Maths and Cryptography SAGE Lab Part F0

N. Courtois, UCL F2F sessions, University College London
n.courtois@cs.ucl.ac.uk

v1.3, 22 October 2020

This document is better viewed in PDF, it contains clickable links, and interactive exercises. Most exercises can also be done with paper pencil directly inside a printed paper version. Running some commands and bit of programming with SAGE software is required all the time.

The exercises are not graded. Students can compare their answers to other students, discuss answers in class or interact with the PDF which contains answer boxes and hints. In many cases the correct answer becomes obvious if we try some command with SAGE.

• Recommended PDF Viewer Software

We recommend either FoxIt reader or Adobe Acrobat or to simply open inside Microsoft Edge browser (tested and works).

1. Getting Acquainted with SAGE

SAGE Math is a free open-source maths software package based on Python, see `http://sagemath.org`. The leader of SAGE is Prof. William A. Stein from University of Washington who left his university in 2016 to work full time on his company SageMath Inc. SAGE is to a tool to experiment with any type of maths and applied maths (e.g. in number theory and cryptography). It is characterized by ultra compact syntax and compatibility. SAGE aggregates the expertise of 1000 mathematicians behind the scenes it embeds many earlier open source computer algebra software packages.

2. How to run a LOCAL version of SAGE?

We recommend to install SAGE natively on your own computer by downloading ready compiled executable SAGE installer from here, part of `http://sagemath.org`. A detailed installation guide (not needed) can be found here.

SAGE has 3 user interfaces:

1. **Jupyter notebook interface. BEST AND PREFERRED.**
   The documentation (not at all needed) can be found here.
2. Old legacy browser notebook interface: outdated, slow to refresh in browsers.
3. Command Line Interface (REPL) using SAGE terminal (type after prompt). REPL means Read-Eval-Print-Loop which is based on IPython.

• Native Install Method 1 for Windows

The best method for large projects/CPU and RAM intensive. Please download according to your OS.

Windows x64 native installer (using cygwin) can be found here, found inside `http://sagemath.org`. For faster speeds select a UK-based mirror.

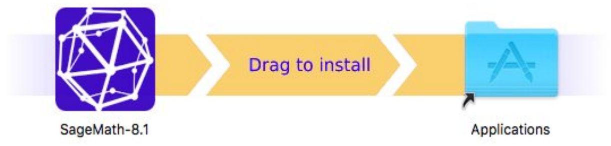
- **How to USE SAGE in Windows**

1. **Starting SAGE.** HOW to use this?
   3 shortcuts called "SageMath 9.1 ... " are created on your desktop. The only one we need is "SageMath 9.1 Notebook". Double click. Wait for 5 seconds, your browser will open.
   Tip: If you selected a non-standard directory for SAGE files and this shortcut fails and closes automatically, a second method is to use SAGE shell and type `jupyter notebook`. If all fails run it again with Administrator privileges.
2. **Opening New.** On the right click on button "New" / "NoteBooks" / "Sage 9.1" Wait for 1 s.
3. **Rename.** In the upper left corner click to rename the worksheet like "exo58_00".
4. **Directories.** A new directory can be created in New button.
5. **Saving.** Files are autosaved.
6. **Exporting Files.** Most author display these worksheet in plaintexts.
   A popular saving format to import from older versions of SAGE is SWS.
7. **Closing.** Just "Close and Halt to exit". Or use Quit button in upper right corner. Then close the browser or browser tab. If a black terminal is still open then type Ctrl+C and then type "exit".
8. **Re-Opening Old.** Restarting jupyter again. Then a file "exo58_00.ipynb" appears in the initial folder, just click to re-open. It is OK also for this file to be in another directory.
9. **Where are My Files.** In windows and by default the files are actually stored in: C:\Users\yourlogingname\.
   For SAGE this directory will be the same as `/home/sage/`.
   These files can be moved as they are to another computer, even with a different OS. This home directory can be freely chosen by the user at installation since version 8.5. This variant may sometimes require to run SAGE with admin privileges (right click or change shortcut properties forever).

- **Native Install Method 1 for MAC (not updated)**

See `http://www.sagemath.org/download-mac.html`, direct link here.
You need to pay attention to choose a correct version (app version) depending on the type of you CPU.

Install it (drag or move it to applications) and double click on the Sage icon.

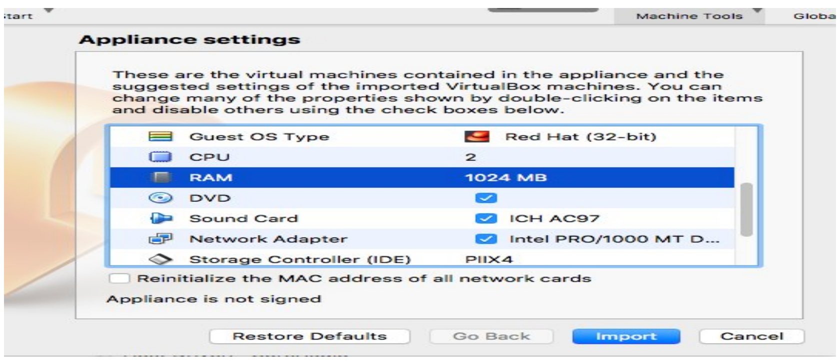


- **Method 2 - Using SAGE Math Cloud**

SAGE can be accessed online via the SAGE Math Cloud.

1. First, click the link above and enter your details to register for a free account.
2. Once you have logged in successfully, click ‘Projects’ in the top left-hand corner.
3. Click ‘New Project’. Add a title (e.g. ‘CryptoExo58’) and a description (e.g. ‘Code for COMPGA18‘) and click ‘Create Project’.
4. Click on the project you have just created.
5. Add a new file by clicking ‘New’. Choose a name for your file (e.g. ‘Lab Session 1’), and select ‘SageMath Worksheet’ as the type.
6. Now you are ready to write code! Type `3+4` and click run.
7. You can write many separate programs in the same window. You can also leave complex programs to run while you log out and do something else.

- **Method 3 - Portable and Clonable Machine with VirtualBox:**
  1. Any OS outside, Linux inside. Download links are here (version number may need to be changed):
     For Windows PC or old builds
     Here are installers for Mac OSX and also for Linux host OS.
  2. The obligatory **VirtualBox Extension Pack** must be downloaded and run AFTER the installation completed. It must have the same version. Click to download here.
  3. Then install SAGE machine/apppliance.
     (a) Method 3.1: one single OVA file. we import the latest OVA installer file from SAGE web site. For example huge file of 3.5 Gigabytes.
     In Virtual Box go to "Import appliance". When importing it is very useful to increase the RAM and the number of CPUs and also to reset MAC address.



Appliance settings

These are the virtual machines contained in the appliance and the suggested settings of the imported VirtualBox machines. You can change many of the properties shown by double-clicking on the items and disable others using the check boxes below.

Guest OS Type	Red Hat (32-bit)
CPU	2
RAM	1024 MB
DVD	✓
Sound Card	✓ ICH AC97
Network Adapter	✓ Intel PRO/1000 MT D...
Storage Controller (IDE)	PIIX4

☐ Reinitialize the MAC address of all network cards

Appliance is not signed

     more about this, in principle not needed.
     (b) Method 3.2: This method is better (for example) if you want to MOVE it from one machine to another or CLONE it with existing SAGE worksheets.

     Copy 2 files from another PC after installation. One is machine definition file, on my PC called "Sage-6.9.vbox" and the other one is the hard drive file, on my PC called sage-6.9-disk1.vmdk, (could also be .vdi, there are several options). Then you have got to "Add" menu option inside the VirtualBox software and add the .vbox file and the check settings/Storage if it points to the right hard drive, or it need also added manually wrt to the current location on your PC.

  More detailed instructions are here (not needed). For example we wanted to do some advanced modifications, change the Linux keyboard layout, make it remotely accessible etc.

- **SAGE and Python**

SAGE is built from Python. Most Python commands will work fine in SAGE, and details of more commands including number theory and algebra can be found here.

- **Python Hint**

In Python and so in SAGE, spaces in front of each line matter, so be careful when you copy and paste code.

- **Learning Python Programming and Learning Cryptography**

We recommend the book "Cracking Codes with Python" by Al Sweigart.

- **Basic Programming in Python under Windows**

In Windows Python can be downloaded from `www.python.org`. The latest version can be run locally from a command line. A better method to edit code is to type IDLE in the Windows search bar and open the IDLE editing environment. Then type

```
print('Hello')
```

this line is automatically executed, a bit similar as in SAGE.

3. Basic Algebra and Modular Arithmetic Crash Course

If you need some help about modular arithmetic and basic vocabulary about groups fields and rings we recommend our UCL crash course on these topics, see http://www.nicolascourtois.com/bitcoin/groups_ECC_7B.pdf and here is direct link: by clicking here.

Quiz Answer each of the following. We do not use SAGE yet, we will do it on the next page. In printed paper version, write your answer inside the box.

In PDF version write the answer inside the box and press enter.

Click on the "Ans" button to get a hint or a partial solution.

1. $3 - 6 \mod 5 =$ ☐ Ans

2. $7 \times -1 \mod 11$

☐ Ans

Definition 1. *We say that set M with an operation $\circ : M \times M \to M$ is a group if it is*

0. closed under $\circ$

1. associative: $(x \circ y) \circ z = x \circ (y \circ z)$

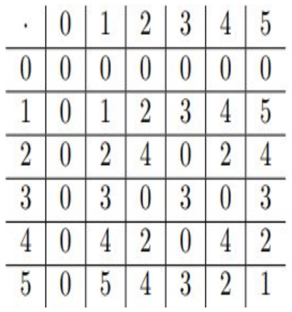
2. has a neutral element **e**

3. each element has an inverse/reciprical for $\circ$

Definition 2. *If we omit the last condition (3.) we call it a monoid.*

3. Is $\mathbb{Z}_6$,+ with addition modulo 6 a group?

☐ True ☐ False



$\cdot$	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	1	2	3	4	5
2	0	2	4	0	2	4
3	0	3	0	3	0	3
4	0	4	2	0	4	2
5	0	5	4	3	2	1

4. Is $\mathbb{Z}_6, \cdot$ with multiplication modulo 6 a group?

☐ True ☐ False

Definition 3. *We consider any group with a multiplicative notation. We call a divisor of zero a non-zero element x such that $\exists y \neq 0$ s.t. $xy = 0$ or $yx = 0$.*

5. How many elements of $\mathbb{Z}_6, \cdot$ are divisors of zero?

☐ 0 ☐ 2 ☐ 3 ☐ 5

List them with commas and without spaces: = ☐

6. How many elements of $\mathbb{Z}_6, \cdot$ are invertible?

☐ 0 ☐ 2 ☐ 4

List them with commas and without spaces: $=$ ☐

Ans

3.1. SAGE and Primes

Theorem 1. *$\mathbb{Z}_N - \{0\}, \cdot$ is a group if and only if N is a prime.*

Quiz Try to guess the answers given by SAGE software. In doubt type the commands and check.

1. list(primes(3,11))[1]
We obtain one integer $=$ ☐ Ans

2. These three constructions are essentially the same:

```
sage: type(Zmod(5))
sage: type(Integers(5))
sage: type(IntegerModRing(5))
```

3. For which of these SAGE commands the result will be strangely NOT at all reduced modulo 5?

1. $(3 + 4)\%5$
2. $3 + 4\%5$
3. $3 * 4\%5$
4. Zr=Zmod(5); Zr(3+4)
5. Zr(3)+Zr(4)
6. F = Integers(5);F(3)+4

The answer is ☐

3.2. SAGE and Rings and Fields

In maths a ring is a set R with two operations denoted by $+$ and $\cdot$. Neutral elements are called 0 and 1. We also requite that:

Definition 4. *A set $R, +, \cdot$ forms a ring if:*

1. $0 \neq 1$

2. *$R, +$ is a* **commutative group** *with $a + b = b + a$.*

3. *$R - \{0\}, \cdot$ is a monoid: without the inverse axiom (3.).*

4. *Distributive law:*
$a(b + c) = ab + ac$
$(b + c)a = ba + ca$.

With two more conditions we get a field:

Definition 5. *A set $R, +, \cdot$ forms a field if:*

5. *$R - \{0\}, \cdot$ is commutative $ab = ba$.*

6. *It is actually a group, because each non-zero element is invertible.*

Quiz

1. Which of these rings is a finite field?

1. Zmod(14); factor(14), divisors(14)
2. ZZ.cardinality()
3. QQ.cardinality()
4. QQ in Fields(); RR in Fields()
5. ZZ in Fields(); ZZ in Rings()
6. FiniteField(12)
7. GF(11).cardinality()

Here GF stands for Galois Field and is the same as FiniteField. The answer is

Theorem 2. *$\mathbb{Z}_N$ is always a ring and it is a field if and only if N is a prime.*

2. Can you predict a result of these SAGE commands? You can run commands with combination of "Shift" + "Enter" keys.

```
sage: Zr=Zmod(5); inv3=Zr(3)^-1; inv3
sage: inv3*6
sage: 2.0*6
sage: parent(inv3)
```

The answer in the first question is $inv3$

3. Fermat's little theorem: we have

$$2^{11} = 2 \mod 11$$

and

$$3^{11} = 3 \mod 11$$

and

$$a^{11} = a \mod 11$$

etc for any integer a. This means that a^{11-1} is either 0 or 1.

Definition 6. *We say that a is a* **primitive root** *mod p if all the numbers $1, a, a^2, a^3, \ldots a^{p-1}$ are distinct values mod p and it generates the whole multiplicative group.*

This is not systematic for example $3^5 = 1 \bmod 11$.
Find one single element that generates $\mathbb{Z}_{11}, \cdot =$

Ans

It could be helpful to type these SAGE commands. The first two lines have the same effect.

```
G11=FiniteField(11)
G11=GF(11) #the same as above
list(G11)
G11(7).multiplicative_order()
primitive_root(11)
[G11(2)^i for i in range(0,11)]
```

4. Is the answer unique? Propose another alternative answer. $=$

5. What is the Discrete Logarithm of 5 in basis 2 in $\mathbb{Z}_{11}, \cdot$? Hint: $2^x \overset{?}{=} 5$.
$=$

Ans

3.3. Counting Invertible Elements mod N

Theorem 3. *In general the number of elements in $\mathbb{Z}_N$ is equal to $\phi(N)$ and this is exactly those elements invertible modulo N, which function was invented by Euler (totient function).*

Quiz Try to guess these answers.

1. We say that a and b are co-primes if $gcd(a, b) = 1$.
a is invertible modulo 10 if and only if $gcd(a, 10) = 1$.
Which number is not co-prime with 10?

☐ 1 ☐ 2 ☐ 3 ☐ 9 ☐ 7

3. Is $\mathbb{Z}_{10}$ a field?

☐ True ☐ False

4. Given the following code:

```
sage: for n in range(1, 21):
sage:     if gcd(n, 20) == 1:
sage:         print(n),
```

What is the value of $\phi(20)$? = ☐ Ans

5. `sage: euler_phi(10)*euler_phi(2)`
Can you guess what we obtain? ☐ Ans

6. We recall the Fermat-Euler theorem:

Theorem 4. *If $GCD(a, n) = 1$ then*

$$a^{\phi(n)} = 1$$

and it is never 1 when $GCD(a, n) \neq 1$.

Def. An order of an element a is the smallest $x > 1$ such that a^x becomes 1.
A famous theorem by Lagrange says that order of an element divides the order of a group. But it can be smaller than $\phi(n)$. What is the order of 9 in $\mathbb{Z}_{20}, \cdot$?

☐ 1 ☐ 2 ☐ 4 ☐ 8

7. What is the order of 7 in $\mathbb{Z}_{20}, \cdot$?
It is = ☐

```
sage: R = Zmod(20); R(7).multiplicative_order()
```

Ans

4. Black Box Groups - Elliptic Curves over Finite Fields

In SAGE it is very easy to define and study elliptic curves. An elliptic curve is a method of defining a group structure over a set points (blue dots on our graph). In addition we add one more point called ∞ or the point at infinity. To define a curve group we need a prime p, and two integer coefficients a, b. For example $p = 5, a = 1, b = 3$.

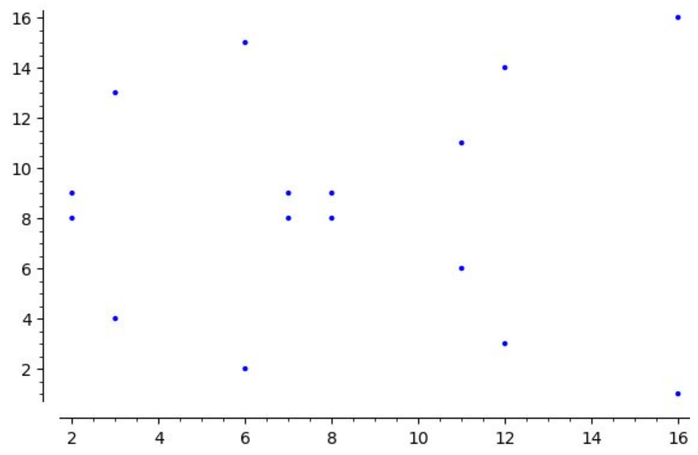
```
sage: E = EllipticCurve(GF(17), [1,3])
sage: plot(E, rgbcolor=(0,0,1))
```

This set of points behaves essentially as a set of random points expect it is symmetric w.r.t. axis x. It satisfies the equation:

$$y^2 = x^3 + ax + b$$

Remarks. Here the plot resembles a continuous "curve", because we cross a boundary with multiples of 17 not very frequently. For a larger p we will cross this boundary more frequently so the word "curve" is a bit of misnomer in finite fields mod p. It works better for elliptic curves over real (floating) numbers. In maths, by convention, a curve means something different than for ordinary people. Mathematicians call a curve any set of points which satisfies some polynomial equation. For a large p this set of points will look more like a galaxy of blue points scattered at random (except it is symmetric).

Point are sometimes stored in short **compressed** form: we just story x and one bit saying if y is even or odd.



Quiz

(a) We have $x = 12$ and y is even, what is the value of y??
The answer is

(b) Usually a + sign is used and we talk about point addition but this is purely conventional, we could also use the $\cdot$ sign. A group law is defined but it is expected to behave like the result of an addition of 2 points operation is essentially random. Points can be defined by pairs or triples of integers where the last integer is 1 by default except for the special point ∞. What is the result of:

```
sage: E(2,8)+E(3,4)
```

The answer is $\boxed{}$

(c) Guess without running it the result of:

```
sage: E(2,8)+(E(3,4)+E(11,6))
```

The answer is $\boxed{}$

(d) Guess the result of:

```
sage: 17*E(2,8)
```

The answer is $\boxed{}$

(e) Without running this command, guess the result of:

```
sage: 16*E(2,8)
```

The answer is $\boxed{}$

(f) What is the representation $x : y : z$ of the point at infinity?
The answer is $\boxed{}$

Special Curves

Some elliptic curves are very special and should be avoided.

```
sage: E.is_ordinary(), E.is_supersingular()
```

In simplified terms, a supersingular curve is one which is isomorphit o one which is unusually simple and abnormal behaviour results. In general if something is special it should be avoided in cryptography, as a dedicated attack could exist (backdoored prime, backdoor curve etc).

4.1. Walks on Curves and ECDLP

If we add one element to itself we obtain $P, 2P, 3P, \ldots$ and this will give essentially a random walk, see slide 57 in here. Typically in cryptography parameters are carefully selected so that this random walk has exactly q steps where q is a very large prime for example on 256 bits. In our example $p = q$ which is because with small numbers we can be lucky. This never happens in practice for large numbers, and $p \neq q$ and the difference large. A famous Hasse theorem from 1933 says that q is an integer inside

$$q \in [p + 1 + 2\sqrt{n}, p + 1 - 2\sqrt{n}].$$

It is in fact more or less uniformly distributed. Very few people have realized that it CAN go very slightly above $2\times$:

```
p=127;
for n in range(6, 50):
   k=(EllipticCurve(GF(p),[1,n])).cardinality();
   if (k>144 or k<110): print( n,float((p+1-k)/sqrt(k))  )
```

Quiz

(a) For the curve from previous exercise which violates the bound,

```
E=EllipticCurve(GF(?),[1,?])).cardinality()
factor(???)
```

How many points has this curve?
The answer is

(b) How many points of order 2?
The answer is

(c) Find an integer x point such that $(x, 1)$ lies on this curve.
The answer is

(d) Compute a point of order 2 which is not ∞.

```
??*E(???,1)
```

(e) Find a generator point of an odd prime order.

```
g=??*E(???,1); g.order()
```

(f) Complete the code for computing a discrete log with a brute force method. What is the discrete logarithm of (110,110) in the basis $(x, 1)$.

```
for n in range(1, 127):
   if ( n*E(???,1) == E(110,110)): print( n  )
```

The answer is

4.2. Exercise to do at Home for Advanced Students

We recall that

$$q \in [p + 1 + 2\sqrt{n}, p + 1 - 2\sqrt{n}].$$

It is easy to see that with the bitcoin elliptic curve seccp256k1 with (a=0,b=7) we have:

```
sage: p=115792089237316195423570985008687907853269984665640564039457584007908834671663
sage: q=115792089237316195423570985008687907852837564279074904382605163141518161494337
```

Private keys in bitcoin are integers modulo q. If you know the right integer you can spend your bitcoins. Just one such number can be worth billions of dollars. If a crypto developer confuses these two numbers inside his bitcoin client producing digital signatures which operations are done modulo q, the results can be devastating. Try to imagine different attack scenarios with standard ECDSA or Schnorr signatures, assuming that the public key of a bitcoin wallet is known to the attacker. Send me your solution to n.courtois@cs.ucl.ac.uk.