Automatic Verification of Assembly Implementations in Crypto Software

Jay Lim
Dept. of Computer Science

5/3/2018 at 10:00 am
Hill 482 Conference Room

Abstract

Cryptographic software such as OpenSSL and BoringSSL implements cryptographic algorithm in assembly language for performance resulting in roughly eighty thousand lines of heavily optimized assembly instructions. The correctness of the implementations is vital to cryptographic softwares and the security of the internet, but proving the correctness is a hard problem. We show how to address this challenge by automatic equivalence checker for heavily optimized assembly implementations and the specification of the implementation. Our technique is able to verify the core algorithm of SHA-256, ChaCha20, and AES128

Examination Committee: Prof. Santosh Nagarakatte (Chair), Prof. Richard Martin, Prof. Uli Kremer, Prof. Bahman Kalantari