Method of Increasing the Security of Smart Parking System

Tetiana Hovorushchenko[1], Olga Pavlova[1,*] and Mariia Kostiuk[2]

[1]*Department of Computer Engineering & Information Systems, Khmelnytskyi National University, Khmelnytskyi, Ukraine*
[2]*Department of Applied Mechanics, Khmelnytskyi National University, Khmelnytskyi, Ukraine*
E-mail: hovorushchenko@khmnu.edu.ua; pavlovao@khmnu.edu.ua; maria@khnu.km.ua
[*]*Corresponding Author*

Received 01 November 2022; Accepted 21 December 2022;
Publication 16 May 2023

Abstract

Currently, the urgent task is developing the methods and tools for increasing Smart Parking software system security. The purpose of this study is conducting analysis of requirements for Smart Parking System software security in order to find the bottlenecks and parts of the software that are most vulnerable to external threats and develop the methods and tools for increasing their security. The paper proposes the method of increasing Smart Parking software system security based on integrating the middleware in Smart Parking System software architecture. The proposed method takes into account all the criteria for Smart Parking System software security, i.e. parameters of safe access to the database, client program security, server security and API security and provides a complex solution for increasing the safety of Smart Parking software system. The proposed method differs from the known ones

Journal of Cyber Security and Mobility, Vol. 12_3, 297–314.
doi: 10.13052/jcsm2245-1439.123.3

in that it allows taking into account all the criteria for increasing the Smart Parking System software security in complex using security middleware.

Keywords: Smart Parking System security, external threats, client-server architecture, API, middleware.

1 Introduction

At the current stage of information and computer technologies development, special attention should be paid to security issues when developing software. This is especially crucial for critical software and cyber-physical systems software, as data loss or malfunctions can have unpredictable and sometimes critical consequences. Smart Parking System, proposed in [1] is based on client-server architecture and uses Convolutional Network image processing method. The images are taken from CCTV camera on a parking-lot and processed by the artificial neural network-based algorithm. Incorrect functioning of the algorithm or errors in the recognition of images by an artificial neural network can lead to an incorrect result providing. Since the client-server architecture is particularly vulnerable to various types of external threats, it is expedient to provide methods and algorithms for the protection and security of the smart parking system at the early stages of the life cycle, i.e. at the software architecture designing stage. It is crucial relevant since according to [5], the cost of error correction increases with each stage of the life cycle.

Therefore, the aim of this work is to analyze the factors that affect the security of smart parking system, the structure of which is presented in Figure 1, and develop the methods and algorithms for protection of this cyber-physical system.

2 State-of-the-Art

Multiple studies have been conducted to solve the problem of Smart Parking System security using different methods and tools. The main criteria that must be followed when developing a security system for smart parking are hardware security, hardware-software connection security, and software security from the standpoint of ensuring the Smart Parking software security system, the following criteria can be identified:

- checking parameters of safe access to the database;
- client program security;

Figure 1 Structure of smart parking system.

- server security;
- API security, if its use is provided by the architecture of the smart parking software system.

Let's conduct the state-of-the-art on known solutions and methods aimed at increasing the security. In paper [6] a secure smart parking system using blockchain technology is proposed, which uses cloaking technique to protect the drivers' location.

In [8] a solution is provided for preventing theft of vehicle from parking using RFID and GSM technology. Energy saving methods based on edge computing and IoT are proposed in [9].

In [10] an inclusive, long-term, effective, and well-performing smart autonomous vehicle parking (SAVP) system is presented. The authors introduce an integrated smart parking system that brings multiple parking service providers together under a unified platform aiming to provide one-stop parking information services to the commuters in a smart city.

The principal role of the research in paper [11] is to analyze smart parking solutions from a technical perspective, underlining the systems and sensors that are available, as documented in the literature. The review seeks to provide comprehensive insights into the building of smart parking solutions. A holistic survey of the current state of smart parking systems should incorporate the classification of such systems as big vehicular detection technologies.

The paper [12] proposes a parking management system that is geared towards business entities. The proposed system will focus on privacy for the

different entities that use the system. The paper aims to improve on already existing research on smart parking using blockchain. This paper proposes a parking management system that will be based on JPMorgan Quorum.

The paper [13] aims to design a secure and smart parking monitoring management system (SPMS) based on integration of WSN, RFID, and IoT.

Inspired by Blockchain and AI technology, the authors of [14] propose a Blockchain-enabled Secure Framework for Energy-Efficient Smart Parking in Sustainable City Environment.

The JMU Secure Smart Parking via the Cloud Environment is proposed in [15]. Using a Radio-frequency identification scanner, our system is able to count the number of vehicles entering and leaving each parking lot on campus.

An intelligent parking system in city based on the 5th Generation Mobile Communication Networks (5G) is proposed in [16]. 5G mobile communication technology has two important advantages of high transmission rate and low transmission latency, so it can better satisfy the rapid development of the Internet of Things (IoT).

This paper [19] presents a work-in-progress agenda that contributes to new business solutions and state-of-the-art research impacts. The authors reveal a multi-layered system of PSP-business model through interdisciplinary research blocks where original results are expected to be made at each layer.

We conducted the analysis of the recent studies [6–19] and highlighted the most frequently used methods for Smart Parking Security assurance. They are: blockchain technology using [6, 7, 10, 14], applying of biometric security mechanisms [7], radio frequency identification (RFID) and using of wireless sensor network (WSN) [8, 13], based on cloud environment [15, 19], 5G Technology [16], General Regression Neural Networks (GRNN) [23], fuzzy logic and uncertain data [24], multiple-valued logic [25]. But all these studies are focused on solving one or two criteria of Smart Parking System security and do not provide the solution of all the above mentioned criteria in complex. Consequently, with the purpose of increasing Smart Parking System security, it is necessary to conduct a software architecture analysis. The purpose of such analysis is the selection of parts of the software that are most vulnerable to external threats and providing the solution for Smart Parking System software security assurance in terms of following all criteria in complex.

Thus, *the urgent task* is the analysis of the requirements for Smart Parking System software security in order to find the parts of the software that

are most vulnerable to external threats and develop the methods and tools for increasing their security. Given the above, *the purpose of this study* is developing the methods of increasing the security of Smart Parking System, taking into account bottlenecks in software system and parts that are most vulnerable to external threats factors.

3 Client-Server Architecture and Possible Security Threat Factors

Since the Smart Parking System shown in Figure 1 consists of two parts – hardware part (cameras and all the and all physical devices necessary for functioning) and software part (client and server subsystems), it is necessary it is necessary to investigate possible factors that may affect the security of this system. If the hardware can simply be checked for reliability and performance, the software subsystem needs a deeper investigation. Taking into account the fact that the system has both server part and client part, an analysis of factors that affect the security of both parts of this cyber-physical system was carried out. They are: security misconfiguration, client-side injections (insecure authentification data, malwares), insufficient transport layer protection (MITM attacks), insecure data storage (database), device rooting/jailbreak, reverse engineering, sensitive data exposure (private data breaches), inadequate logging and monitoring. The results of the analysis are presented in a schematic form in Figure 2.

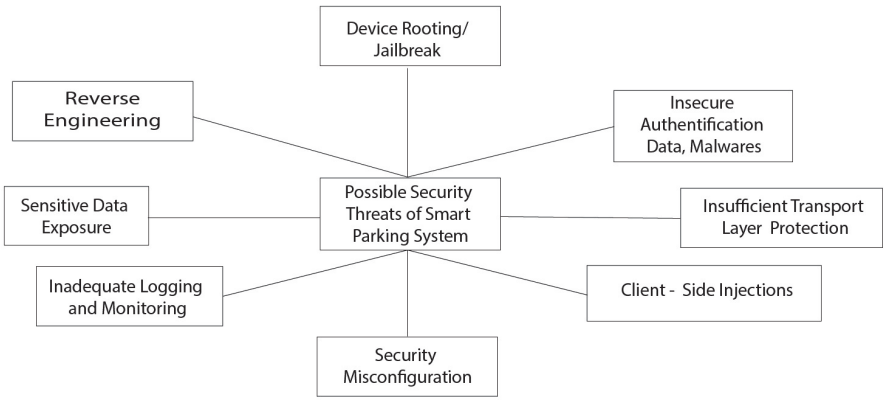

Figure 2 Possible security threats of smart parking system.

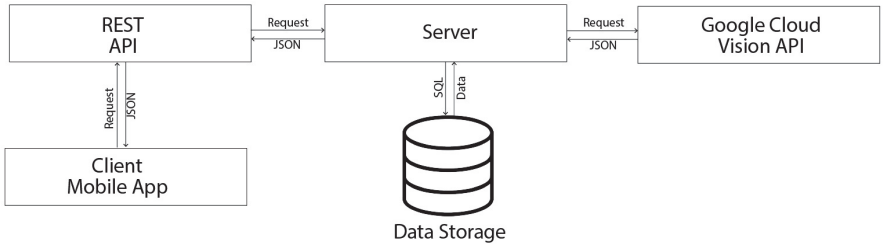

Figure 3 Architecture of smart parking software system.

Since the client part is supposed to be developed in the form of a cross-platform mobile application, which does not involve storing any private information, such as a personal phone number, login and password, it may seem that is less susceptible to attacks by hackers or information leaks. However the possibility of risks related to system security on the client's side cannot be completely excluded. The server part of the software, on the other hand, is very sensitive because it contains algorithms for recognizing car images using an artificial neural network. Unauthorized access to the database, program code or system files can lead to incorrect operation of algorithms and, as a result, to the provision of incorrect information about the occupancy or non-occupancy of a parking space to the client part. That is, incorrect operation of the entire system as a whole. Also, since the connection between the client and server parts is supposed to be implemented using an application programming interface (API), additional bottlenecks appear in the smart parking security system. The architecture of Smart Parking software system is represented in Figure 3.

3.1 Client-side Security Risks

For greater user convenience and faster access to the system, it was decided to develop the client part in the form of a cross-platform mobile application. Over the last decade, the industry of mobile application development has sufficiently increases but cybercrimes also have not stayed at their previous stage. All this caused the fact that it is not possible to upload the mobile application to Google Play Store or Apple App Store without checking the security metrics and being confident that the application will not be accused in information leaks or personal data fraud. But mobile application security is more than just protecting them from malicious software and external threats. First we need to define the main Open Web Application Security principles

and their main safety threats to be able to analyze the security measures and develop methods and tools to increase the level of their security [3].

- Improper platform usage

Not proper usage of smartphone functionality or unpredictable crashes while using security control settings. This includes privacy settings, permissions, incorrect use of Touch ID, FaceID, Keychain, etc.

- Data storage insecurity

A sufficient bottleneck that can be frequently found while cope with mobile application security issues is a lack of a secure data storage system. Mobile developers usually rely on client device storage for some private and internal data. But if the hackers get the access to device or the device may be stolen or lost, this data can be accessed and used for some malicious purposes. As a result it leads to such cybercrimes as privacy policy violation and personal data theft for the purpose of its malicious use.

- Insecure Client-Server Communication

During mobile application development, data communication occurs using a client-server model. Therefore, when the data is transferred, it can be intercepted by attackers via Internet. Malicious agents can also intercept data during wire transfer. Transmitting data via unreliable communication channels leads to privacy policy violation, personal data theft, fraud and business reputation loss for the company.

- Risks of Insecure Authentification

Malicious agents or bots can obtain data during authentication and infiltrate user's account. This can lead to personal information leaks, theft of personal data, and unauthorized access to internal data of user's account.

- Insufficient Data Encryption

Malicious agents or advertising bots can have the access to data that was not encrypted or protected properly. It can result to unauthorized access to internal data of the application, data theft, users‘ personal information leak etc.

- Insecure Authorization Risks

Malicious agents can intercept data during the authorization process and use it for unauthorized access to the application. As a result it leads to personal information leaks and loss of business reputation of the company.

- Poor Quality of the Application Code

Poor code quality can lead to program unpredictable crashes or multiple errors occurance while using. Also it decreases application performance and can cause excessive memory usage or slow loading of graphic elements in user interface while operating.

- Risk of Code Forgery

Malicious agents, when obtaining access to the source code, can integrate advertising or malicious scripts into it or replace parts of the code, which can lead to the incorrect functioning of the program, loss of some functions or replacement of certain functionality to use the application for malicious purposes.

- Reversible engineering

Attackers may download a mobile application in order to redesign its functions. That is, the same program can work completely differently in different versions.

- Extraneous Functionality Risk

In this case attackers review the functions of the mobile application in order to find bottlenecks and implement third-party code.

According to statistics given in Figure 4, insecure data storage and insecure client-server communication are the most frequent reasons of mobile application security risks.

The dependence of smart parking system security can be displayed as a tuple of factors that affect the security of the client part:

$$Csec = <ef, re, ct, ccq, ia, ic, iac, icm, ids, ipu>, \tag{1}$$

where:

ef – Extraneous Functionality,
re – Reverse Engineering,
ct – Code Tampering,
ccq – Client Code Quality,
ia – Insecure Authorization,
ic – Insufficient Cryptology,
iac – Insecure Authentification,
icm – Insecure Communication,
ids – Insecure Data Storage,
ipu – Improper Platform Usage

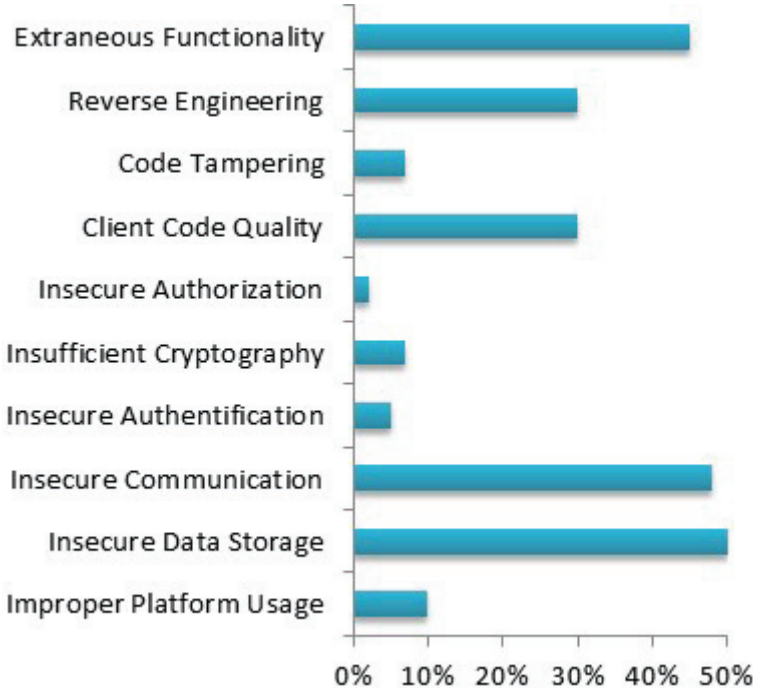

Figure 4 Frequency of factors manifestation that affect the security of mobile applications [3].

3.2 API Security Risks

Application Programming Interface (API) is a type of software that connects to the functionality of the application and saves developers time. Often with the help of an API, a functionality created by other developers or a frequently used functionality is connected or a client-server connection is performed. This helps developers save time and not develop from scratch features that are already in the public domain. When it comes to connecting several parts of software system together, it is really helpful [17]. However there are also security risks in using APIs. There are two primary reasons why security issues should be taken into account while using API.

- A simple way to obtain access to internal information of the application – via APIs stored data, including user's private information (login, password etc.) can be accessed for the purpose of unauthorized distribution or malicious activity.

- A simple way for attackers to bypass security measures, even if the firewall is enabled. Therefore, a well-thought-out security strategy should not be neglected.

There is a significant difference in security measures for traditional web-based applications and API-based web applications. This difference is in their architecture and how they are built. Previously, securing web applications only required protecting HTTP and HTTPS ports. Current applications, which use multiple APIs and different protocols, need be thought of comprehensive protection of all parts of the application, taking into account all its bottlenecks. This is especially important when the API expands its functionality, making security more difficult to manage. Also, when the API is replaced, previously developed security measures must be reviewed and reconfigured manually. The difference in the structure of API–based applications makes them sensitive to external threats [4]:

- Non-secure generation of API key

APIs are usually secured using a JSON Web Token or API key. This allows you to protect the API and, in case of detection of unusual or suspicious behavior, close access to the API keys.

- DDoS attacks

Protection against DDoS attacks is mainly built on the principle of deflecting requests from suspicious actors. This becomes more difficult since in API-based applications each traffic looks suspicious.

- Faulty server controls

Server is responsible for the communications between the application and user behind the mobile phone screen. The main reason for server vulnerability is that sometimes developers do not take proper security measures and protection of server connections seriously enough when working with APIs.

- Data Breaches

According to the studies worldwide data breaches and leaks usually occur when insufficient logging takes place.

- Not handling authorization

Unlike authentication, the authorization process in each application has its own logic and this may often be a bottleneck for attackers. If the authorization process is not sufficiently thought out and protected, hackers can log in to the system and access data using the iterative ID selection method [4].

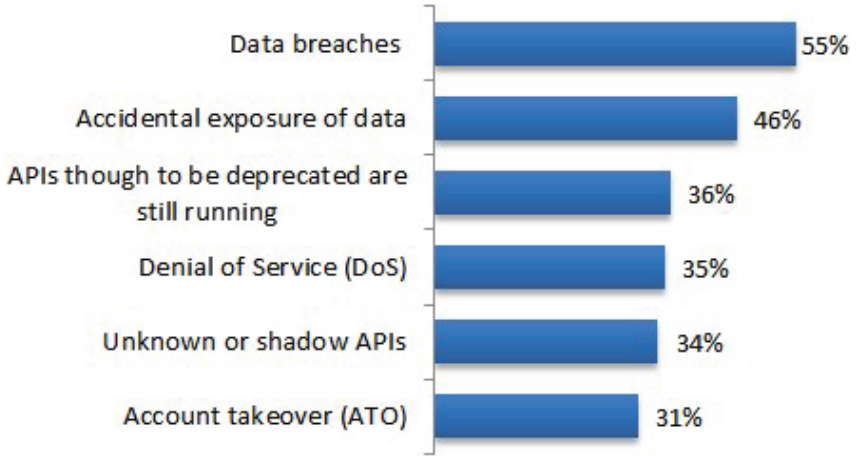

Figure 5 Frequency of factors manifestation that affect the security of application programming interface (API) [4].

According to statistics, the main and the most frequent factors are data breaches, accidental exposure of data, deprecated APIs, Denial of Service, unknown or shadow APIs and account takeover. The frequency of above mentioned factors manifestation that affects the security of Application Programming Interface (API) is shown in Figure 5.

The dependence of smart parking system security can be displayed as a tuple of factors that affect the security of Application Programming Interface:

$$APIsec = <dbr, aed, depapi, dos, unsapi, ato>, \tag{2}$$

where:

dbr – Data Breaches,
aed – Accidental Exposure of Data,
$depapi$ – API thought to be deprecated are still running,
dos – Denial of Service,
$unsapi$ – Unknown or Shadow API,
ato – Account Takeover.

4 Results and Discussion

Considering the factors that affect individual parts of the smart parking software system, it was decided to take into account ones that most often

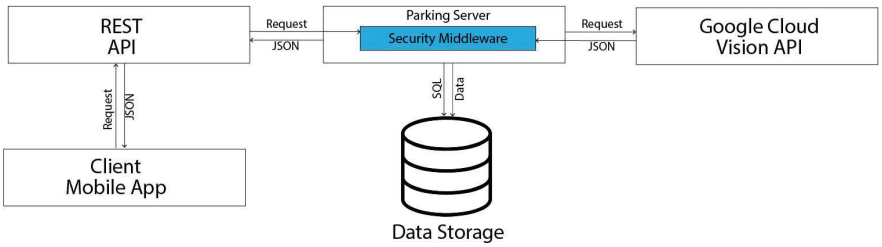

Figure 6 Smart parking software architecture using security middleware.

affect the security of the system and involve its different parts from different angles (for example, data access, client-server communication, bottlenecks when using the API) and arrive at a solution that will help take them into account in the complex.

The proposed solution is middleware for additional verification of requests from the client to the server. It is an efficient tool for performing operations or calculations inside a "request-response" connection in a client-server model of interaction. It should be used when it is necessary to perform a certain operation or check the reliability of the request not directly on a server for security reasons. That is, the middleware provides additional server protection from suspicious or malicious requests by intercepting and checking them. And only if the request is safe, it is sent to the server for further processing.

Therefore, we have improved the architecture of Smart Parking Software System given in Figure 2 by adding a security middleware to the server section of the software. The proposed architecture of the software system for smart parking, including middleware, is presented in Figure 6. With such architecture, it is much easier to determine whether the request was really sent from our client application and check whether it is not malicious and does not contain suspicious code. Also, this middleware will reduce the application's operating time, in case the request is not appropriate, because its result is already known, since the call to the Google Cloud API is not instantaneous.

Let's consider examples of the operation of the proposed smart parking software architecture using Security Middleware. For better understanding and clarity, let us consider the algorithm proposed in Figure 7.

According to Figure 7 the request is sent to the server from the mobile client application. However, the request may not always be secure. To check this, we verify the request using an integrated middleware. This will allow us to verify that the request really came from our client application, and not

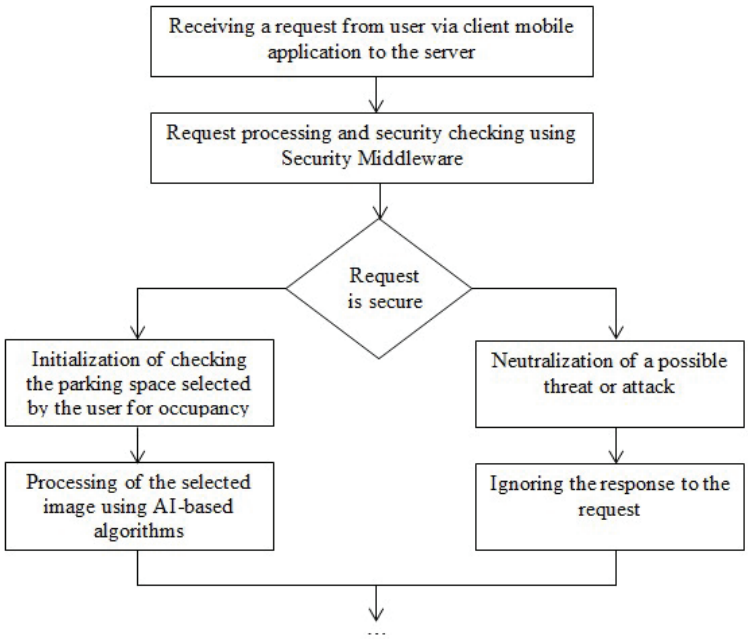

Figure 7 Algorithm for verifying the security of the request using security middleware.

from some third-party resource, and that the request is safe. Also, the use of middleware will reduce the application's operating time, if the request is not appropriate or its result is already known. Since the call to the Google Cloud Vision API is not instantaneous, we may use this time for verifying the security of the obtained request. If the request is secure, it is sent for further processing, namely checking the parking space selected by the user for occupancy. If the request is identified as potentially malicious or extraneous, it is neutralized in the first case, and the response from the server to such a request is ignored.

5 Conclusions

Currently, the actual task is developing the methods and means for increasing Smart Parking software system security.

The purpose of this study is conducting analysis of requirements for Smart Parking System software security in order to find the parts of the software that are most vulnerable to external threats and develop the methods and tools for increasing their security.

The paper proposes the method of increasing Smart Parking software system security based on integrating the middleware in Smart Parking System software architecture. The proposed method takes into account all the criteria for Smart Parking System software security, i.e. parameters of safe access to the database, client program security, server security and API security and provides a complex solution for increasing the security of Smart Parking software system. While using security middleware, it is much easier to determine whether the request was really sent from the native Smart Parking client application and check whether it is not malicious and does not contain suspicious code. In addition, security middleware will reduce the application's operating time, in case the request is not appropriate or malicious.

The main directions for authors' further research are: development and realization of the security middleware; integration of the developed middleware into server part of Smart Parking System; conducting the experiments and providing numerical evaluation of its efficiency.

References

[1] P. Radiuk, O. Pavlova, H. El Bouhissi, V. Avsiyevych, V. Kovalenko. Convolutional Neural Network for Parking Slots Detection. *CEUR Workshop Proceedings*, 2022, 3156, pp. 284–293.

[2] T. Hovorushchenko, A. Boyarchuk, O. Pavlova, K. Bobrovnikova. Agent-Oriented Information Technology for Assessing the Initial Stages of the Software Life Cycle. *ICTERI Workshops*, 2019. pp. 617–632.

[3] Understanding OWASP Mobile Top 10 Risks with Real-world Cases. URL: https://appinventiv.com/blog/owasp-mobile-top-10-real-world-cases/amp (last accessed October 21, 2022).

[4] The top API security risks and how to mitigate them. URL: https://appinventiv.com/blog/how-to-mitigate-api-security-risks/ (last accessed October 21, 2022).

[5] I. Lopatto, T. Hovorushchenko, P. Popov, O. Pavlova. Intelligent Multi-Agent System for Improving the Quality of Software by Taking into Account the Information of the Subject Area at All Stages of its Development. *Proceedings of the 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, IDAACS 2021, 2021, 1, pp. 548–551.

[6] W. A. Amiri, M. Baza, K. Banawan, M. Mahmoud, W. Alasmary and K. Akkaya. Towards Secure Smart Parking System Using Blockchain

Technology, 2020 *IEEE 17th Annual Consumer Communications & Networking Conference* (CCNC), 2020, pp. 1–2, doi: 10.1109/CCNC 46108.2020.9045674.

[7] A. Waheed and P.V. Krishna. Comparing Biometric and Blockchain Security Mechanisms in Smart Parking System, 2020 *International Conference on Inventive Computation Technologies* (ICICT), 2020, pp. 634–638, doi: 10.1109/ICICT48043.2020.9112483.

[8] L. Kumar, M. H. Khan and M. S. Umar. Smart parking system using RFID and GSM technology. *2017 International Conference on Multimedia, Signal Processing and Communication Technologies* (IMPACT), 2017, pp. 180–184, doi: 10.1109/MSPCT.2017.8364000.

[9] CP. Lee, FTJ. Leng, RAA. Habeeb, MAA. Amanullah, M. Rehman. Edge computing-enabled secure and energy-efficient smart parking: A review, Microprocessors and Microsystems, Volume 93, 2022.

[10] S. Ahmed, Soaibuzzaman, M. S. Rahman and M. S. Rahaman. A Blockchain-Based Architecture for Integrated Smart Parking Systems. *2019 IEEE International Conference on Pervasive Computing and Communications Workshops* (PerCom Workshops), 2019, pp. 177–182, doi: 10.1109/PERCOMW.2019.8730772.

[11] C. Biyik, Z. Allam, G. Pieri, G. Moroni, M. O'Fraifer, M. O'Connell, S. Olariu, M. Khalid. Smart Parking Systems: Reviewing the Literature, Architecture and Ways Forward. Smart Cities. 2021; 4(2), pp. 623–642.

[12] G.B. Imbugwa, M. Mazzara. (2021). Towards a Secure Smart Parking Solution for Business Entities. *Advanced Information Networking and Applications*. AINA 2021. Lecture Notes in Networks and Systems, vol. 227. Springer, pp. 469–478.

[13] O. Abdulkader, A. M. Bamhdi, V. Thayananthan, K. Jambi and M. Alrasheedi. A novel and secure smart parking management system (SPMS) based on integration of WSN, RFID, and IoT, *2018 15th Learning and Technology Conference* (L&T), 2018, pp. 102–106, doi: 10.1109/LT.2018.8368492.

[14] SK. Singh, Y. Pan, J. Hyuk Park. Blockchain-enabled Secure Framework for Energy-Efficient Smart Parking in Sustainable City Environment, Sustainable Cities and Society, Volume 76, 2022.

[15] M. Garcia, P. Rose, R. Sung and S. El-Tawab. Secure Smart Parking at James Madison University via the Cloud Environment (SPACE). 2016 *IEEE Systems and Information Engineering Design Symposium* (SIEDS), 2016, pp. 271–276, doi: 10.1109/SIEDS.2016.7489313.

[16] A. Anwar, Ijaz-ul-Haq, N. Saeed and P. Saadati. Smart Parking: Novel Framework of Secure Smart Parking Solution using 5G Technology. *2021 IEEE International Smart Cities Conference* (ISC2), 2021, pp. 1–4, doi: 10.1109/ISC253183.2021.9562776.

[17] T. Hovorushchenko, O. Pavlova, V. Avsiyevych. Method of Assessing the Impact of External Factors on Geopositioning System Operation Using Android GPS API. *2021 International Scientific and Technical Conference on Computer Sciences and Information Technologies* (CSIT), 2021, 1, pp. 295–298.

[18] A. Waheed, P. V. Krishna, J. Gitanjali, B. Sadoun, M. Obaidat. Learning automata and reservation based secure smart parking system: Methodology and simulation analysis, Simulation Modelling Practice and Theory, Volume 106, 2021.

[19] Y. Atif, J. Ding, MA. Jeusfeld, Internet of Things Approach to Cloud-based Smart Car Parking, Procedia Computer Science, Volume 98, 2016.

[20] I.M. Hakim, M. Christover, A.M. Jaya Marindra. Implementation of an image processing based smart parking system using Haar-Cascade method. *2019 IEEE 9th Symposium on Computer Applications Industrial Electronics* (ISCAIE-2019). pp. 222–227. IEEE Inc., Penang, Malaysia, 27–28 April 2019. DOI: 10.1109/ISCAIE.2019.8743906.

[21] G. Manjula, G.G. Rajulu, Anand, J.T. Thirukrishna. Implementation of smart parking application using IoT and machine learning algorithms. Computer Networks and Inventive Communication Technologies. Springer Singapore, Singapore, 2022. pp. 247–257 doi: 10.1007/978-981-16-3728-5_18.

[22] D. Vakula, Y.K. Kolli. Low cost smart parking system for smart cities. *2017 International Conference on Intelligent Sustainable Systems*, 2017. DOI: 10.1109/ISS1.2017.8389415.

[23] R. Tkachenko, I. Izonin, I. Dronyuk, M. Logoyda, P. Tkachenko. Recovery of missing sensor data with grnn-based cascade scheme. International Journal of Sensors, Wireless Communications and Control, 2021, 11(5), pp. 531–541.

[24] Zaitseva E., Levashenko V., Construction of a reliability structure function based onuncertain data, IEEE Transactions on Reliability, vol. 65, no. 4, 2016, pp. 1710–1723.

[25] Zaitseva E., Levashenko V., Reliability analysis of multi-state system with application of multiple-valued logic, International Journal of Quality and Reliability Management, vol. 34, no. 6, 2017, pp. 862–878.

Biographies

Tetiana Hovorushchenko received the master's degree in Computer Engineering Department from Khmelnytskyi National University in 2017 and the doctor of sciences degree in Engineering Science in 2018. She is currently working as a Head of the Department of Computer Engineering and Information Systems, Faculty of Information Technology, Khmelnytskyi National University. Her research areas: quality and safety assurance of software, smart city technologies.

Olga Pavlova received the master's degree in Computer Engineering and System Programming Department from Khmelnytskyi National University in 2017 and the philosophy of doctorate degree in Computer Science from Khmelnytskyi National University in 2021. She is currently working as a Senior Lecturer at the Department of Computer Engineering and Information Systems, Faculty of Information Technology, Khmelnytskyi National University. Her research areas include quality and safety assurance of software, smart city technologies, machine learning and augmented reality.

Mariia Kostiuk received the master's degree in Applied Mechanics and Systems of Computer-Aided Design of Processes from Khmelnytskyi National University in 2019 and currently is a postgraduate student in Applied Mechanics at Khmelnytskyi National University. She is currently working as a junior researcher at the research department of the Khmelnytskyi National University. Her research areas include industrial IoT, application programming interfaces for hardware and computer-integrated technologies.