PWr, Inf. algo. CRYPTOGRAPHY, 2022 assignments list # 2, 9.3.2022, update

1. Assume that S is an encryption scheme S for n-bit blocks creating n-bit ciphertexts. Each key is a m-bit sequence.

   Observe that each encryption key defines a permutation of the set of n-bit block. What can you say about m, if for every permutation π there is a key K so that π is equivalent to encryption with K?

2. Let p stand for the probability to invert a one-way function F. Assume that the computation necessary for inversion takes 1 second, and that you are using one value of F per second.

   What should be the value of p so that you may expect that the adversary will not invert any of your values of F during your lifetime?

3. Propose a one-way function $F : \{0,1\}^{32} \to \{0,1\}^{256}$.

4. In most systems users are authenticated by providing the login and password. How to implement this, if you fear that the system admin is dishonest and sells the users' passwords on the black market?

5. [homework] Explore what are the possibilities to compute discrete logarithm in Python. Search for the libraries that can do it. Once you find something, just test it and find out where it stops to provide the results within, say, a minute.

6. If F is a one-way function, then $F(b)$ cannot be regarded as a commitment for a bit b. However, we have discussed so called Pedersen commitment. Does it mean that we cannot use automatically a one-way function as a commitment of bits?

7. Suppose that F is a one-way function which maps 2048-bit strings to 1024-bit strings. You have a database D with 1024-bit records.

   You have to present a single 1024-bit string so that **later** you can prove for each single record from D that it was really contained in D. You should not reveal other records within the proof.

   Hint: use F, create a tree.

8. You may try some factoring tool (e.g. `www.alpertron.com.ar`). As you observe in most cases the factorization of your input comes very fast. Why? What can be said about average factorization time for n bit numbers?

9. Assume that there is no (polynomial) distinguisher that can distinguish between a PRNG X and a really random generator R. Assume also that there is no (polynomial) distinguisher that can distinguish between a PRNG Y and R.

   Does it follow that there is no such distinguisher that distinguishes between X and Y?

10. Assume that PRNG X is not distinguishable from a random source. Prove that the unpredictability property holds for X.

/-/ Mirosław Kutyłowski