Crypto 2023. Preparation for the final exam (2nd half of the semester).

Exercise 1. Let p be a prime number. Prove that $1 \cdot 2 \cdot 3 \cdot \ldots \cdot (p-1) = -1 \mod p$. For example, for $p = 5$ we have

$$1 \cdot 2 \cdot 3 \cdot 4 = 24, \text{ and we see that } 24 = -1 \mod 5.$$

Exercise 2. Let p, q, r be three (pairwise distinct) prime numbers, each of them is strictly greater than 2, and $n = p \cdot q \cdot r$.

(a) Prove that if $a^2 = 1 \mod n$, then $a^2 = 1 \mod p$.

(b) Prove that there exists 8 numbers $x_1, \ldots x_8$ in the set $\{1, 2, \ldots n-1\}$ such that $x_i^2 = 1 \mod n$.

(c) Let $n = 17 \cdot 19 \cdot 23$. Find at least three different numbers x in $\{1, \ldots, n-1\}$ such that $x^2 = 1 \mod n$.

Exercise 3. Let $n = 41 \cdot 47$ and $k = 3$. Let as take the pair (n, k) as a public key of the scheme RSA. Find the corresponding private key.

Exercise 4. (a) Prove that every pseudo-random generator is a one-way function.

(b) Prove that if there exist one-way functions, then not all of them are pseudo-random generators.

Exercise 5. Assume that there exists a randomized polynomial time algorithm $\mathcal{A}$ such that for every composite number n (represented by its binary expansion), $\mathcal{A}(n)$ with a probability $> 1/2$ returns a non-trivial factor k of n (i.e., $k \neq 1$, $k \neq n$, and k divides n). With a probability $< 1/2$ the algorithm may return a number that is not a factor of n.

Prove that there exists another randomized polynomial time algorithm $\mathcal{B}$ such that for every composite number n (again, represented by its binary expansion), $\mathcal{B}(n)$ with a probability $> 99/100$ returns a non-trivial factor k of n.

Exercise 6. Let $H : \{0,1\}^* \to \{0,1\}^*$ be a function computable in polynomial time by a deterministic algorithm such that for every $x \in \{0,1\}^*$ of even length the value $y = f(x)$ is a binary string twice shorter than x.

Assume that this function is *not pre-image resistant* in the following sense. There exists a polynomial-time algorithm $\mathcal{A}$ such that for every n, for a randomly chosen $x \in \{0,1\}^n$, with probability > 0.1 on the input $y = f(x)$

$$\mathcal{A}(y) \text{ returns an } x' \text{ such that } f(x') = y$$

(algorithm $\mathcal{A}$ finds an f-pre-image of y, which is possibly not equal to the original x).

(a) Prove that this function is *not collision-resistant*: there exists a polynomial-time algorithm $\mathcal{B}$ such that for every even number n

- with probability > 0.1 : $\mathcal{B}(n)$ stops in poly(n) steps and returns two numbers x_1, x_2 of length n such that $x_1 \neq x_2$ and $f(x_1) = f(x_2)$ (i.e., $\mathcal{B}$ finds a *collision* for f)
- with probability < 0.9 : $\mathcal{B}(n)$ returns symbol $\perp$

(b) Prove a stronger property: there exists a polynomial-time algorithm $\mathcal{B}'$ that for every even number n

- with probability > 0.99 : $\mathcal{B}'(n)$ stops in poly(n) steps and returns two numbers x_1, x_2 of length n such that $x_1 \neq x_2$ and $f(x_1) = f(x_2)$ (i.e., $\mathcal{B}$ finds a *collision* for f)
- with probability < 0.01 : $\mathcal{B}'(n)$ returns symbol $\perp$

Comment: If you are a student attended the course, by December 24 you can request a solution of one of these exercises *in exchange to your own solution of any other exercise from this list.*