JOURNAL OF INFORMATION SCIENCE AND ENGINEERING XX, XXXX-XXXX (2016)
DOI: XX.XXX.XXXX.XX

Secure Aggregation with Computational Scalability based on Additive Homomorphic Encryption*

ZE YANG, YOULIANG TIAN+, KUN NIU
The State Key Laboratory of Public Big Data and the College of Computer Science and Technology
Guizhou University
Guiyang, China
E-mail: {gzuyangze@gmail.com, yltian@gzu.edu.cn}

Cloud-based computing framework resolves the dilemma of users' computational capabilities mismatching the demand for high-quality models. However, the explosive growth of data due to the massive popularity of terminal devices not only imposes higher requirements on the performance of the cloud but also increases the risk of private data leakage in the cloud. In this paper, we focus on cloud-based secure aggregation with private data on cloud devices, where the cloud can only handle fixed bit lengths. First, we propose a truncate-mapping scheme for private data to alleviate the resource limitations of the cloud by encoding big data into different shares for parallel computing. Second, we define the calculations on the encoded data, involving the secure addition and secure comparison, to achieve secure aggregation on private data. It is worth noting that not only the results of the computations on the encoded data are the same as on the raw private data, but also the proposed scheme can make full use of the power of the computing devices. That is the more devices concerning the computation, the more precise results can be obtained instead of steadfastly enhancing the computing power of the cloud. Finally, the theoretical analysis and experiments show that the proposed scheme is secure, effective and suitable for practical applications.

Keywords: secure aggregation, ciphertext parallel computing, security, distributed systems

1. INTRODUCTION

With the rapid development of computing devices and next-generation communication technologies, intelligent control can be achieved by deploying millions of devices in the real world to provide real-time data and feedback[1]. As the most basic and core, aggregation requires timely processing of real-time data generated by end devices to achieve the evaluation and optimisation of the system and is widely used in the fields of finance[2], control[3], and insurance[4]. While aggregation promotes the sharing and utilisation of data and brings great convenience to the development of the industry, the concentration of privacy issues, which is essentially caused by the decentralisation of privacy, is becoming more and more serious. For example, in a control system (control system[5]), sensor nodes upload the collected data to regulate the system to be always in a secure and stable

Received January 29, 2024.
+Corresponding author
Communicated by the editor.

state. If the processing of real-time data and the state privacy of the system are compromised, an adversary can launch an attack on the system, leading to loss of control of the control system or even system paralysis.

In particular, the most generalised problem in aggregation is weighted sum aggregation, where the aggregator collects data from agents and gets the aggregation result based on their corresponding contributions. This not only solves the data island well but also lays the foundation for obtaining high-quality data services. In particular, scenarios such as a) federated learning[6, 7], b) decentralised and collaborative linear control systems[8, 9, 10], c) smart grid scheduling services[11, 12, 13], etc. require weighted aggregation. Among them, a) can be regarded as a secure aggregation scheme for weighted information is public, i.e., only the privacy of the input data and result needs to be protected. In b), the weight information can only be held by the aggregator for economic and other considerations. In c), the weight information corresponding to the input data is also sensitive, and it is desired to achieve secure aggregation by the aggregator under the condition of non-disclosure, which can be modelled as the private weighted sum aggregation, which is the focus of this paper due to its stringent requirements on privacy.

Further, to implement the secure aggregation, researchers have proposed secure computation protocols based on cryptographic primitives, such as Secure Multi-Party Computing (SMPC)[14], Homomorphic Encryption (HE)[15], and Differential Privacy[16]. The agent providing the data performs privacy-preserving secure computations by interfering with private input, which is then outsourced to the cloud to provide computational services for revenue or computational results. Among them, protocols based on SMPC suffer from excessive communication overhead, while differential privacy-based schemes require a trade-off between privacy and the accuracy of results. HE as a cryptographic of secure computation is widely used in weighted aggregation, a common form of data sharing, but suffers from the disadvantage of high computational overhead. To address this, partially homomorphic-based privacy aggregation schemes have been proposed in[17].

However, the above scheme has two significant drawbacks constraining application. One is that it does not support secure comparison operations of ciphertexts, which is particularly significant when computing on real-time data (e.g., updating the ciphertexts variable to determine whether the iterations continue or stop). The second is that even if more devices can handle $l_i (l_i \in \mathbb{Z})$ bits are involved in the computation. The only way to get more accurate results is to enhance the computational power of the devices instead of increasing the number of devices, which would result in more expensive expenses. Therefore, summarising the contributions of this paper is as follows:

- We propose a truncate-mapping scheme, which can map the real number with l-bits into β shares with l_i bits, such that $\sum_{i=1}^{\beta} l_i = l$, allowing devices that can only process l_i bits to collaboratively compute a result with higher accuracy than one.
- We construct a secure comparison protocol for secure updating of ciphertexts, which can be implemented by resorting the random numbers and encryption in only two rounds of communication regardless of the bits of the data.
- Based on the agent-cloud-target interaction framework, we simulate a realistic secure aggregation scenario with the Raspberry Pi and desktop to demonstrate the

computational overhead and communication overhead of the proposed protocol, which shows the efficiency of our proposal as well as security in private weighted sum aggregation.

The remainder of this paper is organized as follows. We show the related work in Section 2. In Section 3, we briefly present the problem statement. We describe the proposed scheme in section 4. In Section 5, we show the private weighted sum aggregation scheme. In section 6, we give to analyze the security and complexity. Finally, Section 7 experiments and concludes this paper in 8.

2. Related Work

2.1 Secure aggregation

To preserve data privacy in aggregation, the straightforward way is to resort to cryptography tools. Homomorphic encryptions(HE)[18], which can support the computation over ciphertext without decryption, have been adopted to secure aggregationl[19]. Similarly, Regueiro *et al*.[20] designed a privacy-enhanced distributed secure data aggregation protocol based on blockchain and HE, which uses blockchain as a distributed ledger and facilitates efficient data aggregation through smart contracts. Although secure aggregation protocols constructed based on homomorphic encryption (HE) guarantee confidentiality not only for computation but also for transmission and storage processes. However, due to the ciphertext refresh operation, it results in too high computational overhead to be applied in reality.

To enhance the efficiency of the protocols, secure aggregation schemes based on differential privacy have received much attention[21]. Additionally, Wu *et al*.[22] proposed a federated learning method that combines an adaptive gradient descent strategy and differential privacy to ensure that the federated learning scheme can be efficiently trained with limited communication costs. Further, to make the differential privacy-based secure aggregation scheme further more accurate, Wang *et al*.[23] formulate and derive the optimal dummy variable sizes for both non-adaptive and adaptive dummy variables to adjust the amount of noise that needs to be added. Compared with homomorphic encryption-based aggregation schemes, differential privacy gains a great improvement in computational efficiency, but requires a trade-off between data quality and privacy, and does not yield more accurate results in certain scenarios with stringent requirements on data accuracy.

Partially homomorphic encryptions(PHE)[24] to be designed for secure computation protocols to improve the efficiency of the protocols. Merad-Boudia *et al*.[25] designed an efficient and secure multidimensional data aggregation protocol based on the paillier encryption scheme, which aims to reduce the multidimensional data cipher aggregation overhead, and also employs batch validation techniques to ensure the correctness of the results. Further, Tjell *et al*.[26] reduce the communication and computational overhead of private aggregation protocols by performing the aggregation computation in the preprocessing phase, and their scheme achieves efficient privacy aggregation without the need to introduce any trusted third party. Similarly, Park *et al*.[27] proposed a privacy-preserving federated learning algorithm that enables centralized servers to aggregate encrypted lo-

cal model participation without decryption, and the proposed algorithm allows each node to use different private keys in the same FL-based system. It should be deserved that the above scheme achieves a satisfactory performance in terms of efficiency, however, the privacy analysis is not comprehensive especially when the weight is unknown to the data provider and aggregator. Up to now, the most thorough analysis of privacy-weighted aggregation is presented in [17], which analyses in detail the challenges of secure aggregation under different privacy requirements and designs a secure protocol based on the paillier encryption scheme. Meanwhile, it is the scheme that we mainly focus on for comparison.

It is worth noting that secure aggregation protocols based on additive homomorphic encryption schemes do not support the operations concerning plaintexts that are real numbers, the straightforward solution is to preset the number of valid bits of data to satisfy the requirement of the computation result precision, which will sacrifice the accuracy of the result in some special scenarios. Further, most schemes do not support non-linear operations such as secure comparisons. Meanwhile, most existing schemes have not yet fully utilized the computational power of devices in distributed computing scenarios.

2.2 Secure aggregation in federated learning

As a research hotspot in recent years, federated learning is becoming more and more popular in contemporary practices due to its ability to allow decentralized data to jointly train a high-quality model [29]. Its core is secure aggregation, and commonly used means are secret sharing [30], DP [31] and homomorphic computation [32], in which, in the secret sharing-based approach, zero-sum masking of private values is used to realize model training and ensure the security of private data. However, it requires additional interactions and is sensitive to client dropout, which is unrealistic for which in reality is always facing challenges of client dropout, network latency and software blocking [33]. Meanwhile, the DP-based federated learning is limited in the promotion scenario because it faces the need to trade-off between model quality and data security. As a post-quantum era security solution, full homomorphic encryption has been progressively introduced into federated learning scenarios [34]. Unfortunately, the scalability of cryptographic computation and communication becomes a bottleneck, severely limiting the feasibility in the real world [35]. To reduce the overhead of the training process and make it more practical, Jin *et al*.[36] and others selectively encrypt sensitive data to provide customized privacy protection. Subsequently, Fang *et al*.[37] used the improved Paillier to implement a multi-party privacy federated learning framework, and their experimental results demonstrated that the homomorphic scheme based on the homomorphic scheme is basically the same as the non-homomorphic scheme in terms of model accuracy. Subsequently, in order to reduce computational and communication overheads, Zhang *et al*.[38] improved the Paillier scheme based on the Chinese Residual Theorem to speed up the computation. Subsequently, Zhang *et al*.[35] utilized batch quantization and gradient encryption to reduce the number of model parameters that need to be encrypted. However, it is not the best solution in terms of reducing the communication overhead, to solve this problem, Jiang *et al*.[39] proposed FLASHE, which involves only modular addition operations on random numbers to optimize the computational efficiency in order to improve the performance of the scheme. Meanwhile, another limitation against homomorphic schemes in federated learning is thatwhich cannot resist attacks from curious internal clients as

well as collusion attacks between clients and the server due to the fact that all the clients use the same key pairfor encryption and decryption. To overcome these drawbacks, Du *et al*.[40] proposed a threshold multi-key homomorphic scheme, tMK-CKKS, which not only allows clients to join or exit during the training process, but also achieves the goal of reducing the communication overhead while resisting collusion attacks from no more than t (threshold) internal clients by packing multiple messages into a single ciphertext. Subsequently, a large number of federated learning based on multi-key homomorphisms have been proposed to cope with this type of collusion attacks to enhance the robustness of the system[41, 42, 43]. Further, Wang *et al*.[44] solves the data overload problem by analyzing users' functional preferences for wireless mobile network device types, and constructs a formal application function model based on homomorphic encryption to prevent privacy leakage of model parameters. Unfortunately, its scheme has not yet taken into account the computational capacity constraints of devices when offloading tasks, and its scheme is difficult to be applied to devices with limited computational resources in reality.

In a word, all the above schemes focus on reducing the computation overhead by designing appropriate encrypted data and reducing the communication overhead by ciphertext packing. However, they are ineffective in the face of the increasing number of complicated computation tasks faced by a large number of devices with limited computational resources. Meanwhile, how to judge the range of ciphertext without decrypting it without disclosing the key and private data is of great interest in practical control. Guided by the above secure aggregation scheme, under the requirements of data privacy, the efficiency of the computation, quality of results, and especially the full use of the computational power of distributed devices, this paper develops a privacy computation scheme that can be applied to support multi-device collaborative secure aggregation using partially homomorphic encryption.

3. Problem Statement

3.1 Problem Setup

The private weighted sum aggregation is illustrated in Fig. 1. We consider a system with n agents, including one aggregator with β devices that are capable of handling l bits, a target that accesses the result and a system administrator. Each agent $i \in [n]$, having private data $x_i(t) \in R^{n_i}$ at time t and the aggregator having a mass of computing devices wants to calculate a sum of private data in the system $\mathbf{x}(t) \in R^{n_i}$, where $\mathbf{W}_i \in R^{n_i \times n_i}$ is the corresponding weights of the local data of agent i and the system administrator is a trusted party responsible for generating the required system parameters, the data aggregation can be described as,

$$\mathbf{x}(t) = \sum_{i=1}^{n} \mathbf{W}_i x_i(t) \tag{1}$$

Each agent $i \in [n]$ collects or performs calculations locally. At time t, the agent has access to its local data $\mathbf{x}_i(t)$(e.g., location, energy consumption, or gradient of the FL) and the aggregator holding numbers of computing devices, provides higher precision computations for the sake of interests.

3.2 Security Model

Inspired by [17], our proposed scheme can support the privacy requirement for private weighted sum aggregation. According to E.q.(1), we can categorize the private weighted aggregation into three types based on the knowledge possessed by agents and aggregators, i.e., the agents' private data, the weight, the intermediate results and the aggregation result.

Privacy-preserving Weighted Sum Aggregation with unknown weights(pWSAh):

a) Agent i: can not infer other agents' private data $\mathbf{x}_j(t), j \in [n]\backslash\{i\}$ and the aggregation result $\mathbf{x}(t)$ and the weights $\mathbf{W}_i, i \in [n]$, especially the result of $\mathbf{W}_i x_i(t)$.

b) The aggregator knew neither the agent's private data $\mathbf{x}_i(t)$ nor the weight $\mathbf{W}_i, i \in [n]$, including the intermediate result $\mathbf{W}_i x_i(t)$ except that compute $\mathbf{x}(t)$.

Privacy-preserving Sum Aggregation: In this case, we substitute from: a) Agent i knows its corresponding weight $\mathbf{W}_i$ and can't infer anything about the other agents' private data $x_j(t), j \in [n]\backslash i$, especially the partial information $\mathbf{W}_i x_i(t)$ and the aggregation result $\mathbf{x}(t)$.

Private Weighted Sum Aggregation with centralized weights: In this case, we modify b) slightly to the following: The aggregator knows the corresponding weight $\mathbf{W}_i$ of the agent and cannot infer anything else about the private data of the agent $x_i(t), i \in [n]$, including $\mathbf{W}_i x_i(t)$.

In this paper, we mainly focus on the problem of **pWSAh**, due to its privacy requirement of being the most stringent of the three cases. Furthermore, we consider a semi-honest adversary $\mathscr{A}$ and define that $\mathscr{A}$ cannot corrupt all agents at the same time. The honest agents' private data input and the intermediate results of pWSAh are prevented from the aggregator where the sum of secure aggregation results is revealed only to the target. It should be noted that the privacy requirements should even hold under the case where $\mathscr{A}$ corrupts the aggregator and up to $n-2$ agents.

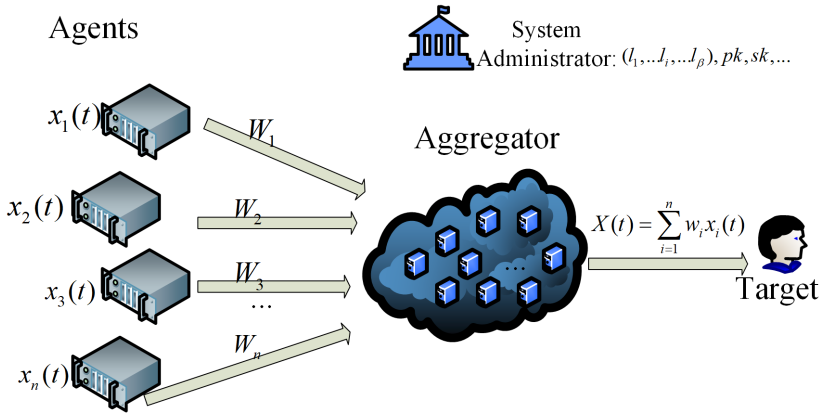


Fig. 1. Architecture of the three parties.

4. Proposed Scheme

In order to obtain a high-precision computation result under the devices with limited power. We will first design a privacy data mapping scheme for the real number that satisfies the ciphertext computation scalability, concerning a mapping and truncating scheme. Then, we present the basic privacy-preserving real number calculation protocols as the basis of constructing secure aggregation, involving secure addition and secure comparison protocols. The detail is as follows, $[[\cdot]]$ denotes the ciphertext of paillier. In particular, we give an example based on our proposed protocol throughout each section to aid understanding.

4.1 Mapping

In application, sensor nodes in the Industrial Internet of Things, financial data, or real-time data in power systems generally are rational numbers. For the sake of computation, while preserving the privacy of the agent's input, we first convert rational numbers to non-negative integers based on [29].

We first define the binary format of floating-point numbers in this paper as follows. Specifically, we specify p denote the number of bits of the real number, where the q bit is used for indicating the fractional.

$$b_p b_{p-1} \cdots b_{q+1} . b_q b_{q-1} \cdots b_1 \tag{2}$$

for given integers $p, q \in \mathbb{N}$ and $p > q$, $b_i \in \{0,1\}\ \forall\ i \in \{1,2,\cdots,p\}$. The set of all such numbers is said to be:

$$Q(p,q) := \{x \in Q | x = -2^{p-q-1} b_p + \sum_{i=1}^{p-1} 2^{i-q-1} b_i\}. \tag{3}$$

Therefore, the set $Q(p,q)$ describes all the rational numbers that can be expressed by the E.q.(2), i.e., from -2^{p-q-1} to $2^{p-q-1} - 2^{-q}$. According to E.q.(2), negative numbers are represented in complement form, which means that the subtraction can be regarded as the addition.

Instance: Given 5.25 and -1.125 for aggregation, the (2) of the private data corresponding is 0101.0100 and 1110.1110 for $p = 4$ and $q = 4$. After **Mapping**, the private data mapped is 01010100 and 11101110.

4.2 Truncating

As the main contribution of this paper, we propose the truncating algorithm based on significant bit splitting for integers by sending the block data that truncated data to different computing devices for a higher accuracy result. Different from the common preformation of converting large data into small data for computation (e.g., China's Residual Theorem, CRT), truncated data can not only support comparative operations without recovering the original data but also enable numerous devices with limited computing that can only process data with fewer significant bit to collaborate to compute a result with

a more significant bit, with the greater the number of devices the higher the precision. It stems from the fact that the performance of the device is preset when it is produced, but the number and complexity of tasks are becoming increasingly intense. Therefore, how to fully utilize the limited power but number of devices to obtain a high-precision calculation result is the focus of this paper.

Given the bits of the result required accuracy, l. Firstly, we truncate the non-negative integers with more bits into β block data, denoted by Δ_i, for boosting the devices with limited computing power to perform efficient calculations. For a plaintext data x, we have:

$$\begin{aligned} &\Delta_i = [x \bmod 2^{\sum_{i=1}^{i} l_i} - x \bmod 2^{\sum_{j=1}^{i-1} l_j}] \cdot 2^{-\sum_{j=1}^{i-1} l_j}, \\ &\sum_{i=1}^{\beta} l_i = l, (i \in \{1,2,...,\beta\}) \end{aligned} \tag{4}$$

Where, l_i denotes the number of bits of the block data Δ_i(l_j means too) i.e, the binary form of the decimal number 23 is 10111, which can be truncated into $\underbrace{1}_{\Delta_3}||\underbrace{01}_{\Delta_2}||\underbrace{11}_{\Delta_1}$. It is means that $l = l_3 + l_2 + l_1 = 1 + 2 + 2 = 5$. Then, the corresponding block data as follows: $Data = (23)_{10} = (10111)_2 = (\Delta_3||\Delta_2||\Delta_1) = (1||1||3)$.

Based on E.q.(5), it is possible to combine β devices that can only handle the data with only l_i bits to compute a result with l bits, wherein $\sum_{i=1}^{i=\beta} l_i = l$.

Instance: In this section, we assume that the data is truncated into tree blocks and the bit length of each is 2, 3, 3. That is $(01010100)_2 = (\Delta_3||\Delta_2||\Delta_1) = (010||101||00) = (2||5||0)$ and $(11101110)_2 = (\Delta_3||\Delta_2||\Delta_1) = (111||011||10) = (7||3||2)$

4.3 Secure comparison protocol C_{mp}

As a fundamental step in many data analysis algorithms, it is important to design secure and efficient privacy data comparison protocols. We give the design details of the secure comparison protocol in Algorithm 1.

The core of the secure comparison protocol is based on the fact that given two encrypted data under Paillier 's scheme $[[a]]$ and $[[b]]$, the algorithm 1 is to extract the encrypted bit $[[t]]$ that is the most significant bit of $[[a-b]]$ such that $(t = 1) \Leftrightarrow (a \leq b)$. In C_{mp}, after mapping in section 4.1, we get the non-negative numbers and the block data Δ_β contain the bit that indicates the sign of raw data, l_β-th bit in Δ_β. When $l_\beta = 1$, the block data corresponding raw data is negative and vice versa. Therefore, when $\Delta_\beta \geq 0$, $\Delta_\beta \in [0, 2^{l_\beta} - 1]$ and $\Delta_\beta < 0$, $\Delta_\beta \in [-1, 1 - 2^{l_\beta}]$.

Instance: In this case, we denote that $a = 5.25$ and $b = 1.125$, the block data containing the significant bit of the result of private data addition is $2 < 2^{l_3} - 1 = 2^3 - 1 = 7$ hold(Why the result is 2 will be given in the next section). Therefore, the compare result between 5.25 and -1.125 is 5.25 being larger.

4.4 Secure addition protocols for block data(SABD)

To realize the scalability of ciphertext computation that can compute high-precision aggregations cooperatively across multiple devices. In this section, we propose the secure

Algorithm 1 Secure comparison protocol C_{mp}

Input: $[[(\Delta)^{data}]]$, $[[(\Delta)^{DATA}]]$, pk

Output: Encrypted bit $[[t]]$: $t = 1 \Leftrightarrow (data \leq DATA)$

1: **Aggregator:** Choose $r_\beta \in \mathbb{Z}_{2^{p-1}}$, $\{r'_\beta \in \mathbb{Z}_{2^{p-1}} \backslash i' \cdot 2^{l_i-1}; i = 1, \cdots, \beta, i' \in \mathbb{Z}^+\}$ get $[[\Delta_\beta]] = [[(\Delta)^{data}_\beta - (\Delta)^{DATA}_\beta]]$, choose a random bit *RandomCoin* and set $Coin_0 = 0$
2: **Aggregator:** Compute: $R' = r'_\beta // 2^{l_\beta - 1}$ and get $[[R']]^{-1}$
3: **Aggregator:** Compute: $R = r'_\beta mod 2^{l_\beta - 1}$ and get $[[R]]^{-1}$
4: **Aggregator:** Send $[[\Delta_{\beta'}]] = [[r'_\beta]] \cdot [[\Delta_\beta]] \cdot [[2^\beta]]^{-1} \cdot [[Coin_{i-1}]]$ to *Target*
5: **Target:** Decrypt: $[[\Delta_{\beta'}]]$
6: **Target:** Compute: $\gamma = \Delta_{\beta'} \bmod 2^{l_\beta - 1} = (\Delta_\beta + r'_\beta) \bmod 2^{l_\beta - 1}$
7: **Target:** Compute: $\gamma' = \Delta_{\beta'} // 2^{l_\beta - 1} = (\Delta_\beta + r'_\beta) // 2^{l_\beta - 1}$
8: **Target:** Send $[[Coin_{\beta-1}]]$, $[[\gamma]]$, $[[\gamma']]$ to *Cloud*
9: **Aggregator:**
10: **if** $RandomCoin = 1$ **then**
11: Set $[[AAA]] = [[R]]$
12: Set $[[BBB]] = [[\gamma]]$
13: **else**
14: Set $[[AAA]] = [[\gamma]]$
15: Set $[[BBB]] = [[R]]$
16: **end if**
17: **Aggregator:** Send: $[[AB]] = [[AAA]] \cdot [[BBB]]^{-r_\beta}$ to *Target*
18: **Target:** Decrypt: $[[AB]]$,
19: **if** $AB > 0$ **then**
20: Set $RE = -1$
21: **else**
22: Set $RE = 0$
23: **end if**
24: **Target:** Send $[[RE]]$ to *Cloud*
25: **Aggregator:**
26: **if** *RandomCoin* = 1 **then**
27: Set $[[RE]] = [[RE]]$
28: **else**
29: Set $RE = [[RE]]^{-1} \cdot [[1]]$
30: **end if**
31: **Aggregator:** Compute: $[[t]] = [[\gamma']] \cdot [[R']]^{-1} \cdot [[RE]]$

addition protocol *SABD* for block data in Algorithm 2, which defines the addition of the truncated data to achieve secure aggregation.

Algorithm 2 Secure addition protocols for block data(**SABD**)

Input: $[[(\Delta)^{data}]]$, $[[(\Delta)^{DATA}]]$

Output: The result of $[[data+DATA]]$

1: **Aggregator:** Choose $\mathbf{r} := \{r | r_i \in \mathbb{Z}_{2^{p-1}}, i = 1, \cdots, \beta\}$, $\mathbf{r}' := \{r' | r'_i \in \mathbb{Z}_{2^{p-1}} \backslash i' \cdot 2^{l_i - 1}; i = 1, \cdots, \beta, i' \in \mathbb{Z}^{+}\}$ get $[[\Delta_i] = [(\Delta)_i^{data}] \cdot [(\Delta)_i^{DATA}]]$, choose a random bit *RandomCoin* and set $t_{i-1} = 0$

2: **for** For $i = 1$ to β **do**

3: **Aggregator:** Compute: $R' = r'_i // 2^{l_i}$ and get $[[R']]^{-1}$

4: **Aggregator:** Compute: $R = r'_i mod 2^{l_i}$ and get $[[R]]^{-1}$

5: **Aggregator, Target:** Running Algorithm 1, the cloud gets $[[t_{i-1}]]$

6: **Aggregator:** Compute: $[[\Delta_i]] = [[\Delta_i^{data+DATA}]] \cdot (2^{l_i} [[Coin_{i-1}]]^{-1})$

7: **end for**

8: **Aggregator:** Computate the aggregation $\sum_{i=1}^{\beta} [[\Delta_i]]$

The main idea of the algorithm 2 is that given two block data ciphertexts $[[(\Delta)_{data}]]$ and $[[(\Delta)_{DATA}]]$ with the unencrypted data are l_i bits, the block data of the corresponding positions of the sum between data and DATA can be computed. More generally, considering the case where $(\mathbb{Z}_1 \times [[(\Delta)_{data}]]) \cdot (\mathbb{Z}_2 \times [[(\Delta)_{DATA}]])$, in which $\mathbb{Z}_{1,2} \in \mathbb{Z}^{+}$, we only need to replace 2^{l_i} with $\mathbb{Z}_1 \mathbb{Z}_2 2^{l_i}$ in *SABD*, leaving the rest steps unvaried.

Instance: Carrying on from the example in the previous section, we compare the $a = 5.25$ and $b = 1.125$, i.e., we compute $a - b$, which can be given abbreviated as follows. In which, the result of the block data Δ_2 is not less than $2^{l_2} = 2^3 = 8$, the forward one bit produces the carrying. Thus, the block data containing the most significant bit results is $2 = (2+7+1) mod 2^{l_3} = 10 mod 8 = 2$.

$$\begin{array}{rr} & 2||5||0 \\ + & 7||3||2 \\ \hline & 2||0||2 \end{array} \tag{5}$$

4.5 Recovery

In this section, we will recover the computation of mapped and truncated private data. It can be recovered by recursively weighting the sum of block data and dividing by 2^{-q} as shown in E.q.(6). Moreover, after the aggregator computes the block data of the data sum to be solved, the secure aggregation result $X(t)$ can be recovered by line (8) in algorithm *SABD*.

$$\begin{aligned} [[data+DATA]] &= 2^{-q} \cdot [[data(p,q)]][[DATA(p,q)]] \\ &= 2^{-q} \cdot \sum_{i=1}^{\beta} 2^{l_{i-1}} \cdot [[\Delta_i]], (2^{l_0} = 1, i = 1, \cdots, \beta) \end{aligned} \tag{6}$$

Instance: we recover the aggregation results as follows, $2^{-q} \cdot (2||0||2) = 2^{-4} \cdot (010||000||10) = 2^{-4} \cdot (01000010) = 0100.0010 = 4.125$.

5. Private Weighted Sum Aggregation

In this section, we will show how to apply the privacy-preserving scheme proposed in Section 4 to solve the **pWSAh** problem mentioned in Section 3.2 to achieve private weighted sum aggregation.

- $Setup(1^{\kappa}, T, w_{i\in\{1,\ldots,\beta\}}, (l, l_1, ..l_\beta), pk, sk)$:

  System Administrator: Input the security parameter κ, the period T, the precision l bits required for the aggregation, and the number of devices β owned by the aggregator. Output the public key pk and the number of bits for which the aggregation device should perform the computation that satisfies $\sum_{i=1}^{i=\beta} l_i = l$ for broadcast, secret key sk for target and the ciphertext $[[w_i]]$ for agent i.

- $Enc(pk, x_i(t), [[w_i]])$:

  Agents: take as input the public parameters, agent i's private data $x_i(t)$ in time step t which is the private data that performs the **Mapping** mentioned in Section 4.1 to convert a floating-point real number to a non-negative integer. Then, the mapped non-negative integer is truncated to block data $\Delta_i(i = 1, \cdots, \beta)$ based on **Truncating** in Section 4.2. Finally, the agent i computates the block data ciphertext $[[w_i]]^{\Delta_i}(i = 1, \cdots, \beta)$ and send it to the cloud offline.

- $Computing(\mathbf{x}(t))$:

  Aggregator, Target: running the algorithm **SABD** in Section 4.4 to get the private weighted sum aggregation result $\mathbf{x}(t)$. Furthermore, after φ iteration, running the algorithm C_{mp} in Section 4.3 to compare the ciphertext result with the ϑ, which is the maximum data that can be processed by the aggregator that consisting of multiple computing devices, to determine whether to continue with the computation.

- $Recover(\mathbf{x}(t))$:

  Target: perform the **Recovery** in Section 4.5 to get the aggregation $\mathbf{x}(t)$.

In the above algorithmic steps for solving the problem **pWSAh**, we focus on how to apply the scheme proposed in Section 4 for secure aggregation. It is worth noting that the proposed scheme can work for other application scenarios due to its ciphertext computational scalability that supports multi-device co-computation

6. Theoretical Analysis

6.1 Security Analysis

According to different privacy requirements in 3.2, $\mathscr{A}$ can corrupt different parties. We assume that the $\mathscr{M}$, which is a probabilistic polynomial time(PPT) simulator, can generate the computationally distinguished between the real view and ideal view for $\mathscr{A}$(e.g., the coalition of the cloud and some of the agents), then the transcript obtained by running our proposed scheme in private weighted sum aggregation is uniformly distributed

random number. It is worth noting that the security of privacy weighted sum aggregation procedure has been proved in detail in [17], and we will focus on the security of the secure aggregation concerning the schemes proposed in this paper, in particular, the secure addition protocol **SABD**, as well as the secure comparison protocol C_{mp} is the core component. Further, the security proof relies on the following.

Lemma 1. All the sub-protocols that consist of a protocol can be simulated perfectly, and then the protocol is perfectly simulatable.

Lemma 2. If a random element a is uniformly distributed on $\mathbb{Z}_p$ and independent from any variable $b \in \mathbb{Z}_p$, then $a \pm b$ is also uniformly random and independent from b.

Theorem 1. C_{mp} protocol is secure in semi-honest model.

Proof. The real view $View_{real}$ of cloud is $\{[[r_\beta]], [[r_\beta{}']], R, R', RandomCoin, [[\Delta_i]], [[\Delta_{\beta'}]], [[Coin_{\beta-1}]], [[\gamma]], [[\gamma']], [[RE]], [[t]]\}$, where r_β, $r_\beta{}'$ and $RandomCoin$ are randomly select from $\mathbb{Z}_p$. Since $\Delta_\beta \in \mathbb{Z}_p$, $\Delta_{\beta'}$ is random according to Lemma 2. Further, the simulator $\mathscr{M}$ can output the view for $\mathscr{A}$ by selecting the random data necessary to encrypt the inputs. Hence, $\mathscr{A}$ that does not have the decryption key cannot infer information about the $[\Delta_\beta]$ by simply computing between $[\gamma]$, $[\gamma']$ and $\Delta_{\beta'}$. For cloud corrupted with $\mathscr{A}$, $\mathscr{A}$ cannot infer to $[[\gamma]]$ and $[[\gamma']]$ for which the ciphertexts are refreshed after the target computed. Since $RandomCoin$, $[[Coin_{\beta-1}]]_{pk}$ and $[[RE]]$, the output $[[t]]$ of cloud is random. $\mathscr{M}$ can generate the simulatable view $View_{sim}$ for $\mathscr{A}$, which is computationally indistinguish between $View_{real}$ and $View_{sim}$. Meanwhile, $\mathscr{A}$ cannot distinguish $View_{real}$ and $View_{sim}$ that corrupted the target. Therefore, C_{mp} protocol is secure in the semi-honest model.

Theorem 2. SABD protocol is secure in our proposed scheme.

Proof. Being the essential compoent of the Algorithm **SABD**, C_{mp} has been shown to be secure in Theorem 1 and the output is uniformly random, and for all $[[\Delta_i]](i = 1, \cdots, \beta)$, it is a variable in $[0, 2^{l_i-1}]$, $\mathscr{M}$ can simulate the real view $View_{real}$ by choosing random numbers for encryption. The $View_{real}$ of *SCBD* protocol and $View_{sim}$ generated by $\mathscr{M}$ are computationally indistinguish for $\mathscr{A}$. Therefore, the *SCBD* protocol is secure in our proposed scheme.

Theorem 3. The private weighted sum aggregation used *Our proposed scheme* is secure in semi-honest.

Proof: Consider an iteration in private aggregation. Firstly, the Paillier cryptosystem is semantically secure, therefore, $\mathscr{A}$ with not the secret key can not learn any private information from any two ciphertexts. Secondly, a secret share of zero and the different random data that uniformly sampled from $\mathbb{Z}_p$ for each time step ensure the intermediate values had been blinded which is computationally indistinguishable for $\mathscr{A}$ corrupted with the cloud or the cloud, some of the agents or the target. Thirdly, based on Lemma 1 and Lemma 2, the simulator $\mathscr{M}$ can generate the view on the real inputs are computationally indistinguishable. Hence, after many rounds of iterative computation or large-scale computation, we can prove the security of the privacy weight sum aggregation scheme in the semi-honest model after using the proposed scheme.

6.2 Complexity Analysis

In this section, we compare the computational and communication complexity of our proposal with that of [15] in TABLE 1, where β is the number of blocks that are truncated

for mapping, and l denotes the number of bits that the device can process. Compared to [15], the computational and communication overhead of the proposed scheme increases linearly with the blocks while the bits that can be processed also increase, the increase is tolerable, and it also demonstrates that we can get results with higher accuracy by concentrating numbers of computationally weak devices for collaborative computation.

Table 1. Comparison of computational and communication complexity

Protocol	Computational	Communication	Range	Comparison
[15]	$O(1)$	$O(1)$	$O(l)$	No
C_{mp}	$O(3)$	$O(4)$	$O(l\beta)$	Yes
SCBD	$O(6\beta-1)$	$O(8\beta-1)$	$O(l\beta)$	Yes

6.3 Functional Analysis

In particular, to better demonstrate the functional differences between the proposed protocol and schemes with similar functionality, especially homomorphism-based federated learning schemes, we give TABLE 2 as follows.

Table 2. Comparison of the function

Protocol	Ciphertext Comparison	Distributed aggregation	Parallel computation for multiple devices
[15]	✗	✗	✗
[33]	✗	✗	✗
[35]	✗	✗	✗
Ours	✓	✓	✓

As shown in Table 2 above, compared to schemes such as homomorphic-based federated learning that focus on secure aggregation, the proposed protocol supports the joint execution of secure aggregation tasks on multiple devices, which is consistent with our aim of jointly utilising a large number of low-performance devices to accomplish complex aggregations that require complex aggregations. Meanwhile, to ensure that the scope of computation during the aggregation process does not exceed the capacity limit of the devices, we propose a homomorphic-based secure comparison operation, which is not only beneficial to extend the homomorphic ciphertexts to perform nonlinear computation tasks, but also helps to achieve the problem of limiting the number of computations of homomorphic ciphertexts. Next, we are going to show the practical overhead of the proposed protocol concerning the related comparison schemes in the next section.

6.3 Correctness Analysis

In this section, we will show the correctness of our proposed protocol. According to our proposed protocol, we aim to prove that $2^q \cdot w_i \cdot \sum_{i=1}^{i=n} x_i = w_i \cdot \sum_{i=1}^{i=n} \sum_{j=1}^{j=\beta} \Delta_{ij}$, that is $2^q \cdot$

$$\sum_{i=1}^{i=n} x_i = \sum_{i=1}^{i=n} \sum_{j=1}^{j=\beta} \Delta_{ij}.$$

$$\begin{aligned}
2^q \cdot \sum_{i=1}^{i=n} x_i &= \sum_{i=1}^{i=n} \sum_{j=1}^{j=\beta} \Delta_{ij} \\
&= \sum_{i=1}^{i=n} \sum_{k=0}^{l-1} 2^k \cdot b_{ik} \\
&= \sum_{i=1}^{i=n} \sum_{j=1}^{j=\beta} \sum_{k=0}^{k=l_j-1} b_{ijk} \cdot 2^{k+\sum_{u=l_0}^{u=l_j} u} \\
&= \sum_{i=1}^{i=n} \sum_{j=1}^{j=\beta} \Big(\sum_{k=0}^{k=l_j-1} 2^k \cdot b_{ijk} \Big) \cdot 2^{l_{j-1}} \cdot t_{j-1} \\
&= \sum_{i=1}^{i=n} (-2^{p-1} b_{ip} + \sum_{j=1}^{p-1} 2^{j-1} b_{ij})
\end{aligned} \tag{7}$$

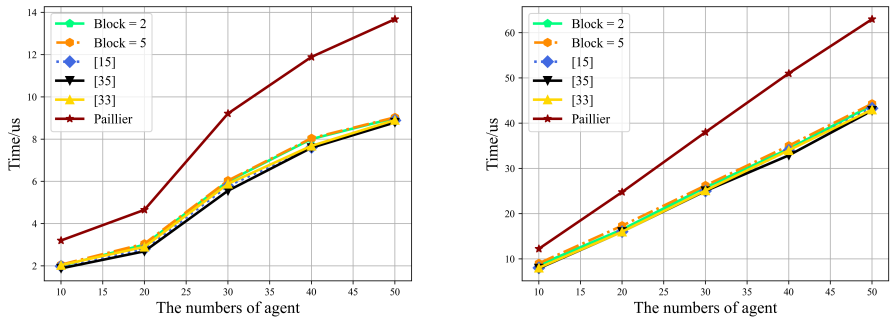
Wherein, $b_{ik}(b_{ij})$ denotes the k-th(j-th) bit of the binary bit string of the Mapped private data of the i-th client and b_{ijk} represents the k-th bit in the binary of the j-th block for client i. As shown in (7), the second equals indicate the binary form of raw private data, the third equals indicate the principle of addition between blocks of private data, while the fourth equals the aggregation under realistic distributed computing when carrying between block data is considered. These demonstrate the correctness of the proposed protocol.

7. Experimental Results

We mainly analyze the performance of the proposed scheme in private weighted sum aggregation from efficiency and accuracy. More specifically, we use a Raspberry Pi, which is a processor with CPU 1.5GHz, 4 GB RAM, and 500MHz GPU as computationally resource-constrained terminal equipment. Meanwhile, a laptop with Intel Core i7-7500 U, 8 GB RAM and a desktop with AMD Ryzen 7 processor and 16 GB RAM for experiment. In the simulation, the security parameter of length is 80 bits and the Paillier cryptosystem parameter, N, is set to 2048 bits. Furthermore, we specify the message representation to be on $l = 32$ bits, with 16 bits for integers and 16 bits for decimals and show the simulation for solving private sum aggregation as follows.

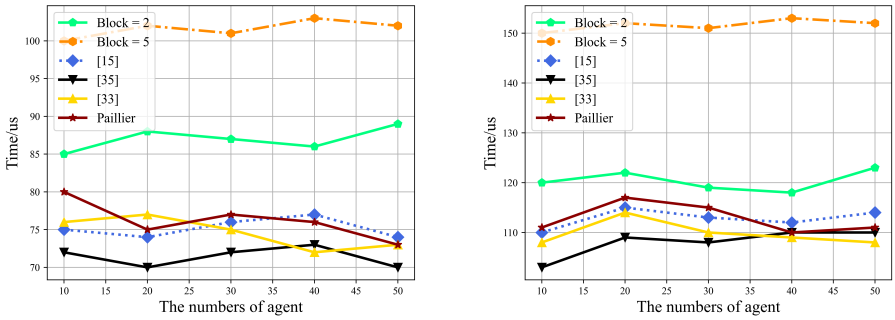
To better demonstrate the characteristics of the proposed protocols, we conducted comparative experiments in [15], [35], [33] and directly based on the encryption scheme Paillier cryptosystem. Among them, [35] and [33] are homomorphism-based federated learning aggregation schemes. As shown in Fig. 2., we do not count the training overhead in the homomorphism-based federated learning scheme for fairness judgement and focus on the overhead used for aggregation in the different schemes. It is worth noting that the running overhead of our proposed protocol is the parallel computation time of the devices, i.e., block = 2 means that the original private data is truncated into two chunks of data and computed by two different devices at the same time under the same power, which is different from the centralised computation on a single server in [15], [35], [33]. To ensure the correctness of the proposed protocol, our proposed protocol needs to perform carrying between block data, which leads to an increase in its computational overhead as the

number of blocks increases. Meanwhile, we note that the schemes other than the original Paillier-based scheme have optimised the encryption scheme to varying degrees, which leads to a significant improvement in the efficiency of ciphertext aggregation compared to the original Paillier system scheme. However, in this paper, we do not focus on accelerating the efficiency of the encryption system, so we can conclude that the running time of the aggregation is not much different in the case of the selected encryption scheme. Additionally, the difference in computing device capabilities has a greater impact on the efficiency of the secure aggregation scheme, as shown in Fig. 2(b). However, the increase in the overhead of the proposed protocol in terms of runtime is not significant compared to other schemes, the only increase is only the increase in the carry perforation between blocks of data, i.e., the addition between blocks of data however this is negligible.



(a) Running time on desktop (b) Running time on raspberry pi

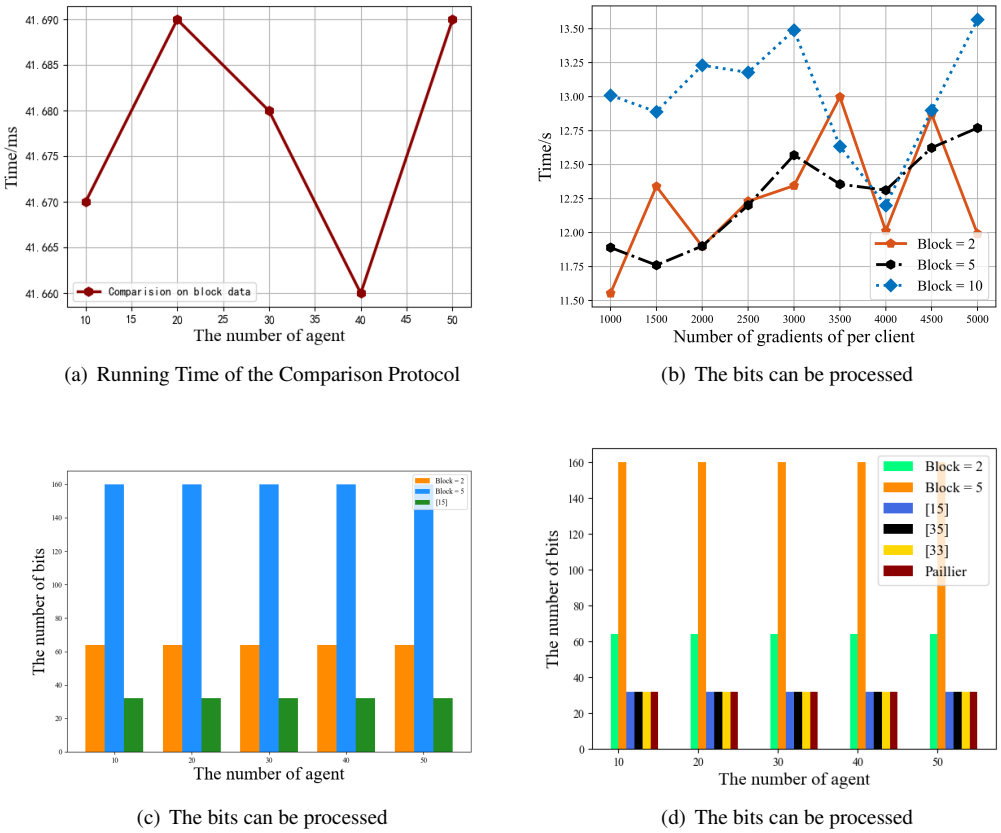
Fig. 2. Running time on different device



(a) Receive and transmit on desktop (b) Receive and transmit on raspberry pi

Fig. 3. Communication overhead on different device

Furthermore, we analyse the communication overhead of the proposed scheme and [15], [35], [33] and the directly based encryption scheme Paillier cryptosystem in Fig. 3. From Fig. 3(a) and (b), it can be seen that schemes other than our proposed scheme only



(a) Running Time of the Comparison Protocol

(b) The bits can be processed

(c) The bits can be processed

(d) The bits can be processed

Fig. 4. Running time on different Scheme

need to encrypt the data and then fully delegate it to the server, which is different from our proposed scheme, which requires the necessary communication between computing devices. Therefore, we have the highest communication overhead. Meanwhile, due to the packing and batch computation of ciphertexts used in [15], [35], [33], this has a more obvious advantage in terms of communication compared to schemes based directly on the Paillier cryptosystem. From the experimental data, it can be obtained that as the number of blocks in the proposed protocol increases, this means that more computing devices are needed to complete the task, and therefore the communication time is the longest, i.e., the communication cost of 'block = 5'. Meanwhile, in conjunction with Fig. 3(b), it can be seen that the more data processing power of the computing devices, the lower the communication overhead. It is worth noting that the communication overhead of the proposed protocol can be greatly reduced when implementing joint multi-device centralised execution of computing tasks. Meanwhile, the increased communication overhead of the proposed protocol is acceptable.

To fully evaluate the efficiency of the proposed scheme, we simulated the secure

comparison protocol on a desktop computer with a different number of agents as shown in Fig. 4(a). Unlike computational and communication protocols, in the safe comparison protocol, we only need to perform operations on Δ_β containing sign bits based on the fact that the comparison result satisfies $\Delta_\beta < 2^{l_\beta - 1}$. The efficiency of the secure comparison protocol is demonstrated by the fact that it requires only 41us compared to a computational protocol that runs for about 0.2 milliseconds per round. Further, to analyse the performance of the proposed protocol with different aggregators and different numbers of clients, we simulate the proposed protocol with different numbers of gradients at 2,5 and 10 aggregators and clients respectively as shown in Fig. 4(b). There is no significant difference in the overhead due to distributed parallel computation and the biggest difference is in the overhead added by the rounding operation between the block data in the last step. Meanwhile, we show in Fig. 4(c)(d) the range of data that can be processed by the computing device, which is still limited even with more devices compared to the scalable computation of protocols other than the proposed protocol, which does not take into account the ciphertext. In contrast, as the number of devices increases, the range of data that can be processed based on the proposed protocol increases. In other words, as the number of computing devices increases, the computable range increases linearly.

As discussed above, we can see that the private weighted sum aggregation based on our proposed scheme, despite a trivial increase in computational and communication overheads, expands the range of data that can be processed significantly, which is consistent with our original intention i.e., to obtain more accurate results by integrating existing computational resources.

8. Conclusion

In this paper, we proposed a private weighted sum aggregation approach aiming to fully exploit the computational resources of existing devices to obtain more accurate computational results. Next, we will extend our proposed scheme to support more complex computations to be applied in more domains.

ACKNOWLEDGMENT

The work has been supported by the National Key Research and Development Program of China under Grant 2021YFB3101100; National Natural Science Foundation of China under Grant 62272123; Project of High-level Innovative Talents of Guizhou Province under Grant [2020]6008; Science and Technology Program of Guizhou Province under Grant [2020]5017, [2022]065; the Guizhou University Talent Fund (2022) No.21; Open project of the State Key Laboratory of Public Big Data under Grant PBD2023-23.

REFERENCES

1. Ren H R, Ma H, Li H Y, et al. A disturbance observer based intelligent control for nonstrict-feedback nonlinear systems[J]. Science China Technological Sciences, 2023, 66(2): 456-467.

2. Babel K, Daian P, Kelkar M, et al. Clockwork finance: Automated analysis of economic security in smart contracts[C]//2023 IEEE Symposium on Security and Privacy (SP). IEEE, 2023: 2499-2516.
3. Alexandru A B, Burbano L, Çeliktuğ M F, et al. Private Anomaly Detection in Linear Controllers: Garbled Circuits vs. Homomorphic Encryption[C]//2022 IEEE 61st Conference on Decision and Control (CDC). IEEE, 2022: 7746-7753.
4. Zheng H, You L, Hu G. A novel insurance claim blockchain scheme based on zero-knowledge proof technology[J]. Computer Communications, 2022, 195: 207-216.
5. Degue K H, Le Ny J. Cooperative Differentially Private LQG Control With Measurement Aggregation[J]. IEEE Control Systems Letters, 2022, 7: 1093-1098.
6. Böhler J, Kerschbaum F. Secure multi-party computation of differentially private heavy hitters[C]//Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security. 2021: 2361-2377.
7. Fereidooni H, Marchal S, Miettinen M, et al. SAFELearn: Secure aggregation for private federated learning[C]//2021 IEEE Security and Privacy Workshops (SPW). IEEE, 2021: 56-62.
8. Dong Y, Li Z, Zhao X, et al. Decentralised and cooperative control of multi-robot systems through distributed optimisation[J]. arXiv preprint arXiv:2302.01728, 2023.
9. Van Cutsem O, Dac D H, Boudou P, et al. Cooperative energy management of a community of smart-buildings: A Blockchain approach[J]. International Journal of electrical power & energy systems, 2020, 117: 105643.
10. Xiang K, Tian Y L, Gao S, et al. Game-based Theory Rational Delegation Learning Scheme[J]. Journal of Information Science & Engineering, 2022, 38(1).
11. Shi X, Xu Y, Chen G, et al. An Augmented Lagrangian-based Safe Reinforcement Learning Algorithm for Carbon-Oriented Optimal Scheduling of EV Aggregators[J]. IEEE Transactions on Smart Grid, 2023.
12. Zhang Z, Deng R, Tian Y, et al. SPMA: Stealthy Physics-Manipulated Attack and Countermeasures in Cyber-Physical Smart Grid[J]. IEEE Transactions on Information Forensics and Security, 2023, 18: 581–596.
13. Niu K, Peng C, Tian Y, et al. K-Implicit Tracking Data Publishing Scheme Against Geo-Matching Attacks[J]. Journal of Information Science & Engineering, 2022, 38(1).
14. Rathee M, Shen C, Wagh S, et al. Elsa: Secure aggregation for federated learning with malicious actors[C]//2023 IEEE Symposium on Security and Privacy (SP). IEEE, 2023: 1961-1979.
15. Young M, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.
16. Fan L, Xiong L. An adaptive approach to real-time aggregate monitoring with differential privacy[J]. IEEE Transactions on knowledge and data engineering, 2013, 26(9): 2094-2106.
17. Alexandru A B, Pappas G J. Private weighted sum aggregation[J]. IEEE Transactions on Control of Network Systems, 2021, 9(1): 219-230.
18. Zhang L, Xu J, Vijayakumar P, et al. Homomorphic encryption-based privacy-preserving federated learning in iot-enabled healthcare system[J]. IEEE Transactions on Network Science and Engineering, 2022.

19. Luo X, Xue K, Xu J, et al. Blockchain based secure data aggregation and distributed power dispatching for microgrids[J]. IEEE Transactions on Smart Grid, 2021, 12(6): 5268-5279.
20. Regueiro C, Seco I, de Diego S, et al. Privacy-enhancing distributed protocol for data aggregation based on blockchain and homomorphic encryption[J]. Information Processing & Management, 2021, 58(6): 102745.
21. Bindschaedler V, Rane S, Brito A E, et al. Achieving differential privacy in secure multiparty data aggregation protocols on star networks[C]//Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy. 2017: 115-125.
22. Wu X, Zhang Y, Shi M, et al. An adaptive federated learning scheme with differential privacy preserving[J]. Future Generation Computer Systems, 2022, 127: 362-372.
23. Wang S, Luo X, Qian Y, et al. Shuffle differential private data aggregation for random population[J]. IEEE Transactions on Parallel and Distributed Systems, 2023, 34(5): 1667-1681.
24. Alanwar A, Shoukry Y, Chakraborty S, et al. PrOLoc: Resilient localization with private observers using partial homomorphic encryption[C]//Proceedings of the 16th ACM/IEEE International Conference on Information Processing in Sensor Networks. 2017: 41-52.
25. Merad-Boudia O R, Senouci S M. An efficient and secure multidimensional data aggregation for fog-computing-based smart grid[J]. IEEE Internet of Things Journal, 2020, 8(8): 6143-6153.
26. Tjell K, Wisniewski R. Private aggregation with application to distributed optimization[J]. IEEE Control Systems Letters, 2020, 5(5): 1591-1596.
27. Park J, Lim H. Privacy-preserving federated learning using homomorphic encryption[J]. Applied Sciences, 2022, 12(2): 734.
28. Zhang Z, Che X, Jiao X, et al. Quadratic Optimization Using Additive Homomorphic Encryption in CPS[C]//2022 13th Asian Control Conference (ASCC). IEEE, 2022: 1995-2000.
29. McMahan B, Moore E, Ramage D, et al. Communication-efficient learning of deep networks from decentralized data[C]//Artificial intelligence and statistics. PMLR, 2017: 1273-1282..
30. Wang H, Zhang Y, Cheng Y, et al. A Data Privacy Protection Scheme Integrating Federal Learning and Secret Sharing[C]//2023 IEEE 5th International Conference on Power, Intelligent Computing and Systems (ICPICS). IEEE, 2023: 311-315..
31. Wei K, Li J, Ding M, et al. Federated learning with differential privacy: Algorithms and performance analysis[J]. IEEE transactions on information forensics and security, 2020, 15: 3454-3469..
32. Hardy S, Henecka W, Ivey-Law H, et al. Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption[J]. arxiv preprint arxiv:1711.10677, 2017.
33. Li T, Sahu A K, Talwalkar A, et al. Federated learning: Challenges, methods, and future directions[J]. IEEE signal processing magazine, 2020, 37(3): 50-60.
34. Yang S, Chen Y, Tu S, et al. A post-quantum secure aggregation for federated learning[C]//Proceedings of the 2022 12th International Conference on Communication and Network Security. 2022: 117-124.

35. Zhang C, Li S, Xia J, et al. BatchCrypt: Efficient homomorphic encryption for Cross-Silo federated learning[C]//2020 USENIX annual technical conference (USENIX ATC 20). 2020: 493-506.
36. Jin W, Yao Y, Han S, et al. FedML-HE: An Efficient Homomorphic-Encryption-Based Privacy-Preserving Federated Learning System[J]. arXiv preprint arXiv:2303.10837, 2023.
37. Fang H, Qian Q. Privacy preserving machine learning with homomorphic encryption and federated learning[J]. Future Internet, 2021, 13(4): 94.
38. Zhang J, Chen B, Yu S, et al. PEFL: A privacy-enhanced federated learning scheme for big data analytics[C]//2019 IEEE Global Communications Conference (GLOBECOM). IEEE, 2019: 1-6.
39. Jiang Z, Wang W, Liu Y. Flashe: Additively symmetric homomorphic encryption for cross-silo federated learning[J]. arXiv preprint arXiv:2109.00675, 2021.
40. Du W, Li M, Wu L, et al. A efficient and robust privacy-preserving framework for cross-device federated learning[J]. Complex & Intelligent Systems, 2023, 9(5): 4923-4937.
41. Eltaras T, Sabry F, Labda W, et al. Efficient verifiable protocol for privacy-preserving aggregation in federated learning[J]. IEEE Transactions on Information Forensics and Security, 2023, 18: 2977-2990.
42. Li Y, Lai J, Zhang R, et al. Secure and efficient multi-key aggregation for federated learning[J]. Information Sciences, 2024, 654: 119830.
43. Pham C H, Huynh-The T, Sedgh-Gooya E, et al. Extension of physical activity recognition with 3D CNN using encrypted multiple sensory data to federated learning based on multi-key homomorphic encryption[J]. Computer Methods and Programs in Biomedicine, 2024, 243: 107854.
44. Wang Y, Feng Y, Xiao Y, et al. Privacy Block-Streaming: A Novel DEX File Loading Scheme Based on Federated Learning[J]. Journal of Information Science & Engineering, 2024, 40(1).