

# COMMUNICATION-COMPUTATION EFFICIENT SECURE AGGREGATION FOR FEDERATED LEARNING (SUPPLEMENTARY MATERIALS)

**Anonymous authors**

Paper under double-blind review

## A DETAILED DESCRIPTION OF THE CCESA PROTOCOL

---

### Algorithm 1: Communication-Computation Efficient Secure Aggregation (CCESA) Protocol

---

**Input:** Number of clients  $n$ , assignment graph  $G$ , privacy thresholds  $t_i$  of all clients  $i \in [n]$ , local models  $\theta_i$  of all clients  $i \in [n]$ , Diffie-Hellman key pairs  $(c_i^{PK}, c_i^{SK})$ ,  $(s_i^{PK}, s_i^{SK})$  of all clients  $i \in [n]$  and corresponding key agreement function  $f$ , pseudo-random generator **PRG**

#### Step 0. Advertise Keys

**Client  $i$ :**

Sends  $(i, c_i^{PK}, s_i^{PK})$  to the server

**Server:**

Collects the messages from clients (denote this set of clients as  $V_1$ )

Sends  $\{(i, c_i^{PK}, s_i^{PK})\}_{i \in Adj(j) \cap V_1}$  to all clients  $j \in V_1$ ;

#### Step 1. Share Keys

**Client  $i$ :**

Generates a random element  $b_i$

Applies  $t_i$ -out-of- $(|Adj(i)| + 1)$  secret sharing schemes to  $b_i$  and  $s_i^{SK}$

$$b_i \xrightarrow{(t_i, |Adj(i)|+1)} (b_{i,j})_{j \in (Adj(i) \cup \{i\})}, \quad s_i^{SK} \xrightarrow{(t_i, |Adj(i)|+1)} (s_{i,j}^{SK})_{j \in Adj(i) \cup \{i\}}$$

Encrypts  $[b_{i,j}, s_{i,j}^{SK}]$  to  $[\bar{b}_{i,j}, \bar{s}_{i,j}^{SK}]$  using the authenticated encryption with key  $f(c_j^{PK}, c_i^{SK})$

Sends  $\{(i, j, \bar{b}_{i,j}, \bar{s}_{i,j}^{SK})\}_{j \in Adj(i) \cap V_1}$  to the server

**Server:**

Collects the messages from clients (denote this set of clients as  $V_2$ )

Sends  $\{(i, j, \bar{b}_{i,j}, \bar{s}_{i,j}^{SK})\}_{i \in Adj(j) \cap V_2}$  to all clients  $j \in V_2$

#### Step 2. Masked Input Collection

**Client  $i$ :**

Computes  $s_{i,j} = f(s_j^{PK}, s_i^{SK})$  and

$$\tilde{\theta}_i = \theta_i + \text{PRG}(b_i) + \sum_{j \in V_2 \cap Adj(i); i < j} \text{PRG}(s_{i,j}) - \sum_{j \in V_2 \cap Adj(i); i > j} \text{PRG}(s_{i,j})$$

Sends  $(i, \tilde{\theta}_i)$  to the server

**Server:**

Collects the messages from clients (denote this set of clients as  $V_3$ )

Sends  $V_3$  to all clients  $j$  in  $V_3$

#### Step 3. Unmasking

**Client  $i$ :**

Decrypts  $\bar{b}_{i,j}$  with key  $f(c_j^{PK}, c_i^{SK})$  to obtain  $b_{i,j}$  for all  $j \in Adj(i) \cap V_3$

Decrypts  $\bar{s}_{i,j}^{SK}$  with key  $f(c_j^{PK}, c_i^{SK})$  to obtain  $s_{i,j}^{SK}$  for all  $j \in Adj(i) \cap (V_2 \setminus V_3)$

Sends  $\{b_{i,j}\}_{j \in Adj(i) \cap V_3}, \{s_{i,j}^{SK}\}_{j \in Adj(i) \cap (V_2 \setminus V_3)}$  to the server

**Server:**

Collects the messages from clients

Reconstructs  $b_i$  from  $\{b_{i,j}\}_{j \in Adj(i) \cap V_3}$  for all  $i \in V_3$

Reconstructs  $s_i^{SK}$  from  $\{s_{i,j}^{SK}\}_{j \in Adj(i) \cap (V_2 \setminus V_3)}$  for all  $i \in V_2 \setminus V_3$

Computes  $s_{i,j} = f(s_j^{PK}, s_i^{SK})$  for all  $j \in Adj(i) \cap V_3$

Computes the aggregated sum of local models

$$\sum_{i \in V_3} \theta_i = \sum_{i \in V_3} \tilde{\theta}_i - \sum_{i \in V_3} \text{PRG}(b_i) - \sum_{i \in V_2 \setminus V_3, j \in Adj(i) \cap V_3; i > j} \text{PRG}(s_{i,j}) \\ + \sum_{i \in V_2 \setminus V_3, j \in Adj(i) \cap V_3; i < j} \text{PRG}(s_{i,j})$$


---

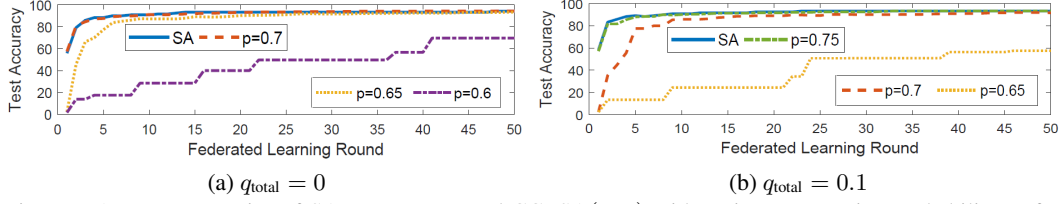


Figure B.1: Test accuracies of SA versus proposed CCESA( $n, p$ ) with various connection probability  $p$ , for federated learning using the AT&T face dataset. Here, we set  $n = 40$  and  $t = 21$ . The suggested CCESA achieves the ideal test accuracy by using only 70% of the communication/computational resources used in the conventional SA.

Schemes \ Number of training data ( $n_{\text{train}}$ )	5000	10000	15000	50000
<b>Federated Averaging</b> (McMahan et al., 2017)	70.41%	65.82%	65.89%	60.62%
<b>Secure Aggregation (SA)</b> (Bonawitz et al., 2017)	49.78%	49.97%	49.91%	49.10%
<b>CCESA (Suggested)</b>	49.48%	50.07%	49.16%	50.00%

Table B.1: Precision of the membership inference attack on local models trained on CIFAR-10. The scheme with a higher attack precision is more vulnerable to the inference attack. For the proposed CCESA, the attacker is no better than the random guess with precision = 50%, showing the privacy-preserving ability of CCESA.

## B ADDITIONAL EXPERIMENTAL RESULTS

### B.1 RELIABILITY

In Fig. 4 of the main paper, we provided the experimental results on the reliability of CCESA on CIFAR-10 dataset. Similarly, Fig. B.1 shows the reliability of CCESA in AT&T Face dataset, where the model is trained over  $n = 40$  clients. We plotted the test accuracies of SA and the suggested CCESA( $n, p$ ) for various  $p$ . In both settings of  $q_{\text{total}}$ , selecting  $p = 0.7$  is sufficient to achieve the test accuracy performance of SA when the system is trained for 50 rounds. Thus, the required communication/computational resources for guaranteeing the reliability, which is proportional to  $np$ , can be reduced to 70% of the conventional wisdom in federated learning.

### B.2 PRIVACY

In Section 5.3 and Fig. 2 of the main paper, we provided the experimental results on the AT&T Face dataset, under the model inversion attack. In Fig. B.2, we provide additional experimental results on the same dataset for different participants. Similar to the result in Fig. 2, the model inversion attack successfully reveals the individuals identity in federated averaging (McMahan et al., 2017), while the privacy attack is not effective in both SA and the suggested CCESA.

In the main manuscript, we have also considered another type of privacy threat called membership inference attack (Shokri et al., 2017), where the attacker observes masked local model  $\tilde{\theta}_i$  sent from client  $i$  to the server, and guesses whether a particular data is the member of the training set. We measured three types of performance metrics of the attacker: *accuracy* (the fraction of the records correctly estimated the membership), *precision* (the fraction of the responses inferred as members of the training dataset that are indeed members) and *recall* (the fraction of the training data that the attacker can correctly infer as members). Table 3 in the main manuscript summarizes the attack accuracy result, while Table B.1 shows the attack precision for CIFAR-10 dataset. We also observed that recall is close to 1 for all schemes. Similar to the results on the attack accuracy, Table B.1 shows that the attack precision of federated averaging reaches near 70%, while that of SA and CCESA remain around the baseline performance of random guess. This shows that both SA and CCESA do not reveal any clue on the training set.

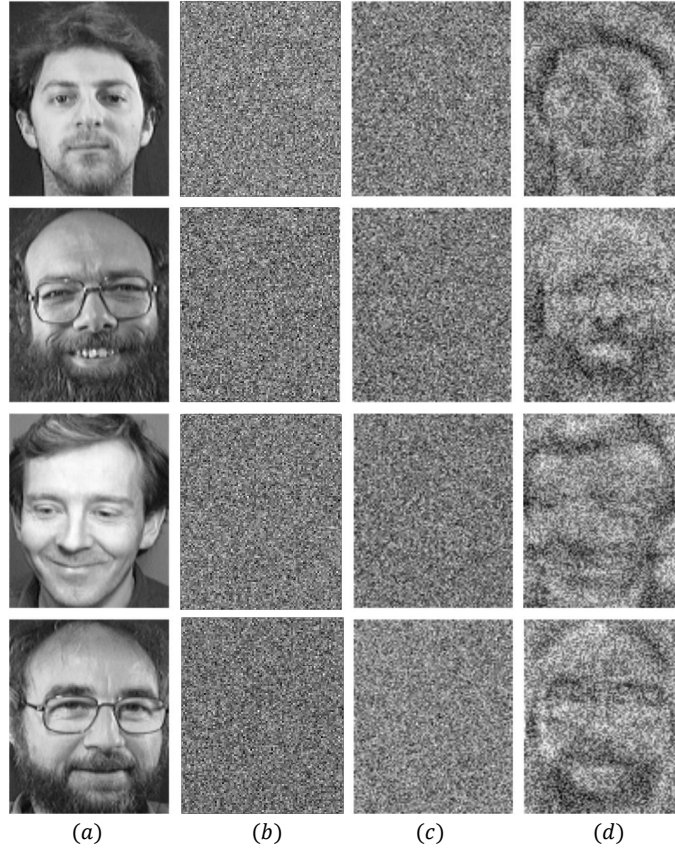


Figure B.2: The result of model inversion attack to three schemes, (b) the suggested scheme (CCESA), (c) SA (Bonawitz et al., 2017) and (d) federated averaging (McMahan et al., 2017), for AT&T Face dataset. The original training images at (a) can be successfully reconstructed by the attack only in federated averaging setup, i.e., both SA and CCESA achieve the same level of privacy.

## C PROOFS

### C.1 PROOF OF THEOREM 1

*Proof.* Note that the sum of masked local models obtained by the server is expressed as

$$\sum_{i \in V_3} \tilde{\theta}_i = \sum_{i \in V_3} \theta_i + \sum_{i \in V_3} \mathbf{PRG}(b_i) + \mathbf{z}$$

where

$$\mathbf{z} = \sum_{i \in V_3} \sum_{j \in V_2 \cap \text{Adj}(i); i < j} \mathbf{PRG}(s_{i,j}) - \sum_{i \in V_3} \sum_{j \in V_2 \cap \text{Adj}(i); i > j} \mathbf{PRG}(s_{i,j}).$$

Here,  $\mathbf{z}$  can be rewritten as

$$\begin{aligned} \mathbf{z} &= \sum_{i \in V_3} \sum_{j \in V_2 \cap \text{Adj}(i); i < j} \mathbf{PRG}(s_{i,j}) - \sum_{i \in V_3} \sum_{j \in V_2 \cap \text{Adj}(i); i > j} \mathbf{PRG}(s_{i,j}) \\ &= \sum_{i \in V_3} \sum_{j \in V_2 \cap \text{Adj}(i); i < j} \mathbf{PRG}(s_{i,j}) - \sum_{j \in V_2} \sum_{i \in V_3 \cap \text{Adj}(j); i > j} \mathbf{PRG}(s_{i,j}) \\ &\stackrel{(a)}{=} \sum_{i \in V_3} \sum_{j \in V_2 \cap \text{Adj}(i); i < j} \mathbf{PRG}(s_{i,j}) - \sum_{i \in V_2} \sum_{j \in V_3 \cap \text{Adj}(i); i < j} \mathbf{PRG}(s_{i,j}) \\ &= \sum_{i \in V_3} \sum_{j \in V_3 \cap \text{Adj}(i); i < j} \mathbf{PRG}(s_{i,j}) + \sum_{i \in V_3} \sum_{j \in (V_2 \setminus V_3) \cap \text{Adj}(i); i < j} \mathbf{PRG}(s_{i,j}) \\ &\quad - \sum_{i \in V_3} \sum_{j \in V_3 \cap \text{Adj}(i); i < j} \mathbf{PRG}(s_{i,j}) - \sum_{i \in V_2 \setminus V_3} \sum_{j \in V_3 \cap \text{Adj}(i); i < j} \mathbf{PRG}(s_{i,j}) \\ &= \sum_{i \in V_3} \sum_{j \in (V_2 \setminus V_3) \cap \text{Adj}(i); i < j} \mathbf{PRG}(s_{i,j}) - \sum_{j \in V_3} \sum_{i \in (V_2 \setminus V_3) \cap \text{Adj}(j); i < j} \mathbf{PRG}(s_{i,j}) \\ &\stackrel{(b)}{=} \sum_{i \in V_3} \sum_{j \in (V_2 \setminus V_3) \cap \text{Adj}(i); i < j} \mathbf{PRG}(s_{i,j}) - \sum_{i \in V_3} \sum_{j \in (V_2 \setminus V_3) \cap \text{Adj}(i); i > j} \mathbf{PRG}(s_{i,j}) \\ &= \sum_{i \in V_3} \sum_{j \in (V_2 \setminus V_3) \cap \text{Adj}(i); i < j} \mathbf{PRG}(s_{i,j}) - \sum_{i \in V_3} \sum_{j \in (V_2 \setminus V_3) \cap \text{Adj}(i); i > j} \mathbf{PRG}(s_{i,j}) \\ &= \sum_{i \in V_3} \sum_{j \in (V_3^+ \setminus V_3) \cap \text{Adj}(i); i < j} \mathbf{PRG}(s_{i,j}) - \sum_{i \in V_3} \sum_{j \in (V_3^+ \setminus V_3) \cap \text{Adj}(i); i > j} \mathbf{PRG}(s_{i,j}), \end{aligned}$$

where (a) and (b) come from  $s_{i,j} = s_{j,i}$ . In order to obtain the sum of unmasked local models  $\sum_{i \in V_3} \theta_i$  from the sum of masked local models  $\sum_{i \in V_3} \tilde{\theta}_i$ , the server should cancel out all the random terms in  $\sum_{i \in V_3} \mathbf{PRG}(b_i) + \mathbf{z}$ . In other words, the server should reconstruct  $b_i$  for all  $i \in V_3$  and  $s_j^{SK}$  for all  $j \in V_3^+ \setminus V_3$ . Since the server can obtain  $|(\text{Adj}(i) \cup \{i\}) \cap V_4|$  secret shares of client  $i$  in **Step 3**,  $|(\text{Adj}(i) \cup \{i\}) \cap V_4| \geq t_i$  for all  $i \in V_3^+$  is a sufficient condition for reliability.

Now we prove the converse part by contrapositive. Suppose there exists  $i \in V_3^+$  such that  $|(\text{Adj}(i) \cup \{i\}) \cap V_4| < t_i$ . In this case, note that the server cannot reconstruct both  $s_i^{SK}$  and  $b_i$  from the shares. If  $i \in V_3$ , the server cannot subtract  $\mathbf{PRG}(b_i)$  from  $\sum_{i \in V_3} \tilde{\theta}_i$ . As a result, the server cannot obtain  $\sum_{i \in V_3} \theta_i$ . If  $i \in V_3^+ \setminus V_3$ , the server cannot subtract  $\mathbf{PRG}(s_{i,j})$  for all  $j \in V_3$  since the server does not have any knowledge of neither  $s_i^{SK}$  nor  $s_j^{SK}$ . Therefore, the server cannot compute  $\sum_{i \in V_3} \theta_i$ , which completes the proof.  $\square$

### C.2 PROOF OF THEOREM 2

*Proof.* Let  $\mathcal{T} \subset V_3$  be a set of compromised clients by eavesdropper satisfying  $\mathcal{T} \notin \{\emptyset, V_3\}$ . It is sufficient to prove the following statement: given a connected graph  $G_3$ , an eavesdropper cannot

obtain the partial sum of local models  $\sum_{i \in \mathcal{T}} \theta_i$  from the sum of masked models  $\sum_{i \in \mathcal{T}} \tilde{\theta}_i$ . Note that the sum of masked local models  $\sum_{i \in \mathcal{T}} \tilde{\theta}_i$  accessible to the eavesdropper is expressed as

$$\sum_{i \in \mathcal{T}} \tilde{\theta}_i = \sum_{i \in \mathcal{T}} \theta_i + \sum_{i \in \mathcal{T}} \mathbf{PRG}(b_i) + \mathbf{z},$$

where

$$\begin{aligned} \mathbf{z} &= \sum_{i \in \mathcal{T}} \sum_{j \in V_2 \cap \text{Adj}(i); i < j} \mathbf{PRG}(s_{i,j}) - \sum_{i \in \mathcal{T}} \sum_{j \in V_2 \cap \text{Adj}(i); i > j} \mathbf{PRG}(s_{i,j}) \\ &= \sum_{i \in \mathcal{T}} \sum_{j \in V_2 \cap \text{Adj}(i); i < j} \mathbf{PRG}(s_{i,j}) - \sum_{j \in V_2} \sum_{i \in \mathcal{T} \cap \text{Adj}(j); i > j} \mathbf{PRG}(s_{i,j}) \\ &\stackrel{(a)}{=} \sum_{i \in \mathcal{T}} \sum_{j \in V_2 \cap \text{Adj}(i); i < j} \mathbf{PRG}(s_{i,j}) - \sum_{i \in V_2} \sum_{j \in \mathcal{T} \cap \text{Adj}(i); i < j} \mathbf{PRG}(s_{i,j}) \\ &= \sum_{i \in \mathcal{T}} \sum_{j \in \mathcal{T} \cap \text{Adj}(i); i < j} \mathbf{PRG}(s_{i,j}) + \sum_{i \in \mathcal{T}} \sum_{j \in (V_2 \setminus \mathcal{T}) \cap \text{Adj}(i); i < j} \mathbf{PRG}(s_{i,j}) \\ &\quad - \sum_{i \in \mathcal{T}} \sum_{j \in \mathcal{T} \cap \text{Adj}(i); i < j} \mathbf{PRG}(s_{i,j}) - \sum_{i \in V_2 \setminus \mathcal{T}} \sum_{j \in \mathcal{T} \cap \text{Adj}(i); i < j} \mathbf{PRG}(s_{i,j}) \\ &= \sum_{i \in \mathcal{T}} \sum_{j \in (V_2 \setminus \mathcal{T}) \cap \text{Adj}(i); i < j} \mathbf{PRG}(s_{i,j}) - \sum_{j \in \mathcal{T}} \sum_{i \in (V_2 \setminus \mathcal{T}) \cap \text{Adj}(j); i < j} \mathbf{PRG}(s_{i,j}) \\ &\stackrel{(b)}{=} \sum_{i \in \mathcal{T}} \sum_{j \in (V_2 \setminus \mathcal{T}) \cap \text{Adj}(i); i < j} \mathbf{PRG}(s_{i,j}) - \sum_{i \in \mathcal{T}} \sum_{j \in (V_2 \setminus \mathcal{T}) \cap \text{Adj}(i); i > j} \mathbf{PRG}(s_{i,j}) \\ &= \left\{ \sum_{i \in \mathcal{T}} \sum_{j \in (V_2 \setminus V_3) \cap \text{Adj}(i); i < j} \mathbf{PRG}(s_{i,j}) - \sum_{i \in \mathcal{T}} \sum_{j \in (V_2 \setminus V_3) \cap \text{Adj}(i); i > j} \mathbf{PRG}(s_{i,j}) \right\} \\ &\quad + \left\{ \sum_{i \in \mathcal{T}} \sum_{j \in (V_3 \setminus \mathcal{T}) \cap \text{Adj}(i); i < j} \mathbf{PRG}(s_{i,j}) - \sum_{i \in \mathcal{T}} \sum_{j \in (V_3 \setminus \mathcal{T}) \cap \text{Adj}(i); i > j} \mathbf{PRG}(s_{i,j}) \right\}. \end{aligned}$$

Here, (a) and (b) come from  $s_{i,j} = s_{j,i}$ . If  $G_3 = (V_3, E_3)$  is connected, there exists an edge  $e = \{p, q\}$  such that  $p \in \mathcal{T}$  and  $q \in (V_3 \setminus \mathcal{T})$ . As a consequence, there exists  $\mathbf{PRG}(s_{p,q})$  term in  $\sum_{i \in \mathcal{T}} \tilde{\theta}_i$ . In order to unmask random term  $\mathbf{PRG}(s_{p,q})$ , the eavesdropper need to know at least one of the secret keys of clients  $p$  and  $q$ . However, the eavesdropper cannot obtain any shares of the secret keys by eavesdropping links since the server requests the shares of  $b_p$  and  $b_q$  instead of  $s_p^{SK}$  and  $s_q^{SK}$ . Therefore, the eavesdropper cannot reconstruct the partial sum of local models  $\sum_{i \in \mathcal{T}} \theta_i$  from the partial sum of masked local models  $\sum_{i \in \mathcal{T}} \tilde{\theta}_i$ , which completes the proof.  $\square$

### C.3 PROOF OF THEOREM 3

*Proof.* Consider Erdős-Rényi assignment graph  $G \in G(n, p)$ . Let  $N_i := |\text{Adj}(i)|$  be the degree of node  $i$ , and  $X_i := |\text{Adj}(i) \cap V_4|$  be the number of clients (except client  $i$ ) that successfully send the shares of client  $i$  to the server in **Step 3**. Then,  $N_i$  and  $X_i$  follow the binomial distributions

$$N_i \sim B(n-1, p), \quad X_i \sim B(N_i, (1-q)^4) = B(n-1, p(1-q)^4),$$

respectively. By applying Hoeffding's inequality on random variable  $X_i$ , we obtain

$$P(X_i < (n-1)p(1-q)^4 - \sqrt{(n-1)\log(n-1)}) \leq 1/(n-1)^2.$$

Let  $E$  be the event that the system is not reliable, i.e., the sum of local models  $\sum_{i \in V_3} \theta_i$  is not reconstructed by the server, and  $E_i$  be the event  $\{(|\text{Adj}(i) \cup \{i\}| \cap V_4) < t\}$ , i.e., a secret of client

$i$  is not reconstructed by the server. For a given  $p > \frac{t + \sqrt{(n-1)\log(n-1)}}{(n-1)(1-q)^4}$ , we obtain

$$\begin{aligned} P(E) &\stackrel{(a)}{=} P(\cup_{i \in V_3^+} E_i) \leq P(\cup_{i \in V_3^+} \{X_i < t\}) \leq \sum_{i \in V_3^+} P(X_i < t) \\ &\leq \sum_{i \in [n]} P(X_i < t) = nP(X_1 < t) \\ &\leq nP(X_1 < (n-1)p(1-q)^4 - \sqrt{(n-1)\log(n-1)}) \leq \frac{n}{(n-1)^2} \xrightarrow{n \rightarrow \infty} 0, \end{aligned}$$

where (a) comes from Theorem 1. Therefore, we conclude that CCESA( $n, p$ ) is asymptotically almost surely (a.a.s.) reliable if  $p > \frac{t + \sqrt{(n-1)\log(n-1)}}{(n-1)(1-q)^4}$ . Furthermore, based on the parameter selection rule of  $t$  in Section F, we obtain a lower bound on  $p$  as

$$p > \frac{t + \sqrt{(n-1)\log(n-1)}}{(n-1)(1-q)^4} \geq \frac{\frac{(n-1)p + \sqrt{(n-1)\log(n-1)} + 1}{2} + \sqrt{(n-1)\log(n-1)} - 1}{(n-1)(1-q)^4}.$$

Rearranging the above inequality with respect to  $p$  yields

$$p > \frac{3\sqrt{(n-1)\log(n-1)} - 1}{(n-1)(2(1-q)^4 - 1)}.$$

□

#### C.4 PROOF OF THEOREM 4

*Proof.* Let  $X := |V_3|$  be the number of clients sending its masked local model in **Step 2**. Then,  $X$  follows Binomial random variable  $B(n, (1-q)^3)$ . Given assignment graph  $G$  of CCESA( $n, p$ ), note that the induced subgraph  $G_3 = G - V \setminus V_3$  is an Erdős-Rényi graph  $G(X, p)$ .

First, we prove

$$P(G_3 \text{ is connected} \mid |X - n(1-q)^3| \leq \sqrt{n \ln n}) \xrightarrow{n \rightarrow \infty} 1, \quad (6)$$

if  $p > p^* = \frac{\ln(\lceil n(1-q)^3 - \sqrt{n \ln n} \rceil)}{\lceil n(1-q)^3 - \sqrt{n \ln n} \rceil} (1 + \epsilon)$ . The left hand side of (6) can be rewritten as

$$\begin{aligned} &P(G_3 \text{ is connected} \mid |X - n(1-q)^3| \leq \sqrt{n \ln n}) \\ &= \frac{\sum_{l \in [n(1-q)^3 - \sqrt{n \ln n}, n(1-q)^3 + \sqrt{n \ln n}]} P(X = l) P(G(l, p) \text{ is connected})}{\sum_{l \in [n(1-q)^3 - \sqrt{n \ln n}, n(1-q)^3 + \sqrt{n \ln n}]} P(X = l)}. \end{aligned}$$

Here, we use a well-known property of Erdős-Rényi graph:  $G(l, p)$  is asymptotically almost surely (a.a.s.) connected if  $p > \frac{(1+\epsilon) \ln l}{l}$  for some  $\epsilon > 0$ . Since  $\frac{\ln l}{l}$  is a decreasing function,  $G(l, p)$  is a.a.s. connected for all  $l \in [n(1-q)^3 - \sqrt{n \ln n}, n(1-q)^3 + \sqrt{n \ln n}]$  when  $p > \frac{\ln(\lceil n(1-q)^3 - \sqrt{n \ln n} \rceil)}{\lceil n(1-q)^3 - \sqrt{n \ln n} \rceil}$ .

Thus, for given  $p > p^*$ , we can conclude

$$P(G_3 \text{ is connected} \mid |X - n(1-q)^3| \leq \sqrt{n \ln n}) \xrightarrow{n \rightarrow \infty} 1.$$

Now, we will prove that CCESA( $n, p$ ) is a.a.s. private when  $p > p^*$ . The probability that CCESA( $n, p$ ) is private is lower bounded by

$$\begin{aligned} P(\text{CCESA}(n, p) \text{ is private}) &\stackrel{(a)}{\geq} P(G_3 \text{ is connected}) \\ &= P(|X - n(1-q)^3| \leq \sqrt{n \ln n}) P(G_3 \text{ is connected} \mid |X - n(1-q)^3| \leq \sqrt{n \ln n}) \\ &\quad + P(|X - n(1-q)^3| > \sqrt{n \ln n}) P(G_3 \text{ is connected} \mid |X - n(1-q)^3| > \sqrt{n \ln n}) \\ &\stackrel{(b)}{\geq} (1 - 2/n^2) P(G_3 \text{ is connected} \mid |X - n(1-q)^3| \leq \sqrt{n \ln n}) \xrightarrow{n \rightarrow \infty} 1, \end{aligned}$$

where (a) comes from Theorem 2 and (b) comes from Hoeffding's inequality

$$P(|X - n(1-q)^3| \leq \sqrt{n \ln n}) \geq 1 - 2/n^2,$$

which completes the proof. □

## C.5 PROOF OF THEOREM 5

*Proof.* Consider an Erdős-Rényi assignment graph  $G \in G(n, p)$ . Let  $N_i := |Adj(i)|$  be the degree of node  $i$ , and  $X_i := |Adj(i) \cap V_4|$  be the number of clients (except client  $i$ ) that successfully send the shares of client  $i$  to the server in **Step 3**. Then,  $N_i$  and  $X_i$  follow the binomial distributions

$$N_i \sim B(n-1, p), \quad X_i \sim B(N_i, (1-q)^4) = B(n-1, p(1-q)^4),$$

respectively. Let  $E_i$  be an event  $\{|(Adj(i) \cup \{i\}) \cap V_4| < t\}$ , i.e., a secret of client  $i$  is not reconstructed by the server. We obtain an upper bound on  $P(E_i)$  as

$$\begin{aligned} P(E_i) &\leq P(X_i < t) = \sum_{i=0}^{t-1} \binom{n-1}{i} (p(1-q)^4)^i (1-p(1-q)^4)^{(n-1-i)} \\ &\stackrel{(a)}{=} e^{-(n-1)D(\frac{t-1}{n-1} || p(1-q)^4)} \end{aligned}$$

where (a) comes from Chernoff bound on the binomial distribution. Thus,  $P_e^{(r)}$  is upper bounded by

$$\begin{aligned} P_e^{(r)} &\stackrel{(b)}{=} P(\cup_{i \in V_3^+} E_i) \leq P(\cup_{i \in V_3^+} \{X_i < t\}) \leq \sum_{i \in V_3^+} P(X_i < t) \\ &\leq \sum_{i \in [n]} P(X_i < t) = nP(X_1 < t) = ne^{-(n-1)D(\frac{t-1}{n-1} || p(1-q)^4)}, \end{aligned}$$

where (b) comes from Theorem 1. □

## C.6 PROOF OF THEOREM 6

*Proof.* Let  $P_{dc}(n, p)$  be the probability of an event that Erdős-Rényi graph  $G \in G(n, p)$  is disconnected. Then,  $P_{dc}(n, p)$  is upper bounded as follows.

$$\begin{aligned} P_{dc}(n, p) &= P(G(n, p) \text{ is disconnected}) \\ &= P(\cup_{k=1}^{\lfloor n/2 \rfloor} \{\text{there exists a subset of } k \text{ nodes that is disconnected}\}) \\ &\leq \sum_{k=1}^{\lfloor n/2 \rfloor} P(\text{there exists a subset of } k \text{ nodes that is disconnected}) \\ &\leq \sum_{k=1}^{\lfloor n/2 \rfloor} \binom{n}{k} P(\text{a specific subset of } k \text{ nodes is disconnected}) \\ &= \sum_{k=1}^{\lfloor n/2 \rfloor} \binom{n}{k} (1-p)^{k(n-k)} \end{aligned}$$

Therefore,  $P_e^{(p)}$  is upper bounded by

$$\begin{aligned} P_e^{(p)} &\stackrel{(a)}{\leq} P(G_3 = G - V \setminus V_3 \text{ is disconnected}) \\ &= \sum_{m=0}^n P(G_3 \text{ has } m \text{ vertices}) P_{dc}(m, p) \\ &= \sum_{m=0}^n \binom{n}{m} (1-q)^{3m} (1-(1-q)^3)^{(n-m)} \cdot P_{dc}(m, p) \\ &= \sum_{m=0}^n \binom{n}{m} (1-q)^{3m} (1-(1-q)^3)^{(n-m)} \sum_{k=1}^{\lfloor m/2 \rfloor} \binom{m}{k} (1-p)^{k(m-k)}, \end{aligned}$$

where (a) comes from Theorem 2. □

## D REQUIRED RESOURCES OF CCESA

### D.1 COMMUNICATION COST

Here, we derive the additional communication bandwidth  $B_{CCESA}$  used at each client for running CCESA, compared to the bandwidth used for running federated averaging (McMahan et al., 2017). We consider the worst-case scenario having the maximum additional bandwidth, where no client fails during the operation.

The required communication bandwidth of each client is composed of four parts. First, in **Step 0**, each client  $i$  sends two public keys to the server, and receives  $2|Adj(i)|$  public keys from other clients. Second, in **Step 1**, each client  $i$  sends encrypted  $2|Adj(i)|$  shares to other nodes, and receives  $2|Adj(i)|$  shares from other nodes through the server. Third, in **Step 2**, each client  $i$  sends a masked data  $y_i$  of  $mR$  bits. Here,  $m$  is the dimension of model parameters where each parameter is represented in  $R$  bits. Fourth, in **Step 3**, each client  $i$  sends  $|Adj(i)| + 1$  shares to the server. Therefore, total communication bandwidth of client  $i$  can be expressed as

$$(\text{total communication bandwidth}) = 2(|Adj(i)| + 1)a_K + (5|Adj(i)| + 1)a_S + mR,$$

where  $a_K$  and  $a_S$  are the number of bits required for exchanging public keys and the number of bits in a secret share. Since each client  $i$  requires  $mR$  bits to send the private vector  $\theta_i$  in the federated averaging (McMahan et al., 2017), we have

$$B_{CCESA} = 2(|Adj(i)| + 1)a_K + (5|Adj(i)| + 1)a_S.$$

If we choose the connection probability  $p = (1 + \epsilon)p^*$  for a small  $\epsilon > 0$ , we have  $B_{CCESA} = O(\sqrt{n} \log n)$ , where  $p^*$  is defined in (5). Note that the additional bandwidth  $B_{SA}$  required for SA can be similarly obtained as  $B_{SA} = 2na_K + (5n - 4)a_S$  having  $B_{SA} = O(n)$ . Thus, we have

$$\frac{B_{CCESA}}{B_{SA}} \rightarrow 0$$

as  $n$  increases, showing the scalability of CCESA. These results are summarized in Table 1 in the main manuscript.

### D.2 COMPUTATIONAL COST

We evaluate the computational cost of CCESA. Here we do not count the cost for computing the signatures since it is negligible. First, we derive the computational cost of each client. Given the number of model parameters  $m$  and the number of clients  $n$ , the computational cost of client  $i$  is composed of three parts: (1) computing  $2|Adj(i)|$  key agreements, which takes  $O(|Adj(i)|)$  time, (2) generating shares of  $t_i$ -out-of- $|Adj(i)|$  secret shares of  $s_i^{SK}$  and  $b_i$ , which takes  $O(|Adj(i)|^2)$  time, and (3) generating masked local model  $\tilde{\theta}_i$ , which requires  $O(m|Adj(i)|)$  time. Thus, the total computational cost of each client is obtained as  $O(|Adj(i)|^2 + m|Adj(i)|)$ . Second, the server's computational cost is composed of two parts: (1) reconstructing  $t_i$ -out-of- $|Adj(i)|$  secrets from shares for all clients  $i \in [n]$ , which requires  $O(|Adj(i)|^2)$  time, and (2) removing masks from masked sum of local models  $\sum_{i=1}^n \tilde{\theta}_i$ , which requires  $O(m|Adj(i)|^2)$  time in the worst case. As a result, the total computational cost of the server is  $O(m|Adj(i)|^2)$ . If we choose  $p = (1 + \epsilon)p^*$  for small  $\epsilon > 0$ , the total computational cost per each client is  $O(n \log n + m\sqrt{n} \log n)$ , while the total computation cost of the server is  $O(mn \log n)$ . The computational cost of SA can be obtained in a similar manner, by setting  $Adj(i) = n - 1$ ; each client requires  $O(n^2 + mn)$  time while the server requires  $O(mn^2)$  time. These results are summarized in Table 1 in the main manuscript.

## E RELIABILITY AND PRIVACY OF CCESA

Here, we analyze the asymptotic behavior of probability that a system is reliable/private. In our analysis, we assume that the connection probability is set to  $p^*$  and the parameter  $t$  used in the secret sharing is selected based on the rule in Section F. First, we prove that a system is reliable with probability  $\geq 1 - O(ne^{-n \log n})$ . Using Theorem 5, the probability that a system is reliable can be directly derived as

$$\begin{aligned} P(\text{A system is reliable}) &= 1 - P_e^{(r)} \\ &\geq 1 - ne^{-(n-1)D_{KL}(\frac{t-1}{n-1} || p^*(1-q)^4)}. \end{aligned}$$



Using the fact that Kullback-Leibler divergence term satisfies

$$\begin{aligned} D_{KL}\left(\frac{t-1}{n-1} \parallel p^*(1-q)^4\right) &= \frac{t-1}{n-1} \log\left(\frac{\frac{t-1}{n-1}}{p^*(1-q)^4}\right) + \left(1 - \frac{t-1}{n-1}\right) \log\left(\frac{1 - \frac{t-1}{n-1}}{1 - p^*(1-q)^4}\right) \\ &= \Theta(\sqrt{\log n/n}), \end{aligned}$$

we conclude that  $\text{CCESA}(n, p^*)$  is reliable with probability  $\geq 1 - O(ne^{-\sqrt{n \log n}})$ .

Now we prove that a system is private with probability  $\geq 1 - O(n^{-C})$  for an arbitrary  $C > 0$ . Using Theorem 6, the probability that a system is private can be obtained as

$$\begin{aligned} P(\text{A system is private}) &= 1 - P_e^{(p)} \\ &\geq 1 - \sum_{m=0}^n a_m b_m, \end{aligned}$$

where  $a_m = \binom{n}{m}(1-q)^{3m}(1 - (1-q)^3)^{(n-m)}$  and  $b_m = \sum_{k=1}^{\lfloor m/2 \rfloor} \binom{m}{k}(1-p^*)^{k(m-k)}$ . Note that the summation term  $\sum_{m=0}^n a_m b_m$  can be broken up into two parts:  $\sum_{m=0}^{m_{th}} a_m b_m$  and  $\sum_{m=m_{th}+1}^n a_m b_m$ , where  $m_{th} = \lfloor n(1-q)^3/2 \rfloor$ . In the rest of the proof, we will prove two lemmas, showing that  $\sum_{m=0}^{m_{th}} a_m b_m = O(e^{-n})$  and  $\sum_{m=m_{th}+1}^n a_m b_m = O(n^{-C})$ , respectively.

**Lemma 1.**

$$\sum_{m=0}^{m_{th}} a_m b_m = O(e^{-n})$$

*Proof.* Since  $b_m \leq 1$  for all  $m$ , we have

$$\sum_{m=0}^{m_{th}} a_m b_m \leq \sum_{m=0}^{m_{th}} a_m.$$

Note that  $a_m = P(X = m)$  holds for binomial random variable  $X = B(n, (1-q)^3)$ . By utilizing Hoeffding's inequality, we have

$$\sum_{m=0}^{m_{th}} a_m = P(X \leq m_{th}) \leq e^{-2(n(1-q)^3 - m_{th})^2} \leq e^{-n(1-q)^6/2}.$$

Therefore, we conclude that  $\sum_{m=0}^{m_{th}} a_m b_m = O(e^{-n})$ .  $\square$

**Lemma 2.**

$$\sum_{m=m_{th}+1}^n a_m b_m = O(n^{-C})$$

*Proof.* Since  $a_m \leq 1$  for all  $m$ , we have

$$\sum_{m=m_{th}+1}^n a_m b_m \leq \sum_{m=m_{th}+1}^n b_m.$$

Let  $C > 0$  be given. Then, the upper bound on  $b_m$  can be obtained as

$$\begin{aligned} b_m &= \sum_{k=1}^{\lfloor m/2 \rfloor} \binom{m}{k} (1-p^*)^{k(m-k)} \leq \sum_{k=1}^{\lfloor m/2 \rfloor} \binom{m}{k} e^{-k(m-k)p^*} \\ &= \sum_{k=1}^{\lfloor m/2 \rfloor} \binom{m}{k} m^{-\lambda k(m-k)/m} = c_m + d_m \end{aligned}$$

where  $\lambda = p^* n / \log n$ ,  $c_m = \binom{m}{k} \sum_{k=1}^{k^*} m^{-\lambda k(m-k)/m}$ ,  $d_m = \binom{m}{k} \sum_{k=k^*+1}^{\lfloor m/2 \rfloor} m^{-\lambda k(m-k)/m}$ , and  $k^* = \lfloor m(1 - \frac{C+2}{\lambda}) \rfloor$  for some  $C > 0$ . The first part of summation is upper bounded by

$$\begin{aligned} c_m &= \sum_{k=1}^{k^*} \binom{m}{k} m^{-\lambda k(m-k)/m} \leq \sum_{k=1}^{k^*} m^{-k \lfloor \lambda(m-k)/m-1 \rfloor} \leq \sum_{k=1}^{k^*} m^{-k \lfloor \lambda(m-k^*)/m-1 \rfloor} \\ &\leq \frac{m^{-\lfloor \lambda(m-k^*)/m-1 \rfloor}}{1 - m^{-\lfloor \lambda(m-k^*)/m-1 \rfloor}} = \frac{m^{-(C+1)}}{1 - m^{-(C+1)}}. \end{aligned}$$

The second part of summation, we will use the bound  $\binom{n}{k} \leq (\frac{en}{k})^k$ . Using this bound,  $d_m$  is upper bounded by

$$\begin{aligned} d_m &= \sum_{k=k^*+1}^{\lfloor m/2 \rfloor} \binom{m}{k} m^{-\lambda k(m-k)/m} \leq \sum_{k=k^*+1}^{\lfloor m/2 \rfloor} \left( \frac{em^{1-\lambda(m-k)/m}}{k} \right)^k \leq \sum_{k=k^*+1}^{\lfloor m/2 \rfloor} \left( \frac{em^{1-\lambda(m-k)/m}}{k^*+1} \right)^k \\ &\leq \sum_{k=k^*+1}^{\lfloor m/2 \rfloor} \left( \frac{em^{-\lambda(m-k)/m}}{1 - \lambda^{-1}(C+2)} \right)^k \leq \sum_{k=k^*+1}^{\lfloor m/2 \rfloor} \left( \frac{em^{-\lambda/2}}{1 - \lambda^{-1}(C+2)} \right)^k. \end{aligned}$$

For sufficiently large  $\lambda$ , we have  $em^{-\lambda/2}/(1 - \lambda^{-1}(C+2)) < \delta$  for some  $\delta < 1$ . Therefore,  $d_m$  is upper bounded by

$$d_m \leq \sum_{k=k^*+1}^{\infty} \delta^k = \frac{\delta^{k^*}}{1 - \delta} = O(\delta^{mC'})$$

where  $C' = (1 - \lambda^{-1}(C+2)) > 0$ . Combining upper bounds on  $c_m$  and  $d_m$ , we obtain  $b_m = O(m^{-(C+1)})$ . Since  $b_m$  is a decreasing function of  $m$ ,

$$\sum_{m=m_{th}+1}^n b_m \leq \sum_{m=m_{th}+1}^n b_{m_{th}+1} = (n - m_{th})b_{m_{th}+1} \stackrel{(a)}{=} O(n^{-C})$$

holds where (a) comes from  $m_{th} = \lfloor n(1-q)^3/2 \rfloor$ .

□

Combining the above two lemmas, we conclude that  $\text{CCESA}(n, p^*)$  is private with probability  $\geq 1 - O(n^{-C})$  for arbitrary  $C > 0$ . These results on the reliability and the privacy are summarized in Table 1 of the main manuscript.

## F DESIGNING THE PARAMETER $t$ FOR THE SECRET SHARING

Here we provide a rule for selecting parameter  $t$  used in the secret sharing. In general, setting  $t$  to a smaller number is better for tolerating dropout scenarios. However, when  $t$  is excessively small, the system is vulnerable to the *unmasking attack* of adversarial server; the server may request shares of  $b_i$  and  $s_i^{SK}$  to disjoint sets of remaining clients simultaneously, which reveals the local model  $\theta_i$  to the server. The following proposition provides a rule of designing parameter  $t$  to avoid such unmasking attack.

**Proposition 1** (Lower bound on  $t$ ). *For  $\text{CCESA}(n, p)$ , let  $t > \frac{(n-1)p + \sqrt{(n-1)\log(n-1)+1}}{2}$  be given. Then, the system is asymptotically almost surely secure against the unmasking attack.*

*Proof.* Let  $E$  be the event that at least one of local models are revealed to the server, and  $E_i$  be the event that  $i^{\text{th}}$  local model  $\theta_i$  is revealed to the server. Note that  $\theta_i$  is revealed to the server if  $t$  clients send the shares of  $b_i$  and other  $t$  clients send the shares of  $s_i^{SK}$  in **Step 3**. Therefore,

$$\begin{aligned} P(E_i) &\leq P(|\text{Adj}(i) \cup \{i\}| \cap V_4| \geq 2t) \\ &\leq P(|\text{Adj}(i) \cup \{i\}| \geq 2t) = P(|\text{Adj}(i)| \geq 2t-1) \\ &\leq P(|\text{Adj}(i)| > (n-1)p + \sqrt{(n-1)\log(n-1)}) \stackrel{(a)}{\leq} \frac{1}{(n-1)^2}, \end{aligned}$$

where (a) comes from Hoeffding’s inequality of binomial random variable. As a result, we obtain

$$P(E) = P(\cup_{i \in [n]} E_i) \leq \sum_{i \in [n]} P(E_i) = nP(E_1) = \frac{n}{(n-1)^2} \xrightarrow{n \rightarrow \infty} 0,$$

which completes the proof.  $\square$

As stated above, setting  $t$  to a smaller number is better to tolerate the dropout of multiple clients. Thus, as in the following remark, we set  $t$  to be the minimum value avoiding the unmasking attack.

**Remark 4** (Design rule for  $t$ ). *Throughout the paper, we set  $t = \lceil \frac{(n-1)p + \sqrt{(n-1)\log(n-1)+1}}{2} \rceil$  for  $CCSA(n, p)$ , in order to secure a system against the unmasking attack and provide the maximum tolerance against dropout scenarios.*

## G DETAILED EXPERIMENTAL SETUP

### G.1 AT & T FACE DATASET

AT&T Face dataset contains images of 40 members. We allocated the data to  $n = 40$  clients participating in the federated learning, where each client contains the images of a specific member. This experimental setup is suitable for the practical federated learning scenario where each client has its own image and the central server aggregates the local models for face recognition. Following the previous work (Fredrikson et al., 2015) on the model inversion, we used softmax regression for the classification. Both the number of local training epochs and the number of global aggregation rounds are set to  $E_{local} = E_{global} = 50$ , and we used the SGD optimizer with learning rate  $\gamma = 0.05$ .

### G.2 CIFAR-10 DATASET

#### G.2.1 RELIABILITY EXPERIMENT IN FIG. 4

We ran experiments under the federated learning setup where 50000 training images are allocated to  $n = 100$  clients. Here, we considered two scenarios for data allocation: one is partitioning the data in the i.i.d. manner (i.e., each client randomly obtains 500 images), while the other is non-i.i.d. allocation scenario. For the non-i.i.d. scenario, we followed the procedure of (McMahan et al., 2017). Specifically, the data is first sorted by its category, and then the sorted data is divided into 200 shards. Each client randomly chooses 2 shards for its local training data. Since each client has access to at most 2 classes, the test accuracy performance is degraded compared with the i.i.d. setup. For training the classifier, we used VGG-11 network and the SGD optimizer with learning rate  $\gamma = 0.1$  and momentum  $\beta = 0.5$ . The local training epoch is set to  $E_{local} = 3$ .

#### G.2.2 PRIVACY EXPERIMENTS IN TABLE 3 AND TABLE B.1

We conducted experiments under the federated learning setup where  $n_{train}$  training images are assigned to  $n = 10$  clients. We considered i.i.d. data allocation setup where each client randomly obtains  $n_{train}/10$  training images. The network architecture, the optimizer, and the number of local training epochs are set to the options used in Sec. G.2.1.

### G.3 CONNECTION PROBABILITY SETUP IN FIG. 3

In Fig. 3, connection probability  $p$  is chosen as  $p > p^*$  where  $p^*$  is the minimum connection probability for achieving both reliability and privacy. Recall that  $p^*$  is a function of  $n$  and  $q_{total}$  as in (5). We select different  $p$  values for various  $n$  in a way that each value of  $p$  is slightly above the threshold value  $p^*$  for all  $q_{total}$ . The detailed values of  $p$  with respect to  $n$  are provided in Table G.2.

$n$	100	200	300	400	500	600	700	800	900	1000
$p$	0.80	0.63	0.53	0.46	0.42	0.39	0.37	0.35	0.33	0.32

Table G.2: Connection probability setup in Fig. 3