Paul M. Washburn
CSCI E-11 Unit 3: IoT
December 18, 2019

**In Depth Review: Secure Multi-Party Computation**

This Office Networking App will be privy to large amounts of data that, if leaked, might cause real damages to professionals like Dr. Brown that utilize it. However, there will be many administrators (such as employers, public & private) that will want to benefit from the collective knowledge that might be gained by gathering together all of the collective data generated using the application and analyzing it in various ways. We'll discuss secure multi-party computation and how it will be applied to the app to deliver more value without sacrificing security.

Classical cryptography addresses authenticity and privacy of communication, or messages sent in plain text. As the velocity of data increases in society, however, there is more demand for a new goal: to compute arbitrary functions using data owned by many users — without divulging un-owned data to any player. Often in these situations there will exist subsets of colluding players that are curious to learn about data that is owned by another party. These colluding players may be malicious or honest & curious; either way we will need a mechanism to keep data private. An obvious but perhaps naive approach might be to implement a "trusted center" solution to perform computation on the combined dataset, but this approach leaves us exposed to the potentiality that the center node might be faulty and act in malicious ways.

Here is where secure multi-party computation enters the scene. Secure multi-party computation enables us to compute via a decentralized protocol that emulate a "trusted center" solution while ensuring correctness, privacy and independence of inputs into the global function. A key tool to successfully implementing a secure multi-party computation scheme is secret sharing among all of the parties. In this protocol the secret *S* must have three properties: each party receives only a share of the secret *S*, no player can recover the secret on their own, and if all players act together then they will able to recover the secret.

Sum sharing is the mechanism that allows us to achieve a secret with these properties. Sum sharing includes choosing a prime number *P* that is greater than the secret *S*, then generating a vector of random numbers between 1 and the prime number *P* that is of length N-1 (where N is the number of parties). The final value in the vector will then be set to summation of this vector subtracted from the secret *S*, modulo the prime number *P*. Each player is then distributed their allocation of the random vector. The dealer then splits the secret *S* and distributes share of this vector. Since the all values in the random vector (except the final value) were chosen

independently of the secret $S$ they contain no information about $S$. Only when you have the last value can you extract $S$.

Via the Completeness Theorems we are ensured of the following. First, if there is an honest majority of players (less than half are faulty), then any multiparty function can be computed securely — even when there are colluding parties that are curious but non-deviating. Second, assuming an honest 2/3 majority, then any multi-party function can be securely computed even when colluding players maliciously deviate from the secure protocol. Finally, assuming an honest majority and an ability to broadcast, any multi-party function can be computed securely even if faulty players deviate. This provides *unconditional security* — meaning no complexity assumptions were made on the power of the adversary.

Secure multi-party computation techniques will be leveraged in this Office Networking App to distribute trust amongst different organizations that are utilizing the platform. Program administrators of this Office Networking App will be hesitant to share their data, despite the fact that sharing data across organizations will enable them to learn more about the workforce via the signal present in other organizations' data. By leveraging this protocol we will be able to perform arbitrary functions on the combined super-dataset while enjoying the unconditional security the protocol provides.