LELEC 2770 - Session 2 - ORAM

October 10, 2017

Problem 1. Consider the following scheme to obtain an ORAM for a 100 blocks memory:

- At the start the ORAM computes a random permutation π. All the blocks are shuffled according to π.
- When the i-th memory block is queried, the ORAM picks the $\pi(i)$ memory block along with nine other memory blocks selected randomly with indexes $(j_1, ..., j_9)$. Those ten blocks are read from the server, they are decrypted and rencrypted with a fresh random value. The i-th block is returned to the client.
- Then, the indexes $(i, j_1, ..., j_9)$ are shuffled according to a random permutation ρ (which is the identity on the 90 memory position whose blocks were not picked). And the blocks with indexes $(\rho(i), \rho(j_1), ..., \rho(j_9))$ are writen to the server.
- Now the blocks are shuffled according to $\rho \circ \pi$.

Prove that the scheme is secure and compute its efficiency or show an explicit attack against this scheme.

Problem 2. Consider the file binaryTreeOram.py. The code implements the Binary-Tree ORAM and the query function. Write the evict function.

Problem 3. Consider the pathOram.py implementing the pathORAM protocol. By changing the size of the client stash, the size of the buckets, the number of children of the nodes and the number of memory blocks stored in the tree (number of words in the code) obtain empirical observations about the probability that a the client stash remains of constant size, or not. Use and complete the benchmark method of benchmarking.py and the matplotlib.pyplot module.