International Institute of Information Technology, Hyderabad.
Principles of Information Security

Evaluation I

March 27, 2020

Due: **April 3, 2020**.

Instructions : Two Evaluation sheets will be released every week (on Tuesdays and Fridays). Each evaluation sheet consists of three categories of questions, namely: **[P]** stands for *programming* assignment, **[Q]** stands for *question* with written solution to be submitted and **[R]** stands for *research* problem. You need to submit the source-code for **[P]** along with a screen-recorded video that demonstrates its execution and for **[Q]** you may submit a pdf-file solution, all by the due-date. The research problems are *optional*, and anyone who solves any *one* of the **[R]** problems among *all* evaluation sheets will directly be awarded an **A**-grade.

[Q] Design a zero-knowledge proof for the Discrete-Logarithm Problem (DLP), that is, given prime p, generator g and the element $y = g^x \bmod p$, how does a prover claiming to know x, convince the verifier, without revealing x? Moreover, using hash-functions (and assuming them to be random oracles) show how would to build a *digital signature* scheme based on your above zero-knowledge proof and the hardness of DLP? Also, show how would you design collision-resistant hash functions based on the hardness of DLP.

[P] Implement (in any popular programming language of your choice) your newly designed digital signature scheme in **[Q]** above including a function/method for choosing a random prime p of length n, a function/method for collision-resistant *Hashing* and a function/method for *Signing* the hash of the message, and a function/method for *Verifying* the messsge and its signature.

ALL THE BEST

[R] Let p be a prime such that $p \bmod 4 = 3$. Define a sequence of numbers $d_1, d_2, d_3, \ldots$ as: $d_1 = \frac{p-3}{4}$ and d_{i+1} is defined based on d_i as:

$$d_{i+1} = \begin{cases} \frac{p-3-d_i}{2} & \text{if} \quad d_i < \frac{p-3}{2} \text{ and } d_i \text{ is even.} \\ \frac{p-4-d_i}{2} & \text{if} \quad d_i > \frac{p-3}{2} \text{ and } d_i \text{ is odd.} \\ \frac{2p-5-d_i}{2} & \text{if} \quad d_i \leq \frac{p-3}{2} \text{ and } d_i \text{ is odd.} \\ \frac{2p-6-d_i}{2} & \text{if} \quad d_i \geq \frac{p-3}{2} \text{ and } d_i \text{ is even.} \end{cases}$$

Given p and an integer y, the question is: does there exist an index $i < \frac{p-3}{2}$ such that $d_i = y$. In other words, consider the language $L = \{\langle p, y\rangle | \exists i < \frac{p-3}{2}, y = d_i\}$. Either design an *efficient* algorithm (that is, polynomial-time in $\log p$) for deciding L, *or* prove that L is **NP-hard**.