SA-EAST: Security-Aware Efficient Data Transmission for ITS in Mobile Heterogeneous Cloud Computing

KEKE GAI, Pace University
LONGFEI QIU, Nanjing Foreign Language School
MIN CHEN, Huazhong University of Science and Technology
HUI ZHAO, Henan University
MEIKANG QIU, Pace University

The expected advanced network explorations and the growing demand for mobile data sharing and transferring have driven numerous novel applications in *Cyber-Physical Systems* (CPSs), such as *Intelligent Transportation Systems* (ITSs). However, current ITS implementations are restricted by the conflicts between security and communication efficiency. Focusing on this issue, this article proposes a *Security-Aware Efficient Data Sharing and Transferring* (SA-EAST) model, which is designed for securing cloud-based ITS implementations. In applying this approach, we aim to obtain secure real-time multimedia data sharing and transferring. Our experimental evaluation has shown that our proposed model provides an effective performance in securing communications for ITS.

Categories and Subject Descriptors: D.4.6 [**Security and Protection**]: Information flow controls; D.2.1 [**Network Architecture and Design**]: Distributed networks; H.2.0 [**General**]: Security, integrity, and protection

General Terms: Security, Theory

Additional Key Words and Phrases: Security-aware, efficient data sharing and transferring, mobile heterogeneous cloud computing, intelligent transportation system, cyber-physical systems

ACM Reference Format:
Keke Gai, Longfei Qiu, Min Chen, Hui Zhao, and Meikang Qiu. 2017. SA-EAST: Security-aware efficient data transmission for ITS in mobile heterogeneous cloud computing. ACM Trans. Embed. Comput. Syst. 16, 2, Article 60 (January 2017), 22 pages.
DOI: http://dx.doi.org/10.1145/2979677

1. INTRODUCTION

The dramatic growth of wireless networks has been advanced in enormous applications from different dimensions. The *Intelligent Transportation System* (ITS) is one of the significant fields in *Cyber-Physical Systems* (CPSs), which has a tight relationship with real-time data sharing and transferring within Web-based solutions [Boban et al. 2011]. The rising applications of ITS have enabled a great exploration of *Vehicular*

This work is supported by the National Science Foundation, under grant NSF CNS-1457506 and NSF CNS-1359557.
M. Qiu is the corresponding author of this work.
Authors' addresses: K. Gai and M. Qiu, Department of Computer Science, Pace University, NY, 10038, USA; emails: {kg71231w, mqiu}@pace.edu; L. Qiu, Nanjing Foreign Language School, Nanjing, China; email: long feiqiu2012@gmail.com; M. Chen, School of Computer Science and Technology, Wuhan, China; email: minchen2012@hust.edu.cn; H. Zhao, Software School, Henan University, Henan, 475000, China; email: zhh@henu.edu.cn.

DOI: http://dx.doi.org/10.1145/2979677

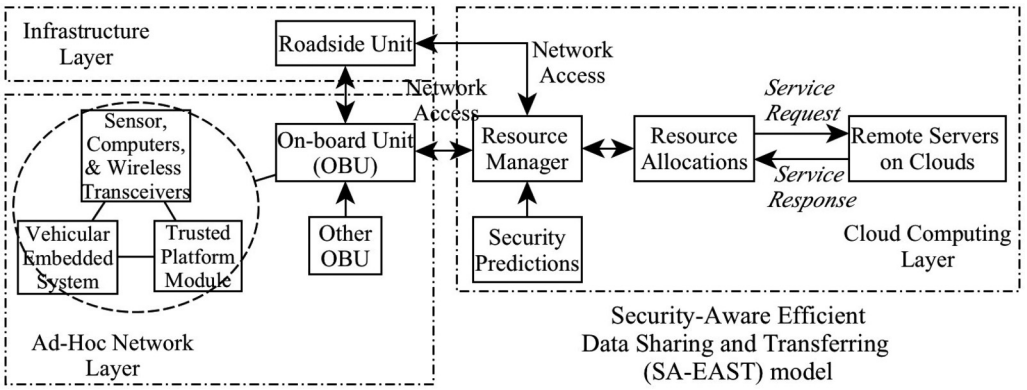

Fig. 1. Architecture of *Security-Aware Efficient Data Sharing and Transferring* (SA-EAST) Model. It includes *Infrastructure*, *Ad-Hoc Network* (AHN), and *Cloud Computing* layers.

Communication Systems (VCSs) for multiple purposes, such as safety enhancement, traffic efficiency, and driving assistance. Deploying a smart VCS is also considered a critical aspect of employing a smart city [Wu et al. 2012] because the communications between vehicles play an important role in achieving scalable service scopes [Calandriello et al. 2011]. For example, an emerging technology *Vehicular Digital Video Recorder* (VDVR) has been broadly applied in monitoring vehicles' behaviors for the purpose of protecting drivers. However, implementing mobile systems on a rapid moving object is still facing a major concern regarding unexpected adversaries while high-performance communications are needed as well. A cloud-based solution exhibits its potential effectiveness even though it encounters a few constraints caused by multiple factors [Gai and Li 2012; Gai et al. 2016b]. A high moving speed generates a dynamic manipulating context that makes the whole system complicated [Karagiannis et al. 2011]. This article focuses on both security and efficiency issues when executing ITS and proposes a novel approach focusing on real-time secure communications.

Currently, there exist a number of challenges in deploying mobile vehicular digital systems due to the limitations of wireless communications. For example, migrating data to cloud servers requires a stable communication performance in a dynamic operating environment. The rapidly moving objects can lead to an intermitted or unstable connection to cloud resources, which implies that the quality of cloud services are changing with network dynamics. Maintaining the operations of VCS securely is a challenge in ITS. For example, adversaries can attack objects when mobile users switch communication channels. To address this issue, we propose a novel secure approach of applying VCS in a smart city, which is called the *Security-Aware Efficient Data Sharing and Transferring* (SA-EAST) model.

Figure 1 represents the architecture of the SA-EAST model, illustrating a brief relationship mapping for critical entities in the system. Three layers are involved in the model: the *Infrastructure*, *Ad-Hoc Network* (AHN), and *Cloud Computing* layers. At the AHN layer, the mobile users who are attached to *On-Board Units* (OBUs) also receive messages from other OBUs. The main components of the OBU have sensors and wireless transceivers, vehicular embedded systems, and trusted platform modules. Next, at the Infrastructure layer, *Roadside Units* (RSUs) are stations that receive signals from mobile users and connect the users with a *Resource Manager* (RM). This layer is usually deployed on the highway, where vehicles move at high speed. Finally, an RM at the *Cloud Computing Layer* (CCL) is responsible for assigning tasks to different remote cloud servers in proportion to real-time server capacities. An operation of resource allocations is completed to address the service requests and responses by

communicating with the remote servers on clouds. The security prediction operations are done by RM.

Furthermore, the proposed mechanism focuses on the functionality and performance issues of ITS, in which the services are delivered in a dynamic cloud-based environment. Our SA-EAST model uses a dynamic resource allocation mechanism for supporting an efficiency-aware data migration toward cloud servers. The proposed mechanism is based on vehicular embedded systems capturing video data via vehicular cameras and sensors, and dynamically selects the optimal cloud server for data transmissions. Before the tasks are assigned to different cloud servers, RMs will map all available computing sources by weighting both security and efficiency. The weight of security is determined by security predictions, which is a mechanism analyzing unusual communication behaviors. Moreover, the distributed cloud server deployment can enable real-time communications as well as a long-term large-sized data storage. After the distributed data receivers obtain the video data, they will send the data package to a datacenter in which users' location-aware data are maintained.

In addition, our model focuses on the transmission restrictions that mainly take place when transferring video data to the clouds as well as gaining real-time digital video management. Focusing on this problem, we propose two crucial algorithms used in our proposed model. The first is the *Cloud Resource Mapping (CRM)* Algorithm, which targets having a panoramic view of cloud servers, such as capacity, data processing time, transmission costs, and geographic distances. The mapping process is a regular check for the availability of cloud computing resources. The second algorithm in SA-EAST is the *Security-aware Computing Resource Assignment (SCRA)* Algorithm, which executes an optimal resource allocation among distributed cloud servers. This algorithm derives from the *Min-Min Scheduling Preemptable Task* algorithm [Li et al. 2012].

The proposed model is significant by the use of a jumping-off point of security-aware efficient data transmissions in ITS. The deployment of the proposed model relies on the cloud-based server using dynamic server selection methodology to reach real-time data migrations. The security level can be improved by analyzing communication states. Next, the solution supports concurrent multiple tasks generated by VCS, which can reduce the latency caused by large-sized video file transmissions. Meanwhile, the utilizations of the resource allocations mainly concentrate on task scheduling among multiple cloud servers. This field has rarely been explored by prior research; we evaluate the proposed model in experiments.

The main contributions of this article are threefold:

(1) We propose a mobile heterogeneous cloud implementation using dynamic task assignments to achieve high-security performance of wireless transmissions in ITS. The concentration of the proposed scheme is increasing both security and efficiency for CPS applications with heterogenous cloud computing, in which a higher-level performance can be achieved by our proposed distributed parallel computing method.
(2) We propose an approach of mapping cloud resources that can be implemented in other systems for security-aware efficient solutions. We estimate the communication security states by selectively encrypting sensitive data in order to achieve real-time secure services.
(3) This article presents a novel ITS deployment that can be employed for securing ubiquitous CPS by using mobile heterogeneous cloud computing. The proposed scheme can not only ensure that sensitive data are encrypted, but also increase the amount of encrypted basic data, which depends on latency tolerance.

The remainder of the article is organized as follows. Section 2 presents related works in the relevant fields. Section 3 summarizes the main security issues of ITS in mobile

heterogeneous cloud computing. Section 4 presents a motivational example illustrating an implementation of the proposed model. Section 5 defines the main concepts used in the proposed model and explains the proposed mechanism. The main algorithms in the proposed model are displayed in Section 6. The experimental configurations and results are demonstrated in Section 7. We present our conclusions in Section 8.

2. RELATED WORK

2.1. Vehicular Cyber-Physical Systems Implementations and Improvements

Many studies on multimedia over VCS have been undertaken in recent years [Huang et al. 2011]. The performance of transmission is one of the main concerns in the research of VCS [Jafari et al. 2012]. The limitation of the wireless bandwidth is that the data exchange is executed in the continuous dynamic communication environment. One approach using compressive sensors was proposed for efficient transmissions, which used a time domain synchronous frequency-division multiplexing technique [Dai et al. 2013]. Considering the scenario of delay tolerance, a vehicular delay-tolerant network [Pereira et al. 2012] was proposed to overcome the limitations caused by the sparse networking connectivities. However, this field of research usually produces a trade-off between efficiency and security.

The explorations of video wireless transmissions have also been addressed by prior research in various fields [Carpi et al. 2011]. An approach using multipath transmission of video streaming traffic over multi-radio mobile devices has been proposed [Song and Zhuang 2012]. This research was based on probabilistic performance analysis. Other research focused on predicting wireless networks for gaining energy-efficient video transmission [Abou-zeid et al. 2014]. Similar research was also done by proposing a cross-layer resource allocation for scalable video transmission [Cicalo and Tralli 2014]. However, using heterogeneous mobile cloud computing along with dynamic cloud resource allocations has not been addressed by previous research.

Additionally, some prior research had addressed video streaming performance enhancement using cloud computing. One approach was migrating the hazards by using a graphical model to detect *Distributed Denial-of-Service* (DDoS) attacks [Wang et al. 2015]. The integration of SDN with cloud computing is utilizing the characteristic of networking virtualization via cloud systems [Jain and Paul 2013]. The main benefit of the integration is gaining advantages from both SDN and clouds, such as adding other cloud-based service types to SDN [Akyildiz et al. 2015]. However, these approaches cannot solve the problem of vehicular adversaries because of the challenge of wireless networking governance and threat detections.

Moreover, the efficiency of the *Mobile Digital Vehicular Recorder* (MDVR) has been studied from various perspectives in recent years [Wang et al. 2012a]. Data allocations for memory is one of the research directions that had been verified as an effective approach for efficiency enhancement from a hardware perspective [Gai et al. 2016a; Qiu et al. 2014]. Optimizations could be made when workloads were real-time constrained [Li et al. 2013]. However, dynamic different input datatypes can result in significant gaps between applications. Traditional data allocation optimizations can hardly satisfy the demands of task assignments to continuously changing servers.

2.2. Monitoring Unexpected Behaviors in Cyber-Physical Systems

Some improper driving behaviors were considered unexpected incidents for OBUs, and security issues was one of the major concerns explored by prior research [Gai et al. 2015; Batistatos et al. 2012]. One approach securing the video content proposed using joint compression and encryptions [Pande et al. 2013]. A lightweight intrusion detection mechanism was developed for service-oriented vehicular networks [Sedjelmaci et al.

2014]. Nonetheless, limited prior research attempted to use DVR to monitor real-time hazards.

Considering detection of malicious behaviors, many previous works had studied techniques for CPS. One approach finding temporal logic falsification of the systems was to detect falsifying behaviors in CPS using a predefined robustness metric that was confirmed by the metric temporal logic property [Abbas et al. 2013]. Using a reference providing constraints was the operating principle of this type of solutions, which could also be applied in increasing task scheduling efficiency for distributed CPS [Tang et al. 2012]. It also implied that the confirmed estimated criteria of the opportunistic parallel data processing could increase both reliability and operability [Balani et al. 2014]. However, the stated criteria detecting unusual behaviors for cloud servers has rarely been addressed as yet.

A few prior works also investigated the surveillance of unexpected or unusual operations in CPS. For instance, a study has been done for assessing security performance when the mixed criterion of criticality is applied in a *Radar Surveillance System* (RSS) [Lakshmanan et al. 2012]. This investigation used a formal overload-tolerance metric to locate unanticipated conditions and allocate computing resources. Similar research was processed by modeling and validating the applications' abstract manners in order to increase reliability and trustworthiness [Malik et al. 2012]. Nevertheless, surveillance-based solutions in CPS were rarely explored in securing dynamic wireless interconnected objects.

Next, applying mathematical models in VCS has been investigated by previous research in a few dimensions. An approach was proposed to increase highway safety and efficiency using the coordinated controls of vehicle platoons [Wang et al. 2012b]. Another study focused on scheduling electric station usage. One approach used a fluid dynamic traffic model as well as M/M/S queueing theory to optimize the schedule and distribute the charging stations for electric vehicles [Bae and Kwasinski 2012]. In a different perspective, another approach is adding an electric vehicular operator to have a holistic view of the charging demands [Ortega-Vazquez et al. 2013].

In summary, the problem we investigated has an urgent demand for solutions in both academic and industrial fields. The proposed model is an innovative approach for solving the adversary detection problem in a dynamic vehicular communications environment.

3. ITS SECURITY ISSUES IN MOBILE HETEROGENEOUS CLOUD COMPUTING

3.1. Main Security Threats

The main security threats impacting on the implementations of ITS in mobile heterogeneous clouds are mainly caused by the vulnerabilities of mobile networking communications. There are several types of malicious attacks perpetrated on mobile networks.

—*Channel-related attacks:* This type of threat mainly occurs when communication channels are maliciously occupied or monitored [Gebotys and White 2015]. The target attack channels can be either public or private. Attackers tap into communications to damage data packets, such as sending duplicate messages, adjusting data packages, and inserting harm messages. Channel congestion is also a common attack method: channels are maliciously occupied for disabling data deliveries. Channel bandwidth is fully expended so that the interconnections between communication nodes are interfered with or disconnected.

—*Node-based attacks:* Some malicious methods attack mobile networks via intruding or controlling communication nodes. There are a few approaches to launching this type of attack. First, some adversaries pretend that they are one of the nodes in the communications, in which the harmful information is spread. Second,

the attacker intrudes into the networks to duplicate and send out the same message as the manner of multiple nodes, by which the malicious node swindles other nodes. The receivers will hardly be able to identify the trustworthiness since the same message is repeatedly received. Third, the adversarial party can also interfere with the nodes' collaboration by impeding broadcasting messages to certain nodes in the networks, such as some *Denial of Service* (DoS) attacks [Serpanos and Voyiatzis 2013; Xiang et al. 2011].

—*Infrastructure vulnerabilities in networks:* Malicious activities also take place when the infrastructure in traffic management systems is abused or in improper operations. In ITS, RSUs play an interconnection role that is also an attack target for adversaries. Attackers usually use three intrusion methods. First, controlling an RSU to generate conflict monitor messages can disable other RSUs' functions. In addition, combining capturing infrastructure with node-based attacks can increase harm while the infrastructure can be considered a node in the network [Dua et al. 2014]. Finally, physical damage can result in dramatic unexpected abuses due to irregular operations.

3.2. Constraints of Security-Aware Efficiency Enhancement

The fundamental approach to protecting sensitive information is encrypting data before the data are transmitted in the networks, such that the adversaries can hardly have direct decipherings on the nodes. The constraints of this mechanism is that there is a contradiction between security and efficiency [Wang et al. 2014; Qiu et al. 2011]. Time consumption increases when the security level goes up. For reaching real-time services with low latency time, it is almost impossible to encrypt all transmissions due to the large-size data and continuous data generation. Addressing this conflict, our proposed model uses the *Computation Time-Oriented* (CTO) method to classify encryption targets. The operating principle of using the CTO method is dynamically selecting encrypted objects while ensuring that sensitive data are encrypted. The details of the proposed CTO operations are given in Section 5.

4. MOTIVATIONAL EXAMPLE

This motivational example simulates an OBU running on a highway that transmits video data to cloud servers via connecting RSUs [Karagiannis et al. 2011]. Concurrent wireless communications also include other tasks of VCSs due to other vehicular functionalities. It implies that the tasks are independent from each other, but some tasks have predecessor–successor relations.

In this example, there are eight independent data packets, marked {*A*, *B*, *C*, *D*, *E*, *F*, *G*, *H*}. Table I contains the security requirements of the input data packets and their corresponding time consumptions. There are eight data packets, from A to H. Each data packet X can select encrypt data (X1) or nonencrypt data (X2). Three working modes are offered by three cloud service providers: M1, M2, and M3. Service providers have various performances for different data packets. Moreover, there are three working modes provided by different cloud vendors, namely, *M1*, *M2*, and *M3*. Two options for each data packet are available, which means different security levels' operations by the selections of the encryptions. In the table, *1* means nonencryption and *2* means with encryptions.

In addition, we assume that A, D, and H are sensitive data that need encryptions. Other data packets have two options, either encrypting or nonencrypting data. Therefore, data packets A, D, and H only have the higher-level security options, A2, D2, and H2. In Table I, *Length* refers to the number of data units in the packet. The time consumption under each cloud resource is the time cost of each data unit. Figure 2 illustrates the *Data Flow Graph* (DFG) with predecessor–successor relations. Figure 2(a)

Table I. Security Requirements and Time Consumptions

DP	Length	TCPU		
		M1	M2	M3
A1	10	-	-	-
A2		2	3	4
B1	4	2	2	4
B2		4	4	6
C1	10	1	2	3
C2		2	3	5
D1	20	-	-	-
D2		1	2	5
E1	5	2	3	5
E2		4	5	7
F1	2	1	1	2
F2		2	2	4
G1	7	2	3	3
G2		4	5	6
H1	15	-	-	-
H2		2	2	3

Note: DP = Data Packets; TCPU = Time Consumption Per Unit; the length is counted by units.

Table II. Data Packets Assignment

Layers	Length	DP	TC		
			M1	M2	M3
i	10	A2	20	30	40
ii	15	H2	30	30	45
	10	C1	10	20	30
		C2	20	30	50
	4	B1	8	8	16
		B2	16	16	24
ii	20	D2	20	40	100
	7	G1	14	21	21
		G2	28	35	42
	5	E1	10	15	25
		E2	20	25	35
	2	F1	2	2	4
		F2	4	4	8

Note: DP = Data Packets; TC = Time Consumption; the length is counted by units.

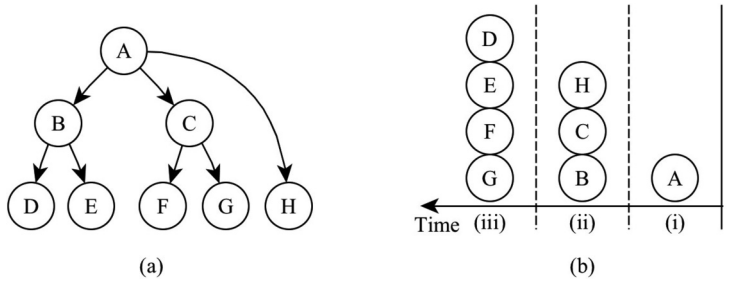

Fig. 2. Dataflow for the motivational example. Figure 2(a) represents a dataflow graph for the given data processing tasks. Figure 2(b) represents task flow using parallel computing.

represents the DFG of the given eight data packets. As shown in this figure, data A needs to be transmitted before B, C, and H. Data D and E can be processed after B. Data C must be processed before F and G. Therefore, we can allocate data packet transmissions to different clouds by different layers. Figure 2(b) displays the task flow using parallel computing. For instance, B, C, and H can be parallel computed because all these data packets can be executed after A, as shown by the layer *ii* in Figure 2(b).

Furthermore, we generate a table for assigning data packets based on these elements. Table II illustrates the optimizations of the task assignments using distributed cloud resources. First, we group the data packet into three layers, which derives from Figure 2(b). Next, at each layer, we sort the data packets in a descending order according to the number of units. Once all time costs are finalized, we start determining the data packet assignments. The assignments are in descending order and the succeeding working mode selection needs to be associated with the preceding assignment.

Next, repeat the steps outlined earlier and we eventually obtain the following assignments: $A2 \rightarrow M1$, $H2 \rightarrow M1$, $C2 \rightarrow M2$, $B2 \rightarrow M3$, $D2 \rightarrow M1$, $G1 \rightarrow M2$,

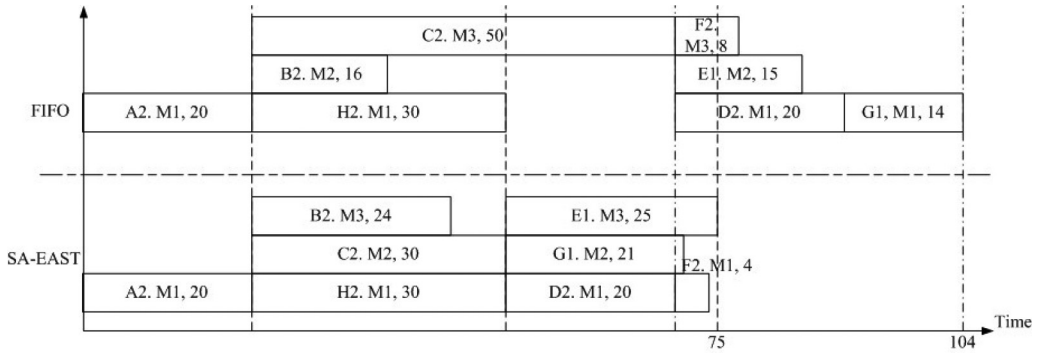

Fig. 3. Efficiency performance comparison at the same security level between FIFO and SA-EAST.

$E1 \rightarrow M3$, and $F2 \rightarrow M1$. The total time is 75 from (20+30+25) by encrypting data A, D, and H. Figure 3 is an efficiency performance comparison between the *First-In-First-Out* (FIFO) method and SA-EAST. The order of task processing is $A2 \rightarrow C2 \rightarrow B2 \rightarrow H2 \rightarrow F2 \rightarrow E1 \rightarrow D2 \rightarrow G1$. The examined targets have the same security performances, as well as the same cloud service providers. The result is that SA-EAST has a shorter execution time than FIFO, which has a 29 time unit difference deriving from (104 – 75).

5. CONCEPTS AND THE PROPOSED MODEL

5.1. Problem Definition and Main Concepts Used in the SA-EAST Model

Definition 1 (*Data Packets Assignment Problem on Heterogeneous Clouds*). Given a set of data packets that need to be transmitted, which consist of both sensitive data and basic data. The data transmission timing costs are varied and the information of transmission capacities is available. The problem is to determine an approach minimizing the execution time by assigning data packets to heterogeneous cloud servers, ensuring that the sensitive data are encrypted and partial basic data are encrypted.

The inputs are cloud server availability, historical execution time for each type of data, and input data packets. The input data packets need to be separated into several independent data subpackets. The required information includes the length of each data packet and the corresponding time consumption operated by each working mode. At least two working modes are offered by cloud service providers, including encryption and nonencryption. The output is an assignment plan that assigns data packets to heterogeneous clouds for minimizing execution time as well as increasing security level. The main concepts used in our proposed SA-EAST model are defined. The following itemized definitions are crucial entities and contexts in the model.

—*Cloud Manager:* An interconnector managing and assigning tasks to cloud resources depending on the OBUs' geographic positions and service contents. We use notation $\mathbf{C}=\{C_1, C_2, \ldots, C_n\}$, where $n \in N$.
—*Terminal Cloud Server:* A group of cloud servers in which the video data are eventually stored, operated, and maintained for specific purposes of the usage.
—*Heterogeneous Mobile Cloud Computing:* In the SA-EAST model, this term refers to various cloud vendors offering OBUs different service contents, qualities, or performances due to the various techniques and locations.
—*Cloud Server:* Cloud servers in SA-EAST is an exchangeable term with *Distributed Cloud Servers* (DCS) and cloud resources. We use the notation $\mathbf{R}=\{R_1, R_2, \ldots, R_n\}$, where $n \in N$. The capacity information for each server is the input of SA-EAST.

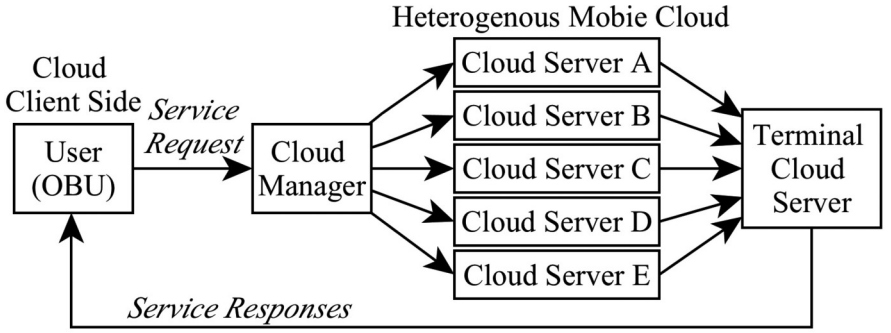

Fig. 4. Mapping process structure. Wireless connections are processed by deploying RSUs, which connect cloud clients with the remote cloud.

—*Sensitive Data:* Those data must be encrypted in wireless communications.
—*Basic Data:* Refers to the data that can be either encrypted or nonencrypted, which depends on latency tolerance. Encrypting these data can increase the system's entire security level.

We consider time consumption a critical factor since the key concentration of our approach is efficiency. We use $\mathbf{T}$, $\mathbf{T} = \{T_1; T_2; \ldots; T_n, n \in N\}$ to denote the total time consumption, which consists of a variety of timing consumptions caused by different sources, such as wireless transmissions, video operations, audio operations, and data analysis. For each observed time spot, we use $\{t_1, t_2, \ldots, t_n\}$, where $n \in N$.

Figure 4 is a structure illustration of the mapping process. As illustrated in the figure, the running vehicle is an OBU that captures video data during the driving period. The data inputs may consist of a variety of video file formats with different container formats since the VDVR may support multiple video recording devices synchronously. Therefore, various coded video and audio data may be in different data types. Other data are also in the queue since VCS is applied by deploying embedded systems [Qiu and Sha 2009] with other wireless services, such as the *Global Positioning System* (GPS), *Vehicle Speed Limit Alarm System* (VSLAS), or *Mobile Intrusion Detection System* (MIDS).

Meanwhile, the communications are delivered by connecting to a set of RSUs. The received signals are sent to the cloud manager, which arranges task assignments for heterogeneous mobile cloud computing. The cloud manager looks for available distributed cloud servers and maps their conditions with capacities [Qiu et al. 2015]. Data analyses are done on distributed cloud servers. The results are sent to the terminal cloud server for real-time data usage or maintenance. For instance, a real-time camera-view monitor can be implemented.

In order to ensure real-time wireless video transmissions, the critical part is to minimize the latency when transmitting a large packed video file. It is a major restriction for MVCR real-time transmissions, since a long queue can cause a large latency and influences other functions of VCS. Moreover, an important factor is that the computation time can vary due to the executions operated by different hardware, even though the same task is performed. For instance, compared with CPUs, *Graphics Processing Units* (GPUs) can have a higher performance when encountering a similar operation. This phenomenon forms a heterogeneous cloud environment that offers selections for cloud users. Our proposed algorithm mainly focuses on this issue and provides an efficient video data transmission. This issue is related to not only networking bandwidth, but also cloud server computation capacities and conditions.

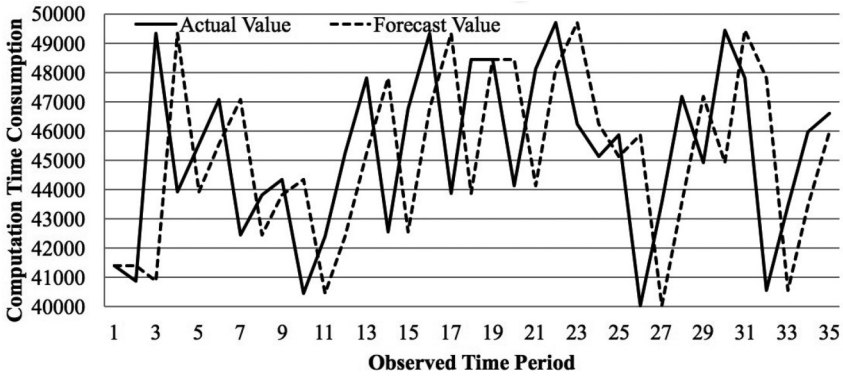

Fig. 5. Example of forecasting values using the exponential smoothing technique. $\alpha = 0.6$.

5.2. Computation Time-Oriented Security Enhancement Using Greedy Algorithm

The mechanism of CTO consists of two steps. The first step is categorizing all data packets into two groups, including sensitive and basic data. A table mapping all time consumptions under cloud resources needs to be generated. The results will be used to determine the server selection later. Next, we use the DFG method to identify the layers by which the data packets are grouped. At each layer, we sort the data packets depending on their lengths; then, the data packets are assigned one by one. For each data packet, the cloud server offering the shortest execution time will be selected. During the manipulative process, the server is occupied and other data packets need to select other available servers. The cloud manager uses this method to assign all data packets until all layers are accomplished.

5.3. Mapping Cloud Resources Using Exponential Smoothing Techniques

The first phase of our proposed model is to identify the cloud servers' capacities. It requires a prediction since the performances of the cloud servers are influenced by a variety of factors. The technique that we use for forecasting performance trends is the *Exponential Smoothing Technique* (EST). EST is a technique that analyzes historical trends for forecasting the value of the next time period [Chan et al. 2012]. The equation of EST is Equation (1):

$$F_{t+1} = F_t + \alpha(R_t - F_t), \tag{1}$$

where F refers to a forecasting value. F_{t+1} and F_t denote the forecasting values at time spots/periods *t+1* and t, where t refers to a time spot/period. R_t denotes the real value at time t. α is a *Smoothing Weight* (SW) that is scoped in $0 < \alpha < 1$. Therefore, we gain the function as follows:

$$F(t+1) = f(F(t)). \tag{2}$$

Figure 5 is an example of using exponential smoothing technique to forecast computation time consumptions with $\alpha = 0.6$. In the figure, the broken line refers to the forecast values and the solid line refers to the actual value gained from the observations. The actual values can be obtained from cloud managers in our model.

Therefore, addressing the total time consumption, we use the forecast technique to map all available cloud resources. Assume that there are group cloud resources $\mathbf{R}$. $F_t^{R_i}$ denotes the forecast time consumption. Since the time consumption derives from various components, we consider the total time a sum of time consumptions generated from a set of data transmission and processing, such as video, audio, vehicular position, and other functions [Pang et al. 2014]. We use $F_t^{R_i}(j)$ to denote each component's

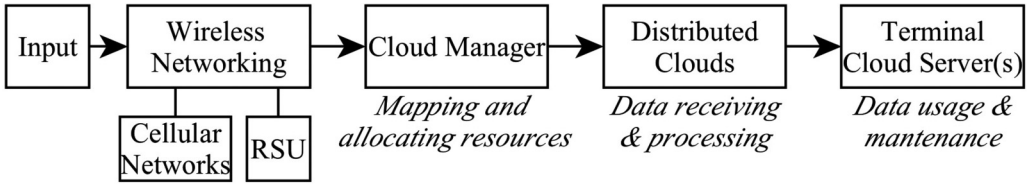

Fig. 6. Operating structure of efficiency-aware cloud computing resource allocations.

time consumption. Correspondingly, the actual time cost is $T_t^{R_i}(j)$. Therefore, we can calculate the forecasting time cost on each cloud server by using Equation (3), and the actual time cost is gained by Equation (4):

$$F_t^{R_i} = \sum_{j=1}^{m} F_t^{R_i}(j) \tag{3}$$

$$T_t^{R_i} = \sum_{j=1}^{m} T_t^{R_i}(j). \tag{4}$$

The total time consumptions of all cloud servers are gained from summing up $F_t^{R_i}$ and $T_t^{R_i}$. To minimize the execution time, we always select the shortest execution time, which is denoted as j_{min}. We predict the total time consumption value from Equation (5) and the actual time cost is gained by Equation (6):

$$T_{forecast} = \sum_{i=1}^{n} F_t^{R_i} = \sum_{i=1}^{n}\sum_{j=1}^{m} F_t^{R_i}(j_{min}) \tag{5}$$

$$T_{actual} = \sum_{i=1}^{n} T_t^{R_i} = \sum_{i=1}^{n}\sum_{j=1}^{m} T_t^{R_i}(j_{min}). \tag{6}$$

We use Equations (5) and (6) for the purpose of mapping real-time cloud severs' status and availabilities.

5.4. Cloud Resource Allocations Minimizing Time Consumptions

We also propose an approach allocating tasks to various cloud resources in order to accomplish tasks in the shortest time period. As described in Section 4, data collections are sent to a layer called the cloud manager that is responsible for assigning tasks to distributed cloud servers. The main challenge of this step is that the manipulative process is dynamic due to dramatic position changes that can result in the varied performances. For the purpose of proper selections, we use the technique introduced in Section 5.3 as well as a greedy algorithm, the SCRA algorithm.

Figure 6 is an operating diagram of cloud computing resource allocations. We divide the long waiting queue into a number of subtasks when the tasks are independent from each other. As mentioned earlier, we select the shortest time consumption from a set of available cloud servers, $\{R_1, R_2, \ldots, R_i\}$ $(i \in N)$. We define each subtask as an S_k, and there is a set $\mathbf{S}=\{S_1, S_2, \ldots, S_k\}$ $(k \in N)$. A selection operation determining the shortest time period is denoted as $\theta(j, S, t)$, which refers to the cloud server j, subtask S, and time period t. The computation of θ follows the following formulation, for which *Min* refers to the shortest time.

$$\theta = ^{Min}\left[\prod_{i=1}^{j} F(t+1, S_k, R_i)\right] \tag{7}$$

Therefore, considering the wireless transmission time T_{trans}, using the SCRA algorithm can further formulate the calculations as follows:

$$T_{\text{forecast}} = \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{s} \theta \left\{ Min \left[\prod_{i=1}^{j} F(t+1, S_k, R_i) \right] \right\} + T_{trans}. \tag{8}$$

The following section focuses on explaining the crucial algorithms used in the model.

6. ALGORITHMS

6.1. Cloud Resource Mapping (CRM) Algorithm

ALGORITHM 1: Cloud Resource Mapping (CRM) Algorithm

Require: $\mathbf{R}$, $\{T_i^h(S_k)\}$.
Ensure: $\mathbb{M}$

```
The OBU sends out the service request to the Cloud Manager that searches the idle cloud servers.
if cloud server is idle/available then
    R_temp ← R_i
end if
List all R_i in R_temp; input T_i^h(S_k)
for all target measured data input types I_i in R_i ∈ R_temp do
    for all cloud servers R_i do
        if historical dataset is not empty then
            MeasureContainer ← {T_i^h(S_k)}
            Using Equation (1) to generate predictions, obtain the shortest time consumption T_i^short(S_k)
            M ← T_i^short(S_k)
        end if
    end for
end for
RETURN M
```

The CRM algorithm is designed for mapping the cloud resources by tagging resource capacities and predicting accomplishment time. The inputs of the algorithm include the historical trend data from cloud servers. Addressing the computing capability. we aim to measure the server capacity by considering a few data input scenarios. For instance, we measure the performance differences caused by the diversity of hardware, such as CPU, GPU, and memory. The measurement is based on predictions using the historical data.

Algorithm 1 contains pseudocodes for the CRM algorithm. The target measured cloud servers are within the set $\mathbf{R}$. The corresponding historical execution time T for each type of data input is represented as $T_i^h(S_k)$. Different data inputs are denoted as I_i. The output will be a table $\mathbb{M}$ that maps all forecasting computation time with the corresponding data type inputs. As shown in Algorithm 1, there are a few main phases for the implementations.

(1) Input required datasets, including recent historical reference datasets and cloud server information, which will be used for server capacity analysis.
(2) Search all idle cloud servers and add the information to a temporary dataset R_{temp}.
(3) Input the historical reference dataset and start the prediction process using exponential smoothing techniques.
(4) Try all situations if the reference dataset is nonempty and add the results to the table $\mathbb{M}$.
(5) Output the $\mathbb{M}$ after all predictions are done.

6.2. Security-aware Computing Resource Assignment (SCRA) Algorithm

The SCRA algorithm is designed to use the outcomes of the CRM algorithm and dynamically select the cloud servers. The computation time is mapped by predictions from analyzing the historical datasets. The input data packages, *Data*, consist of a number of components that are defined as *fields*. The output will be an allocation plan for assigning tasks to heterogeneous cloud resources, which is represented as *AssignmentPlan*.

Algorithm 2 contains pseudocodes of the SCRA algorithm. In the algorithm, *machine* denotes a cloud server.

ALGORITHM 2: Security-aware Computing Resource Assignment (SCRA) Algorithm

Require: A data package: *data*, $\mathbb{M}$
Ensure: Allocation plan: *plan*

```
Initialize an endTimeList, input M, Temp← ∅
while ∃ data are not assigned do
  for ∀ fields in data do
    for ∀ machine do
      endTimeList.add(sum up machine.freeTime and machine.processTime(field))
    end for
  end for
  for ∀ input data packets do
    if the data packet does not have preceding task then
      Temp ← data packet
    end if
  end for
  for all data packets in the set Temp do
    Sort data packets according to the lengths in a descending order
    Select the minimum endTime in endTimeList
    Assign the field process on the machine that has minimum endTime
    machines.freeTime ← minimum endTime
    /*Forward the tasks to other cloud servers*/
  end for
end while
RETURN AssignmentPlan
```

The main phases of the SCRA include the following steps:

(1) Input data packages and the mapping table generated from the output of executing Algorithm 1. Obtain data from $\mathbb{M}$, list cloud servers' available time, and predict processing time consumptions.
(2) Calculate the time consumptions and sort them to a list according to the forecasting accomplishment time.
(3) Group data packets by using CTO methods. Sort the data packets at each layer based on the lengths of the data packets.
(4) Select the minimum accomplishment time from the list and determine the cloud servers to which the task is assigned. Forwarded tasks must be accomplished within a shorter time period.
(5) While there exist data that are not assigned, repeat the preceding steps till all data are assigned.
(6) Output the *AssignmentPlan* and execute the plan.

7. EXPERIMENT AND THE RESULTS

For the purpose of evaluation, we perform a series of experiments to simulate the practical implementations. Experimental configurations are given in Section 7.1 and

crucial experimental results are presented in Section 7.2. Finally, Section 7.3 presents discussions about findings and future work.

7.1. Experimental Configurations

We conducted experiments on our own written simulation environment assuming that vehicles send out a package every second. The hardware used in our experiments included an HP server having the following hardware configuration: 8-core CPU, a 16GM memory, and a MySQL 5.7. We used the VMWare workstation operated within a Ubuntu 15.04 LTS server in order to simulate cloud computing. In addition, the experiments consisted of a few examinations. First, we evaluated the performance of the CRM algorithm. Two settings simulate four cloud servers: clouds *A, B, C*, and *D*. We aimed to measure computation time consumption from two aspects: accuracy at the specific time period and performance at different data inputs. Accuracy was examined by comparing the actual time consumptions with the forecasting values.

We also evaluated the performance of the SCRA algorithm by simulating the total time consumptions, including data transmission and data processing. In a data package, there were 5 characteristic fields measured in our experiment, including video data, audio data, the vehicle's coordinates, the vehicle's velocity, and gasoline volume. Moreover, we set up a number of cloud servers using different configurations in this simulation system. The configurations are based on defining five different levels of capacities that are associated with the targeted characteristic fields.

We measured a variety of experimental scenarios using a group of experimental settings to evaluate the performance of the proposed resource allocation scheme. Two comparison targets were traditional cloud approaches and utilizing a supercomputer as cloud server. The performances of supercomputers and normal servers were configured by putting different parameters in our simulator. The configurations of the settings are as follows:

—*Setting 1:* We simulated a scenario of common cloud services, which cloud users connect with only one fixed performance cloud server.
—*Setting 2:* We simulated a supercomputer that has higher-level performances. The calculation speed is n times faster than one single cloud server.
—*Setting 3:* We simulated our proposed approach using m distributed cloud servers.
—*Setting 4:* We simulated a set of experiments to compare the SA-EAST with the FIFO method, which were based on the same security requirements and performances.

7.2. Experimental Results

In this section, we present some results gained from our experimental evaluations. Both algorithms were examined and partial outcomes are exhibited for the demonstration purpose.

First, Figure 7 represented a time comparison between the actual execution time costs and the forecasting total time costs using four cloud servers—assumed as *Cloud A, B, C*, and *D*—when the value α was 0.6. The figure displays 10 rounds of experimental results, which could prove that our mechanism could fit in the requirements of certain cases. The time gaps were all in the acceptable range. For measuring large-sized continuous input data, Figure 7 illustrates partial experimental results for comparing values between actual and forecasting consumptions within a great amount of experimental runs. The value of α was selected as 0.6 as well. The findings derived from Figures 7 and 8 could prove that prediction of value fluctuations was applicable for mapping server capacities.

Figure 9 presented a comparison of forecasting values for four cloud servers. According to the figure, cloud B usually performed better than other clouds due to shorter

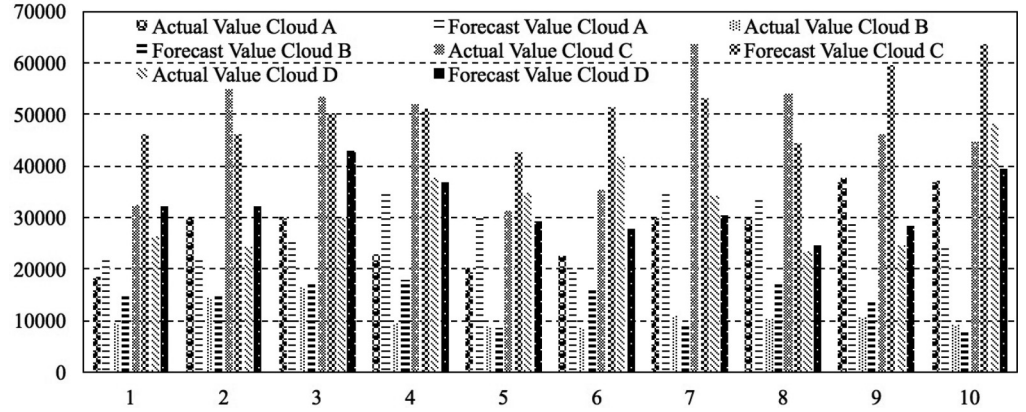

Fig. 7. Time difference comparing the actual total time consumptions with forecasting total time consumptions for four cloud servers, including clouds A, B, C, and D. Time is measured by milliseconds. Experimental runs: 10. $\alpha = 0.6$.

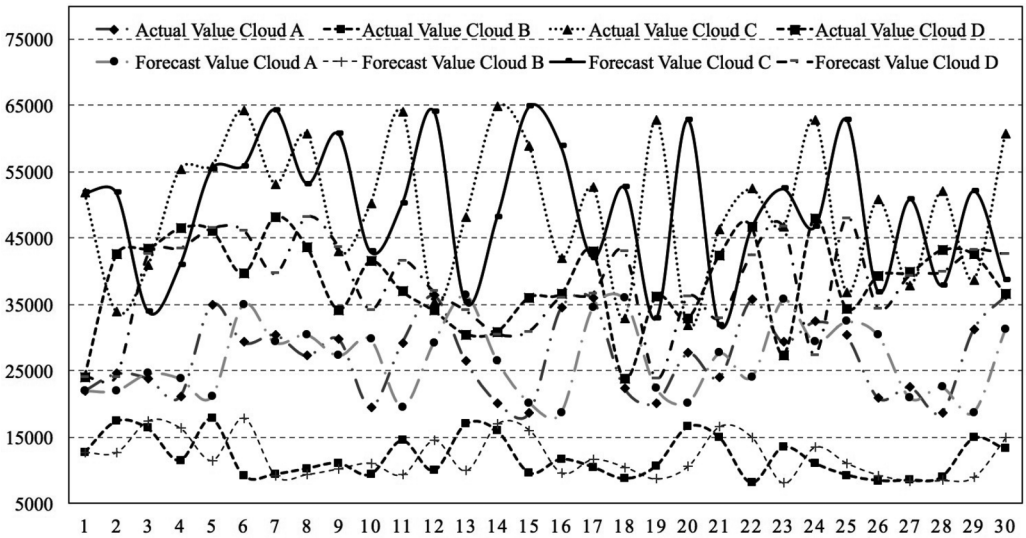

Fig. 8. Experimental results for comparing values between actual and forecasting consumptions within a great amount of experimental runs by using exponential smoothing techniques. Four cloud servers include clouds A, B, C, and D. $\alpha = 0.6$.

time consumptions. This cloud server was usually selected as the main processor in the clouds. The demonstrations from Figures 7 to 9 proved the feasibility of our proposed scheme. The following figures were generated from the evaluations of the SCRA algorithm.

Figure 10 presents a comparisons of delays among settings 1, 2, and 3. According to the figure, delay grew dramatically fast on setting 1 when execution time increased. Settings 2 and 3 had similar performances. It implied that traditional cloud services using limited fixed cloud servers had great difficulty in real-time video transmission and data processing. Our proposed scheme using distributed cloud resources and supercomputer-based solutions were two potential solutions.

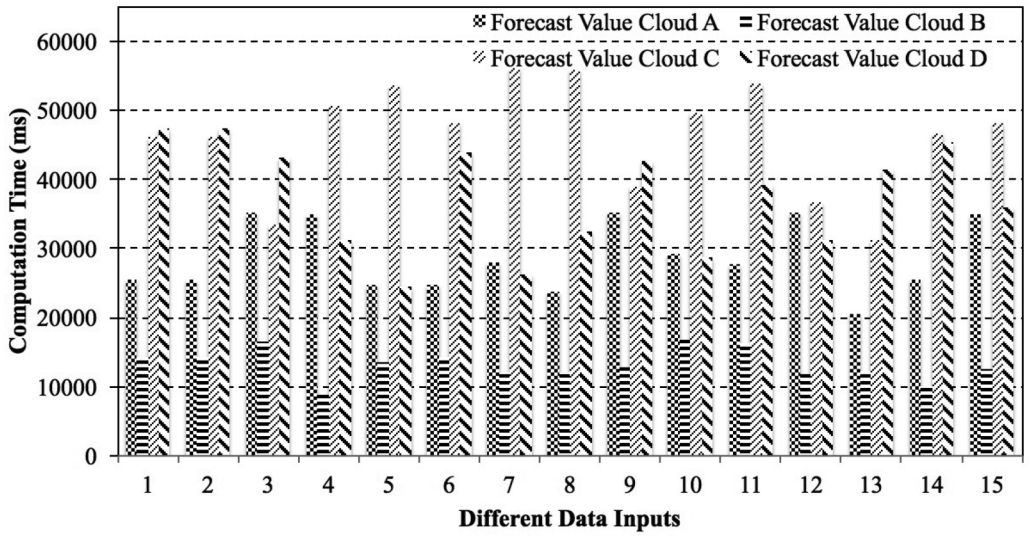

Fig. 9. A comparison of forecasting time consumptions on different data inputs for four cloud servers, including clouds A, B, C, and D, using exponential smoothing techniques. $\alpha = 0.6$.

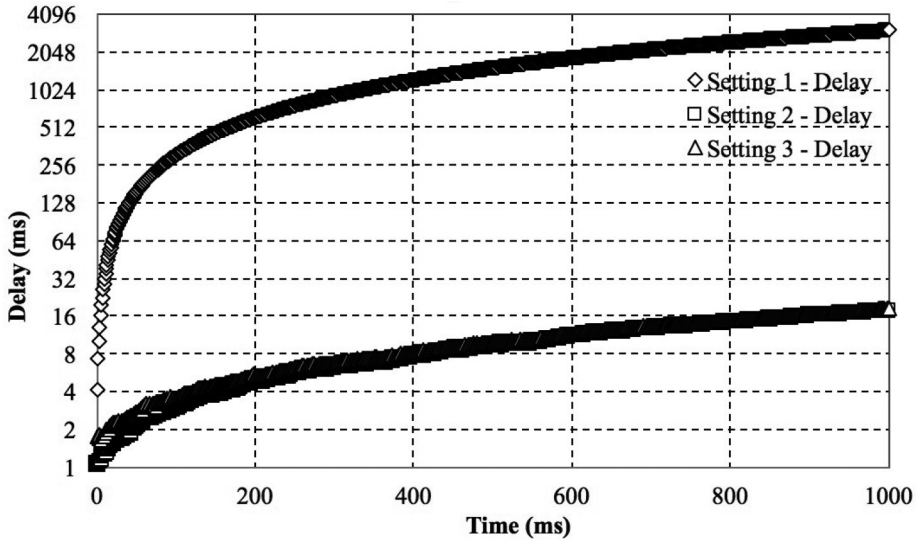

Fig. 10. Comparisons of delay distributions among settings 1, 2, and 3, associated with the execution time. Supercomputer capacity: 4 times faster than a single regular server. The number of distributed cloud resources: 4.

Figure 11 presented a comparison of time delay between our proposed approach and implementing supercomputers. The figure showed that our approach had a better performance than that of the supercomputer. The delay was limited into an acceptable range when executing SA-EAST. In addition, we examined the distribution of the data pack ages by measuring the dataset volumes. Figure 12 displayed a data sample showing the data volumes between different data types. *Data* 0 had the biggest volume, which was a video data file. Other data types required much less transmissions, which meant that the main transmission costs were produced by transferring video data.

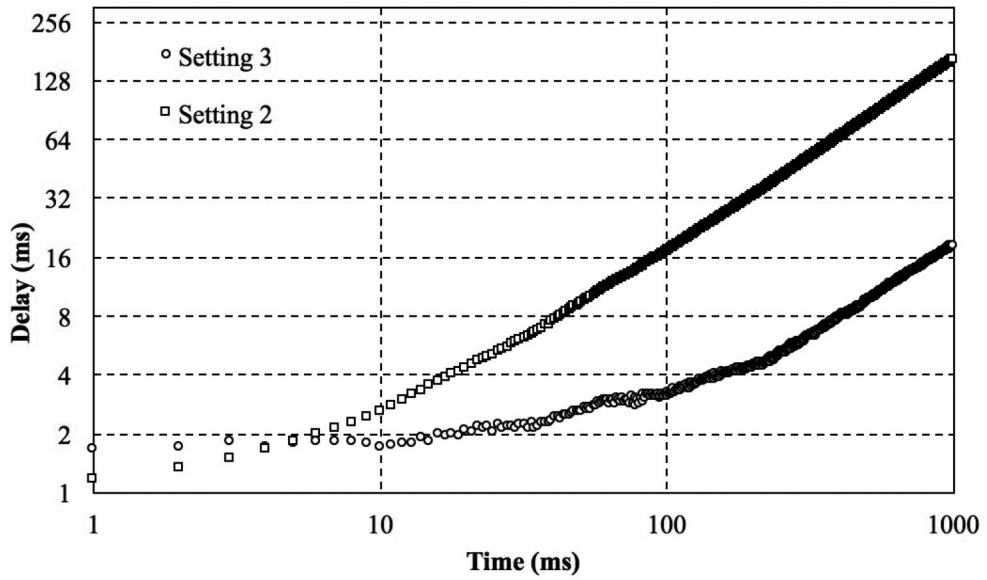

Fig. 11. Comparison of delay distributions between SA-EAST and using a supercomputer. Supercomputer capacity: 3 times faster than a single regular server. The number of distributed cloud resources: 4.

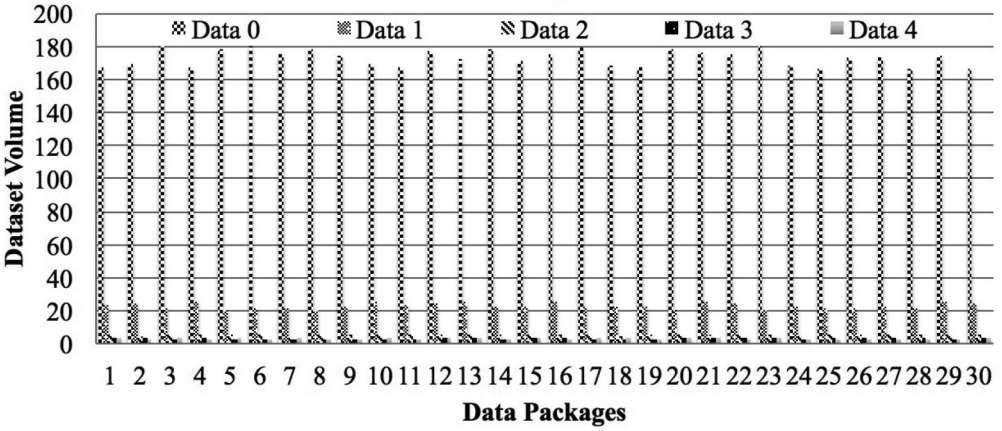

Fig. 12. Data volume distributions between various data types.

More experiments were done for evaluating the performance differences between SA-EAST and using supercomputers. Figure 13 illustrates a number of experimental results produced by simulating different supercomputers. The configurations were based on the speed of the supercomputers, which were 2, 3, 4, and 5 times faster than regular cloud servers. The figure shows that our proposed approach had a stable performance. However, the performances of supercomputer-based solutions had a strong relationship with the computation capability of the supercomputer.

Figure 14 demonstrates that our proposed scheme had an advantage of reducing execution time while security performance is the same compared with FIFO methods. According to the figure, the SA-EAST approach performed better in saving time than

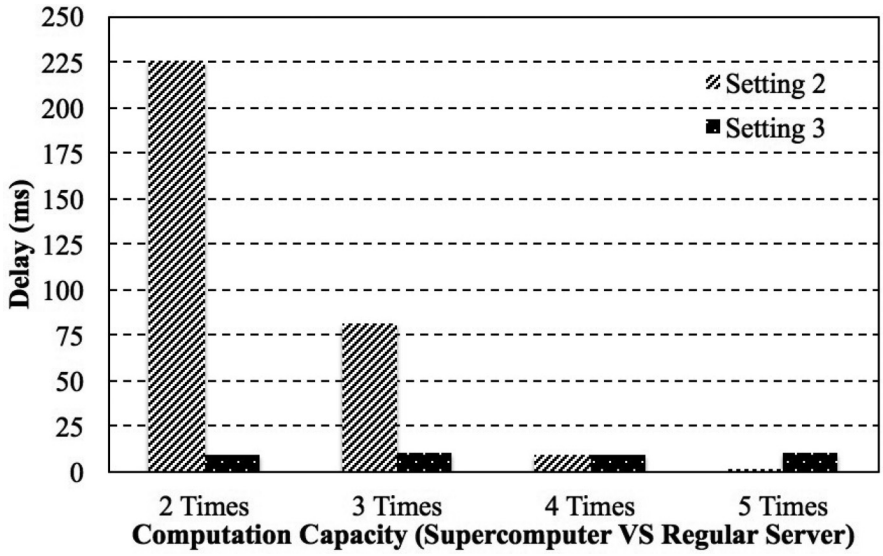

Fig. 13. Performance comparisons between settings 2 and 3 using different parameters. The number of distributed cloud resources: 4.

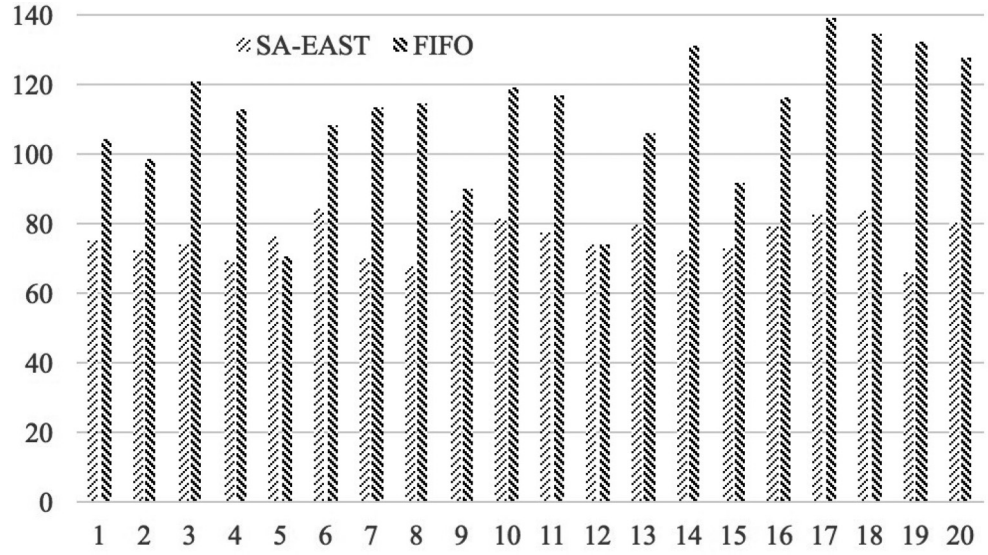

Fig. 14. Transmission time consumption (ms) comparisons between SA-EAST and FIFO with the same security performances. Experimental setting: 4.

FIFO in most situations. The influencing factors included the length of the data packets and the cloud servers' capabilities.

Finally, we focused on examining the proposed approach by using different cloud resource capabilities. The experiments were done by deploying different amounts of cloud server as well as the computation speeds. Our experimental results verified two assertions. The first was that the number of cloud servers had a positive relationship with efficiency. The second was that there was a critical point for the computation

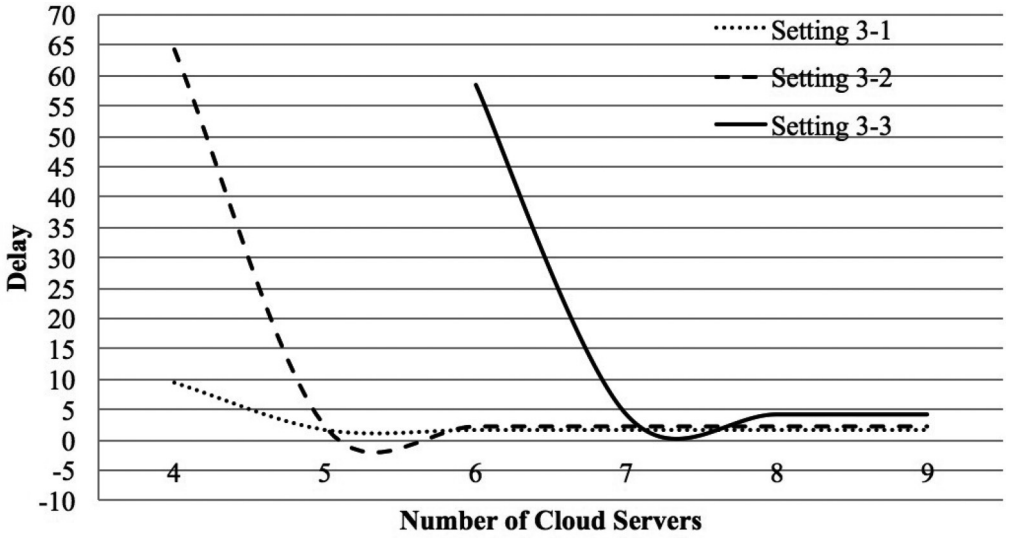

Fig. 15. Performance comparisons for setting 3 using different cloud resources.

capabilities. Performance would become stable once the calculation capabilities pass the critical point.

In summary, our experimental evaluations proved that our proposed approach could be a solution for supporting real-time video transmission attached to VCS. Our approach was superior to traditional cloud service offerings and had more stable performances than using supercomputers.

7.3. Discussions

Based on the results gained from our experimental evaluations, we found that our proposed approach was an efficient cloud-based solution for achieving real-time communications in ITS. Heterogeneous mobile cloud computing is deployed in our proposed model, which could work efficiently in most operation scenarios. For the purpose of a successful model implementation, we presented two suggestions to engineers and system designers.

First, our approach provided a method of using regular clouds for reaching high performance. In general, deploying supercomputers or high-performance computing facilities could assist in gaining a high-speed execution. However, this deployment was not affordable for most organizations. Thus, using a regular computing resource is one of the major benefits of using our approach. The whole performance could be similar or superior to using supercomputer-based solutions due to distributed parallel computing. Second, from a practical perspective, collecting historical data is significant for establishing an effective forecasting system. A proper configuration of dividing data in a periodic manner was an alternative for system designers.

Finally, our future work will be focused in two directions. The first will be to converge the proposed approach into the existing ITS system in order to examine practical performances and implementation feasibility. Multiple dimensions will be covered by this direction, such as big data, data fusion, and distributed data storage. The evaluation in the real-world context would be completed in our future work. The other research focus will be attempting to leverage the SA-EAST model in other systems, such as *Social Cyber-Physical Systems* (SCPSs), the *Internet-of-Things* (IoT), and smart cities.

8. CONCLUSIONS

This article proposes a novel cloud-based approach supporting real-time vehicular multimedia data transmissions when implementing VCS. The proposed model, SA-EAST, was an exploration of dynamically assigning data packet to cloud resources based on security requirements. Implementing the proposed scheme could not only protect sensitive data, but also increase the entire security level, which depends on the cloud servers' performances. Two main algorithms in SA-EAST were CRM and SCRA algorithms. The experimental evaluation proved the implementation feasibility and adaptability of the proposed scheme.

REFERENCES

H. Abbas, G. Fainekos, S. Sankaranarayanan, F. Ivančić, and A. Gupta. 2013. Probabilistic temporal logic falsification of cyber-physical systems. *ACM Transactions on Embedded Computing Systems* 12, 2s, 95.

H. Abou-zeid, H. Hassanein, and S. Valentin. 2014. Energy-efficient adaptive video transmission: Exploiting rate predictions in wireless networks. *IEEE Transactions on Vehicular Technology* 63, 5, 2013–2026.

I. Akyildiz, P. Wang, and S. Lin. 2015. SoftAir: A software defined networking architecture for 5G wireless systems. *Computer Networks* 85, 1–18.

S. Bae and A. Kwasinski. 2012. Spatial and temporal model of electric vehicle charging demand. *IEEE Transactions on Smart Grid* 3, 1, 394–403.

R. Balani, L. Wanner, and M. Srivastava. 2014. Distributed programming framework for fast iterative optimization in networked cyber-physical systems. *ACM Transactions on Embedded Computing Systems* 13, 2s, 66.

M. Batistatos, G. Tsoulos, and G. Athanasiadou. 2012. Mobile telemedicine for moving vehicle scenarios: Wireless technology options and challenges. *Journal of Network and Computer Applications* 35, 3, 1140–1150.

M. Boban, T. Vinhoza, M. Ferreira, J. Barros, and O. Tonguz. 2011. Impact of vehicles as obstacles in vehicular ad hoc networks. *IEEE Journal on Selected Areas in Communications* 29, 1, 15–28.

G. Calandriello, P. Papadimitratos, J. Hubaux, and A. Lioy. 2011. On the performance of secure vehicular communication systems. *IEEE Transactions on Dependable and Secure Computing* 8, 6, 898–912.

F. Carpi, N. Kastelein, M. Talcott, and C. Pappone. 2011. Magnetically controllable gastrointestinal steering of video capsules. *IEEE Transactions on Biomedical Engineering* 58, 2, 231–234.

K. Chan, T. Dillon, J. Singh, and E. Chang. 2012. Neural-network-based models for short-term traffic flow forecasting using a hybrid exponential smoothing and Levenberg–Marquardt algorithm. *IEEE Transactions on Intelligent Transportation Systems* 13, 2, 644–654.

S. Cicalo and V. Tralli. 2014. Distortion-fair cross-layer resource allocation for scalable video transmission in OFDMA wireless networks. *IEEE Transactions on Multimedia* 16, 3, 848–863.

L. Dai, Z. Wang, and Z. Yang. 2013. Compressive sensing based time domain synchronous OFDM transmission for vehicular communications. *IEEE Journal on Selected Areas in Communications* 31, 9, 460–469.

A. Dua, N. Bulusu, W. Feng, and W. Hu. 2014. Combating software and Sybil attacks to data integrity in crowd-sourced embedded systems. *ACM Transactions on Embedded Computing Systems* 13, 5s, 154.

K. Gai and S. Li. 2012. Towards cloud computing: A literature review on cloud computing and its development trends. In *IEEE 4th International Conference on Multimedia Information Networking and Security*. IEEE, Nanjing, China, 142–146.

K. Gai, L. Qiu, H. Zhao, and M. Qiu. 2016a. Cost-aware multimedia data allocation for heterogeneous memory using genetic algorithm in cloud computing. *IEEE Transactions on Cloud Computing* PP, 99, 1.

K. Gai, M. Qiu, L. Tao, and Y. Zhu. 2015. Intrusion detection techniques for mobile cloud computing in heterogeneous 5G. *Security and Communication Networks* PP, 99, 1–10.

K. Gai, M. Qiu, H. Zhao, L. Tao, and Z. Zong. 2016b. Dynamic energy-aware cloudlet-based mobile cloud computing model for green computing. *Journal of Network and Computer Applications* 59, 46–54.

C. Gebotys and B. White. 2015. A sliding window phase-only correlation method for side-channel alignment in a smartphone. *ACM Transactions on Embedded Computing Systems* 14, 4, 80.

X. Huang, Y. Xiang, A. Chonka, J. Zhou, and R. Deng. 2011. A generic framework for three-factor authentication: Preserving security and privacy in distributed systems. *IEEE Transactions on Parallel and Distributed Systems* 22, 8, 1390–1397.

A. Jafari, S. Al-Khayatt, and A. Dogman. 2012. Performance evaluation of IEEE 802.11 p for vehicular communication networks. In *8th International Symposium on Communication Systems, Networks & Digital Signal Processing*. IEEE, Poznan, Poland, 1–5.

R. Jain and S. Paul. 2013. Network virtualization and software defined networking for cloud computing: A survey. *IEEE Communications Magazine* 51, 11, 24–31.

G. Karagiannis, O. Altintas, E. Ekici, G. Heijenk, B. Jarupan, K. Lin, and T. Weil. 2011. Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions. *IEEE Communications Surveys & Tutorials* 13, 4, 584–616.

K. Lakshmanan, D. De Niz, R. Rajkumar, and G. Moreno. 2012. Overload provisioning in mixed-criticality cyber-physical systems. *ACM Transactions on Embedded Computing Systems* 11, 4, 83.

J. Li, M. Qiu, Z. Ming, G. Quan, X. Qin, and Z. Gu. 2012. Online optimization for scheduling preemptable tasks on IaaS cloud systems. *Journal of Parallel and Distributed Computing* 72, 5, 666–677.

J. Li, M. Qiu, J. Niu, L. Yang, Y. Zhu, and Z. Ming. 2013. Thermal-aware task scheduling in 3D chip multiprocessor with real-time constrained workloads. *ACM Transactions on Embedded Computing System* 12, 2, 24.

A. Malik, Z. Salcic, C. Chong, and S. Javed. 2012. System-level approach to the design of a smart distributed surveillance system using systemj. *ACM Transactions on Embedded Computing Systems* 11, 4, 77.

M. Ortega-Vazquez, F. Bouffard, and V. Silva. 2013. Electric vehicle aggregator/system operator coordination for charging scheduling and services procurement. *IEEE Transactions on Power Systems* 28, 2, 1806–1815.

A. Pande, P. Mohapatra, and J. Zambreno. 2013. Securing multimedia content using joint compression and encryption. *IEEE MultiMedia* 20, 4, 50–61.

L. Pang, X. Li, J. Chai, and Y. Liu. 2014. A high performance video frame extractor for omni media monitoring system. In *IEEE 12th International Conference on Signal Processing*. IEEE, Hangzhou, China, 399–402.

P. Pereira, A. Casaca, J. Rodrigues, V. Soares, J. Triay, and C. Cervelló-Pastor. 2012. From delay-tolerant networks to vehicular delay-tolerant networks. *IEEE Communications Surveys & Tutorials* 14, 4, 1166–1182.

M. Qiu, Z. Chen, Z. Ming, X. Qin, and J. Niu. 2014. Energy-aware data allocation with hybrid memory for mobile cloud systems. *IEEE Systems Journal* PP 99, 1–10.

M. Qiu, W. Gao, M. Chen, J. Niu, and L. Zhang. 2011. Energy efficient security algorithm for power grid wide area monitoring system. *IEEE Transactions on Smart Grid* 2, 4, 715–723.

M. Qiu and E. Sha. 2009. Cost minimization while satisfying hard/soft timing constraints for heterogeneous embedded systems. *ACM Transactions on Design Automation of Electronic System* 14, 2, 25.

M. Qiu, M. Zhong, J. Li, K. Gai, and Z. Zong. 2015. Phase-change memory optimization for green cloud with genetic algorithm. *IEEE Transactions on Computers* 64, 12, 3528–3540.

H. Sedjelmaci, S. Senouci, and M. Abu-Rgheff. 2014. An efficient and lightweight intrusion detection mechanism for service-oriented vehicular networks. *IEEE Internet of Things Journal* 1, 6, 570–577.

D. Serpanos and A. Voyiatzis. 2013. Security challenges in embedded systems. *ACM Transactions on Embedded Computing Systems* 12, 1s, 66.

W. Song and W. Zhuang. 2012. Performance analysis of probabilistic multipath transmission of video streaming traffic over multi-radio wireless devices. *IEEE Transactions on Wireless Communications* 11, 4, 1554–1564.

Q. Tang, S. Gupta, and G. Varsamopoulos. 2012. A unified methodology for scheduling in distributed cyber-physical systems. *ACM Transactions on Embedded Computing Systems* 11, S2, 57.

B. Wang, Y. Zheng, W. Lou, and Y. Hou. 2015. DDoS attack protection in the era of cloud computing and software-defined networking. *Computer Networks* 81, 308–319.

L. Wang, A. Syed, G. Yin, A. Pandya, and H. Zhang. 2012b. Coordinated vehicle platoon control: Weighted and constrained consensus and communication network topologies. In *IEEE 51st Annual Conference on Decision and Control*. IEEE, Maui, HI, 4057–4062.

M. Wang, R. Hong, G. Li, Z. Zha, S. Yan, and T. Chua. 2012a. Event driven web video summarization by tag localization and key-shot identification. *IEEE Transactions on Multimedia* 14, 4, 975–985.

Y. Wang, Y. Xiang, J. Zhang, W. Zhou, G. Wei, and L. T. Yang. 2014. Internet traffic classification using constrained clustering. *IEEE Transactions on Parallel and Distributed Systems* 25, 11, 2932–2943.

C. Wu, H. Mohsenian-Rad, and Jianwei J. Huang. 2012. Vehicle-to-aggregator interaction game. *IEEE Transactions on Smart Grid* 3, 1, 434–442.

Y. Xiang, K. Li, and W. Zhou. 2011. Low-rate DDoS attacks detection and traceback by using new information metrics. *IEEE Transactions on Information Forensics and Security* 6, 2, 426–437.

Received November 2015; revised May 2016; accepted July 2016