CRYPTOGRAPHY BASICS

NAME:
SNEHA MAGANAHALLI RAJENDRANATH.

ROLL NO: CSR1M522

ASSIGNMENT NO: 1.

What is format preserving Algorithm / Encryption?

Answer:

Format preserving Encryption [FPE] refers to encrypting in such a way that the output (ciphertext) is in the same format as the input (the plaintext)

Eg:

① 16 digit credit card number is encrypted to give another 16 digit number.

② Encrypting an english word so that the ciphertext is another English word.

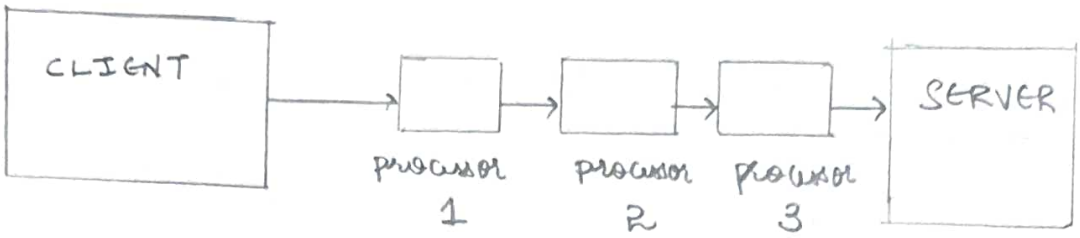
Why format preserving Encryption is needed?

① COBOL Applications:
A small change in the structure of the record, then there will be huge change all over the modules.

② Database Applications:
They take only character strings.

③ Compression:
Since format is preserved, compression is easy.
It is difficult for AES, DES since format is not preserved

(9) Credit card encryption.



Steps:

i) person will swipe his credit card at client.

ii) The processor 1, processor 2, processor 3 will process it. These processors expect the input in the form of credit card number only. Hence format preserving Encryption is useful.

iii) If we did-not implement FPE, then we need to change the software on all the processors. Now with the help of FPE, we can change the software only on client and server.

P.T.O.

Let us use FFI algorithm to encrypt and decrypt the credit card numbers.

Encryption Algorithm :-

1. Let $u = \lfloor n/2 \rfloor$ $v = n - u$
2. Let $A = X[1 \cdots u]$ $B = X[u+1 \cdots n]$
3. Let $b = \lceil \lceil v \cdot \log_2(\text{radix}) \rceil / 8 \rceil$
4. Let $d = 4\lceil b/4 \rceil + 4$
5. Let $P = [1]^1 \| [2]^1 \| [1]^1 \| [\text{radix}]^3 \| [10]^1 \| [u \bmod 256]^1 \| n^4 \| [t]^4$
6. For i from 0 to 9.
   i) Let $Q = T \| [0]^{(-t-b-1) \bmod 16} \| [i]^1 \| [\text{NUM}_{\text{radix}}(B)]^b$
   ii) Let $R = \text{PRF}(P \| Q)$
   iii) Let S be the first d bytes of the following string of $\lceil d/16 \rceil$ blocks:

$R \| \text{ciph}(R \oplus [1]^{16}) \| R \| \text{ciph}(R \oplus [2]^{16}) \cdots \cdots \text{ciph}(R \oplus [\lceil d/16 \rceil - 1]^{16})$

   iv) Let $y = \text{NUM}(S)$
   v) If i is even, let $m = u$; else let $m = v$
   vi) Let $c = (\text{Num}_{\text{radix}}(A) + y) \bmod \text{radix}^m$
   vii) Let $C = \text{STR}^m_{\text{radix}}(c)$
   viii) Let $A = B$
   ix Let $B = C$.
7. Return $A \| B$.

FF1 Decryption Algorithm:

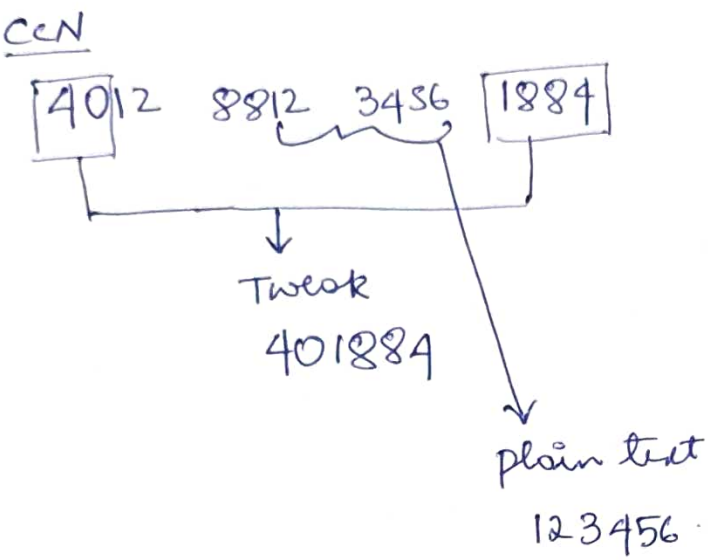
It has a similar approach as encryption algorithm

1. Let $u = \lfloor n/2 \rfloor$; $v = n - u$.
2. Let $A = X[1 \cdots u]$; $B = X[u+1 \cdots n]$
3. Let $b = \lceil \lceil v \log_2(radix) \rceil / 8 \rceil$
4. Let $d = 4\lceil b/4 \rceil + 4$
5. Let $P = [1]^1 \,||\, [2]^1 \,||\, [1]^1 \,||\, [radix]^3 \,||\, [10]^1 \,||\, [u \bmod 256]^1 \,||\, n^4 \,||\, t^4$
6. For $i = 9$ to 0.
   i) $Q = T \,||\, 0^{(-t-b-1) \bmod 16} \,||\, [i]^1 \,||\, [NUM_{radix}(A)]^b$
   ii) Let $R = PRF(P||Q)$
   iii) Let S be the string of the first d bytes of the following string $[d/16]$ blocks:
   $R \,||\, ciph_K(R \oplus [1]^{16}) \,||\, Ciph_K(R \oplus [2]^{16}) \cdots ciph_K(R \oplus [\lceil d/16 \rceil - 1]^{16})$.
   iv) Let $y = NUM(S)$
   v) If i is even, let $m = u$; else let $m = v$
   vi) Let $c = (NUM_{radix}(B) - y) \bmod radix^m$
   vii) Let $C\ STR^m_{radix}(c)$
   viii) Let $B = A$
   ix) Let $A = C$
7. Return $A \,||\, B$.

How credit card encryption is done??

Answer:

Let us consider an example.



plaintext + Tweak.

$$\begin{array}{r} 123456 \\ +\ 401884 \\ \hline 524230 \end{array}$$ ---- If the number exceeds 10, then apply mod 10

Why tweak is necessary?

If tweak is not present, then Eve can map the credit card number (plain text) to cipher text. He can maintain the record of it. Because many credit cards will have the same middle numbers.

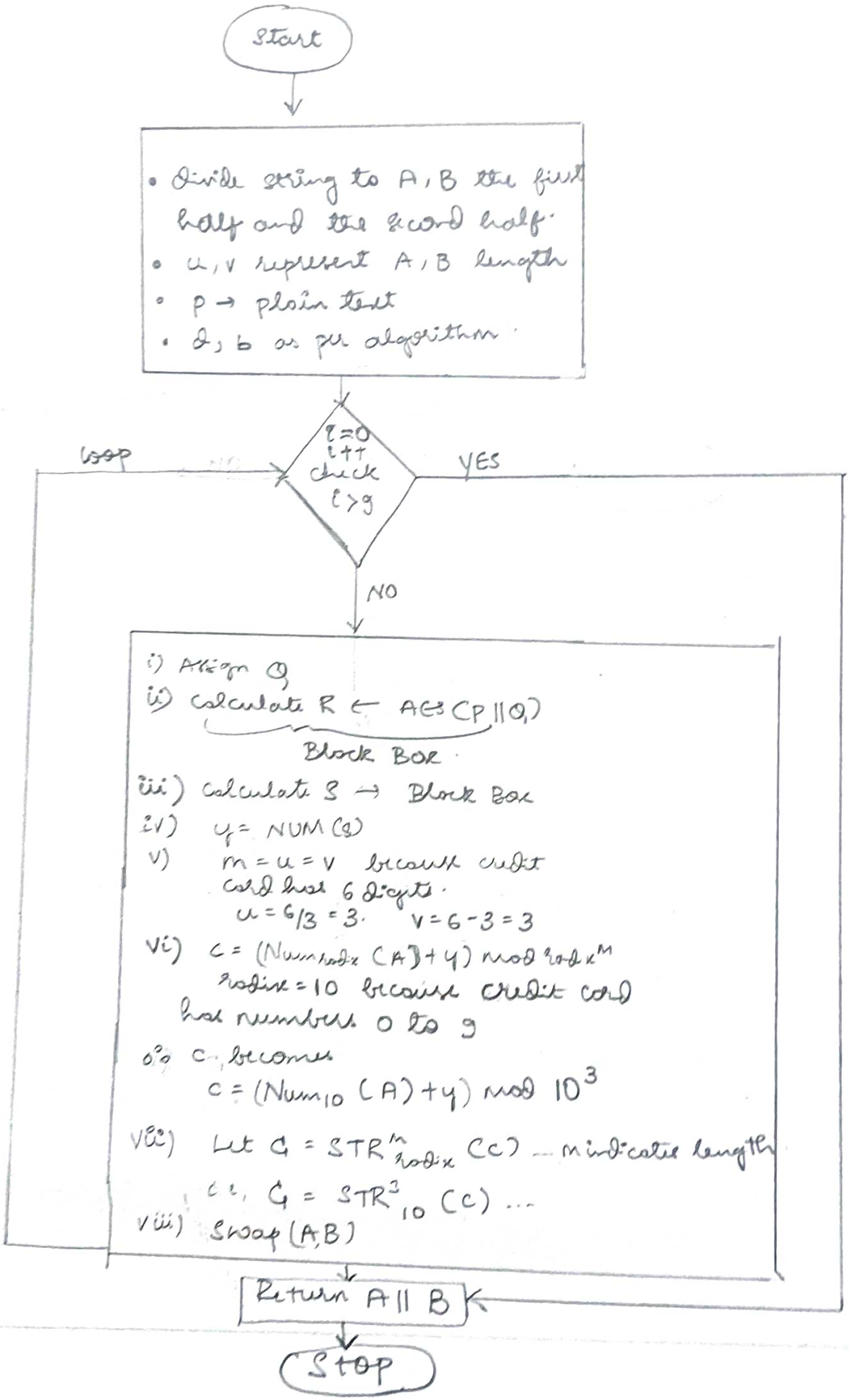
Why only 6 digits are encrypted?

This is because other digits represent the issuer identification number.

Only 6 digits represent the user account number.

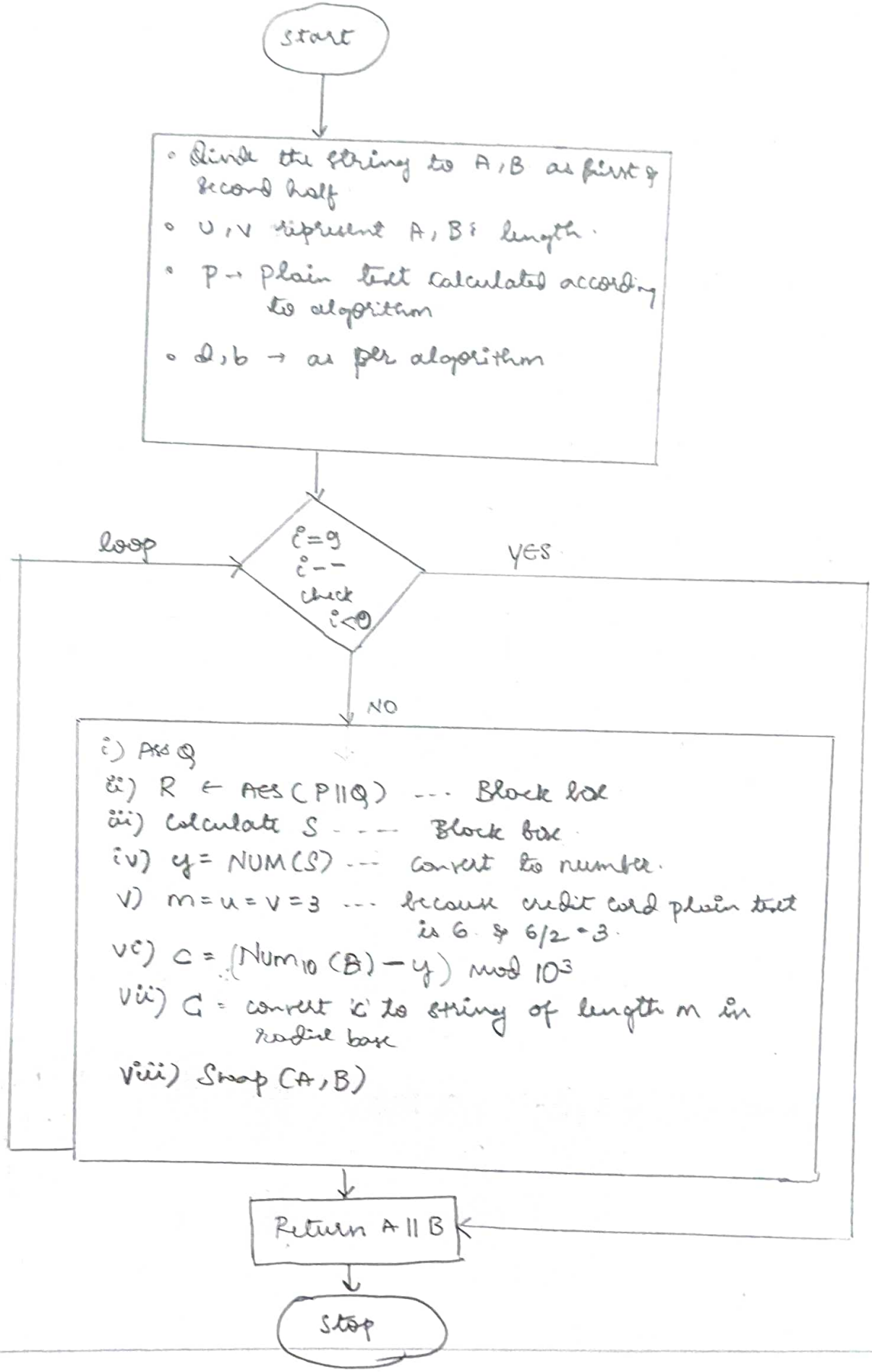
Flow chart:

Encryption:



Flowchart

Decryption:



Let us take an example and trace the algorithm.

Key: 2b 7e 15 16 28 ae d2 a6 ab f7 15 88 09
cf 4f 3c .... this is AES key (32 digits)

plain text: ccn: 12 3456 7890 12 3456

Solution:

Twtak = 123456

plaintext = 789012

Twtak + plain text

802468

Encryption Algorithm trace:

1. $u = \lfloor \frac{n}{2} \rfloor = \lfloor \frac{6}{2} \rfloor = 3$

$v = n - u$
$= 6 - 3$
$v = 3$

2. A 802
B 468] ..... "802468" we have to encrypt

3. $b = \lceil \frac{v \lceil \log_2(\text{radix}) \rceil}{8} \rceil$

$b = \lceil \frac{3 \lceil \log_2 10 \rceil}{8} \rceil$

$= \lceil \frac{3 \lceil 3.32 \rceil}{8} \rceil$

$= \lceil 3 \lceil 4 \rceil / 8 \rceil$

$= \lceil 12/8 \rceil$

$= \lceil 1.5 \rceil$

$b = 2$

4. $d = 4 \lceil b/4 \rceil + 4$

$= 4 \lceil 3/4 \rceil + 4$

$= 4 \lceil 0.75 \rceil + 4$

$= 4 \cdot 1 + 4$

$= 8$

5.

P[0] = [1]' = 1

P[1] = [2]' = 2

P[2] = [1]' = 1

P[3] = 0 ⎫
P[4] = 0 ⎪
P[5] = 10 ⎪
P[6] = 10 ⎭ ..... = port << 8 | 10
= 10 << 8 | 10
= 2560 | 10
= 2570

↓

2570 is the ans in Big Endian notation.

Convert 2570 to little Endian form

P[3] = (2570 >> 24) & FF = 0

P[4] = (2570 >> 16) & 0XFF = 0

P[5] = (2570 >> 8) & 0XFF = 10 & 0XFF = 10

P[6] = (2570 >> 0) & 0XFF = 0XA0A & 0XFF = 10

P[7] = 6 MOD 256
= 3 MOD 256
= 3

(credit card len) in little Endian form

P[8] = 0 ⎫ ... (6 >> 24) & 0XFF = 0
P[9] = 0 ⎪ (6 >> 16) & 0XFF = 0
P[10] = 0 ⎪ (6 >> 8) & 0XFF = 0
P[11] = 6 ⎭ (6 >> 0) & 0XFF = 6

P[12] = 0 ⎫ ... [tweak length / 2] in little Endian
P[13] = 0 ⎪ 6/2 = 3 in little Endian
P[14] = 0 ⎪
P[15] = 3 ⎭ ... (3 >> 0) & 0XFF = 3

Consider loop for 10 times

when $i = 0$

(c)

Q Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Q Value	18	52	86	0	0	0	0	0	0	0	0	0	0	0	1	212

52 86 → tweak in BigEndian

0 0 0 0 0 0 0 0 0 0 → $(-t - b - 1) \bmod 16$

$(-3 - 2 - 1) \bmod 16$

$-6 \bmod 16$

$10 \bmod 16$

$\underline{10}$

0 → i Value

1 212 → "468" — B in BigEndian

↓

We have to place 10 '0's

(ii) $R = PRF(P \| Q)$

$= AES(P \| Q)$

$= BlockBox$

(iii) S is also a Block Box.

(iv) S in number format is Y

$Y = 78967442033607605O1$

(v) $m = u = v = 3$

(vi) $C = (NUM_{radix}(A) + y) \bmod radix^m$

$= (802 + 78967442033607605O1) \bmod 10^3$

$= 78967442033607613O3 \bmod 1000$

802 468" → Tweak + plain text

$C = \underline{303}$

(vii) c in string form

$C = "303"$

viii) Let A = B

A = "802" "468" ....

ex) Let B = C

∴ B = "303" ....

"802468"

↓ A ↓ B

∴ C = "305"

#) Repeat the loop for 10 time in the same way and return A||B.

Decryption tracing:

ciphered string : 453284

(1) $u = \left[\frac{n}{2}\right] = \lfloor 6/2 \rfloor = 3$

$v = n - u$

$= 6 - 3$

$= 3$

(2) A = 453
B = 284] .... "453284" ÷ as 2 halfs

(3) $b = \lceil [v \log_2(\text{radix})] / 8 \rceil$

(4) b = 2 ---- By proof of encryption trace.

d = 8 --- By proof of encryption trace

(5)

P index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
P value	1	2	1	0	10	10	0	03	0	0	0	6	0	0	0	3	

---- By proof of encryption trace.

Consider loop of 10 time.

i = 9 to 0

Consider i = 9

(i)

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
value	18	52	86	0	0	0	0	0	0	0	0	0	0	9	1	97

i value A = "453" in Big Endian

by encryption trace.
It is same as encryption trace.

(ii) $P = PRF(P \| Q)$
$= AES(P \| Q)$ --- Block Box

(iii) S is also a block box.

(iv) S in number format is y

$y = 181052021603754998862$

(v) $m = u = v = 3$.

(vi) $c = (NUM_{radix}(B) - y) \bmod radix^m$
$= (284 - 181052021603754998862) \bmod 10^3$
$= \underline{422}$

--- B is "284" i.e; 483 284

(vii) c in string form

$C = 422$

(viii) $A = B$
$\therefore A = '284'$

(ix) $B = C$
$\therefore B = 422$ --- By (vii)

Repeat the loop for 10 times and return A || B.

∴ plain text : 789012

cipher text : 453284

[ccn was] : 123456 [789012] 3456

[Encrypted text ccn] : 123456 [453284] 3456 (replace)