Exercise 9.1 (Properties of hash functions). (7 points)

Let \( h_1 \) and \( h_2 \) be two hash functions. Let \( h = h_1 \mid h_2 \) be the concatenation of them.

(i) Prove that if at least one of \( h_1 \) and \( h_2 \) is collision resistant, then \( h \) is collision resistant. \( \square \)

(ii) Determine whether an analogous claim holds for second pre-image resistance and inversion resistance, respectively. Prove your claims. \( \square \)

Now assume \( h \) is any collision resistant hash function.

(iii) Is the composition \( h \circ h \) necessarily collision resistant? \( \square \)

Exercise 9.2 (Energy cost). (0+4 points)

Estimate the total energy consumed by performing \( 2^{128} \) computations of the SHA-256 compression function with modern high-end CPUs. Extrapolate that to 10, 20, 30 years from now. Do the same for \( 2^{256} \) and \( 2^{512} \) such computations. \( \square +4 \)

Exercise 9.3 (The ElGamal signature scheme). (12 points)

In this exercise you will get some hands-on experience with the ElGamal signature scheme.

Let \( p = 2^{28} + 3 \) and \( g = 3 \) a generator of \( G = \mathbb{Z}_p^* \). The injective encoding function \( G \to \mathbb{Z}_{p-1}, x \mapsto x^* \) is given by

\[
    x^* = \begin{cases} 
        0 & \text{for } x = p - 1 \\
        x & \text{else.}
    \end{cases}
\]

Our message \( m \) will be the ASCII-string "2014".

(i) Look up the 7-bit ASCII encodings for each letter and concatenate them for the 28-bit number \( m \). \( \square \)
Let us take the role of Alice and let \( a = 100 \) be our secret key.

(ii) Choose a random session key \( k \) (of at least three digits) and generate a signature for your message \( m \).

(iii) What is your public key? Use it to verify the signature you just produced.

We will now explore how Eve can sign a given message if additional information is provided.

(iv) Alice sends the signed message

\[
(m, x, b) = (500, 10296631, 248708422).
\]

By accident the secret session key \( k = 787 \) is revealed. Compute Alice’s secret key \( a \).

(v) After this experience, Alice changes her secret key and the public version is now \( y = 138309740 \). Unfortunately a bug/feature in the random number generator revealed that the same value for \( k \) was generated twice in a row. This is known for the signed messages

\[
(501, 32067479, 51030675)
\]

and

\[
(502, 32067479, 60076072)
\]

Compute Alice’s secret key.