided into group of 8 bits. Totally there are 16 8-bit blocks. For each group we perform EX-OR operation. The higher ordered four bytes are EX-ORed with lower order four bytes. At the end of this step we get 64 bit block.[5]

For example at the end of the sixteenth round the 16 digits credit card number is encrypted as 128 bit cipher text.

7A 56 3F 4E 22 4B 5A 65 32 4D 9F 84 A2 FB DD 7B



Similarly, Applying the EX-OR operation to the remaining groups we get 8 hexa digits as a result.
D 3 C A 0 F F 3 1 9 6 C 8 4 0 C

8.2.5.2 Conversion

In the last step applying 2421 coding to the hexa digits we get the exact 16 decimal digits and each digit is in the valid range of 0 to 9. An ordinary hexa to decimal conversion cannot be applied here. In hex the letters A TO F represents 2two digits number. Instead of ordinary conversion we apply 2421 decimal conversion to get the valid decimal digits.

At the end of the sixth step we get the following 16 digits number as a ciphertext.

D ---------- 1 1 0 1 ---------- 2 + 4 + 0 + 1 ---------- 7
3----------- 0 0 1 1 -----------0 + 0 + 2 + 1----------- 3
C ---------- 1 1 0 0 ---------- 2 + 4 + 0 + 0 ---------- 6
A ---------- 1 0 1 0 ---------- 2 + 0 + 2 + 0 ---------- 4
0 ---------- 0 0 0 0 ---------- 0 + 0 + 0 + 0 ---------- 0
F ---------- 1 1 1 1----------- 2 + 4 + 2 + 1---------- 9
F ---------- 1 1 1 1----------- 2 + 4 + 2 + 1 --------- 9
3 ---------- 0 0 1 1----------- 0 + 0 + 2 + 1 -------- 3
1 ---------- 0 0 0 1----------- 0 + 0 + 0 + 1 -------- 1
9 ---------- 1 0 0 1----------- 2 + 0 + 0 + 1 -------- 3
6 --------- 0 1 1 0----------- 0 + 4 + 2+ 0 -------- 6
C --------- 1 1 0 0----------- 2 + 4 + 0 + 0 -------- 6
8 --------- 10 0 0------------ 2 + 0 + 0 + 0 ------- 2
4 ---------- 0 1 0 0----------- 0 + 4 + 0 + 0 -------- 4
0 --------- 0 0 0 0----------- 0 + 0 + 0 + 0 -------- 0
C --------- 1 1 0 0----------- 2 + 4 + 0 + 0 -------- 6

The 16 digits cipher text is **7 3 6 4 0 9 9 3 1 3 6 6 2 4 0 6**

At the end of the sixth step the plaintext and cipher text are same in format and data type. No need to change the database structure, queries and application programs to handle this cipher text.

IX. PERFORMANCE ANALYSIS OF PROPOSED SYSTEM

IX.1 Comparitive study

The following table represents the comparative study of existing FPE techniques and Proposed FPE technique.[6]

TABLE 2 :Comparitive Study

Constraints	**Existing FPE techniques**			**Proposed FPE technique**
	Prefix	**Cycle walking**	**Feeistel and cyclic**	
Input plaintext size	Small range	Small range	Closest to the Feistel network	No limitations
Number of iterations	Number of digits in plaintext	Not deterministic	Not deterministic	One
Storage requirements	Additional storage to hold the tables	Additional storage to store random keys for each iteration	Additional storage to store random keys for each iteration	No need for any additional storage
Implementation	Sorting is required for each encryption	Repeated encryption	Generating pseudo random function(prf) for each iterations	Simple ex-or and conversion function.

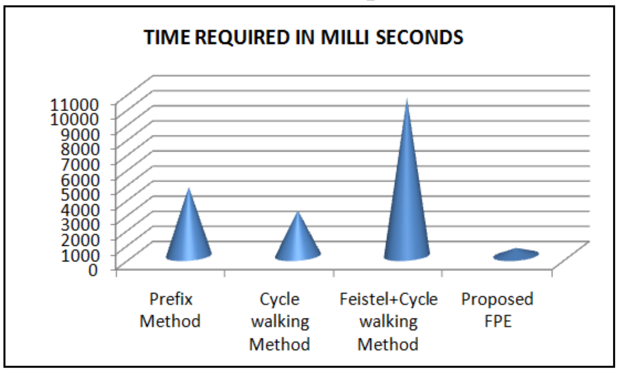
IX.2 Time Requirements

The following table shows the performance of the existing FPE techniques and Proposed FPE on a 2.34 Ghz Pentium IV with 1 GB memory running Microsoft Windows XP Professional to encrypt 128 bit plaintext. There is no iterations and table build time in proposed FPE technique. So it requires limited time for encryption.[7]

TABLE 3: Time Requirements

FPE TECHNIQUES	TIME REQUIRED IN MILLI SECONDS
Prefix Method	4560
Cycle walking Method	3000
Feistel+Cycle walking Method	10500
Proposed FPE	500

FIGURE 9:Time Requirements



X. CONCLUSION

The proposed FPE algorithm is very useful for real time applications such as encrypting credit card number. This algorithm retains the format and data type of plaintext after encryption. The database structure, queries and application programs never changed after encryption. In future try to apply this algorithm for all the data types not only for numeric data type. Try to making some changes in AES technique to implement FPE instead of adding extra work after the encryption process.

REFERENCES

[1] Format Preserving Encryption Terence SpiesVoltage Security, Inc.
[2] M. Bellare, T. Ristenpart, P. Rogaway, and T. Stegers. Format-preserving encryption. SAC 2009. LNCS 5867,Springer, 2009.
[3] V. Hoang and P. Rogaway. On generalized Feistel networks. Conference version of this paper. CRYPTO 2010,Springer, 2010.
[4] Information security management Hnadbook, volume 6 by Harold F.Tipton, Micki Krause nozaki
[5] A new integer FPE scheme based on Feistel Network (Jia, Z Liu, J Li, Z Dong, X you advances in Electric and Electronic, 2012 Springer.
[6] M. Bellare, P. Rogaway, and T. Spies. The FFXmode of operation for format-preserving encryption(Draft 1.1). February, 2010. Manuscript (standards proposal) submitted to NIST.
[7] Performance Comparison of the AES Submissions byBruce Schneier, John Kelsey, Doug Whiting, David Wagner, Chris Hall, Niels Ferguson