A PROOF OF THEOREM 1

Proof: We use a standard hybrid argument to prove our **Theorem 1**. We first define a simulator *SIM* through a series of subsequent modifications to the random variable *REAL*, so that any two subsequent random variables are computationally indistinguishable.

Hyb_0: The *SIM* random variable is distributed exactly as *REAL*, i.e., the joint view of the parties U in a real execution of the protocol.

Hyb_1: In this hybrid, the simulator changes the behavior of all honest clients belonging to $C_2\backslash U$. Specifically, for each client n, a uniformly random number $v_{n,m}$ is selected to replace the shared key $\mathrm{KA.Agree}\left(P_n^{SK}, P_m^{PK}\right)$ between client n and client m in the same set, and to perform the function of encryption and decryption. The DDH assumption [9] ensures that this hybrid possesses the indistinguishability from real protocol.

Hyb_2: In this hybrid, the simulator replaces all encrypted data (i.e., the encrypted SK and encrypted shares of b_n and N_n^{SK}) sent by honest clients (in the set $C_2\backslash U$) to other clients with encrypted random values (e.g., 0, with appropriate length). However, all honest clients continue to respond using the correct SK_n to encrypt the correct shares **in Round 3**. Since the simulator just changes the content of ciphertext, IND-CPA security of the Symmetric Encryption (SE) scheme [8], [4] guarantees the indistinguishability between this hybrid with the previous one.

Hyb_3: Define:

$$C^* = \begin{cases} C_2\backslash U, & \text{if } z = \perp \\ C_2\backslash C_3\backslash U, & \text{otherwise} \end{cases}$$

Then, in the **Round 1**, for all honest clients in the set C^*, the simulator replaces all the shares of b_n with random values (e.g., 0, with appropriate length). It is obvious that the adversary cannot get extra share of b_n, either because the honest clients do not reveal their shares of b_n (resp. $C_3 \geq t, C^* = C_2\backslash C_3\backslash U$), or because all the honest clients are offline (resp. $C_3 < t, C^* = C_2\backslash U$, where $z = \perp$). The security of Shamir's secret sharing scheme guarantees that it is infeasible to recover the secret even possessing any $t-1$ shares of current secret, which means that even the honest but curious clients have $|U| < t$ shares of b_n, they still cannot tell whether the shares submitted from honest clients come from the real b_n or not.

Hyb_4: In this hybrid, instead of generating $\mathrm{PRG}(b_n)$ for all clients in the set C^*, the simulator uses uniformly random number r_n with appropriate size to replace it. It is easy to understand that the simulator just substitutes the output of Pseudorandom Generator (PRG) [7]. Therefore, the security of PRG [8] ensures that this hybrid is indistinguishable from the previous one.

Hyb_5: In this hybrid, for each client n in the set C^*, the simulator generates the masked input as below:

$$g_n^- = r_n + \sum_{m\in C_2:n<m} \mathrm{PRG}\left(s_{n,m}\right) - \sum_{m\in C_2:n>m} \mathrm{PRG}\left(s_{n,m}\right)$$

instead of utilizing

$$g_n^- = g_n + \mathbf{PRG}\left(b_n\right) + \sum_{m\in C_2:n<m} \mathbf{PRG}\left(s_{n,m}\right) - \sum_{m\in C_2:n>m} \mathbf{PRG}\left(s_{n,m}\right)$$

Since $\mathrm{PRG}(b_n)$ has been changed in the previous hybrid with a uniformly random number, we know that $g_n + r_n$ is also a random value, and it is easy to deduce that the distribution of r_n and g_n+$\mathrm{PRG}(b_n)$ is indistinguishable.

Note that, if $z = \perp$, the simulator has already completed the simulation (describe as Hyb_5) since *SIM* successfully simulates *REAL* without knowing g_n for all clients $n \notin U$. Therefore in the following hybrids we assume $z \neq \perp$.

Hyb_6: In this hybrid, for every client $u^- \in C_3\backslash U$, the simulator substitutes the shares of N_n^{SK} with shares of random values (e.g., 0, with appropriate length). Similar to Hyb_3, the security of Shamir's secret sharing protocol guarantees that this hybrid is indistinguishable from the previous one.

Hyb_7: In this hybrid, given a client $u^- \in C_3\backslash U$, for all other clients $n \in C_3\backslash U$, the simulator uniformly selects a random number to replace the sharked key (i.e., $s_{u^-,n} = s_{n,u^-} \leftarrow \mathrm{KA.Agree}$ $(N_{u^-}^{SK}, N_n^{PK})$) between client n and u^-, and this random number will be used as the seed of **PRG** for both client n and u^-. Specifically, a random value s_{n,u^-}^- is selected for each client $n \in C_3\backslash U\backslash\{u^-\}$. Instead of sending

$$g_n^- = g_n + \mathbf{PRG}\left(b_n\right) + \sum_{m\in C_2:n<m} \mathbf{PRG}\left(s_{n,m}\right) - \sum_{m\in C_2:n>m} \mathbf{PRG}\left(s_{n,m}\right)$$

the simulator submits

$$g_n^- = g_n + r_n + \sum_{m\in C_2\backslash\{u^-\}:n<m} \mathbf{PRG}\left(s_{n,m}\right) - \sum_{m\in C_2\backslash\{u^-\}:n>m} \mathbf{PRG}\left(s_{n,m}\right) + \Delta_{n,u^-} \cdot \mathbf{PRG}(s_{n,u^-}^-)$$

where $\Delta_(n,u^-) = 1$ if $n < u^-$. Otherwise, $\Delta_(n,u^-) = -1$. Correspondingly, we have

$$g_{u^-}^- = g_{u^-} + r_{u^-} + \sum_{n\in C_2} \Delta_{u^-,n} \cdot \mathbf{PRG}\left(s_{n,m}\right)$$

It is easy to see that the DDH assumption [9] guarantees that this hybrid is indistinguishable from the previous one.

Hyb_8: In this hybrid, for the same client u^- selected in the previous hybrid and all other clients $n \in C_3\backslash U$, the simulator also uniformly selects a random number r_{n,u^-} to replace the computation of $\mathrm{PRG}(s_{n,u^-}^-)$. Similar to Hyb_4, it is easy to understand that the simulator just substitutes the output of PRG. Hence, the security of PRG [8] ensures this hybrid is indistinguishable from the previous one.

Hyb_9: In this hybrid, for each client n in the set $C_3\backslash U$, the simulator submits

$$x_n^- = x_n + \mathbf{PRG}\left(b_n\right) + \sum_{m\in C_2\backslash C_3\backslash U} \Delta_{n,m} \cdot \mathbf{PRG}\left(s_{n,m}\right)$$

instead of sending

$$g_n^- = g_n + \mathbf{PRG}\left(b_n\right) + \sum_{m\in C_2:n<m} \mathbf{PRG}\left(s_{n,m}\right) - \sum_{m\in C_2:n>m} \mathbf{PRG}\left(s_{n,m}\right)$$

where $\{x_n\}_{n\in C_3\backslash U}$ are uniformly random and subjected to $\sum_{C_3\backslash U} x_n = \sum_{C_3\backslash U} g_n = z$. Moreover, the simulator submits signature of masked input $\sigma_n \leftarrow SIG.Sign\left(K_n^{SK}, x_n^-\right)$ instead of $\sigma_n \leftarrow SIG.Sign\left(K_n^{SK}, g_n^-\right)$. As a result, each client in $C_3\backslash U$ can pass the verification. Therefore, the simulator has already completed the simulation since SIM successfully simulates REAL without knowing g_n for all clients $n \in C_3\backslash U$. Based on the hybrid 0 to 9, we can infer that the distribution of this hybrid is identical to the previous one, completing the proof.