Encryption

Daan Snoeken

July 2020

1 Encryption

Let's say we have a random string we want to encrypt:

```
Daan Snoeken
```

We can convert this into numbers according to the ASCII table and to make sure we always have 3 digits for each character, we add 101 to the value of each character and an additional value every time the position of a character in the string increases, in this case 1 is picked. However, we may also pick larger numbers to add. Next we concatenate all the numbers we get in order. This will give us the following number:

$$a_0 = 1691992002141371892172192102172122222. \tag{1}$$

Now consider the following first degree polynomial,

$$y = a_0 + a_1 x \tag{2}$$

and note that the value of a_0 is the same in eq. 2 as in eq. 1. Next, pick a random value for a_1 and find two points at e.g. $x = 1$ and $x = 2$ that lie on this line. These will be your encrypted numbers. Only somebody who knows both numbers can figure out what the original number a_0 was that you started with. This can of course be extended to higher order equations, e.g.

$$y = a_0 + a_1 x + a_2 x^2 + ... + a_n x^n. \tag{3}$$

Note that for a polynomial of degree n, we need (at least) $n+1$ points to describe it. In all cases, a_0 needs to be the original number (eq. 1) and all the others can be picked at random. As was done with the linear equation, pick random x values for every point and solve equation 3 for each x value and save the output. Now you can store every number on a separate location and only when someone has all the numbers, will they be able to solve the original message. So, if you and your friend have a secret, you can do this and give one number to your friend. Then keep one number yourself. Now neither of you can decrypt the message on your own, but you can solve it when you are together.

2 Decryption

I will describe the specific case first and then a general case.

2.1 Specific case

In order to decrypt a certain message, remember that we have two points. Let's say the points P_1 and P_2 are defined by the following coordinates:

$$\begin{aligned} P_1 &= (1, 32701467143075497732091078049795910 9), \\ P_2 &= (2, 48483114465037677043060935878771600 7). \end{aligned} \tag{4}$$

Since there are only two points, our best guess is that the original was formed using a first degree polynomial (eq. 2). So in order to get the value for a_0, we need to derive the value of a_1 from the given points. This is defined by:

$$a_1 = \frac{\Delta y}{\Delta x} = \frac{P_2^{(y)} - P_1^{(y)}}{P_2^{(x)} - P_1^{(x)}}, \tag{5}$$

where $P_n^{(y,x)}$ refers to the y and x coordinate of the n'th point, respectively. Then filling in all the values gives

$$a_1 = 157816473219621793109698578289756898. \tag{6}$$

Then a_0 will be

$$a_0 = P_1^{(y)} - a_1 = 169198198211133184211212202208202211. \tag{7}$$

Note that each character will be represented by three digits and that the ASCII value can be generated by subtracting 101. Doing this will get our original string back, which was

```
Daan Snoeken
```

2.2 General case

Let's say we are given any number of points, defined by

$$\{P_1 = (x_1, y_1), P_2 = (x_2, y_2), ..., P_n = (x_n, y_n)\}. \tag{8}$$

These points define the n'th degree polynomial, defined by

$$y(x) = a_0 + a_1 x + a_2 x^2 + ... + a_n x^n. \tag{9}$$

In order to decrypt the message, a_0 must be found. Combining eqs. 8 and 9 gives a system of equations with n unknowns and n equations defined by

$$\begin{aligned} y_1 &= a_0 + a_1 x_1 + a_2 x_1^2 + ... + a_n x_1^n \\ y_2 &= a_0 + a_1 x_2 + a_2 x_2^2 + ... + a_n x_2^n \\ &\vdots \\ y_n &= a_0 + a_1 x_n + a_2 x_n^2 + ... + a_n x_n^n. \end{aligned} \tag{10}$$

Note that all the values for x_n and y_n are known from the points defined in eq. 8. Now solve the system of equations shown in equation 10. This should give a value for a_0 that gives the numerical representation of the original string. Once again, this number should be split into three digits for each character and can then be converted by subtracting 101 and using the ASCII table.