AES

Piotr Karamon

March 24, 2024

Contents

1 Basic information

- Advanced encryption standard.
- The more general algorithm is called rijndael.
- 128 bit cipher (takes 128 bits and returns 128 bits of cipher text)
- in the ciphering process a key is used
- this key is (128|192|256) bits long
- 128 bits is 16 bytes
- those 16 bytes are often though as a 4 by 4 grid

- AES is a SP network (meaning we have some substitution and some permutation)
- it is in implmeneted in hardware (Intel, Amd chips)

```
Key: randomly generated (that is the secret)
each [] represents a byte
Plain text:
[00][04][08][12]
[01][05][09][13]
[02][06][10][14]
[03][07][11][15]
```

2 Steps of the algorithm

1. We first do part of our Key xor Plain text
2. Substitute bytes.
3. We then shift rows
4. We mix the columns
5. Add Round key (The key is expanded into round keys) We do it for N rounds. Number of rounds depends on the key.

Key size in bits	N rounds
128	10
192	12
256	14

3 Key

Key is expanded using something called key schedule. Which takes the 128/192/256 bit key and expands it in a way that it can be used to add it in every round (addition-xor).

4 Finite fields(in a nutshell)

Say we have a list of all posible values (for example $[1-10]$). We also have some operations which we can do on those fields. The idea is that any of those operations return values which are in the predifined set (in this example results of all operations would be values $[1 - 10]$). AES using $GF(2^8)$ which is a Galois field of order 256. Thats a byte. Because a byte is $00000000 - 11111111$ There are 256 values in this range. Operations are:

- `+`
- `-`
- `*`
- `/` (which is really x^{-1})

It does not matter which operation you choose you will never leave the field. There are no overflows, underflows.

5 Substitution

It's just a lookup table mapping a byte to a different byte. It's a cleverly designed lookup table. It is very non linear. It is really quick (just a lookup).

Fixed point `x` is a value such that `x -> x`

Opposite fixed points all the bits get flipped

There are no fixed points meaning for example `(1 -> 1)`
There are no values such that `(1 -> 2, 2 -> 1)`
There are no opposite fixed points

6 Shifting rows

- first row -> do nothing
- second row -> move 1 to the left (wrap around)
- third row -> move 2 to the left (wrap around)
- forth row -> move 3 to the left (wrap around)

7 Mixing columns

It is done doing matrix multiplcation.

```
              some column   result
[2 3 1 1]     [c0]          [n0]
[1 2 3 1] *   |c1| =        |n1|
[1 1 2 3]     |c2|          |n2|
[3 1 1 2]     [c3]          [n3]
```

There is also an inverse matrix which done the opposite. It is used to decrypt back the data Add - xor Multiplication - Multiplication inside the finite field (modulo polynomial) Mixing columns is **not** done in the last round.

8 More Finite fields

Say we have a number $b = 10100110$ We would encode it as

$$1 * x^7 + 0 * x^6 + 1 * x^5 + \cdots + 1x + 0$$

A presence of 1 at position i means we add $1 * x^i$ to the polynomial.

Addition in this field means adding the polynomials - xor

$$(x^7 + x) + (x^7 + x^2) = (1 + 1)x^7 + x^2 + x = x^2 + x$$

$2x^7$ gets cancelled because we are dealing with binary

```
1010 0110 +
1000 0001 =
0010 0111
```

Addition - xor

Multiplication - multiplcation ($\mod (x^8 + x^4 + x^3 + x + 1)$)