DATE: 2019/01/08

Department of Electrical and Electronic Information Engineering		ID	M153257
Name	Hiroki Fujita		

Supervisor	Shuichi Ichikawa Naoki Fujieda

Abstract

Title	An investigation of light-weight implementations of Path ORAM and its security

Oblivious RAM (hereinafter referred to as ORAM) is a technology to prevent an access pattern analysis, which cannot be prevented by an encryption-based secure processor, by using dummy access and data rearrangement. Path ORAM has been extensively studied as a lightweight ORAM protocol. Since Path ORAM has been pointed out that there is significant bandwidth overhead, research to reduce it has been widely done. However, there are few discussions about the deterioration of security due to the application of the reduction method.This research focuses on pseudo random number generators (PRNGs) in Path ORAM and aims to quantitatively evaluate the security.

Path ORAM manages data blocks with a binary tree organized in external memory. Blocks along a route from the root to a leaf in the tree is called a Path. A data block is mapped to one path and stored to one of the blocks in that path. An access to the block is translated to reading and writing of the mapped path. The correspondence relationship between Paths and blocks is managed by internal memory called Position Map. After that, the access sequence to blocks is replaced with an independent access sequence to Paths, and the access pattern is concealed. If a series of Paths with high randomness can be generated using a light-weight PRNG engine, the amount of hardware for the Path ORAM architecture can be reduced while maintaining security. In this research, a security requirement is defined that a Path sequence for block access is statistically random. Even though the random numbers generated by the PRNG engine are of low quality, the requirement can be met if the Path sequence is sufficiently random.

For the evaluation, a simple Positon Map simulator is made from scratch. As it only reproduces the operation of Positon Map, a sequence of Paths corresponding to the input addresses is written out along with a sequence of random numbers from a PRNG. Statistic properties of sequences of Paths and random numbers are evaluated. As a result, that randomness of Path sequence is maintained in some PRNG engines even if the original random number sequence is of low quality. In response to this result, experiment was carried out in another ORAM simulator closer to actual Path ORAM. There, methods to reduce bandwidth overhead, Treetop Caching and Last Path Caching, were apopted to Path ORAM. They have a cache to the binary tree, where accesses of the tree can be omitted when the target block is found in the cache. As a result, the difference of the number of passed tests by adopting them was within margin of error.

Finally, a part of Path ORAM controller is implemented on an FPGA, using three PRNG engines, and the amount of hardware is evaluated. As a result, the effect of the selection of PRNG engines to the whole controller was modest: the maximum difference of the numbers of LUTs and Flip-flops were 193 and 130, respectively.