SaUCy Reference: Reliable Broadcast

1 Overview

Adaptively secure broadcast [2]. Bracha [1].

2 Ideal Functionalities

Functionality $\mathcal{F}_{\text{ACAST}}$

$\mathcal{F}^{t}_{\text{ACAST}}$ interacts with an adversary $\mathcal{S}$ and a set $\mathcal{P} = \{P_1, \ldots, P_N\}$ of parties.

1. Upon receiving (Bcast, sid, m) from P_D:
   - If P_D is honest, then, for each P_i in $\mathcal{P}$, send $(\text{Bcast}, sid, P_i, m)$ eventually.
   - If P_D is corrupted, then possibly, for each P_i in $\mathcal{P}$, send $(\text{Bcast}, sid, P_i, m)$ eventually.

ILC $\mathcal{F}_{\text{ACAST}}$

```
let F_acast = λ P .
  ...
```

3 Protocol Definition

Protocol Π_{Bracha}

Π_{Bracha} interacts with a set $\mathcal{P} = \{P_1, \ldots, P_n\}$ of parties and can tolerate up to f failures.

1. Upon receiving (Value, v) from P_i, send $(\text{Initial}, v)$ to all parties in $\mathcal{P}$.
2. Upon receiving an $(\text{Initial}, v)$ message or $\left\lceil \frac{n+f}{2} \right\rceil$ (Echo, v) messages or $\left\lceil \frac{f+1}{2} \right\rceil$ (Ready, v) messages, send (Echo, v) to all parties in $\mathcal{P}$.
3. Upon receiving $\left\lceil \frac{n+f}{2} \right\rceil$ (Echo, v) messages or $\left\lceil \frac{f+1}{2} \right\rceil$ (Ready, v) messages, send (Ready, v) to all parties in $\mathcal{P}$.
4. Upon receiving $\left\lceil \frac{f+1}{2} \right\rceil$ (Ready, v) messages, accept v.

4 Protocol Emulation

Theorem 1. *Protocol Π_{Bracha} t-securely realizes the functionality $\mathcal{F}_{\text{ACAST}}$ for $t < N/3$.*

Proof sketch. Let $\mathcal{A}$ be an adversary attacking Π_{Bracha}. We build a corresponding simulator $\mathcal{S}$ as follows.

Simulator $\mathcal{S}_{\text{ACAST}}$

□

References

1. Gabriel Bracha. Asynchronous byzantine agreement protocols. *Information and Computation*, 75(2):130–143, 1987.
2. Juan A Garay, Jonathan Katz, Ranjit Kumaresan, and Hong-Sheng Zhou. Adaptively secure broadcast, revisited. In *Proceedings of the 30th annual ACM SIGACT-SIGOPS symposium on Principles of distributed computing*, pages 179–186. ACM, 2011.