Satisfaction-Maximized Secure Computation Offloading in Multi-Eavesdropper MEC Networks

Shumei Liu, Yao Yu, *Member, IEEE,* Lei Guo, *Senior Member, IEEE,* Phee Lep Yeoh, *Member, IEEE,* Branka Vucetic, *Life Fellow, IEEE,* Yonghui Li, *Fellow, IEEE,* and Trung Q. Duong, *Senior Member, IEEE*

***Abstract*—In this paper, we consider a mobile edge computing (MEC)-based secure computation offloading system, and design a practical multi-eavesdropper model including two specific scenarios of non-colluding and colluding eavesdropping. Furthermore, we design a requirement satisfaction model by exploring practical variations in user request patterns for security provisioning, delay reduction and energy saving. Based on these, we propose a satisfaction-maximized secure computation offloading (SMax-SCO) scheme, and then formulate an optimization problem aiming at maximizing users' requirement satisfactions subject to secrecy offloading rate, tolerable delay, task workload and maximum power constraints. Since the optimization problem is nonconvex, we present an efficient successive convex approximation (SCA)-based algorithm to obtain suboptimal solutions. We demonstrate that the proposed SMax-SCO scheme achieves a significant improvement in security performance and requirement satisfaction compared with existing schemes. Moreover, we conclude that SMax-SCO can resist eavesdropping attacks of multiple eavesdroppers and even colluding eavesdroppers.**

***Index Terms*—Mobile edge computing (MEC), computation offloading, multiple eavesdroppers, security provisioning, delay and energy consumption**

I. Introduction

THE past decade has witnessed explosive demand for Internet of Things (IoT) networks to support a large number of computation-intensive applications such as autonomous driving, interactive gaming, and augmented reality [1, 2]. Such applications usually require large amounts of computation resources, which is challenging to mobile devices with limited computing capabilities [3]. Mobile edge computing (MEC) has been proposed to address this challenge by deploying MEC servers at wireless access points (APs) to provide high computation resources at the IoT network edge [4]. MEC enables mobile devices to offload their computation tasks via wireless transmissions to the APs for efficient execution [5]. Specifically, by exploiting the large amounts of computation resources available at APs, MEC can support a variety of emerging computation-intensive applications and bring significant benefits to users, such as lower service delay, lower energy consumption and better quality of experience (QoE) [6, 7].

In MEC-based computation offloading systems, there are two offloading modes, namely binary and partial offloading [8]. In binary mode [9], the computation tasks are indivisible, and each user can either complete them using local computing or offload them entirely to the MEC server for processing. In partial mode [10], each user can divide their computation tasks into two parts, which can be processed in parallel by offloading one part to the MEC server and completing the other part using local computing. In this paper, we consider the more general and flexible mode of partial offloading. Computation offloading in MEC systems with partial offloading requires careful allocation of various wireless resources (e.g., transmit power, bandwidth and computing resources) and offloading rate [11]. It is known that different task offloading requirements correspond to strikingly different resource allocation solutions [12]. For mobile users, security provisioning is an essential prerequisite for MEC-enabled applications, and service performance regarding processing delay and energy consumption is important to promote effective long-term task offloading for IoT users.

Security provisioning is essential due to the broadcast nature of wireless communications, where the computation tasks offloaded from mobile devices to APs via wireless channel could be overheard by malicious attackers nearby [13-15]. Moreover, traditional cryptographic techniques are challenging to implement in mobile devices with limited computing capacities [16]. Against this background, extensive studies have explored the advantages of physical layer security (PLS) [17, 18]. PLS aims to reinforce the security of communication systems by exploiting differences in channel conditions between each legitimate user and eavesdropper, which is a viable solution to guarantee the security of wireless offloading [18]. Unlike encryption-based algorithms, the security metric for PLS does not rely on any computationally expensive encryption techniques. In other words, the secrecy level provided by PLS will not be compromised by the mobile devices'

Manuscript received June 28, 2021; revised August 27, 2021; accepted November 08, 2021. This work was supported in part by the National Key Research and Development Program under Grant 2018YFB1702003, in part by the National Natural Science Foundation of China under Grant 62171113, 61941113 and 61775033, and in part by the Fundamental Research Funds for the Central Universities under Grant N2116011 and N2116003. *(Corresponding author: Yao Yu.)*

S. Liu and Y. Yu are with both the School of Computer Science and Engineering, Northeastern University and Key Laboratory of Intelligent Computing in Medical Image, Ministry of Education, Northeastern University, Shenyang 110819, China (E-mail: liusmneu@163.com; yuyao@mail.neu.edu.cn).

L. Guo is with the School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China (E-mail: guolei@cqupt.edu.cn).

P. L. Yeoh, B. Vucetic and Y. Li are with the School of Electrical and Information Engineering, University of Sydney, Sydney NSW 2006, Australia (E-mail:{phee.yeoh; branka.vucetic; yonghui.li}@sydney.edu.au).

T. Q. Duong is with the School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, Belfast, BT7 1NN, U.K. (E-mail: trung.q.duong@qub.ac.uk).

limited computation resources [19]. Recently, many works on MEC-based computation offloading consider eavesdropping attacks and exploit the nature of wireless channels in PLS to enhance security [4, 5, 20-26]. Most of these works have focused on the eavesdropping scenario with only a single eavesdropper in MEC networks [4, 5, 20-24]. The authors in these works aimed at minimizing processing delay or energy consumption in a single-eavesdropper system, and ensure the security of offloaded tasks using PLS technologies. Actually, there are usually multiple eavesdroppers in realistic MEC networks. More recently in [25, 26], the authors considered a UAV-enabled MEC system with multiple non-colluding eavesdroppers. The authors investigated the security problems of maximizing secrecy capacity under some performance constraints, where one UAV helps mobile users to compute the offloaded tasks and the others act as jammers to suppress the non-colluding eavesdroppers. We note that previous works on secure computation offloading have not considered the worst-case scenario of colluding eavesdroppers in multi-eavesdropper scenarios. Furthermore, they have not considered the impact of various users' requirements for delay reduction and energy saving.

In practical IoT applications, mobile users always place great concerns about the performance experiences regarding delay reduction and energy saving under the premise of security provisioning. There has been substantial research on MEC resource management to optimize system delay [4] or energy consumption [5] while ensuring offloading security. To reduce energy consumption and extend the life cycle of devices, the authors in [5] and [22] investigated a weighted sum-energy consumption minimization problem with security provisioning. The authors in [20] and [21] defined secrecy energy efficiency (SEE) as the total number of secure computation offloading bits per Joule, which aims to maximize secrecy capacity and minimize devices' energy consumption. Moreover, to achieve excellent delay performance in IoT applications, the authors in [4] and [23] formulated the overall delay minimizing problem with security provisioning. We note that the degrees and urgencies of delay and energy consumption requirements may vary greatly in different scenarios and situations. For example, the applications in vehicular networks and tactile Internet usually have very stringent requirements to guarantee low delay performance, while the requirements on energy consumption are relatively less strict [27]. Moreover, a device with short battery life always pays more attention to energy saving than delay reduction [28]. However, the above existing works on MEC-based secure computation offloading only support fixed service of minimizing delay or energy consumption with security provisioning. Given this background, how to dynamically balance the tradeoff performance between delay reduction and energy saving on the basis of security provisioning is a very challenging problem, which directly affects users' requirement satisfaction in MEC systems.

To sum up, due to the security vulnerability of wireless communication and the wide variety of IoT applications, typical users usually have both security requirements and various task preferences regarding delay and energy consumption. However, few existing works have jointly addressed these concerns, and most have considered relatively ideal eavesdropping scenarios. Motivated by these challenges, in this paper, we consider a practical requirement-based multi-eavesdropper computation offloading system. We focus on the specific requirements of each user regarding security, delay and energy consumption for their tasks. Then, we aim to maximize users' requirement satisfaction by providing each user with a unique and optimal computation offloading strategy. In this context, we provide an optimal solution to satisfy both the delay-energy balance and security requirements by answering the following three questions: 1) What is the volume of the computation tasks that should be offloaded? 2) How much power should be assigned to the offloaded task? 3) What is the transmission rate that can ensure offloading security? As such, unlike the fixed service strategies in the existing works, we flexibly design a requirement-adaptive service pattern by considering users' current task preferences and eavesdropping scenarios, and we believe the problems formed and solved in this work are of great practical significance. The main technical contributions of this paper are as follows:

1) To the best of our knowledge, this is the first work to study the delay and energy consumption balance for the secure computation offloading in MEC-based multi-eavesdropper computation offloading systems, and further provide offloading solutions for users with various individual requirements.
2) We design a multi-eavesdropper model including two practical scenarios to reflect different eavesdropping capabilities: the non-colluding (independent) eavesdroppers who do not share received information and colluding (cooperating) eavesdroppers who can combine and jointly process the received information.
3) We design a requirement satisfaction model by exploring practical variations in the user request patterns to security provisioning, delay reduction and energy saving. Based on the above two models, we propose a satisfaction-maximized secure computation offloading (SMax-SCO) scheme.
4) We formulate an optimization problem aiming at maximizing the requirement satisfactions of users subject to secrecy offloading rate, tolerable delay, task workload and maximum power constraints. To tackle the nonconvex optimization problem, we develop an efficient successive convex approximation (SCA)-based algorithm to obtain the solutions.
5) Through exhaustive numerical simulations, we demonstrate that our proposed SMax-SCO scheme significantly outperforms other benchmarks DMin-SCO [4] and EMin-SCO [5] regarding the security performance and user's requirement satisfaction. Furthermore, we show that SMax-SCO can guarantee offloading security when there are multiple eavesdroppers or even colluding eavesdroppers.

The structure of the remainder of this paper is as follows. Section II introduces the system models of secure computation offloading. Section III proposes a SMax-SCO scheme. Specifically, we introduce the multi-eavesdropper and requirement

satisfaction models in detail, and the problem formulation of SMax-SCO. In Section IV, we present a SCA-based algorithm to solve the resulting nonconvex problem. In Section V, we provide a detailed overview description of our SMax-SCO scheme. We evaluate our proposed SMax-SCO scheme by experimental simulation in Section VI. We conclude and discuss the paper in Section VII.

II. System Models and Problem Formulation

A. System Model

As shown in Fig. 1, we consider an illustrative MEC-based multi-eavesdropper computation offloading system where user i offloads its computation tasks to an AP in the presence of $M > 1$ malicious eavesdroppers. Let $\mathcal{M} = \{1, ..., m, ..., M\}$ denote the set of eavesdroppers. All nodes are assumed to be equipped with a single antenna. The AP is integrated with a MEC server, and thus has a high amount of computation and energy resources. User i with limited computation and energy resources has some computation-intensive and delay-aware tasks that need to be completed. Its computation tasks are characterized by (L_i, D_i). L_i is the total number of bits to be processed in the computation tasks. D_i is the tolerable delay, which means the tasks must be completed within this time. We focus on secure partial offloading in this paper. In other words, the computation task of user i can be partitioned into two portions with l_i and $L_i - l_i$ bits, which are locally computed at user i and securely offloaded to the AP in the presence of multiple eavesdroppers, respectively. Physical layer security technology is adopted to mitigate eavesdropping and improve security during the offloading process.

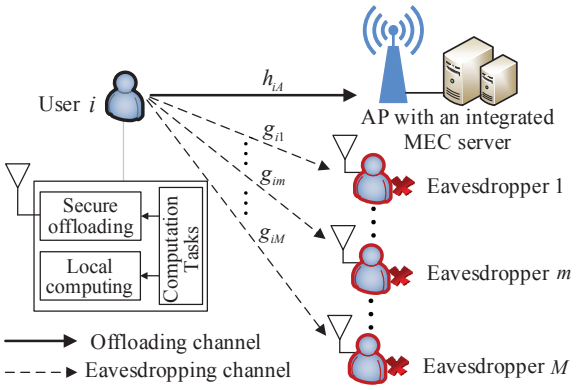


Fig. 1: MEC-based multi-eavesdropper computation offloading system

We consider a quasi-static Rayleigh fading channel model, in which the channels remain constant within each offloading process. Then, the channel gains from user i to the AP and the eavesdropper m are denoted by h_{iA} and g_{im}, respectively. Since the AP performs real-time control signal interactions with its serviced users, we further assume that the tasks information and the instantaneous channel state information (CSI) (i.e., h_{iA}) of each legitimate user are perfectly known by the AP. However, due to the fact that the eavesdroppers might intentionally hide their position, accurate instantaneous CSI (i.e., g_{im}) of each eavesdropper is difficult to obtain. Therefore, similarly to previous works [5, 13], we apply a deterministic CSI uncertainty model for g_{im}, where $g_{im} = \bar{g}_{im} + \Delta g_{im}, m \in \mathcal{M}$. Here, $\bar{g}_{im}$ is the estimated (i.e., average) channel gain of eavesdropper m at the AP and Δg_{im} is the estimation error.

B. Local Computing Model

For the local computing at user i, we use C_i to denote the number of central processing unit (CPU) cycles required to compute one bit of tasks. Hence, the total number of CPU cycles required for computing l_i bits is $C_i l_i$. We assume that f_i is the average CPU frequency (in CPU cycles per second) of user i. Then, the total computing time used for local computing is $\frac{C_i l_i}{f_i}$. Due to the delay limitation requirement of the computation tasks, the delay constraint of local computing at user i is given by

$$t_i^{\text{loc}} = \frac{C_i l_i}{f_i} \leq D_i. \tag{1}$$

Next, the energy consumption for local computing at user i is written as

$$E_i^{\text{loc}} = \varsigma C_i l_i f_i^2, \tag{2}$$

where $\varsigma > 0$ is the effective capacitance coefficient that depends on the chip processor architecture of user i's device.

C. Secure Offloading Model

Apart from local computing, user i can offload the remaining computation tasks of $L_i - l_i$ bits to the AP for computing. As stated before, we apply physical layer security to quantify the communications security against eavesdroppers when user i offloads its computation tasks to the AP. According to the Shannon's channel capacity theory [29], the channel capacity $R_{i,\text{AP}}$ from user i to the AP and the capacity $R_{i,m}$ from user i to eavesdropper m are shown as

$$R_{i,\text{AP}} = B\log_2\left(1 + \frac{p_i h_{iA}}{n_A}\right) \tag{3}$$

$$R_{i,m} = B\log_2\left(1 + \frac{p_i g_{im}}{n_E}\right), \tag{4}$$

where B is the system bandwidth, p_i denotes the transmission power at user i for offloading the partial computation tasks, n_A and n_E are the background noise powers at the AP each eavesdropper, respectively.

Based on (3) and (4), the secrecy capacity $S_{i,m}^{\text{sec}}$ of user i against eavesdropper m is given by

$$S_{i,m}^{\text{sec}} = [R_{i,\text{AP}} - R_{i,m}]^+, \tag{5}$$

where $[x]^+$ represents $\max\{x, 0\}$. We denote r_{iA} as the offloading rate, which must satisfy the condition of $S_{i,m}^{\text{sec}} \geq r_{iA}$ to prevent eavesdropper m from eavesdropping on the offloading tasks.

Moreover, the energy consumption of user i during the offloading process is expressed as

$$E_i^{\text{off}} = p_i \frac{L_i - l_i}{r_{iA}}. \tag{6}$$

D. Edge Computing Model

For the edge computing at the AP, we use $C_{\rm AP}$ and $f_{\rm AP}$ to represent the number of required CPU cycles at the AP for computing one bit of tasks and its average CPU frequency, respectively. The time consumption at the AP is $t_i^{\rm AP} = \frac{C_{AP}(L_i - l_i)}{f_{AP}}$. To satisfy the delay limitation of user i's computation tasks, the delay constraint of edge computing at the AP is defined as

$$t_i^{\rm AP} + \frac{L_i - l_i}{r_{iA}} \leq D_i. \tag{7}$$

III. Proposed SMax-SCO Scheme

In this section, we propose a satisfaction-maximized secure computation offloading (SMax-SCO) scheme to accommodate users' various requirements on the computation tasks. In particular, we consider a multi-eavesdropper model to accommodate more general and practical wireless task environments. Besides the security provisioning, we consider specific requirements of users in terms of processing delay and energy consumption. Finally, we formulate the optimization problem for SMax-SCO.

A. Multi-Eavesdropper Model

We specifically consider two eavesdropping scenarios of non-colluding and colluding eavesdroppers. The colluding eavesdropping scheme includes partial colluding and complete colluding, where the latter is a special case of the former.

- ***Non-colluding eavesdropping***

In this case, each eavesdropper is considered to intercept the computation offloading messages independently and there are no information exchanges between them. As such, the secrecy capacity $S_{i,\tilde{m}}^{\rm sec}$ from user i to multiple non-colluding eavesdroppers is constrained by the best eavesdropper, which can be expressed as

$$\begin{aligned} S_{i,\tilde{m}}^{\rm sec} &= [R_{i,\rm AP} - R_{i,\tilde{m}}]^+ \\ &= \begin{cases} R_{i,\rm AP} - R_{i,\tilde{m}}, & \text{if } h_{iA} \geq \frac{n_A}{n_E} g_{i\tilde{m}} \\ 0, & \text{otherwise} \end{cases}, \end{aligned} \tag{8}$$

where $\tilde{m} = \arg\max_{m \in \mathcal{M}} (B\log_2(1 + \frac{p_i g_{im}}{n_E}))$ is the best eavesdropper among the multiple eavesdroppers. To ensure secure offloading of the computation tasks, the offloading rate r_{iA} of user i must satisfy $S_{i,\tilde{m}}^{\rm sec} \geq r_{iA}$.

- ***Partial and complete colluding eavesdropping***

In this case, we consider that the AP and other legitimate users can detect that some eavesdroppers are displaying colluding behaviours (e.g., gathering and exchanging eavesdropping information), and therefore the multi-eavesdropper scenario is regarded as partial colluding eavesdropping. In general, when the channel gains of multiple eavesdroppers to the legitimate users are poor, the eavesdroppers could collude to improve their detection of the computation offloading messages. In addition, the complete colluding eavesdropping scenario is the worst-case scenario when all identified eavesdroppers are involved in malicious colluding.

For the partial colluding eavesdropping scenario, let $\mathcal{C}_{\mathcal{M}} (\mathcal{C}_{\mathcal{M}} \subset \mathcal{M})$ denote the set of colluding eavesdroppers. In this case, these eavesdroppers can be regarded as one super eavesdropper m' with $|\mathcal{C}_{\mathcal{M}}|$ distributed antennas [30], where $|\mathcal{C}_{\mathcal{M}}|$ is the number of colluding eavesdroppers in set $\mathcal{C}_{\mathcal{M}}$. Similarly to [30], the channel gain from user i to the super eavesdropper m' is equivalent to the sum of the channel gains of the colluding eavesdroppers in $|\mathcal{C}_{\mathcal{M}}|$, i.e., $g_{im'} = \sum_{j \in \mathcal{C}_{\mathcal{M}}} g_{ij}$. As such, the best channel capacity $R_{i,\rm Coll}$ among the eavesdroppers for this partial colluding eavesdropping scenario can be expressed as

$$R_{i,\rm Coll} = B\log_2(1 + p_i \gamma_{i,\rm Coll}) \tag{9}$$

$$\gamma_{i,\rm Coll} = \begin{cases} \frac{g_{im'}}{n_E}, & \text{if } g_{im'} \geq \max\{g_{im}\} \\ \max\{\frac{g_{im}}{n_E}\}, & \text{otherwise} \end{cases}, \forall m \in \mathcal{M} \backslash \mathcal{C}_{\mathcal{M}}, \tag{10}$$

where $\mathcal{M} \backslash \mathcal{C}_{\mathcal{M}}$ means removing $\mathcal{C}_{\mathcal{M}}$ from $\mathcal{M}$, and is the set of the remaining eavesdroppers (i.e., eavesdroppers who are not involved in the collusion) besides the colluding eavesdroppers in the partial colluding eavesdropping scenario. Specifically, when the channel gain of the super eavesdropper m' is larger than that of any of the remaining eavesdroppers in set $\mathcal{M} \backslash \mathcal{C}_{\mathcal{M}}$, the super eavesdropper m' becomes the best eavesdropper $\tilde{m}$. Otherwise, the best eavesdropper $\tilde{m}$ is the one with the largest channel gain in set $\mathcal{M} \backslash \mathcal{C}_{\mathcal{M}}$.

Then, secrecy performance $S_{i,\tilde{m}}^{\rm sec}$ from user i to the best eavesdropper $\tilde{m}$ is

$$\begin{aligned} S_{i,\tilde{m}}^{\rm sec} &= [R_{i,\rm AP} - R_{i,\rm Coll}]^+ \\ &= \begin{cases} R_{i,\rm AP} - R_{i,\rm Coll}, & \text{if } h_{iA} \geq n_A \gamma_{i,\rm Coll} \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \tag{11}$$

To ensure security, the offloading rate r_{iA} must satisfy $S_{i,\tilde{m}}^{\rm sec} \geq r_{iA}$.

B. Requirement Satisfaction Model

For MEC users, security provisioning, delay reduction and energy saving are three main concerns in computation offloading systems. Hence, in this paper we design a requirement satisfaction model including the above three concerns. Among them, security provisioning is the basic requirement of users during the computation offloading process. We introduce a binary security variable $x_{iA} = \{0, 1\}$ to represent the offloading security of user i's computation tasks. $x_{iA} = 1$ means that the security of the offloading process can be guaranteed, i.e., $S_{i,\tilde{m}}^{\rm sec} \geq r_{iA}$, and otherwise $x_{iA} = 0$, i.e., $S_{i,\tilde{m}}^{\rm sec} < r_{iA}$.

For energy consumption, we focus on user since the MEC server is typically powered by grid energy and has sufficient energy resources. The energy consumption of each user can be expressed in two parts given by local computing and secure computation offloading. From (2) and (6), the total energy consumption E_i of user i with secure partial offloading is given by

$$E_i = E_i^{\rm loc} + E_i^{\rm off} = \varsigma C_i l_i f_i^2 + p_i \frac{L_i - l_i}{r_{iA}}. \tag{12}$$

If user i does not offload computation tasks to the AP, the security variable will be $x_{iA} = 1$ and the total energy consumption $E_i^{\rm all-loc}$ using only local computing can be calculated as

$$E_i^{\rm all-loc} = \varsigma C_i L_i f_i^2. \tag{13}$$

As mentioned before, this paper considers the service requirements of users regarding delay reduction and energy saving on the basis of ensuring offloading security. Since the units and dimensions of time and energy are quite different, a simple weighted combination cannot provide users with a preference index to fairly balance the delay reduction and energy saving. Therefore, it is necessary to address the dimensional differences between them and provide on-demand service. To this end, we introduce a normalized energy consumption γ_i^E and a normalized delay γ_i^D for user i when using secure partial computation offloading, which is given by

$$\gamma_i^E = \frac{E_i}{E_i^{\mathrm{all-loc}}}. \tag{14}$$

$$\gamma_i^D = \frac{\max\left(t_i^{\mathrm{loc}}, t_i^{\mathrm{AP}} + \frac{L_i - l_i}{r_{iA}}\right)}{D_i},\ \gamma_i^D \leq 1, \tag{15}$$

where γ_i^D should be less than or equal to one to comply with the user i's delay limitation for the computation task. By doing so, γ_i^E and γ_i^D can intuitively reflect the degrees of energy saving and delay reduction due to secure partial computation offloading, respectively.

According to the above analysis, we further define a *Requirement Satisfaction (RS)* index including security provisioning, delay reduction and energy saving for user i as

$$RS(i) = \frac{x_{iA}}{\alpha_i \gamma_i^E + (1-\alpha_i)\gamma_i^D}. \tag{16}$$

We highlight that security provisioning is a basic requirement for all users during the computation offloading process. Therefore, each user i assigns the value of x_{iA} as 1. Besides, $0 \leq \alpha_i \leq 1$ is a control weight, and its value is given by user i according to his/her device state and current tasks preferences regarding delay and energy consumption. As such, we find that maximizing the requirement satisfaction of each user is equivalent to minimizing the costs of normalized processing delay and energy consumption of the user for completing tasks while ensuring security (i.e., $x_{iA} = 1$).

Obviously, when a user has energy-intensive tasks or has limited battery power, he/she would like to choose a larger α_i to save more energy. Conversely, when a user is participating in some delay-sensitive applications (e.g., online games), he/she may set a smaller α_i to reduce the processing delay. Therefore, the value of α_i greatly affects the offloading decisions of the computation tasks. To meet the specific requirement of each user, we allow each user to freely assign weights α_i and x_{iA}. To illustrate this, we present the following three-user example:

1) User 1 has delay-sensitive tasks, and he/she places a higher priority on the tasks' processing delay than energy consumption. The security offloading process is a further consideration due to the presence of eavesdroppers in the system. Therefore, user 1 sets $0 \leq \alpha_1 < \frac{1}{2}$ and $x_{iA} = 1$, where the setting of $\alpha_1 = 0$ and $x_{iA} = 1$ means that the goal is to minimize the processing delay with security provisioning [4];
2) User 2 has computation tasks, and he/she places equal priority on energy consumption and delay during the secure offloading process. Therefore, user 2 sets $x_{iA} = 1$ and $\alpha_2 = \frac{1}{2}$;
3) User 3 has energy-intensive tasks and is in a low battery state, thus he/she places higher priority on energy consumption than delay during the secure offloading process. Therefore, user 3 sets $x_{iA} = 1$ and $\frac{1}{2} < \alpha_3 \leq 1$, where the setting of $\alpha_3 = 1$ and $x_{iA} = 1$ means that the goal is to minimize energy consumption while meeting the delay constraint and ensuring offloading security [5].

C. Problem Formulation for SMax-SCO

In the proposed SMax-SCO scheme, we aim to provide computation offloading solutions for participating users according to the user's requirements including security, delay and energy consumption. In our optimization problem, the goal is to find an optimal allocation decision set $\{l_i, p_i, r_{iA}\}$ for each user i's computation tasks, which maximizes the requirement satisfaction $RS(i)$. Specifically, we aim to minimize the computation cost of each user regarding delay and energy consumption while ensuring offloading security in the presence of eavesdropper $\tilde{m}$. Here, $\tilde{m}$ refers to the best eavesdropper among multiple eavesdroppers in the non-colluding eavesdropping scenario, or among the super eavesdropper and the remaining eavesdroppers in the colluding scenario. To this end, the optimization problem can be formulated as

$$\begin{aligned}
\underline{\textbf{(P1)}}\text{:}\ \max_{l_i, p_i, r_{iA}}\ & RS(i) && \text{(17a)}\\
\text{s.t.}\ & t_i^{\mathrm{loc}} \leq D_i, && \text{(17b)}\\
& t_i^{\mathrm{AP}} + \frac{L_i - l_i}{r_{iA}} \leq D_i, && \text{(17c)}\\
& 0 \leq l_i \leq L_i, && \text{(17d)}\\
& 0 \leq p_i \leq p_i^{\max}, && \text{(17e)}
\end{aligned}$$

where in (17a), x_{iA} is 1, corresponding to $S_{i,\tilde{m}}^{\mathrm{sec}} \geq r_{iA}$. It ensures the requirement of secure offloading in the presence of eavesdropper $\tilde{m}$. Constraints (17b) and (17c) ensure the delay limitation D_i. Constraint (17d) ensures the offloading size limit, and constraint (17e) restricts the transmission power of user i. Problem **(P1)** is a nonconvex problem due to the complexity of the objective function (17a) and the coupling of variables.

We introduce a variable $\mathcal{R}_i^{\mathrm{den}}$ to represent the denominator of $RS(i)$ in (16), i.e., $\mathcal{R}_i^{\mathrm{den}} \triangleq \alpha_i \gamma_i^E + (1-\alpha_i)\gamma_i^D$. $\mathcal{R}_i^{\mathrm{den}}$ can be seen as the computation cost of user i regarding the delay and energy consumption during the tasks completion. Then, we equivalently transform the above maximization problem to a minimization problem as

$$\begin{aligned}
\underline{\textbf{(P2)}}\text{:}\ \min_{l_i, p_i, r_{iA}}\ & \mathcal{R}_i^{\mathrm{den}} && \text{(18a)}\\
\text{s.t.}\ & t_i^{\mathrm{loc}} \leq D_i, && \text{(18b)}\\
& t_i^{\mathrm{AP}} + \frac{L_i - l_i}{r_{iA}} \leq D_i, && \text{(18c)}\\
& 0 \leq l_i \leq L_i, && \text{(18d)}\\
& 0 \leq p_i \leq p_i^{\max}, && \text{(18e)}\\
& S_{i,\tilde{m}}^{\mathrm{sec}} \geq r_{iA}, && \text{(18f)}
\end{aligned}$$

where (18f) is the equivalent form of $x_{iA} = 1$.

However, problem **(P2)** is still a nonconvex problem due to the coupling of variables in objective function (18a) and constraints (18c) and (18f), which is belong to NP-hard problems.

IV. Proposed SCA-Based Algorithm

In this section, we first derive the optimal secure offloading rate in Theorem 1 and define an auxiliary variable to approximately transform objective function (18a) and constraints (18c), (18f) in **(P2)**. Then, we discuss the advantages of the SCA approach from [6] and introduce several Lemmas. Finally, we transform the nonconvex problem into a convex one, and propose an efficient SCA-based algorithm to solve the optimization in **(P2)**.

A. Problem Transformation

We aim to transform problem **(P2)** into an equivalent but more tractable form. First, given a generally reasonable assumption of $S_{i,\tilde{m}}^{\mathrm{sec}} > 0$ (i.e., $n_E h_{iA} - n_A g_{i\tilde{m}} > 0$ always holds), we expand $S_{i,\tilde{m}}^{\mathrm{sec}}$ in (18f) based on (3)-(5) as

$$\begin{aligned} S_{i,\tilde{m}}^{\mathrm{sec}} &= B\log_2(1+\frac{p_i h_{iA}}{n_A}) - B\log_2(1+\frac{p_i g_{im}}{n_E}) \\ &= B\log_2(\frac{n_A n_E + n_E h_{iA} p_i}{n_A n_E + n_A g_{i\tilde{m}} p_i}) \end{aligned} \tag{19}$$

Then based on (19), constraint (18f) can be rewritten as

$$r_{iA} \le B\log_2\left(\frac{n_A n_E + n_E h_{iA} p_i}{n_A n_E + n_A g_{i\tilde{m}} p_i}\right). \tag{20}$$

Next, our solution for transforming problem **(P2)** relies on the following *Theorem*.

Theorem 1 (Optimal Offloading Rate r_{iA}^*): For the optimal solution of problem **(P2)**, the optimal offloading rate $r_{i,A}^*$ for minimizing $\mathcal{R}_i^{\mathrm{den}}$ should satisfy

$$r_{iA}^* = B\log_2\left(\frac{n_A n_E + n_E h_{iA} p_i}{n_A n_E + n_A g_{i\tilde{m}} p_i}\right). \tag{21}$$

Proof: Refer to Appendix. ∎

Based on *Theorem* 1, we introduce an auxiliary variable z_i for user i as

$$z_i \triangleq \max(t_i^{\mathrm{loc}}, t_i^{AP} + \frac{L_i - l_i}{B\log_2(\frac{n_A n_E + n_E h_{iA} p_i}{n_A n_E + n_A g_{i\tilde{m}} p_i})}). \tag{22}$$

We can linearize the processing delay term in $\mathcal{R}_i^{\mathrm{den}}$ using z_i. Based on z_i and *Theorem* 1, problem **(P2)** can be reformulated as (23), shown at the bottom of this page.

However, problem **(P3)** is still a nonconvex optimization problem because the objective function (23a) and constraint (23c) are nonconvex.

B. SCA-Based Algorithm

In this subsection, we develop an SCA-based optimization algorithm to transform the nonconvex objective function and constraint into suitable convex approximations. We specifically introduce how to build the convex approximation for the nonconvex terms in problem **(P3)** while preserving the local first-order behaviour of the original nonconvex problem. Before developing our SCA-based algorithm, we first introduce the following three Lemmas from [8].

Lemma 1: [8, Example 4]: Suppose that g has a separable structure as the product of two convex and non-negative real-valued functions f_1 and f_2, i.e., $g(x,y) = f_1(x)f_2(y)$. For any (x_0, y_0) in the domain of g, a convex upper bound $\tilde{g}(x,y;x_0,y_0)$ of function $g(x,y)$ can be written as

$$\begin{aligned} &\tilde{g}(x,y;x_0,y_0) \\ &\quad\triangleq \frac{1}{2}(f_1(x)+f_2(y))^2 - \frac{1}{2}(f_1(x_0)^2 + f_2(y_0)^2) \\ &\quad - f_1(x_0)f_1'(x_0)(x-x_0) - f_2(y_0)f_2'(y_0)(y-y_0). \end{aligned} \tag{24}$$

Then, we get $\tilde{g}(x,y;x_0,y_0) \ge g(x,y)$.

Lemma 2: [8, Example 6]: If no convexity whatsoever is present in $U(x)$, we can follow proximal-gradient methods for equivalently convex transformation, i.e., first order approximation. Hence, for any x_0 in the domain of U, a convex approximation $\tilde{U}(x;x_0)$ of function $U(x)$ can be written as

$$\tilde{U}(x;x_0) \triangleq \nabla_x U(x_0)(x-x_0) + \frac{\tau}{2}(x-x_0)^2, \tag{25}$$

where $\tau > 0$ is a positive constant. $\nabla_x U(x_0)$ denotes the partial gradient of U with respect to argument x evaluated at x_0.

Lemma 3: [8, Example 8]: Function U is often written as the product of functions. We consider here the product of two functions, i.e., $U(x,y) = h_1(x)h_2(y)$. Functions h_1 and h_2 are positive but not necessarily convex. For any (x_0,y_0) in the domain of $U(x,y)$, a convex approximation $\tilde{U}(x,y;x_0,y_0)$ of $U(x,y)$ can be written as

$$\tilde{U}(x,y;x_0,y_0) \triangleq \tilde{h}_1(x;x_0)h_2(y_0) + h_1(x_0)\tilde{h}_2(y;y_0), \tag{26}$$

where $\tilde{h}_1(x;x_0)$ and $\tilde{h}_2(y;y_0)$ are the legitimate approximations using *Lemma 2* above for h_1 and h_2.

$$\begin{aligned} \underline{\textbf{(P3)}}\text{:}\ \min_{p_i,l_i,z_i}\ & \hat{\mathcal{R}}_i^{\mathrm{den}} = \min_{p_i,l_i,z_i}\left(\frac{\alpha_i l_i}{L_i} + \frac{\alpha_i(L_i-l_i)}{\varsigma C_i L_i f_i^2}\frac{p_i}{B\log_2(\frac{n_A n_E + n_E h_{iA} p_i}{n_A n_E + n_A g_{i\tilde{m}} p_i})} + \frac{(1-\alpha_i)z_i}{D_i}\right) && \text{(23a)} \\ \text{s.t.}\ & t_i^{\mathrm{loc}} \le z_i \le D_i, && \text{(23b)} \\ & z_i \ge t_i^{\mathrm{AP}} + \frac{L_i - l_i}{B\log_2(\frac{n_A n_E + n_E h_{iA} p_i}{n_A n_E + n_A g_{i\tilde{m}} p_i})}, && \text{(23c)} \\ & 0 \le l_i \le L_i, && \text{(23d)} \\ & 0 \le p_i \le p_i^{\max}. && \text{(23e)} \end{aligned}$$

Based on the above *Lemmas*, we transform nonconvex constraint (23c) and objective function (23a) in problem **(P3)** into suitable approximations, and develop a SCA-based algorithm.

For constraint (23c), we observe that the second term of the right hand side is nonconvex, which we represent as $g(l_i, p_i)$. Here, $g(l_i, p_i)$ can be written as the product of two functions

$$g(l_i, p_i) = \frac{1}{B} f_1(l_i) f_2(p_i) \tag{27}$$

$$f_1(l_i) = L_i - l_i \tag{28}$$

$$f_2(p_i) = \frac{1}{\log_2(\frac{n_A n_E + n_E h_{iA} p_i}{n_A n_E + n_A g_{i\tilde{m}} p_i})}. \tag{29}$$

Obviously, $f_1(l_i)$ is an affine function. We next establish the convexity of function $f_2(p_i)$ in the following *Proposition*.

Proposition 1: Function $f_2(p_i)$ is convex with respect to p_i.

Proof: We define the denominator of function $f_2(p_i)$ as

$$f_2^{\text{den}}(p_i) = \log_2(\frac{n_A n_E + n_E h_{iA} p_i}{n_A n_E + n_A g_{i\tilde{m}} p_i}), \tag{30}$$

where $f_2^{\text{den}}(p_i) > 0$ due to the premise of $n_E h_{iA} - n_A g_{i\tilde{m}} > 0$. Then, the first derivative of function $f_2^{\text{den}}(p_i)$ is given by

$$\frac{df_2^{\text{den}}(p_i)}{dp_i} = \frac{n_A n_E (n_E h_{iA} - n_A g_{i\tilde{m}})}{\ln 2 (n_A n_E + n_E h_{iA} p_i)(n_A n_E + n_A g_{i\tilde{m}} p_i)}, \tag{31}$$

where due to $n_E h_{iA} - n_A g_{i\tilde{m}} > 0$, we can easily find that $\frac{df_2^{\text{den}}(p_i)}{dp_i} > 0$. Thus, $f_2^{\text{den}}(p_i)$ is an increasing function with respect to p_i. Moreover, since $n_A n_E (n_E h_{iA} - n_A g_{i\tilde{m}})$ is a constant greater than zero, we can further observe that $\frac{df_2^{\text{den}}(p_i)}{dp_i}$ is a decreasing function with respect to p_i.

Next, we derive the first derivative of function $f_2(p_i)$ as

$$\begin{aligned} \frac{df_2(p_i)}{dp_i} &= -\frac{\frac{n_A n_E (n_E h_{iA} - n_A g_{i\tilde{m}})}{\ln 2 (n_A n_E + n_E h_{iA} p_i)(n_A n_E + n_A g_{i\tilde{m}} p_i)}}{\left(\log_2(\frac{n_A n_E + n_E h_{iA} p_i}{n_A n_E + n_A g_{i\tilde{m}} p_i})\right)^2} \\ &= -\frac{\frac{df_2^{\text{den}}(p_i)}{dp_i}}{\left(f_2^{\text{den}}(p_i)\right)^2}, \end{aligned} \tag{32}$$

since $f_2^{\text{den}}(p_i)$ and $\frac{df_2^{\text{den}}(p_i)}{dp_i}$ are respectively a positive increasing and a decreasing functions, $\frac{df_2(p_i)}{dp_i}$ is a negative increasing function with respect to p_i. Consequently, we conclude that the increase of p_i results in an increased $\frac{df_2(p_i)}{dp_i}$. Hence, $f_2(p_i)$ is a convex function. ■

Based on *Proposition* 1, we conclude that $g(l_i, p_i)$ is the product of two convex and nonnegative functions. Hence, given a feasible solution $l_i(k)$ and $p_i(k)$ for the kth iteration of the SCA-based algorithm, we derive a convex upper bound $\tilde{g}(l_i, p_i; l_i(k), p_i(k))$ of $g(l_i, p_i)$ using *Lemma* 1, which is given by (33), shown at the bottom of this page.

In (33), $\tilde{g}(l_i, p_i; l_i(k), p_i(k))$ can equivalently replace $g(l_i, p_i)$, and then non-convex constraint (23c) can be transformed into a convex one.

For objective function (23a), the second term is nonconvex and we use $U(l_i, p_i)$ to represent it. As such, $U(l_i, p_i)$ can be written as the product of two functions as

$$U(l_i, p_i) = \frac{\alpha_i}{\varsigma C_i L_i f_i^2 B} h_1(l_i) \cdot h_2(p_i) \tag{34}$$

$$h_1(l_i) = L_i - l_i \tag{35}$$

$$h_2(p_i) = \frac{p_i}{\log_2(\frac{n_A n_E + n_E h_{iA} p_i}{n_A n_E + n_A g_{i\tilde{m}} p_i})}. \tag{36}$$

Obviously, $h_1(l_i)$ is an affine function, and $h_2(p_i)$ is a nonconvex function. According to *Lemma* 2, we derive a convex approximation of $h_2(p_i)$ as

$$\tilde{h}_2(p_i; p_i(k)) = h_2{}'(p_i(k))(p_i - p_i(k)) + \frac{\tau_1}{2}(p_i - p_i(k))^2, \tag{37}$$

where $\tau_1 > 0$ is a positive constant, and $h_2{}'(p_i(k))$ is given by equation (38), shown at the bottom of this page.

Based on (37) and (38), the transformed objective function $\hat{U}(l_i, p_i)$ can be written as a product of two convex functions

$$\hat{U}(l_i, p_i) = h_1(l_i) \tilde{h}_2(p_i; p_i(k)). \tag{39}$$

Then, given a feasible solution $l_i(k)$ and $p_i(k)$ for the kth iteration of our SCA-based algorithm, we derive a convex

$$\begin{aligned} &\tilde{g}(l_i, p_i; l_i(k), p_i(k)) \\ &= \frac{1}{B}\left[\frac{1}{2}\left(L_i - l_i + \frac{1}{\log_2(\frac{n_A n_E + n_E h_{iA} p_i}{n_A n_E + n_A g_{i\tilde{m}} p_i})}\right)^2 - \frac{1}{2}\left((L_i - l_i(k))^2 + \frac{1}{\left(\log_2(\frac{n_A n_E + n_E h_{iA} p_i(k)}{n_A n_E + n_A g_{i\tilde{m}} p_i(k)})\right)^2}\right)\right. \\ &\quad \left. + (L_i - l_i(k))(l_i - l_i(k)) + \frac{n_A n_E (n_E h_{iA} - n_A g_{i\tilde{m}})}{\ln 2 (n_A n_E + n_E h_{iA} p_i(k))(n_A n_E + n_A g_{i\tilde{m}} p_i(k))} \frac{p_i - p_i(k)}{\left(\log_2(\frac{n_A n_E + n_E h_{iA} p_i(k)}{n_A n_E + n_A g_{i\tilde{m}} p_i(k)})\right)^3}\right] \end{aligned} \tag{33}$$

$$h_2{}'(p_i(k)) = \left(\log_2\left(\frac{n_A n_E + n_E h_{iA} p_i(k)}{n_A n_E + n_A g_{i\tilde{m}} p_i(k)}\right) - \frac{p_i(k) n_A n_E (n_E h_{iA} - n_A g_{i\tilde{m}})}{\ln 2 (n_A n_E + n_E h_{iA} p_i(k))(n_A n_E + n_A g_{i\tilde{m}} p_i(k))}\right) \times \frac{1}{\left(\log_2(\frac{n_A n_E + n_E h_{iA} p_i(k)}{n_A n_E + n_A g_{i\tilde{m}} p_i(k)})\right)^2} \tag{38}$$

approximation of $\hat{U}(l_i, p_i)$ using *Lemma* 3 as

$$\begin{aligned}
&\tilde{U}(l_i, p_i; l_i(k), p_i(k)) \\
&= \frac{\alpha_i}{\varsigma C_i L_i f_i^2 B} \left(\tilde{h}_1(l_i; l_i(k)) h_2(p_i(k)) + h_1(l_i(k)) \right) \\
&= \frac{\alpha_i}{\varsigma C_i L_i f_i^2 B} \Bigg(\frac{(L_i - l_i) p_i(k)}{\log_2(\frac{n_A n_E + n_E h_{iA} p_i(k)}{n_A n_E + n_A g_{i\tilde{m}} p_i(k)})} \\
&\quad + (L_i - l_i(k)) \tilde{h}_2(p_i; p_i(k)) \Bigg),
\end{aligned} \tag{40}$$

where $\tilde{h}_2(p_i; p_i(k))$ is given by (37) and (38). In (40), $\tilde{h}_1(l_i; l_i(k)) = L_i - l_i$ because $h_1(l_i)$ is an affine function.

Therefore, the convex surrogate objective function $\hat{\mathcal{R}}_i^{\text{den}}$ of (23a) can be transformed as a nonnegative weighted sum of convex functions, which is given by

$$\widetilde{\mathcal{R}}_i^{\text{den}} = \frac{\alpha_i l_i}{L_i} + \tilde{U}(l_i, p_i; l_i(k), p_i(k)) + \frac{(1 - \alpha_i) z_i}{D_i}, \tag{41}$$

where the convexity is preserved.

Based on the above derivations, problem **(P3)** can be equivalently converted to

$$\underline{\textbf{(P4)}}: \min_{p_i, l_i, z_i} \ \widetilde{\mathcal{R}}_i^{\text{den}} \tag{42a}$$

$$\text{s.t.} \ \ t_i^{\text{loc}} \le z_i \le D_i, \tag{42b}$$

$$z_i \ge t_i^{\text{AP}} + \tilde{g}(l_i, p_i; l_i(k), p_i(k)), \tag{42c}$$

$$0 \le l_i \le L_i, \tag{42d}$$

$$0 \le p_i \le p_i^{\max}. \tag{42e}$$

The above optimization problem **(P4)** is convex, and the SCA-based algorithm is summarized in Algorithm 1, where the feasible solution for kth iteration is $(l_i(k), p_i(k)) = (l_i', p_i')$.

Algorithm 1 SCA-Based Algorithm for Solving Problem **(P4)**

Input:
δ: tolerance for convergence
K: maximum number of iterations
(l_i', p_i'): initial feasible solution for resource allocation
Output:
(l_i^*, p_i^*): optimal resource allocation strategy for user i

1: Initialize: $k = 1$, $\widetilde{\mathcal{R}}_i^{\text{den}(0)} = -\delta$ and $Converge$ = false;
2: **while** $Converge$ = false and $k \le K$ **do**
3: Solve problem **(P4)** based on (l_i', p_i') using CVX;
4: Obtain objective function value $\widetilde{\mathcal{R}}_i^{\text{den}(k)}$ and solution (l_i^k, p_i^k) of the kth iteration;
5: **if** $|\widetilde{\mathcal{R}}_i^{\text{den}(k)} - \widetilde{\mathcal{R}}_i^{\text{den}(k-1)}| \le \delta$ **then**
6: $Converge$ = true
7: **else**
8: $Converge$ = false
9: **end if**
10: Update $l_i' = l_i^k$ and $p_i' = p_i^k$;
11: $k = k + 1$;
12: **end while**
13: Return optimal solution $(l_i^*, p_i^*) = (l_i', p_i')$.

In Algorithm 1, δ and K are the tolerance for convergence (i.e., the desired accuracy) and the maximum number of iterations, respectively. The iteratively updated l_i' and p_i' are the initial feasible solution for each iteration of the SCA-based algorithm. l_i' and p_i' can be any values within the given ranges (i.e., $0 \le l_i' \le L_i$ and $0 \le p_i' \le p_i^{\max}$), and different initial values will not affect the final convergent optimal solution of Algorithm 1. After the k–1th iteration, the updated l_i' and p_i' is the input feasible solution for the kth iteration. Then, given an initial feasible solution (l_i', p_i') for resource allocation, we iteratively approximate the optimal solution until the termination condition is satisfied, i.e., $|\widetilde{\mathcal{R}}_i^{\text{den}(k)} - \widetilde{\mathcal{R}}_i^{\text{den}(k-1)}| \le \delta$ or $k > K$, where $\widetilde{\mathcal{R}}_i^{\text{den}(k)}$ is the objective value of the kth iteration.

Complexity analysis: The complexity of Algorithm 1 is theoretically analyzed as follows. As mentioned above, by using the SCA approach, the original nonconvex problem can be transformed into a strictly convex optimization problem **(P4)**. It can be solved by the interior point method of the CVX toolkit. Therefore, the overall complexity of solving problem **(P4)** in Algorithm 1 can be written as $O(K\log(\frac{1}{\delta}))$, where K are the number of iterations to solve problem **(P4)** using CVX, and δ denotes the accuracy required for the convergence of Algorithm 1. Therefore, our proposed Algorithm 1 can solve the resulting problem in polynomial time complexity.

V. Solution Overview

In this section, we provide a detailed overview description of our SMax-SCO scheme. Specifically, we first introduce the application scenario and problem construction of SMax-SCO. Then, we present a visual flow diagram to illustrate the main steps in the problem transformation process.

In a MEC-based computation offloading application, due to limited computing capacity, each participating user offloads part of the computation tasks to the MEC server in the presence of multiple non-colluding or colluding eavesdroppers. During the task offloading and processing, each user has both security requirements and various task preferences regarding delay and energy consumption.

- For security, the MEC server and legitimate users can monitor and discover malicious eavesdroppers. If some eavesdroppers are displaying colluding behaviours (e.g., gathering and exchanging eavesdropping information), the server will assume a partial or complete colluding eavesdropping scenario. For this case, security provisioning is the basic requirement of users.
- For delay and energy consumption, different users have various computation tasks and device statuses, and thus their requirements for delay reduction and energy saving are diverse. To this end, we design an adaptive weight factor, which can be assigned as needed by each user according to their emphasis on energy saving compared to delay reduction. The details of a three-user example are given in Section III-B.

As such, we measure the requirement satisfaction of users using security provisioning, delay reduction and energy saving. Then, we propose the SMax-SCO scheme, and construct a requirement satisfaction maximization problem, as shown in problem **(P1)**. Because **(P1)** has high computational complexity, and is an intractable non-convex problem, we must

transform the problem using a tractable approximation. The specific problem transformation process is shown in Fig. 2.

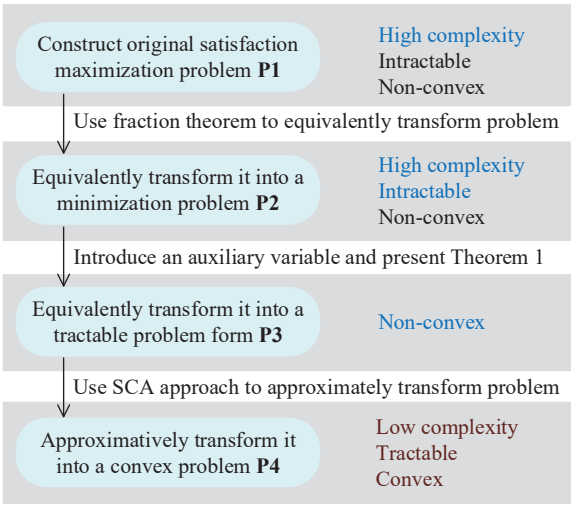


Fig. 2: Problem transformation process of SMax-SCO

First, we use the fraction theorem to transform the maximizing fraction problem into a minimizing denominator problem (the numerator is a constant), and rewrite the original problem as **(P2)**. Second, to further reduce the computational complexity and express the problem in a more tractable form, we introduce an auxiliary variable z_i and present Theorem 1. Then, the problem is transformed as problem **(P3)**. But **(P3)** is still a non-convex problem. Finally, we apply the SCA approach to approximatively transform it into a strictly convex problem **(P4)**. It can be quickly solved by the interior point method in the CVX toolkit. The specific solution process of SMax-SCO is detailed in Algorithm 1.

VI. Performance Evaluation

In this section, we evaluate the performance of the proposed SMax-SCO scheme via extensive simulations. All the simulations are conducted in MATLAB using CVX on a desktop computer with an Intel Core i7-7700 3.60GHz CPU and 16GB RAM. The simulation parameters settings are summarized in Table I unless otherwise stated.

In the following simulation experiments, we use L, D and ς to describe the characteristics of various computation tasks belonging to different users. Specifically, L_i is the workload of the computation tasks required by user i, D_i is the tolerable delay of i's tasks, and ς is the effective capacitance coefficient of i's device. A larger ς represents more energy consumed for computing one bit of tasks. Besides, we use α_i to measure user i's preference for energy consumption compared to delay regarding the current computational tasks. A larger α_i (i.e., $\alpha_i > 0.5$) means that user i places relatively higher priority on energy consumption than delay under security provisioning. One reason for setting a larger α_i is that the current tasks are more energy-consuming or user i's device has low power.

We validate the advantages of SMax-SCO from three aspects: convergence analysis, requirement satisfaction analysis and secrecy capacity analysis. In terms of convergence analysis, we show the number of iterations and convergence time of the proposed Algorithm 1 when solving the optimal solution. In terms of requirement satisfaction analysis, we highlight that security provisioning, delay reduction and energy saving are the three main requirements of users when offloading computation tasks. Hence, the requirement satisfaction in this paper is measured by the gains in the above three requirements of the offloading solution, and is specifically given by equation (16). In terms of secrecy capacity analysis, we calculate the secrecy capacity of the offloading process in different eavesdropping scenarios using equations (8) and (11).

TABLE I: Simulation Parameter Settings

Parameters	Values
Maximum number of iterations in Algorithm 1: K	50
Tolerance for convergence in Algorithm 1: δ	0.0001
Channel bandwidth: B	20MHz
Channel gain from user i to the AP: h_{iA}	2.5×10^{-7}
Estimation error of g_{im}: Δg_{im}	0.2×10^{-8}
Background noise at AP: n_A	-43dbm
Background noise at each eavesdropper: n_E	-50dbm
CPU cycles required for computing 1-bit tasks at user i: C_i	100 cycles/bit
CPU cycles required for computing 1-bit tasks at AP: C_{AP}	100 cycles/bit
Maximum transmission power of user i: $p_i^{\max}$	2W
Average CPU frequency at user i: f_i	1GHz
Average CPU frequency at AP: f_{AP}	3GHz

A key contribution of this paper is to consider different requirements for secure offloading by each user regarding delay and energy consumption, and provide the corresponding offloading strategy. In the current literature, depending on the user requirement of delay or energy consumption, the existing works on MEC-based secure computation offloading can be generally categorized into either security-based delay minimization schemes or security-based energy consumption minimization schemes. Therefore, we compare our proposed SMax-SCO scheme with these two types of schemes to highlight the advantages of our proposed scheme. Specifically, the two schemes used for comparisons are configured as follows:

1) Delay minimization secure computation offloading (DMin-SCO) scheme [4]: In this scheme, there is one eavesdropper in the MEC system. Moreover, user i adopts partial offloading mode in the single-eavesdropper MEC-based computation offloading system, and the AP aims to provide solutions with minimum delay and security provisioning for user i's tasks.
2) Energy consumption minimization secure computation offloading (EMin-SCO) scheme [5]: In this scheme, there is one eavesdropper in the MEC system. Moreover, user i also adopts partial offloading mode in the single-eavesdropper MEC-based computation offloading system, and the AP aims to provide solutions with minimum energy consumption and security provisioning for user i's tasks.

TABLE II: Convergence Performance of Algorithm 1

	$\alpha_i = 0.5$, $L_i = 1\times10^6$	$\alpha_i = 0.5$, $L_i = 2.5\times10^6$	$\alpha_i = 0.5$, $L_i = 4\times10^6$	$\alpha_i = 0.2$, $L_i = 3\times10^6$	$\alpha_i = 0.5$, $L_i = 3\times10^6$	$\alpha_i = 0.8$, $L_i = 3\times10^6$	Average
Number of Iterations	5	3	5	5	3	3	4
Convergence Time (s)	2.375864	1.78303	2.370782	2.388341	1.72754	1.799234	2.074132

A. Convergence analysis

First, we verify the convergence of Algorithm 1 for the SMax-SCO scheme in the presence of an eavesdropper m, shown in Table II. In this case, we set $D_i = 0.6s$ and $\varsigma = 10^{-27}$. The average channel gain $\bar{g}_{im}$ of eavesdropper m is set to 1.05×10^{-7}. Then, we consider different α_i and L_i (bits), and then analyze the convergence performance of Algorithm 1 with these different task requirements. From Table II, we can observe that the optimal solution is achieved within 5 iterations at most, regardless of the values of α_i and L_i. Meanwhile, under our simulation conditions of a standard desktop computer, the algorithm convergence time in Table II for different task requirements does not exceed 2.5 seconds, and the average is only about 2 seconds. In reality, the MEC server has far more computational resources than a desktop computer, so it is possible to obtain the optimal solution of Algorithm 1 almost in real time. Therefore, the proposed Algorithm 1 for the SMax-SCO scheme can quickly converge to a stable value with a fast convergence speed, thus verifying the convergence and feasibility of our Algorithm 1.

B. Requirement Satisfaction Analysis

Subsequently, we validate the advantages in requirement satisfactions of our proposed SMax-SCO scheme compared to DMin-SCO and EMin-SCO schemes regarding different tolerable delays, computation task sizes, α and number of users.

- **Regarding changes in tolerable delays**

First, we evaluate the performance of different tolerable delays in different eavesdropping scenarios. We set $\alpha_i = 0.2$, $L_i = 3\times10^6$ bits and $\varsigma = 10^{-27}$ to describe the characteristics of user i's current computation tasks. Obviously, the tasks are delay-sensitive for user i as $\alpha_i = 0.2 < 0.5$.

For the single-eavesdropper scenario, Fig. 3 shows the requirement satisfaction $RS(i)$ of user i in the presence of an eavesdropper m changing with the tasks' tolerable delay D_i in the three schemes. The average channel gain $\bar{g}_{im}$ of eavesdropper m is set to 1×10^{-7}. All three schemes can achieve security provisioning (i.e., $x_{iA} = 1$) due to $h_{iA} \geq \frac{n_A}{n_m^E} g_{im}$ ($g_{im} = \bar{g}_{im} + \Delta g_{im}$). In Fig. 3, we find that the requirement satisfaction increases with increasing tolerable delay D_i in the three schemes. This is because when task workload L_i is constant and $x_{iA} = 1$, a larger tolerable delay D_i leads to a smaller delay term (i.e., $(1-\alpha_i)\gamma_i^D$) of the denominator in (16). Moreover, compared with EMin-SCO, DMin-SCO shows relatively better performance because the current tasks are delay-sensitive. Furthermore, we show that SMax-SCO and DMin-SCO achieve a similar $RS(i)$ when the tolerable delay is very small. This is because the computation tasks with smaller tolerable delay are more delay-sensitive. In this case, our SMax-SCO will place more focus on minimizing delay to satisfy the strict delay constraint, which the resulting solution is closer to DMin-SCO. Overall, it is obvious that our proposed SMax-SCO scheme outperforms the other schemes. The reason for the improvement is that SMax-SCO can accommodate different user preferences with security provisioning, while other schemes can only provide a fixed strategy of minimizing delay or energy consumption. When $\alpha_i = 0.2$, SMax-SCO places a higher priority on minimizing user i's task processing delay compared to energy consumption to maximize the requirement satisfaction of user i. In other words, instead of ignoring energy saving as DMin-SCO, we give energy consumption relatively low attention on the basis of ensuring security provisioning. DMin-SCO and EMin-SCO default to $\alpha_i = 0$ and 1, respectively, and thus they show lower requirement satisfaction than our SMax-SCO scheme.

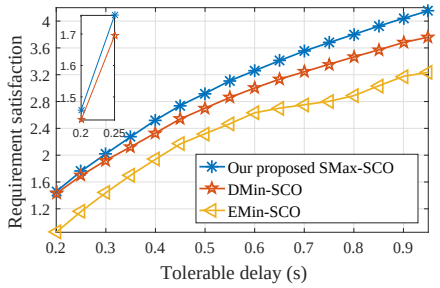
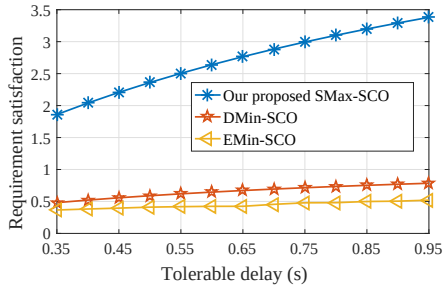


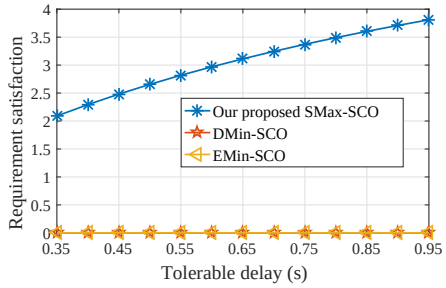
Fig. 3: Requirement satisfaction in the presence of an eavesdropper versus tolerable delays

For the multi-eavesdropper scenario, Fig. 4 shows the requirement satisfaction $RS(i)$ of user i in the presence of three eavesdroppers changing with the tasks' tolerable delay D_i in the three schemes. Fig. 4 (a) is the case of three non-colluding eavesdroppers, and the average channel gains are $\bar{g}_{im} = \{1, 1.6, 2.2\} \times 10^{-7}$. Fig. 4 (b) is the case of three colluding eavesdroppers, and the average channel gains are $\bar{g}_{im} = \{0.5, 0.5, 0.5\} \times 10^{-7}$ (The channel gain of each colluding eavesdropper is generally small, which is why they choose to collude). Since DMin-SCO and EMin-SCO are the single-eavesdropper schemes, we randomly select a value of $\bar{g}_{im}$ from three eavesdroppers for each experimental trial and present the average results for a large number of trials.

In Fig. 4 (a), we observe that our proposed SMax-SCO scheme outperforms the other schemes. There are two reasons for this improvement. First, SMax-SCO can balance user i's preference regarding delay and energy consumption, while DMin-SCO and EMin-SCO can only provide a fixed solution. Second, in the case of non-colluding eavesdroppers, SMax-



(a) Case of non-colluding eavesdroppers



(b) Case of colluding eavesdroppers

Fig. 4: Requirement satisfaction in the presence of three eavesdroppers versus tolerable delays.

SCO is designed to prevent eavesdropping attacks by the best eavesdropper $\tilde{m}$, and the channel gain $g_{i\tilde{m}} = \max\{\bar{g}_{im} + \Delta g_{im}, \forall m \in \mathcal{M}\} = 2.22 \times 10^{-7}$. By contrast, DMin-SCO and EMin-SCO default that there is one eavesdropper in the considered system. When their default eavesdropper is not the best eavesdropper, the provided solutions cannot ensure the security of offloading tasks (i.e., $x_{iA} = 0$), and then the requirement satisfaction $RS(i)$ of user i is zero.

In Fig. 4 (b), the requirement satisfactions of DMin-SCO and EMin-SCO are always zero regardless of the tolerable delays. This is because they ignore the multi-eavesdropper scenarios in the task environments, and therefore cannot guarantee a secure offloading rate for the user. Moreover, we observe that the requirement satisfactions of SMax-SCO in Fig. 4 (b) are smaller compared with those in Fig. 4 (a). This is because in the case of colluding eavesdroppers, SMax-SCO is designed to prevent eavesdropping attacks by the super eavesdropper $\tilde{m}$, and the channel gain $g_{i\tilde{m}} = \sum_{m\in\mathcal{M}} (\bar{g}_{im} + \Delta g_{im}) = 1.52 \times 10^{-7}$. A larger channel gain $\bar{g}_{i\tilde{m}}$ leads to the consumption of more wireless resources to ensure security provisioning, and thus occupying part of the resources previously used to improve the service satisfaction of delay and energy consumption.

- **Regarding changes in computation task sizes**

Next, we evaluate the performance of different computation task sizes in different eavesdropping scenarios. We set $\alpha_i = 0.8$, $D_i = 0.8s$, $\varsigma = 2 \times 10^{-27}$ to describe the characteristics of user i's current tasks. This describes user i having energy-intensive tasks or the residual energy of his/her device is low due to the setting of $\alpha_i = 0.8 > 0.5$ and the effective capacitance coefficient $\varsigma = 2 \times 10^{-27}$.

For the single-eavesdropper scenario, Fig. 5 shows the requirement satisfaction $RS(i)$ of user i in the presence of an eavesdropper m changing with the task workload L_i. The average channel gain $\bar{g}_{im}$ of eavesdropper m is 1×10^{-7}. All three schemes can achieve security provisioning (i.e., $x_{iA} = 1$) due to $h_{iA} \geq \frac{n_A}{n_m^E} g_{im}$. This figure shows that the requirement satisfaction decreases with increasing task workload. This is because when tolerable delay D_i is constant and $x_{iA} = 1$, a larger task workload L_i leads to a larger energy consumption term (i.e., $\alpha_i \gamma_i^E$) of the denominator in (16). Meanwhile, we observe that EMin-SCO performs better than DMin-SCO because the current tasks are energy-intensive for user i's device. More importantly, our proposed SMax-SCO scheme outperforms EMin-SCO, and performs far better than DMin-SCO. The reason is that user i in SMax-SCO is not completely ignoring the processing delay of current tasks, but the delay is of relatively low concern compared to energy consumption under the premise of ensuring security provisioning. Therefore, SMax-SCO can accommodate each user's preference, and then improve the requirement satisfactions of users.

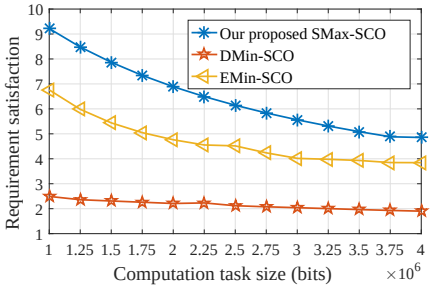
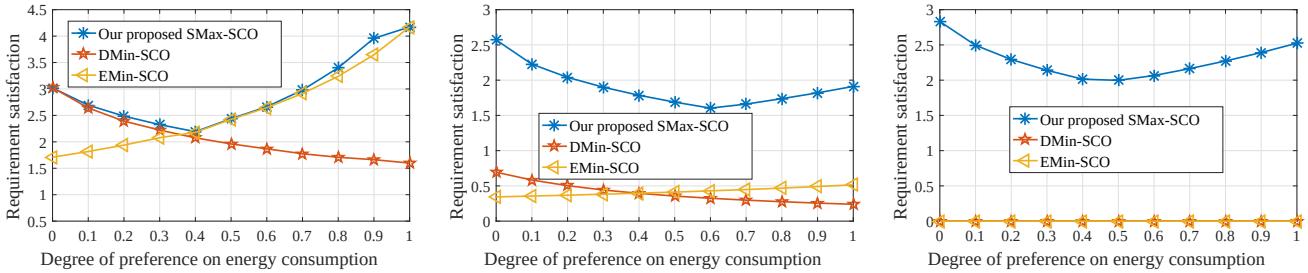


Fig. 5: Requirement satisfaction in the presence of an eavesdropper versus task sizes

For the multi-eavesdropper scenario, Fig. 6 shows the requirement satisfaction $RS(i)$ of user i in the presence of three eavesdroppers changing with the computation task workload L_i. Fig. 6 (a) is the case of three non-colluding eavesdroppers with the same average channel gains as Fig. 4 (a), and the channel gain of the best eavesdropper $\tilde{m}$ is $g_{i\tilde{m}} = 2.22 \times 10^{-7}$. Fig. 6 (b) is the case of three colluding eavesdroppers with the same average channel gains as Fig. 4 (b), and the channel gain of the best (super) eavesdropper $\tilde{m}$ is $g_{i\tilde{m}} = 1.52 \times 10^{-7}$. In Fig. 6 (a), we observe that our proposed SMax-SCO scheme outperforms the other schemes. This is because SMax-SCO can accurately determine the worst-case eavesdropper in the non-colluding eavesdropper scenario, while DMin-SCO and EMin-SCO are assumed to randomly select an eavesdropper. In Fig. 6 (b), the requirement satisfactions of DMin-SCO and EMin-SCO are always zero regardless of computation task workload due to the fact that they do not consider the impact of the multiple eavesdroppers colluding as a single super eavesdropper.

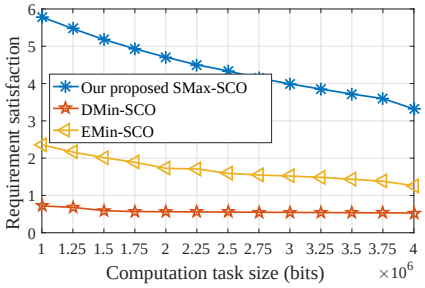
- **Regarding changes in α and number of users**

Finally, we evaluate the performance of different α and number of users in different eavesdropping scenarios. We verify our advantages when there are multiple participating users with various task preferences in the MEC system. We

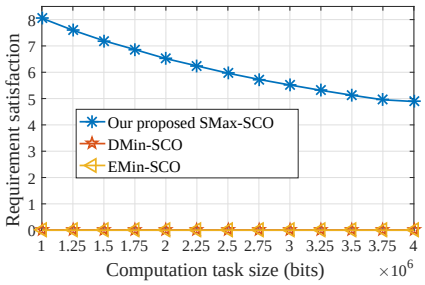


(a) Case of a single eavesdropper (b) Case of non-colluding eavesdroppers (c) Case of colluding eavesdroppers

Fig. 7: Requirement satisfaction versus the degree of emphasis on energy consumption.



(a) The case of non-colluding eavesdroppers



(b) The case of colluding eavesdroppers

Fig. 6: Requirement satisfaction in the presence of three eavesdroppers versus task sizes.

aim to highlight the positive impact of considering each user's requirement preference regarding delay and energy consumption on the basis of security provisioning. We set $L = 3 \times 10^6$ bits, $D = 0.4s$, $\varsigma = 10^{-27}$ for each user's tasks. We specifically consider the three cases of single eavesdropper, non-colluding eavesdropper and colluding eavesdropper scenarios. The average channel gains in the three cases are the same as in Figs. 3 and 4. Since DMin-SCO and EMin-SCO are the single-eavesdropper schemes, we randomly select a value of $\bar{g}_{im}$ from three eavesdroppers for each multiple-eavesdropper experimental trial and present the average results for a large number of trials.

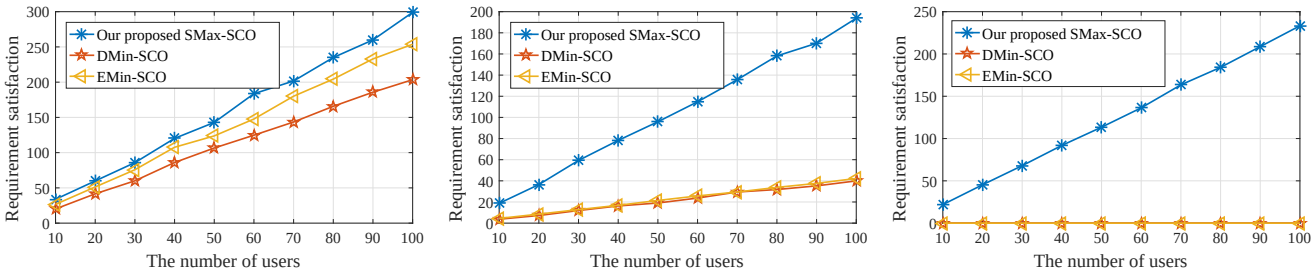
Fig. 7 shows the requirement satisfaction $RS(i)$ of user i changing with the degree of preference on energy consumption α_i in different eavesdropper scenarios. In Fig. 7 (a), the three schemes can achieve security provisioning (i.e., $x_{iA} = 1$) due to the single-eavesdropper scenario and $h_{iA} \geq \frac{n_A}{n_E^m} g_{i\tilde{m}}$. Obviously, our proposed SMax-SCO scheme is equivalent to the DMin-SCO scheme when $\alpha_i = 0$, and is equivalent to the EMin-SCO scheme when $\alpha_i = 1$. Moreover, in Fig. 7 (a), we find that SMax-SCO always achieves a higher requirement satisfaction than other schemes regardless of the preference degrees α_i, because SMax-SCO can accommodate the varying requirements of users. Furthermore, we observe that the turning point of the data curve in SMax-SCO occurs approximately at $\alpha_i = 0.4 < 0.5$. This phenomenon shows that the computation tasks with the above settings of L, D, ς and $g_{i\tilde{m}}$ are relatively more energy-consuming. In Figs. 7 (b) and (c), we observe that our proposed SMax-SCO scheme outperforms the other schemes, and the requirement satisfactions of DMin-SCO and EMin-SCO in Fig. 7 (b) are always zero regardless of the preference degrees α_i. The specific reasons are the same as those in Fig. 4 (a) and (b).

Fig. 8 shows the requirement satisfaction $RS(i)$ of user i changing with the number of users in different eavesdropper scenarios. The α_i value of each user i is randomly generated between 0 and 1 to represent the different requirement preferences of different users. We observe that our SMax-SCO scheme consistently outperforms the other schemes regardless of the number of users and the eavesdropper scenarios. Besides, in Fig. 8 (a), EMin-SCO has better performance than DMin-SCO, which supports the conclusion in Fig. 7 (a) that the computation tasks are more energy-consuming. More importantly, we find that under our above simulation settings, the requirement satisfactions of SMax-SCO in the colluding eavesdropper case are greater than those in the non-colluding eavesdropper case and less than those in the single eavesdropper case. This is because the requirement satisfaction is negatively correlated with the channel gain $g_{i\tilde{m}}$, and $2.22 \times 10^{-7} > 1.52 \times 10^{-7} > 1.02 \times 10^{-7}$.

C. Secrecy Capacity Analysis

In this subsection, we validate the advantages of our proposed SMax-SCO scheme in terms of secrecy capacity. We set $L = 3 \times 10^6$ bits, $D = 0.4s$ and $\varsigma = 10^{-27}$.

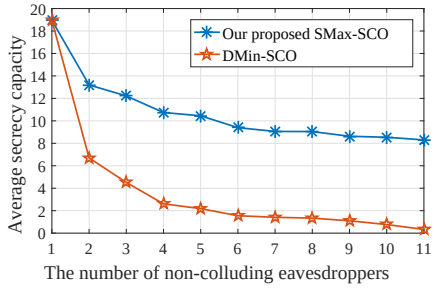
First, we evaluate the performance of the three schemes in non-colluding eavesdropping scenarios. We consider 11 non-colluding eavesdroppers with average channel gains of $\bar{g}_{im} = \{1, 1.2, 1.4, 1.6, 1.8, 2, 2.2, 2.4, 2.6, 2.8, 3\} \times 10^{-7}$, and randomly select a certain number of eavesdroppers in each experiment. We show the average results of a large number experiments. In Fig. 9 (a), we set $\alpha = 0$, and show the



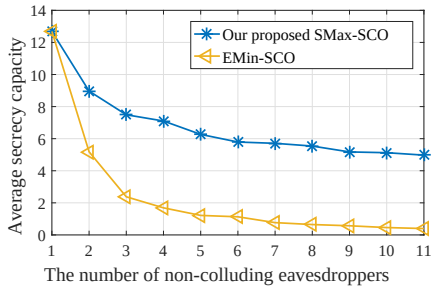
(a) Case of a single eavesdropper (b) Case of non-colluding eavesdroppers (c) Case of colluding eavesdroppers

Fig. 8: Requirement satisfaction versus the number of users.

average secrecy capacity decreasing with increasing number of non-colluding eavesdroppers in the SMax-SCO and DMin-SCO schemes. In Fig. 9 (b), we set $\alpha = 1$, and similarly show that the average secrecy capacity decreases with the increasing number of non-colluding eavesdroppers for both the SMax-SCO and EMin-SCO schemes. This is generally because a larger number of non-colluding eavesdroppers leads to a larger channel gain $g_{i\tilde{m}}$ for the worst-case eavesdropper. Meanwhile, we observe from Fig. 9 that SMax-SCO outperforms DMin-SCO and EMin-SCO, especially when the number of eavesdroppers is large. The reason is that SMax-SCO considers multiple-eavesdropper scenarios and provides corresponding security offloading strategies, while the other schemes focus only on single-eavesdropper scenarios.



(a) $\alpha = 0$



(b) $\alpha = 1$

Fig. 9: Average secrecy capacity versus the number of non-colluding eavesdroppers.

Next, we evaluate the performance of the three schemes in colluding eavesdropping scenarios. We set 8 colluding eavesdroppers with the same average channel gain of $\bar{g}_{im} = 0.5 \times 10^{-7}$. Fig. 10 shows the secrecy capacity changing with the number of colluding eavesdroppers in DMin-SCO, EMin-SCO, and SMax-SCO with different values of α. Obviously, the secrecy capacities in DMin-SCO and EMin-SCO are always zero because they cannot deal with the colluding eavesdropping attacks. Meanwhile, we observe that the secrecy capacity decreases with increasing α in our proposed SMax-SCO scheme. The reason is that with the increase of α, SMax-SCO pays more attention to minimizing the energy consumption of users when completing computing tasks. Specifically, when α increases, the users will try to offload more of their computation tasks to the AP for processing, while SMax-SCO assigns a relatively low offloading power p_i to the offloading tasks to reduce the energy consumption of the offloading process. Then, a smaller offloading power p_i leads to a small secrecy capacity.

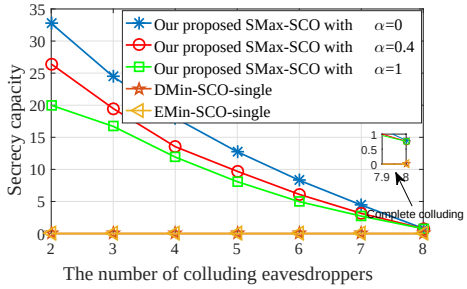


Fig. 10: Secrecy capacity versus the number of colluding eavesdroppers

VII. Conclusion and Discussion

MEC-based computation offloading is a promising paradigm for supporting ubiquitous mobile users. Due to the broadcast nature of wireless communications, the computation tasks offloaded from mobile devices to MEC servers are likely to be overheard by malicious attackers nearby. To address this challenge, there has been substantial research on MEC-based secure computation offloading. However, these works ignore the varying requirements of individual IoT users in terms of delay and energy consumption on the basis of security provisioning. Furthermore, the eavesdropping model considered in these works does not consider the impact of collusion. To this end, we considered a practical multi-eavesdropper model including non-colluding and colluding

scenarios and a requirement satisfaction model. Furthermore, we proposed a satisfaction-maximized secure computation offloading (SMax-SCO) scheme, which aims to maximize the requirement satisfaction of individual users subject to secrecy offloading rate, tolerable delay, task workload and maximum power constraints. To tackle the original nonconvex problem, we developed an efficient iterative algorithm based on the SCA approach to obtain near-optimal solutions. Finally, we conducted extensive simulations, and verified that our proposed SMax-SCO scheme provides better performance in terms of secrecy capacity and requirement satisfaction of users than the existing schemes.

APPENDIX

PROOF OF THEOREM 1

We prove this *Theorem* via contradiction. Denoting the optimal offloading size and optimal offloading power as l_i^* and p_i^* respectively, regarding (20), we assume the optimal offloading rate $r_{iA}^* < B\log_2(\frac{n_A n_E + n_E h_{iA} p_i^*}{n_A n_E + n_A g_{i\tilde{m}} p_i^*})$.

Based on the optimal values l_i^* and p_i^*, one can find that the objective function (23a) is related to variable r_{iA}. In particular, we have

$$\mathcal{R}_i^{\mathrm{den}}(r_{iA}) = \begin{cases} \alpha_i \frac{\varsigma C_i l_i^* f_i + p_i^* \frac{L_i - l_i^*}{r_{iA}}}{\varsigma C_i L_i f_i} + (1-\alpha_i)\frac{\frac{C_i l_i^*}{f_i}}{D_i}, \\ \qquad\qquad r_{iA} \geq |\frac{L_i - l_i^*}{\frac{C_i l_i^*}{f_i} - \frac{C_{AP}(L_k - l_k^*)}{f_{AP}}}| \\ \alpha_i \frac{\varsigma C_i l_i^* f_i + p_i^* \frac{L_i - l_i^*}{r_{iA}}}{\varsigma C_i L_i f_i} + (1-\alpha_i)\frac{\frac{C_{AP}(L_i - l_i^*)}{f_{AP}} + \frac{L_i - l_i^*}{r_{iA}}}{D_i}, \\ \qquad\qquad r_{iA} < |\frac{L_i - l_i^*}{\frac{C_i l_i^*}{f_i} - \frac{C_{AP}(L_i - l_i^*)}{f_{AP}}}| \end{cases}$$

When $r_{iA} = |\frac{L_i - l_i^*}{\frac{C_i l_i^*}{f_i} - \frac{C_{AP}(L_i - l_i^*)}{f_{AP}}}|$, we have $\frac{C_i l_i^*}{f_i} = t_{AP} + \frac{L_i - l_i^*}{r_{iA}}$. Therefore, the above piecewise functions are continuous at the piecewise point. Moreover, r_{iA} is a real number greater than zero. Since $f(x) = a + \frac{b}{x}, (x > 0, x, a, b$ are real numbers$)$ is a continuous function, we can conclude that objective value $\mathcal{R}_i^{\mathrm{den}}$ is continuous for variable r_{iA}, and decreases with r_{iA}. This is because when the transmission power is constant, a larger offloading rate r_{iA} leads to a shorter offloading time and a smaller offloading energy consumption. Based on this, we believe that there must exist another $r_{iA}^{*\prime} = r_{iA}^* + \tau$, ($\tau$ is a small positive value), such that $\mathcal{R}_i^{\mathrm{den}\prime}(r_{iA}^{*\prime}) < \mathcal{R}_i^{\mathrm{den}}(r_{iA}^*)$. It is proved that the assumed offloading rate is not optimal. Therefore, the assumption we made above is incorrect, the equation $r_{iA}^* = B\log_2(\frac{n_A n_E + n_E h_{iA} p_i'}{n_A n_E + n_A g_{i\tilde{m}} p_i'})$ must be held in the optimal solution. Here, we have completed the proof of *Theorem* 1.

REFERENCES

[1] B. Yang, X. Cao, Z. Han and L. Qian, "A Machine Learning Enabled MAC Framework for Heterogeneous Internet-of-Things Networks," *IEEE Trans. Wireless Commun.*, vol. 18, no. 7, pp. 3697-3712, Jul. 2019.

[2] B. Lin, X. Wang, W. Yuan and N. Wu, "A Novel OFDM Autoencoder Featuring CNN-Based Channel Estimation for Internet of Vessels," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 7601-7611, Apr. 2020.

[3] 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, 4th Quart., 2017.

[4] Y. Wu, J. Shi, L. Qian, W. Zhu, Z. Shi and L. Meng, "Secrecy-Based Delay-Aware Computation Offloading via Mobile Edge Computing for Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4201-4213, Jun. 2019.

[5] J. Xu and J. Yao, "Exploiting Physical-Layer Security for Multiuser Multicarrier Computation Offloading," *IEEE Wireless Commun. Lett.*, vol. 8, no. 1, pp. 9-12, Feb. 2019.

[6] Z. Yu, Y. Gong, S. Gong, Y. Guo, "Joint Task Offloading and Resource Allocation in UAV-Enabled Mobile Edge Computing," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 3147-3159, Apr. 2020.

[7] N. Wu, X. Wang, B. Lin and K. Zhang, "A CNN-Based End-to-End Learning Framework Toward Intelligent Communication Systems," *IEEE Access*, vol. 7, pp. 110197-110204, Jul. 2019.

[8] G. Scutari, F. Facchinei and L. Lampariello, "Parallel and Distributed Methods for Constrained Nonconvex Optimization-Part I: Theory," *IEEE Trans. Signal Process.*, vol. 65, no. 8, pp. 1929-1944, Apr. 2017.

[9] S. Bi and Y. J. Zhang, "Computation Rate Maximization for Wireless Powered Mobile-Edge Computing With Binary Computation Offloading," *IEEE Trans. Wireless Commun.*, vol. 17, no. 6, pp. 4177-4190, Jun. 2018.

[10] Z. Ning, P. Dong, X. Kong and F. Xia, "A Cooperative Partial Computation Offloading Scheme for Mobile Edge Computing Enabled Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4804-4814, Jun. 2019.

[11] J. Peng, H. Qiu, J. Cai, W. Xu and J. Wang, "D2D-Assisted Multi-User Cooperative Partial Offloading, Transmission Scheduling and Computation Allocating for MEC," *IEEE Trans. Wireless Commun.*, DOI: 10.1109/TWC.2021.3062616, Mar. 2021.

[12] S. Nath and J. Wu, "Deep Reinforcement Learning for Dynamic Computation Offloading and Resource Allocation in Cache-Assisted Mobile Edge Computing Systems," *Intelligent and Converged Networks*, vol. 1, no. 2, pp. 181-198, Oct. 2020.

[13] B. Li, W. Wu, W. Zhao and H. Zhang, "Security Enhancement With a Hybrid Cooperative NOMA Scheme for MEC System," *IEEE Tran. Veh. Technol.*, vol. 70, no. 3, pp. 2635-2648, Mar. 2021.

[14] X. He, R. Jin, H. Dai, "Peace: Privacy-Preserving and Cost-Efficient Task Offloading for Mobile Edge Computing," *IEEE Trans. Wireless Commun.*, vol. 19, no. 3, pp. 1814-1824, Mar. 2020.

[15] L. Xiao, X. Lu, T. Xu, X. Wan, W. Ji and Y. Zhang, "Reinforcement Learning-Based Mobile Offloading for Edge Computing Against Jamming and Interference," *IEEE Trans. Commun.*, vol. 68, no. 10, pp. 6114-6126, Oct. 2020.

[16] Z. Ding, P. Fan, and H. V. Poor, "Impact of Non-orthogonal Multiple Access on the Offloading of Mobile Edge Computing," *IEEE Trans. Commun.*, vol. 67, no. 1, pp. 375-390, Jan. 2019.

[17] Z. Sheng, H. D. Tuan, A. A. Nasir, T. Q. Duong and H. V. Poor, "Secure UAV-Enabled Communication Using Han-Kobayashi Signaling," *IEEE Trans. Wireless Commun.*, vol. 19, no. 5, pp. 2905-2919, May. 2020.

[18] L. Qing, H. Guangyao and F. Xiaomei, "Physical Layer Security in Multi-Hop AF Relay Network Based on Compressed Sensing," *IEEE Commun. Lett.*, vol. 22, no. 9, pp. 1882-1885, Sept. 2018.

[19] M. Kamel, W. Hamouda and A. Youssef, "Physical Layer Security in Ultra-Dense Networks," *IEEE Wireless Commun. Lett.*, vol. 6, no. 5, pp. 690-693, 2017.

[20] H. Lin, Y. Cao, Y. Zhong and P. Liu, "Secure Computation Efficiency Maximization in NOMA-Enabled Mobile Edge Computing Networks," *IEEE Access*, vol. 7, pp. 87504-87512, Jul. 2019.

[21] S. Han, X. Xu, S. Fang, Y. Sun, Y. Cao, X. Tao and P. Zhang, "Energy Efficient Secure Computation Offloading in NOMA-Based mMTC Networks for IoT," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 5674-5690, Jun. 2019.

[22] W. Wu, F. Zhou, R. Q. Hu and B. Wang, "Energy-Efficient Resource Allocation for Secure NOMA-Enabled Mobile Edge Computing Networks," *IEEE Trans. Commun.*, vol. 68, no. 1, pp. 493-505, Jan. 2020.

[23] Y. Liu, W. Wang, H. H. Cheny, F. Lyu, L. Wang, W. Meng and X. Shen, "Physical Layer Security Assisted Computation Offloading in Intelligently Connected Vehicle Networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 6, pp. 3555-3570, Jun. 2021.

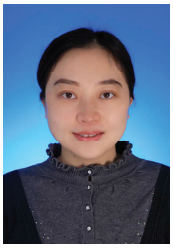
[24] T. Bai, J. Wang, Y. Ren and L. Hanzo, "Energy-Efficient Computation Offloading for Secure UAV-Edge-Computing Systems," *IEEE Trans. Veh. Technol.*, vol. 68, no. 6, pp. 6074-6087, Jun. 2019.

[25] Y. Zhou, C. Pan, P. L. Yeoh, K. Wang, M. Elkashlan, B. Vucetic and Y. Li, "Secure Communications for UAV-Enabled Mobile Edge Computing Systems," *IEEE Trans. Commun.*, vol. 68, no. 1, pp. 376-388, Jan. 2020.

[26] Y. Xu, T. Zhang, D. Yang, Y. Liu and M. Tao, "Joint Resource and Trajectory Optimization for Security in UAV-Assisted MEC Systems," *IEEE Trans. Commun.*, vol. 69, no. 1, pp. 573-588, Jan. 2021.

[27] M. A. Rodriguez and R. Buyya, "Deadline Based Resource Provisioningand Scheduling Algorithm for Scientific Workflows on Clouds," *IEEE Tran. Cloud Comput.*, vol. 2, no. 2, pp. 222-235, Jul. 2014.
[28] H. Wu, J. Zhang, Z. Cai, Q. Ni, T. Zhou, J. Yu, H. Chen and F. Liu, "Resolving Multi-task Competition for Constrained Resources in Dispersed Computing: A Bilateral Matching Game," *IEEE Internet Things J.*, DOI: 10.1109/JIOT.2021.3075673, Apr. 2021.
[29] R. Esmailzadeh, "Information Theory," *Broadband Telecommunications Technologies and Management*, 2016.
[30] K. Jiang, T. Jing, Y. Huo, F. Zhang and Z. Li, "SIC-based secrecy performance in uplink NOMA multi-eavesdropper wiretap channels," *IEEE Access*, vol. 6, pp. 19664-19680. Apr. 2018.

Shumei Liu received the B.S. degree in electronic and information engineering from Shanxi University, Taiyuan, China, in 2016 and the M.S. degree in electronics and communication engineering from Northeastern University, Shenyang, China, in 2018. She is currently pursuing the Ph.D. degree in communication and information system at Northeastern University. Her research interests include the Internet-of-Things (IoT), mobile edge computing, and radio resource management for enhancing the physical layer security. She has received the Best Paper Award in National Postdoctoral Academic Forum in China (2018).



Yao Yu (Member, IEEE) received B.S. degree in communication engineering from the Northeastern University, Shenyang, China in 2005, and the Ph.D. degree in communication and information system from the Northeastern University, Shenyang, China in 2010. From 2010 to 2011, she was a Postdoctoral Fellow with Department of Computing at Hong Kong Polytechnic University, Hong Kong, China. Also she was a visiting scholar in the University of Sydney from 2019 to 2020. She is currently a Professor at the School of Computer Science and Engineering, Northeastern University, Shenyang, China. Her current research interests include network security and big data. She is a member of the IEEE.

Lei Guo (Senior Member, IEEE) received the Ph.D. degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2006. He is currently a Full Professor with Chongqing University of Posts and Telecommunications, Chongqing, China. He has authored or coauthored more than 200 technical papers in international journals and conferences. He is an Editor for several international journals. His current research interests include communication networks, optical communications, and wireless communications.

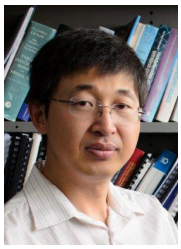
Phee Lep Yeoh (Member, IEEE) received the B.E. degree with University Medal and the Ph.D. degree from the University of Sydney (USYD), Australia, in 2004 and 2012, respectively. From 2005 to 2008, he was a Wireless Technology Specialist at Telstra, Australia. From 2012 to 2016, he was a Lecturer and Research Fellow in Wireless Communications at the University of Melbourne, Australia. Since 2016, he has been a Senior Lecturer with the School of Electrical and Information Engineering at USYD. His research interests include secure communications for the Internet-of-Things (IoT), ultra-reliable and low-latency communications (URLLC), and multi-scale molecular communications.

Dr. Yeoh is a recipient of the 2020 USYD Robinson Fellowship, the 2018 Alexander von Humboldt Research Fellowship for Experienced Researchers, and the 2014 Australian Research Council (ARC) Discovery Early Career Researcher Award (DECRA). He has received best paper awards at IEEE ICC 2014 and IEEE VTC-Spring 2013, and best student paper awards with his supervised students at the 2013 and 2019 Australian Communications Theory Workshop (AusCTW).

Branka Vucetic (Life Fellow, IEEE) received the Ph.D. degree from the University of Belgrade, Belgrade, Serbia, in 1982.

She is an ARC Laureate Fellow and Director of the Centre of Excellence for IoT and Telecommunications at the University of Sydney. Her current research work is in wireless networks and the Internet of Things. In the area of wireless networks, she works on communication system design for millimetre wave frequency bands. In the area of the Internet of Things, Vucetic works on providing wireless connectivity for mission critical applications.

Prof. Vucetic is a life Fellow of IEEE, the Australian Academy of Technological Sciences and Engineering and the Australian Academy of Science.



Yonghui Li (Fellow, IEEE) received his Ph.D. degree from Beijing University of Aeronautics and Astronautics, Beijing, China, in November 2002. From 1999 - 2003, he was affiliated with Linkair Communication Inc, where he held a position of project manager with responsibility for the design of physical layer solutions for the LAS-CDMA system. Since 2003, he has been with the Centre of Excellence in Telecommunications, the University of Sydney, Australia. He is now a Professor in School of Electrical and Information Engineering, University of Sydney. He is the recipient of the Australian Queen Elizabeth II Fellowship in 2008 and the Australian Future Fellowship in 2012. His current research interests are in the area of wireless communications, with a particular focus on MIMO, millimeter wave communications, machine to machine communications, coding techniques and cooperative communications. He holds a number of patents granted and pending in these fields.

Prof. Li is a Fellow of IEEE. He is now an editor for IEEE TRANSACTIONS ON COMMUNICATIONS and IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY. He also served as a guest editor for several special issues of IEEE journals, such as IEEE JSAC special issue on Millimeter Wave Communications. He received the best paper awards from IEEE International Conference on Communications (ICC) 2014, IEEE PIMRC 2017 and IEEE Wireless Days Conferences (WD) 2014.

Trung Q. Duong (Senior Member, IEEE) is a Chair Professor of Telecommunications at Queen's University Belfast (UK), where he was a Lecturer (Assistant Professor) (2013-2017), a Reader (Associate Professor) (2018-2020), and Full Professor from August 2020. He also holds a prestigious Research Chair of Royal Academy of Engineering. His current research interests include wireless communications, machine learning, realtime optimisation, and data analytic. He is the author or co-author of over 400 publications.

Dr. Duong currently serves as an Editor for the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, and an Executive Editor for IEEE COMMUNICATIONS LETTERS. He has served as an Editor/Guest Editor for IEEE TRANSACTIONS ON COMMUNICATIONS, IEEE WIRELESS COMMUNICATIONS, IEEE COMMUNICATIONS MAGAZINES, IEEE COMMUNICATIONS LETTERS, and IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS. He was awarded the Best Paper Award at the IEEE Vehicular Technology Conference (VTC-Spring) in 2013, IEEE International Conference on Communications (ICC) 2014, IEEE Global Communications Conference (GLOBECOM) 2016 and 2019, IEEE Digital Signal Processing Conference (DSP) 2017, and International Wireless Communications & Mobile Computing Conference (IWCMC) 2019. He is the recipient of prestigious Royal Academy of Engineering Research Fellowship (2015-2020) and has won a prestigious Newton Prize 2017.