3. $f_j(i), f'_j(i)$ for $j \in H_0, i \in B$
4. C_{jm} for $j \in H_0, m \in \{0, ..., t-1\}$
5. A_{jm} for $j \in H_0, m \in \{0, ..., t-1\}$
6. $(f_i(j), f'_i(j))$ for $j \in \{1, ..., n\}, i \in B'$

Now we show how to construct a simulator *SIM* that can act in the protocol as the real players, such that the resulting view has the same probability distribution (we use the same simulator as in [11]). Note that *SIM* does not have to compute the sharing polynomials (2) itself since they are chosen by the adversary. The same holds for the content of the random tape (1) which is part of the adversary's internal state that does not have to be simulated.

1. (3, 4) Perform step 1 of the protocol on behalf of the uncorrupted players $P_t, ..., P_n$ exactly as specified in the protocol. This includes receiving and processing the information sent privately and publicly from corrupted players to honest ones.After this step, *SIM* knows all polynomials $f_i(\cdot), f'_i(\cdot)$ for $i \in H_0$ (this holds also for $i \in H_0 \cap B$, since *SIM* received enough consistent shares from these parties to compute their polynomials). In particular, *SIM* knows all the shares $f_i(j), f'_i(j)$, the coefficients a_{ik}, b_{ik} and the public values C_{ik}.
2. (5) When extracting the values $r_i G$, the simulator acts as follows:
   - Compute $A_{ik} = a_{ik} G$ for $i \in H_0 \setminus \{n\}, k \in \{0, ..., t-1\}$
   - Compute $A_{n0} = Y - \sum_{i \in H \setminus \{n\}} A_{i0}$
   - Compute $A_{nk} = \lambda_{k0} A_{n0} + \sum_{i=1}^{t-1} \lambda_{ki} f_n(i) G$ for $k \in \{1, ..., t-1\}$, where λ_{ki}'s are the Lagrange interpolation coefficients of the set H_0.
   - Broadcast A_{ik} for $i \in H_0, k \in \{0, ..., t-1\}$
3. (6) To handle the messages resulting from complaints, *SIM* acts as follows:
   - Perform for each uncorrupted player the verifications of Eq. (2) on the values A_{ik} for $i \in B$, broadcast by the players controlled by the adversary. If the verification fails for some $i \in B, j \in H_0 \setminus B$, broadcast a complaint $(f_i(j), f'_i(j))$. (Notice that the corrupted players can publish a valid complaint only against one another, and there will be no complaints against an honest player that is simulated by *SIM*).
   - For each valid complaint against P_i, perform the reconstruction phase of Pedersen's VSS to compute r_i and Y_i in the clear.

After step 1, the polynomials $f_i(\cdot), f'_i(\cdot)$ for $i \in H_0 \setminus B$ are chosen at random. All associated values $(C_{ik}, f_i(j), f'_i(j), a_{ik}, b_{ik})$ therefore have the exact same probability distribution as in a real run of the protocol.

The broadcasted values A_{ik} are all uniformly random since the corresponding a_{ik} are random. This holds also for the specially computed A_{nk} for $k \in \{0, ..., t-1\}$, since, for each such coefficient, there is at least one random value it depends on. Notice that the fact that these A_{nk}'s are not consistent with the corresponding a_{nk}'s does not appear in the adversary's view: he never sees the a_{nk}'s but only the consistent public commitments of these values.