Cryptanalysis of SIDH Cryptosystem using Heterogeneous Hardware II

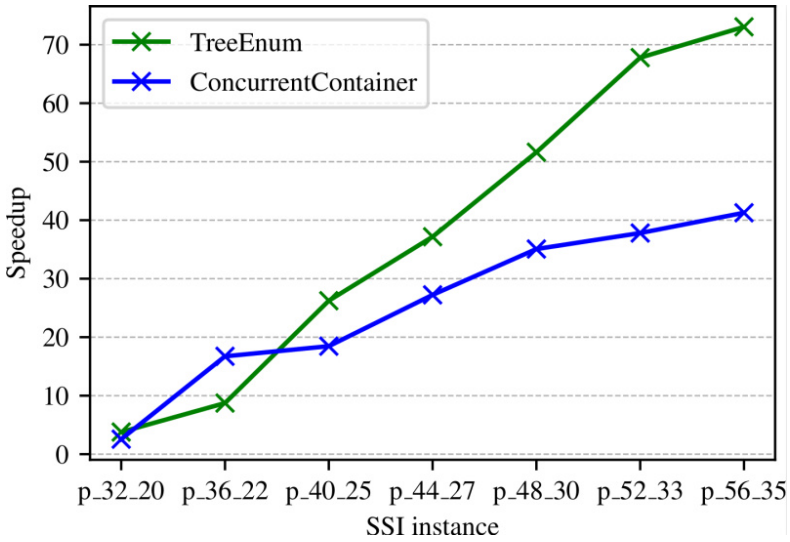


Figure 1: Speedup achieved by our proposed approaches TreeEnumeration and ConcurrentContainer for the Meet-in-the-Middle attack on seven SSI instances. Experiments conducted using 96 threads running on a 96-cores CPU.

Introduction

The project focuses on the cryptanalysis of the Supersingular Isogeny (SSI) path problem, which underlies the isogeny-based post-quantum cryptography. The target hardware architecture is multicore CPUs with shared memory.
The SSI path problem is a hard problem corresponding to the problem of finding a path connecting two nodes in graphs with exponentially many nodes.
The fastest known attack against the SSI path problem is the Meet-in-the-Middle attack that generates two isogeny trees from two given nodes and searches for a collision on the two sets of leaf nodes. This algorithm demands memory of exponentially large size, which renders it unrealistic
for large problem sizes.
The van Oorschot-Wiener (vOW) attack, a random walk collision-finding algorithm, uses memory more efficiently than the MiTM attack at the cost of longer runtime. In theory, vOW is perfectly parallelizable, while in practice, the parallelization aspect of vOW on shared-memory NUMA systems is not sufficiently studied.
We also consider the generation of random torsion points of given orders on elliptic curves, which serves as an utility in many isogeny-based cryptographic schemes.

Methods

Project Manager
Giang Nam Nguyen

Principal Investigator
Prof. Dr. Christian Bischof

Project Term
2022 - 2023

Clusters
Lichtenberg Cluster Darmstadt

Additional Software
OpenMP

Institute
Institute of Scientific Computing

University
Technische Universität Darmstadt

JUSTUS-LIEBIG-UNIVERSITÄT GIESSEN
TECHNISCHE UNIVERSITÄT DARMSTADT
UNIKASSEL VERSITÄT
Philipps Universität Marburg

We aim to improve the node-level performance of the vOW attack by employing multithreading more efficiently than the up-to-date implementation which is published in [1].
Thus, we analyze the state-of-the-art implementation mentioned above using performance analysis tools available on the Lichtenberg cluster, such as gprof and Intel vTune.
Based on the analysis results, we identify the isogeny tree generation as the most
computing-intensive building block in code. We conclude that this building block does not utilize the CPU cores available.
To that end, we apply a range of shared-memory parallelization techniques supported by OpenMP and thread-safe data structures to accelerate this building block.
For the torsion point generation, we consider a prototype written in SageMath, a computer algebra system, and follow a similar performance engineering approach with the help of the profiling methods and the multithreading support in Python and SageMath.

Results

The MiTM attack and the vOW attack share a compute-intensive building block called the isogeny tree generation, which is conceptually similar to a tree with a constant branching factor.
Regarding the isogeny tree generation, we develop a task-based parallel version with OpenMP, which runs an order of magnitude faster than the state-of-the-art implementation.
The container runtime Singularity on the Lichtenberg cluster provides HPC users a wide range of options to develop applications in an HPC environment without concern about the incompatibility of the software stack. The containerization eases the deployment of codes that require a different software stack than that currently available on the Lichtenberg cluster. This work demonstrates a manageable effort to run a containerized application written in the computer algebra system SageMath in a HPC environment.

Discussion

Our parallelization approaches for the isogeny tree generation show speedups ranging from 40 to 75, which is quite promising on the test system with 96 CPU cores running 96 threads.
Because there is a trade-off in the memory demanded by the precomputation and the memory needed for the random walk phase, it is still open which choice of the precomputation depth in vOW leads to an optimal speed up.

Publications

Nguyen, G.N. Bischof, C.: "Task-based Parallelization Approach for Attacking the Supersingular Isogeny Path Proble", In Proceedings of the 2023 Australasian Computer Science Week (ACSW '23). Association for Computing Machinery, New York, NY, USA, 40–49, 2023
https://doi.org/10.1145/3579375.3579381

Nguyen, G.N.: "Task-based Parallelization Approach for Attacking the Supersingular Isogeny Path Problem”, ACSW 2023, 2023

Reference

[1] Costello, C.; Longa, P.; Naehrig, M.; Renes, J.; Virdia F.: "Improved Classical Cryptanalysis of SIKE in Practice. In Public-Key Cryptography – PKC 2020, Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas (Eds.). Springer International Publishing, Cham, 505–534, 2020 https://doi.org/10.1007/978-3-030-45388-6_18

Last Update: 2023-03-17 12:38