Different cases:

1. Clients may collude with each other to learn the honest party’s set elements (under threshold assumption)
2. Client B may refuse paying his share, but it does not collude with any other clients.
3. A subset of clients may collude with client B and help him not paying his share. In this case, the colluding clients may get some reward, more than their real shares, from client B.
4. Clients may want to misbehave to earn more than they deserve, i.e. their share, but without colluding with client B.

**Remark**: for now, we do not consider the case where a client wants to ruin the result so nobody, including itself, would get paid. Therefore, we consider economically rational adversaries.

*Case 1* is easy to address, as client B sends the money directly to clients, so no smart contract (or blockchain) is needed.

*Case 2* is easy to address under the assumption that the other clients are semi-honest- To address the requirement of this case, client B sends a deposit to a smart contract. Then, when the result is computed one of the clients just ask the contract to send their shares.

Note that the current version of the protocol captures case1, 2, and 4. Note that in case 3, client B cannot be sure that the elements provided by a colluding client is indeed in the intersection.