How to solve MixColumns

Asked 7 years, 10 months ago Active 5 years, 11 months ago Viewed 50k times

22

12

I can't really understand MixColumns in Advanced Encryption Standard, can anyone help me how to do this?

I found some topic in the internet about MixColumns, but I still have a lot of question to ask.

ex.

$$\begin{bmatrix}\mathtt{d4}\\\mathtt{bf}\\\mathtt{5d}\\\mathtt{30}\end{bmatrix}\cdot\begin{bmatrix}\mathtt{02}&\mathtt{03}&\mathtt{01}&\mathtt{01}\\\mathtt{01}&\mathtt{02}&\mathtt{03}&\mathtt{01}\\\mathtt{01}&\mathtt{01}&\mathtt{02}&\mathtt{03}\\\mathtt{03}&\mathtt{01}&\mathtt{01}&\mathtt{02}\end{bmatrix}=\begin{bmatrix}\mathtt{04}\\\mathtt{66}\\\mathtt{81}\\\mathtt{e5}\end{bmatrix}$$

Here, the first element is calculated as

$$(\mathtt{d4}\cdot\mathtt{02})+(\mathtt{bf}\cdot\mathtt{03})+(\mathtt{5d}\cdot\mathtt{01})+(\mathtt{30}\cdot\mathtt{01})=\mathtt{04}$$

First we will try to solve $\mathtt{d4}\cdot\mathtt{02}$.

We will convert $\mathtt{d4}$ to it's binary form, where $\mathtt{d4}_{16}=1101\,0100_2$.

$$\begin{aligned} \texttt{d4}\cdot\texttt{02} &= \texttt{1101 0100} \ll 1 && (\ll \text{is left shift, 1 is the number of bits to shift} \\ &= \texttt{1010 1000} \oplus \texttt{0001 1011} && (\text{XOR because the leftmost bit is 1 before shift} \\ &= \texttt{1011 0011} && (\text{answe} \end{aligned}$$

Calculation:

$$\begin{aligned} &\texttt{1010 1000} \\ &\texttt{0001 1011} \oplus \\ =&\texttt{1011 0011} \end{aligned}$$

The binary value of $\texttt{d4}$ will be XORed with $\texttt{0001 1011}$ after shifting if the left most bit of the binary value of $\texttt{d4}$ is equal to 1 (before shift).

My question is, what if the left most bit of the binary value is equal to 0, what do I XOR it with then? ex. $\texttt{01}_{16} = \texttt{0000 0001}_2$ ..?

aes block-cipher

edited Mar 14 '14 at 19:28
Richie Frame
11.1k 1 18 36

asked Apr 19 '12 at 17:04
goldroger
1,489 8 25 39

2 Please don't cross post John... – Maarten Bodewes ♦ Apr 19 '12 at 18:31

I am new here so I don't know what is cross post. – goldroger Apr 20 '12 at 16:05

1 @JohnPaulParreño: *Cross posting* means posting the same question to multiple newsgroups/sites/etc. In particular, posting the same question on multiple Stack Exchange sites (like with your question on Stack Overflow) is frowned upon, since it splits the audience in parts, each with different (and maybe conflicting answers). Welcome to Cryptography Stack Exchange, though. – Paŭlo Ebermann Apr 20 '12 at 20:12

Yeah, forgot to say welcome first, sorry for that :) – Maarten Bodewes ♦ Apr 20 '12 at 21:50

1 @JohnPaulParreño It is not the same site (obviously), but is run by the same company (Stack Exchange), has a partially overlapping community, and uses the same software (with some modifications, for example we have TeX formatting here). There are quite some more sites in the Stack Exchange Network, which you can find by clicking on the Stack Exchange button on the top left. – Paŭlo Ebermann Apr 21 '12 at 16:06

1 Answer

29

Well, it sounds like you're close.

The multiplications implicit within the MixColumns operation are $GF(2^8)$ multiplication operations, using the same field representation as they use in the inverse within the sbox.

However, because they're multiplying by the fixed constants $\texttt{1}$, $\texttt{2}$ and $\texttt{3}$, it's easier to implement than a general $GF(2^8)$ multiplication

Multiplying by 2 is what your question is about: it is equivalent to shifting the number left by one, and then exclusiving-or'ing the value 0x1B if the high bit had been one (where, in case you're wondering, the value 0x1B came from the field representation). And so, that is the answer to the question you asked; if the high bit was a zero, then you don't need to exclusive or anything (or equivalently, you exclusive-or in a 0x00 constant).

And, your next question would likely be "how do a multiply by 3"? Well, you can do that by multiplying by 2 (see above), and then exclusive-or-ing that with the original value, since $3 = 2 \oplus 1$. Or, in other words:

$$3 \times x = (2 \oplus 1) \times x = (2 \times x) \oplus x$$

So, once we've gotten the multiplication by 2 operation solved, the multiplication by 3 is solved as well.

Once you've multiplied all the vector elements, then you need to add them. Now, you might be tempted to add them modulo 256, but that'd be wrong. This "addition" operation is actually "exclusive-or". They've written it as $+$ because, in $GF(2^n)$ fields, exclusive-or is considered the addition operation; it acts an awful lot like an addition operation in conjunction with the multiplication operation.

So, if we look at the calculation:

$$(\mathtt{d4} \times \mathtt{02}) + (\mathtt{bf} \times \mathtt{03}) + (\mathtt{5d} \times \mathtt{01}) + (\mathtt{30} \times \mathtt{01})$$

- $\mathtt{d4} \times \mathtt{02}$ is $\mathtt{d4} << 1$, exclusive-ored with $\mathtt{1b}$ (because the high bit of $\mathtt{d4}$ is set), giving $\mathtt{b3}$;
- $\mathtt{bf} \times \mathtt{03}$ is $\mathtt{bf} << 1$ exclusive-ored with $\mathtt{1b}$ (because the high bit of $\mathtt{bf}$ is set) and $\mathtt{bf}$ (because we're multiplying by 3), giving us $\mathtt{da}$;
- $\mathtt{5d} \times \mathtt{01}$ is $\mathtt{5d}$, and $\mathtt{30} \times \mathtt{01}$ is $\mathtt{30}$.

Now, we add (rather, exclusive or) $\mathtt{b3}$, $\mathtt{da}$, $\mathtt{5d}$ and $\mathtt{30}$ together, and that gives us $\mathtt{04}$.

edited Apr 20 '12 at 20:24
Paŭlo Ebermann
20.7k 7 68 110

answered Apr 19 '12 at 18:15
poncho
105k 3 169 272

thanks poncho, if I have a another question, can I ask it to you? – goldroger Apr 20 '12 at 7:23

2 @John: If your other question could be considered a part of this question, you can edit your original question. Otherwise better post a new question on the site, so everyone (including poncho) has a chance of answering it (and learning from the answers). If necessary, link to this question from there. – Paŭlo Ebermann Apr 20 '12 at 7:38

How do you XOR the 4 values at the end together? Do you XOR the first pair, then the result with the third input, and that result XOR the fourth input? Or do you do them all in one go (i.e. output '1' if there is literally one '1' in all 4 of the inputs)? – Loic Verrall Oct 22 '17 at 11:16

1 @LoicVerrall: xor is both associative and commutative, and so you can use whatever order is convenient. However, if you decide to xor all 4 together at once, you don't "output '1' if there is 1 in all 4 inputs", instead, it's "output 1 if there is a 1 in an odd number of the numbers" – poncho Oct 22 '17 at 12:40

protect this question from spam and non-answer activity.