Magma Assignment 04

Combinatorics and Cryptography

Some 'simple' discrete logarithms mod p

Solve the following tasks using (as subroutine) a Magma function called `mydLog` which takes as inputs a triple of positive integers $(g, h, p) \in \mathbb{N} \times \mathbb{N} \times \mathbb{P}$ and returns the *discrete logarithm* of h to the base g in $\mathbb{Z}_p^*$. Remember that if g is a generator in $\mathbb{Z}_p^*$, then the discrete logarithm to the base g exists for each element $h \in \mathbb{Z}_p^*$, otherwise it exists only for elements in the subgroup $< g >$ of $\mathbb{Z}_p^*$ generated by g.

The use of Magma functions related to discrete logarithms is (of course) forbidden.

Task 1 [4 pts.]

Consider the simulation of the *Diffie-Hellman key-exchange protocol* showed during Lecture 6 between Professor and Davide, related to the following Magma code:

```
> p1:=NextPrime(2^37+1);
> p1;
137438953481

> R:=Integers(p1);

// this finds a generator of the cyclic multiplicative
// group (Z_p1)^*

> g:=PrimitiveRoot(p1);
> g;
3
```

```
> g:=R!g;
> a:=Random(p1);
> Rpub:=g^a;
> Rpub;
114534287914

> Dpub := R!47963704417;
> s:=Dpub^a;
```

Knowing that the shared secret `s` has been used to encrypt a 5-letter message using the method described during the lecture (see Section 2.3 of lecture notes):

(for every student different from Davide)

- recover Professor's private exponent `a` or Davide's private exponent `b` (using your Magma subroutine `mydLog`),
- reconstruct the shared secret `s` (using Magma),
- recover the message, knowing that the encrypted message is `FUAUM`;

(only for Davide [volunteering pays off])

- recover Professor's private exponent `a` (using your Magma subroutine `mydLog`).

Task 2 [1 pt.]

Test your function on a harder task. Execute `mydLog(g2,h2,p2)` and compute $\mathrm{dlog}_{g_2} h_2$ in $\mathbb{Z}_{p_2}^*$ where

```
p2:=1309169629861283354959 33;
g2:=1234567;
h2:=8987654321;
```

Task 3 [hard and not mandatory]

Test your function on an even harder task. Check if `mydLog(g3,h3,p3)` terminates in a reasonable time and compute $\mathrm{dlog}_{g_3} h_3$ in $\mathbb{Z}_{p_3}^*$ where

```
p3:=4143048263908946630850772177750697667379840457068343 93;
g3:=3;
h3:=350245296247225487828410712825238569959396265055090989;
```

Points

Submitting a working solution will give you up to five point, distributed as detailed above.

Hints

- The requested main function `mydLog` can be implemented in many ways. Find the most suitable method(s), choosing from those explained during the lectures. Please explain you method in detail. More efficient methods will allow you to solve more difficult tasks.
- These exercises will not be corrected using a code-correcting code. Please make sure that your code is readable or well commented.
- The orders of the involved groups can be factorized in Magma.
- Mind the order of the elements!
- The following Magma code will check if $g \in \mathbb{Z}_p^*$ is a generator:

```
>R:=Integers(101);
> IsPrimitive(R!3);
true

> IsPrimitive(R!9);
false

> IsPrimitive(R!27);
true
```

- All the requested tasks are solvable: the requested discrete logarithms exist and, provided that the correct algorithm is chosen, can be found in a short time.