Security-aware job allocation in mobile cloud computing

1st Piotr Nawrocki
Institute of Computer Science
AGH University of Science and Technology
Krakow, Poland
piotr.nawrocki@agh.edu.pl

2nd Jakub Pajor
Institute of Computer Science
AGH University of Science and Technology
Krakow, Poland
jay.pajor@gmail.com

3rd Bartlomiej Sniezynski
Institute of Computer Science
AGH University of Science and Technology
Krakow, Poland
bartlomiej.sniezynski@agh.edu.pl

4th Joanna Kolodziej
Research and Technology Transfer Unit
Research and Academic Computer Network – NASK
Warsaw, Poland
joanna.kolodziej@nask.pl

***Abstract*—The ultimate goal of Mobile Cloud Computing is to allow users of mobile devices to execute their applications and complex numerical tasks on a broad range of cloud services and resources. One of the most challenging problems in the flow of mobile tasks related to remote cloud services is the security of all aspects of communication, service security and the reliability of cloud resources. In this paper, we developed a new security-aware job flow model for mobile computational clouds. In our model, we defined dedicated algorithm models such as the Filtration Algorithm and Prediction Module to generate an optimal secure system architecture for task and data processing and to ensure optimal cloud resource and service utilization. The robust performance of our model has been demonstrated by experimental analysis. Results of the experiments performed show that our flow model significantly enhances the security level of computations compared to a configuration in which computation time is the major criterion for job processing optimization.**

***Index Terms*—security, mobile cloud computing, task allocation, machine learning, adaptation**

I. Introduction

Cloud computing entails the exchange of computer and data resources across global networks; it constitutes a new value-added paradigm for network computing where higher efficiency, massive scalability and speed rely on effective software development. Cloud computing is rapidly becoming a popular infrastructure of choice among all types of organizations. Despite some initial security concerns and technical issues, an increasing number of institutions are considering moving their applications and services into the cloud. Consequently, mainstream Information and Communication Technologies (ICT) powerhouses such as Amazon, Microsoft, IBM, Apple, and Google are heavily investing in the provision and support of public cloud infrastructures [1].

Although significant effort has been devoted to migrating generic web-based applications into the Cloud, scant research and development have been invested in creating a unified tool-based framework for Mobile Cloud Computing (MCC) [2]. MCC entails the seamless enhancement of mobile device capabilities via flexible cloud resources (storage and computing), with special emphasis on the dynamic, on-demand offloading of operations under variable environmental conditions in order to maintain communication and interactive use.

Through offloading intensive computations or data storage, which saves considerable amounts of energy and storage space on the end-users' mobile devices, MCC can be very beneficial for mobile cloud users. However, just as in conventional cloud systems, the use of the cloud for mobile devices poses a lot of questions in terms of security and trust issues. In general, security in mobile clouds encompasses two main issues, i.e. platform reliability and data and privacy protection [3]. In the case where the MCC environment is defined as a set of mobile devices connected to a remote cloud, a remote cloud server would be the same as a conventional cloud computing provider, with the same general cloud security threats. Potential attackers, which may include parties injecting malware, end users or internal users, aim to destroy cloud services and MCC components. Examples of such attacks are battery exhaustion attacks [4] and mobile botnets [5]. The offloading of sensitive data to the cloud usually means the loss of direct control over these data. The same happens when jobs are offloaded through wireless communication channels: computationally intensive tasks are sent directly to remote cloud servers and then distributed among the entire environment for low-cost execution. In both cases, it is important for cloud providers to ensure proper recovery mechanisms for cloud data and services in case of any failure of cloud servers and services or in case of an external attack, while keeping resource utilization cost relatively low. On the other hand, end users may have specific requirements concerning the protection of their data, including authentication procedures, which make the entire problem even more challenging.

In order to support security-aware resource mapping and deal with evolving requirements, secure mobile cloud re-

source management systems are needed that are capable of continuously managing the reservation process by monitoring current service and data requests, amending future service and data requests and automatically adjusting schedules to accommodate the dynamically changing demand for resources. Whereas users need to make decisions to select suitable providers and negotiate with providers to achieve "perfect" secure service contracts, providers need to make decisions on selecting the right requests to accept and execute depending on the availability of resources, current and future demand for services, and data storage. All this should be done within a relatively short time. Since machine learning (ML) has become a promising methodology for solving such complex decision-making problems, all the above challenges provide motivation for adopting ML methods to securely allocate resources in MCC amid dynamically changing resource demands in accordance with dedicated security policies.

The goal of this paper is to present our recent achievements in secure task allocation in MCC. We developed a new security-aware job flow model for MCC. We define MCC as a set of mobile devices remotely connected to virtual cloud services through wireless communication channels. For each mobile device and its connections to the cloud, security level parameters are specified and a list of ordered communication wireless protocols based on security level parameter values is defined. We also defined security parameters for the cloud services utilized by a given task (and connected to a given mobile device) and security demands for the data processed using such services. We developed a new security Filtration Algorithm, which generates the optimal secure configuration of communication protocols and services in order to meet the specified data confidentiality demands. The other criterion for generating the optimal mobile system architecture configuration for handling job flows is the minimization of utilization cost of cloud services and resources. To predict such costs, we used a machine learning algorithm.

Summarizing the above, the main contributions of this paper can be defined as follows:

- the development of the new security-aware job flow model in the MCC environment;
- the development of a novel secure configuration selection model for the MCC system, ensuring the lowest (optimal) predicted resource utilization by using machine learning algorithms that can be adapted online;
- comprehensive experimental analysis of the model developed and a comparison of system performance and security levels for different user preference priorities.

The rest of the paper is organized as follows. Section II presents a short state-of-the-art analysis of secure MCC. In Section III, we define the Security Filtration algorithm for ensuring service security during MCC optimization. Our method has been validated by a simple experimental analysis presented in Section IV. The paper ends in Section V with a summary of research results and plans for further developments in this area.

II. Related work

Ensuring an appropriate level of security in the MCC environment is an important and challenging problem that has already been discussed in the literature. Therefore, in this Section we present a short state-of-the-art analysis of secure MCC.

Some articles survey the existing frameworks that ensure security in the Mobile Cloud Computing (MCC) environment. In [6], the authors present a list of the most important features of MCC environment necessary for ensuring the security of MCC frameworks. Their list includes confidentiality and privacy, integrity, authorization, network security and secure data communication channels. The main weakness of that publication is that it does not include even a simple comparable experimental analysis of the methods surveyed. The authors only focus on the general definition and theoretical analysis of those methods.

A much more detailed analysis of data security in various MCC frameworks can be found in [7]. The authors of that article thoroughly analyze security aspects in the MCC environment, including infrastructural and architectural issues (for example virtualization security or insecure applications and interfaces) as well as privacy. They further investigate data security issues including cryptographic security schemes and compare different MCC frameworks in terms of the cryptographic techniques and tools used. The article presents a very detailed analysis of security aspects, especially cryptography, but there is no comparison of how security mechanisms work in practice in the different frameworks.

Another article that analyzes various security aspects of multiple MCC frameworks is [8]. The authors of that article present the criteria that a secure MCC framework should meet and compare selected MCC solutions in terms of these criteria. Similar studies are described in [9] where over fifty MCC frameworks were analyzed for security aspects. However, none of the solutions analyzed allows the user to easily define the security requirements which must be met by a specific service in MCC.

Another aspect in the context of MCC security is presented in [10]. That article includes a summary of multiple works in the context of various security and privacy issues related to multimedia big data in mobile and cloud computing. The growing importance of multimedia which are commonly used on mobile devices makes the security aspects of the use of this type of data in MCC increasingly important.

In [11], the authors analyze available secure user authentication schemes for mobile cloud computing services. They propose an improved model designed with bilinear pairing operations, countering the identified threats as posed to the Tsai scheme. They also compare security features of different ID-based cryptographic protocols. However, the proposed model lacks the ability for the user to define the security requirements which must be met by a specific service, which is possible in our solution. Also, in [12] the authors analyze security and privacy issues in three layers: the mobile terminal, the mobile

network and the mobile cloud, without comparing any existing frameworks.

The solution presented in [13] enables the optimization of application / service execution in the MCC environment. The decision on the location of application / service execution is made dynamically and takes into account memory usage, CPU utilization, energy consumption and execution time. The framework developed also has a security layer, which uses AES encryption to protect the data during offloading to the cloud. However, apart from introducing the AES algorithm and checking its impact on system performance, the authors do not take into account other security aspects such as the type of network connection, the level of trust in the cloud or the user's requirements concerning application / service security. Similar solutions involving the optimization of application / service execution, but using additional machine learning mechanisms are presented in other articles such as [14]–[16]. Additionally, in the article [17] the optimization method in MCC using machine learning was complemented by a code offloading mechanism.

In [18], the authors analyze security aspects, focusing on how to protect MCC resources from illegitimate access. For this purpose, they put forward a biometric authentication framework that enables pre-processing and algorithms for extracting features in order to match biometric traits. However, the authors of that article do not touch upon other important security aspects, including a secure network connection or the level of trust in the cloud.

The presented analysis of publications shows that there is a scarcity of research (including experiments) into the impact of security mechanisms on the efficiency of services in MCC. Therefore, we focused our work on this area of research.

III. Security-aware job allocation model in MCC

In this section developed system architecture, its formal model and algorithms are presented.

A. Architecture

The general concept of our new model of secure job allocation in MCC is presented in Fig. 1. In this model, we assume that MCC is defined as a set of mobile devices connected through wireless communication channels with remote servers and cloud resources. Boxes in the figure represent processes/main procedures, while cylinders represent data storage. The process receives a "Task to be Executed" as an input request. In order to ensure security while considering performance optimization, the request must be handled by the "Security Filtration" algorithm (see Algorithm 1). This is a filter that only selects those configurations that meet specific data confidentiality demands from among all connections and execution services. Subsequently, "Resource Consumption Prediction" is performed for these selected configurations using stored "Models". These predictions are used by the "Optimization" module to select those from among the connections and services available which best suit the user's execution preferences. With these data at its disposal, the "Execution and Data Collection" stage follows and all the data gathered are saved to the "Experience" database so that prediction models can be updated in the future using the "Learning Models" function. The interaction with the complete task execution process is implemented in Algorithm 2.

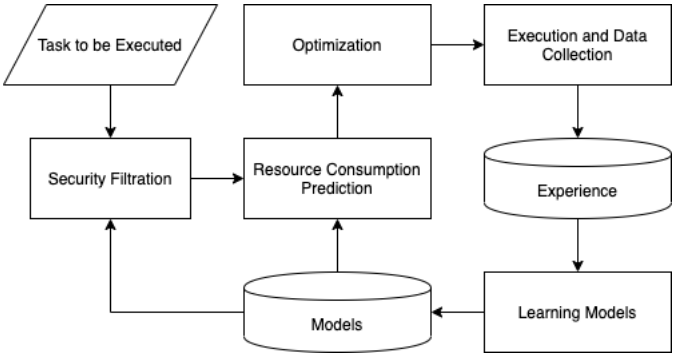

Fig. 1. The concept of the secure job allocation process

B. Model

The main procedures and modules in our secure job workflow model in MCC can be defined as follows. Let us denote by C an ordered list with n_c items which contains the available wireless communication protocols that can be used to connect a given mobile device to MCC:

$$C = (c_0, c_1, \ldots, c_{n_c}). \tag{1}$$

Such protocols are sorted based on their communication security level: c_0 denotes the protocol with the lowest security, while c_{n_c} is the highest security level communication protocol. A simple example of C is the following list of wireless communication protocols:

- c_0 – public Wi-Fi with no access restrictions;
- c_1 – public Wi-Fi with access restrictions;
- c_2 – cellular connection;
- c_3 – known Wi-Fi access point (e.g. defined by an SSID and a MAC address).

Mobile devices can use the VPN (Virtual Private Network) protocol to protect the connection against potential confidential data leaks. Therefore, we can define a feature parameter v with binary values from the set $V = \{0, 1\}$ where 0 means a connection without a VPN, and 1 – a connection using a VPN.

The job should be allocated to one of the services from the set U:

$$U = (u_0, u_1, \ldots, u_{n_u}). \tag{2}$$

For each service u_i, the following pair of parameters can be defined:

$$l_i = l(u_i) \in \mathbb{N} \tag{3}$$

$$p_i = p(u_i) \in \mathbb{N} \tag{4}$$

where l_i denotes service security level parameters (the higher, the more secure) and p_i – the service processing cost in MCC.

The following is a simple example of a service list:

- u_0 – public (free) cloud;
- u_1 – commercial (paid) cloud;
- u_2 – proprietary computation cluster;
- u_3 – local PC (connected to the same Wi-Fi access point);
- u_4 – mobile device.

Data confidentiality level for a job j is described by $d_j \in D = \{0, 1, 2, 3, n_d\}$, where 0 means that data is not confidential and the n_d value represents maximal confidentiality. It is assumed that we have at least one service providing sufficient security: $d_j \leq n_u$.

Using the features described above, we can define a set of all possible security policies:

$$S = U \times C. \tag{5}$$

To provide a secure execution environment for the job, we need a method to select $s \subset S$ which fulfils the security conditions expressed by the data confidentiality level $d_j \in D$ and VPN usage $v \in V$. This method is defined in Algorithm 1. Security policies have a property that more secure policies meet less strict needs, which can be expressed as:

$$\forall_{i \in \{1 \ldots n_d\}} \quad s_{d_{i-1}} \supseteq s_{d_i}. \tag{6}$$

If d_j value is the lowest possible, no security filtration takes place. If d_j value is the highest, it means that the task can only be executed on a mobile device. In the remaining range of d_j values, the connection types and execution services available are ranked starting with the least secure ones. VPN usage makes it possible to add one more connection type to the filtered pool.

To make a decision on task allocation, the mobile device context should be taken into account. It can be defined as:

$$ctx = (b, c) \in Ctx = B \times C, \tag{7}$$

where $b \in B = [0, 1]$ represents the current battery level and $c \in C$ represents the current Internet connection.

We would like to select a service and connection with low resource consumption. The problem is that this consumption is not known in advance. Machine learning algorithms can be applied to learn models for predicting the resources needed to process the task like execution time (M_t), battery usage (M_b) or money (M_p) for a given job (task) $j \in J$. This can be done using classification and regression algorithms. Details can be found in [16]. These predictions depend on the selected connection type, execution service and task features $f(j)$ (size of the data to be processed, data quality etc.). Therefore, we can assume that the predicted resource consumption is calculated as follows:

$$r(j) = M_r(u_j, c_j, f(j)), \tag{8}$$

where $r \in \{t, b, p\}$.

Job execution consumes a certain amount of resources. This consumption can be expressed by an element of the following set:

$$Res = \mathbb{R} \times [0, 100] \times \mathbb{N}. \tag{9}$$

As a result, the set of resources required to process a task j is represented by

$$res(j) = (t(j), b(j), p(j)) \in Res. \tag{10}$$

All information related to the efficiency of j execution can be collected in the following e_j tuple, containing data confidentiality d_j, execution service and task features $f(j)$ and required by task j resources $res(j)$:

$$e_j = (d_j, f(j), res(j)). \tag{11}$$

Input :
C - all possible connections;
U - all possible services;
v - if VPN can be used;
d_j - data confidentiality level ($d_j \leq n_u$);
Result:
C_j - filtered connections;
U_j - filtered services;

```
Function filterSecurityPolicies(
  C, U, v, d_j):
  if d_j == 0 then
    U_j = U;
    C_j = C;
  else
    if d_j > n_c + v then
      U_j = [u_{d_j}, u_{d_j}, ..., u_{n_u}];
      C_j = [];
    else
      m = d_j − v;
      U_j = [u_{d_j}, u_{d_j+1} ..., u_{n_u}];
      C_j = [c_m, c_{m+1}, ..., c_{n_c}];
    end
  end
  return (C_j, U_j);
```

Algorithm 1: Security Filtration algorithm

C. Service Execution Algorithm

Algorithm 2 represents the main system procedure that selects the best service and connection to execute a given job j_x and executes j_x there. In order for the algorithm to work, secure connection types C and execution services U must be determined so that only those which fulfill data confidentiality criteria d_j will be selected. Currently available connections C_r must be checked and passed to the function because connection types vary depending on mobile device location.

The first usage of Algorithm 2 triggers the training of the resource consumption model. In this process, predefined training data (training tasks) are used to simulate and test the behavior of the mobile device used. Such data is persisted in $ExperienceDB$ and used to train the model CM to predict resource consumption for a given task j_x through the selected

connection type c and calculations performed in service u. In every subsequent run of the algorithm, security is considered the main focus of interest with respect to the configuration preferences $pref$ set by user. This preprocessing is followed by running the task and persistence of the execution details (task feature $f(t)$, connection type and execution service m_x selected as well as consumed resources rc).

To represent the security level achieved for a job j executed using service u_j through connection c_j (if no connection was used and j was executed on a mobile device, $c_j = \emptyset$) we define:

$$SL_j = \begin{cases} u_j, & \text{for } c_j = \emptyset \\ \min(c_j + v, u_j), & \text{for } c_j \neq \emptyset \end{cases}. \quad (12)$$

With regard to equation (6), the security level SL_j achieved is always equal to, or higher than, the data confidentiality level d_j set by the user for j. This is because connections and services are by definition sorted by confidentiality level and Algorithm 1 allows c_j and u_j to be selected from subsets U_j and C_j starting with u_{d_j} and $c_{d_j - v}$ respectively (lines 12–13).

IV. Evaluation

The new secure job workflow model developed for MCC has been validated through simple experimental analysis. The main purpose of the experiments performed was to investigate the impact of security mechanisms on the efficiency of services in MCC. For these experiments, a scenario was selected in which an OCR (optical character recognition) service runs in MCC and can be executed on a mobile device, in the cloud or on a local PC. An OCR service based on the "tesseract OCR" open-source engine was deployed to the cloud, the local PC and as a mobile app for the Android OS. Graphics files for testing the OCR service were prepared as well. These graphics files can be described using the following feature set f: *file_size*, *image_resolution*, *difficulty*. The difficulty metric was proposed because the other characteristics were not correlated with OCR execution time. This metric is defined as the sum of numbers of changes between black and the background color for all rows of the scanned image. This makes it possible to estimate the number of characters in the scanned image. In experiments, this metric appeared to be a good approximation of the difficulty the OCR engine encountered in recognizing all characters, and a linear regression ML algorithm was able to predict resource consumption very well. File resolutions and the difficulty metric are shown in Table I. Using that data, the services' performance was measured depending on the connection type C and execution service U selected.

The test environment consisted of a mobile device (OnePlus 3T running Android 9 with a 4-core 2.35GHz CPU and 6 GB RAM) and of a local device (a 2017 13" MacBook Pro running macOS 11.0.1 with a 2-core 2.3 GHz CPU and 16 GB RAM). Both these devices were connected to the same local area network. The Amazon EC2 service (a t2.micro instance running Ubuntu 20.04 with 1 vCPU and 1 GB RAM) was used as the computing cloud. During the experiments, Wi-Fi 802.11ac and LTE Advanced connectivity were used.

Input :
- C - all possible connections;
- U - all possible services;
- v - if VPN can be used;
- d_j - data confidentiality level ($d_j \leq n_u$);
- j_x - task chosen by user to execute;
- C_r - currently available connections;
- CM - Context Model;
- $ExperienceDB$ - database with task execution contexts and consumed resources (training data for CM);
- $pref$ - user's priority preference (security/execution time etc.)

Result:
- $result$ - output of executed task j_x;

```
1  Function executeTaskUsingModule(
2    C, U, v, d_j, j_x, C_r, CM, ctx_a, pref):
3    if CM = ∅ then
4      if ExperienceDB = ∅ then
5        J_a = predefined tasks prepared to test
           demand on resources;
6        foreach (u, c, j) ∈ U × C × J_a do
7          addToDB(ExperienceDB,
             contextUsage(u, c, f(j)));
8        end
9      end
10     trainModel(CM, ExperienceDB);
11   end
12   if CM should be updated then
13     updateModel(CM, ExperienceDB);
14   end
15   s = filterSecurityPolicies(C, U, v, d_j);
16   r_x = predictResourcesConsumption(CM, f(j_x), C,
       U);
17   m_x = chooseOptimalService(r_x, s, c, f(j_x), pref,
       τ);
18   Execute task j_x at m_x and collect output as result
       and resources consumed rc ;
19   addToDB(ExperienceDB, j_x, m_x, rc);
20   return result;
```

Algorithm 2: Algorithm for service execution selection based on security and mobile device context

To simulate a real-life scenario in which a mobile device user executes a similar OCR job several times a day, a set of tasks (jobs) $J = (j_0, j_1, \ldots, j_n)$ was created. We assume that these executions demand various security settings and require appropriate allocation optimization. Every simulated day consisted of 10 tasks chosen randomly from J and was repeated for 30 consecutive days with randomly available connections $C_r \in C$, randomized data confidentiality levels $d \in D$ and also randomized VPN connection availability $v \in V$. In total, 300 tasks were prepared and used in the

TABLE I
PARAMETERS OF THE FILES USED IN EXPERIMENTS

	Resolution	Difficulty metric
1.	540x40	3183
2.	580x680	8463
3.	600x120	8493
4.	720x960	83883
5.	960x480	33574
6.	960x1280	98774
7.	1200x1200	17473
8.	1600x1200	22769

experiment.

To measure the influence of the security module on task (job) performance and the security level SL_j achieved, the following three user priority preference configurations representing different priorities were developed:

- P_S: Time and Security – security is the top priority, execution time is optimized if possible;
- P_T: Time – only execution time is optimized;
- P_R: Random Decision – the decision is made randomly.

The results demonstrate that the use of the security module increases service execution time. As shown on Figure 2, the P_S configuration which prioritizes the security aspect is approximately two times slower than the P_T configuration which only optimizes execution time, with the P_R configuration between the other two.

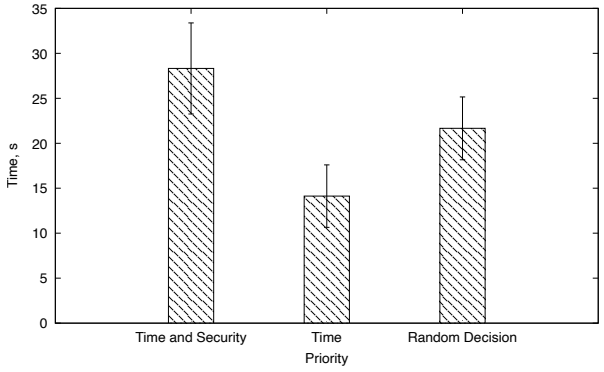

Fig. 2. Configuration impact on execution time

Figure 3 shows that in the P_S configuration where security is considered crucial, the average security level achieved reaches around 3.25, which means that for some jobs the maximum security level (4) was achieved.

Another conclusion from the experiments conducted is that in the P_T configuration (optimizing execution time) the security level drops by around 12% compared to P_R (random decision) and by around 36% compared to P_S.

Initially, image resolution was chosen as one of the meaningful features describing the task (element of $f(j)$). However, it appeared that this had no strict correlation with the additional time required (Figure 4). Therefore, a difficulty metric was added to $f(j)$. Figure 5 shows that with an increase in task difficulty (the difficulty metric), computation time also increases.

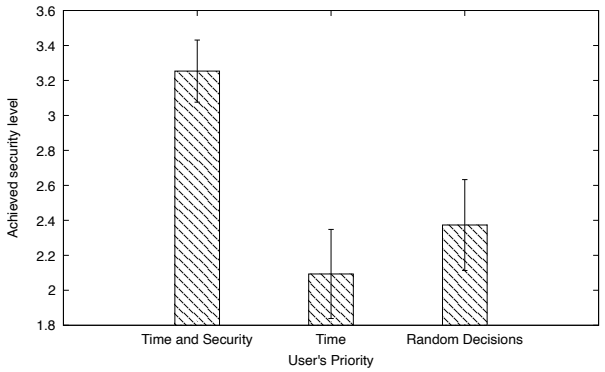

Fig. 3. Configuration and mean achieved security level

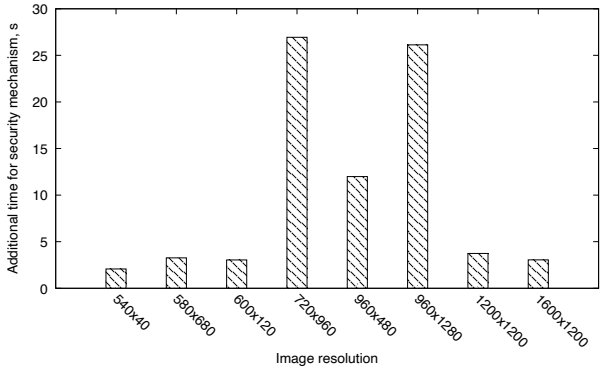

Fig. 4. Additional execution time due to the use of security mechanisms (in the context of the image resolution)

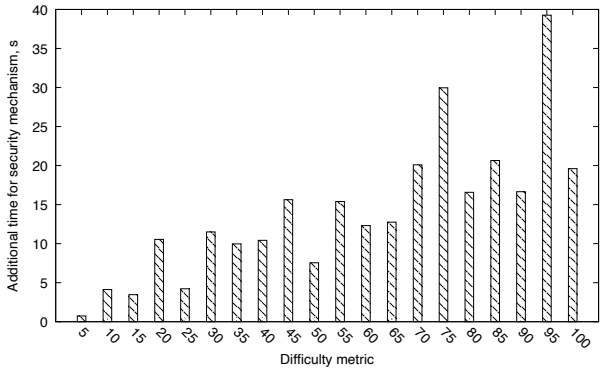

Fig. 5. Additional execution time due to the use of security mechanisms (in the context of the difficulty metric)

V. CONCLUSIONS

In this paper, we have proposed a novel security-aware job flow model in mobile computational clouds. This model makes it possible to take into account security parameters of the services and connections used and of the data processed. The proposed algorithm filters out configurations that are not

sufficiently secure and from the remaining ones, it selects one with the lowest predicted resource consumption. The information collected during job execution makes it possible to adapt the models used for prediction locally on the mobile device. This improves solution scalability and privacy because there is no need to send additional data related to user behavior. Experimental results demonstrate that our system works well and improves the security level of computations significantly compared to a configuration in which computation time is the highest priority.

Further, we plan to research the impact of our security mechanism on mobile device power consumption. We also intend to conduct additional experiments, especially taking into account service execution costs.

ACKNOWLEDGMENT

The research presented in this paper was supported by funds from the Polish Ministry of Science and Higher Education allocated to the AGH University of Science and Technology. Joanna Kolodziej's work was supported in part by the European Commission under Grant Agreement no. 833456.

REFERENCES

[1] J. Carolan, S. Gaede, J. Baty, G. Brunette, A. Licht, J. Remmell, L. Tucker, and J. Weise, "Introduction to cloud computing architecture," *White Paper, 1st edn. Sun Micro Systems Inc*, 2009.

[2] M. Satyanarayanan, "Mobile computing: the next decade," in *Proceedings of the 1st ACM workshop on mobile cloud computing & services: social networks and beyond*, 2010, pp. 1–6.

[3] P. Pranav and N. Rizvi, "Security in mobile cloud computing: A review," *International Journal of Computer Science and Information Technologies*, vol. 7, no. 1, pp. 34–39, 2016.

[4] B. R. Moyers, J. P. Dunning, R. C. Marchany, and J. G. Tront, "Effects of wi-fi and bluetooth battery exhaustion attacks on mobile devices," in *2010 43rd Hawaii International Conference on System Sciences*. IEEE, 2010, pp. 1–9.

[5] J. Oberheide and F. Jahanian, "When mobile is harder than fixed (and vice versa) demystifying security challenges in mobile environments," in *Proceedings of the Eleventh Workshop on Mobile Computing Systems & Applications*, 2010, pp. 43–48.

[6] P. Kulkarni, R. Khanai, and G. Bindagi, "Security frameworks for mobile cloud computing: A survey," in *2016 international conference on electrical, electronics, and optimization techniques (ICEEOT)*. IEEE, 2016, pp. 2507–2511.

[7] T. Bhatia and A. Verma, "Data security in mobile cloud computing paradigm: a survey, taxonomy and open research issues," *The Journal of Supercomputing*, vol. 73, no. 6, pp. 2558–2631, 2017.

[8] A. N. Khan, M. Mat Kiah, S. U. Khan, and S. A. Madani, "Towards secure mobile cloud computing: A survey," *Future Generation Computer Systems*, vol. 29, no. 5, pp. 1278 – 1299, 2013, special section: Hybrid Cloud Computing. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X12001598

[9] M. B. Mollah, M. A. K. Azad, and A. Vasilakos, "Security and privacy challenges in mobile cloud computing: Survey and way ahead," *Journal of Network and Computer Applications*, vol. 84, pp. 38 – 54, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1084804517300632

[10] B. B. Gupta, S. Yamaguchi, and D. P. Agrawal, "Advances in security and privacy of multimedia big data in mobile and cloud computing," *Multimedia Tools and Applications*, vol. 77, no. 7, pp. 9203–9208, 2018.

[11] A. Irshad, S. A. Chaudhry, M. Shafiq, M. Usman, M. Asif, and A. Ghani, "A provable and secure mobile user authentication scheme for mobile cloud computing services," *International Journal of Communication Systems*, vol. 32, no. 14, p. e3980, 2019.

[12] H. Suo, Z. Liu, J. Wan, and K. Zhou, "Security and privacy in mobile cloud computing," in *2013 9th International Wireless Communications and Mobile Computing Conference (IWCMC)*. IEEE, 2013, pp. 655–659.

[13] I. Elgendy, W. Zhang, C. Liu, and C. Hsu, "An efficient and secured framework for mobile cloud computing," *IEEE Transactions on Cloud Computing*, pp. 1–1, 2018.

[14] P. Nawrocki, B. Sniezynski, and J. Czyzewski, "Learning agent for a service-oriented context-aware recommender system in a heterogeneous environment," *Computing and Informatics*, vol. 35, no. 5, 2016.

[15] P. Nawrocki and B. Sniezynski, "Autonomous context-based service optimization in mobile cloud computing," *Journal of Grid computing*, vol. 15, no. 3, pp. 343–356, 2017.

[16] ——, "Adaptive service management in mobile cloud computing by means of supervised and reinforcement learning," *Journal of Network and Systems Management*, vol. 26, no. 1, pp. 1–22, 2018.

[17] P. Nawrocki, B. Sniezynski, and H. Slojewski, "Adaptable mobile cloud computing environment with code transfer based on machine learning," *Pervasive and Mobile Computing*, vol. 57, pp. 49–63, 2019.

[18] L. Ramavathu, M. Bairam, and S. Manchala, "A framework for secure mobile cloud computing," in *Proceedings of the First International Conference on Computational Intelligence and Informatics*. Springer, 2017, pp. 353–363.