Assignment1

LI Yang(ylikp@connect.ust.hk) 20750699

Problem 1

Denition 1 (Dolev-Strong Protocol). Given a network with at most *f* corrupt nodes out of a total of n nodes and an input message m,

1. At round 0: Initialize $S_i = \varnothing$ for each node i. Leader sends m to all nodes with its signature.
2. For each round r = 1 to f + 1: If a node i receives an unseen message m signed by r signatures, it (i) updates $S_i = S_i \cup \{m\}$ and (ii) sends m to all other nodes by adding its signature.
3. At round f + 2: Node i outputs S_i.

Question. Consider that honest nodes follow the protocol and corrupt nodes behave arbitrarily. Prove the consistency of the Dolev-Strong protocol, i.e., outputs S_i and S_j satisfying $S_i = S_j$ for any two honest nodes i and j. (Note: leader may be corrupt.)

Solution:

Honest nodes: 1, 2, …, n-f; Malicious nodes: 1', 2', …, f'

Case1: If the leader is honest

In round 0, the leader(node 1) will send the message with his signature: $\{m(sig_1)\}$, to every other node.

In round r = 1 ~ f + 1, all honest nodes will receive and accept $\{m(sig_1, sig_2,\ldots, sig_r)\}$ and keep their set $S_i = \{m\}$. While other malicious messages will not be accepted because they are not signed by the leader.

So all the honest nodes will have the same ouput S finally.

Case2: If the leader is corrupt

In round 0, the leader(node 1') will send the message with his signature: $\{m(sig_{1'})\}$, to every other node.

In round r = f + 1 (final round), if node i (honest) adds the wrong message m' to its S, then the message b' must contain at least f+1 distinct signatures including the leader's one. However, there are only f corrupt nodes to generate f signatures on the wrong message. So there must be another honest node (p) to sign on the wrong message m'. It means that the honest node p has attached its signature to m' in some early round r < f + 1.

But we have known that if in the round r < f+1 an honest node accepted and signed on a message, all other nodes will receive this message in the next round (r + 1) and honest nodes will accept this message and update their S.

Combining these two points we can have the conclusion that 1) corrupt nodes decide not to attack => all honest nodes will have the same message m, 2) corrupt nodes try to attack => all honest nodes will have the same message m'. So with f + 1 rounds, all honest nodes will have the same output S (consistency).

Problem 2

Definition 2 (Streamlet Protocol). The Streamlet protocol works as follows:

- **Propose-Vote.** In every epoch:
  - The epoch's designated leader proposes a new block extending from the longest notarized chain it has seen (if there are multiple, break ties arbitrarily). The notion “notarized” is defined below.
  - Every player votes for the first proposal they see from the epoch's leader, as long as the proposed block extends from (one of) the longest notarized chain(s) that the voter has seen. A vote is a signature on the proposed block.
  - When a block gains votes from at least 2n/3 distinct players, it becomes notarized. A chain is notarized if its constituent blocks are all notarized.
- **Finalize.** Notarized does not mean final. If in any notarized chain, there are three adjacent blocks with consecutive epoch numbers, the prefix of the chain up to the second of the three blocks is considered final. When a block becomes final, all of its prex must be final too.

Question. If we finalize the chain when there are two adjacent blocks with consecutive epoch numbers, is consistency still achieved? If yes, please prove it. Otherwise, please show a counter example.

Solution:

Consider three latest blocks K, N, N+1. Assume < n/3 corrupt nodes.

Chain: 0 <= … <= K <= N <= N+1

<= S <= none (a potential fork)(m may be 1 or not)

If we finalize the chain when there are 2 adjacent blocks with consecutive epoch numbers it means that when N+1 is notarized, N will be finalized. We only need to make sure that a possible fork S from K will not have its later blocks.

Over 2n/3 nodes have voted for block N+1 and that means > n/3 honest nodes voted for block N+1. Those honest nodes who voted for N+1 must have seen the previous consecutive notarized block N (otherwise they will not agree on the previous hash of N+1).

If S > N+1, then those honest nodes who voted for block N+1 (>n/3) will not vote for S so S can not be notarized. There will not exist a later fork from K.

If S < N, similarly those honest nodes who voted for N+1 (>n/3) will not vote for any block that follows S. So the fork will stop at block S.

Finally we can finalize the block N once N+1 is notarized. <=> Achieve consistency with two adjacent blocks with consecutive epoch numbers.