Secure Computations of Trigonometric and Inverse Trigonometric Functions

Fattaneh Bayatbabolghani[1], Marina Blanton[2], Mehrdad Aliasgari[3], and Michael Goodrich[4]

[1]Computer Science and Engineering, University of Notre Dame
[2]Computer Science and Engineering, University at Buffalo (SUNY)
[3]Computer Engineering and Computer Science, California State University, Long Beach
[4]Computer Science, University of California, Irvine

Motivation

- There is a need to compute trigonometric and inverse trigonometric functions on private data in a number of applications such as secure fingerprint recognition.

Goals

- To develop new and efficient secure protocols for trigonometric and inverse trigonometric functions such as the sine and the arctangent functions.
- To develop secure protocols in both the two-party and multi-party computational settings in the semi-honest adversarial model.

Secure Protocols

- We build our solutions based on garbled circuit evaluation techniques for the two-party setting and linear secret sharing techniques for the multi-party setting.
- Our solutions use for fixed-point arithmetic which are represented using l bits, k of which are stored after the radix point.

Sine Protocol

- We use $xP(x^2)$ to approximate sine function for some polynomial P over variable x.

Input: a
Output: $\text{Sin}(a)$
Computation:

1. Apply a range reduction on a to compute x where $0 \le x \le 1$ and keep range reduction information
2. Compute $w = x^2$
3. Lookup the minimum polynomial degree N which precision of approximation is at least k' bits
4. Lookup polynomial coefficients $p_0, \dots, p_N$ for sine approximation
5. Compute $(z_1, \dots, z_N) \leftarrow PreMul(w, N)$
6. Set $y = p_0 + \sum_{i=1}^{N} p_i z_i$
7. Set the output as xy and adjust it based on the range reduction information in step 1.

Complexity in Two-Party Setting:
XOR Gates: $O(Nl^2)$
Non-XOR Gates: $O(Nl^2)$

Complexity in Multi-Party Setting:
Rounds: $O(\log N)$
Interactive Operations : $O(Nk + l)$

Arctangent Protocol

- We use $h_N(x)$ to approximate arctangent function for some polynomial h of degree N over variable x.

Input: a
Output: $\text{Arctan}(a)$
Computation:

1. Apply a range reduction on a to compute x where $0 \le x \le 1$ and keep range reduction information
2. Lookup the minimum polynomial degree N which precision of approximation is at least k' bits
3. Lookup polynomial coefficients $p_0, \dots, p_N$ for arctangent approximation
4. Compute $(z_1, \dots, z_N) \leftarrow PreMul(x, N)$
5. Set $y = p_0 + \sum_{i=1}^{N} p_i z_i$
6. Set the output as y and adjust it based on the range reduction information in step 1.

Complexity in Two-Party Setting:
XOR Gates: $O(Nl^2)$
Non-XOR Gates: $O(Nl^2)$

Complexity in Multi-Party Setting:
Rounds: $O(\log N + \log l)$
Interactive Operations: $O(Nk + l \log l)$

This work was supported in part by grants 1223699, 1228639,1319090, and 1526631 from the National Science Foundation and FA9550-13-1-0066 from the Air Force Office of Scientific Research, as well as DARPA agreement no. AFRL FA8750-15-2-0092.