**Lecture 4: Irreducible Polynomials in AES**

**LEARNING OUTCOME**

**By the end of the lesson the student will be able to:**

1. understand a concept of irreducible polynomial in AES
2. to multiply two polynomials
3. compute a matrix multiplication in AES.
4. Compute one round of AES

A ring over an irreducible polynomial has been used in modern cryptosystem, namely, ECC, AES and NTRU. In AES algorithm, this irreducible polynomial is

=1000110112 = or {01}{1B} in hexadecimal notation.

In the S-box of AES, take the multiplicative inverse in the finite field GF(28) first where

element {00} is mapped to itself {00}.

Let us take  from the top left corner of the S-box and then 

One more time, Let us take  from the bottom right corner of the S-box and then we need to take multiplicative inverse first modulo the irreducible polynomial =1000110112



Let us give an overview of multiplication between two bytes.

{57} • {83} = {C1}

= (0101 0111)⋅(1000 0011) written little endian

10000011

10000011

10000011

10000011

10000011

= 101001121101221 mod 2 = 10101101101001

The convolution will result in

(0101 0111)⋅(1000 0011)=10101101101001

In polynomial, it is written as

 mod 

=10101101101001

100011011

= 100000011001

100011011

= 11000001=C1.

Let us review the mix-column operation in AES encryption. At a certain round, let the state

 and the mix-column matrix .

It might be a good idea to write the matrix side by side.



The whole mix-column operation is a matrix multiplication



=

And now, we can see the inverse mix column during the decryption process,



=

Let take an example from the top left corner,



|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Round** | **Start of Round** | | | |  | **After SubBytes** | | | |  | **After ShiftRows** | | | |  | **After Mix Columns** | | | |  | **Round Key Value** | | | |
| **0**  **input** | **00** | **44** | **88** | **CC** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | **⊕** | **00** | **04** | **08** | **0C** |
| **11** | **55** | **99** | **DD** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | **01** | **05** | **09** | **0D** |
| **22** | **66** | **AA** | **EE** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | **02** | **06** | **0A** | **0E** |
| **33** | **77** | **BB** | **FF** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | **03** | **07** | **0B** | **0F** |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **1** | **00** | **40** | **80** | **C0** |  | **63** | **09** | **CD** | **BA** |  | **63** | **09** | **CD** | **BA** |  | **5F** | **57** | **F7** | **1D** | **⊕** | D6 | D2 | DA | D6 |
| **10** | **50** | **90** | **D0** |  | **CA** | **53** | **60** | **70** |  | **53** | **60** | **70** | **CA** |  | **72** | **F5** | **BE** | **B9** | AA | AF | A6 | AB |
| **20** | **60** | **A0** | **E0** |  | **B7** | **D0** | **E0** | **E1** |  | **E0** | **E1** | **B7** | **D0** |  | **64** | **BC** | **3B** | **F9** | 74 | 72 | 78 | 76 |
| **30** | **70** | **B0** | **F0** |  | **04** | **51** | **E7** | **8C** |  | **8C** | **04** | **51** | **E7** |  | **15** | **92** | **29** | **1A** | FD | FA | F1 | FE |

From the above standard sample,

The whole mix-column operation is a matrix multiplication



=

And now, we can see the inverse mix column during the decryption process,



=

Let take an example from the top left corner,



**Lab Test 4: One round of AES~LT (10%) –C3 PO2**

Do the initial and first round of AES Encryption on the string plaintext M using a given symmetric key K. Take the plaintext M as the first 16 character of your name instead. You are also given symmetric key K written in hexadecimals. Compute for full Round 1 until Initial Round 2 State Array.