SCVM: An Efficient, Automated RAM-Model Secure Computation Framework

Memory Trace Oblivious Program Execution. [CSF’13]
Chang Liu, Michael Hicks, Elaine Shi

Automating Efficient RAM-Model Secure Computation. [S&P’14]
Chang Liu, Yan Huang, Elaine Shi, Jonathan Katz, Michael Hicks

Oblivious Data Structures. [CCS’14]
Xiao Shaun Wang, Kartik Nayak, Chang Liu, T-H. Hubert Chan, Elaine Shi, Emil Stefanov, Yan Huang

SCORAM: Oblivious RAM for Secure Computation. [CCS’14]
Xiao Shaun Wang, Yan Huang, T-H. Hubert Chan, abhi shelat, and Elaine Shi.

More to come soon!
(A Subset of) Our Team

Chang Liu
Kartik Nayak
Xiao Shaun Wang
T-H. Hubert Chan (HKU)
Yan Huang (IUB)

Jonathan Katz
Michael Hicks
Elaine Shi
“One year ago, we took four months to design efficient oblivious algorithms for matrix factorization, and implement them on a garbled circuit backend.”

— Nina Taft (Distinguished Scientist) and Udi Weinsberg (Researcher)

Technicolor Research
Our Ultimate Goals

Usability

Non-expert programmers can accomplish secure computation tasks in a few hours.
Our Ultimate Goals

**Usability**
Non-expert programmers can accomplish secure computation tasks in a few hours.

**Formal security**
Guaranteed through type systems.

**Efficiency**
Competitive to customized circuits for a large class of algorithms.
Compile-Time Optimizations

Instruction-trace obliviousness:
Eliminate universal next-instruction circuit

Memory-trace obliviousness:
Minimize use of ORAM

Mixed-mode execution
Local computation for local/public data

[Liu et al. Oakland 14]
Watch out for our open source release!

SCVM Compiler

Efficient ORAM Constructions

Efficient Garbled Circuit Backend
Watch out for our open source release!

- **SCVM Compiler**
- **Rich Libraries**
  - data structures, floating point, machine learning, matrix operations, graph algorithms
- **Efficient ORAM Constructions**
- **Efficient Garbled Circuit Backend**
Thank You

wangxiao@cs.umd.edu