Analysis of Implementation Results of the Distributed Access Control System

Maksym Vladymyrenko
Dept. of Inform. and Cyber Security
Borys Grinchenko Kyiv University
Kyiv, Ukraine
ORCID: 0000-0001-8852-7939

Volodymyr Buriachok
Dept. of Inform. and Cyber Security
Borys Grinchenko Kyiv University
Kyiv, Ukraine
ORCID: 0000-0002-4055-1494

V. Yu. Sokolov
Dept. of Inform. and Cyber Security
Borys Grinchenko Kyiv University
Kyiv, Ukraine
ORCID: 0000-0002-9349-7946

Artem Platonenko
Dept. of Inform. and Cyber Security
Borys Grinchenko Kyiv University
Kyiv, Ukraine
ORCID: 0000-0002-2962-5667

Abstract—**This paper attempts to create software and hardware complex that can work autonomously and designed to simplify the organization of scientific conferences. The goal of developing a complex is to give an opportunity to attendees of the conference to register to conference sections using radio frequency identification (RFID) tags and collect statistics of sections attendance in real time. The paper describes the development process of the complex, problems that encounters during the development process and ways to fix it.**

Keywords—radio frequency identification; distributed access control system; attendee tracking.

I. INTRODUCTION

Today's technology of RFID is used in a wide variety of industries. In particular, one of the most common ways of using technology is the systems of physical access control, in which spectrum measurements are used to identify users located in a particular area [1]. Such a system could be used to control the movement of attendees of a scientific conference that would provide the opportunity to get detailed real-time statistics about the activity of attendees. However, there is a shortage of open solutions that anyone could use. Therefore, we were offered the idea of developing an open solution for attendee's control. So let us look at how to implement such a system and check the validity of this idea in practice. The purpose of this research is to study the reliability of access control systems on the example of the automation of the accounting system for conducting scientific conferences.

The paper is structured as follows. Section II gives an overview of related works. In section III, has presented our vision of the work of designed control system and requirements to it. Section IV gives an overview of software and related hardware used during development. In section V, is described algorithm of work of the system. Section VI gives information about hardware assembly features and section VII presents results of the testing of our developed complex. This paper ends with section VIII were concluded current state of development of this system and described directions of future development.

II. RELATED WORKS

The results of several works in a similar direction were analyzed before starting the work [2–5].

Wang X. and Wang Y. in their work addressed the problem of building an access control system for employees in the office. Their article describes the main elements of such system and also described algorithm of system operation. Familiarization with this paper made it possible to look closer at the architecture of such a system. However, in this work, the practical part of the development and testing of the system is literally not affected, which does not fully evaluate the advantages and disadvantages of such a solution [6].

In turn, Z. Kotevski *et al* addresses the issue on the example of developing a system for monitoring attendance of students. The main usefulness of this publication is that the authors have considered in detail various ways of identifying attendees, as a result of which was provided a comparison of the characteristics of these ways. According to this conclusion, RFID tags are the best option for implementing an access control system. Based on this, we will further consider just such a method of identifying visitors [7].

III. PROBLEM STATEMENT

In order to understand the requirements for the developing software and hardware complex, is necessary to analyze the specifics of the conditions in which the system will work at the conference. The optimal model of interaction with the complex during registration should be approximately as follows: attendees are added to the database beforehand, or by hand right before the conference begin; during the registration, the attendee of the conference is issued a card, which contains his personal data, including his ID in the system; attendee's account is activated by the operator (the flag of activation in the database is changed).

At each section must be installed special controller for attendee's check in and when registered attendee visits some section, the sequence of steps should be as follows: controller displays greeting prompt on its screen;

2019 International Scientific-Practical Conference
Problems of Infocommunications. Science and Technology

PIC S&T'2019

participant scans the card on the controller when enters the section; attendee's data are read from the card to the controller's memory; controller sends an HTTP request to the server containing the identity data of the attendee and the controller; attendance flag is set in the database; controller displays the attendee's name on the screen for some seconds and then returns for displaying prompt.

Thus, in accordance with the description of the system's operation, the following requirements must be made to the controller presence of an RFID reader on the controller to get information from the attendee's cards; a display to inform a visitor about successful reading of the card in case of success or an error that has occurred in the opposite case; Wi-Fi module, because the wired network requires additional administrative and technical resources.

And for the server part of the complex, requirements should be as follows: presence of an RFID reader to record user data on a card; the server must have sufficient computing power to be able to handle a large number of requests at a time; software running on the server should have a convenient web interface to give users ability to interact with this system, including viewing the list of registered members.

In addition, considering the specifics of holding event such as scientific conferences, it is necessary to provide to the system mobility and autonomy. This is due to the fact that the conference can be held in different premises with different conditions and level of convenience for deployment of the network infrastructure. Therefore, the complex should be able to work both using the network infrastructure of the premises where it will be deployed, and in conditions where this is impossible. Therefore, the complex should be able to independently deploy the infrastructure for its work. In addition, if necessary, it should to be used as a workstation of an operator that will serve attendees. In accordance with the conditions listed above, we can form the following additional requirements to the server: minimum dimensions and weight of the server device; server must have USB and video interfaces to have ability to connect the monitor and keyboard to the device and use it both as server and workstation; presence of a Wi-Fi module and ability of the device to work in the AP mode to create a network for controllers.

The requirements listed above are a sufficient minimum for building a planned complex and will be used further as criteria for evaluating devices. Generalized scheme of a system's components is present in Fig. 1.
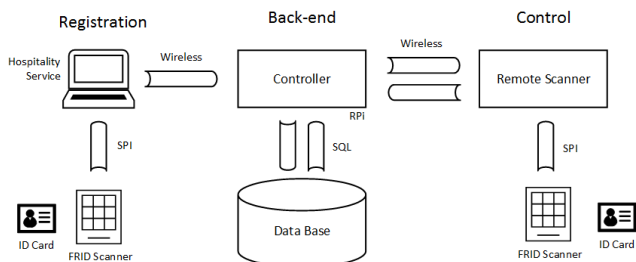

Figure 1. Generalized scheme of access control system

IV. DEVELOPMENT TOOLKIT

In accordance with the requirements listed above, the perfect device for implementing the server-side of developing complex is a single-board computer. The most popular one today is Raspberry Pi 3 Model B. This device is a full-fledged ARM-based computer that can work as a workstation as it has all the necessary connectors for connecting the monitor and the periphery or as a controller for embedded systems, since it has 40 GPIOs that can be used for connect additional equipment. The computer has support of such wireless technologies as Wi-Fi or Bluetooth, and has an Ethernet interface [8].

To implement the controller, several platforms with different characteristics were considered, namely:

- NodeMCU 1.0 development board based on ESP8266 MCU.

- Witty Cloud WiFi development board based on ESP8266 MCU.

- WEMOS Lolin32 development board based on ESP32 MCU that has integrated OLED display.

- Single-board computer Onion Omega 2 based on Mediatek MT7688AN CPU.

To select a platform for the role of controller, we must understand that there should be many of such devices in the system. And the controller's functionality is quite simple: it should read the data from the card, connect to the Wi-Fi network and send data to the server. All the modules examined have done this job equally well. Therefore, comparing the price for one controller is more characteristic than the technical capabilities of the microcontroller. Therefore, for this experiment it is decided to use ESP8266 microcontroller boards that provide all the necessary functionality, while having the lowest price among similar devices.

For this system was selected RFID modules operating at a frequency of 13.56 MHz and supporting the MIFARE standard [9], since it is one of the most commonly used standard for control systems. The most widely available on the chip market that works with this type of RFID tags is MFRC522, developed by NXP. On this chip are implementing several variants of ready-made modules, which can be used for prototyping. These modules have prepared pins for SPI interface through which it is possible to connect to the control unit. The one most commonly used by these modules was RFID-RC522 so we also will use it.

To show information from a controller, we selected OLED displays based on the SSD1306 controller. These displays are available in two sizes: 128×64 and 128×32 pixels. The benefits of the display of 128×64 include the ability to output twice the amount of information and the ability to purchase ready-made modules with pins for connection through an SPI or I²C. At the same time, the display size of 128×32 comes only in the form of modules with the I²C interface. This is because SPI needs more contacts than I²C (4 vs. 2 and extra contacts for power on both modules), so all the necessary contacts simply do not fit into the smaller module. Between two variants of displays, we chose the smaller one, since the screen is

planned to display only 1–2 lines of information, which eliminates the advantages of 128×64. At the same time, smaller display may be more convenient for the layout of the components of the controller.

Linux Raspbian was chosen for the server operating system because it is an official distribution of Raspberry Pi computers and is supported by developers. For this project is expedient to use one of two variants, namely *Raspbian with Desktop* or *Raspbian Lite*. The first one should be used if the minicomputer will be used not only as a server for the control system, but also as a workstation for the operator that will register new attendees. However, if the operators will work on individual computers, it is also possible to use *Raspbian Lite*, which needs a small amount of memory. In this case, the entire system can fit on a memory card of just 2 GB.

Requirements for programming language and the tools for writing the server part of the software are as follows:

- Availability of a ready-made library for working with RFID chip MFRC522.

- Convenient tools for creating web applications and REST APIs.

- Extensive standard library.

- Active community of developers that can contribute to this project in future.

These requirements are intended to simplify the development phase in the beginning by using a ready-made code base and facilitate the distribution of the project in future. After analyzing the accordance of the currently popular programming languages to the above requirements, was concluded that the main popular among programming languages such as Ruby, JavaScript, Python, Go, today can offer roughly the same set of benefits [10]. In the repositories of each of the languages you can find a library for working with RFID modules, have sufficiently developed standard tools and in each of these languages a lot of programs are written. Given this situation, was decided to use the Python language for this project. To implement the web interface and API were reviewed the main frameworks for developing web applications, such as *Django* and *Flask*. However, despite the fact that they are excellent tools for most web-related tasks, in this case, these frameworks are redundant and cumbersome, so the *Bottle* micro framework is chosen. In particular, for its minimalism (the source code of the framework consists of one file of only a few thousands of lines) and laconism, so you can learn this tool literally in a couple of hours, having read the official documentation [11].

Nearly any of the modern relational databases (PostgreSQL, MySQL, MariaDB, etc.) can be used to store and process data. However, since their functionality is redundant for the implementation of our experimental system, the choice was made to the side of *SQLite*, which occupies the minimum amount of memory on the disk and is easily administrated. In the future, it will be possible to transfer the project to another database without any difficulty, changing only the database's interface.

V. Algorithm of Interaction Between Elements of the System

The next stage of the experiment was the development of an algorithm that will be used by the system. According to the task, our system can be divided into two separate parts: controllers whose task is to read the card and transfer data from it to the server and the server itself, which will collect, process and store data from different controllers and provide the necessary information to the user. In turn, the server part can also be divided into several parts by the functions that each of them performs including: database—saves data about visitors, their movement; HTTP server—receives HTTP requests and passes them to the corresponding component; API—gives possibility to interact with the database and other components by sending HTTP requests to the server; web interface—component responsible for displaying information in a browser in some user-friendly interface; RFID tagging—component that is responsible for interacting with the RFID module connected to the server. More detailed scheme of a system's components is present in Fig. 2.
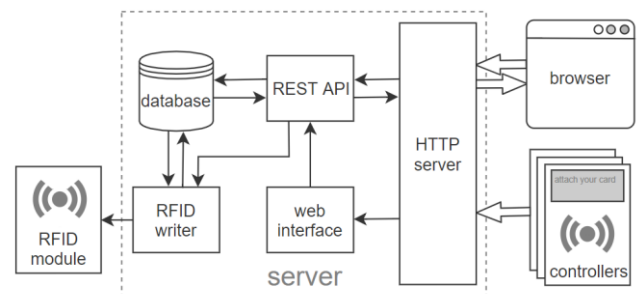

Figure 2. Scheme of connections between components of the system

To work with the database, the *sqlite3* driver from the standard Python library was chosen. For the web server work, including the implementation of API and web page generation, the *Bottle* micro-framework was chosen. And the entire functionality of the client part of the web interface was written without the use of external libraries, only with the use of pure JavaScript language. This was done to minimize the system's dependence on the external code base, so the system is completely autonomous and can correctly display the interface without connecting to the Internet.

Initially, it was planned to use the RFID module on the server for both writing and reading of cards. This should display the data from the card on the web interface or on an additional display. This functionality could be very useful, since it would allow to immediately check the correctness of the card record. However, when implementing such an algorithm, the system became less stable. This is due to the overall complexity of implementation of the multi-threaded control of RFID module, and with some instability of the used library *pi-rc522*, which can to work incorrectly under certain conditions. In response to the above facts, it was decided to abandon the described functional at this stage, not to install an additional display to the server and leave only the ability to write cards. Perhaps, with the further evolution of the project, this function will still be implemented.

2019 International Scientific-Practical Conference
Problems of Infocommunications. Science and Technology

PIC S&T'2019

VI. Assembly of the System Prototype

The next step was creating a fully functional model that would allow testing of the system's performance. First was the issue of combining Raspberry Pi with the RFID module in one case. To do this, the standard case for this minicomputer was used. The hole in the upper part of the case, located there for connecting the display, especially approached for mounting the RFID module in it. It turned out that the size of this hole fits greatly for the dimensions of the RFID-RC522 module, as seen in Fig. 3.

Figure 3. Raspberry Pi 3 with RFID-RC522 module as a server of our system

Two different modules were used to make controller models, and as a result, it is given the possibility to assemble two different models. The first model was made using a one-sided prototyping board. However, because of the low quality of this board, bad metallization on it, as well as the inequality of the board itself, the assembled module was unsuccessful. And after a short time of the modules work, the soldering joints began to drop and connection with periphery began to disappear. Therefore, the model was assembled again, but already with a prototype board of better quality (Fig. 4).

Figure 4. Model of controller with the NodeMCU development board

In general, this controller was performed well, but during its exploitation some constructive weakness was found: RFID module protrudes beyond the dimensions of the device; NodeMCU obscures the display, if you look at the controller not at a right angle due to high placement; the bottom of the board is clinging to the surface.

The last item in the list was immediately corrected by adding nicks from the screws. Others demanded a change in the layout of the elements on the board. As a result, a second version of the board was created, shown in Fig. 5. In order to allow the controller to be more steady, four

long nicks were mounted to the board. The only drawback in comparison with the first version of the controller is a removable UART module, so at the same time the module can either be connected to the UART module, or to the controller. Because of this, to download a new version of the firmware board you need to remove it from the controller each time. In addition, it makes impossible adding a controller's data exchange over UART in future.
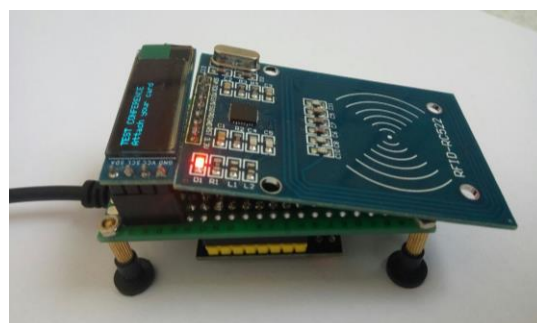

Figure 5. Model of controller with Witty Cloud WiFi board

VII. Implementation of the System

After assembling the system, a number of tests were carried out to assess its performance. Therefore, the first serious testing of the system was its application at the scientific conference "*Problems of Infocommunications. Science and Technology*" (PIC S&T'2018), held October 9–12, 2018 at the Kharkiv National University of Radio Electronics (Ukraine) [12]. Therefore, the results of this test revealed a number of programmatic issues that were not taken into account during development. In particular, the first prototype of the Web interface was not given enough attention, it was as minimalistic as possible, without adding CSS, which greatly complicated the work with it for the unprepared person. Therefore, after receiving feedback, a new intuitive interface was developed. In addition, a number of stability problems have been identified. So while using the system at a conference, the server work was unexpectedly stopped due to code errors. Therefore, after the testing, the server part of the system was re-written to improve it and avoid similar problems in the future. In addition, it is planned to cover the code with tests, for faster and faster detection of errors during further development.

The main problem at the design level of the system was the lack of alternative ways to add new attendees. The fact is that the first prototype of the web server was not clearly divided into "model" and "view" parts, and the interaction was followed by a mostly monolithic web interface. Therefore, each attendee was registering separately, which, at the scale of the people flow in the big event, slowed down the speed of the registration process. The full REST API was assigned to programmatically interact with the server, thus to expand its capabilities, including the massive addition of data.

For testing this statement, the Python script with name *send_data.py* was written. This script reads an attendees or sections data from a CSV file and sends it to the server using HTTP requests. It should be noted that instead of a CSV file, the source of data can be any of formats, starting with the XML table and ending with an external database, it will only be necessary to write the corresponding script.

By adding attendees using the API, the operator who registers attendee should only to find his data in the list and write an RFID card for itself, instead of registering the attendee from scratch by manually entering all his data.

To understand how much improvement has affected the time spent on a single attendee, an experiment was done in which a sample of 12 visitors and 3 measurements under different conditions was taken. In the first two cases, each attendee called his name, title and sections on which he would like to register to the operator. So time to add this data was measured. In the second case, the attendee received the card with his data from operator and time need to write card is also being measured in this case. In the second case, attendees' data were already added to the system in advance, and when he called his name, the operator should find this data in the list, write to card and give it to attendee. The time was measured from the moment the attendee appealed to the operator and before the transfer of the card.

It was interesting to find that the fields for entering attendees' data were not cleared automatically after adding, that allows to go to the input of new data immediately. This negatively affected the accuracy of measurements of the experiment. If in one case the addition of a data of the next attendee took more time because it was necessary to manually clear all the fields, and in the other one, addition process could be accelerated if some part of the data (the name or the same set of sections) coincided with the data of previous attendee, so in that case, these data did not need to be re-entered. This problem has been fixed and the auto-cleanup has been added to the form to attendee's registration, and the experiment was repeated. Then, from the three samples obtained from measurements, the arithmetic mean and the mean square error were taken. The obtained data are presented in the Table 1.

TABLE 1. MEASUREMENTS OF THE MEAN TIME NEED
TO REGISTER NEW ATTENDEE

Card-writing	Data addition method	Arithmetic mean, sec.	Mean square error, sec.
Yes	API	6.0	1.1
	Manual	22.0	5.2
No		14.8	2.5

The obtained data demonstrate that the procedure for adding personal data of the attendee into the database takes about 14 seconds, with a mean square deviation of this index of 2.5 s, indicating a significant variation of the time of the data entry. This is due to the heterogeneity of each user's data. In this case, this value is quite large, if you look at it, multiplying by several hundred visitors. However, it should be noted that in such scenario, adding an attendee to the database is only half of work, so also it is necessary to write down the card for the attendee. This procedure requires an average of 22 s, with a deviation from the average value of 5.2 s. This is due to the fact that in this case the factor of the correct work of the RFID module is added to the data heterogeneity factor. In practical experiments, it was found that some percentage of attempts to write data to a card fail due to various failures of the RFID module. Experimentally was reproduced one of the situations when the data was read and writes incorrectly. It turned out that applying the card

to the reader at an angle close to the direct (more than 60°), the RFID module notices this card within a radius of action and starts reading, but in this case reading or writing data will not succeed. Therefore, the number of unsuccessful attempts to use the card directly depends on how well it is attached to the reader. Finally, it should be noted that with the prior writing of the data in the database through API, serving a single attendee takes only 6 seconds. The average deviation is only 1.08 s, which is the best result among the three measurements, which in turn clearly demonstrates the success of the analysis of the problem parts of the system and improvement of it.

The memory of the card of MIFARE 1 K standard consists of 16 sectors with 4 blocks in each. Each block consists of 16 bytes. At the same time, the last block of each sector is allocated for storing system data, where are stored access keys and access flags of the sector. In addition, the first block is not writable and allocated to store information about the manufacturer. Thus, the maximum amount of useful data is 752 bytes.

For the experiment was written a Python script that generates an array for the required number of random hashes and sequentially writes them into the appropriate blocks of the card. Then the written data are read from the card and verified with the data from our generated array. For generating the hash, the MD5 algorithm was chosen, since it gives a 16-byte hash at the output, which corresponds to the block size. The measurements were carried out with a gradual increase in the number of sectors used for recording. The following number of blocks was chosen for using in each measurement: 5, 11, 17, 23, 29, 35, 41, and 47. For the experiment was taken three different cards which were alternated each other after each recording. The results of the measurements are listed below in the Table 2.

In the calculations presented in the table, sigma equal to 3 is used, the level of significance is 5%, the sample size is 30 elements.

TABLE 2. MEASUREMENTS OF THE WRITE
AND READ TIME INTERVALS

Block size, bytes	Action	Time, sec.	ΔTime, sec.	Confidence interval, sec.	
				Left	Right
80	write	0.135	0.035	–0.938	1.209
	read	0.100		–0.973	1.174
176	write	0.293	0.055	–0.780	1.367
	read	0.238		–0.835	1.312
272	write	0.460	0.347	–0.613	1.534
	read	0.627		–0.726	1.421
368	write	0.627	0.501	–0.447	1.700
	read	0.501		–0.572	1.575
464	write	0.787	0.591	–0.287	1.860
	read	0.591		–0.483	1.664
560	write	0.929	0.714	–0.145	2.003
	read	0.714		–0.360	1.787
656	write	1.136	0.838	0.063	2.210
	read	0.838		–0.236	1.911
752	write	1.298	0.998	0.224	2.371
	read	0.998		–0.075	2.072

2019 International Scientific-Practical Conference
Problems of Infocommunications. Science and Technology

PIC S&T'2019

In addition, in Fig. 6 you can visually see a comparison of the time required for writing and reading data from the card. It shows that this dependence is linear, exactly like the delta between the time for writing and reading a data block of the same size.
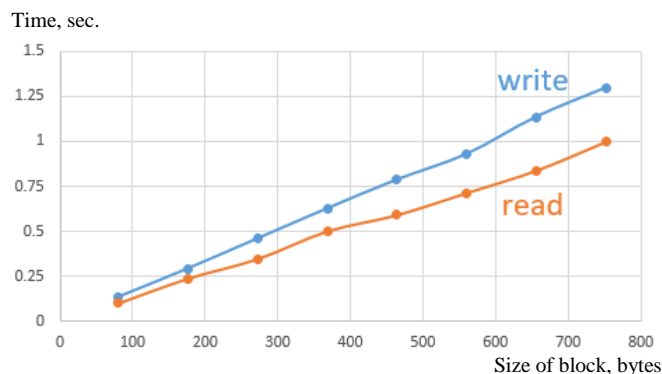
Time, sec.

Figure 6. The dependence of time needed for operation on the data size written to the card

During the experiment was observed that writing a large number of data blocks at a time significantly increased the number of errors. The dependence of the number of errors on the size of the written and readable data can be observed in Fig. 7. It follows from this that when working it is recommended to use the minimum possible amount of data read at a time.
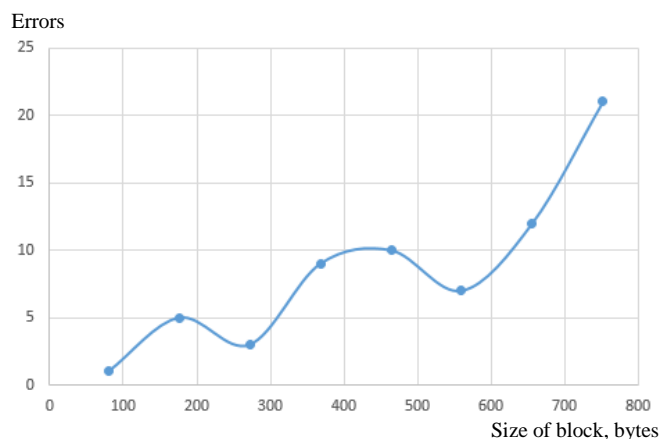
Errors

Figure 7. The dependence of the number of errors on the data size stored written to the card

VIII. CONCLUSIONS AND FUTURE WORK

Testing the complex in practice and its further improvement has demonstrated the validness of the proposed idea. Now we have a minimal working system that can fulfill the main task set for it: to track the movement of conference attendees between the sections of this conference.

In future, it is planned to add a number of improvements, which should improve the characteristics of this system and make it more safely and conveniently. In particular, it is considered to implement a secure system of controller identification in the system, which should protect the data from interception and substitution. In addition, it is planned to add a number of other improvements to information security, such as the separation of users' privileges and encryption of traffic transmitted between the server, controllers and users.

Such improvements in the future should facilitate the widespread use of this complex by anyone involved in the organization of conferences or similar events.

REFERENCES

[1] T. Sathishkumar, G. Prabhakara Rao, and P. Arumugam, "Verifying the authenticity of Employees entering nuclear complex," in *2016 IEEE First International Conference on Control, Measurement and Instrumentation*, Jan. 2016, pp. 146–150. DOI: https://doi.org/10.1109/CMI.2016.7413728

[2] S. Saha, "Evaluation of Network Parameters of a Sensor Node Deployment Strategy in Wireless Sensor Network for Hilly Terrains," *International Journal of Information Technology and Computer Science*, vol. 11, no. 1, pp. 31–38, Jan. 2019. DOI: https://doi.org/10.5815/ijitcs.2019.01.04

[3] V. Ferro, A. Luz, and A. Lucrecio, "Small long range UHF tag for metal applications," in *2013 IEEE International Conference on RFID-Technologies and Applications (RFID-TA)*, Sep. 2013. DOI: https://doi.org/10.1109/RFID-TA.2013.6694521

[4] M. Moshref, "Improved Anti-Collision Algorithm for Tag Identification in Future Internet of Things," *International Journal of Computer Network and Information Security (IJCNIS)*, vol. 9, no. 3, pp. 11–20, 2017. DOI: https://doi.org/10.5815/ijcnis.2017.03.02

[5] G. Kaura and G. Babbar, "A Novel RTS/CTS based Relative Time Synchronization Technique for RFID based Wireless Sensor Network," *International Journal of Wireless and Microwave Technologies*, vol. 5, no. 5, pp. 35–46, Sep. 2015. DOI: https://doi.org/10.5815/ijwmt.2015.05.04

[6] X. Wang and Y. Wang, "An Office Intelligent Access Control System based on RFID," in *2018 Chinese Control and Decision Conference (CCDC)*, Jun. 2018. DOI: https://doi.org/10.1109/CCDC.2018.8407206

[7] Z. Kotevski, N. Blazheska-Tabakovska, A. Bocevska, and T. Dimovski, "On the Technologies and Systems for Student Attendance Tracking," *International Journal of Information Technology and Computer Science*, vol. 10, no. 10, pp. 44–52, Oct. 2018. DOI: https://doi.org/10.5815/ijitcs.2018.10.06

[8] G. Loyse, *Raspberry-Pi Documentation*, Release 0.0, Nov 16, 2017, [Online]. Available: https://media.readthedocs.org/pdf/raspberry-pi-intro/latest/raspberry-pi-intro.pdf

[9] NXP Semiconductors, *MIFARE Classic EV1 1K—Mainstream contactless smart card IC for fast and easy solution development* Rev. 3.2, May 23, 2018, [Online]. Available: https://www.nxp.com/docs/en/data-sheet/MF1S50YYX_V1.pdf

[10] T. Crawford and T. Hussain, "A Comparison of Server Side Scripting Technologies," in *International Conference on Software Engineering Research and Practice (CSCE)*, 2017, pp. 69–76. ISBN: 1-60132-468-5

[11] M. Hellkamp, *Bottle Documentation*, Release 0.13, Jan 07, 2019, [Online] Available: https://bottlepy.org/docs/dev/bottle-docs.pdf

[12] V. Sokolov, A. Carlsson, and I. Kuzminykh, "Scheme for dynamic channel allocation with interference reduction in wireless sensor network," in *4th Int. Scientific-Practical Conf. Problems of Infocommunications. Science and Technology (PIC S&T)*, Oct. 2017, pp. 564–568. DOI: https://doi.org/10.1109/INFOCOMMST.2017.8246463

[13] IEEE. (2018). International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T) [Online]. Available: https://ieeexplore.ieee.org/xpl/conhome/8628958/proceeding