Secure Computation

Arpita Patra
Department of Computer Science and Automation
Indian Institute of Science, India
arpita@csa.iisc.ernet.in

1. Name of the Course. Secure Computation

2. Course Level and No. of Credits. 300 / 3:1

3. Instructor. Arpita Patra

5. Motivation and Objectives of the Course. The fantabulous journey of Secure Computation had originated with the seminal work of Andew Chi Chih Yao published in Foundation of Computer Science (FOCS) 1982. The idea of secure computation is so groundbreaking that Yao was bestowed with the prestigious Turing Award in 2000.

Many compelling applications involve computations that require sensitive data from two or more entities. Consider the following example. The Earth is orbited by nearly 7000 man-made satellites and more than 21000 orbital debris larger than 10 centimeters. The growing number of satellites and space debris orbiting the planet is increasing the danger of collisions. This is not a hypothetical scenario. There are many such reported collisions. Most recently in 2009, two communication satellites belonging to the US and Russia collided in orbit. Given how expensive the satellites are, in terms of replacing the satellite, the host countries want to avoid collision. A collision can only be predicted if the detailed orbit information of the satellites are known. However, the detailed location of each satellite can be a closely guarded secret data; it can even be a national secret. So what is needed is a way to determine whether two satellites are about to clash with each other based on the detailed locations of the satellites, but *without* the need of disclosing the locations of the satellites. The problem of *secure computation* models such applications that make simultaneous demands for the privacy and usability of sensitive data. More such real life applications include e-election, e-auction, secure signal-processing, secure bioinformatics, secure biometrics, secure machine-learning, secure outsourcing to name a few. Secure computation is also a tool to foil server security breach. To counter ubiquitous data security breaches and identity thefts, often the sensitive cryptographic key or the secret data is split into shares and stored in multiple servers so that the secret is protected from an adversary attacking a quorum of servers. The computation on the shared secret data without compromising its privacy is a serious challenge and can be tamed via secure computation protocols. The problem of secure computation abstracts out the aforementioned applications and alike, goes beyond the capabilities of conventional cryptography to offer the dual demands of privacy and computation on secret data as required. The problem is defined as follows: We have a set of n distrusting parties $\{P_1, \ldots, P_n\}$, each with its own private input $x_1, \ldots, x_n$. They want to compute some publicly known function f on their inputs without disclosing their inputs. The distrust among the parties is formalized by having an adversary that may corrupt some of the parties. Being such a powerful abstraction, the problem of secure computation is known as the "holy-grail" of cryptography. Yet, as far as my knowledge is concerned with, we are yet to see a formal course on this topic in India. So this course is first of its kind in India and promises to offer a comprehensive understanding on this topic. It will unfold the evolution of this topic since 1982 to till date and teach the fundamental and intricate results in this area. The results will include some of the groundbreaking

works done by secure computation exponents and Turing Award winners Shafi Goldwasser and Silvio Micali from MIT. It is my sincere hope that some of the students will aspire to become experts on this area after completing this course.

6. Syllabus.

- *Why Secure Computation?* Introduction, Motivation, History
- *Models of Secure Computation:* Honest vs. Dishonest majority settings, semi-honest vs active(malicious) adversary, static vs. adaptive computation, computational vs. information theoretic security, synchronous vs. asynchronous network.
- *Defining Secure Computation:* Computational/statistical indistinguishability, Real-Ideal World or Simulation-based Security notions.
- *Secure computation with semi-honest security:*
  - **Honest Majority Setting.** Secret Sharing, BenOr-Goldwasser-Wigderson (BGW) Construction, Optimizations (MPC in preprocessing mode and circuit randomization), Cramer-Damgaard-Neilsen (CDN) Construction.
  - **Dishonest majority Setting.** Impossibility of information-theoretic secure computation in dishonest majority setting, Oblivious Transfers (OT), two-party Goldreich-Micali-Wigderson (GMW) construction, Optimizations of GMW (Random input OT and OT extension), Yao construction, Optimizations of Yao (free XOR technique, point and permute technique), BMR construction and multi-party GMW construction.
- *Secure computation with Active security:*
  - **Honest Majority Setting.** Verifiable Secret Sharing, BGW Construction with active security, Hyper-invertible Matrices and Beerliova-Hirt (BH) Construction, Information Checking Protocol.
  - **Dishonest majority Setting.** Commitment Schemes, Zero-knowledge, GMW Compiler for active corruption, Cut-and-Choose OT and Lindell-Pinkas Construction.
- *Broadcast & Byzantine Agreement (BA):* Impossibility results. Dolev-Strong (DS) Broadcast, Exponential Information Gathering (EIG) construction for BA, Berman-Garay-Perry (BGP) construction for BA. Multi-valued Broadcast and BA.

7. References. There is no standard textbook to cover the entire syllabus. Many of the results discussed are very much state-of-the-art. So in most cases, the materials are taken from research papers. A part of the course may be covered from the following materials

1. *Book:* "Efficient Two-part Protocols- Techniques and Constructions" by Carmit Hazay and Yehuda Lindell.
2. *Book Draft:* "Secure Multiparty Computation and Secret Sharing - An Information Theoretic Appoach" by Ronald Cramer, Ivan Damgaard and Jesper Buus Nielsen

8. Prerequisites. My intention is to make the course as self-content as possible. However, those who have attended the basic level crypto course will be more comfortable than those who have not. Mathematical maturity will be assumed.