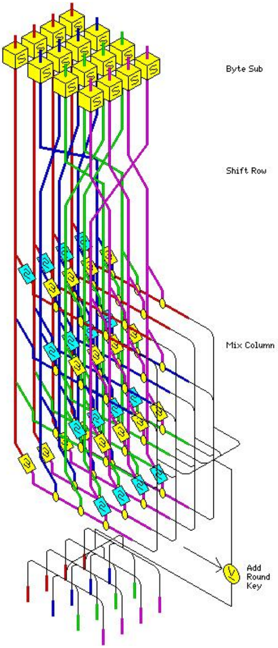


Image Source: Google

CS 553
CRYPTOGRAPHY

Lecture 12
The Story of AES Continues

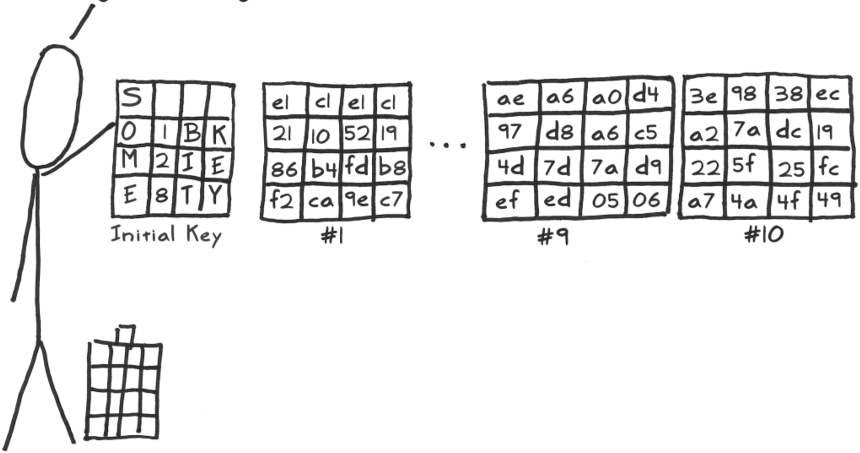
Instructor
Dr. Dhiman Saha

The Story So Far

- `DES` and Modern Crypto
- The issue with `DES`
- The need for `AES`
- Rijndael: The Winner of `AES`
- The Key Expansion
- The Round Function
- `SubBytes`
- `ShiftRows`

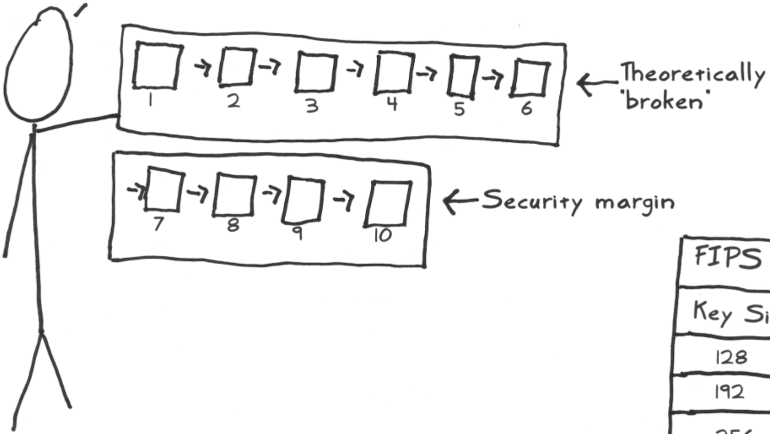
Key Expansion: Part 1

I need lots of keys for use in later rounds. I derive all of them from the initial key using a simple mixing technique that's really fast. Despite its critics,* it's good enough.



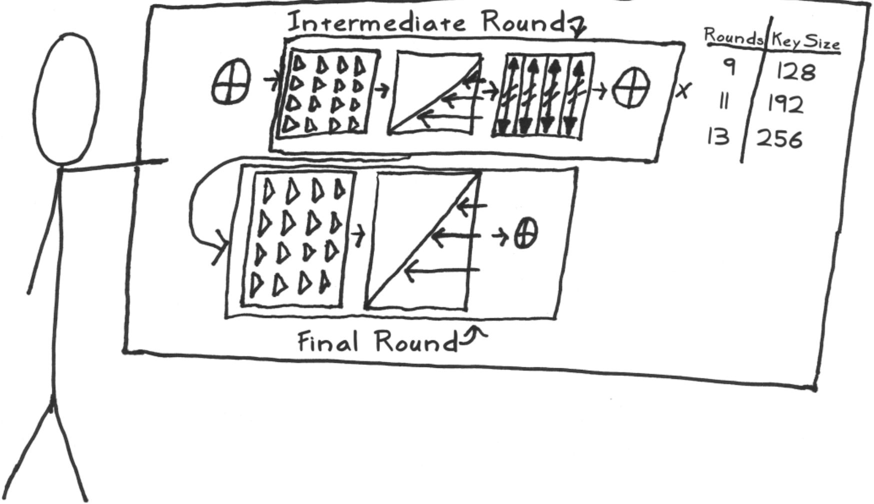
* By far, most complaints against AES's design focus on this simplicity.

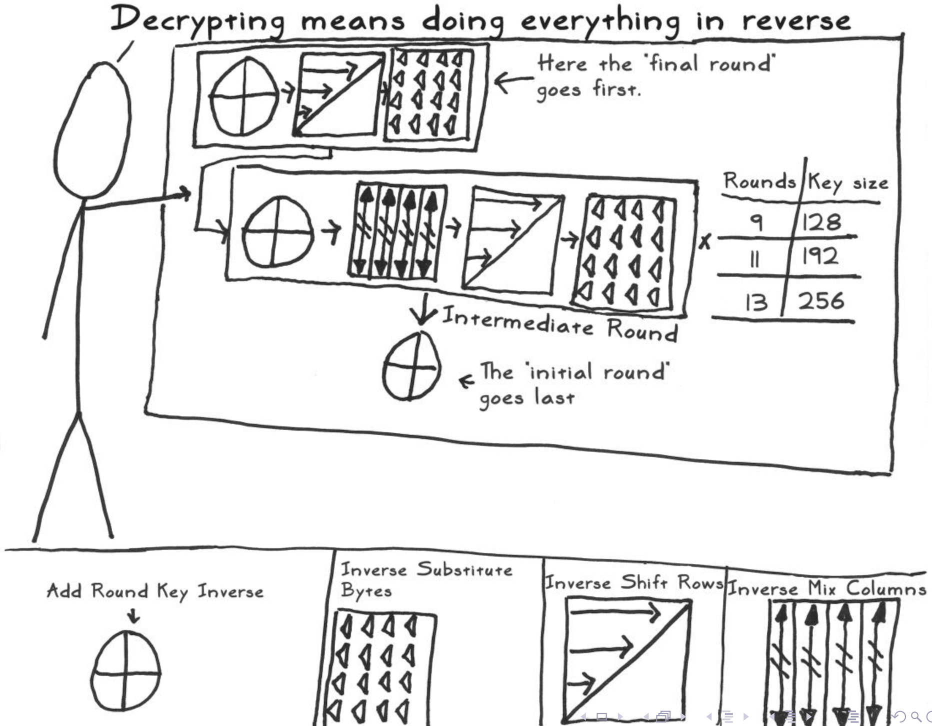
When I was being developed, a clever guy was able to find a shortcut path through 6 rounds. That's not good! If you look carefully, you'll see that each bit of a round's output depends on every bit from two rounds ago. To increase this diffusion "avalanche," I added 4 extra rounds. This is my "security margin."



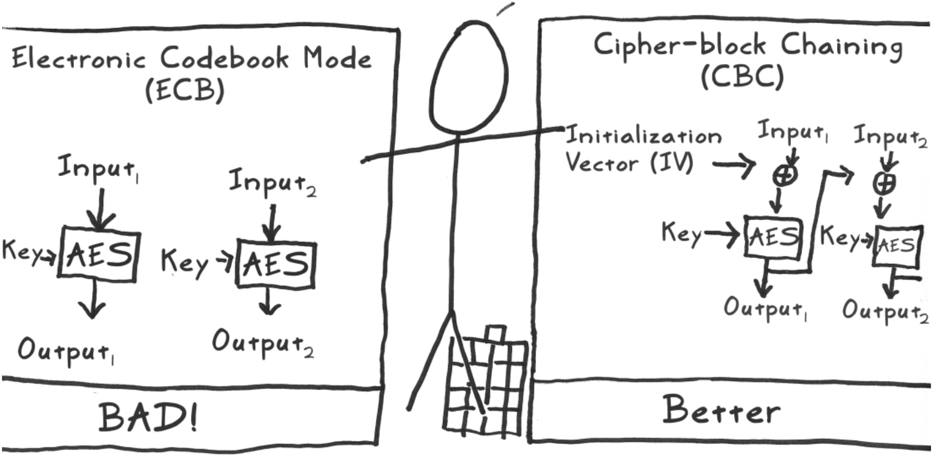
FIPS 197 Spec	
Key Size	Rounds
128	10
192	12
256	14

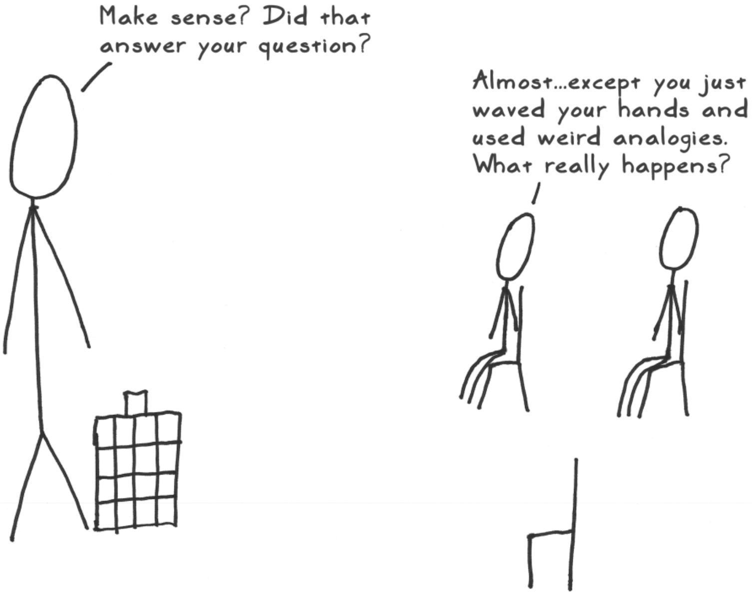
So in pictures, we have this:

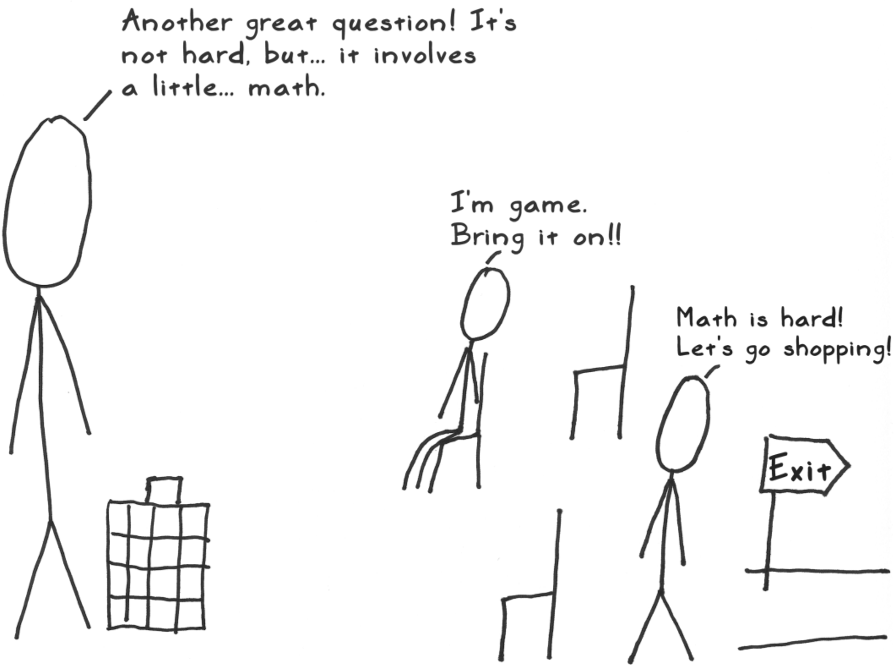



Decrypting means doing everything in reverse
Here the "final round" goes first.
Rounds Key size
9 128
11 192
13 256
x
Intermediate Round
The "initial round" goes last
Add Round Key Inverse
Inverse Substitute Bytes
Inverse Shift Rows
Inverse Mix Columns

One last tidbit: I shouldn't be used as-is, but rather as a building block to a decent "mode."

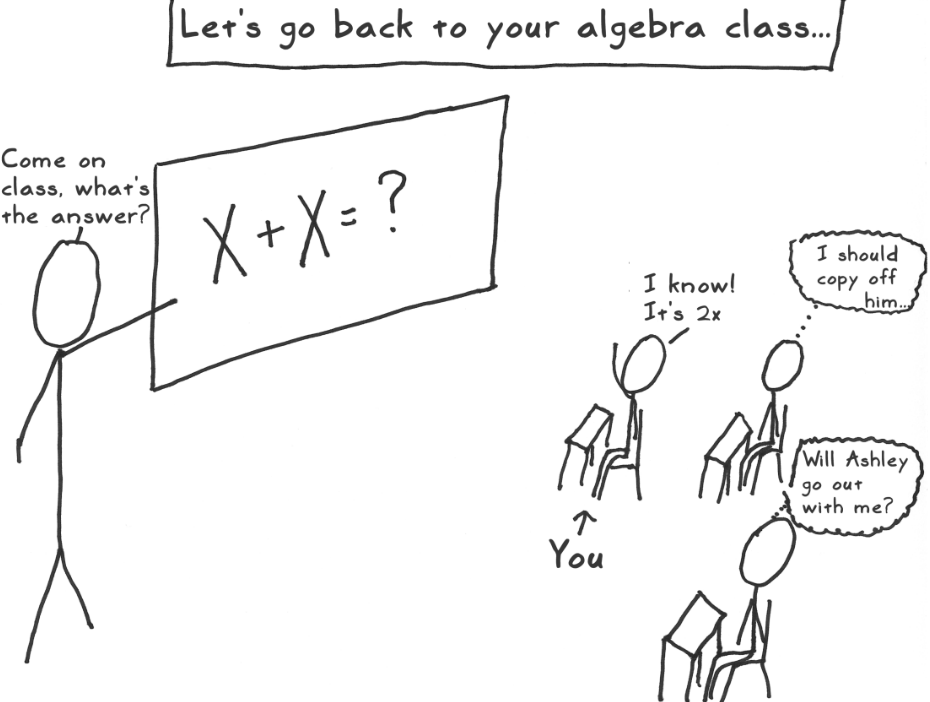



Make sense? Did that answer your question?
Almost...except you just waved your hands and used weird analogies. What really happens?

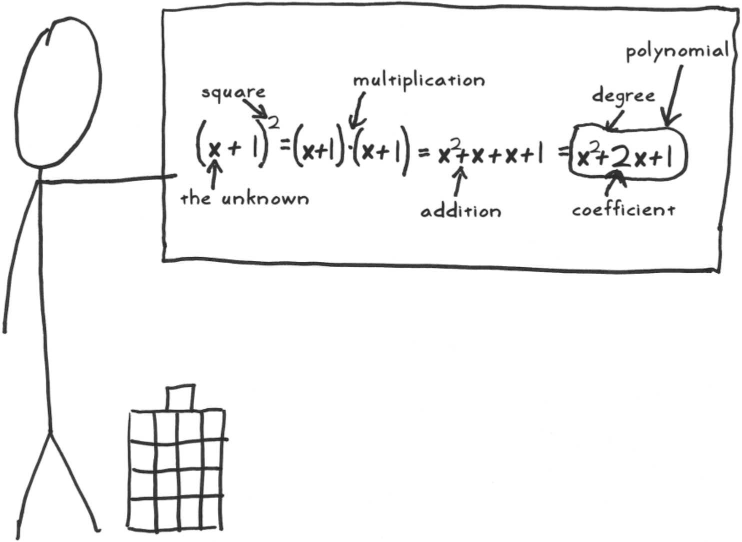


Another great question! It's not hard, but... it involves a little... math.
I'm game. Bring it on!!
Math is hard! Let's go shopping!
Exit

Act 4: Math!


Let's go back to your algebra class...
Come on class, what's the answer?
X+X=?
I know! It's 2x
I should copy off him...
Will Ashley go out with me?
You

Reviewing the Basics...



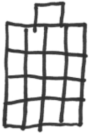
We'll change things slightly. In the old way, coefficients could get as big as we wanted. In the new way, they can only be 0 or 1:

Old Way

$$123x^2+45x^2+678x+9x+10$$

$$=168x^2+687x+10$$

Big coefficients



New Way

$$x^2 \oplus x^2 \oplus x^2 \oplus x \oplus x \oplus 1$$

$$= x^2 \oplus 1$$

The "new" add*

Small coefficients

$$x^2 \oplus x^2 \oplus x^2 = (x^2 \oplus x^2) \oplus x^2$$

$$= 0 \oplus x^2$$

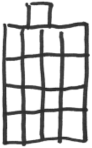
$$= x^2$$

*Nifty Fact: In the new way, addition is the same as subtraction (e.g. $x \oplus x = x - x = 0$)

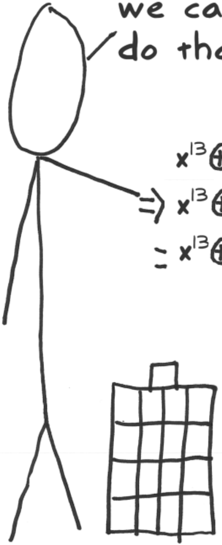
Remember how multiplication could make things grow fast?

$$
\begin{aligned}
&(x^7+x^5+x^3+x)\cdot(x^6+x^4+x^2+1)\\
&= x^{7+6}+x^{7+4}+x^{7+2}+x^{7+0}+x^{5+6}+x^{5+4}+x^{5+2}+x^{5+0}\\
&\quad +x^{3+6}+x^{3+4}+x^{3+2}+x^{3+0}+x^{1+6}+x^{1+4}+x^{1+2}+x^{1+0}\\
&= x^{13}+x^{11}+x^9+x^7+x^{11}+x^9+x^7+x^5+x^9+x^7+x^5+x^3+x^7+x^5+x^3+x\\
&= x^{13}+x^{11}+x^{11}+x^9+x^9+x^9+x^7+x^7+x^7+x^7+x^5+x^5+x^5+x^3+x^3+x\\
&= x^{13}+2x^{11}+3x^9+4x^7+3x^5+2x^3+x
\end{aligned}
$$

Big and yucky!

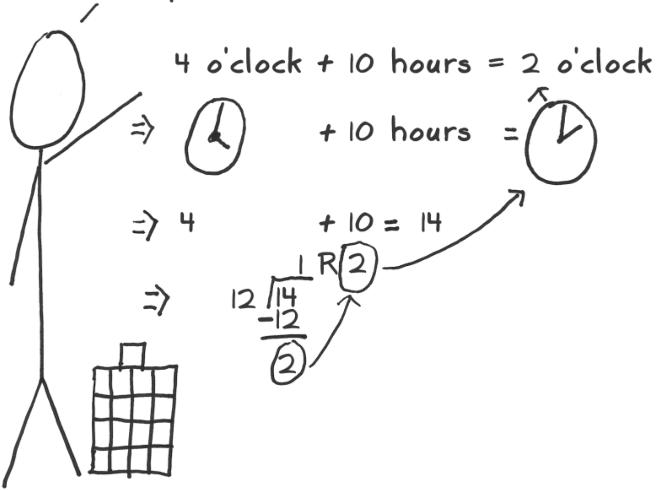


With the "new" addition, things are simpler, but the x^{13} is still too big. Let's make it so we can't go bigger than x^7. How can we do that?



$$
\begin{aligned}
& x^{13} \oplus 2x^{11} \oplus 3x^{9} \oplus 4x^{7} \oplus 3x^{5} \oplus 2x^{3} \oplus x \\
\Rightarrow\ & x^{13} \oplus 0x^{11} \oplus x^{9} \oplus 0x^{7} \oplus x^{5} \oplus 0x^{3} \oplus x \\
=\ & x^{13} \oplus x^{9} \oplus x^{5} \oplus x
\end{aligned}
$$

We use our friend, "clock math*," to do this. Just add things up and do long division. Keep a close watch on the remainder:



* This is also known as "modular addition." Math geeks call this a "group." AES uses a special group called a "finite field."

We can do 'clock' math with polynomials. Instead of dividing by 12 my creators told me to use $m(x) = x^8 \oplus x^4 \oplus x^3 \oplus x \oplus 1$. Let's say we wanted to multiply $x \cdot b(x)$ where $b(x)$ has coefficients $b_7 \ldots b_0$:

$$
\begin{aligned}
& x \cdot b(x) \\
&= x \cdot (b_7x^7 \oplus b_6x^6 \oplus b_5x^5 \oplus b_4x^4 \oplus b_3x^3 \oplus b_2x^2 \oplus b_1x \oplus b_0) \\
&= b_7x^8 \oplus b_6x^7 \oplus b_5x^6 \oplus b_4x^5 \oplus b_3x^4 \oplus b_2x^3 \oplus b_1x^2 \oplus b_0x)
\end{aligned}
$$

Eeek! x^8 is too big. We must make it smaller.

* Remember that each b_n (e.g. b_7) is either 0 or 1.

We divide it by $m(x) = x^8 \oplus x^4 \oplus x^3 \oplus x \oplus 1$ and take the remainder:

$$
\begin{array}{r|l}
 & b_7 \\
\hline
\underbrace{x^8 \oplus x^4 \oplus x^3 \oplus x \oplus 1}_{m(x)} & b_7x^8 \oplus b_6x^7 \oplus b_5x^6 \oplus b_4x^5 \oplus b_3x^4 \oplus b_2x^3 \oplus b_1x^2 \oplus b_0x \\
\oplus & b_7x^8 \qquad\qquad\qquad\qquad\qquad \oplus b_7x^4 \oplus b_7x^3 \oplus b_7x \oplus b_7 \\
\hline
 & b_6x^7 \oplus b_5x^6 \oplus b_4x^5 \oplus (b_3 \oplus b_7)x^4 \oplus (b_2 \oplus b_7)x^3 \\
 & \qquad\qquad \oplus b_1x^2 \oplus (b_0 \oplus b_7)x \oplus b_7 \\
\hline
\end{array}
$$

Remainder

$$b_6x^7 \oplus b_5x^6 \oplus b_4x^5 \oplus b_3x^4 \oplus b_2x^3 \oplus b_1x^2 \oplus b_0x$$

$$\oplus\, b_7 \cdot (x^4 \oplus x^3 \oplus x \oplus 1)$$

Note how the b's are shifted left by 1 spot.

This is just b_7 multiplied by a small polynomial.

Now we're ready for the hardest blast from the past: <u>logarithms</u>. After logarithms, everything else is cake! Logarithms let us turn multiplication into addition:

$$\log(x\cdot y)=\log(x)+\log(y)$$

So... $\log(10\cdot 100)=\log(10^1)+\log(10^2)$

$$=2+1=3$$

In reverse:

$$\log^{-1}(1)=10^1=10$$
$$\log^{-1}(2)=10^2=100$$
$$\log^{-1}(3)=10^3=1{,}000$$

$\Rightarrow 10\cdot 100=1{,}000$

We can use logarithms in our new world. Instead of using 10 as the base, we can use the simple polynomial of $x \oplus 1$ and watch the magic unravel.*

$$(x \oplus 1)^1 = x \oplus 1$$

$$(x \oplus 1)^2 = (x \oplus 1) \cdot (x \oplus 1) = x^2 \oplus x \oplus x \oplus 1 = x^2 \oplus 1$$

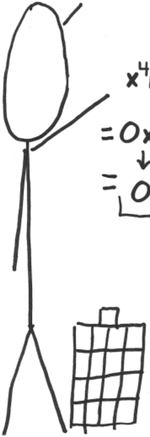
$$(x \oplus 1)^3 = (x \oplus 1)^2 \cdot (x \oplus 1) = x^3 \oplus x^2 \oplus x \oplus 1$$

So...

$$\log_{x \oplus 1}(x \oplus 1) = 1,\ \log_{x \oplus 1}(x^2 \oplus 1) = 2,\ \log_{x \oplus 1}(x^3 \oplus x^2 \oplus x \oplus 1) = 3$$

*If you keep multiplying by $(x \oplus 1)$ and then take the remainder after dividing by $m(x)$, you'll see that you generate all possible polynomial below x^8. This is very important!

Why bother with all of this math?* Encryption deals with bits and bytes, right? Well, there's one last connection: a 7th degree polynomial can be represented in exactly 1 byte since the new way uses only 0 or 1 for coefficients:



$$x^4 \oplus x^3 \oplus x \oplus 1$$

$$= 0x^7 \oplus 0x^6 \oplus 0x^5 \oplus 1x^4 \oplus 1x^3 \oplus 0x^2 \oplus 1x \oplus 1$$

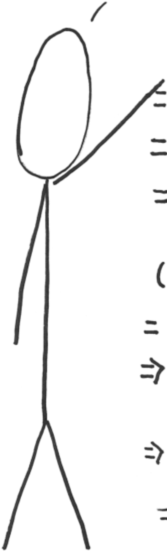
$$= \underbrace{0\ \ 0\ \ 0\ \ 1}_{1}\ \ \underbrace{1\ \ 0\ \ 1\ \ 1}_{1011_2 = 11_{10} = b_{16}}$$

← hexadecimal

$= 1b$ ← A single byte!!

*Although we'll work with bytes from now on, the math makes sure everything works out.

With bytes, polynomial addition becomes a simple xor. We can use our logarithm skills to make a table for speedy multiplication.*



$$(x^4 \oplus x^3 \oplus x \oplus 1) \oplus (x^7 \oplus x^5 \oplus x^3 \oplus x)$$

$$= \mathtt{1b} \oplus \mathtt{aa}$$ ← byte xor

$$= \mathtt{b1}$$

$$= x^7 \oplus x^5 \oplus x^4 \oplus 1$$

$$(x^4 \oplus x^3 \oplus x \oplus 1) \cdot (x^7 \oplus x^5 \oplus x^3 \oplus x)$$

$$= \mathtt{1b} \cdot \mathtt{aa}$$ logarithm table lookup

$$\Rightarrow \log(\mathtt{1b}) + \log(\mathtt{aa}) = \mathtt{c8} + \mathtt{1f} = \mathtt{e7}$$

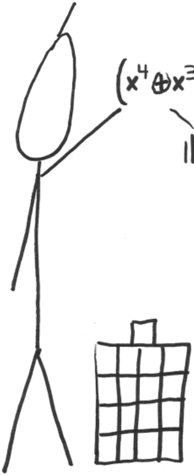
inverse table lookup

$$\Rightarrow \log^{-1}(\mathtt{e7}) = \mathtt{8c} \Rightarrow \mathtt{1b} \cdot \mathtt{aa}$$

$$= x^7 \oplus x^3 \oplus x^2$$

* We can create the table as we keep multiplying by $(x \oplus 1)$.

Since we know how to multiply, we can find the "inverse" polynomial byte for each byte. This is the byte that will undo/invert the polynomial back to 1. There are only 255* of them, so we can use brute force to find them:



$$(x^4 \oplus x^3 \oplus x \oplus 1) \cdot ? = 1$$

$$1b \cdot cc = 1$$

found using a brute force for-loop

* There are only 255 instead of 256 because 0 has no inverse.

Now we can understand the mysterious s-box. It takes a byte "a" and applies two functions. The first is "g" which just finds the byte inverse. The second is "f" which intentionally makes the math uglier to foil attackers.

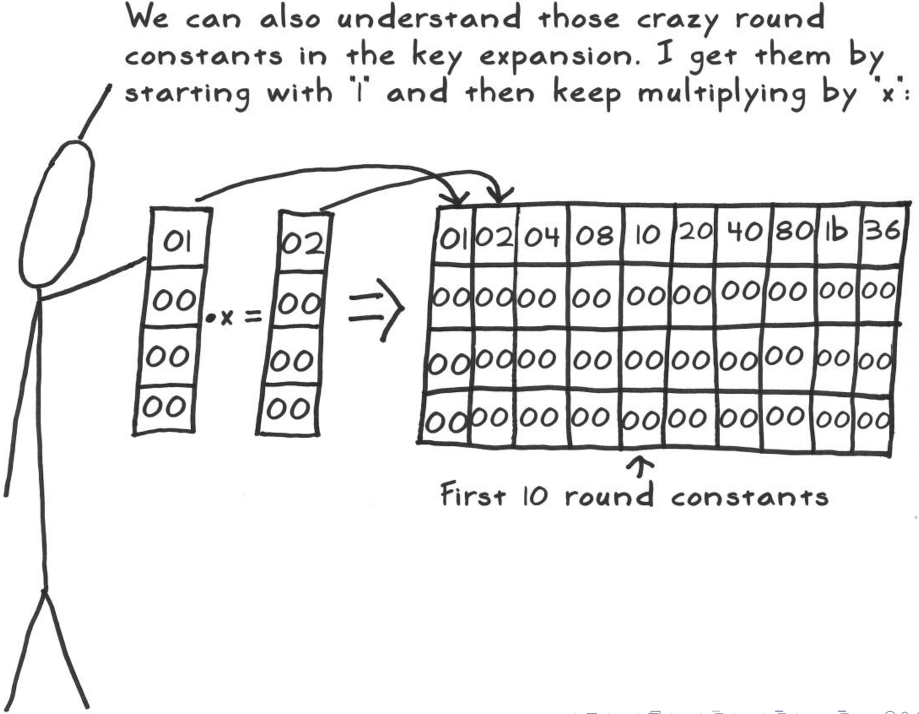
$$g(a) = a^{-1}$$

$$f(a) = \begin{bmatrix} 1&1&1&1&1&0&0&0 \\ 0&1&1&1&1&1&0&0 \\ 0&0&1&1&1&1&1&0 \\ 0&0&0&1&1&1&1&1 \\ 1&0&0&0&1&1&1&1 \\ 1&1&0&0&0&1&1&1 \\ 1&1&1&0&0&0&1&1 \\ 1&1&1&1&0&0&0&1 \end{bmatrix} \cdot \begin{bmatrix} a_7 \\ a_6 \\ a_5 \\ a_4 \\ a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

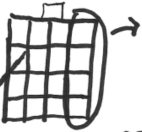
$$\text{sbox}[a] = f(g(a))$$

$$\text{sbox}[58] = f(g(58))$$

$$\text{sbox}[58] = f(\underbrace{18}_{58 \cdot 18 = 01}) = 6a$$


We can also understand those crazy round constants in the key expansion. I get them by starting with "1" and then keep multiplying by "x":
01
00
00
00
•x =
02
00
00
00
01 02 04 08 10 20 40 80 1b 36
00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00
First 10 round constants

Mix Columns is the hardest. I treat each column as a polynomial. I then use our new multiply method to multiply it by a specially crafted polynomial and then take the remainder after dividing by x^4+1. This all simplifies to a matrix multiply:



$$\begin{bmatrix}a_3\\a_2\\a_1\\a_0\end{bmatrix}$$

$$b(x) = c(x)\cdot a(x) \bmod x^4+1$$

$$= \underbrace{(03x^3+01x^2+01x+02)}_{\text{special polynomial}}\cdot\underbrace{(a_3x^3+a_2x^2+a_1x+a_0)}_{\text{the column}} \bmod x^4+1$$

$$=$$

Quotient: $03a_3\cdot x^2+(3a_2+a_3)x+(3a_1+a_2+a_3)$

Dividing by x^4+1:

$$03a_3x^6+03a_2x^5+03a_1x^4+03a_0x^3+01a_3x^5+01a_2x^4+01a_1x^3+01a_0x^2$$
$$+01a_3x^4+01a_2x^3+01a_1x^2+01a_0x+02a_3x^3+02a_2x^2+02a_1x+02a_0$$

$$\oplus\ 03a_3x^6+03a_3x^2$$

$$3a_2x^5+3a_1x^4+3a_0x^3+a_3x^5+a_2x^4+a_1x^3+a_0x^2+a_3x^4+a_2x^3+a_1x^2+a_0x+2a_3x^3$$
$$+2a_2x^2+2a_1x+2a_0+3a_3x^2$$

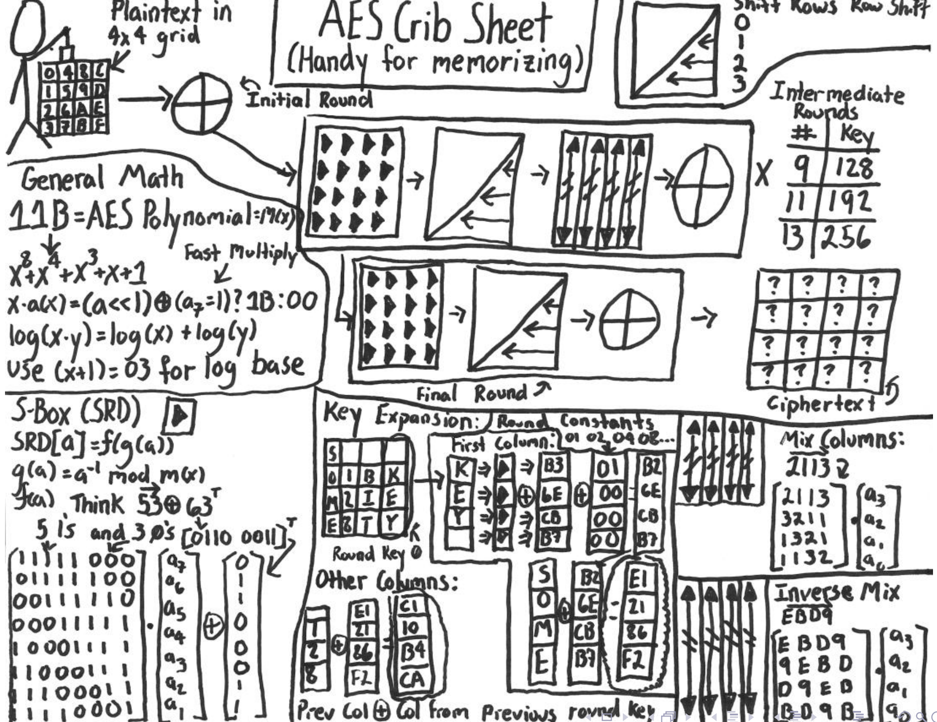
$$\oplus\ 3a_2x^5+a_3x^5+3a_2x+a_3x$$

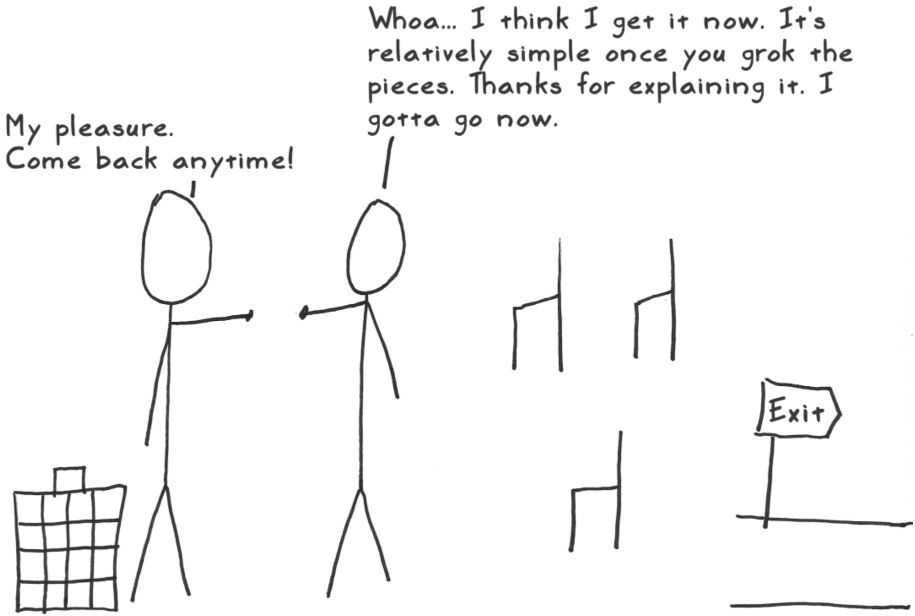
$$3a_1x^4+3a_0x^3+a_2x^4+a_1x^3+a_0x^2+a_3x^4+a_2x^3+a_1x^2+a_0x+2a_3x^3+2a_2x^2+2a_1x+2a_0$$
$$+3a_3x^2+3a_2x+a_3x$$

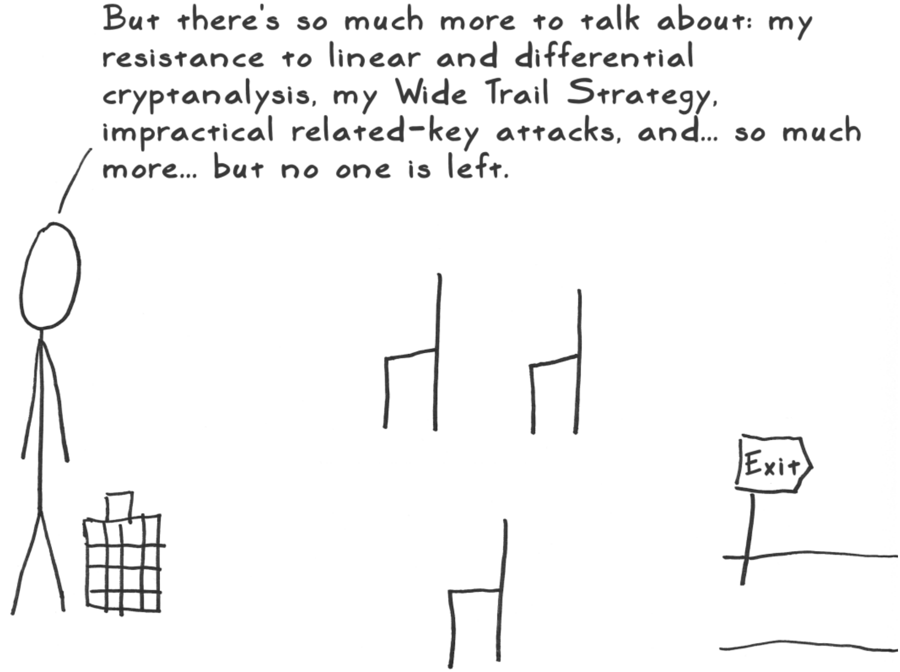
$$\oplus\ (3a_1+a_2+a_3)x^4+(3a_1+a_2+a_3)$$

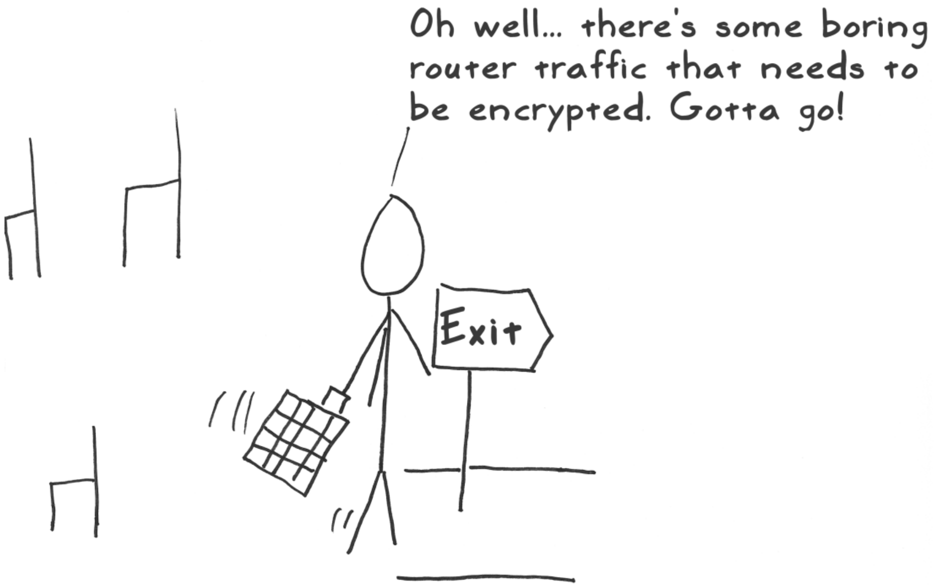
$$(2a_3+a_2+a_1+3a_0)x^3+(3a_3+2a_2+a_1+a_0)x^2$$
$$+(a_3+3a_2+2a_1+a_0)x+(a_3+a_2+3a_1+2a_0)$$

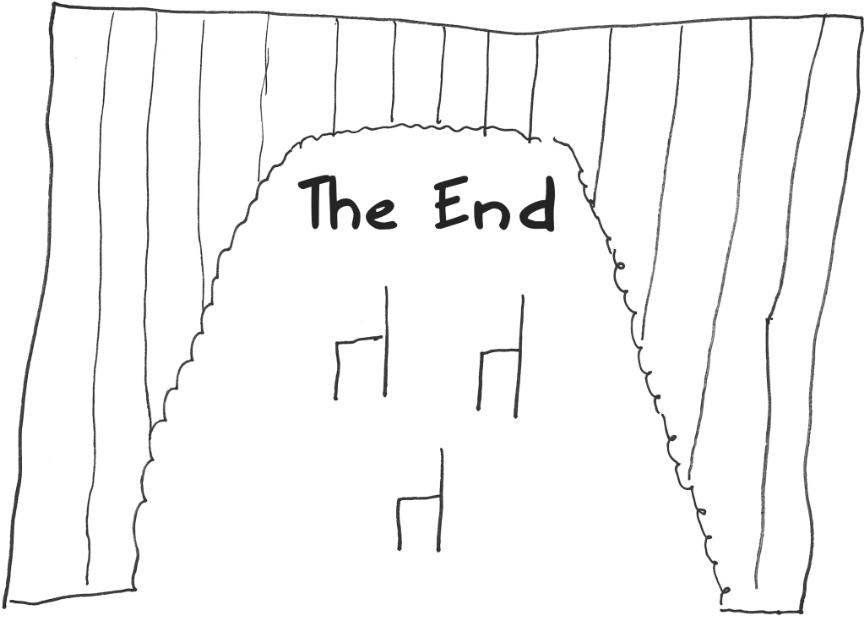
$$\Longrightarrow \begin{bmatrix}2&1&1&3\\3&2&1&1\\1&3&2&1\\1&1&3&2\end{bmatrix}\cdot\begin{bmatrix}a_3\\a_2\\a_1\\a_0\end{bmatrix}=\begin{bmatrix}b_3\\b_2\\b_1\\b_0\end{bmatrix}$$


AES Crib Sheet
(Handy for memorizing)
Plaintext in 4x4 grid
0 4 8 C
1 5 9 D
2 6 A E
3 7 B F
Initial Round
Shift Rows
Row Shift
0
1
2
3
Intermediate Rounds
Key
9 128
11 192
13 256
X
General Math
11B = AES Polynomial = m(x)
x8+x4+x3+x+1
Fast Multiply
x·a(x) = (a<<1) ⊕ (a7=1) ? 1B : 00
log(x·y) = log(x) + log(y)
Use (x+1) = 03 for log base
Final Round
? ? ? ?
? ? ? ?
? ? ? ?
? ? ? ?
Ciphertext
S-Box (SRD)
SRD[a] = f(g(a))
g(a) = a-1 mod m(x)
f(a) Think 53 ⊕ 63T
5 1's and 3 0's [0110 0011]T
11111000
01111100
00111110
00011111
10001111
11000111
11100011
11110001
a7 a6 a5 a4 a3 a2 a1
0 1 1 0 0 0 1 1
Key Expansion:
Round Constants 01 02 04 08...
First Column:
S
0 1 B K
M 2 I E
E 8 T Y
Round Key 0
K E Y
B3 6E CB B7
01 00 00 00
B2 6E CB B7
Other Columns:
T 2 8
E1 21 86 F2
C1 10 B4 CA
S O M E
B2 6E CB B7
E1 21 86 F2
Prev Col ⊕ Col from Previous round key
Mix Columns:
2113
2113
3211
1321
1132
a3 a2 a1 a0
Inverse Mix
EBD9
EBD9
9EBD
D9EB
BD9E
a3 a2 a1 a0


Whoa... I think I get it now. It's relatively simple once you grok the pieces. Thanks for explaining it. I gotta go now.
My pleasure. Come back anytime!
Exit


But there's so much more to talk about: my resistance to linear and differential cryptanalysis, my Wide Trail Strategy, impractical related-key attacks, and... so much more... but no one is left.
Exit


Oh well... there's some boring router traffic that needs to be encrypted. Gotta go!
Exit


The End