3. Exercise sheet

Hand in solutions until
Wednesday, 19 November 2014, 23:59:59

Note the modified hand-in time!

Exercise 3.1 (Properties of hash functions). (6 points)

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

(i) Is \( h \) collision resistant if at least one of \( h_1 \) and \( h_2 \) is collision resistant? 

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

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

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

Exercise 3.2 (A discrete log hash function). (8 points)

A prime number \( q \) so that \( p = 2q + 1 \) is also prime, is called a Sophie Germain prime. We choose \( q = 7541 \) and \( p = 2 \cdot 7541 + 1 \) both prime and want to define a hash function on the set \( \mathbb{Z}_q \times \mathbb{Z}_q \).

(i) Let \( g = 604 \) and \( z = 3791 \). Prove that \( \text{ord}(g) = \text{ord}(z) = q \).

The elements \( g \) and \( z \) actually generate the same subgroup of \( \mathbb{Z}_p^\times \), i.e. \( \langle g \rangle = \langle z \rangle \). Call this subgroup \( G \).

(ii) Now, we can define a hash function

\[
h : \mathbb{Z}_q \times \mathbb{Z}_q \to G, (a, b) \mapsto g^a z^b.
\]

Compute \( h(7431, 5564) \) and \( h(1459, 954) \) and compare them.
(iii) In (ii) you found a collision for the hash function \( h \). This enables you to compute the discrete logarithm \( \text{dlog}_g z \). Do it.

(iv) Show the converse: If you can compute discrete logarithms in \( G \), find a way to generate a collision.

(v) Explain whether you would employ such a hash function in practice.

**Exercise 3.3 (Expected number of iterations).** (9 points)

We are given a discrete random variable \( X \), for example the result of a single roll of a fair die. The values that \( X \) can take are denoted by \( x \) and the respective probability is given by \( \text{prob}(X = x) \). For the example, the \( x \) are taken from the set \( A = \{1, 2, 3, 4, 5, 6\} \) each with \( \text{prob}(X = x) = \frac{1}{6} \).

We are interested in the expected value \( E(X) \) defined as

\[
E(X) = \sum x \cdot \text{prob}(X = x),
\]

where the sum is taken over all possible values of \( X \). In the example above, this returns as the expected value for the roll of a single die

\[
E(X) = \sum_{x \in A} x \cdot \frac{1}{6} = \frac{21}{6} = 3.5.
\]

Next, we roll the die until a certain number, say "2", appears for the first time. The random variable \( Y \) is now the number of rolls that are performed, until this happens.

(i) What is \( \text{prob}(Y = i) \), i.e. the probability that "2" appears for the first time in the \( i \)th roll?

(ii) Prove that \( E(Y) = 6 \). (You may have use for the generalization of the formula for the geometric series \( \sum_{k=n}^\infty q^k = q^n/(1 - q) \) for \(|q| < 1 \).

(iii) Generalize the preceding steps to prove the more general proposition

**Proposition.** Suppose that an event \( A \) occurs in an experiment with probability \( p \), and we repeat the experiment until \( A \) occurs. Then the expected number of executions until \( A \) happens is \( 1/p \).

**Exercise 3.4 (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.