E0 337: Topics in Advanced Cryptography Sep 27 2023

Lecture 15

Instructor: Bhavana Kanukurthi *Scribe: Bhavish Raj Gopal*

1 Welcome to E0 337

1.1 Course Information

Contact information and office hours:

- **Lecturer:** Bhavana Kanukurthi (bhavana@iisc.ac.in).
- **Presenter:** Bhavish Raj Gopal (bhavishraj@iisc.ac.in).

1.2 Course Topics

The course will cover the following topics:

- Byzantine agreement
- Rational Byzantine agreement

1.3 Reference

Groce A, Katz J, Thiruvengadam A, Zikas V. *"Byzantine Agreement with a Rational Adversary"*. In Automata, Languages, and Programming: 39th International Colloquium, ICALP 2012

2 Introduction

2.1 Byzantine Agreement

The Byzantine agreement problem was proposed by Lamport, Peace and Shostak in 1982. It was motivated by the so-called Byzantine Generals' problem which is as follows. Consider a situation where the Byzantine generals have encircled an enemy city. Each general is far away from the rest and messengers are used for communication. The generals must agree upon a common plan (to attack or to retreat), though one or more of the generals may be traitors who will attempt to foil the plan. The good generals do not know who the traitors are. If all the generals agree on a plan then the plan will succeed. However, if there is a disagreement then the Byzantine army will lose. The traitors, however, may choose to coordinate in a manner that would mislead the good generals into disagreement. The Byzantine agreement problem when formally defined comes in two different flavours— Consensus and Broadcast. It finds use in applications such as Consensus in blockchain, fault tolerance in distributed systems etc. We next formally define Consensus and Broadcast.

Definition 1 (Consensus)**.** We refer to player P_1 as the sender, who is transmitting his input v_1 to the remaining $n-1$ receivers. A protocol is a perfectly secure broadcast protocol if it satisfies the following properties:

1. *(Consistency)*: All honest players output the same value w.
2. *(Correctness)*: If sender P_1 is honest, then $w = v$.

.

Definition 2 (Broadcast)**.** Each player P_i initially has input v_i. A protocol is a perfectly secure consensus protocol if it satisfies the following properties:

1. *(Consistency)*: All honest players output the same value w.
2. *(Correctness)*: If all honest players begin with the same input value, i.e. $v_i = v$ for all i, then $w = v$.

.

2.2 Results in traditional setting

Here we discuss some of the important results established for consensus and broadcast in the rational setting i.e when considering a malicious adversary. Here we assume that the parties are connected via a synchronous network and there is a centralized adversary controlling a subset of the parties.

1. To get a perfectly or statistically secure broadcast or consensus protocol we require $t < \frac{n}{3}$.
2. When we assume a setup (such as IT MACs), we can get a statistically secure consensus protocol for any $t < \frac{n}{2}$. The bound for broadcast is not known.
3. For $t < \frac{n}{2}$, broadcast $\iff$ consensus.
   (a) Broadcast $\implies$ consensus: Each party Pi broadcasts his and the Parties output the majority.
   (b) consensus $\implies$ broadcast: The sender sends his input to all parties. They then run a consensus protocol with the received value.

3 Rational Byzantine Agreement

RBA considers the problem of Byzantine agreement in the presence of a rational adversary instead of a malicious adversary. A rational adversary is characterized by some utility function that describes his preference over possible outcomes of the protocol execution. The utility should be natural, reasonable, and can be worked with easily. For simplicity, the paper only considers BA for single-bit values only. The adversary's utility is defined on the three possible events: (1) All honest players output (agree on) 1, (2) all honest players output (agree on) 0, and (3) honest players have disagreeing output. These utilities are denoted as $u_0, u_1, u_2 \in R$ respectively. The strategies and the protocol can be randomized, a particular set of strategies will imply not a particular outcome but a particular distribution over outcomes. The utility of a distribution is then the expected value of the utility of an outcome drawn from that distribution. If D is the resulting distribution of the outcome, then we use $U(D)$ to denote the expected utility.

We assume that all corrupted players are colluding. Equivalently, there is a single adversary that directs the actions of up to t (non-adaptively) corrupted players. The other players are honest, meaning that rather than acting according to their selfish interests they simply run the protocol as specified. Thus from a game theoretic perspective, the "game" we are considering actually only has one player.

Definition 3 (Perfect security)**.** A protocol for broadcast or consensus is perfectly secure against rational adversaries controlling t players with utility U if for every t-adversary there is a strategy S such that for any choice of input for honest players

1. *(S is tolerable)*: S induces a distribution of final outputs D in which no security condition is violated with nonzero probability.
2. *(S is Nash)*: For any strategy $S' \neq S$ with induced output distribution D':$U(D) \geq U(D')$.

.

Definition 4 (Statistical security)**.** A protocol for broadcast or consensus is statistically secure against rational adversaries controlling t players with utility U if for every t-adversary there is a strategy S such that for any choice of input for honest players

1. *(S is statistically tolerable)*: S induces a distribution of final outputs D in which no security condition is violated except with negligible probability.
2. *(S is statistically Nash)*: For any strategy $S' \neq S$ with induced output distribution D':$U(D)+neg(\lambda) > U(D')$.

.

A rational BA reduces to a traditional BA. The proof is based on the observation that if a protocol is secure according to the traditional definition of BA, then in RBA every adversarial strategy is Nash.

Theorem 5. *If protocol* Π *perfectly securely realizes traditional consensus (resp., broadcast) in the presence of a (non-rational) t-adversary, then* Π *is perfectly secure for consensus (resp., broadcast) against rational t-adversaries with utility U.*

We next study RBA in two different scenarios one with full knowledge of the adversary's preferences, the other where the adversary's preferences are not fully known.

4 Assuming Complete Knowledge

It is well-known that when $t \geq n/2$, it is impossible to get a consensus protocol even if there is a broadcast protocol. This is because consider the setting where the first $n/2$ of the parties have input 0, and the remaining have input 1. Assume the following adversarial scenarios: (A) the adversary corrupts the first $n/2$ or (B) the adversary corrupts the last $n/2$ parties; in both scenarios the adversary has the corrupted parties execute their correct protocol. In Scenario A, the honest parties should all output 1, whereas in Scenario B they should output 0. Consider now a third scenario (Scenario C) where the adversary does not corrupt any party. Because this Scenario is indistinguishable from Scenario B, the first half of the parties should output 0; however, because Scenario C is indistinguishable from Scenario A, the second half of the parties should output 1, which leads to a contradiction.

However, this is not true in the rational setting when the knowledge of the adversary's utility is known. Consider a rational adversary with utility $u_2 > u_1 > u_0$. Now consider a protocol that works as follows: (1) Every party P_i broadcasts his input v_i. (2) If all parties broadcast the same value then output it, otherwise output 0. Here, the adversary will never try to introduce an inconsistency, as if he does so he will be punished with his worst preferred outcome (i.e., 0). Thus we have the following theorem.

Theorem 6. *There exists a protocol for perfectly secure Byzantine agreement tolerating a rational t-adversary, where the utilities $u_0, u_1, u_2 \in R$ are known, tolerating arbitrarily many corruptions, i.e., $t < n$. The statement holds both for broadcast and consensus.*

5 Partial Knowledge

In this case, we assume that it is known whether or not the adversary wishes to create disagreement between the parties, but it is not known what the adversary's preferences are among different potential agreeing outputs. In this case, there are two possible scenarios (1) Disagreement is the Adversary's Most Favorable Option (2) Disagreement is the Adversary's Least Favorable Option.

Case 1: In this setting where disagreement is known to be the adversary's most preferred outcome, all the impossibility proofs from the traditional world apply. Therefore the bounds for both broadcast and consensus are the same as in the traditional setting.

Case 2: In the case where disagreement is the adversary's least-preferred outcome. Broadcast can be achieved, by the trivial multi-send protocol, tolerating an arbitrary number of corruptions. However the same is not true for consensus, this shows that in this setting perfectly secure broadcast is easier to achieve than consensus. This results in the following theorems.

Theorem 7. *' Assuming $n \geq 3$, there exists a perfectly secure rational consensus protocol tolerating any t-adversary with disagreement as the least-preferred outcome if and only if $t < \frac{n}{2}$. The statement holds also for statistical security.*

Theorem 8. *' If there exists a perfectly secure rational broadcast protocol tolerating any t-adversary for t ¡ n with disagreement as the least preferred outcome. The statement holds for statistical security as well.*

consider the following protocol: The sender sends his input to every party who outputs the value received from the sender. For ant $t < n$ there are two possible cases. In the first case, the sender is honest. As a result, all honest players are sent the correct output, and no error is made. In the second case, the sender is not honest. In this case, the adversary would not have the sender send disagreeing messages to honest parties, since disagreement is the least preferred outcome. However, because the sender is dishonest, any agreeing output from the honest parties is consistent with the security conditions, so no security violation can occur. Hence the protocol is perfectly secure against a rational adversary.