Chuong Ngo - Summary for Matthew Hammer’s Colloquium

WYSTERIA: A Programming Language for Generic Mixed-Mode Multiparty Computations

**Summary**

Secure multi-party computation (SMC) is a subfield of cryptography that allows multiple parties to jointly compute a public function over their private input data sets without revealing the private data sets. One example of a problem where a SMC function would be used is called the millionaire problem. First used by Andrew C. Yao, the millionaire problem involves two millionaires who want to see which one is richer. Typically, this would mean that the millionaires must reveal their net worths to each other or a common trusted medium who then makes the decision on who is wealthier. However, the millionaires don’t want to reveal their net worths to anyone. This is where SMCs come in.

Typically, SMCs are implemented using garbled circuits, homomorphic encryption, or cooperative computation among a group of servers. Current libraries or intermediate languages have also been created to help in making SMCs, but they are complex and error prone, which limits their use in realistic programs. Realistic programs would use mixed mode computations, where only some of the computations would be done in a *secure* mode, using SMCs, while the remaining computations will be done normally and not cooperatively. Mixed mode computation would allow for the distribution of the secure computations load to be spread across multiple nodes, increasing performance. Currently, mixed mode computations can be achieved by creating the application in two parts. The normal computations would be done in the typical way, being implemented in some sort of host language like C or Java. That application would then call a separate server application that implements the SMCs for the secure computations. This splitting of the program into two separate parts increases the complexity of the overall system, making them harder to create, error-prone, makes dynamic execution not viable, and makes it harder to reuse the SMC code.

Wysteria is a functional language that supports the writing of mixed-mode programs without using additional frameworks or languages. It allows for computations to operate in parallel and secure modes accross multiple nodes while allowing the program to conceptually view the program as a single thread, making it easier to write. Wysteria also supports the dynamic determination of multiple parties, dynamic circuit generation, and the representation of inputs and outputs as wire bundles. The language also supports secret shares and a refinement type system.

Wysteria was used to implement several n-party example programs in two sets of experiments to test the language’s performance. The tests show that the example programs implemented using Wysteria were quite fast and scaled very well as the number of participants increased. Additionally, mixed mode computations out performed monolithic implementations as the number of elements per principal increased. All in all, Wysteria looks to be a good language for mixed mode computation programs.

**Reference**

Rastogi, Aseem, Matthew A. Hammer, and Michael Hicks. "Wysteria: A programming language for generic, mixed-mode multiparty computations." *Security and Privacy (SP), 2014 IEEE Symposium on*. IEEE, 2014.