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

Evaluation II

March 31, 2020

Due: **April 7, 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] To store k blocks of data/information (say each block is of b bits) in a fault-tolerant way, you may encode the k blocks into n blocks (using some error-correction code) such that if any e of the n blocks are corrupted, it is still possible to retrieve the original k blocks of information. Specifically (for large enough b), coding theory suggests that this is possible if and only if $n \geq (k + 2e)$. However, *show that using digital signatures, it is possible to achieve the above fault-tolerant storage even when* $(k+e) \leq n < (k+2e)$, assuming a PPTM-adversary and a negligible probability of error is permitted.

[P] Implement (in any popular programming language of your choice) your newly designed fault-tolerant storage scheme to store any given data, by dividing the data into k blocks and encoding them into n blocks tolerating upto any e erroneous blocks where $e \leq (n-k)$, using the solution from **[Q]** above, and your own collision resistent hash function and signature scheme (implemented by you in *Evaluation I*).

ALL THE BEST

[R] *General Secure Fault-Tolerant Storage:* Given a monotone function $f : \{0,1\}^n \to \{0,1\}$, is it possible to design a storage scheme where a block of plaintext data can be encocded into n blocks such that for any PPTM adversary that chooses to corrupt all the blocks in any subset $E \subset [1, 2, \ldots, n]$ where $f(E) = 0$, the following hold: (a) *Confidentiality:* the adversary is oblivious of the plaintext data (even under CPA; you may need to define the security accordingly via a indistinguishability game) and (b) *Integrity:* it is still possible to retrieve the original plaintext data (from the n blocks) and (c) *Efficiency:* the complexity of reconstructing the plaintext is bounded by a polynomial in $t(n)$, where $t(n)$ is the time-complexity of f (that is, the fastest algorithm that computes $f(E)$ for any given subset E, runs in time $O(t(n))$) — or alternatively, you may characterize (by giving a necessary and sufficient condition for) the set of all monotone functions f for which a polynomial (in $t(n)$) retrieval is possible.