Implementation of Unconditionally Secure Multi-Party Computation with Guaranteed Output Delivery with Communication Complexity O(Cnϕ) bits
15-300, Fall 2020

Fan Pu Zeng
`https://fanpu.io/research`

December 18, 2020

1 Changes

The implementation of the multi-party computation (MPC) protocol given in the paper Guaranteed Output Delivery Comes Free in Honest Majority MPC by Vipul et al. [3] first requires a working implementation of a secure-with-abort protocol, which was not accounted for in the original project proposal. This can be achieved by implementing the currently best-known semi-honest protocol outlined in [2], which requires 5.5 field elements per multiplication gate per party. The secure-with-abort protocol itself then relies on an implementation of a Shamir secret sharing scheme, which fortunately already has an existing open-source Python implementation called `sss` (`https://github.com/dsprenkels/sss`).

2 Accomplishments

I have scrutinized the papers [1], [2], and [3] several times, with particular emphasis on [1] as it is considered the seminal paper in unconditionally secure multi-party computation.

3 Milestone Progress

I have gone through each of the papers several times and discussed various parts of the protocol in [3] with my collaborator Albert Gao.

4 Surprises

It is not entirely a surprise, but there is some additional math background that I need to attain in order to fully understand [1], in particular field theory and super linear algebra.

I also realised that it is quite difficult to break into the papers (in comparison to the papers I read for the critique, which was less mathematical in nature), especially given the other commitments from schoolwork throughout the semester, as it requires several days of focused reading to work through each of the algorithms and proofs fully.

5 Revisions

Given the scale of the project, it makes sense to split up the protocol into the following additional individual segments:

- **Dispute control.** This is a scheme to achieve unconditional security efficiently, as there can be dishonest parties participating in the protocol
- **Secret sharing.** This can be a wrapper around the open-source secret sharing protocol, but requires the addition of generating challenges with high min-entropy for all parties, and generating random double strings
- **Circuit evaluation.** After each round it is necessary to evaluate the circuit to verify the results of the computation. This comprises the following:
  - Evaluating a single multiplication gate
  - Evaluating a single segment
  - Checking the correctness of the `REFRESH` phase. This includes introducing randomness in the sharing of the `REFRESH` transcripts to preserve the privacy of participants
  - Checking the correctnes of `PartialMult`, to verify a batch of multiplications efficiently
- **Verifiability.** This is a protocol invoked by all parties in a pairwise-manner to determine if a share is correct or not. This helps to guard against collaborative corrupted parties.
  - Adding verifiability to sharings dealt by each party.
  - $\mathcal{F}_{Tag}$ protocol mentioned in [3]
  - Key distribution and maintenance
  - Generating authentication tags
  - Fault localization

  As such, I plan to make the following revisions to the milestones:

5.1 First Biweekly Milestone: February 15th

- Meet Hanjun Li together with collaborator Albert Gao, and get up to speed on the work done previously and the difficulties faced. Understand which protocols are working and well-tested, and which ones requires work.
- Acquire the math background required to understand [1] to an intuitive level

5.2 Second Biweekly Milestone: March 1st

- Make half-way progress on the secure-with-abort protocol
- Get familiar with the Shamir secret sharing (`sss`) library
- Peruse [3] with closer attention to details

5.3 Third Biweekly Milestone: March 15th

- Complete secure-with-abort protocol
- Write tests for the secure-with-abort protocol
- Begin implementation on the dispute control protocol

5.4 Fourth Biweekly Milestone: March 29th

- Complete dispute control protocol
- Write unit tests for dispute control protocol
- Begin work on secret sharing protocol

5.5 Fifth Biweekly Milestone: April 12th

- Complete secret sharing protocol
- Write unit tests for secret sharing protocol
- Begin work on verifiability protocols

5.6 Sixth Biweekly Milestone: April 26th

- Make half-way progress on verifiability protocol
- Begin work on drafting a paper based on the results of the implementation so far

5.7 Seventh Biweekly Milestone: May 10th

- Complete all portions of the verifiability protocols
- Complete first draft of paper

6 Resources Needed

I have all resources required to complete the project.

References

[1] Ivan Damgård and Jesper Buus Nielsen. Scalable and unconditionally secure multiparty computation. In Alfred Menezes, editor, *Advances in Cryptology - CRYPTO 2007*, pages 572–590, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[2] Vipul Goyal and Y. Song. Malicious security comes free in honest-majority mpc. *IACR Cryptol. ePrint Arch.*, 2020:134, 2020.

[3] Vipul Goyal, Yifan Song, and Chenzhi Zhu. Guaranteed output delivery comes free in honest majority mpc. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology – CRYPTO 2020*, pages 618–646, Cham, 2020. Springer International Publishing.