Scribes?

lecture recording

no class next week

---

**Broadcast definition**

An \( n \)-party protocol run by \( P_1, \ldots, P_n \) with a designated party \( P^* \in \{ P_1, \ldots, P_n \} \) is a broadcast protocol if:

- **Validity**: If \( P^* \) is honest, it has input \( m \), then all honest parties output \( m \).
- **Consistency**: All honest parties output the same value.

---

Securing realizing \( \overline{\text{FCast}} \) (with full security)

!!

broadcast

(assuming \( t < 2/3 \) corrupted parties)

---

Last time: \( t \leq \) setup, broadcast is impossible if \( t \geq n/3 \)

Today:

- broadcast is possible if \( t < n/3 \) (w/ setup)
- w/ setup, broadcast is possible for \( t < n \)
Protocol for BA tolerating $T < n/2$ corruptions:

- Construct a phase-King sub-protocol, with designated party called “King”.
- Overall protocol:
  - run phase-King sub-protocol $t+1$ times w/ $P, \ldots, P_{t+1}$ as the successive kings

Phase/King sub-protocol w/ King $P_k$:

Round 1) Every $P_i$ sends its input $v_i$ to everyone else.
   Each $P_i$ sets $C_{i}^{b} = 1$ iff $\geq n-t$ parties sent it the bit $b$.

Round 2) Each $P_i$ sends $C_{i}^{0}, C_{i}^{1}$ to everyone else.
   Each $P_i$ sets $D_{i}^{b} = \#$ of parties who sent $C_{i}^{b} = 1$.
   If $D_{i}^{b} > t$, set $v_i = 1$; else $v_i = 0$.

Round 3) $P_k$ sends $v_k$ to all parties.
   Each $P_i$ does: if $D_i^{v_i} < n-t$, then output $v_k$.
   Else output $v_i$.
   $\Rightarrow$ output $v$.

Lemma. If $T < n/2$ and all honest parties begin holding input $v$, then they all output $v$.

Lemma. If $T < n/3$ and the King is honest, then all honest parties agree on their output.

Proof. $P_k$ sends the same $v_k$ to everyone.

The only possible way agreement can fail is if some honest $P_i$ does not adopt the King's value.
I.e., if \( D^v_i \geq n-t \).

Claim: if \( D^v_i \geq n-t \), then \( v_i = v_k \)

**Case 1** \( v_i = 1 \). Because \( D^v_i \geq n-t \)

\[ \Rightarrow \text{any other honest party } p_j \text{ has } D^v_j \geq n-t-t \]

\[ \Rightarrow D^c_k > t \Rightarrow v_k = 1 \]

**Case 2** \( v_i = 0 \), \( D^v_i \geq n-t \Rightarrow D^0_k \geq n-2t > t \).

\[ \Rightarrow \text{at least one honest party sent } C^0 = 1 \text{ in round 1} \]

\[ \Rightarrow \text{at least one honest party received } 0 \text{ from } \]

\[ \geq n-t \text{ parties in round 1, } t \text{ received } 1 \text{ from } \]

\[ \leq t \text{ parties in round 1} \]

\[ \Rightarrow \text{every honest party received } 1 \text{ from } \leq 2t \text{ parties in round 1, } 2t < n-t \]

\[ \Rightarrow \text{every honest party sends } C^i = C \text{ in round i} \]

\[ \Rightarrow \text{every honest party has } D^i \leq t \]

\[ \Rightarrow v_k = 0 \]

\[ PKI = \text{public key infrastructure} \]

\[ \text{every party } p_i \text{ has } (sk_i, pk_i) \text{ for signature scheme} \]

\[ \text{every party has the same vector } (pk_1, pk_2, \ldots, pk_n) \]

**Dolev–Strong (retro)**

Assume \( p_i \) is the sender

\( (m, i) \)-valid message is received in round \( i \) if it has the form:

\[ m, (\sigma), a_2, \ldots, a_i \text{ by parties different from the receiver} \]
\( m \)-valid message \( \equiv \) \( (m, i) \)-valid for some \( i \);

**Round 1:** \( P_i \) signs \( m \) and sends \( m, s_i \) to everyone.

**Round 2, \ldots, n-1:** If \( P_i \) received an \( m \)-valid message in the previous round, it appends its signature and sends an \( m \)-valid message in the current round.

**Conclusion:** Let \( S_i \) denote the set of \( m \) for which \( P_i \) received an \( m \)-valid message.

- If \( |S_i| = 0 \) or \( |S_i| > 1 \), output 1.
- If \( |S_i| = 1 \), then output the value in \( S_i \).

Validity is immediate.

Consistency: Claim: All honest parties agree on the set of \( m \)-valid messages.

Efficiency?

The protocol as described is not necessarily efficient, but can be modified so that it is