University of London International Programmes
CO3326 Computer Security 2016-17

Coursework assignment 1

IMPORTANT: all students will receive an email containing a unique set of numbers to use for this coursework. We will be sending these out immediately after posting the assignment. These will be sent the email address you have registered with the University of London (you can check what address you have registered via the Portal.) If you do not receive this email, please check your spam folder, before emailing us at: (intcomp@gold.ac.uk)

The aim of this coursework is to show evidence of an understanding of what Elliptic Curve Cryptography (ECC) is, how it works and why it is considered secure.

Today, we can find elliptic curves cryptosystems in TLS, PGP and SSH, which are three of the main technologies on which the modern web and IT world are based (not to mention Bitcoin and other cryptocurrencies).

The web has plenty of information on the subject, so you will not find it difficult to research. You are not required to dive deeper into the mathematics of the subject than that covered in your subject guide (modular arithmetic). Some simple maths covered in high school (geometry and simple group theory) will be needed though. Your research should cover topics including:

- what is an elliptic curve,
- how to define a group law in order to do mathemtical operations with points on the elliptic curve,
- how to restrict elliptic curves to finite fields of integers modulo a prime - with this restriction, you'll see that the points of elliptic curves generate cyclic subgroups and you'll understand the terms: *base point*, *order* and *cofactor*.

Please write your report using the following skeleton:

Elliptic curves over real numbers

1. Define briefly what elliptic curves are and plot the elliptic curve based on the *a* and *b* parameters you were given.
2. Explain in simple terms, using your own words, the group law for elliptic curves and demonstrate
    - geometric addition and
    - scalar multiplication with arbitrary points on your curve.

Elliptic curves over finite fields

1. Explain in simple terms, using your own words, how elliptic curves are restricted to a finite field $\mathbb{F}_k$.

2. Explain what *order* of the field means and calculate the order of your field, using *k* that you were given ($\mathbb{F}_k$).
3. Using your curve and your field, that you have calculated, demonstrate:
    - point addition, by calculating $R = P + Q$ with the values you were given (points *P* and *Q*)
    - scalar multiplication, by calculating $S = n * P$ over the field $\mathbb{F}_k$, using the same point *P* as above and *n*, both of which you were given.
4. Finally, in the ECC context explain, using your own words, what the easy problem is, and what seems to be hard problem; explain how this relates to the discrete logarithm problem.
5. Explain how ECC can be used for cryptography, and more specifically for key exchange.

Show all your work. You must include in-text citation and provide a detailed references section at the end of your coursework (see How to avoid plagiarism in Study Support on the VLE.) Please note, that there is no need to copy-paste entire explanations or mathematical proofs. A simple sketch that demonstrates your understanding is enough and it should be based around the calculations with your own numbers (the numbers you were given).

Submission requirements

The report should be submitted as a PDF document, following a *strict naming scheme*: *StudentName_{srn}_CO3326_cw1.pdf*. For example, if your name is *Mark Zuckerberg* and your SRN is *000000001*, your report submission will be *MarkZuckerberg_000000001_CO3326_cw1.pdf*. Your report will count as **60%** of your CW1 mark.

In addition to your report, you will also submit a JSON file, which also follows a *strict format* and *naming scheme* and summarizes the results of your calculations. This will count as **40%** of your CW1 and will be automatically checked by an algorithm, so pay particular attention to its format.

The name of the file should be *StudentName_{srn}_CO3326_cw1.json*; for example, if your name is *Mark Zuckerberg* and your SRN is *000000001*, your JSON submission will be *MarkZuckerberg_000000001_CO3326_cw1.json*. You can use the following well-formed JSON and adapt it for your numbers and your calculation results.

A hypothetical student, *Mark Zuckerberg*, with SRN *000000001*, received the following numbers:

SRN	Name	a	b	k	Px	Py	Qx	Qy	n
000000001	MARK ZUCKERBERG	-2	13	103	19	97	27	81	3

He would submit the following JSON, which reflects a correct solution:


```
{
  "name": "MARK ZUCKERBERG",
  "srn": "000000001",
  "ecc": {
    "a": -2,
    "b": 13,
    "k": 103,
    "order": 109
  },
  "assignment": {
    "modk-add": {
      "p": {
        "x": 19,
        "y": 97
      },
      "q": {
        "x": 27,
        "y": 81
      },
      "r": {
        "x": 61,
        "y": 90
      }
    },
    "modk-mul": {
      "n": 3,
      "p": {
        "x": 19,
        "y": 97
      },
      "s": {
        "x": 63,
        "y": 46
      }
    }
  }
}
```


Using his *a* and *b* values, the generic elliptic curve, defined by $y^2 = x^3 + ax + b$, becomes $y^2 = x^3 - 2x + 13$. This curve is used throughout the coursework.

NOTE: You may rely on any graph plotting package (or online tool) to plot curves (for example, try typing "x^3 - 2x + 13" into Google). You will have to demonstrate the mathematical operations (geometric addition and scalar multiplication) using points of your choice on your curve. The calculations, showing step by step detail, should be included in your report. Please remember, that calculation results are best shown using plots.

He uses $k = 103$ to restrict the curve onto the $\mathbb{F}_k = \mathbb{F}_{103}$ prime field. He determines the *order* of the field, by working out how many points the discrete curve has (including the point at infinity.) In this case it is 109 (observe `"order": 109` in the JSON result).

NOTE: This will most probably involve some programming. You may choose a programming language of your liking, whatever you're the most comfortable with. Most programming languages have very good support for JSON parsing and output

(including Java, C#, Python and JavaScript), so you can even rely on your programme to produce the required JSON output.

Then, using his 2 points, *P(19, 97)* and *Q(27, 81)*, he does the modulo addition (`modk-add`) over the $\mathbb{F}_k = \mathbb{F}_{103}$ field: *R = P + Q*, which gives = *R(61, 90)*. Note the result under the `modk-add` hash in the JSON result. He also does the modulo multiplication (`modk-mul`): *S = nP*, using *n = 3* and *P(19, 97)*, which gives = *S(63, 46)*. Note the result under the `modk-mul` hash in the JSON result.

[END OF COURSEWORK ASSIGNMENT 1]