Reliability of Distributed Systems

Ex3, due on January

version 2 (updated GABCA properties to mention grades)

Graded Asynchronous Binding Crusader Agreement

There are n parties and each party i has an input $x_i \in \{0, 1\}$, the goal of the protocol is to output a value $\in \{0, 1, \perp\}$ and a grade $\in \{2, 1, 0\}$. A protocol that solves **Graded Asynchronous Binding Crusader Agreement (GABCA)** that is resilient to $f < n/3$ Byzantine faults has the following properties:

1. Validity:

   1. if a non-faulty party outputs $x \neq \perp$, then some non-faulty party had x as input.

   2. If all non-faulty parties have the same input value then each non-faulty party outputs this value with grade 2.

   3. If a non-faulty party outputs a value with grade 2, then all non-faulty parties will output this value.

2. Termination: if all non-faulty parties start the protocol, then all non-faulty parties output a value and terminate. We will reach termination in a constant number of rounds.

3. Binding: when the first non-faulty party terminates, the adversary must commit to one of three events:

   A. No non-faulty will output 0

   B. No non-faulty will output 1

   C. all parties will output $\perp$.

   The important aspect of this property is that the adversary has to commit to b (or to all $\perp$) when the first non-faulty party completes the protocol (before seeing the random coin).

Weak coin

A weak coin with parameter α is a protocol where each party outputs a value $\in \{0, 1\}$ such that:

1. Agreement: for each $b \in \{0, 1\}$, with probability at least α all non-faulty parties output b.

2. Unpredictability: if no non-faulty has started the protocol, the the adversary cannot predict if the coin will reach agreement and to which value

Question set 1

1. Show that the trivial protocol where each party chooses a coin uniformly at random is a weak coin protocol.

2. What is the parameter α as a function of n and f.

Binary Asynchronous Byzantine Agreement

There are n parties and each party i has an input $x_i \in \{0, 1\}$, the goal of the protocol is to output a value $\in \{0, 1\}$. A protocol that solves **Binary Asynchronous Byzantine Agreement (BABA)** that is resilient to $f < n/3$ Byzantine faults has the following properties:

1. Validity: If all non-faulty parties have the same input value then each non-faulty party outputs this value.
2. Agreement: All non-faulty parties output the same value.
3. Termination: if all non-faulty parties start the protocol, then all non-faulty parties output a value and terminate.

Question 2

1. Provide a protocol for solving Binary Asynchronous Byzantine Agreement using a weak coin protocol and a GABCA protocol.
2. Prove all the properties. Hint: your proof should be using all the properties of all the building blocks.

The following code can provide hints but may be incomplete and incorrect.
Party i with input x_{k-1} for round k:

1. $(x_k, g_k) = GABCA(x_{k-1}, k)$
2. $coin = WeakCoin(k)$
3. If $g_k = 2$ then decide x_k and send $< decide, x_k >$ to all
4. If $x_k = \perp$ then $x_k = coin$
5. If you hear $n - 2f$ decide then decide and send decide
6. If you hear $n - f$ decide then terminate
7. $k + +$; goto 1.

One idea: 1. if the adversary chooses to bind to b and the coin equals b for all then the next round all decide due to validity. 2. if the adversary chooses all $\perp$ and the next coin equals b for all then next round all decide due to validity.

GABCA protocol

The following sketch for **GABCA** can provide hints but may be incomplete or incorrect:

1. Send: send $<$ val, $x_i >$ to all parties.
2. Echo1: (at most two values)
   1. If you hear $<$ val, $x >$ from $f + 1$ parties and you did not send $<$ echo1, $x >$ yet then send $<$ echo1, $x >$ to all parties.
3. Echo2: (at most one value and one $\perp$)
   1. *First time:* if you hear $<$ echo1, $x >$ from $n - f$ parties and you did not send any $<$ echo2, $\star >$, then send $<$ echo2, $x >$ to all parties.
   2. *Second time*: if you hear $<$ echo1, $x >$ from $n - f$ parties and you already sent exactly one $<$ echo2, $y >$ with $y \neq x$, then send $<$ echo2, $\perp >$ to all parties.

4. Echo3: (at most one value)
   1. if you hear $< \text{echo2}, x >$ from $n - f$ parties and you did not send any $< \text{echo3}, \star >$ yet, then send $< \text{echo3}, x >$ to all parties.
5. Echo4: wait for $n - f$ echo3 messages, then wait for either:
   1. $< \text{echo3}, x >$ from $n - f$ parties, then send echo4 x.
   2. $< \text{echo2}, \perp >$ from $n - f$ parties, then send echo2 $\perp$.
6. Echo5:
   1. if you receive $n - t < \text{echo2}, \perp >$, then send $< \text{echo5}, \perp >$
   2. if you receive $n - t < \text{echo4}, v >$ then send $< \text{echo5}, v >$
7. Output: wait for $n - f < \text{echo5}, v >$ messages, then wait for:
   1. $n - f < \text{echo5}, v >$ then output $(v, 2)$
   2. at least one $< \text{echo5}, v >$ then output $(v, 1)$
   3. $n - t < \text{echo5}, bot >$ then output $(\perp, 0)$

question set 3

1. provide a full protocol for solving GABCA.
2. Prove all the properties. Hint, try to first understand what each round provides.
3. How many bits does each party send in the worst case?
4. (bonus) Can you reduce the number of rounds to less than 6 and still keep a total of $O(n^2)$ bits? Make sure the binding property holds.