Example 126. (bonus challenge!) Find the smallest (pseudo)prime with 100 decimal digits, all of which are 3 or 7.

(Send me an email by next week with the prime, and how you found it, to collect a bonus point. Earn an extra bonus point if you can find it using a single line of Sage code [artificial concatenations not allowed].)

AES

Finite fields

Example 127. We have already seen xor in several cryptosystems. Note that a single xor operation as in the one-time pad or stream ciphers provides no diffusion.

When designing a cipher it may be nice to replace xor of N bit blocks with an operation that does provide some diffusion.

- A tiny amount of diffusion is provided by instead using addition modulo 2^N.
  Due to carries, one bit flip in the input can propagate to more than one bit flipped in the output.
- More diffusion can be achieved using operations (multiplication/inversion) in finite fields like $\mathrm{GF}(2^N)$.
  [We only need to make sure in our design that we don't multiply with zero.]

A **field** is a set of elements which can be added/subtracted as well as multiplied/divided by according to the usual rules.

In particular, a field always has distinguished elements 0 and 1, which are the neutral elements with respect to addition and multiplication, respectively.

Example 128.

- The rational numbers $\mathbb{Q}$, the real numbers $\mathbb{R}$, and the complex numbers $\mathbb{C}$ all are fields, which you have seen before. They contain infinitely many elements.
- The integers $\mathbb{Z}$ are not a field because, for instance, 3 is not invertible (since $\frac{1}{3}$ is not an integer itself). Quotients of integers (rational numbers!) are a field.
  Since addition/subtraction and multiplication work as they should, $\mathbb{Z}$ is what is called a **ring**.
- Polynomials are not a field (they are a ring like $\mathbb{Z}$). Quotients of polynomials (rational functions!) are a field.

Cryptographic applications require finite structures. Correspondingly, our focus will be on **finite fields**, that is, fields consisting of only a finite number of elements.

Example 129. Let p be a prime. The residues modulo p form a field, often denoted as $\mathrm{GF}(p)$.

GF is short for **Galois field**, which is another word for finite field.

Note that we can divide by any element! (Except the zero residue but, of course, we can never divide by 0).

Example 130. The residues modulo 21 (or any other composite number) are not a field.

We can add/subtract and multiply these numbers, but we cannot always divide. Specifically, we cannot divide by elements like $3, 6, 7, \ldots$ even though these are nonzero (we can, of course, never divide by zero).

Note. We have already seen that this seemingly slight deficiency has "terrible" consequences. For instance, the quadratic equation $x^2 = 1$ has more than the two solutions $x = \pm 1$ modulo 21 (namely, ± 8 as well).

AES is built upon byte operations (in contrast to DES, which is built on bit operations). Each of the 2^8 bytes represents one of the 2^8 elements of the finite field $\mathrm{GF}(2^8)$.

Note. We do not yet know what $\mathrm{GF}(2^8)$ is. It cannot be the residues modulo 2^8, because we just observed that the residues modulo n are a field only if n is prime.

To construct the finite field $\mathrm{GF}(p^n)$ of p^n elements, we can do the following:

- Fix a polynomial $m(x)$ of degree n, which is irreducible modulo p (i.e. cannot be factored modulo p).
- The elements of $\mathrm{GF}(p^n)$ are polynomials modulo $m(x)$ modulo p.

We will discuss the irreducibility condition on $m(x)$ next time. For now, see Example 133.

Comment. Actually, all finite fields can be constructed in this fashion. Moreover, choosing different $m(x)$ to construct $\mathrm{GF}(p^n)$ does not really matter: the resulting fields are always isomorphic (i.e. work in the same way, although the elements are represented differently). That justifies writing down $\mathrm{GF}(p^n)$, since there is exactly one such field.

Example 131. AES is based on representing bytes as elements of the field $\mathrm{GF}(2^8)$. It is constructed using the polynomial $x^8+x^4+x^3+x+1$ (which is indeed irreducible mod 2).

From bits to polynomials. For instance, the polynomial x^7+x^4+x corresponds to the bits 10010010 while x^6+1 corresponds to 01000001.

Example 132. The polynomial x^2+x+1 is irreducible modulo 2, so we can use it to construct the finite field $\mathrm{GF}(2^2)$ with 4 elements.

(a) List all 4 elements, and make an addition table. Then realize that this is just xor.

(b) Make a multiplication table.

(c) What is the inverse of $x+1$?

Solution.

(a) The four elements are $0, 1, x, x+1$.

For instance, $(x+1)+x=2x+1=1$ (in $\mathrm{GF}(2^2)$, since we are working modulo 2). The full table is below.

Each of the four elements is of the form $ax+b$, which can be represented using the two bits ab (for instance, $(10)_2$ represents x and $(11)_2$ represents $x+1$).

Then, addition of elements $ax+b$ in $\mathrm{GF}(2^2)$ works in the same way as xoring bits ab.

(b) For instance, $(x+1)^2=x^2+2x+1\equiv x^2+1\equiv(x+1)+1\equiv x$.

Here, the key is to realize that reducing modulo x^2+x+1 is the same as saying that $x^2=-x-1$, i.e. $x^2=x+1$ in $\mathrm{GF}(2^2)$. That means all polynomials of degree 2 and higher can be reduced to polynomials of degree less than 2.

$+$	0	1	x	$x+1$
0	0	1	x	$x+1$
1	1	0	$x+1$	x
x	x	$x+1$	0	1
$x+1$	$x+1$	x	1	0

$\times$	0	1	x	$x+1$
0	0	0	0	0
1	0	1	x	$x+1$
x	0	x	$x+1$	1
$x+1$	0	$x+1$	1	x

(c) We are looking for an element y such that $y(x+1)=1$ in $\mathrm{GF}(2^2)$. Looking at the table, we see that $y=x$ has that property. Hence, $(x+1)^{-1}=x$ in $\mathrm{GF}(2^2)$.

Example 133. What if we proceed as in the previous example but used $m(x)=x^2+1$ instead?

Solution. The addition table would be the same. The multiplication table would be different and a crucial difference would be that $(x+1)\cdot(x+1)=x^2+2x+1\equiv x^2+1\equiv 0$, which implies that $x+1$ cannot be invertible. That means our construction is not a field.

Comment. Note how, here, $m(x)$ factors modulo 2 as $x^2+1\equiv(x+1)(x+1)$. Hence the condition of irreducibility in the construction of $\mathrm{GF}(p^n)$ is violated.

Review. $\mathrm{GF}(p^n)$ is "the" finite field with p^n elements.

Recall that, in the construction of $\mathrm{GF}(p^n)$, the polynomial $m(x)$ has to be such that it cannot be factored modulo p. We also say that $m(x)$ needs to be **irreducible** mod p.

For instance. The polynomial x^2+2x+1 can always be factored as $(x+1)^2$.

On the other hand. For the polynomials $m(x)=x^2+x+1$ things are more interesting:

- x^2+x+1 cannot be factored over $\mathbb{Q}$ because the roots $\frac{-1\pm\sqrt{-3}}{2}$ are not rational.
- However, $x^2+x+1\equiv(x+2)^2$ modulo 3, so it can be factored modulo 3.
- On the other hand, x^2+x+1 is irreducible modulo 2 (that is, it cannot be factored: the only linear factors are x and $x+1$, but x^2, $x(x+1)$ and $(x+1)^2$ are all different from x^2+x+1 modulo 2).

In general, it follows from the formula $\frac{-1\pm\sqrt{-3}}{2}$ for the roots that x^2+x+1 can be factored modulo a prime $p>2$ if and only if $\sqrt{-3}$ exists as a residue modulo p. In other words, if and only if -3 is a quadratic residue modulo p.

For instance. Modulo $p=7$, we have $-3\equiv 2^2$ and $\frac{1}{2}\equiv 4$, so that $\frac{-1\pm\sqrt{-3}}{2}\equiv 4\cdot(-1\pm 2)\equiv 2,4$. Indeed, we have the factorization $(x-2)(x-4)=x^2-6x+8\equiv x^2+x+1$ modulo 7.

Example 134. The polynomial x^3+x+1 is irreducible modulo 2, so we can use it to construct the finite field $\mathrm{GF}(2^3)$ with 8 elements.

(a) List all 8 elements.

(b) Reduce x^5+1 in $\mathrm{GF}(2^3)$.

(c) Multiply each element of $\mathrm{GF}(2^3)$ with x^2+x.

(d) What is the inverse of x^2+x in $\mathrm{GF}(2^3)$?

Solution.

(a) The elements are $0,1,x,x+1,x^2,x^2+1,x^2+x,x^2+x+1$.
[Note that $x^3=-x-1=x+1$ in $\mathrm{GF}(2^3)$. That means all polynomials of degree 3 and higher can be reduced to polynomials of degree less than 3. See next part.]

(b) We divide x^5+1 by x^3+x+1 (long division!) to find $x^5+1=(x^2-1)(x^3+x+1)+(-x^2+x+2)$.
It follows that x^5+1 reduces to $-x^2+x+2\equiv x^2+x$ in $\mathrm{GF}(2^3)$.
Important. We can simplify things by performing the long division modulo 2. We then find $x^5+1\equiv(x^2+1)(x^3+x+1)+(x^2+x)$.

(c) We multiply the polynomials as usual, then reduce as in the previous part.
For instance, $(x^2+x)(x^2+x+1)\equiv x^4+x$ and, by long division, $x^4+x\equiv x(x^3+x+1)+x^2$, which reduces to just x^2 in $\mathrm{GF}(2^3)$.

$\times$	0	1	x	$x+1$	x^2	x^2+1	x^2+x	x^2+x+1
x^2+x	0	x^2+x	x^2+x+1	1	x^2+1	$x+1$	x	x^2

(d) We are looking for an element y such that $y(x^2+x)=1$ in $\mathrm{GF}(2^3)$. Looking at the table, we see that $y=x+1$ has that property. Hence, $(x^2+x)^{-1}=x+1$ in $\mathrm{GF}(2^3)$.
Important. To find the inverse, we essentially tried all possibilities. That's not sustainable. Instead, we can (and should!) proceed as we did for computing the inverse of residues modulo n. That is, we should use the Euclidean algorithm as indicated in the next examples. Here, this is just one step: modulo 2, we have $\boxed{x^3+x+1}\equiv(x+1)\cdot\boxed{x^2+x}+1$, so that $(x^2+x)^{-1}=x+1$ in $\mathrm{GF}(2^3)$.

Armin Straub
straub@southalabama.edu

Example 135.

(a) Apply the extended Euclidean algorithm to find the gcd of x^2+1 and x^4+x+1, and spell out Bezout's identity.

(b) Repeat the previous computation but always reduce all coefficients modulo 2.

(c) What is the inverse of x^2+1 in $\mathrm{GF}(2^4)$? Here, $\mathrm{GF}(2^4)$ is constructed using x^4+x+1.

Solution.

(a) We use the extended Euclidean algorithm:

$$\begin{aligned}&\gcd(x^2+1, x^4+x+1) && \boxed{x^4+x+1} = (x^2-1)\cdot\boxed{x^2+1} + (x+2)\\ &= \gcd(x+2, x^2+1) && \boxed{x^2+1} = (x-2)\cdot\boxed{x+2} + 5\\ &= 5\end{aligned}$$

Backtracking through this, we find that Bézout's identity takes the form

$$\begin{aligned}5 \;=\; 1\cdot\boxed{x^2+1} - (x-2)\cdot\boxed{x+2} \;&=\; 1\cdot\boxed{x^2+1} - (x-2)\cdot\left(\boxed{x^4+x+1} - (x^2-1)\cdot\boxed{x^2+1}\right)\\ &=\; (x^3-2x^2-x+3)\cdot\boxed{x^2+1} - (x-2)\cdot\boxed{x^4+x+1}\end{aligned}$$

If we wanted to, we could divide both sides by 5.

(b) We repeat the exact same computation but reduce modulo 2 at each step:

$$\begin{aligned}\boxed{x^4+x+1} \;&\equiv\; (x^2+1)\cdot\boxed{x^2+1} + x\\ \boxed{x^2+1} \;&\equiv\; = x\cdot\boxed{x} + 1\end{aligned}$$

Backtracking through this, we find that Bézout's identity takes the form

$$\begin{aligned}1 \;=\; 1\cdot\boxed{x^2+1} + x\cdot\boxed{x} \;&=\; 1\cdot\boxed{x^2+1} + x\cdot\left(\boxed{x^4+x+1} + (x^2+1)\cdot\boxed{x^2+1}\right)\\ &=\; (x^3+x+1)\cdot\boxed{x^2+1} + x\cdot\boxed{x^4+x+1}\end{aligned}$$

(c) We can now read off that $(x^2+1)^{-1} = x^3+x+1$ in $\mathrm{GF}(2^4)$.

Example 136. (HW) Find the inverses of x^2+1 and x^3+1 in $\mathrm{GF}(2^8)$, constructed as in AES.

Solution. Recall that for AES, $\mathrm{GF}(2^8)$ is constructed using $x^8+x^4+x^3+x+1$.

(a) We use the extended Euclidean algorithm for polynomials, and reduce all coefficients modulo 2:

$$\boxed{x^8+x^4+x^3+x+1} \equiv (x^6+x^4+x)\cdot\boxed{x^2+1} + 1$$

Hence, $(x^2+1)^{-1} = x^6+x^4+x$ in $\mathrm{GF}(2^8)$.

(b) We use the extended Euclidean algorithm, and always reduce modulo 2:

$$\begin{aligned}\boxed{x^8+x^4+x^3+x+1} \;&\equiv\; (x^5+x^2+x+1)\cdot\boxed{x^3+1} + x^2\\ \boxed{x^3+1} \;&\equiv\; x\cdot\boxed{x^2} + 1\end{aligned}$$

Backtracking through this, we find that Bézout's identity takes the form

$$\begin{aligned}1 \;\equiv\; 1\cdot\boxed{x^3+1} - x\cdot\boxed{x^2} \;&\equiv\; 1\cdot\boxed{x^3+1} - x\cdot\left(\boxed{x^8+x^4+x^3+x+1} - (x^5+x^2+x+1)\cdot\boxed{x^3+1}\right)\\ &\equiv\; (x^6+x^3+x^2+x+1)\cdot\boxed{x^3+1} + x\cdot\boxed{x^8+x^4+x^3+x+1}.\end{aligned}$$

Hence, $(x^3+1)^{-1} = x^6+x^3+x^2+x+1$ in $\mathrm{GF}(2^8)$.

Armin Straub
straub@southalabama.edu

Basics of AES

The block cipher AES (short for **advanced encryption standard**) replaced DES. By now, it is the most important symmetric block cipher.

1997: NIST requests proposals for AES (receives 15 submissions) [very different from how DES was selected!]

2000: Rijndael (by Joan Daemen and Vincent Rijmen) selected (from 5 finalists)

https://csrc.nist.gov/csrc/media/publications/fips/197/final/documents/fips-197.pdf

- 128 bit block size (as per NIST request)
- The key size of AES can be 128, 192 or 256 bit. The corresponding choices are referred to as AES-128, AES-192 and AES-256, and have 10, 12 and 14 rounds, respectively.
- AES-192/256 is first (and only) public cipher allowed by NSA for top secret information.
- No known attacks on AES which are substantially better than brute-force.

Attacks better than brute-force known if the number of rounds was 6 (instead of 10) for AES-128.

- Unlike DES, AES is not a Feistel network.

While for a Feistel network, each round only encrypts half of the bits, all bits are being encrypted during each round. That's one indication why AES requires fewer rounds than DES.

Internals of AES

Each round consists of 4 **layers**. Each layer takes 128 bits input and outputs 128 bits in a reversible way (so that we can decrypt as long as we know the key).The 128 bit state consists of 16 bytes. These 16 bytes $c_{0,0}, c_{1,0}, c_{2,0}, c_{3,0}, c_{0,1}, \dots, c_{3,3}$ are often arranged in a 4x4 matrix as

$$\begin{bmatrix} c_{0,0} & c_{0,1} & c_{0,2} & c_{0,3} \\ c_{1,0} & c_{1,1} & c_{1,2} & c_{1,3} \\ c_{2,0} & c_{2,1} & c_{2,2} & c_{2,3} \\ c_{3,0} & c_{3,1} & c_{3,2} & c_{3,3} \end{bmatrix}.$$

Each byte is identified with an element of $\mathrm{GF}(2^8)$.

Example 137. $(0000\ 0101)_2$ represents the element x^2+1 in $\mathrm{GF}(2^8)$.

The 4 layers are:

- ByteSub

  each byte gets substituted with another byte (like a single S-box in DES); provides confusion and guarantees non-linearity of AES

- ShiftRow

  the 16 bytes are permuted (like a P-box in DES but on bytes, not bits); provides diffusion

- MixCol

  the 4x4 matrix is linearly transformed; provides diffusion

- AddRoundKey

  the state is xored with a 128 bit round key

Slight deviations. Before the first round, AddRoundKey is applied with the 0th round key (which equals the AES key). Otherwise, our first step would be ByteSub, which wouldn't have any cryptographic effect since the plaintext bytes would just be changed in a fixed manner (no key involved yet).

Also, the last round has no MixCol layer. This has the effect that decryption can be made to look very much like encryption (see Section 5.3 in our book for the details).

ShiftRow

The layer **ShiftRow** permutes the 16 bytes $c_{0,0}, c_{1,0}, c_{2,0}, c_{3,0}, c_{0,1}, \dots, c_{3,3}$ as follows:

$$\begin{bmatrix} c_{0,0} & c_{0,1} & c_{0,2} & c_{0,3} \\ c_{1,0} & c_{1,1} & c_{1,2} & c_{1,3} \\ c_{2,0} & c_{2,1} & c_{2,2} & c_{2,3} \\ c_{3,0} & c_{3,1} & c_{3,2} & c_{3,3} \end{bmatrix} \mapsto \begin{bmatrix} c_{0,0} & c_{0,1} & c_{0,2} & c_{0,3} \\ c_{1,1} & c_{1,2} & c_{1,3} & c_{1,0} \\ c_{2,2} & c_{2,3} & c_{2,0} & c_{2,1} \\ c_{3,3} & c_{3,0} & c_{3,1} & c_{3,2} \end{bmatrix}$$

MixCol

Again, arrange the 16 bytes as a 4×4 matrix with entries in $\mathrm{GF}(2^8)$. The **MixCol** layer transform this 4×4 matrix by multiplying it with another, fixed, 4×4 matrix:

$$\begin{bmatrix} c_{0,0} & c_{0,1} & c_{0,2} & c_{0,3} \\ c_{1,0} & c_{1,1} & c_{1,2} & c_{1,3} \\ c_{2,0} & c_{2,1} & c_{2,2} & c_{2,3} \\ c_{3,0} & c_{3,1} & c_{3,2} & c_{3,3} \end{bmatrix} \mapsto \begin{bmatrix} x & x+1 & 1 & 1 \\ 1 & x & x+1 & 1 \\ 1 & 1 & x & x+1 \\ x+1 & 1 & 1 & x \end{bmatrix} \begin{bmatrix} c_{0,0} & c_{0,1} & c_{0,2} & c_{0,3} \\ c_{1,0} & c_{1,1} & c_{1,2} & c_{1,3} \\ c_{2,0} & c_{2,1} & c_{2,2} & c_{2,3} \\ c_{3,0} & c_{3,1} & c_{3,2} & c_{3,3} \end{bmatrix}$$

Example 138. For instance, the new byte at the position $c_{2,1}$ is

$$\begin{bmatrix} 1 & 1 & x & x+1 \end{bmatrix} \begin{bmatrix} c_{0,1} \\ c_{1,1} \\ c_{2,1} \\ c_{3,1} \end{bmatrix} = c_{0,1} + c_{1,1} + x\,c_{2,1} + (x+1)c_{3,1},$$

where all computations are to be done in $\mathrm{GF}(2^8)$.

AddRoundKey

The **AddRoundKey** layer simply xors the current 128 bit state with a 128 bit round key.

The **key schedule** for AES-128 is as follows. Like for the states, arrange the original 16 byte AES key k in a 4×4 matrix with columns $W(0), W(1), W(2), W(3)$.

The ith round key is then obtained from the matrix with columns $W(4i), W(4i+1), W(4i+2), W(4i+3)$, where $W(4), W(5), \ldots$ are recursively constructed:

$$W(i) = \begin{cases} W(i-4) + W(i-1), & \text{if } 4 \nmid i, \\ W(i-4) + \tilde{W}(i-1), & \text{if } 4 | i. \end{cases}$$

Here, $\tilde{W}(i-1)$ is obtained from $W(i-1)$ as follows:

$$W(i-1) = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix} \quad \Longrightarrow \quad \tilde{W}(i-1) = \begin{bmatrix} S(w_1) + x^{(i-4)/4} \\ S(w_2) \\ S(w_3) \\ S(w_4) \end{bmatrix}.$$

Note that w_1, w_2, w_3, w_4 each are bytes. The function S is the ByteSub substitution. Since that substitution is nonlinear, the round keys are constructed from k in a nonlinear manner (unlike in DES).

As usual, the computation $S(w_1) + x^{(i-4)/4}$ happens in $\mathrm{GF}(2^8)$.

ByteSub

The **ByteSub** layer takes each of the 16 bytes y and replaces it with the byte $S(y)$. As in DES, we could simply describe the (invertible) map by a lookup table. However, like the other steps of AES, it has a very simple mathematical description which we'll discuss next time.

Comment. As for the S-boxes in DES, ByteSub can be implemented in hardware as a lookup table. Since we have $2^8 = 256$ inputs, with 1 byte of output each, this table is 256 bytes large. (See Table 5.1 in our book.)

For comparison, each of the eight S-boxes in DES occupies 2^6 times 4 bits, which is 32 bytes. In total, these are also 256 bytes.

[In contrast to DES it is the case (and necessary for decryption!) that different inputs have different outputs.]