

Galois field

Galois field

For cryptography we process bytes (integers in the interval $[0, 255]$). The appropriate Galois field is called $GF(2^8)$. We will use cofactors for the polynomials drawn from the set $\{0, 1\}$ as before, but we go up to degree 8.

We closed the last chapter by showing polynomials defined over $GF(2^3)$ modulo the irreducible polynomial $x^3 + x + 1$, which consist of the finite set:

$$0$$

$$1$$

$$x$$

$$x + 1$$

$$x^2$$

$$x^2 + 1$$

$$x^2 + x$$

$$x^2 + x + 1$$

Kak says:

Our conceptualization of $GF(2^3)$ is analogous to our conceptualization of the set Z_8 . The eight elements of Z_8 are to be thought of as integers modulo 8. So, basically, Z_8 maps all integers to the eight numbers in the set Z_8 . Similarly, $GF(2^3)$ maps all of the polynomials over $GF(2)$ to the eight polynomials shown above.

He continues:

$GF(2^3)$ contains a unique multiplicative inverse for every non-zero element for the same reason that Z_7 contains a unique multiplicative inverse for every non-zero integer in the set. (For a counterexample, recall that Z_8 does not possess multiplicative inverses for 2, 4, and 6.) Stated formally, we say that for every non-zero element $a \in GF(2^3)$ there is always a unique element $b \in GF(2^3)$ such that $a \times b = 1$.

The above conclusion follows from the fact if you multiply a non-zero element a with each of the eight elements of $GF(2^3)$, the result will be the eight distinct elements of $GF(2^3)$. Obviously, the results of such multiplications must equal 1 for exactly one of the non-zero elements of $GF(2^3)$. So if $a \times b = 1$, then b must be the multiplicative inverse for a .

The same thing happens in Z_7 . If you multiply a non-zero element a of this set with each of the seven elements of Z_7 , you will get seven distinct answers. The answer must therefore equal 1 for at least one such multiplication. When the answer is 1, you have your multiplicative inverse for a .

.

In fact

For a more formal proof (by contradiction) of the fact that

if you multiply a non-zero element a of $GF(2^3)$ with every element of the same set, no two answers will be the same, let's assume that this assertion is false. That is, we assume the existence of two distinct values b and c in the set such that

$$a \times b \equiv a \times c \pmod{x^3 + x + 1}$$

which implies

$$a \times (b - c) \equiv 0 \pmod{x^3 + x + 1}$$

which implies either $a = 0$ or $b = c$, in either case this is a contradiction.

In exploring multiplication in $GF(2^8)$ constructed with the irreducible polynomial

$$x^8 + x^4 + x^3 + x + 1$$

my approach was to show by exhaustive search that every number has a unique multiplicative inverse, and furthermore for every product other than 1 and every factor a there is a unique b such that $a \times b = p$.

addition

The fundamental definition of addition for our Galois field $GF(2^8)$ is that it is the same as the XOR operation:

$$a \oplus b$$

Since $a \oplus a = 0$, it follows that each number is its own additive inverse, so that $a + a = 0$. Addition is the same as subtraction.

And from this a clear implication is that multiplication is *not* repeated addition.

The irreducible polynomial that is used for $GF(2^8)$ is

$$x^8 + x^4 + x^3 + x + 1$$

It is claimed that the finite field $GF(2^8)$ contains 256 distinct polynomials over $GF(2)$.

representations

The common way to write numbers in $GF(2^8)$ is like this

$$= x^7 + x^6 + x^5 + x^2 + x^1 + x^0$$

the exponents on the last two terms are typically suppressed:

$$= x^7 + x^6 + x^5 + x^2 + x + 1$$

If a term is present, then a 1 is present in the place corresponding to the exponent, counting from right to left and starting with index 0. So

$$x^7 + x^6 + x^5 + x^2 + x + 1$$

is the same as binary 1110 0111 or **0xe7**. In fact, exactly the integers 0-255 or hex **00** to **ff** are in this field.

multiplication

Multiplication by 1 is the easiest: $b * 1 = b$, which is a great relief.

Multiplication by any number ≥ 2 consists of two steps: the multiplication itself, possibly followed by a mod operation.

multiplication by 2

For multiplication by 2 there are two cases: if the most significant bit (MSB) of b is not set ($b \leq 127$) then we simply do a left-shift: throw away the left-hand 0 bit from b and insert a 0 bit on the right

$$0111 \ 1111 * 2 = 1111 \ 1110 = b'$$

If b *does* have its most-significant bit set ($b > 127$), then after the shift, we do the mod operation, which can be achieved by:

$$b' \oplus 27$$

We show later why this works later. For now, we move on to higher powers.

multiplication by powers of 2

To multiply by 4 simply carry out two successive multiplications by 2.

$$b * 4 = (b * 2) * 2$$

For higher powers, lather, rinse and repeat:

$$b * 8 = (b * 4) * 2 = ((b * 2) * 2) * 2$$

multiplication by any number in GF(2⁸)

Suppose we want to multiply $1000 \ 0011 * b$. First use the distributive law to write

$$\begin{aligned} & 1000 \ 0011 * b \\ &= [1000 \ 0000 + 0000 \ 0010 + 0000 \ 0001] * b \end{aligned}$$

$$= 1000\ 0000 * b + 0000\ 0010 * b + 0000\ 0001 * b$$

Break the multiplier into its constituent powers of two, multiply as before, and then carry out a final addition step (which is just \oplus).

And that is really all we need.

When we understand the theory better, we hope to be able to explain why multiplying a 4-byte sequence like **db 13 53 45** by the matrix below to yield **8e 4d a1 bc**

$$fM = \begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} * \begin{bmatrix} db \\ 13 \\ 53 \\ 45 \end{bmatrix} = \begin{bmatrix} 8e \\ 4d \\ a1 \\ bc \end{bmatrix}$$

is reversed by multiplying the latter by

$$rM = \begin{bmatrix} 14 & 11 & 13 & 9 \\ 9 & 14 & 11 & 13 \\ 13 & 9 & 14 & 11 \\ 11 & 13 & 9 & 14 \end{bmatrix}$$

The reason is that

$$fM \times fM = rM$$

and

$$fM \times fM \times fM = fM \times rM = I$$

And I wrote a matrix multiplication routine to prove it.

More about Galois Field math

I found some other write-ups that provide a different perspective and a little more insight.

<http://www.cs.utsa.edu/~wagner/laws/FFM.html>)

Consider this multiplication: **0xb6 * 0x53**. Write it in binary as:

$$1011\ 0110\ * \ 0101\ 0011$$

We adopt a new notation. Rewrite this as

$$(7\ 5\ 4\ 2\ 1) * (6\ 4\ 1\ 0)$$

This is a shorthand version of the Galois field notation:

$$(x^7 + x^5 + x^4 + x^2 + x) * (x^6 + x^4 + x + 1)$$

The first line below restates the problem, then we do the multiplication one digit at a time. It turns out that we can multiply by each of the digits, XOR the result, and not worry about the mod until the end.

$$(7\ 5\ 4\ 2\ 1) * (6\ 4\ 1\ 0)$$

$$\begin{array}{rcll} (7\ 5\ 4\ 2\ 1) * 6 = & 13 & 11\ 10 & 8\ 7 \\ (7\ 5\ 4\ 2\ 1) * 4 = & & 11 & 9\ 8\ 6\ 5 \\ (7\ 5\ 4\ 2\ 1) * 1 = & & & 8\ 6\ 5\ 3\ 2 \\ (7\ 5\ 4\ 2\ 1) * 0 = & & & 7\ 5\ 4\ 2\ 1 \\ \text{XOR} & 13 & 10\ 9\ 8 & 5\ 4\ 3\ 1 \end{array}$$

Recall above where we said that multiplying by decimal 2, which is the same as multiplying by x in the field notation, is also the same as a left-shift of 1 in the binary representation. In the line $(7\ 5\ 4\ 2\ 1) * 6$, we simply add 6 to each of the digits in the first number to give the result (13,11,10,8,7). The spacing is used to simplify the XOR on the last line.

Now, we do the mod operation.

		13	10 9 8	5 4 3 1
(8 4 3 1 0) * 5		13	9 8	6 5
XOR			10	6 4 3 1
(8 4 3 1 0) * 2			10	6 5 3 2
XOR				5 4 2 1

The special divisor is (8, 4, 3, 1, 0).

$$(x^8 + x^4 + x^3 + x + 1)$$

We multiply by 5 to get it to align with the 13 in the interim result. This is the same as multiplying the irreducible polynomial by x^5 or multiplying its binary equivalent (1 0001 1011) by 2^5 .

After the XOR, the 13 is gone. We repeat the process, lining up with the 10 and zeroing it out. If there had been an 8 we would go that far, but no farther.

Our answer, then, is 0011 0110 which is **0x36**.

The complete multiplication is: **0xb6 * 0x53 = 0x36**.

Effectively what we've done is to do repeated divisions by powers of two of the super duper special number (1 0001 1011). We guarantee that every place higher than 8 will be turned to zero by this operation.

wikipedia example

Another view of the modulus or division operation comes from the wikipedia example: **0x53 * 0xca**. We write this in binary: 0101 0011 * 1100 1010.

In our new notation, this would be


```

(6 4 1 0) * (7 6 3 1)

{6 4 1 0} * 7 =   13   11   8 7
{6 4 1 0} * 6 =      12   10   7 6
{6 4 1 0} * 3 =          9   7   4 3
{6 4 1 0} * 1 =            7   5   2 1

XOR                      13 12 11 10 9 8   6 5 4 3 2 1

```

which looks like it's pretty special. And it is.

Here is the mod step:

```

(8 4 3 1 0) * 5 =   13 12 11 10 9 8   6 5 4 3 2 1
                    13           9 8   6 5

XOR
(8 4 3 1 0) * 4 =   12 11 10           4 3 2 1
                    12           8 7   5 4

XOR
(8 4 3 1 0) * 3 =   11 10   8 7   5   3 2 1
                    11           7 6   4 3

XOR
(8 4 3 1 0) * 2 =   10   8   6 5 4   2 1
                    10           6 5   3 2

XOR
(8 4 3 1 0) * 0 =           8           4 3   1
                    8           4 3   1 0

XOR                                                    0

```

That 0 at the end is not really zero. It is a power of 2, or $2^0 = \mathbf{0x01}$. What we've shown is that $\mathbf{0x53} * \mathbf{0xca} = \mathbf{0x01}$. In other words, $\mathbf{0x53}$ is the *multiplicative inverse* of $\mathbf{0xca}$.

Let's take a look at doing the same mod operation as long division, which should hammer home the point we've been absorbing.

$$\begin{array}{r}
0011 \ 1111 \ 0111 \ 1110 \\
^10 \ 0011 \ 011 \\
\hline
01 \ 1100 \ 0001 \ 1110 \\
^1 \ 0001 \ 1011 \\
\hline
0 \ 1101 \ 1010 \ 1110 \\
^1000 \ 1101 \ 1 \\
\hline
0101 \ 0111 \ 0110 \\
^100 \ 0110 \ 11 \\
\hline
1 \ 0001 \ 1010 \\
1 \ 0001 \ 1011 \\
\hline
1
\end{array}$$

Question: why is 1 0011 1011 special?

This one is hard. Fields constructed using irreducible polynomials as the modulus have really great properties. Showing why this is true is beyond me.

Choose any number in the range 0-255 except 0, and multiply that number by any number in the range 0-255. The resulting set is exactly 0-255. In particular, every number has a unique multiplicative inverse in this field.

Question: why does the mod 27 trick work?

I thought this one was hard, but it turns out to be easy. We use it when multiplying by 2, but only in the case where the original number is > 127 .

$$b_7.b_6.b_5.b_4 \ b_3.b_2.b_1.b_0 \times 2 = b_7 \ b_6.b_5.b_4.b_3 \ b_2.b_1.b_0.0$$

If the number is < 128 , then $b_7 = 0$ and the result of the multiplication is smaller than the irreducible polynomial $m(x)$ so nothing more needs

to be done. On the other hand, if $n > 127$, after the left-shift, we get a 1 in the 8th position, and we need to do the mod operation to clear that bit. Since there aren't any higher bits to the left of the 8th position to clear, the mod operation is a simple subtraction, which is just an \oplus . We clear the 1 at b_7 and get the rest of our result by subtracting 0011 1011 (that is, XOR or \oplus 27).