Final Exam

1. (a) Describe the simple substitution cipher. **(2 points)**

   (b) Alice and Bob are planning to use a simple substitution cipher. They want a key which encrypts vowels to vowels, consonants to consonants and (to avoid debates) Y to Y. How many choices do they have? **(4 points)**

Solution. (a) In the simple substitution cipher, both $\mathcal{M}$ and $\mathcal{C}$ are set of the letters of the alphabet $\mathfrak{A}$:

$$\mathcal{M} = \mathcal{C} = \mathfrak{A} = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}, \mathtt{e}, \mathtt{f}, \mathtt{g}, \mathtt{h}, \mathtt{i}, \mathtt{j}, \mathtt{k}, \mathtt{l}, \mathtt{m}, \mathtt{n}, \mathtt{o}, \mathtt{p}, \mathtt{q}, \mathtt{r}, \mathtt{s}, \mathtt{t}, \mathtt{u}, \mathtt{v}, \mathtt{w}, \mathtt{x}, \mathtt{y}, \mathtt{z}\}.$$

The key set $\mathcal{K}$ is the group of permutations of $\mathfrak{A}$:

$$\mathcal{K} = \{k : k \in \mathrm{Perm}(\mathfrak{A})\}.$$

Given a letter, the key k acts on it via the permutation, i.e.

$$e_k(m) = k(m).$$

As for the decryption, it is given by the inverse permutation. Formally,

$$d_k(c) = k^{-1}(c).$$

(b) Apart from Y, there are 5 vowels and 20 consonants. Their key is a pair of two permutations: the first component permutes the 5 vowels, the second one permutes the 20 consonants. Since these can be chosen independently from each other, the number of pairs in total is simply the product $5! \cdot 20!$.

2. (a) Describe the elliptic curve Diffie-Hellman key exchange. **(2 points)**

   (b) Let the base field be $\mathbf{F}_7$, and consider the elliptic curve E given by the equation $X^3 + XZ^2 = Y^2 Z$. How many points lie on E? **(4 points)**

Solution. (a) The elliptic curve Diffie-Hellman key exchange is the following. Alice and Bob agree on a (large) prime number p, an elliptic curve E over $\mathbf{F}_p$, and a point $P \in \mathbf{F}_p$. Then Alice chooses a (large) positive integer n_A (keeps it in secret), computes

$$Q = n_A P = \underbrace{P + \ldots + P}_{n_A \text{ times}},$$

and sends it to Bob. In the meantime, Bob similarly chooses a secret n_B, computes $R = n_B P$, and sends it to Alice. Now Alice computes $n_A R$, Bob computes $n_B Q$. Both of them arrives at $n_A n_B P$, which will be their key (for example, for a later cryptosystem). An eavesdropper should compute $n_A n_B P$ from the information $n_A P$ and $n_B P$.

(b) First, the ideal point $\mathcal{O}$ is on E, and this is the only point of E which is not in the affine xy-plane. So we may consider the equation $x^3 + x = y^2$. We see that (computing in $\mathbf{F}_7$) $x = 0$ gives $y = 0$; $x = 1$ gives $y = 3, 4$; $x = 2$ gives no y; $x = 3$ gives $y = 3, 4$; $x = 4$ gives no y; $x = 5$ gives $y = 2, 5$; $x = 6$ gives no y. Together with $\mathcal{O}$, this is 8 points.

3. (a) Describe the RSA crpytosystem. **(2 points)**

   (b) Alice uses an RSA, but Eve learns that the prime numbers are chosen in the unfortunate way that $p = a + 1$, $q = a^2 + 1$ for some $a \in \mathbf{N}$. How can Eve break the cryptosystem in polynomial time? (Clarification: Eve does not know what a is, she only knows that there is such a positive integer a.) **(4 points)**

Solution. (a) Alice takes two (large) prime numbers p, q, then computes their product N. She also computes $\varphi(N) = (p-1)(q-1)$. Then she takes an exponent $e \in \mathbf{N}$ coprime to $\varphi(N)$, and computes its inverse d modulo $\varphi(N)$. She publishes N, e and keeps $p, q, \varphi(N), d$ in secret.

Now anyone (say, Bob) can send her a message m (a residue class modulo N) using the following protocol. Bob raises the message to power e modulo N and sends $c \equiv m^e \bmod N$ to Alice.

Now Alice raises the incoming cipher c to power d modulo N. With high probability, m is coprime to N, and then, by Euler-Fermat,

$$c^d \equiv (m^e)^d \equiv m^{\varphi(N)u+1} \equiv (m^{\varphi(N)})^e \cdot m \equiv 1 \cdot m \equiv m \bmod N,$$

which is the original message.

(b) Eve can do the following. The value a can be revealed by binary search, since $N = a^3 + a^2 + a + 1$, that is, N is a monotone function of a. Also, this function is computable in polynomial time. Therefore, binary search applies (starting out from the interval $[0, N]$, we can always halve it by trying the midpoint, after $O(\log N)$ steps, the interval containing a gets smaller than 1). Then p, q are computable and Eve can play the role of Alice.

4. (a) Define entropy. **(2 points)**

   (b) Can the entropy be infinite? (Clarification: in view of that for finitely many probabilities, the answer is obviously no, your task is to decide in the infinite setup. That is, are there nonnegative numbers $p_1, p_2, \ldots$ summing up to 1 such that the calculated entropy sums up to infinity?) **(4 points)**

Solution. (a) The entropy function H is defined on finite sets of positive numbers summing up to 1, i.e. on tuples $(p_1, \ldots, p_n) \in \mathbf{R}_+^n$ if $p_1 + \ldots + p_n = 1$, for any $n \in \mathbf{N}$. For such a tuple,

$$H(p_1, \ldots, p_n) = -\sum_{j=1}^{n} p_j \log_2 p_j.$$

(b) Yes, we construct such a setup. Our plan is to give a decomposition, for each $n \in \mathbf{N}$,

$$2^{-n} = \underbrace{p_n + \ldots + p_n}_{b_n \text{ times}},$$

such that the b_n many p_n altogether contribute 1 to the calculated entropy. Then since there are infinitely many n's, the total entropy will be infinite. Obviously $p_n = 2^{-n}/b_n$, then the contribution for a fixed n is

$$-\sum_{j=1}^{b_n} p_n \log_2 p_n = -b_n \frac{2^{-n}}{b_n} \log_2 \frac{2^{-n}}{b_n} = 2^{-n} \log_2(2^n b_n).$$

Clearly, if we choose b_n large enough, for example, taking $b_n = 2^{2^n}$, then

$$2^{-n} \log_2(2^n b_n) = 2^{-n} \log_2(2^n 2^{2^n}) > 2^{-n} \log_2(2^{2^n}) = 2^{-n} 2^n = 1,$$

which shows that one can construct infinite entropy.