Abstract No: 277 *IT, Mathematics and Statistics*

Implementing a leakage-resilient storage scheme and a refreshing protocol to prevent continuous leakage attacks

D. Eranga, N. Jayanath, C. Somathilaka, J. Alawatugoda* and R. Ragel

Department of Computer Engineering, Faculty of Engineering, University of Peradeniya, Sri Lanka
**janaka@ce.pdn.ac.lk*

Although the cryptographic schemes are designed in a way that they are hard to break computationally, leaking information from their implementations (timing, EM radiation, power traces) may give sufficient power to the attacker to break the system by fully or partially recovering the secret parameters such as secret keys. Such attacks are known as side-channel attacks. The secret data can leak to the attacker while they are stored in the memory or involved in computations. The leakage-resilient cryptography aims to design proven-secure cryptographic schemes against side-channel attacks.

If the secret value has less number of bytes, the attacker can obtain bounded amount of bytes from a side-channel attack, and get rest of the bytes by brute-forcing. In this work, we implement a leakage-resilient (LR) storage scheme and its refreshing protocol. The LR storage scheme can securely store a secret in the memory against side-channel attacks (bounded memory leakage attacks), and the refreshing protocol can protect the secret from repeatedly occurring side-channel attacks (continuous leakage attacks). Above LR storage scheme and protocol work as follows: The LR storage scheme expands the number of bytes of the secret into very large amount of bytes, without damaging the actual value. Then the attacker has to steal a lot of bytes to recover the secret. Usually, such a large amount of bytes cannot be obtained by side-channel attacks. If the attacker obtains continuous leakage, then he has a chance of revealing a large amount of bytes of the secret. Therefore, the refreshing protocol continuously refreshes the expanded value, without damaging the actual secret value.

In order to achieve high security, we have to sacrifice the efficiency by introducing additional computations in expanding the secret value and refreshing. As a solution for that, we use the GPU for those computations by implementing them in CUDA. As a real-world application of this implementation we can integrate the LR storage scheme and the refreshing protocol with a Diffie-Hellman-based key exchange protocol and RSA algorithm, to implement them in the leakage-resilient manner.

This project is supported by the National Research Council (Grant NRC 16-020).