AES Key Schedule, Explanation of GF(2^8) Field

Here is pseudocode which shows how to produce the round keys (from the Stallings book Cryptography and Network Security)

```
KeyExpansion(byte key[16], word w[44]) {

  word temp;

  for (i=0; i<4; i++)
    w[i] = (key[4i], key[4i+1], key[4i+2], key[4i+3]);

  for (i=4; i<44; i++) {
    temp = w[i-1];
    if (i%4 == 0)
        temp = SubWord(RotWord(temp)) XOR Rcon[i/4];
    w[i] = w[i-4] XOR temp;
  }
}
```

Normally, we simply XOR two previous words (32 bits – the last four, and the fourth to last word) to get the new word. But, for each multiple of 4, we do a special operation on temp. Namely, we first perform a **cyclic left-shift of one byte** to it (this is the RotWord), then we perform a byte substitution on each byte in it based on the original S-box, also used in the beginning of the algorithm, and finally we XOR it with a value stored in the array Rcon. Here are the values:

j	1	2	3	4	5	6	7	8	9	10
RCon[j]	01	02	04	08	10	20	40	80	1B	36

This array starts with the value 01 in the first index, and all subsequent indexes store a value obtained by doubling the previous value in the field discussed earlier. In all cases except for going from index 8 to 9, this is just regular doubling. Here's how we calculate index 9:

80 x 02 = 10000000 x 00000010 = 00000000 XOR 00011011 = 00011011 = 1B.

This turns out NOT to be an exception because it's just multiplication by 2 in the AES field.

Let's look at a couple examples of the Key Expansion Algorithm:

Example 1 – calculating w[26]
Let's say that in HEX w[22] = `26 35 A4 B8` and that
w[25] = `A3 C7 5B B3`

We can use this information to calculate w[26] as follows: Notice that when we go through the pseudocode, the if statement doesn't trigger because 26 isn't divisible by 4. Thus all we do is XOR the two words. Using the Hex XOR chart expedites this process and we get:

```
w[22] = 26 35 A4 B8
w[25] = A3 C7 5B B3
-------------------
w[26] = 85 F2 FF 0B
```

Example 2 – calculating w[40]
Consider calculating w[40] given the following information:

w[36] = B1 89 C4 07 (in hex)
w[39] = 9C 2F 63 DE (in hex)

Notice that this time, 40 is divisible by 4, so there are several steps to perform before the final XOR. Here are the steps:

1. temp = RotWord(w[39])
2. temp = SubWord(temp)
3. temp = Rcon[40] XOR temp
4. w[40] = w[36] XOR temp

Here is how we can fill this information out in a table:

RotWord	SubWord	Rcon[i/4]	XOR	FinalResult
2F 63 DE 9C	15 FB 1D DE	36 00 00 00	23 FB 1D DE	92 72 D9 D9

First, we take w[39] and move its first byte (9C) to the end.
Next, we substitute for each byte from the S-box. (Note S-box(2F) = 15, etc.)
Next, we XOR Rcon[10] with the result from the subword – notice that only the first byte changes. This is always the case, because the Rcon array items always have 24 0 bits at their end.

Finally, XOR the result from the previous step with w[36], using the HEX xor chart:

```
B1 89 C4 07
23 FB 1D DE
-----------
92 72 D9 D9
```

Field GF(2^8) used for AES

A field is a special type of group. Group Theory is a branch of Number Theory (which of course is a branch of Mathematics.)

Group Definition

A group in mathematics is a set of elements (G) paired with an operation (°) for which the following properties hold:

A1. Closure – If a and b are elements of G, then a°b is as well.
A2. Associative – a ° (b ° c) = (a ° b) ° c is true for all a,b and c in G
A3. Identity Element – there is an element e in G such that a°e = e°a = a for all a in G.
A4. Inverse Element – for each a in G, there is an a' in G such that a°a' = e.

An example of a group would be addition mod n of the elements 0, 1, 2, …, n-1. When we mod (using the function) we always arrive at another element in the set. The order of parentheses in addition doesn't matter. The identity element is 0. Each element x (except 0) has the inverse element n – x, and 0's inverse is 0.

A group is said to be finite if it has a finite number of elements and infinite if it has an infinite number of elements.

Abelian Group

An Abelian Group is a group that also satisfies the following property:

A5. Commutative - a°b = b°a for all elements a and b in G.

The previous example is also an Abelian Group, since the order of addition doesn't matter.

Cyclic Group

For an element a in a group G, define a^k = a°a°a…°a, k times total.

A group is cyclic if and only if there exists some element a in G such that for every other element b in G, b = a^k for some integer k. The element a is said to be a generator for the group. Note that groups may have multiple generators.

For addition under mod n, any value that is relatively prime to n in G is a generator. For example, let n = 8 and a = 5. Here is a table with the values of a added k times, mod n:

k	0	1	2	3	4	5	6	7	8
a^k	0	5	2	7	4	1	6	3	0

We can see that the table eventually cycles, and each item in the set {0,1,2,3,4,5,6,7} can be obtained by "exponentiating" a some number of times.

Ring

A ring is a set of elements, but with two operations, addition (+) and multiplication (x). A ring satisfies the following properties:

A1 – A5: These properties with the addition operator
M1. Closure under multiplication: if a and b are in G, then a x b is in G also.
M2. Associativity under multiplication: a x (b x c) = (a x b) x c, for all a, b, and c in G.
M3. Distributive Law: a x (b + c) = a x b + a x c, for all a, b and c in G.

If a ring is also commutative under multiplication, we call it a Commutative Ring:

M4. a x b = b x a for all a, b in G.

An Integral Domain is a Commutative Ring which also satisfies the two following properties:

M5. There is an element 1 in G such that for all a in G, a x 1 = 1 x a = a.
M6. No zero divisors: If a and b are in G, and if a x b = 0, then either a = 0 or b = 0. (0 is the additive identity.)

Field

A field is an Integral Domain which satisfies one additional property:

M7. Multiplicative inverse: For each a in G, except 0, there exists an element a^{-1} such that a x a^{-1} = 1 (multiplicative identity)

Note that addition and multiplication mod p, for a prime number p forms a field with the elements in the set {0, 1, 2, …, p-1}.

Use of Polynomials for AES

A regular polynomial, f(x), of degree d is of the form:

$$f(x) = c_d x^d + c_{d-1} x^{d-1} + \cdots + c_1 x + c_0$$

where each c_i is a constant (coefficient).

When we're dealing with modular arithmetic, we will limit the output of the polynomials to valid remainders mod n. We can do this by reducing any coefficients out of range to the unique equivalent value in range.

For example, if n = 5 (the mod value), then:

$$(3x^2 + 4x + 2) + (4x^2 + x + 1) = 2x^2 + 3$$

When we add 3 + 4 = 7, we immediately reduce this to 2 mod 5, producing the first term. For the second term, since 4 + 1 = 5 and 5 is equivalent to 0 mod 5, the term isn't there. Finally, 2 and 1 get added normally since 3 is already in range.

In GF(2^8), we do the following:

1. Limit polynomial coefficients to be 0 or 1 (mod 2).
2. Limit the degree of the polynomial by "modding" it by a polynomial of degree 8.

Modding by a polynomial

Let's quickly define modding by a polynomial. Just like numbers, where we can define a unique remainder when dividing a by b:

$a = bq + r, 0 \leq r < b$

we can do the same for dividing polynomial a(x) by b(x):

$a(x) = b(x)q(x) + r(x)$, where degree of r is less than degree of b.

Here is a quick example of doing a mod for two polynomials $a(x) = x^4 + x^3 + 1$ and $b(x) = x^2 + x + 1$:

$$
\begin{array}{rl}
 & x^2 + 1 \\
 & \text{----------} \\
x^2 + x + 1 \mid & x^4 + x^3 + 1 \\
 & x^2 + x^3 + x^2 \\
 & \text{----------} \\
 & x^2 + 1 \\
 & x^2 + x + 1 \\
 & \text{----------} \\
 & x
\end{array}
$$

Note that in the field GF(2^2), the coefficient -1 doesn't exist as its equivalent to 1.

This means that when we calculate a(x) mod b(x) we get just x because we have:

$x^4+x^3+1 = (x^2 + x + 1)(x^2 + 1) + x$

Specifically, for AES the mod polynomial is $m(x) = x^8 + x^4 + x^3 + x + 1$

One key calculation that will be important is calculating x^8 mod m(x). For convenience, I've included the work here:

$$
\begin{array}{r|l}
 & 1 \\
 & \text{----------} \\
x^8 + x^4 + x^3 + x + 1 & x^8 \\
 & x^8 + x^4 + x^3 + x + 1 \\
 & \text{----------} \\
 & x^4 + x^3 + x + 1
\end{array}
$$

It follows that x^8 mod $m(x) = x^4 + x^3 + x + 1$ because

$x^8 = (x^8 + x^4 + x^3 + x + 1) \times 1 + (x^4 + x^3 + x + 1)$

in the AES field.

Multiplication in the AES field

Thus, we can finally define multiplication in the AES field: Given two polynomials a(x) and b(x) in $GF(2^8)$, their product will be a(x) x b(x) mod m(x). Note that when we multiply, we immediately reduce all coefficients mod 2.

Then, if the result is a polynomial of degree 8 or greater, we must reduce the result mod m(x) via long division. (Though, in code there's a much easier way to do it.)

Once we note that $x^8 = x^4 + x^3 + x + 1 \pmod{m(x)}$

Then we can figure out that

$x^9 = x(x^8) = x(x^4 + x^3 + x + 1) = x^5 + x^4 + x^2 + x.$

We can similarly figure out other powers of x. If one of these calculations produces a term of the form x^8, we just substitute that with $x^4 + x^3 + x + 1$.

Here are a couple examples:

$$
\begin{aligned}
(x^4 + x^3 + x)(x^3 + x^2 + 1) &= x^7 + x^6 + x^5 + x^6 + x^5 + x^4 + x^4 + x^3 + x^2 \\
&= x^7 + 2x^6 + 2x^5 + 2x^4 + x^3 + x^2 \\
&= x^7 + x^3 + x^2
\end{aligned}
$$

$$
\begin{aligned}
(x^6 + x)(x^4+1) &= x^{10} + x^6 + x^5 + x \\
&= x^2(x^8) + x^6 + x^5 + x \\
&= x^2(x^4 + x^3 + x + 1) + x^6 + x^5 + x \\
&= x^6 + x^5 + x^3 + x^2 + x^6 + x^5 + x \\
&= x^3 + x^2 + x
\end{aligned}
$$

AES Mix Columns, S-Box

The manner in which mix columns works is that when we specify a multiplication, such as 03 x D8, we are really multiplying two polynomials in the field GF(2^8) with the polynomial $m(x) = x^8 + x^4 + x^3 + x + 1$. One other note: this polynomial is irreducible. This means that there are no polynomials a(x) and b(x), each of degree 1 or greater, that multiply to m(x) using mod 2 for the coefficients.

An example of a reducible polynomial from a previous example is
$x^7 + x^3 + x^2 = (x^4 + x^3 + x)(x^3 + x^2 + 1) = x^2(x^5 + x + 1)$

Note that for this particular polynomial, there are multiple ways to express it as the product of two polynomials of degree 1 or greater. I've included a second, more obvious example at the right side of the equation.

When working the mix columns step forward, we never have to multiply by any polynomial greater than x+1. This means that no term gets created greater than x^8. Thus, the "quick fix" we discussed earlier (replacing x^8 with x^4+x^3+x+1, or 00011011) suffices to be able to make all necessary calculations.

The corresponding decryption matrix however, has terms such as 0E, 0B, 0D and 09. Thus, doing these by hand would necessitate a slightly better understanding of multiplication in the field. But, the iterative trick shown previously (writing x^{10} as x^2x^8, and then doing our substitution for x^8) will suffice eventually.

The S-Box is constructed as follows, as a matrix multiplication (looking at values mod 2) followed by an addition:

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

Let the input to the X-Box be $x_7x_6\ldots x_0$, in bits from most to least significant. First, we find the inverse of be $x_7x_6\ldots x_0$ in the field GF(2^8) mod m(x). Assign this inverse the value $a_7a_6\ldots a_0$. Then, multiply this stored in the column (in reverse as shown above) by the matrix shown, then add the corresponding column matrix shown. This computation is how the S-Box for AES is constructed.

As a quick example, consider the entry for {01}. The inverse of 1 in the field is 1, so we can store $a_0 = 1$ $a_1 = 0$, $a_2 = 0$, … $a_7 = 0$. When we do the matrix multiply, we get 1's for the first 5 entries and 3 zeros. When we add to 11000110, we get 00111110, which, when read in reverse is 0111 1100, or 7C, which is the entry in row 0, column 1, of the S-box.