ALGORITHMICS, CRYPTOGRAPHY & COMPUTER SECURITY, graduate programs 2023
CRYPTOGRAPHY, 2022, assignments list # 3, 21.3.2023

1. Check what the consequences are of flipping a single bit of the input of Keccak's function f, say $a[0,0][0]$. You have to find which positions in the 3-dimensional matrix a may be affected. Follow the situation after each step of the first round of f. If the number of affected positions is low, then trace the 2nd round of f.

   Background for the question: A good hash function has the property that for similar inputs, the outputs do not look similarly. So, we will check how far a single round of function f helps to achieve this property. (Let me recall that f consists of $12 + 2\ell$ rounds, if Keccak's internal state consists of $5 \times 5 \times 2^\ell$ bits).

2. (a) Show that the f function of Keccak is invertible.

   (b) Consider an adversary that gets access to the internal state of Keccak (say, at the sponge phase). Then, of course, the adversary can derive the hash bits that are created afterwards. But can we recover the hash bits that were created before this moment? Discuss the potential consequences (brain storming in the class, please!)

3. Poly1305 is a modern authenticator for messages (and an alternative to the simple HMAC presented during the lecture). For a message m and secret key r, s, it works as follows:

   - message m is divided into 16-byte words $m_1, \ldots, m_q$,
   - for each i, get a 17-byte number c_i by adding byte 1 to m_i at the most significant position
   - create a polynomial $F(x) = c_1 \cdot x^q + c_2 \cdot x^{q-1} + \cdots + c_q \cdot x$
   - compute $\mathsf{Poly1305}_r(m) := F(r) \bmod 2^{130} - 5$
   - compute $a := (\mathsf{Poly1305}_r(m) + s) \bmod 2^{128}$

   Remarks: $2^{130} - 5$ is a prime number. There are some restrictions on r (certain bits are 0).

   - check how many messages of length $16q$ bytes may have the same authenticator for r, s; is it (nearly) true that the preimage of each authenticator value has the same cardinality?
   - if r, s are random, what is the probability that the authenticators for given messages m, m' are equal?
   - can r, s be reused? How to ensure that s will not repeat for two messages? (redesign the scheme)

4. Ethereum cryptocurrency uses so-called Proof-of-Stake to determine the roles in a group on nodes: one random node will create a new block of blokchain while the rest will check it.

   (a) Design a scheme based on random number generator, commitments to determine the roles: there are nodes $A_1, \ldots, A_n$ and assign them roles $R_1, \ldots, R_n$ at random. Assume that there is a shared unpredictable but public value r available to each node during the execution of the protocol. (e.g., r might be the current stock exchange index) The goal is to avoid any advantage of the Byzantine nodes trying to influence the assignment of roles. (Note that the solution discussed during the lecture is imperfect in this sense.)

   (b) If you have time, check the mechanism used by Ethereum. Is it fair and resilient to coalitions of nodes trying to influence the choice of the roles in an unfair way?

/-/ Mirosław Kutyłowski