Magma Tutorial. Part II

Luis J Dominguez Perez

Cinvestav, Mexico

September, 2011

Luis J Dominguez Perez

Table of contents

Surprise hands-on

Luis J Dominguez Perez

Code the extended Euclidean algorithm

Rings

Supported types of rings

- IntegerRing(.)
- Integers()
- Rationals()
- RealField()

```
Z11:=IntegerRing(11);
w:=PrimitiveElement(Z11); w;
[w*i:i in [1..10]];
[w^i:i in [1..10]];
Order(w);
Category(w);
Parent(w);
```

Polynomial Rings

Luis J Dominguez Perez


```
Z:=Integers();
S:=PolynomialRing(Z);
AssignNames(~S, ["x"]);

S<x>:=PolynomialRing(Z);

S<x,y>:=PolynomialRing(Integers(),2);

Z<x>:=PolynomialRing(Rationals());
```

Hands-on

Luis J Dominguez Perez

Define the following polynomials:

```
Z<x>:=PolynomialRing(Integers());
px:=36*x^4+36*x^3+24*x^2+6*x+1;
z:=2^64;
p:=Evaluate(px,z);
rx:=x^8-x^4+1;
r:=Evaluate(rx,z);
```

The exercise:

- Write a short program to find a z such that p is a prime of 256 bits.
- Modify your program to find r *almost prime* of 256 bits.

Finite fields

- FiniteField()
- GaloisField()
- GF()

If the field size is a prime power, then it is preferred to use FiniteField.

Extension field (not yet covered in the Arithmetic Module)

```
ext<F | n>
ExtensionField< F, x | polynomial>
```

Elements in $\mathbb{F}_p$

Example usage of elements in a field

```
p:=101;
Fp:=FiniteField(p);
a:=Random(p);
b:=Fp!103;
c:=Fp!-1000;
(a+b) in Fp;
(b*c) in Fp;
a^(1999);
Fp!a^(1999);
a^(1999) eq Fp!a^(1999);
```

Other functions for finite fields

Luis J Dominguez Perez

Try these functions over Fp

- One(Fp);
- Identity(Fp);
- Zero(Fp);
- Fp.1;

Now, create a quadratic extension of Fp and try the same functions. Then:

```
a2:=Random(Fp2); a2;
myseq:=Eltseq(a2); myseq;
myseq[1]*1+myseq[2]*Fp2.1;
```

Hands-on

Luis J Dominguez Perez

Finite fields

- Generate a Finite field of 256 bits
- Generate a random element
- Find its inverse using Fermat's little theorem

Bonus

```
> p:=RandomPrime(30);
> p;
471164429
> Factorization(p-1);
[ <2, 2>, <7, 1>, <16827301, 1> ]
> K:=GF(p);
> m:=Random(K)^7;
> m;
264249637
> P<x>:=PolynomialRing(K);
> f:=x^7-m;
> Factorization(f);
[
    <x + 168185695, 1>,
    <x + 266248969, 1>,
    <x + 301801881, 1>,
    <x + 331419936, 1>,
    <x + 426819080, 1>,
    <x + 427671341, 1>,
    <x + 433675243, 1>
]
> fact:=$1;
```

```
> for i:=1 to 7 do
for> z := -Evaluate (fact[i][1], 0);
for> print z, "^7 -", m, "=", z^7 - m
for> end for;
302978734 ^7 - 264249637 = 0
204915460 ^7 - 264249637 = 0
169362548 ^7 - 264249637 = 0
139744493 ^7 - 264249637 = 0
44345349 ^7 - 264249637 = 0
43493088 ^7 - 264249637 = 0
37489186 ^7 - 264249637 = 0
>
```

Primitive element

Let g be a multiplicative group of order n, and $g \in G$. The order of g divides n.

If $b \in \mathbb{Z}_p^*$, then $b^{\Phi(n)} \equiv 1 \bmod n$, $\#\mathbb{Z}_n^* = \Phi(n)$.

Let p-prime, $b \in \mathbb{Z}_p$, then $b^p \equiv b \bmod p$, and $\Phi(p) = p - 1$. Also, $b \not\equiv 0 \bmod p$ and $\#\mathbb{Z}_p^* = p - 1$

Then, $\exists \alpha \in \mathbb{Z}_p^*$ s.t. $|\alpha| = p - 1$, such an α is called a *primitive element*.

Hands on

Exercise

- Make a program to find the primitive elements of a very small finite field, and verify the order of the generated group with your Euler totient function.

Finding primitive elements in $\mathbb{Z}_p^*$

An easy way to determine if a random element $\alpha \in \mathbb{Z}_p^*$ is primitive arises when the factorization (in prime powers) of $(p-1)$ is known.

Let $p = p_1^{e_1} \cdots p_k^{e_k}$

An element $\alpha \in \mathbb{Z}_p^*$ is primitive iff

$$\alpha^{(p-1)/p_j} \not\equiv 1 \bmod p.$$

Finding primitive elements in $\mathbb{Z}_p^*$ II

Proof

Let $d = |\alpha|$. We know $d|(p-1)$, and α is primitive iff $d = p - 1$.

Suppose $\alpha^{(p-1)/p_j} \equiv 1 \bmod p$ for some j, then $d \leq (p-1)/p_j$, so $d \neq (p-1)$.

Now, suppose $\alpha^{(p-1)/p_j} \not\equiv 1 \bmod p$ for $1 \leq j \leq k$. Suppose $d \neq p-1$, since d is a divisor of $p-1$, and $d < p-1$, $\exists p_j (1 \leq j \leq k)$ s.t. p_j is a divisor of $(p-1)/d$, but this implies d is a divisor of $(p-1)/p_j$.

Hence, $a^{(p-1)/p_j} \equiv a^d \equiv 1 \bmod p$, which is a contradiction $\square$

Finding primitive elements in $\mathbb{Z}_p^*$ III

One way to find a primitive element is using *Las Vegas* algorithm: choosing random α and testing them, until a primitive element is found.

There are exactly $\Phi(p-1)$ primitive elements in $\mathbb{Z}_p^*$, the probability to find a random primitive element is $\Phi(p-1)/(p-1)$.

Suppose p and p_1 are prime, and $p = 2p_1 + 1$. Suppose $\alpha \in \mathbb{Z}_p^*$ and $\alpha \not\equiv 1 \bmod p$, then α is a primitive element iff $a^{(p-1)/2} \not\equiv 1 \bmod p$.

Finding primitive elements in $\mathbb{Z}_p^*$ IV

Proof

Observe that $\alpha^{(p-1)/p_j} \equiv \alpha^2 \bmod p$, and $\alpha^2 \equiv 1 \bmod p$ iff $\alpha \equiv \pm 1 \bmod p$. Follow the previous slides. $\square$

Following

If $\alpha \not\equiv 1 \bmod p$, and α is not primitive, then $\alpha^{(p-1)/2} \equiv 1 \bmod p$, but then:

$$\begin{aligned}(-\alpha)^{(p-1)/2} &\equiv (-1)^{(p-1)/2}\alpha^{(p-1)/2} \bmod p\\ &\equiv (-1)^{(p-1)/2} \bmod p\\ &\equiv -1 \bmod p\end{aligned}$$

Then, $-\alpha$ must be primitive.

Gnuplot

In your research, you may end up needing to plot your results, either from Magma, Maple, C/C++ or Java.

Some languages have support for graphs, it is not the case of Magma, but we can use Gnuplot.

Gnuplot is a portable command-line driven graphing utility for Linux, Windows, and some other platforms. Other options are available.

Hands on

To plot, first we need data.

- Generate 3 finite fields of 80, 112, and 128 bits
- Generate 3 random elements in the field
- time the exponentiation for several exponents
- Generate a table with the results and output to a file

Setting up

```
set border 3

set terminal png picsize 512 512
set output "myfile.png"
set term postscript eps enhanced color
set output "myfile.eps"
```

At the end, one can convert the `epstopdf myfile.eps` to get an embeddable pdf for your article.

Setting up

```
set style line 1 linetype 1 linewidth 1 \
  pointtype 9 linecolor rgb "red"
set style dots...
set style point...

set key right bottom box

set title "My graph"
set xlabel "Equivalent AES security level"
set xrange [80:128]
set xtics 88,96,104,112,120,128
set mxtics 4
```

Luis J Dominguez Perez

Plotting

```
plot "Example" using 1:3 title 'Data1' with \
linespoints linestyle 1,\
     "Example" using 1:4 title 'Data2' with \
      linespoints linestyle 2,\
     "Example" using 1:5 title 'Data3' with \
      linespoints linestyle 3
```

Don't forget to `reset` your settings at the end of your plot!

Multiplot

```
set size 1,1
set origin 0,0
set multiplot
set xlabel "Security level"

set size 0.5,0.5

set origin 0,0.5
set ylabel "CPU cycles"
plot ...

set origin 0.5,0.5
set ylabel "Milli seconds"
plot ...
```

More plot types

If bored, have a look at...
`http://www.phyast.pitt.edu/~zov1/gnuplot/html/intro.html`[1]

[1] Obviously, you need some data to plot...

End

End of Part II

There's no part III