# Optimal Task Allocation and Coding Design for Secure Edge Computing With Heterogeneous Edge Devices

Jin Wang, *Member, IEEE*, Chunming Cao, Jianping Wang, *Member, IEEE*, Kejie Lu, *Senior Member, IEEE*, Admela Jukan, *Member, IEEE*, and Wei Zhao, *Fellow, IEEE*

**Abstract**—In recent years, edge computing has attracted significant attention because it can effectively support many delay-sensitive applications. Despite such a salient feature, edge computing also faces many challenges, especially for efficiency and security, because edge devices are usually heterogeneous and may be untrustworthy. To address these challenges, we propose a unified framework to provide efficiency and confidentiality by coded distributed computing. Within the proposed framework, we use matrix multiplication, a fundamental building block of many distributed machine learning algorithms, as the representative computation task. To minimize resource consumption while achieving information-theoretic security, we investigate two highly-coupled problems, (1) task allocation that assigns data blocks in a computing task to edge devices and (2) linear code design that generates data blocks by encoding the original data with random information. Specifically, we first theoretically analyze the necessary conditions for the optimal solution. Based on the theoretical analysis, we develop an efficient *task allocation* algorithm to obtain a set of selected edge devices and the number of coded vectors allocated to them. Using the task allocation results, we then design *secure coded computing* schemes, for two cases, (1) with redundant computation and (2) without redundant computation, all of which satisfy the availability and security conditions. Moreover, we also theoretically analyze the optimization of the proposed scheme. Finally, we conduct extensive simulation experiments to demonstrate the effectiveness of the proposed schemes.

**Index Terms**—Edge computing, efficiency, confidentiality, coded computing, task allocation, linear coding, optimization

## 1 Introduction

During the past few years, edge computing has become a viable solution to support many delay-sensitive applications, such as Internet-of-Things (IoT), virtual/augmented/mixed reality (VR/AR/MR), crowdsourcing, machine learning, and big data analytics [2] (Fig. 1a). Instead of executing on a remote data center, in edge computing, a computing task of a user device can be distributed and then executed in multiple nearby edge devices. Therefore, the completion time of the computing task can be significantly reduced [3], [4], [5], [6], [7], [8] (Fig. 1b). Moreover, a large computation task can be partitioned into smaller sub-tasks with certain redundancy and then executed on multiple edge devices to further reduce the total task completion time [9], [10], [11].

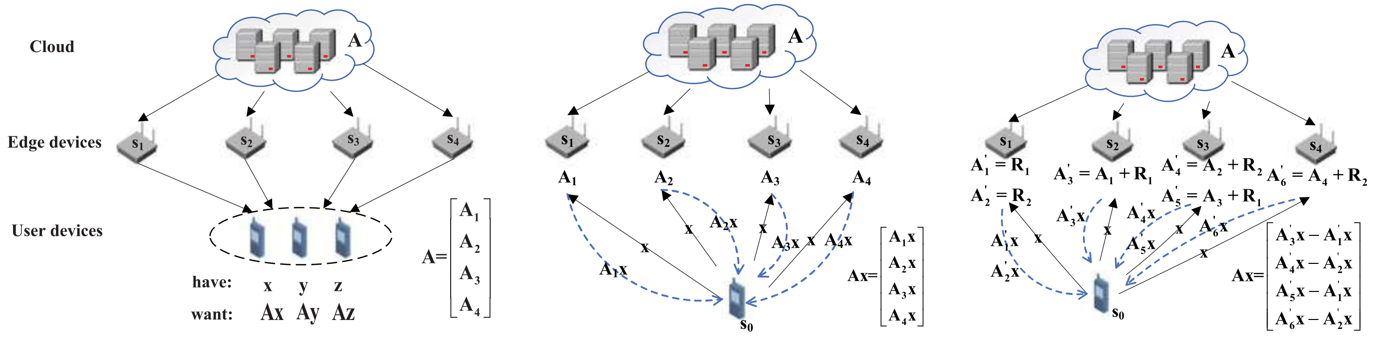
Although such traditional distributed computing schemes can be utilized in edge computing, there are still many challenges to be addressed in practical scenarios. First, many edge devices are heterogeneous and resource-limited, i.e., limited storage space, computing capability, and bandwidth. Therefore, it is important to design optimal task allocation schemes to identify a set of suitable edge devices for computing. Second, edge devices may be untrustworthy, so it is necessary to design security mechanisms to provide data confidentiality in edge computing.

To address both challenges, *coded distributed computing* (CDC) has been proposed in recent years and applied to perform different distributed computing tasks, in which the dominating use case is matrix multiplication [1], [4], [5], [6], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25] because it is a critical and indispensable building block of many distributed machine learning algorithms, e.g., linear regression [4], k-nearest neighbors estimation [17], deep neural network [18], convolution neural network [19], federated learning [20], etc.

For matrix multiplication, most existing studies on CDC focus on the tradeoff between the latency and computing resources [4], [5], [6], [7], but very few efforts have been devoted to the security aspects by fully utilizing linear coding with resource consumption consideration. For instance, in [10],

- *Jin Wang is with the Department of Computer Science and Technology, Soochow University, Suzhou 215006, China. E-mail: wjin1985@suda.edu.cn.*
- *Chunming Cao is with the Department of Computer Science and Technology, Soochow University, Suzhou 215006, China, and also with the China Mobile (Suzhou) Software Technology Company, Ltd., Suzhou 215000, China. E-mail: 20175227063@stu.suda.edu.cn.*
- *Jianping Wang is with the Department of Computer Science, City University, Hong Kong, China. E-mail: jianwang@cityu.edu.hk.*
- *Kejie Lu is with the Department of Computer Science and Engineering, University of Puerto Rico at Mayagüez, Puerto Rico 00682 USA. E-mail: kejie.lu@upr.edu.*
- *Admela Jukan is with the Department of Electrical Engineering, Information Technology, Physics, Technische Universität Carolo-Wilhelmina zu Braunschweig, 38106 Braunschweig, Germany. E-mail: a.jukan@tu-bs.de.*
- *Wei Zhao is with the CAS Shenzhen Institute of Advanced Technology, Shenzhen 518055, China. E-mail: wzhao@aus.edu.*

*Manuscript received 28 May 2020; revised 26 Nov. 2020; accepted 19 Dec. 2020. Date of publication 8 Jan. 2021; date of current version 6 Dec. 2022.*
*(Corresponding author: Jin Wang.)*
*Recommended for acceptance by X. Deng.*
*Digital Object Identifier no. 10.1109/TCC.2021.3050012*



(a) Edge Computing Networks.
(b) Traditional Edge Computing.
(c) Secure Coded Edge Computing.

Fig. 1. An example of coded edge computing.

[11], [12], the authors utilized the random information and the redundant computation resource to provide *information-theoretic security* (ITS) without considering the communication, computation, and storage costs. In this paper, we address the design of secure CDC for edge computing with heterogeneous edge devices to *minimize the total resource usage*, which has not yet been investigated in the literature.

In the literature, homomorphic encryption can be exploited to compute directly on the encrypted data, but it requires high computation overhead and implementation complexity [25], [26], [27]. Specifically, using the latest HElib library developed in 2018, the authors in [27] demonstrated that the running time of multiplying a matrix by a vector in homomorphic encryption mode is more than $10^3$ times slower than the amount of time to directly multiply two unencrypted matrices. Therefore, homomorphic encryption may not be efficient for edge computing, especially for matrix multiplication. In this paper, we consider the secure coded edge computing by fully exploiting the properties of the linear coding itself, which has lower computation complexity.

Specifically, we consider a matrix multiplication model $\mathbf{Ax}$ in which the data matrix $\mathbf{A}$ is pre-defined in the cloud and coded blocks of $\mathbf{A}$ are disseminated to edge devices in advance [10], [11], [12], [13], [28], as shown in Fig. 1. Moreover, we aim to achieve the confidentiality of $\mathbf{A}$ such that the coded blocks assigned to each computing device cannot be used to compute any linear combination of rows in $\mathbf{A}$, which is the ITS requirement. To achieve the ITS of $\mathbf{A}$, the cloud can generate some random blocks and linearly combine them with the blocks in $\mathbf{A}$, as shown in Fig. 1c. Although adding random blocks will lead to more resource usage, the confidentiality of the data matrix $\mathbf{A}$ is provided without sharing any secret keys. Since edge devices are resource-limited, the number of coded blocks processed in each edge device must also be limited. Therefore, we will investigate two highly-coupled problems, task allocation and coding design. Next, we will use an example to show that the task allocation and coding design have significant impacts on the performance of security and the total usage of storage, computation, and communication resources.

In the following example, we consider that a user device $s_0$ wants to multiply *data matrix* $\mathbf{A}$ by its *input vector* $\mathbf{x}$ (or $\mathbf{y}$ or $\mathbf{z}$), and four edge devices can be selected to participate in edge computing. Specifically, suppose that the data matrix $\mathbf{A}$ contains 4 row-vectors, represented as $\{\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3, \mathbf{A}_4\}$. The cloud generates $r$ random row-vectors, represented as $\{\mathbf{R}_1, \mathbf{R}_2, \dots, \mathbf{R}_r\}$, encodes them with row-vectors of $\mathbf{A}$ into $4 + r$ coded row-vectors, and distributes them to the selected edge devices. After the selected edge devices compute and send intermediate results back to $s_0$, $s_0$ can decode $\mathbf{Ax}$.

To simplify the discussions about heterogeneous edge devices, we consider that (1) a unit resource is required to store and process one coded row-vector on each edge device, and (2) each computing device has a different resource limit, which is the maximum number of coded row-vectors that it can store and process. Moreover, the number of intermediate results transmitted to $s_0$ from each edge device is equal to the number of coded row-vectors stored on it. To use one unit resource on different edge devices may involve different costs. Thus, we model the total cost involved on an edge device as the product of unit cost and the number of coded row-vectors stored and computed on that device. Consequently, the total cost of the entire matrix multiplication is the summation of the total cost of all selected edge devices. Table 1 shows the total cost under different task allocations and coding schemes.

- In Case 1, $r = 1$ and three devices are selected to process the task. Case 1 is *valid* because the number of coded row-vectors allocated to each edge device does

TABLE 1
Examples for Task Allocation and Coding Scheme in Secure Coded Edge Computing

| Edge device | $s_1$ | $s_2$ | $s_3$ | $s_4$ | Total cost |
|---|---|---|---|---|---|
| Unit cost | 2 | 3 | 4 | 10 | |
| Resource limit | 4 | 1 | 3 | 2 | |
| Case 1: $r = 1$ | $\mathbf{A}_1 + \mathbf{R}_1$, $\mathbf{A}_1 + \mathbf{A}_2 + \mathbf{R}_1$ | $\mathbf{A}_3 + \mathbf{R}_1$ | $\mathbf{A}_4 + \mathbf{R}_1$, $\mathbf{R}_1$ | | 15, valid, **unsecure** |
| Case 2: $r = 2$ | $\mathbf{R}_1$, $\mathbf{R}_2$ | $\mathbf{A}_1 + \mathbf{R}_1$, $\mathbf{A}_2 + \mathbf{R}_2$ | $\mathbf{A}_3 + \mathbf{R}_1$, $\mathbf{A}_4 + \mathbf{R}_2$ | | 18, **invalid**, secure |
| Case 3: $r = 2$ | $\mathbf{R}_1$, $\mathbf{R}_2$ | $\mathbf{A}_1 + \mathbf{R}_1$ | $\mathbf{A}_2 + \mathbf{R}_2$, $\mathbf{A}_3 + \mathbf{R}_1$ | $\mathbf{A}_4 + \mathbf{R}_2$ | **25**, valid, secure |
| Case 4: $r = 3$ | $\mathbf{R}_1$, $\mathbf{R}_2$, $\mathbf{R}_3$ | $\mathbf{A}_1 + \mathbf{R}_1$ | $\mathbf{A}_2 + \mathbf{R}_1$, $\mathbf{A}_3 + \mathbf{R}_2$, $\mathbf{A}_4 + \mathbf{R}_3$ | | **21**, valid, secure |

not exceed its resource limit and the user device $s_0$ can decode $\mathbf{Ax}$ after receiving all the intermediate results. The total cost is $2 \times 2 + 1 \times 3 + 2 \times 4 = 15$. However, it is *unsecure* because $s_1$ can obtain $\mathbf{A}_2$ by $(\mathbf{A}_1 + \mathbf{A}_2 + \mathbf{R}_1) - (\mathbf{A}_1 + \mathbf{R}_1)$ and $s_3$ can obtain $\mathbf{A}_4$ by $(\mathbf{A}_4 + \mathbf{R}_1) - (\mathbf{R}_1)$.

- Case 2 was first investigated in our previous work [1] for homogeneous devices, in which each edge device has no resource limit and is allocated the same number of coded row-vectors. Case 2 is *secure* with a higher total cost $2 \times 2 + 2 \times 3 + 2 \times 4 = \mathbf{18}$. However, it is *invalid* for this example, because the number of coded row-vectors allocated to $s_2$ exceeds its resource limit, which is 1.
- In Case 3, the cloud generates the same coded row-vectors as those in Case 2 but it distributes the vectors to four edge devices as shown in Fig. 1c. It is *valid* and *secure*. The total cost is $2 \times 2 + 1 \times 3 + 2 \times 4 + 1 \times 10 = \mathbf{25}$.
- In Case 4, the first three edge devices are selected and three random vectors are used. Case 4 is also *valid* and *secure*. The total cost is $3 \times 2 + 1 \times 3 + 3 \times 4 = \mathbf{21}$. Compared to the cost in Case 3, the total cost in this case decreases by **16** percent.

The examples above clearly demonstrate that the task allocation and coding design should be jointly considered to minimize the total cost for *Secure Coded Edge Computing* (SCEC) with heterogeneous edge devices. In this paper, we formulate an optimization problem to *minimize the total resource usage* in SCEC with *ITS guarantee* by *jointly studying* the *task allocation* and *coding design*. Our objectives include: 1) completing the computation task, 2) satisfying the resource and security requirements, and 3) minimizing the total cost of storage, computation, and communication. To the best of the authors' knowledge, no previous work has been conducted to address such a *Minimum Cost SCEC* (MCSCEC) problem. The main contributions of the paper are summarized as follows:

- We adopt linear coding to achieve *secure edge computing* by exploiting the available resources of massive edge devices in edge networks. To this end, we formally define the *Minimum Cost Secure Coded Edge Computing* (MCSCEC) problem for the heterogeneous edge computing system.
- We conduct a solid theoretical analysis to first show the conditions for the existence of the feasible solution. We further prove the necessary conditions for the optimal solution, which enables us to further design the optimal task allocation scheme.
- To achieve the first two objectives of the MCSCEC problem, we develop an efficient optimal algorithm to first obtain a set of selected edge devices, i.e., task allocation, and then design coded computing scheme, i.e., coding design. Moreover, we also prove that the cost achieved by the proposed scheme is the minimum.
- We conduct extensive simulation experiments to demonstrate the effectiveness of the proposed task allocation and code design.

A preliminary version of this paper appears as [1]. Compared with our previous work, the main differences are shown as follows:

First, in this paper, we consider the MCSCEC problem for a heterogeneous edge computing network where the number of vectors stored on each edge device is limited and the limits of various devices may be different, which is more general than the case of a homogeneous edge network considered in [1].

Second, redundant coding can greatly reduce the impact of stragglers on computational latency because the user device does not need to receive all the intermediate results, but only need to receive enough intermediate results to decode [4], [10], [11], [12], [14], [19], [21]. In this paper, we consider a redundant coding scheme, which is more general than the case of the *unredundant* coding scheme considered in [1].

Third, we substantially enhance the performance evaluation in that we not only consider the heterogeneous resource consumption costs, but also the heterogeneous resource limits of different edge devices. In particular, we consider two probability distributions, i.e., the uniform distribution and the normal distribution, for the resource consumption costs and resource limits. On the other hand, we only evaluate the proposed algorithm when the resource consumption costs follow the uniform distribution in [1].

Finally, we add a new section to discuss related work from different aspects.

The rest of the paper is organized as follows. In Section 2, we first introduce the system model for the MCSCEC problem. We then provide a theoretical analysis of the MCSCEC problem in Section 3. Next, in Section 4, we design an efficient optimal scheme that includes task allocation algorithm and secure code designs for both the non-redundancy case and the redundancy case. We further conduct comprehensive simulations in Section 5. Finally, we investigate related work from multiple aspects in Section 6 and conclude the paper in Section 7.

## 2 Problem Modeling

In this section, we first introduce the SCEC model and then present the attack model considered in this paper. At last, we give the formal definition of the MCSCEC problem and provide an overview of the framework solving the problem.

### 2.1 System Model

In this paper, we study an edge computing system $\mathbf{S} = \{s_0, s_1, \dots, s_k\}$, $k \geq 2$, in which $s_0$ denotes a user device and $s_j, \forall j \in \{1, \dots, k\}$, represents the $j$th edge device. Let $w_j$ denotes the resource limit of the number of vectors stored on $s_j$, and let $\mathbf{W} = (w_1, \dots, w_k)$ be the sequence of limits. $s_0$ needs to perform computations on a confidential data set represented by an $m \times l$ dimensional matrix $\mathbf{A}$. Let $\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_m$ be $m$ row-vectors of $\mathbf{A}$, each of which is with dimension $1 \times l$. In our study, without loss of generality, we focus on the multiplication of data matrix $\mathbf{A}$ with one input vector $\mathbf{x}$. We assume that vector $\mathbf{x}$ is also a coded version of the original data, which cannot be used by any edge device to reveal the original data. In this paper, since the security of all data is rather comprehensive, we will only focus on how to achieve the confidentiality of $\mathbf{A}$. The scheme proposed in this paper can also be applied to more general cases that require multiplication of two matrices and multiplication of a data matrix with different input vectors.

To compute $\mathbf{A}\mathbf{x}$ and achieve the ITS requirement, $\mathbf{A}$ needs to be divided into blocks, coded, and stored at edge devices. This pre-processing can be done by a cloud, e.g., a parameter server that has trained a deep-learning model. Specifically, in the encoding process, we first generate a set of $r$ random vectors $\mathbf{R} = \{\mathbf{R}_1, \mathbf{R}_2, \ldots, \mathbf{R}_r\}$, in which each vector has a dimension $1 \times l$. Then, we generate a total of $a(m+r)$ coded vectors, each of which is a linear combination of $\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_m, \mathbf{R}_1, \mathbf{R}_2, \ldots, \mathbf{R}_r$. Here we note that, $a(a \geq 1)$ is the computing redundant rate and $a(m+r)$ is an integer.[1] We also note that $r$ is an important variable to be determined, which not only has great impacts on the total resource usage but also the existence of the secure linear coding scheme for the edge computing. Let $\mathbf{T} = \left[\mathbf{A}_1^\top, \ldots, \mathbf{A}_m^\top, \mathbf{R}_1^\top \ldots, \mathbf{R}_r^\top\right]^\top$ and the $a(m+r) \times (m+r)$ dimensional encoding coefficient matrix $\mathbf{B} = \left[\mathbf{B}_1^\top, \ldots, \mathbf{B}_k^\top\right]^\top$, in which $\mathbf{B}_j$ is the encoding coefficient matrix of the coded vectors to be stored on $s_j$ and $^\top$ denotes matrix transposition. To guarantee the decodability of the redundant computation, the encoding coefficient matrix $\mathbf{B}$ should be full rank and every $m+r$ row-vectors of $\mathbf{B}$ are linearly independent [4]. Finally, coded vectors, i.e., the row-vectors of $\mathbf{B}_j\mathbf{T}$, are distributed and stored on edge device $s_j$, $\forall j \in \{1, \ldots, k\}$. Let $V(\mathbf{B}_j)$ denote the number of rows in $\mathbf{B}_j$. In other words, the number of coded vectors stored on $s_j$ is $V(\mathbf{B}_j)$. We note that the encoding coefficient matrix is an empty matrix for the edge device which will not participate in the computation, i.e., no coded vector is stored on it. Therefore, $\sum_{j=1}^{k} V(\mathbf{B}_j) = a(m+r)$.

To compute $\mathbf{A}\mathbf{x}$, $s_0$ first sends the input vector $\mathbf{x}$ to the selected edge devices. Each edge device $s_j$ then multiplies the coded vectors, i.e., the row-vectors of $\mathbf{B}_j\mathbf{T}$, by $\mathbf{x}$ and sends the intermediate results $\mathbf{B}_j\mathbf{T}\mathbf{x}$ with length $V(\mathbf{B}_j)$ back to $s_0$. Then, $s_0$ can decode $\mathbf{A}\mathbf{x}$ after receiving any $m+r$ intermediate results. Specifically, since the user device $s_0$ receives any $m+r$ intermediate results, the aggregated results will be in the form of $\mathbf{B}_{(m+r)}\mathbf{T}\mathbf{x}$, where $\mathbf{B}_{(m+r)}$ is the $(m+r) \times (m+r)$ dimensional aggregated encoding matrix. The aggregated encoding matrix $\mathbf{B}_{(m+r)}$ is a full rank matrix, the user device can obtain $\mathbf{T}\mathbf{x}$ by Gaussian elimination, in which $\mathbf{A}\mathbf{x}$ is composed by the first $m$ values of $\mathbf{T}\mathbf{x}$. In Section 4.2.2, for $a = 1$, we give a secure linear coding design with much lower decoding complexity, in which the user device only needs to perform $m$ subtractions on the received $m+r$ intermediate results, i.e., values, to obtain the final result $\mathbf{y} = \mathbf{A}\mathbf{x}$.

In this paper, we consider minimizing the storage, computation and communication costs in SCEC. For each edge device $s_j$, let the unit cost of storage be $c_j^s$. We note that $l$ is the number of columns in data matrix $\mathbf{A}$ and the number of rows of input vector $\mathbf{x}$. Let the unit costs of addition and multiplication be $c_j^a$ and $c_j^m$, respectively, where $c_j^a \leq c_j^m$. Let the unit cost of communication from $s_j$ to $s_0$ be $c_j^d$. First, for storage, $s_j$ needs to store (1) an $l \times 1$ dimensional input vector $\mathbf{x}$, (2) $V(\mathbf{B}_j)$ coded vectors of the data matrix, each of which is a $1 \times l$ dimensional row-vector in $\mathbf{B}_j\mathbf{T}$, and (3) $V(\mathbf{B}_j)$ intermediate results (values) in $\mathbf{B}_j\mathbf{T}\mathbf{x}$. Therefore, the storage cost is up to $(l + V(\mathbf{B}_j)l + V(\mathbf{B}_j))c_j^s$. Second, to compute the multiplication between a $V(\mathbf{B}_j) \times l$ coded data matrix $\mathbf{B}_j\mathbf{T}$ and the $l \times 1$ input vector $\mathbf{x}$, the total computation cost is $V(\mathbf{B}_j)(lc_j^m + (l-1)c_j^a)$. Third, after completion of the computing task, $s_j$ shall send $V(\mathbf{B}_j)$ intermediate results (values) in $\mathbf{B}_j\mathbf{T}\mathbf{x}$ to $s_0$, which will lead to up to $V(\mathbf{B}_j)c_j^d$ communication cost. Therefore, the total cost on $s_j$ is

$$\begin{aligned}&\sum_{j=1}^{k}(l + (l+1)V(\mathbf{B}_j))c_j^s + V(\mathbf{B}_j)(lc_j^m + (l-1)c_j^a) + V(\mathbf{B}_j)c_j^d \\ &= \sum_{j=1}^{k}(((l+1)c_j^s + lc_j^m + (l-1)c_j^a + c_j^d)V(\mathbf{B}_j) + lc_j^s).\end{aligned} \tag{1}$$

To simplify the notations, we define $c_j = (l+1)c_j^s + lc_j^m + (l-1)c_j^a + c_j^d$ as the unit cost of each edge device $s_j$, which reflects the involved storage, computation, and communication cost for $s_j$ to handle one row-vector. Since $l$ and $c_j^s$ are given values, $\sum_{j=1}^{k} lc_j^s$ is fixed. Therefore, the problem of minimizing the total cost shown in Eq. (1) is equivalent to the problem of minimizing cost $c = \sum_{j=1}^{k} V(\mathbf{B}_j)c_j$. Let $\mathbf{C} = \{c_1, \ldots, c_k\}$. Without loss of generality, we assume $0 < c_{j_1} \leq c_{j_2}$ if $1 \leq j_1 \leq j_2 \leq k$.

Given an edge computing system $\mathbf{S}$, the resource limits of edge devices $\mathbf{W}$, the costs of edge devices $\mathbf{C}$, the $m \times l$ dimensional data matrix $\mathbf{A}$ and the computing redundant rate $a$, we define the task allocation and the corresponding linear code as an $a(m+r)$ dimensional *Linear Code for Edge Computing* (LCEC) $\phi(\mathbf{S}, \mathbf{W}, \mathbf{C}, \mathbf{A}, a, r)$. To guarantee that the task allocation is available for the edge computing system $\mathbf{S}$, and the user can decode the final result $\mathbf{y}$, we give the following two conditions.

**Definition 1 (Availability Condition).** *A task allocation of an $a(m+r)$ dimensional LCEC $\phi(\mathbf{S}, \mathbf{W}, \mathbf{C}, \mathbf{A}, a, r)$ satisfies the availability condition iff $\sum_{j=1}^{k} V(\mathbf{B}_j) = a(m+r)$ and $V(\mathbf{B}_j) \leq w_j, \forall j \in \{1, \ldots, k\}$.*

**Definition 2 (Decodability Condition).** *An $a(m+r)$ dimensional LCEC $\phi(\mathbf{S}, \mathbf{W}, \mathbf{C}, \mathbf{A}, a, r)$ satisfies the decodability condition iff the encoding coefficient matrix $\mathbf{B}$ is full rank and every $m+r$ row-vectors of $\mathbf{B}$ are linearly independent.*

To facilitate the discussions, we summarize notations in Table 2.

## 2.2 Attack Model and Secure Requirements

In this paper, we consider the case that each edge device can be an attacker or compromised by an attacker, who wants to know the information of data matrix $\mathbf{A}$. For example, in gradient-descent based algorithms, data matrix $\mathbf{A}$ is usually the personal data and input vector $\mathbf{x}$ in each iteration is only a temporary vector for obtaining the final weight vector [1], [4], [10], [11]. We assume that these edge devices do not collude with each other. Similar passive attack models have been investigated in [1], [10], [11], [12], [13], [28]. We note that the ideas proposed in this paper can also be extended to protect both data matrix $\mathbf{A}$ and input vector $\mathbf{x}$ simultaneously.

Let $H(\cdot)$ be entropy and $H(\cdot|\cdot)$ be conditional entropy. We define the *information-theoretic security* (ITS) requirement [1], [10], [11], [29] as follows:

[1] The redundant coded vectors can be used to not only assure security but also provide processing delay guarantee [4], [10], [11], [12], [21].

TABLE 2
Notations

| Notations | Meaning |
|---|---|
| $\mathbf{A}$ | the $m \times l$ dimensional data matrix, in which the $i$th row vector is $\mathbf{A}_i$. |
| $a$ | the computing redundant rate. |
| $\mathbf{B}$ | the encoding coefficient matrix. |
| $\mathbf{B}_j$ | the encoding coefficient matrix for edge device $s_j$. |
| $c_j$ | the unit cost of edge device $s_j$. |
| $\mathbf{C}$ | the costs of edge devices, $\mathbf{C} = \{c_1, \dots, c_k\}$. |
| $\mathbf{E}_m$ | the $m \times m$ dimensional identity matrix. |
| $k$ | the number of edge devices. |
| $L(\cdot)$ | the linear span space of row-vectors of a matrix. |
| $m$ | the number of row-vectors in the data matrix $\mathbf{A}$. |
| $l$ | the number of column-vectors in the data matrix $\mathbf{A}$. |
| $\mathbf{O}_{p,q}$ | the $p \times q$ dimensional zero matrix. |
| $\mathbf{R}$ | The set of random vectors $\mathbf{R} = \{\mathbf{R}_1, \mathbf{R}_2, \dots, \mathbf{R}_r\}$. |
| $r$ | the number of random vectors to be encoded with the data vectors. |
| $Rank(\cdot)$ | the rank of a vector set or matrix. |
| $\mathbf{S}$ | the set of edge devices and a user device, $\mathbf{S} = \{s_0, s_1 \cdots, s_k\}$. |
| $\mathbf{T}$ | the new matrix composed of data matrix $\mathbf{A}$ and random vectors $\mathbf{R}$, $\mathbf{T} = \left[\mathbf{A}_1^\top, \dots, \mathbf{A}_m^\top, \mathbf{R}_1^\top \dots, \mathbf{R}_r^\top\right]^\top$. |
| $V(\cdot)$ | the number of row-vectors in a matrix. |
| $\mathbf{W}$ | the sequence of resource limits of edge devices, $\mathbf{W} = \{w_1, \dots, w_k\}$. |
| $\mathbf{x}$ | the $l \times 1$ dimensional input vector. |
| $\mathbf{y}$ | $\mathbf{y} = \mathbf{A}\mathbf{x}$. |
| $\theta_r$ | the number of edge devices to participate in the SCEC. |
| $\top$ | matrix transposition. |
| $\{\cdot\}_b^a$ | the matrix composed by the set of row-vectors with indexes from $a$ to $b$ in a matrix. |
| $(\cdot)_{p,q}$ | the element in the $p$th row and $q$th column of a matrix. |

**Definition 3 (Security Condition).** *An $a(m+r)$ dimensional LCEC $\phi(\mathbf{S}, \mathbf{W}, \mathbf{C}, \mathbf{A}, a, r)$ satisfies the requirements of ITS iff*

$$H(\mathbf{A}|\mathbf{B}_j\mathbf{T}) = H(\mathbf{A}), \forall j \in \{1, \dots, k\}. \tag{2}$$

Let $\mathbf{E}_m$ be the $m \times m$ dimensional identity matrix and $\mathbf{O}_{p,q}$ be the $p \times q$ dimensional zero matrix. Let $\overline{\boldsymbol{\lambda}} = \left[\mathbf{E_m} \,\vdots\, \mathbf{O_{m,r}}\right]$ and $L(\cdot)$ be the span space of row-vectors of a matrix. According to [29], Eq. (2) is equivalent to: $dim(L(\mathbf{B}_j) \cap L(\overline{\boldsymbol{\lambda}})) = 0$, $\forall j \in \{1, \dots, k\}$. Suppose $\overline{\mathbf{B}}_j = \begin{bmatrix} \overline{\boldsymbol{\lambda}} \\ \mathbf{B}_j \end{bmatrix}$, we next present a sufficient and necessary condition of the ITS requirement.

**Theorem 1.** *$\mathbf{B}_j$ satisfies the security requirement shown in Eq. (2), iff $Rank(\overline{\mathbf{B}}_j) = m + Rank(\mathbf{B}_j)$.*

**Proof.** According to the properties of the linear space, we have

$$\begin{aligned} & dim(L(\mathbf{B}_j)) + dim(L(\overline{\boldsymbol{\lambda}})) \\ & = dim(L(\mathbf{B}_j) + L(\overline{\boldsymbol{\lambda}})) + dim(L(\mathbf{B}_j) \cap L(\overline{\boldsymbol{\lambda}})) \\ & = Rank(\overline{\mathbf{B}}_j) + dim(L(\mathbf{B}_j) \cap L(\overline{\boldsymbol{\lambda}})). \end{aligned}$$

Since $dim(L(\mathbf{B}_j)) = Rank(\mathbf{B}_j)$ and $dim(L(\overline{\boldsymbol{\lambda}})) = m$, we have $dim(L(\mathbf{B}_j) \cap L(\overline{\boldsymbol{\lambda}})) = m + Rank(\mathbf{B}_j) - Rank(\overline{\mathbf{B}}_j)$.

If $Rank(\overline{\mathbf{B}}_j) = m + Rank(\mathbf{B}_j)$, then $dim(L(\mathbf{B}_j) \cap L(\overline{\boldsymbol{\lambda}})) = 0$. Therefore, $\mathbf{B}_j$ satisfies the security requirement shown in Eq. (2).

On the other hand, if $\mathbf{B}_j$ satisfies the security requirement shown in Eq. (2), we have $dim(L(\mathbf{B}_j) \cap L(\overline{\boldsymbol{\lambda}})) = 0$. Therefore, $Rank(\overline{\mathbf{B}}_j) = m + Rank(\mathbf{B}_j)$. □

## 2.3 Problem Definition

In this paper, we study the *Minimum Cost Secure Coded Edge Computing* (MCSCEC) problem as follows:

**Definition 4 (The MCSCEC Problem).** *Given an edge computing system $\mathbf{S}$, the resource limits of edge devices $\mathbf{W}$, the costs of edge devices $\mathbf{C}$, a data matrix $\mathbf{A}$ and the computing redundant rate $a$, the MCSCEC problem is to minimize the total cost $c$ by finding a subset of edge devices that satisfy the availability, decodability and security conditions*

$$\begin{aligned} & \min_{\phi(\mathbf{S},\mathbf{W},\mathbf{C},\mathbf{A},a,r)} \quad c \\ \text{subject to} \quad & \text{Availability Condition (Definition 1)} \\ & \text{Decodability Condition (Definition 2)} \\ & \text{Security Condition (Definition 3).} \end{aligned}$$

We note that if an LCEC $\phi(\mathbf{S}, \mathbf{W}, \mathbf{C}, \mathbf{A}, a, r)$ satisfies the availability, decodability and security conditions, then it is a *feasible* solution for the MCSCEC problem.

## 2.4 The MCSCEC Framework

In this section, we provide an overview of the framework to solve the MCSCEC problem where the key components in the framework will be elaborated upon in the following sections.

- *Task Allocation.* In this step, the cloud shall first determine two parameters: $r$ (the number of random vectors to be encoded with data vectors) and $\theta_r$ (the number of edge devices to participate in the SCEC). We will elaborate on this in Section 4.1.
- *Coded Data Distribution.* As explained in the system model, the cloud shall first generate $r$ random vectors $\{\mathbf{R}_1, \mathbf{R}_2, \dots, \mathbf{R}_r\}$, then generate the encoding coefficient matrix $\mathbf{B} = [\mathbf{B}_1^\top, \dots, \mathbf{B}_{\theta_r}^\top]^\top$. Finally, the cloud computes and then distributes $\mathbf{B}_j\mathbf{T}$ to each selected edge device $s_j$. We will present the design of $\mathbf{B}$ in Section 4.2.
- *Coded Edge Computing.* After user device $s_0$ sends the input vector $\mathbf{x}$ to each edge device $s_j$, $s_j$ multiplies the coded data matrix $\mathbf{B}_j\mathbf{T}$ by $\mathbf{x}$ and sends the intermediate results $\mathbf{B}_j\mathbf{T}\mathbf{x}$ back to $s_0$.
- *Original Result Recovery.* When user device $s_0$ receives the first returned $m+r$ intermediate results, i.e., $\mathbf{B}_{(m+r)}\mathbf{T}\mathbf{x}$, it can decode and obtain $\mathbf{A}\mathbf{x}$. In Section 4.2, we will discuss how to use $\mathbf{B}_{(m+r)}\mathbf{T}\mathbf{x}$ to efficiently calculate the desired result $\mathbf{A}\mathbf{x}$.

# 3 Theoretical Analysis

In this section, we will first conduct a solid theoretical analysis to show the conditions for the existence of a feasible solution. We then prove the necessary conditions on the range of $r$ for the optimal solution, which enables us to further design the optimal task allocation scheme.

### 3.1 The Existence of a Feasible Solution

**Lemma 1.** *If LCEC $\phi(\mathbf{S}, \mathbf{W}, \mathbf{C}, \mathbf{A}, a, r)$ is a feasible solution of the MCSCEC problem, then the number of coded vectors allocated to $s_j$ satisfies $Rank(\mathbf{B}_j) = V(\mathbf{B}_j) \leq r$, $\forall j \in \{1, \ldots, k\}$.*

**Proof.** First, since LCEC $\phi(\mathbf{S}, \mathbf{W}, \mathbf{C}, \mathbf{A}, a, r)$ is a feasible solution of the MCSCEC problem, it satisfies the ITS requirements. According to Theorem 1, $Rank(\overline{\mathbf{B}}_j) = m + Rank(\mathbf{B}_j)$, where the dimension of matrix $\overline{\mathbf{B}}_j$ is $(V(\mathbf{B}_j) + m) \times (m + r)$. We have $Rank(\overline{\mathbf{B}}_j) \leq m + r$. Therefore, $Rank(\mathbf{B}_j) = Rank(\overline{\mathbf{B}}_j) - m \leq r$, $\forall j \in \{1, \ldots, k\}$. Next, since LCEC $\phi$ also satisfies the decodability condition, according to Definition 2, every $m + r$ row-vectors of $\mathbf{B}$ are linearly independent, we have $Rank(\mathbf{B}_j) = \min(m + r, V(\mathbf{B}_j))$. Consequently, $Rank(\mathbf{B}_j) = \min(m + r, V(\mathbf{B}_j)) \leq r$. Since $m > 0$, we have $\min(m + r, V(\mathbf{B}_j)) = V(\mathbf{B}_j) \leq r$, $\forall j \in \{1, \ldots, k\}$. Therefore, $Rank(\mathbf{B}_j) = V(\mathbf{B}_j) \leq r$. □

**Remark 1.** Lemma 1 shows that the decodability condition and the security condition require that the number of coded vectors allocated to each edge device is no more than $r$.

Before we show the following theorem, we let

$$\hbar_i^r = \sum_{j=1}^{i} \min(w_j, r), \forall i \in \{1, \ldots, k\} \text{ and } \forall r \geq 1. \tag{3}$$

We note that $\hbar_i^r$ is monotonically increasing with respect to $i$ and $r$.

**Theorem 2.** *If LCEC $\phi(\mathbf{S}, \mathbf{W}, \mathbf{C}, \mathbf{A}, a, r)$ is a feasible solution of the MCSCEC problem, then $\hbar_k^r \geq a(m + r)$ and $V(\mathbf{B}_j) \leq \min(w_j, r), \forall j \in \{1, \ldots, k\}$.*

**Proof.** Since LCEC $\phi(\mathbf{S}, \mathbf{W}, \mathbf{C}, \mathbf{A}, a, r)$ is a feasible solution of the MCSCEC problem, it satisfies the availability condition. Therefore, $V(\mathbf{B}_j) \leq w_j$, $\forall j \in \{1, \ldots, k\}$ and $\sum_{j=1}^{k} V(\mathbf{B}_j) = a(m + r)$. According to Lemma 1, $V(\mathbf{B}_j) \leq r$. Therefore, $V(\mathbf{B}_j) \leq \min(w_j, r)$, $\forall j \in \{1, \ldots, k\}$. Moreover, since $\hbar_k^r = \sum_{j=1}^{k} \min(w_j, r) \geq \sum_{j=1}^{k} V(\mathbf{B}_j), \hbar_k^r \geq a(m + r)$. □

**Remark 2.** Theorem 2 shows the necessary conditions for the feasible solution of the MCSCEC problem.

Before we give the following theoretical analysis, we let

$$\theta_r = \min(\{i | i \in \{1, \ldots, k\} \text{ and } \hbar_i^r \geq a(m + r)\}), \tag{4}$$

$$c_{sum}(r) = \sum_{j=1}^{\theta_r - 1} \min(w_j, r) c_j + \left[a(m + r) - \hbar_{\theta_r - 1}^r\right] c_{\theta_r}. \tag{5}$$

From the definition of $\theta_r$, we have $1 \leq a(m + r) - \hbar_{\theta_r - 1}^r \leq \min(w_{\theta_r}, r)$. We next give the sufficient and necessary condition that a feasible solution exists for the MCSCEC problem.

**Theorem 3.** *For a given $r$, there exists a feasible solution $\phi(\mathbf{S}, \mathbf{W}, \mathbf{C}, \mathbf{A}, a, r)$ of the MCSCEC problem, iff $\hbar_k^r \geq a(m+ r)$.*

**Proof.** For a given $r$, if there exists a feasible solution $\phi(\mathbf{S}, \mathbf{W}, \mathbf{C}, \mathbf{A}, a, r)$ of the MCSCEC problem, according to Theorem 2, $\hbar_k^r \geq a(m + r)$.

On the other hand, if $\hbar_k^r \geq a(m + r)$ for a given $r$, then $\theta_r \leq k$. We can obtain a task allocation as follows:

$$V(\mathbf{B}_j) = \begin{cases} \min(w_j, r) & , \forall j \in \{1, \ldots, \theta_r - 1\}; \\ a(m + r) - \hbar_{\theta_r - 1}^r & , j = \theta_r; \\ 0 & , \forall j \in \{\theta_r + 1, \ldots, k\}, \end{cases} \tag{6}$$

which satisfies the availability condition. Based on the task allocation, we can also obtain a secure coding scheme as designed in Section 4.2, which satisfies the decodability and security conditions (Theorem 8). Therefore, the scheme composed by the task allocation and the secure coding is a feasible solution of the MCSCEC problem. □

**Remark 3.** For a given $r$, Theorem 3 shows the sufficient and necessary condition that a feasible solution exists for the MCSCEC problem. It will be used to further determine the range of $r$ in the feasible solution of the MCSCEC problem.

Let $w_{max} = \max(\mathbf{W})$, we have the following theorem.

**Theorem 4.** *There exists a feasible solution for the MCSCEC problem, iff there exists a parameter $r$ that satisfies $r \in \{1, \ldots, w_{max}\}$ and $\hbar_k^r \geq a(m + r)$.*

**Proof.** If there exists a parameter $r$ that satisfies $r \in \{1, \ldots, w_{max}\}$ and $\hbar_k^r \geq a(m + r)$, according to Theorem 3, there exists a feasible solution of the MCSCEC problem.

Next, if there exists a feasible solution $\phi(\mathbf{S}, \mathbf{W}, \mathbf{C}, \mathbf{A}, a, r)$ of the MCSCEC problem, we have $r \geq 1$. In $\phi$, since $V(\mathbf{B}_j)$ is the number of coded vectors allocated to $s_j$, according to Theorem 2, we have $V(\mathbf{B}_j) \leq \min(w_j, r)$ and $\hbar_k^r \geq a(m + r)$. Consequently, when $1 \leq r \leq w_{max}$, then $r$ satisfies $r \in \{1, \ldots, w_{max}\}$ and $\hbar_k^r \geq a(m + r)$.

On the other hand, when $r > w_{max}$, then $\min(w_j, r) = w_j$, $\forall j \in \{1, \ldots, k\}$. Since $V(\mathbf{B}_j) \leq \min(w_j, r) = w_j$. we have $\sum_{j=1}^{k} V(\mathbf{B}_j) \leq \sum_{j=1}^{k} w_j$. Since $\phi$ satisfies the availability condition, $\sum_{j=1}^{k} V(\mathbf{B}_j) = a(m + r)$ and consequently $\sum_{j=1}^{k} w_j \geq a(m + r)$. Suppose that $r' = w_{max}$, we have $\hbar_k^{r'} = \sum_{j=1}^{k} \min(w_j, r') = \sum_{j=1}^{k} w_j \geq a(m + r) > a(m+ r')$. Therefore, according to Theorem 3, there exists a feasible solution $\phi(\mathbf{S}, \mathbf{W}, \mathbf{C}, \mathbf{A}, a, r')$, where $r'$ satisfies that $r' \in \{1, \ldots, w_{max}\}$ and $\hbar_k^{r'} \geq a(m + r')$.

In summary, there exists a feasible solution for the MCSCEC problem, if and only if there exists a parameter $r$ that satisfies $r \in \{1, \ldots, w_{max}\}$ and $\hbar_k^r \geq a(m + r)$. □

**Remark 4.** Theorem 4 shows the sufficient and necessary condition that the MCSCEC problem has a feasible solution, which will be used to judge whether the MCSCEC problem has a feasible solution or not in the design of task allocation shown in Section 4.1.

### 3.2 The Range of $r$ in the Optimal Solution

In the previous subsection, we only consider the case that there exists a feasible solution for the MCSCEC problem, i.e., there exists a parameter $r$ that satisfies $r \in \{1, \ldots, w_{max}\}$ and $\hbar_k^r \geq a(m + r)$. In the following theoretical analysis, we try to figure out the range of $r$ in the optimal solution of the MCSCEC problem, for which we define two parameters:

$$r_{min} = \min(\{r | r \in \{1, \dots, w_{max}\} \text{ and } \hbar_k^r \geq a(m+r)\}),$$
$$r_{max} = \max(\{r | r \in \{1, \dots, w_{max}\} \text{ and } \hbar_k^r \geq a(m+r)\}).$$

**Lemma 2.** *If there exists an optimal solution $\phi^*(\mathbf{S}, \mathbf{W}, \mathbf{C}, \mathbf{A}, a, r)$ of the MCSCEC problem, then there exists an optimal solution $\phi(\mathbf{S}, \mathbf{W}, \mathbf{C}, \mathbf{A}, a, r)$, in which the task allocation satisfies Eq. (6), and the cost of $\phi$ is $c_{sum}(r)$.*

**Proof.** We prove this statement in a constructive way. If $\phi^*(\mathbf{S}, \mathbf{W}, \mathbf{C}, \mathbf{A}, a, r)$ is one of the optimal solutions for the MCSCEC problem and the minimum cost is $c$, according to Theorem 3, $\hbar_k^r \geq a(m+r)$. Moreover, based on the given $r$, we can allocate the task according to Eq. (6) and obtain a feasible LCEC $\phi(\mathbf{S}, \mathbf{W}, \mathbf{C}, \mathbf{A}, a, r)$ according to the secure coding designs shown in Section 4.2. Specifically, in Section 4.2, Theorem 8 shows that $\phi(\mathbf{S}, \mathbf{W}, \mathbf{C}, \mathbf{A}, a, r)$ is a feasible solution because it satisfies the availability, decodability and security conditions.

We now show that $\phi(\mathbf{S}, \mathbf{W}, \mathbf{C}, \mathbf{A}, a, r)$ is also optimal. According to Eq. (6), $\sum_{j=1}^k V(\mathbf{B}_j) = a(m+r)$ and the cost of $\phi$ is $c_{sum}(r)$ shown in Eq. (5). Since the optimal solution of the MCSCEC problem $\phi^*$ satisfies the availability condition, we have $\sum_{j=1}^k V(\mathbf{B}_j^*) = a(m+r)$, where we let $V(\mathbf{B}_j^*)$ be the number of the coded vectors stored on $s_j$ in $\phi^*$. According to Theorem 2, for both the optimal $\phi^*$ and the feasible solution $\phi$, $V(\mathbf{B}_j^*) \leq \min(w_j, r)$ and $V(\mathbf{B}_j) \leq \min(w_j, r)$, for $\forall j \in \{1, \dots, k\}$. Moreover, $\sum_{j=1}^k V(\mathbf{B}_j^*) = \sum_{j=1}^k V(\mathbf{B}_j) = a(m+r)$. Given $r$, for $\forall j \in \{1, \dots, k\}$, $\min(w_j, r)$ and $a(m+r)$ are fixed. When $1 \leq j_1 \leq j_2 \leq k$, $c_{j_1} \leq c_{j_2}$. Therefore, $c_{sum}(r) \leq c$.

On the other hand, since $\phi^*$ is an optimal solution of the MCSCEC problem while $\phi$ is a feasible solution of the MCSCEC problem, we have $c \leq c_{sum}(r)$. Consequently, $c_{sum}(r) = c$, i.e., $\phi$ is also an optimal solution of the MCSCEC problem, in which the task allocation satisfies Eq. (6), and the cost of $\phi$ is $c_{sum}(r)$. □

**Remark 5.** Lemma 2 shows the existence of an optimal solution of the MCSCEC problem that satisfies Eq. (6). It will be used to design task allocation shown in Section 4.1 and coding design in Section 4.2.

Before we continue the following analysis, we let $w_{max}^r = \max(\{w_j | j \in \{1, \dots, \theta_r - 1\}\})$.

**Lemma 3.** *If $\hbar_k^{r_1} \geq a(m+r_1)$, $\hbar_k^{r_2} \geq a(m+r_2)$ and $r_1 > r_2 \geq w_{max}^{r_1}$, then $c_{sum}(r_1) > c_{sum}(r_2)$.*

**Proof.** Since $a(m+r_1) > a(m+r_2)$ and $r_1 > r_2 \geq w_{max}^{r_1}$, we have $\theta_{r_2} \leq \theta_{r_1}$, $\min(w_j, r_1) = w_j$, $\forall r_1 \in \{1, \dots, \theta_{r_1} - 1\}$, and $\min(w_j, r_2) = w_j$, $\forall r_2 \in \{1, \dots, \theta_{r_2} - 1\}$

$$\begin{aligned} c_{sum}(r_1) &= \sum_{j=1}^{\theta_{r_1}-1} \min(w_j, r_1) c_j \\ &\quad + [a(m+r_1) - \sum_{j=1}^{\theta_{r_1}-1} \min(w_j, r_1)] c_{\theta_{r_1}} \\ &= \sum_{j=1}^{\theta_{r_1}-1} w_j c_j + [a(m+r_1) - \sum_{j=1}^{\theta_{r_1}-1} w_j] c_{\theta_{r_1}}. \end{aligned}$$

$$\begin{aligned} c_{sum}(r_2) &= \sum_{j=1}^{\theta_{r_2}-1} \min(w_j, r_2) c_j \\ &\quad + [a(m+r_2) - \sum_{j=1}^{\theta_{r_2}-1} \min(w_j, r_2)] c_{\theta_{r_2}} \\ &= \sum_{j=1}^{\theta_{r_2}-1} w_j c_j + [a(m+r_2) - \sum_{j=1}^{\theta_{r_2}-1} w_j] c_{\theta_{r_2}}. \end{aligned}$$

If $\theta_{r_2} = \theta_{r_1}$, then $c_{sum}(r_1) - c_{sum}(r_2) = a(r_1 - r_2) c_{\theta_{r_1}} > 0$. If $\theta_{r_2} < \theta_{r_1}$, then

$$\begin{aligned} c_{sum}(r_1) &= \sum_{j=1}^{\theta_{r_2}-1} w_j c_j + \sum_{j=\theta_{r_2}}^{\theta_{r_1}-1} w_j c_j \\ &\quad + [a(m+r_1) - \sum_{j=1}^{\theta_{r_1}-1} \min(w_j, r_1)] c_{\theta_{r_1}}. \end{aligned}$$

Since $a(m+r_2) - \sum_{j=1}^{\theta_{r_2}-1} w_j \leq \min(w_{\theta_{r_2}}, r_2) \leq w_{\theta_{r_2}}$, $c_{sum}(r_1) - c_{sum}(r_2) > \sum_{j=\theta_{r_2}}^{\theta_{r_1}-1} w_j c_j - [a(m+r_2) - \sum_{j=1}^{\theta_{r_2}-1} w_j] c_{\theta_{r_2}} \geq \sum_{j=\theta_{r_2}}^{\theta_{r_1}-1} w_j c_j - w_{\theta_{r_2}} c_{\theta_{r_2}} \geq 0$.

Therefore, we have $c_{sum}(r_1) > c_{sum}(r_2)$. □

**Theorem 5.** *If an LCEC $\phi(\mathbf{S}, \mathbf{W}, \mathbf{C}, \mathbf{A}, a, r)$ is an optimal solution of the MCSCEC problem, then $r_{min} \leq r \leq r_{max}$.*

**Proof.** First, we consider the case that $1 \leq r \leq w_{max}$. Since $\phi$ is an optimal solution, $\phi$ must be feasible and $r$ must satisfy $\hbar_k^r \geq a(m+r)$ in Theorem 3. Then, according to the definitions of $r_{min}$ and $r_{max}$, $r_{min} \leq r \leq r_{max}$.

Next, we let $r^* = w_{max}$ and show that $r > r^*$ will lead to a contradiction. According to Lemma 2, for each optimal solution $\phi(\mathbf{S}, \mathbf{W}, \mathbf{C}, \mathbf{A}, a, r)$, there exists an optimal solution $\phi'(\mathbf{S}, \mathbf{W}, \mathbf{C}, \mathbf{A}, a, r)$, in which the task allocation satisfies Eq. (6) and the cost of $\phi'$ is $c_{sum}(r)$. In $\phi'$, suppose that the number of coded vectors allocated to $s_j$ is $V(\mathbf{B}'_j)$, $\forall j \in \{1, \dots, k\}$, we have $\sum_{j=1}^k V(\mathbf{B}'_j) = a(m+r)$.

According to Theorem 2, $V(\mathbf{B}'_j) \leq \min(w_j, r) \leq w_j \leq r^*$. We have $\hbar_k^{r^*} = \sum_{j=1}^k \min(w_j, r^*) \geq \sum_{j=1}^k V(\mathbf{B}'_j) = a(m+r) > a(m+r^*)$. According to the proof of Theorem 3, we can obtain a feasible solution $\phi^*(\mathbf{S}, \mathbf{W}, \mathbf{C}, \mathbf{A}, a, r^*)$ of the MCSCEC problem in which the number of random vectors is $r^*$, the task allocation satisfies Eq. (6), and the total cost of $\phi^*$ is $c_{sum}(r^*)$. Since $\hbar_k^r \geq a(m+r)$, $\hbar_k^{r^*} \geq a(m+r^*)$ and $r > r^* \geq w_{max}^r$, according to Lemma 3, we have $c_{sum}(r) > c_{sum}(r^*)$, which contradicts with the assumption that $\phi'$ is the optimal solution. □

**Remark 6.** Theorem 5 gives the range of $r$ in the optimal solution. It will be used to determine the range of $r$ to find the optimal task allocation shown in Section 4.1.

**Theorem 6.** *For each $r \in \{r_{min}, \dots, r_{max}\}$, there exists a feasible solution $\phi(\mathbf{S}, \mathbf{W}, \mathbf{C}, \mathbf{A}, a, r)$ for the MCSCEC problem.*

**Proof.** When $r = r_{min}$ or $r = r_{max}$, according to the definitions of $r_{min}$ and $r_{max}$, we have $\hbar_k^r \geq a(m+r)$ so a feasible solution exists. When $r \neq r_{min}$ and $r \neq r_{max}$, we can infer that $r_{min} < r < r_{max}$. For this case, we will prove that $\hbar_k^r \geq a(m+r)$ by contradiction.

Before further discussions, we define

$$y_r = |\{s_j | j \in \{1, \ldots, k\} \text{ and } w_j \geq r\}|,$$

which is the number of edge devices with the resource limits larger than or equal to $r$. Then, according to Eq. (3), for $\forall r_1 \geq 2$, we have

$$\begin{aligned} \hbar_k^{r_1} - \hbar_k^{r_1 - 1} &= \sum_{j=1}^{k} \min(w_j, r_1) - \sum_{j=1}^{k} \min(w_j, r_1 - 1) \\ &= \sum_{j=1}^{k} \big(\min(w_j, r_1) - \min(w_j, r_1 - 1)\big). \end{aligned}$$

In the equation above, $\forall j \in \{1, \ldots, k\}$, if $w_j \geq r_1$, $\min(w_j, r_1) - \min(w_j, r_1 - 1) = r_1 - (r_1 - 1) = 1$. Otherwise, if $w_j < r_1$, $\min(w_j, r_1) - \min(w_j, r_1 - 1) = w_j - w_j = 0$. Therefore, $\forall r_1 \geq 2$, we have $\hbar_k^{r_1} - \hbar_k^{r_1 - 1} = y_{r_1}$. Consequently, if $1 \leq r_2 < r_1 \leq w_{max}$, we have $y_{r_1} \leq y_{r_2}$ and $\hbar_k^{r_1} = \hbar_k^{r_2} + y_{r_2+1} + \cdots + y_{r_1} = \hbar_k^{r_2} + \sum_{j=r_2+1}^{r_1} y_j$.

We now construct a contradiction by assuming $\hbar_k^r < a(m + r)$ for an arbitrary $r$ with $r_{min} < r < r_{max}$. According to the definition of $r_{min}$, we have $\hbar_k^{r_{min}} \geq a(m + r_{min})$. Therefore, $\hbar_k^r = \hbar_k^{r_{min}} + \sum_{j=r_{min}+1}^{r} y_j \geq a(m + r_{min}) + \sum_{j=r_{min}+1}^{r} y_j$. Since $\hbar_k^r < a(m + r)$, $a(m + r_{min}) + \sum_{j=r_{min}+1}^{r} y_j < a(m + r)$. Therefore, $\sum_{j=r_{min}+1}^{r} y_j < a(m + r) - a(m + r_{min}) = a(r - r_{min})$. We have $\frac{1}{r - r_{min}} \sum_{j=r_{min}+1}^{r} y_j < a$, where the left hand side is the average of all $y_j$ when $r_{min} < j \leq r$.

For $\forall r_1 \in \{r + 1, \ldots, r_{max}\}$ and $\forall r_2 \in \{r_{min}, \ldots, r\}$, we have $y_{r_1} \leq y_{r_2}$. Therefore, $\frac{1}{r_{max} - r} \sum_{j=r+1}^{r_{max}} y_j \leq \frac{1}{r - r_{min}} \sum_{j=r_{min}+1}^{r} y_j < a$, i.e., $\sum_{j=r+1}^{r_{max}} y_j < a(r_{max} - r)$. Since $\hbar_k^{r_{max}} = \hbar_k^r + \sum_{j=r+1}^{r_{max}} y_j < a(m + r) + a(r_{max} - r) = a(m + r_{max})$, we have $\hbar_k^{r_{max}} < a(m + r_{max})$, which contradicts with the definition of $r_{max}$ that $\hbar_k^{r_{max}} \geq a(m + r_{max})$.

Therefore, for each $r \in \{r_{min}, \ldots, r_{max}\}$, we have $\hbar_k^r \geq a(m + r)$. According to Theorem 3, there exists a feasible solution $\phi(\mathbf{S}, \mathbf{W}, \mathbf{C}, \mathbf{A}, a, r)$ for the MCSCEC problem. □

**Remark 7.** Theorem 6 shows that for each $r \in \{r_{min}, \ldots, r_{max}\}$, there always exists a feasible solution of the MCSCEC problem, which will be used to guarantee that the solution obtained by task allocation algorithm shown in Section 4.1 is feasible.

## 4 The MCSCEC Schemes

In this section, based on the theoretical analysis shown in Section 3, we develop efficient optimal algorithms to first obtain a set of selected edge devices, i.e., task allocation, and then design secure coded computing schemes, i.e., coding designs. Moreover, we also prove that the cost achieved by the proposed schemes is the minimum cost.

### 4.1 Task Allocation (TA) Algorithm

In this subsection, we give the optimal *task allocation* (TA) algorithm for the MCSCEC problem. The details of the TA algorithm are shown in Algorithm 1. First, according to Theorems 4 and 5, we can judge whether the MCSCEC problem has a feasible solution or not and obtain the range of $r$ in the optimal solution, which is shown in Algorithm 1 lines 1-13. For each $r$ in this range, Theorems 3 and 6 show that there exists a feasible solution of the MCSCEC problem, in which the number of random vectors is $r$, the task allocation satisfies Eq. (6) and the total cost is $c_{sum}(r)$. Then, we can obtain the optimal $r$ with the minimum $c_{sum}(r)$ by exploiting the exhaustion algorithm, which is shown in Algorithm 1 lines 14-20. After that, based on the obtained $r$, we compute $\theta_r$ and $c_{sum}(r)$ according to Eqs. (4) and (5). We note that the task can be allocated to the first $\theta_r$ edge devices according to Eq. (6) and the cost of the task allocation is $c = c_{sum}(r)$.

In Algorithm 1, lines 2-8 are looped at most $w_{max}$ times, lines 9-13 are also looped at most $w_{max}$ times, lines 14-20 are looped at most $w_{max}$ times and the complexity of line 15 is $O(k)$. Finally, the complexity of line 21 is $O(k)$. Therefore, the complexity of Algorithm 1 is $O(kw_{max})$.

**Algorithm 1.** Task Allocation (TA) Algorithm

```
Input: S, W, C, $m$, $a$, $r$
Output: $r$, $\theta_r$, $c$
1  $r_{min} = 0$, $r_{max} = 0$, $r$, $c = INT\_MAX$;
2  for $r_{min} = 1$; $r_{min} \leq w_{max}$; $r_{min} = r_{min} + 1$ do
3    if $\hbar_k^{r_{min}} \geq a(m + r_{min})$ then
4      break;
5    else if $r_{min} == w_{max}$ then
6      return ERROR: no feasible solution exists;
7    end
8  end
9  for $r_{max} = w_{max}$; $r_{max} \geq r_{min}$; $r_{max} = r_{max} - 1$ do
10   if $\hbar_k^{r_{max}} \geq a(m + r_{max})$ then
11     break;
12   end
13 end
14 for $r^* = r_{min}$; $r^* \leq r_{max}$; $r^* = r^* + 1$ do
15   $c^* = c_{sum}(r^*)$;
16   if $c^* < c$ then
17     $c = c^*$;
18     $r = r^*$;
19   end
20 end
21 Compute $\theta_r$ and $c_{sum}(r)$ according to Eqs. (4), (5);
22 return $r$, $\theta_r$, $c_{sum}(r)$;
```

We next prove that the task allocation obtained by the proposed TA algorithm is optimal.

**Theorem 7.** *The TA algorithm gives the optimal task allocation of the MCSCEC problem.*

**Proof.** If there does not exist a parameter $r$ that satisfies $r \in \{1, \ldots, w_{max}\}$ and $\hbar_k^r \geq a(m + r)$, according to Theorem 4, there does not exist a feasible solution for the MCSCEC problem. In this case, the TA algorithm returns ERROR as shown in the lines 1-8 of Algorithm 1, which means that there does not exist a valid $r \in \{1, \ldots, w_{max}\}$ with $\hbar_k^r \geq a(m + r)$. □

If there exists a parameter $r$ that satisfies $r \in \{1, \ldots, w_{max}\}$ and $\hbar_k^r \geq a(m + r)$, i.e., the MCSCEC problem has at least one feasible solution, then in the TA algorithm, we can obtain $r_{min}$ and $r_{max}$ as shown in the lines 1-13 of Algorithm 1. Moreover, the TA algorithm will return an $r$ and $r \in$

$\{r_{min}, \ldots, r_{max}\}$. Theorem 6 guarantees that there always exists a feasible solution of the MCSCEC problem, in which the task allocation satisfies Eq. (6), the coding design is shown in Section 4.2 and the total cost is $c_{sum}(r)$. In this case, the task allocation returned by the TA algorithm is a feasible solution.

Next, we will prove that the task allocation obtained by the proposed TA algorithm is optimal. If there exists an optimal solution for the MCSCEC problem, in which $r'$ is the number of random vectors, according to Theorem 5 and Lemma 2, $r' \in \{r_{min}, \ldots, r_{max}\}$ and the total cost of it is $c_{sum}(r')$. Suppose that the TA algorithm returns $r, \theta_r$ and $c_{sum}(r)$, as shown in the lines 14-22 of Algorithm 1, since $c_{sum}(r) = \min_{r^* \in \{r_{min}, \ldots, r_{max}\}} c_{sum}(r^*)$, $c_{sum}(r) \leq c_{sum}(r')$. Therefore, task allocation obtained by the TA algorithm gives the optimal task allocation for the MCSCEC problem.

## 4.2 Secure Linear Coding Designs

From the task allocation algorithm shown in Section 4.1, we have determined the number of random vectors to be encoded with data vectors, i.e., $r$, and the number of edge devices participating in the secure CEC, i.e., $\theta_r$. Moreover, the number of coded vectors allocated to each edge device, i.e., $V(\mathbf{B}_j)$, $\forall j \in \{1, \ldots, \theta_r\}$, can also be obtained according to Eq. (6). Based on the task allocation, in this subsection, we will provide the designs of secure coding for the general case $a \geq 1$ and the special case $a = 1$. Specifically, the secure coding designed for the special case has lower decoding complexity. Furthermore, we also prove that both of them satisfy the availability, decodability, and security conditions.

Let $b_0 = 0$ and $b_j = \sum_{i=1}^{j} V(\mathbf{B}_i)$, $\forall j \in \{1, \ldots, \theta_r\}$. We give the two designs of secure linear coding as follows.

### *4.2.1 Redundant Secure Coding Design for $a \geq 1$*

When $a \geq 1$, we use an $a(m+r) \times (m+r)$ dimensional Vandermonde Matrix as the encoding coefficient matrix $\mathbf{B}$ because the Vandermonde Matrix has the *maximal distance separable* (MDS) property and can guarantee that any $(m+r)$ rows of matrix $\mathbf{B}$ satisfies full-rank with probability 1. Since the task allocation obtained by the proposed TA algorithm satisfies Eq. (6), we have $\mathbf{B}_j = \{\mathbf{B}\}_{b_j}^{b_{j-1}+1}$, $\forall j \in \{1, \ldots, \theta_r - 1\}$ and $\mathbf{B}_{\theta_r} = \{\mathbf{B}\}_{a(m+r)}^{b_{\theta_r - 1}+1}$. We next show the proposed redundant secure coding scheme is feasible for the MCSCEC problem.

**Theorem 8.** *For an LCEC $\phi(\mathbf{S}, \mathbf{W}, \mathbf{C}, \mathbf{A}, a, r)$, if the task allocation of $\phi$ satisfies Eq. (6) and the encoding coefficient matrix of $\phi$ is an $a(m+r) \times (m+r)$ dimensional Vandermonde Matrix, then $\phi$ satisfies the availability, decodability, and security conditions.*

**Proof.** First, LCEC $\phi$ satisfies the availability condition because the task allocation of $\phi$ follows Eq. (6), in which $V(\mathbf{B}_j) \leq \min(w_j, r), \forall j \in \{1, \ldots, k\}$, and $\sum_{j=1}^{k} V(\mathbf{B}_j) = a(m+r)$.

Second, since $\mathbf{B}$ is an $a(m+r) \times (m+r)$ dimensional Vandermonde Matrix and $a \geq 1$, $Rank(\mathbf{B}) = \min(a(m+r), m+r) = m+r$ and every $(m+r)$ rows-vectors of $\mathbf{B}$ are linearly independent. Therefore, the LCEC $\phi$ satisfies the decodability condition.

Third, we prove that $\mathbf{B}_j$ satisfies the security condition for each edge device $s_j$, $\forall j \in \{1, \ldots, \theta_r\}$. For $s_j$, we have

$$\overline{\mathbf{B}}_j = \left[\begin{array}{c} \overline{\lambda} \\ \hdashline \mathbf{B_j} \end{array}\right] = \left[\begin{array}{c:c} \mathbf{E_m} & \mathbf{O_{m,r}} \\ \hdashline \mathbf{B_j^L} & \mathbf{B_j^R} \end{array}\right],$$

in which we can partition matrix $\mathbf{B}_j$ into $\mathbf{B}_j^L$ and $\mathbf{B}_j^R$, using the first $m$ columns and the last $r$ columns in $\mathbf{B}_j$, respectively. Since $\mathbf{B}_j$ is a $V(\mathbf{B}_j) \times (m+r)$ dimensional Vandermonde Matrix, $\mathbf{B}_j^R$ is a $V(\mathbf{B}_j) \times r$ dimensional Vandermonde Matrix. According to Eq. (6), $V(\mathbf{B}_j) \leq r$ so $Rank(\mathbf{B}_j^R) = V(\mathbf{B}_j) = Rank(\mathbf{B}_j)$. Since $\overline{\mathbf{B}}_j$ can be treated as a partitioned matrix, $Rank(\overline{\mathbf{B}}_j) \geq Rank(\mathbf{E}_m) + Rank(\mathbf{B}_j^R) = m + V(\mathbf{B}_j)$. On the other hand, $\overline{\mathbf{B}}_j$ has $m + V(\mathbf{B}_j)$ rows, so $Rank(\overline{\mathbf{B}}_j) \leq m + V(\mathbf{B}_j)$. Therefore, $Rank(\overline{\mathbf{B}}_j) = m + Rank(\mathbf{B}_j)$. According to Theorem 1, $\mathbf{B}_j$ satisfies the security condition, $\forall j \in \{1, \ldots, \theta_r\}$, i.e., the LCEC $\phi$ satisfies the security condition. □

### *4.2.2 Non-Redundant Secure Coding Design for $a = 1$*

Before we provide the design of secure coding for the special case that $a = 1$, we prove that the task allocation obtained from the TA algorithm has the following special properties.

**Lemma 4.** *If $\phi(\mathbf{S}, \mathbf{W}, \mathbf{C}, \mathbf{A}, a, r)$ is an optimal solution of the MCSCEC problem and task allocation satisfies Eq. (6), then there exists an edge device $s_j$ which satisfies $V(\mathbf{B}_j) = r$, $j \in \{1, \ldots, \theta_r\}$.*

**Proof.** If $\phi(\mathbf{S}, \mathbf{W}, \mathbf{C}, \mathbf{A}, a, r)$ is an optimal solution of the MCSCEC problem and the task allocation of $\phi$ satisfies Eq. (6), then we have $V(\mathbf{B}_j) \leq \min(w_j, r), \forall j \in \{1, \ldots, \theta_r\}$, the total cost is $c_{sum}(r)$ and $\hbar_{\theta_r}^{r} = \sum_{j=1}^{\theta_r} \min(w_j, r) \geq \sum_{j=1}^{\theta_r} V(\mathbf{B}_j) = a(m+r)$. Next, we will prove the lemma by contradiction.

We assume that in $\phi$, $V(\mathbf{B}_j) < r, \forall j \in \{1, \ldots, \theta_r\}$. Since task allocation of $\phi$ satisfies Eq. (6), we have $V(\mathbf{B}_j) = \min(w_j, r) < r$, $\forall j \in \{1, \ldots, \theta_r - 1\}$. Therefore, $V(\mathbf{B}_j) = w_j < r, \forall j \in \{1, \ldots, \theta_r - 1\}$. According to Eq. (6), we have $V(\mathbf{B}_{\theta_r}) = a(m+r) - \sum_{j=1}^{\theta_r - 1} w_j \leq \min(w_{\theta_r}, r)$. Therefore, $\sum_{j=1}^{\theta_r - 1} w_j \geq a(m+r) - \min(w_{\theta_r}, r)$.

Let $r' = \max(\{V(\mathbf{B}_j) | j \in \{1, \ldots, \theta_r - 1\}\})$. Since $V(\mathbf{B}_j) = w_j < r$, $\forall j \in \{1, \ldots, \theta_r - 1\}$, we have $r' = w_{max}^{r} < r$. Therefore, $\hbar_k^{r'} \geq \hbar_{\theta_r}^{r'} = \sum_{j=1}^{\theta_r - 1} \min(w_j, r') + \min(w_{\theta_r}, r') = \sum_{j=1}^{\theta_r - 1} w_j + \min(w_{\theta_r}, r')$.

If $w_{\theta_r} \leq r'$, $\hbar_{\theta_r}^{r'} = \sum_{j=1}^{\theta_r} w_j$ and $\hbar_{\theta_r}^{r} = \sum_{j=1}^{\theta_r} w_j$. Therefore, we have $\hbar_{\theta_r}^{r'} = \hbar_{\theta_r}^{r}$. Since $\hbar_{\theta_r}^{r} \geq a(m+r)$ and $a(m+r) > a(m+r')$, $\hbar_{\theta_r}^{r'} > a(m+r')$. If $w_{\theta_r} > r'$, $\hbar_{\theta_r}^{r'} = \sum_{j=1}^{\theta_r - 1} w_j + r'$. Since $\sum_{j=1}^{\theta_r - 1} w_j \geq a(m+r) - \min(w_{\theta_r}, r)$, we have $\hbar_{\theta_r}^{r'} \geq a(m+r) - \min(w_{\theta_r}, r) + r' \geq a(m+r) - r + r' \geq a(m+r) - a(r-r') = a(m+r')$. Therefore, $\hbar_k^{r'} \geq a(m+r')$.

According to Theorem 3, we can obtain a feasible solution $\phi'(\mathbf{S}, \mathbf{W}, \mathbf{C}, \mathbf{A}, a, r')$ of the MCSCEC problem in which the usage of random vectors is $r'$, the task allocation satisfies Eq. (6), and the total cost of $\phi'$ is $c_{sum}(r')$.

Since $\hbar_k^{r} \geq \hbar_{\theta_r}^{r} \geq a(m+r)$, $\hbar_k^{r'} \geq a(m+r')$ and $r > r' \geq w_{max}^{r}$, according to Lemma 3, we have $c_{sum}(r) > c_{sum}(r')$, which contradicts with the assumption that $\phi$ is the optimal solution. Therefore, there exists an edge device $s_j$ that $V(\mathbf{B}_j) = r$ in the optimal solution $\phi$. □

**Theorem 9.** *In the task allocation obtained from the TA algorithm, there exists an edge device $s_z$, $z \in \{1, \ldots, \theta_r - 1\}$, which satisfies $V(\mathbf{B}_z) = r$.*

**Proof.** According to Theorem 7, the task allocation obtained from the TA algorithm is optimal and satisfies Eq. (6). Next, we prove the theorem by contradiction.

Considering an optimal LCEC $\phi(\mathbf{S}, \mathbf{W}, \mathbf{C}, \mathbf{A}, a, r)$ whose task allocation follows Algorithm 1, we assume that $V(\mathbf{B}_j) < r$, $\forall j \in \{1, \ldots, \theta_r - 1\}$. According to Lemma 4, $V(\mathbf{B}_{\theta_r}) = r$. According to Theorem 2, we have $V(\mathbf{B}_{\theta_r}) \leq \min(w_j, r)$. Therefore, $\min(w_{\theta_r}, r) = r$. Since the task allocation of $\phi$ satisfies Eq. (6) and $V(\mathbf{B}_j) < r$, $\forall j \in \{1, \ldots, \theta_r - 1\}$, we know $V(\mathbf{B}_j) = \min(w_j, r) = w_j$, $\forall j \in \{1, \ldots, \theta_r - 1\}$ and $\hbar^r_{\theta_r} = \sum_{j=1}^{\theta_r} \min(w_j, r) = \sum_{j=1}^{\theta_r - 1} w_j + r$. Based on Eq. (6), we also have $\hbar^r_{\theta_r} = \sum_{j=1}^{\theta_r} \min(w_j, r) \geq \sum_{j=1}^{\theta_r} V(\mathbf{B}_j) = a(m + r)$. Therefore, $\sum_{j=1}^{\theta_r - 1} w_j + r \geq a(m+ r)$.

Let $r' = \max(\{V(\mathbf{B}_j) | j \in \{1, \ldots, \theta_r - 1\}\})$. Since $V(\mathbf{B}_j) = w_j < r$, $\forall j \in \{1, \ldots, \theta_r - 1\}$, we have $r' = w^r_{max} < r$. Since $\min(w_{\theta_r}, r) = r$, we have $\min(w_{\theta_r}, r') = r'$. Therefore, $\hbar^{r'}_k \geq \hbar^{r'}_{\theta_r} = \sum_{j=1}^{\theta_r - 1} \min(w_j, r') + \min(w_{\theta_r}, r') = \sum_{j=1}^{\theta_r - 1} w_j + r' = \sum_{j=1}^{\theta_r - 1} w_j + r - r + r' \geq a(m + r) - r + r' \geq a(m + r) - a(r - r') = a(m + r')$.

According to Theorem 3, we can obtain a feasible solution $\phi'(\mathbf{S}, \mathbf{W}, \mathbf{C}, \mathbf{A}, a, r')$ of the MCSCEC problem in which the usage of random vectors is $r'$, the task allocation satisfies Eq. (6), and the total cost of $\phi'$ is $c_{sum}(r')$.

Since $\hbar^r_k \geq \hbar^r_{\theta_r} \geq a(m + r)$, $\hbar^{r'}_k \geq a(m + r')$ and $r > r' \geq w^r_{max}$, according to Lemma 3, we have $c_{sum}(r) > c_{sum}(r')$, which contradicts with the assumption that $\phi$ is the optimal solution. Therefore,there exists an edge device $s_z$, which satisfies $V(\mathbf{B}_z) = r$, $z \in \{1, \ldots, \theta_r - 1\}$. □

To construct an $(m + r) \times (m + r)$ encoding matrix $\mathbf{B}$, we let $\mathbf{E}_r$ be an $r \times r$ dimensional identity matrix. From Theorem 9, there exists an edge device $s_z$, $z \in \{1, \ldots, \theta_r - 1\}$, with $V(\mathbf{B}_z) = r$. For any other edge devise $s_j$ $(j \neq z)$, we can use the first $V(\mathbf{B}_j)$ row vectors in $\mathbf{E}_r$, denoted as $\{\mathbf{E}_r\}^1_{V(\mathbf{B}_j)}$ to construct an $m \times r$ dimensional matrix shown as follows:

$$\mathbf{E}_{m,r} = \left[\begin{array}{c} \{\mathbf{E}_r\}^1_{V(\mathbf{B}_1)} \\ \hline \vdots \\ \hline \{\mathbf{E}_r\}^1_{V(\mathbf{B}_{z-1})} \\ \hline \{\mathbf{E}_r\}^1_{V(\mathbf{B}_{z+1})} \\ \hline \vdots \\ \hline \{\mathbf{E}_r\}^1_{V(\mathbf{B}_{\theta_r})} \end{array}\right]$$

Since $\sum_j^{\theta_r} V(\mathbf{B}_j) = m + r$ and $V(\mathbf{B}_z) = r$, we have $\sum_{\forall j: j \neq z} V(\mathbf{B}_j) = m$. We can then define an $(m + r) \times (m + r)$ dimensional encoding coefficient matrix $\mathbf{B}$ as follows:

$$\mathbf{B} = \left[\begin{array}{c|c} \{\mathbf{E_m}\}^{\mathbf{1}}_{\mathbf{b_{z-1}}} & \{\mathbf{E_{m,r}}\}^{\mathbf{1}}_{\mathbf{b_{z-1}}} \\ \hline \mathbf{O_{r,m}} & \mathbf{E_r} \\ \hline \{\mathbf{E_m}\}^{\mathbf{b_{z-1}+1}}_{\mathbf{m}} & \{\mathbf{E_{m,r}}\}^{\mathbf{b_{z-1}+1}}_{\mathbf{m}} \end{array}\right], \tag{7}$$

in which

$$\mathbf{B_j} = \begin{cases} \left[\{\mathbf{E_m}\}^{\mathbf{b_{j-1}+1}}_{\mathbf{b_j}} \,\vdots\, \{\mathbf{E_r}\}^{\mathbf{1}}_{\mathbf{V(B_j)}}\right] & ,\mathbf{j} \in \{\mathbf{1}, \ldots, \mathbf{z-1}\}; \\ \left[\mathbf{O_{r,m}} \,\vdots\, \mathbf{E_r}\right] & ,\mathbf{j} = \mathbf{z}; \\ \left[\{\mathbf{E_m}\}^{\mathbf{b_{j-1}+1-r}}_{\mathbf{b_j-r}} \,\vdots\, \{\mathbf{E_r}\}^{\mathbf{1}}_{\mathbf{V(B_j)}}\right] & ,\mathbf{j} \in \{\mathbf{z+1}, \ldots, \theta_{\mathbf{r}}\}. \end{cases}$$

Next, we prove that the LCEC designed based on encoding coefficient matrix $\mathbf{B}$ satisfies the availability, decodability, and security conditions. Let $(\mathbf{B})_{p,q}$ be the $p$th row $q$th column element of $\mathbf{B}$.

**Lemma 5.** *If $\mathbf{B_j} = \left[\{\mathbf{E_m}\}^{\mathbf{j_1}}_{\mathbf{j_2}}\{\mathbf{E_r}\}^{\mathbf{1}}_{\mathbf{j_2-j_1+1}}\right]$, $j_1 \geq 1$, $j_2 \geq 1$, $r > j_2 - j_1 \geq 0$, then $\mathbf{B}_j$ satisfies the security condition.*

**Proof.** Using the definition of $\overline{\mathbf{B}}_j$, we have

$$\overline{\mathbf{B}}_j = \begin{bmatrix} \overline{\lambda} \\ \mathbf{B_j} \end{bmatrix} = \begin{bmatrix} \mathbf{E_m} & \mathbf{O_{m,r}} \\ \{\mathbf{E_m}\}^{\mathbf{j_1}}_{\mathbf{j_2}} & \{\mathbf{E_r}\}^{\mathbf{1}}_{\mathbf{j_2-j_1+1}} \end{bmatrix}.$$

Since $j_2 - j_1 + 1 \leq r$, we can further express $\overline{\mathbf{B}}_j$ as

$$\begin{bmatrix} \mathbf{E_m} & \mathbf{O_{m,j_2-j_1+1}} & \mathbf{O_{m,r-(j_2-j_1+1)}} \\ \{\mathbf{E_m}\}^{\mathbf{j_1}}_{\mathbf{j_2}} & \mathbf{E_{j_2-j_1+1}} & \mathbf{O_{j_2-j_1+1,r-(j_2-j_1+1)}} \end{bmatrix}.$$

Clearly, $Rank(\overline{\mathbf{B}}_j) = Rank(\left[\begin{array}{c|c} \mathbf{E_m} & \mathbf{O_{m,j_2-j_1+1}} \\ \hline \{\mathbf{E_m}\}^{\mathbf{j_1}}_{\mathbf{j_2}} & \mathbf{E_{j_2-j_1+1}} \end{array}\right]) = m + j_2 - j_1 + 1$. For the rank of $\mathbf{B}_j$, we note that $j_2 - j_1 + 1 \leq r$, so $Rank(\{\mathbf{E}_r\}^1_{j_2-j_1+1}) = j_2 - j_1 + 1$. Furthermore, since the number of rows in $\mathbf{B}_j$ is $j_2 - j_1 + 1$ and the rank of its submatrix $\{\mathbf{E}_r\}^1_{j_2-j_1+1}$ is $j_2 - j_1 + 1$, we have $Rank(\mathbf{B}_j) = j_2 - j_1 + 1$. Therefore, $Rank(\overline{\mathbf{B}}_j) = m + Rank(\mathbf{B}_j)$. According to Theorem 1, $\mathbf{B}_j$ satisfies the security condition, □

**Theorem 10.** *For an LCEC $\phi(\mathbf{S}, \mathbf{W}, \mathbf{C}, \mathbf{A}, 1, r)$, if the task allocation of $\phi$ satisfies Eq. (6) and the encoding coefficient matrix of $\phi$ is defined in Eq. (7), then $\phi$ satisfies the availability, decodability and security conditions.*

**Proof.** Since the task allocation of $\phi$ satisfies Eq. (6), $V(\mathbf{B}_j) \leq \min(w_j, r)$, $\forall j \in \{1, \ldots, k\}$, and $\sum_{j=1}^k V(\mathbf{B}_j) = m + r$. Therefore, the LCEC $\phi$ satisfies the availability condition.

With row transform, the matrix $\mathbf{B}$ becomes $\mathbf{B}' = \left[\begin{array}{c|c} \mathbf{E_m} & \mathbf{E_{m,r}} \\ \hline \mathbf{O_{r,m}} & \mathbf{E_r} \end{array}\right]$, $Rank(\mathbf{B}) = Rank(\mathbf{B}')$. Since $\mathbf{B}'$ is an upper triangular matrix and $(\mathbf{B}')_{p,p} = 1, \forall p \in \{1, \ldots, m + r\}$, $\mathbf{B}'$ is full rank. Therefore, $\mathbf{B}$ is full rank and the LCEC $\phi$ satisfies the decodability condition.

For edge device $s_j$, first, if $j = z$, since all the elements in the 1th to $m$th column of matrix $\mathbf{B}_z$ are 0, $\mathbf{B}_z\mathbf{Tx}$ are linear combinations of random vectors, i.e., $s_z$ cannot obtain any nonzero vector which is the linear combination of row-vectors of $\mathbf{A}$. Therefore, $\mathbf{B}_z$ satisfies security condition. Second, if $j \in \{1, \ldots, z - 1\}$, we have $\mathbf{B_j} = \left[\{\mathbf{E_m}\}^{\mathbf{b_{j-1}+1}}_{\mathbf{b_j}} \,\vdots\, \{\mathbf{E_r}\}^{\mathbf{1}}_{\mathbf{V(B_j)}}\right]$. Since $b_j - (b_{j-1} + 1) = V(\mathbf{B}_j) - 1$, $0 \leq b_j - (b_{j-1} + 1) < r$. According to Lemma 5, $\mathbf{B}_j$ satisfies the security condition, $\forall j \in \{1, \ldots, z - 1\}$. Third, if

TABLE 3
Simulation Settings

| Comparison Algorithms | *TA*, *MinNode*, *MaxNode*, *RNode* |
|---|---|
| Evaluation Parameters | $a, k, m, c_{max}, c_\mu, c_\sigma, w_{max}, w_\mu, w_\sigma$. |
| Default Values | $a = 1.25$, $k = 25$, $m = 5000$, $c_{max} = 5$, $c_\sigma = 1.25$, $w_{max} = 7000$, $w_\mu = 6000$, $w_\sigma = 420$ |
| Simulations Groups | (1) $\mathbf{C} \sim \mathcal{U}(1, c_{max})$, $\mathbf{W} \sim \mathcal{U}(100, w_{max})$;<br>(2) $\mathbf{C} \sim \mathcal{U}(1, c_{max})$, $\mathbf{W} \sim \mathcal{N}(w_\mu, {w_\sigma}^2)$;<br>(3) $\mathbf{C} \sim \mathcal{N}(c_\mu, {c_\sigma}^2)$, $\mathbf{W} \sim \mathcal{U}(100, w_{max})$;<br>(4) $\mathbf{C} \sim \mathcal{N}(c_\mu, {c_\sigma}^2)$, $\mathbf{W} \sim \mathcal{N}(w_\mu, {w_\sigma}^2)$. |

$j \in \{z+1, \ldots, \theta_r\}$, we have $\mathbf{B_j} = \left[\{\mathbf{E_m}\}_{\mathbf{b_j - r}}^{\mathbf{b_{j-1}+1-r}} \,\vdots\, \{\mathbf{E_r}\}_{\mathbf{V(B_j)}}^{\mathbf{1}}\right]$. Since $b_j - r - (b_{j-1} + 1 - r) = V(\mathbf{B}_j) - 1$, $0 \leq b_j - r - (b_{j-1} + 1 - r) < r$. According to Lemma 5, $\mathbf{B}_j$ satisfies the security condition, $\forall j \in \{z+1, \ldots, \theta_r\}$. Finally, for edge device $s_j$, $\forall j \in \{\theta_r + 1, \ldots, k\}$, since it is not allocated any coded vectors, the security condition is obviously satisfied. Therefore, $\mathbf{B}_j$ satisfies the security condition, $\forall j \in \{1, \ldots, k\}$.

Therefore, the LCEC $\phi$ satisfies the availability, decodability and security conditions. □

We now discuss the efficiency of the decoding process. Based on the encoding coefficient matrix $\mathbf{B}$, the coded data matrix $\mathbf{B}_j\mathbf{T}$ is migrated and stored on each edge device $s_j$, $\forall j \in \{1, \ldots, \theta_r\}$. After user device $s_0$ sends the input vector $\mathbf{x}$ to each edge device $s_j$, $\forall j \in \{1, \ldots, \theta_r\}$, $s_j$ multiplies the coded data matrix $\mathbf{B}_j\mathbf{T}$ by $\mathbf{x}$. Then, it sends the intermediate results $\mathbf{B}_j\mathbf{Tx}$ back to $s_0$. After the user device receives the intermediate results $\{\mathbf{B}_1\mathbf{Tx}, \ldots, \mathbf{B}_{\theta_r}\mathbf{Tx}\}$ from $\theta_r$ edge devices, it can obtain

$$\mathbf{BTx} = \begin{bmatrix} \mathbf{B_1 Tx} \\ \vdots \\ \mathbf{B}_{\theta_r}\mathbf{Tx} \end{bmatrix}.$$

Then, it can decode and recover the required result

$$\mathbf{Ax} = \begin{bmatrix} \mathbf{B_1 Tx} - \{\mathbf{B_z Tx}\}_{\mathbf{V(B_1)}}^{\mathbf{1}} \\ \vdots \\ \mathbf{B}_{z-1}\mathbf{Tx} - \{\mathbf{B}_z\mathbf{Tx}\}_{V(\mathbf{B}_{z-1})}^{1} \\ \mathbf{B}_{z+1}\mathbf{Tx} - \{\mathbf{B}_z\mathbf{Tx}\}_{V(\mathbf{B}_{z+1})}^{1} \\ \vdots \\ \mathbf{B}_{\theta_r}\mathbf{Tx} - \{\mathbf{B}_z\mathbf{Tx}\}_{V(\mathbf{B}_{\theta_r})}^{1} \end{bmatrix}.$$

For the computational complexity of decoding operations in the user device, it only needs to perform $m$ times calculations on the received $m + r$ intermediate results, i.e., values, to obtain the required results $\mathbf{Ax}$, which is much lower than the complexity of decoding $m + r$ intermediate results generated by using the Vandermonde Matrix as the encoding coefficient matrix $\mathbf{B}$.

In summary, the task allocation and coding design have been proposed in Sections 4.1 and 4.2, respectively. Theorem 7 shows that the proposed TA algorithm can derive the optimal task allocation, and the coding designs proposed in Section 4.2 can make the solutions satisfy the availability, security and decodiability conditions. Therefore, the solutions composed by task allocation shown in the TA algorithm and secure coding schemes shown in Section 4.2 are the optimal solutions of the MCSCEC problem.

## 5 Simulations

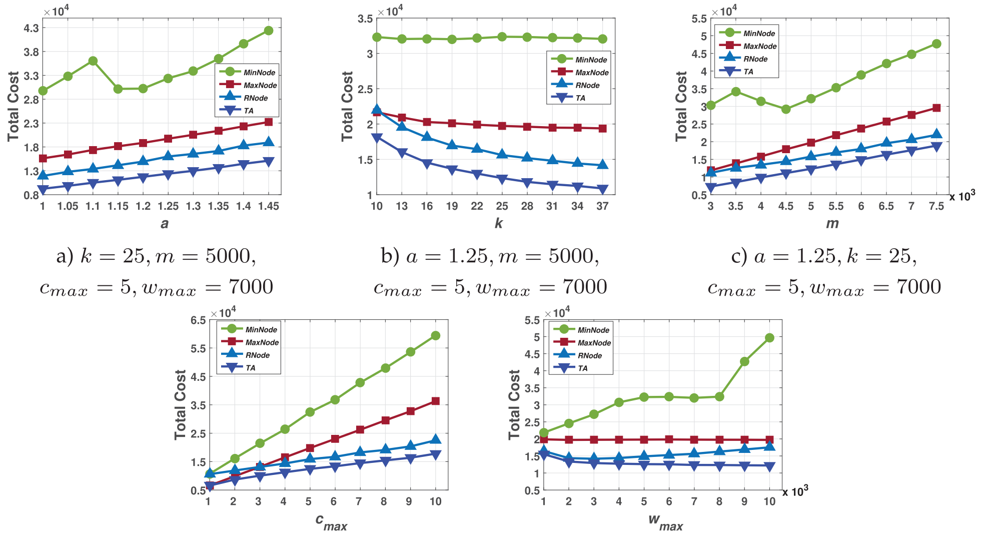
In this section, we conduct simulation experiments to evaluate the performance of the proposed scheme for the MCSCEC problem.

### 5.1 Simulation Settings

In the following simulations, we will compare the performance of the proposed scheme with the following baseline algorithms.

- For the *MinNode* scheme, as few edge devices as possible are selected to participate in the computation. To this end, we first sort edge devices in descending order by their resource limits. Then, we obtain the minimum number of edge devices, i.e., $\theta_{min} = \min_{r \in \{r_{min}, \ldots, r_{max}\}} \theta_r$ that can participate in the computation. The first $\theta_{min}$ edge devices can be allocated with tasks according to Eq. (6).
- For the *MaxNode* scheme, as many edge devices as possible are selected to participate in the computation. Specifically, we first sort edge devices in ascending order by their resource limits. Then, we obtain the maximum number of edge devices, i.e., $\theta_{max} = \max_{r \in \{r_{min}, \ldots, r_{max}\}} \theta_r$ that can participate in the computation. The first $\theta_{max}$ edge devices can be allocated with tasks according to Eq. (6).
- For the *RNode* scheme, we randomly select the value of $r$ from its range $\{r_{min}, \ldots, r_{max}\}$ and the first $\theta_r$ edge devices can be allocated with tasks according to Eq. (6).

We consider the performance of the *TA* algorithm when unit cost and resource limit, i.e., $\mathbf{C}$ and $\mathbf{W}$, obey different distributions, i.e., uniform distribution $\mathcal{U}$ and normal distribution $\mathcal{N}$ [1], [5]. Specifically, we will consider four groups of simulations when (1) $\mathbf{C} \sim \mathcal{U}(1, c_{max})$, $\mathbf{W} \sim \mathcal{U}(100, w_{max})$; (2) $\mathbf{C} \sim \mathcal{U}(1, c_{max}), \mathbf{W} \sim \mathcal{N}(w_\mu, {w_\sigma}^2)$; (3) $\mathbf{C} \sim \mathcal{N}(c_\mu, {c_\sigma}^2), \mathbf{W} \sim \mathcal{U}(100, w_{max})$ and (4) $\mathbf{C} \sim \mathcal{N}(c_\mu, {c_\sigma}^2), \mathbf{W} \sim \mathcal{N}(w_\mu, {w_\sigma}^2)$. Including the above parameters of distributions, we also consider the performances of the *TA* algorithm under different values of the redundant rate, i.e., $a$, the number of edge devices, i.e., $k$, and the number of row vectors in data matrix $\mathbf{A}$, i.e., $m$. The default values of these parameters are: $a = 1.25$, $k = 25$, $m = 5000$, $c_{max} = 5$, $c_\mu = 4$, $c_\sigma = 1.25$, $w_{max} = 7000$, $w_\mu = 6000$, $w_\sigma = 420$. For given $a$, $m$ and $r$, we use $\lceil a(m + r) \rceil$ instead of $a(m + r)$ to represent the total number of coded vectors to be stored on the edge devices. For each combination of parameters, we generate 1,000 instances to report the average results. Next, we will evaluate the performance of the proposed scheme for the MCSCEC problem from the impacts of computing parameters ($a$, $k$ and $m$), cost distributions ($c_\mu$, $c_{max}$ and $c_\sigma$) and resource limits ($w_\mu$, $w_{max}$ and $w_\sigma$), respectively. All simulation settings are shown in Table 3.



a) $k = 25, m = 5000, c_{max} = 5, w_{max} = 7000$

b) $a = 1.25, m = 5000, c_{max} = 5, w_{max} = 7000$

c) $a = 1.25, k = 25, c_{max} = 5, w_{max} = 7000$

d) $a = 1.25, m = 5000, k = 25, w_{max} = 7000$

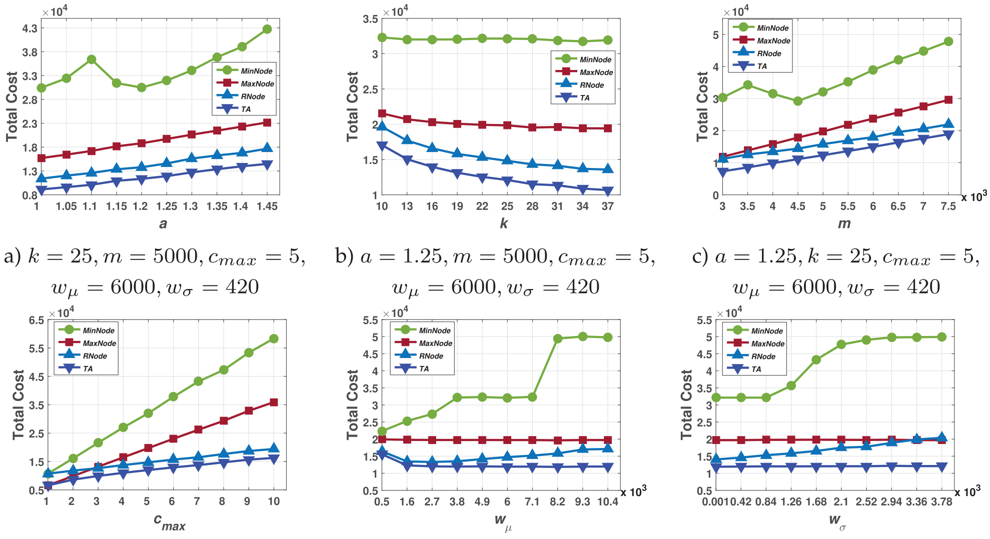
e) $a = 1.25, k = 25, m = 5000, c_{max} = 5$

Fig. 2. Total cost versus different parameters, when $\mathbf{C} \sim \mathcal{U}(1, c_{max}), \mathbf{W} \sim \mathcal{U}(100, w_{max})$.

## 5.2 The Impacts of Computing Parameters

In Figs. 2a, 3a, 4a, and 5a, we demonstrate the impacts of the redundant rate, i.e., $a$. Specifically, experiments show that the total costs of *MaxNode*, *RNode*, and *TA* increase with the increase of $a$, while the cost of *MinNode* does not monotonically increase with $a$. These results are due to the fact that $a$ has a complicated effect on the number of random vectors, i.e., $r$. In particular, when $a$ is near 1, we observe that increasing $a$ does not change the set of selected edge devices in the optimal settings. Therefore, the costs of all schemes increase with the increase of $a$ when $a \leq 1.1$. When $a$ continues to increase, the increase of $a$ leads to a larger set of selected edge devices. In such a scenario, depending on the task allocation scheme, when the selected number of devices is small, each device is allocated with a large number of coded vectors so $r$ must be large to satisfy the security condition. On the other hand, when more new edge devices are selected, $r$ may decrease shapely. For instance, when *MinNode* is used, we observe that the total cost decreases when $a$ increases from 1.1 to 1.15. In the case that the selected number of edge devices is large enough, the decrease of $r$ becomes negligible. Therefore, for all the schemes, the total costs increase as the redundant rate $a$ increases beyond 1.2. Finally, our results show that, when $a$ is sufficiently large, compared with the *MinNode*, *MaxNode*, and *RNode* algorithms, the *TA* algorithm can reduce the total cost by more than 58.2, 23.8, and 15.4 percent under different distributions of unit cost and resource limits, respectively.

Figs. 2b, 3b, 4b, and 5b illustrate the impacts of the number of edge devices, i.e., $k$. These results show that the total costs of *MaxNode*, *RNode*, and *TA* decrease with the increase



a) $k = 25, m = 5000, c_{max} = 5, w_{\mu} = 6000, w_{\sigma} = 420$

b) $a = 1.25, m = 5000, c_{max} = 5, w_{\mu} = 6000, w_{\sigma} = 420$

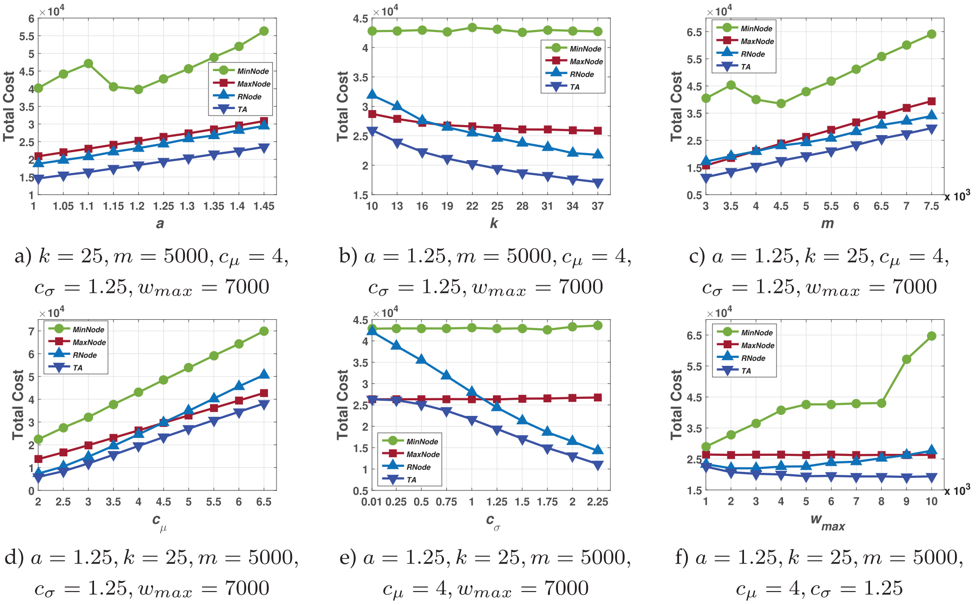
c) $a = 1.25, k = 25, c_{max} = 5, w_{\mu} = 6000, w_{\sigma} = 420$

d) $a = 1.25, k = 25, m = 5000, w_{\mu} = 6000, w_{\sigma} = 420$

e) $a = 1.25, k = 25, m = 5000, c_{max} = 5, w_{\sigma} = 420$

f) $a = 1.25, k = 25, m = 5000, c_{max} = 5, w_{\mu} = 6000$

Fig. 3. Total cost versus different parameters, when $\mathbf{C} \sim \mathcal{U}(1, c_{max}), \mathbf{W} \sim \mathcal{N}(w_{\mu}, {w_{\sigma}}^2)$.



a) $k = 25, m = 5000, c_\mu = 4, c_\sigma = 1.25, w_{max} = 7000$

b) $a = 1.25, m = 5000, c_\mu = 4, c_\sigma = 1.25, w_{max} = 7000$

c) $a = 1.25, k = 25, c_\mu = 4, c_\sigma = 1.25, w_{max} = 7000$

d) $a = 1.25, k = 25, m = 5000, c_\sigma = 1.25, w_{max} = 7000$

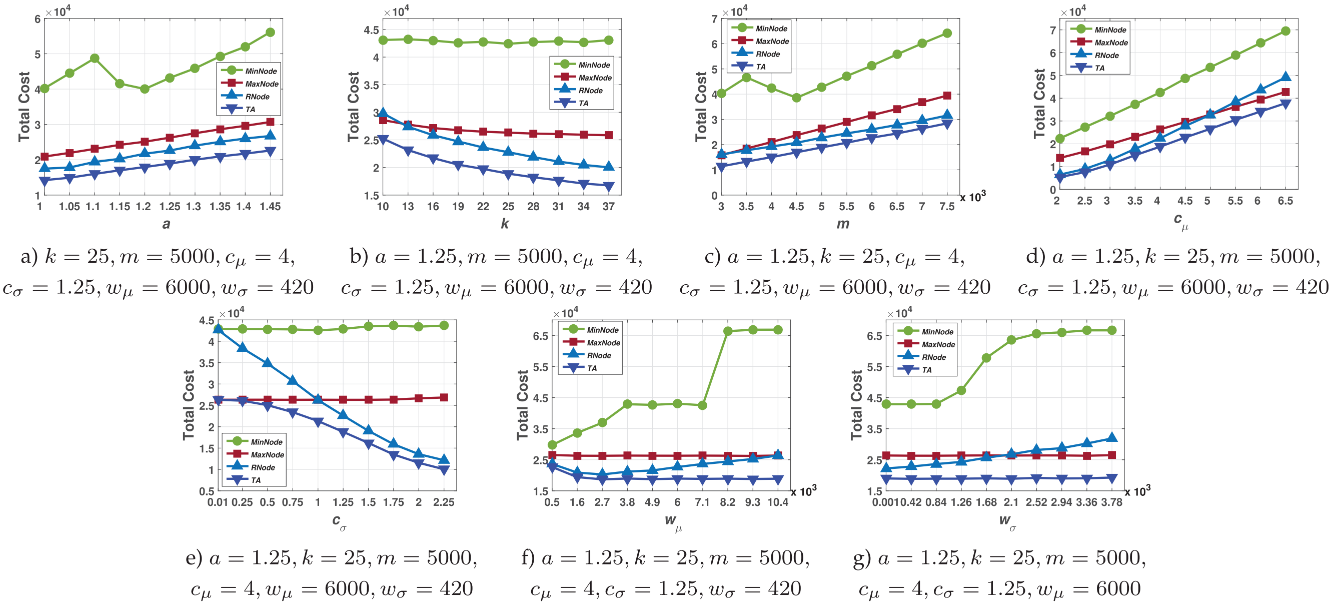
e) $a = 1.25, k = 25, m = 5000, c_\mu = 4, w_{max} = 7000$

f) $a = 1.25, k = 25, m = 5000, c_\mu = 4, c_\sigma = 1.25$

Fig. 4. Total cost versus different parameters, when $\mathbf{C} \sim \mathcal{N}(c_\mu, {c_\sigma}^2), \mathbf{W} \sim \mathcal{U}(100, w_{max})$.

of $k$, while the cost of *MinNode* is almost the same with different $k$. The behaviors of *MinNode* is mainly due to the facts that this scheme aims to select the minimal number of devices that have high resource limits, and that, in our experiments, the increase of $k$ does not significantly affect the characteristics of the set of selected devices. On the other hand, for *MaxNode*, *RNode*, since more edge devices with low unit costs are available for selection, the total costs of them decrease with the increase of $k$. For *TA*, when $k$ increases, the optimization spaces of the *TA* algorithm become larger, which decreases the usage of random vectors $r$ and the total cost of task allocation. Finally, when $k$ is sufficiently large, compared with the *MinNode*, *MaxNode*, and *RNode* algorithms, the proposed *TA* algorithm can reduce the total cost by more than 59.6, 33.5, and 16.3 percent under different distributions of unit cost and resource limits, respectively.

Next, we use Figs. 2c, 3c, 4c, and 5c to evaluate the impacts of the number of rows in data matrix $\mathbf{A}$, i.e., $m$. The experimental results show that the total costs of *MaxNode*, *RNode*, and *TA* increase with the increase of $m$. On the other hand, for *MinNode*, the impact of $m$ is similar to the impact of $a$ on the total cost. We observe that the reasons are also similar, i.e., when $m$ is in a certain range (e.g., $3.5 \times 10^3$ to $4.5 \times 10^3$), increasing $m$ will require more edge devices to participate in the computing so the required $r$ is decreasing significantly, which leads to the decrease of the total cost. In addition to these observations, we can see that the gap between *RNode* and *TA* becomes smaller as $m$ increases. This is because (1) the resource limit of each edge device is fixed, and (2) the increase of $m$ results in a smaller range of $r$ for feasible solutions, i.e., a smaller optimization space of the *TA* algorithm. Nevertheless, the proposed *TA* can still outperform all others. In particular, when $m$ is sufficiently



a) $k = 25, m = 5000, c_\mu = 4, c_\sigma = 1.25, w_\mu = 6000, w_\sigma = 420$

b) $a = 1.25, m = 5000, c_\mu = 4, c_\sigma = 1.25, w_\mu = 6000, w_\sigma = 420$

c) $a = 1.25, k = 25, c_\mu = 4, c_\sigma = 1.25, w_\mu = 6000, w_\sigma = 420$

d) $a = 1.25, k = 25, m = 5000, c_\sigma = 1.25, w_\mu = 6000, w_\sigma = 420$

e) $a = 1.25, k = 25, m = 5000, c_\mu = 4, w_\mu = 6000, w_\sigma = 420$

f) $a = 1.25, k = 25, m = 5000, c_\mu = 4, c_\sigma = 1.25, w_\sigma = 420$

g) $a = 1.25, k = 25, m = 5000, c_\mu = 4, c_\sigma = 1.25, w_\mu = 6000$

Fig. 5. Total cost versus different parameters, when $\mathbf{C} \sim \mathcal{N}(c_\mu, {c_\sigma}^2), \mathbf{W} \sim \mathcal{N}(w_\mu, {w_\sigma}^2)$.

large, compared with the *MinNode*, *MaxNode*, and *RNode* algorithms, the *TA* algorithm can reduce the total cost by more than 54.0, 25.2, and 10.0 percent under different distributions of unit cost and resource limits, respectively.

### 5.3 The Impacts of Cost Distributions

In the next two groups of experiments, we study the impacts of the cost distributions. Figs. 2d, 3d, 4d and 5d show that the total costs of all algorithms increase as $c_{max}$ and $c_{\mu}$ increase. The reason is obviously that the unit cost of each edge device increases. Moreover, when the unit costs of all edge devices are almost the same, i.e., $c_{max}$ is small or $c_{\mu}$ is large, the *TA* algorithm will select as many edge devices as possible to reduce the value of $r$. Therefore, in this case, the total cost of *TA* is very close to that of *MaxNode*. However, with the increase of $c_{max}$ or decrease of $c_{\mu}$, the gap between *TA* and *MaxNode* becomes larger because the range of unit costs increases as $c_{max}$ increases, but decreases as $c_{\mu}$ increases. Interestingly, while *MaxNode* performs well when the range of unit costs is small, *RNode* performs well when the range of unit costs is large. Therefore, the performance relationship of *MaxNode* and *RNode* in Figs. 2d and 3d is opposite to that of Figs. 4d and 5d. Finally, we note that the proposed *TA* algorithm can achieve the best cost performance in all cases. In particular, when $c_{max}$ is sufficiently large, compared with the *MinNode*, *MaxNode*, and *RNode* algorithms, the *TA* algorithm can reduce the total cost by more than 70.0, 50.9, and 16.2 percent under different distributions of unit cost and resource limits, respectively. When $c_{\mu}$ is sufficiently large, compared with the *MinNode*, *MaxNode*, and *RNode* algorithms, the *TA* algorithm can reduce the total cost by more than 45.5, 10.8, and 23.0 percent under different distributions of unit cost and resource limits, respectively.

In Figs. 4e and 5e, we note that the performance of *MinNode* and *MaxNode* almost do not change with the increase of $c_{\sigma}$. This is because we order the edge devices according to their resource limits in *MinNode* and *MaxNode*, so the order of devices is independent to $c_{\sigma}$. Besides, each point in any curve represents the average cost in a large number of experiments. Therefore, the total costs of *MinNode* and *MaxNode* are related more to the average value of $\mathbf{C}$, i.e., $c_{\mu}$. By comparison, the total costs of *RNode* and *TA* decrease as $c_{\sigma}$ increases. The reasons are as follows. First, when $c_{\sigma}$ is sufficiently small, e.g., $c_{\sigma} = 0.01$, the unit costs of all the edge devices are almost the same. In this case, the *TA* algorithm will select as many edge devices as possible to reduce the value of $r$ so the total cost of *TA* is almost the same as that of *MaxNode*, while the *RNode* algorithm tries to find the smallest number of low-cost devices so the total cost of *RNode* is the same as that of *MinNode*. Second, with the increase of $c_{\sigma}$, the number of edge devices with low unit costs increases. In this case, since both *RNode* and *TA* try to allocate tasks to edge devices with low unit costs, their total cost decrease as $c_{\sigma}$ increases. Since the increase of $c_{\sigma}$ does not change the range of $r$ and *RNode* always randomly selects an $r$ from this range, the cost of *RNode* decreases faster than the cost of *TA*. Nevertheless, the cost of *TA* is always lower than that of *RNode*. Finally, when $c_{\sigma}$ is sufficiently large, compared with the *MinNode*, *MaxNode*, and *RNode* algorithms, the *TA* algorithm can reduce the total cost by more than 74.4, 58.2, and 17.4 percent under different distributions of unit cost and resource limits, respectively.

### 5.4 The Impacts of Resource Limits

In the last two groups of experiments, we investigate the impacts of the resource limits. Figs. 2e, 3e, 4f, and 5f illustrate that, with the increase of the average resource limit, the total costs of *MinNode* and *RNode* increase, the total cost of *MaxNode* is almost unchanged, and the total cost of *TA* can slightly decrease. For the *MinNode* algorithm, (1) when $w_{max}$ and $w_{\mu}$ are small, since the *MinNode* algorithm needs to use more edge devices to allocate the task, the performance of *MinNode* is close to that of *MaxNode*; (2) when $w_{max}$ and $w_{\mu}$ become larger, since the *MinNode* algorithm tries to use as few as possible edge devices, the number of random vectors used in the computation becomes larger so the total cost of *MinNode* increases; (3) when $w_{max}$ and $w_{\mu}$ become sufficiently larger, the *MinNode* algorithm will use only two edge devices to complete the task so the increase of $w_{max}$ and $w_{\mu}$ will have no impact on the performance of *MinNode*. For the *MaxNode* algorithm, since it tries to use as many as possible edge devices, the number of coded vectors allocated to each selected edge device is small. Therefore, the increase of $w_{max}$ and $w_{\mu}$ will have a negligible impact on the performance of *MaxNode*. For the *RNode* algorithm and the *TA* algorithm, the increase of $w_{max}$ and $w_{\mu}$ will increase the range of $r$, which leads to the poor performance of *RNode*, but will increase the optimization space of the *TA* algorithm. Therefore, when $w_{max}$ and $w_{\mu}$ increase, the cost of *RNode* increases, and that of *TA* decreases. Finally, when $w_{max}$ and $w_{\mu}$ are sufficiently large, compared with the *MinNode*, *MaxNode*, and *RNode* algorithms, the *TA* algorithm can reduce the total cost by more than 71.2, 27.1, and 28.6 percent under different distributions of unit cost and resource limits, respectively.

In Figs. 3f and 5g, with the increase of $w_{\sigma}$, the total costs of *MinNode* and *RNode* increase, the total costs of *MaxNode* and *TA* are rather stable. For *MinNode*, when $w_{\sigma}$ increases, the number of edge devices with high resource limits increases so the *MinNode* algorithm will select a smaller number of edge devices involved in the computing, which leads to the increase of $r$ and the total cost. The increase of $w_{\sigma}$ will enlarge the range of $r$, which leads to the poor performance of *RNode*. For the *MaxNode* algorithm, since it tries to use as many as possible edge devices, and the number of coded vectors allocated to each selected edge device is small. Therefore, the increase of $w_{\sigma}$ will have less impact on the performance of *MaxNode*. For *TA*, since nodes are sorted according to the unit cost of computing, the impacts of $w_{\sigma}$ will be average out in a large number of experiments. Finally, when $w_{\sigma}$ is sufficiently large, compared with the *MinNode*, *MaxNode*, and *RNode* algorithms, the *TA* algorithm can reduce the total cost by more than 55.8, 28.0, and 18.0 percent under different distributions of unit cost and resource limits, respectively.

From all the above simulation experiments, we can conclude that the proposed *TA* algorithm always outperforms the *MinNode*, *MaxNode*, and *RNode* algorithms.

## 6 Related Work

For the edge computing systems, in many scenarios, the computing devices are operated by service providers. For instance, using the standardized multi-access edge computing (MEC) [2], a mobile network operator can deploy edge computing devices in the edge of its cellular network. Similarly, in Google's gaming platform, Stadia, game engines are running in Google's edge nodes to generate video streams to users [30]. Although the aforementioned edge computing systems are important, open edge computing platforms, such as the KubeEdge system [31] and the Baetyl platform [32] have also attracted more and more attentions. In such scenarios, security concerns, especially the confidentiality and privacy of user data, become more important because some computing devices may be malicious. Therefore, in this paper, we investigate an edge computing system, in which the computing task of one user device can be distributed to multiple nearby user devices using coded distributed computing to protect the confidentiality of user data.

In recent years, coded distributed computing (CDC) has been proposed and applied to efficiently perform different distributed computing tasks. First, to minimize the total computation time, Yu *et al.* proposed an optimal resource allocation scheme for distributed computation [7]. Lee *et al.* proposed *Product Code* to reduce the total computation time of distributed computation [14]. Dutta *et al.* designed a *Short-Dot* coding scheme for matrix-vector multiplication $\mathbf{A}\mathbf{x}$, in which each device only needs a part of vector $\mathbf{x}$ to perform coded computation [15]. Yu *et al.* designed *Polynomial Codes* to achieve the optimum recovery threshold, i.e., the minimum number of workers that the master needs to wait for decoding the final result [16]. Second, to minimize the total communication load, Li *et al.* proposed a CDC framework for *MapReduce* based distributed computing by repetitively mapping tasks to different servers and exploiting coded multicast for results exchange [8].

In addition to reducing computation latency by exploiting coded computation, data confidentiality is also concerned by users. Recently, two kinds of attack models have been studied in the literature, including the external wiretapping attack and the internal eavesdropping attack. For the external wiretapping attack, the transmission channels in the edge computing may be wiretapped and intermediate results exchanged by the edge devices can be acquired by the attackers. Specifically, Zhao *et al.* considered the secure data shuffling problem under the external wiretapping attack model by utilizing the linear coding [33]. In the internal eavesdropping attack, each edge device may be a potential attacker and can eavesdrop data stored by itself. Under this kind of attack, for the secure matrix-vector multiplication problem (i.e., calculating $\mathbf{A}\mathbf{x}$), Bitar *et al.* designed staircase codes to ensure the *information-theoretic security* (ITS) of data matrix $\mathbf{A}$ on each computing device, and analyzed the expected delay of task completion [10], [11]. For the secure matrix multiplication problem (i.e., calculating $\mathbf{AB}$), Yang *et al.* designed a polynomial code scheme to ensure the ITS of $\mathbf{A}$ and $\mathbf{B}$ on each computing node, analyzed the upper and lower bounds of recovery threshold and the total task completion delay [12]. In [22], Kakar *et al.* studied the cooperative attack model and improved the polynomial coding scheme. Although existing studies have considered confidentiality in the distributed computing, they focus on the homogeneous distributed-computing scenario, in which each computing device is allocated to the same size computation task.

Recently, some researchers have started to investigate the more general heterogeneous distributed computing scenario, in which computing devices can have different capabilities. For instance, Reisizadeh *et al.* discussed the optimal task allocation scheme for heterogeneous devices that have various computation latencies [21]. Kiamari *et al.* investigated the MapReduce-based CDC in heterogeneous systems by designing file allocation and the optimal coding scheme to achieve the minimum communication load [34]. However, these existing works have not considered the security issue in the heterogeneous distributed computing scenarios.

About the resource consumption of coded distributed computing, Li *et al.* studied the trade-off between the computational load and the communication load [5]. Li *et al.* designed a unified coding framework to enable a trade-off between the computation latency and the communication load [6]. To effectively deal with "stragglers" in large-scale distributed linear transform problems, Wang *et al.* designed the *Diagonal Code* to achieve minimum recovery threshold and low computation load [23]. In [35], Yan *et al.* analyzed the optimal tradeoff between the storage, the computation, and the communication for the coded distributed computing model proposed in [5]. Mallick *et al.* proposed a rateless fountain coding strategy to reduce latency while reducing computational redundancy and decoding complexity [24]. However, these existing studies have not comprehensively analyzed the total resource consumption, i.e., storage, computing and transmission, of the coded distributed computing, especially, in the secure coded distributed computing.

In this paper, we consider a general edge computing scenario, in which computing devices are heterogeneous, dynamic, and resource-limited. For such a scenario, the task allocation should be designed based on these characteristics and jointly studied with the secure coding scheme design. To this end, we jointly study two highly-coupled problems, i.e., task allocation and secure coding scheme, which not only achieves the ITS but also minimizes the total resource consumption, when different edge devices have different costs for storage, computing and transmission.

## 7 Conclusion

In this paper, we have investigated a secure coded computing problem in heterogeneous edge computing, with the objective to *minimize the total resource usage*, by jointly studying the *task allocation* that assigns data blocks in a computing task to edge devices, and the *linear code design* that generates data blocks by encoding the original data with random information. Specifically, we first theoretically analyzed the necessary conditions for the optimal solution. Based on the theoretical analysis, we developed an efficient task allocation algorithm to obtain a set of selected edge devices and the number of coded vectors allocated to them. Using the task allocation results, we designed two secure coded computing schemes for both the case with redundant computation and the case without redundant computation, and we further proved the feasibility and the optimality of these schemes. Finally, we conducted extensive simulation experiments that demonstrate the effectiveness of the proposed schemes. In the future, we will implement the proposed MCSCEC scheme in real edge computing systems and study a more general case that edge devices can attack cooperatively.

## Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under Grants 62072321, 61672370, in part by National Science Foundation under Grant CNS-1730325, Hong Kong Research Grant Council under Grant GRF 11211519, in part by European Union's Horizon 2020 under Grant 952644, in part by Six Talent Peak Project of Jiangsu Province under Grant XYDXX-084, in part by Suzhou Planning Project of Science and Technology under Grant SNG2020073, SS202023, and in part by Tang Scholar of Soochow University and the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD). A conference version has been published in *the 39th IEEE International Conference on Distributed Computing Systems* (IEEE ICDCS) 2019 [1]. This version contains at least 50 percent additional technical materials.

## References

[1] C. Cao *et al.*, "Optimal task allocation and coding design for secure coded edge computing," in *Proc. Int. Conf. Distrib. Comput. Syst.*, 2019, pp. 1083–1093.

[2] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2322–2358, Fourth Quarter 2017.

[3] S. Li, Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, "Coded distributed computing: Fundamental limits and practical challenges," in *Proc. Asilomar Conf. Signals Syst. Comput.*, 2016, pp. 509–513.

[4] K. Lee, M. Lam, R. Pedarsani, D. Papailiopoulos, and K. Ramchandran, "Speeding up distributed machine learning using codes," *IEEE Trans. Inf. Theory*, vol. 64, no. 3, pp. 1514–1529, Mar. 2018.

[5] S. Li, M. A. Maddah-Ali, Q. Yu, and A. S. Avestimehr, "A fundamental tradeoff between computation and communication in distributed computing," *IEEE Trans. Inf. Theory*, vol. 64, no. 1, pp. 109–128, Jan. 2018.

[6] S. Li, M. A. Maddah-Ali, and A. S. Avestimehr, "Coding for distributed fog computing," *IEEE Commun. Mag.*, vol. 55, no. 4, pp. 34–40, Apr. 2017.

[7] Q. Yu, S. Li, M. A. Maddah-Ali, and A. S. Avestimehr, "How to optimally allocate resources for coded distributed computing?," in *Proc. Int. Conf. Commun.*, 2017, pp. 1–7.

[8] S. Li, M. A. Maddah-Ali, and A. S. Avestimehr, "Coded MapReduce," in *Proc. Annu. Allerton Conf. Commun. Control Comput.*, 2015, pp. 964–971.

[9] S. Li, M. A. Maddah-Ali, and A. S. Avestimehr, "Communication-aware computing for edge processing," in *Proc. Int. Symp. Inf. Theory*, 2017, pp. 2885–2889.

[10] R. Bitar, P. Parag, and S. E. Rouayheb, "Minimizing latency for secure distributed computing," in *Proc. Int. Symp. Inf. Theory*, 2017, pp. 2900–2904.

[11] R. Bitar, P. Parag, and S. E. Rouayheb, "Minimizing latency for secure coded computing using secret sharing via staircase codes," *Arxiv Preprint*, 2018. [Online]. Available: https://arxiv.org/abs/1802.02640

[12] H. Yang and J. Lee, "Secure distributed computing with straggling servers using polynomial codes," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 1, pp. 141–150, Jan. 2019.

[13] C. Wang, K. Ren, and J. Wang, "Secure optimization computation outsourcing in cloud computing: A case study of linear programming," *IEEE Trans. Comput.*, vol. 65, no. 1, pp. 216–229, Jan. 2016.

[14] K. Lee, C. Suh, and K. Ramchandran, "High-dimensional coded matrix multiplication," in *Proc. Int. Symp. Inf. Theory*, 2017, pp. 2418–2422.

[15] S. Dutta, V. Cadambe, and P. Grover, "Short-Dot: Computing large linear transforms distributedly using coded short dot products," *IEEE Trans. Inf. Theory*, vol. 65, no. 10, pp. 6171–6193, Oct. 2019.

[16] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, "Polynomial codes: An optimal design for high-dimensional coded matrix multiplication," 2017. [Online]. Available: https://arxiv.org/abs/1705.10464

[17] U. Sheth *et al.*, "An application of storage-optimal MatDot codes for coded matrix multiplication: Fast k-nearest neighbors estimation," in *Proc. Int. Conf. Big Data*, 2018, pp. 1113–1120.

[18] S. Dutta, Z. Bai, H. Jeong, T. M. Low, and P. Grover, "A unified coded deep neural network training strategy based on generalized PolyDot codes," in *Proc. Int. Symp. Inf. Theory*, 2018, pp. 1585–1589.

[19] S. Dutta, V. Cadambe, and P. Grover, "Coded convolution for parallel and distributed computing within a deadline," in *Proc. Int. Symp. Inf. Theory*, 2017, pp. 2403–2407.

[20] S. Dhakal, S. Prakash, Y. Yona, S. Talwar, and N. Himayat, "Coded federated learning," in *Proc. IEEE Globecom Workshops*, 2019, pp. 1–6.

[21] A. Reisizadeh, S. Prakash, R. Pedarsani, and A. S. Avestimehr, "Coded computation over heterogeneous clusters," *IEEE Trans. Inf. Theory*, vol. 65, no. 7, pp. 4227–4242, Jul. 2019.

[22] J. Kakar, S. Ebadifar, and A. Sezgin, "On the capacity and straggler-robustness of distributed secure matrix multiplication," *IEEE Access*, vol. 7, no. 1, pp. 45 783–45 799, 2019.

[23] S. Wang, J. Liu, N. Shroff, and P. Yang, "Fundamental limits of coded linear transform," 2018. [Online]. Available: http://arxiv.org/abs/1804.09791

[24] A. Mallick, M. Chaudhari, U. Sheth, and J. Gauri, "Rateless codes for near-perfect load balancing in distributed matrix-vector multiplication," 2019. [Online]. Available: https://arxiv.org/abs/1804.10331

[25] P. Mohassel, "Efficient and secure delegation of linear algebra," Cryptology ePrint Archive, Report 2011/605, 2011. [Online]. Available: https://eprint.iacr.org/2011/605

[26] Z. Brakerski and V. Vaikuntanathan, "Efficient fully homomorphic encryption from (standard) LWE," *Soc. Ind. Appl. Math. J. Comput.*, vol. 43, no. 2, pp. 831–871, 2014.

[27] S. Halevi and V. Shoup, "Faster homomorphic linear transformations in HElib," in *Annual International Cryptology Conference*. Cham, Switzerland: Springer, 2018.

[28] R. Gennaro, C. Gentry, and B. Parno, "Non-interactive verifiable computing: Outsourcing computation to untrusted workers," in *Proc. Annu. Cryptol. Conf.*, 2010, pp. 465–482.

[29] N. Cai and T. Chan, "Theory of secure network coding," *Proc. IEEE*, vol. 99, no. 3, pp. 421–437, Mar. 2011.

[30] Stadia. Accessed: 2019. [Online]. Available: https://stadia.google.com

[31] KubeEdge. Accessed: 2020. [Online]. Available: https://kubeedge.io/

[32] Baetyl. Accessed: 2020. [Online]. Available: https://baetyl.io/

[33] R. Zhao *et al.*, "Weakly secure coded distributed computing," in *Proc. Int. Conf. Ubiquitous Intell. Comput.*, 2018, pp. 603–610.

[34] M. Kiamari, C. Wang, and A. S. Avestimehr, "On heterogeneous coded distributed computing," in *Proc. IEEE Global Commun. Conf.*, 2017, pp. 1–7.

[35] Q. Yan, S. Yang, and M. Wigger, "Storage, computation, and communication: A fundamental tradeoff in distributed computing," 2018. [Online]. Available: https://arxiv.org/abs/1806.07565

**Jin Wang** (Member, IEEE) received the BS degree from the Ocean University of China, Qingdao, China, in 2006, and the PhD degree in computer science jointly awarded by the City University of Hong Kong, Hong Kong and the University of Science and Technology of China, Hefei, China, in 2011. He is currently a professor with the Department of Computer Science and Technology, Soochow University, Suzhou, China. His research interests include edge computing, linear coding, and network security.

**Chunming Cao** received the MS degree from Soochow University, Suzhou, China, in 2020. He is currently a researcher with China Mobile (Suzhou) Software Technology Company, Ltd., Suzhou, China. His research interests include security and edge computing.

**Jianping Wang** (Member, IEEE) received the BS and MS degrees in computer science from Nankai University, Tianjin, China, in 1996 and 1999, respectively, and the PhD degree in computer science from the University of Texas at Dallas, Richardson, Texas, in 2003. She is currently a professor with the Department of Computer Science, City University of Hong Kong. Her research interests include security and motion planning in autonomous driving, dependable networking edge computing, and data center networks.

**Kejie Lu** (Senior Member, IEEE) received the BSc and MSc degrees from the Beijing University of Posts and Telecommunications, Beijing, China, in 1994 and 1997, respectively, and the PhD degree in electrical engineering from the University of Texas at Dallas, Richardson, Texas, in 2003. In July 2005, he joined the University of Puerto Rico at Mayaguez, Mayaguez, Puerto Rico, where he is currently a professor with the Department of Computer Science and Engineering. His research interests include computer and communication networks, cyber-physical system, and network-based computing.

**Admela Jukan** (Member, IEEE) was a visiting scientist with Bell Labs, Holmdel, New Jersey, in 1999 and 2000, with the University of Illinois at Urbana-Champaign (UIUC), in 2004, and with MIT in 2015. From 2002 to 2004, she was a program director in computer and networks system research with the National Science Foundation (NSF), Arlington, Virginia. She was research faculty with the Institut National de la Recherche Scientifique (INRS) and Georgia Tech (GaTech). She is currently a chair professor of communication networks with Technische Universität Braunschweig, Germany. She was a recipient of the IBM Innovation Award, in 2009, the Award of Excellence for the BMBF/CELTIC project 100Gb Ethernet in 2013, and the IEEE Optical Network Technical Committee Service Award, in 2018. She has chaired and co-chaired several international conferences, including the IEEE/ACM IWqoS, IEEE ANTS, IFIP ONDM, IEEE ICC, and IEEE GLOBECOM. She serves as a senior editor of the *IEEE Journal of Selected Areas in Communications* (JSAC). She is co-editor-in-chief of the *Journal on Optical Switching and Networking* (OSN) (Elsevier).

**Wei Zhao** (Fellow, IEEE) received the undergraduate degree in physics from Shaanxi Normal University, Xi'an, China, in 1977, and the MSc and PhD degrees in computer and information sciences from the University of Massachusetts at Amherst, Amherst, Massachusetts, in 1983 and 1986, respectively. An internationally renowned scholar, he has served important leadership roles in academic including the chief research officer (i.e., vice president for research) with the American University of Sharjah, the chair of Academic Council, CAS Shenzhen Institute of Advanced Technology, the eighth rector (i.e., president) with the University of Macau, the dean of science with Rensselaer Polytechnic Institute, the director for the Division of Computer and Network Systems, U.S. National Science Foundation, and the senior associate vice president for research with Texas A&M University. He was also on faculty with Shaanxi Normal University, Amherst College, Adelaide University, and Shenzhen Institute of Advanced Technologies. He has made significant contributions in cyber-physical systems, distributed computing, real-time systems, computer networks, and cyberspace security. He led the effort to define research agenda of and to create the very first funding program for cyber-physical systems, when he served as the NSF CNS Division director in 2006. His research group has received numerous awards. Their research results have been adopted in the standard of SAFENET (Survivable Adaptable Fiber Optic Embedded Network). In 2011, he was named by the Chinese Ministry of Science and Technology as the chief scientist of the national 973 Internet of Things Project. He was awarded the Lifelong Achievement Award by the Chinese Association of Science and Technology, in 2005. In 2007, he was honored with the Overseas Achievement Award by the Chinese Computer Federation. He has been conferred honorable doctorates by 12 universities in the world and academician of the International Eurasian Academy of Sciences.