



Internet of Things for Industry and Human Applications Dependability and Security of Internet of Things

# Internet of Things for Industry and Human Applications

Dependability and Security  
of Internet of Things

**PRACTICUM**



**Ministry of Education and Science of Ukraine  
National Aerospace University “Kharkiv Aviation Institute”  
Ternopil National Economic University**

**V. V. Sklyar, V. V. Yatskiv, N. G. Yatskiv**

**Internet of Things for Industry and Human Applications**

# **Dependability and Security of Internet of Things**

**Practicum**

**Edited by V. S. Kharchenko and V. V. Sklyar**

**Project**

**ERASMUS+ ALIOT “Internet of Things: Emerging Curriculum  
for Industry and Human Applications”  
(573818-EPP-1-2016-1-UK-EPPKA2-CBHE-JP)**

2019

UDC 004.415/.416.052.056(076.5)=111  
C48

Reviewers:

Prof., Dr. Felicita Di Giandomenico, ISTI-CNR, Pisa, Italy

Prof., DrS. Volodymyr Opanasenko, State Prize Winner of Ukraine,  
Leading Researcher of the Microprocessor Engineering Department, V.M.  
Glushkov Institute of Cybernetics of NAS, Ukraine

**C48 Sklyar V.V., Yatskiv V.V., Yatskiv N.G. Dependability and Security Internet of Things: Practicum / Kharchenko V.S. and Sklyar V.V. (Eds.)** – Ministry of Education and Science of Ukraine, National Aerospace University “KhAI”, Ternopil National Economic University, 2019. – 98 p.

ISBN 978-617-7361-88-5

The materials of the practical part of the study course PC3 “Dependability and Security of IoT”, developed in the framework of the ERASMUS+ ALIOT project “Internet of Things: Emerging Curriculum for Industry and Human Applications” (573818-EPP-1-2016-1-UK-EPPKA2-CBHE-JP).

The structure of work on verification of residual knowledge in the discipline, the corresponding practical material, examples of tasks and criteria of evaluation are given. In the learning process, the theoretical aspects of dependability and security of IoT-based systems are presented. The basic concepts and approaches to development and implementation of dependable, safe and secure IoT systems, models and methods for dependability and security assurance and assessment of IoT-based systems are examined.

It is intended for engineers, developers and scientists engaged in the development and implementation of IoT-based systems, for postgraduate students of universities studying in areas of IoT, cybersecurity in networks, computer science, computer and software engineering, as well as for teachers of relevant courses.

Ref. – 33 items, figures – 14, tables – 7.

Approved by Academic Council of National Aerospace University “Kharkiv Aviation Institute” (record No 4, December 19, 2018).

ISBN 978-617-7361-88-5

© Sklyar V.V., Yatskiv V.V., Yatskiv N.G., Kharchenko V.S.

This work is subject to copyright. All rights are reserved by the authors, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms, or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar.

**Міністерство освіти і науки України  
Національний аерокосмічний університет  
ім. М. Є. Жуковського «Харківський Авіаційний Інститут»  
Тернопільський національний економічний університет**

**Скляр В.В., Яцків В.В., Яцків Н.Г.**

**Інтернет речей для  
індустріальних і гуманітарних застосунків**

# **Гарантоздатність та безпека систем Інтернету речей**

**Практикум**

**Редактори Харченко В.С. та Скляр В.В.**

**Проект ERASMUS+ ALIOT  
“Інтернет речей: нова освітня програма для потреб  
промисловості та суспільства”  
(573818-EPP-1-2016-1-UK-EPPKA2-SVNE-JP)**

2019

УДК 004.415/.416.052.056(076.5)=111  
С48

Рецензенти: Др. Фелісіта Ді Джандоменіко, ISTI-CNR, Піза, Італія

Д.т.н., проф. Володимир Опанасенко, Лауреат Державної премії  
України, провідний науковий співробітник відділу мікропроцесорної  
інженерії, Інститут кібернетики ім. В.М.Глушкова НАН України

**Скляр В.В., Яцків В.В., Яцків Н.Г.**

С48 **Гарантоздатність та безпека систем інтернету речей: Практикум** / За  
ред. Харченка В.С. та Скляра В.В. – МОН України, Національний аерокосмічний  
університет ім. М. С. Жуковського «ХАІ». – 98 с.

ISBN 978-617-7361-88-5

Викладено матеріали практичної частини курсу PC4 “Розробка та впровадження IoT систем”, підготовленого в рамках проекту ERASMUS+ ALIOT “ Internet of Things: Emerging Curriculum for Industry and Human Applications” (573818-EPP-1-2016-1-UK-EPPKA2-CBHE-JP).

Наведена структура робіт з перевірки знань з курсу, відповідний практичний матеріал, приклади виконання завдань та критерії оцінювання. В процесі навчання наводяться теоретичні аспекти розробки та впровадження надійних та безпечних IoT систем. Вивчаються основні концепції та підходи до розробки і імплементації IoT систем, моделі для надійності та безпеки IoT пристроїв та систем.

Призначено для інженерів, розробників та науковців, які займаються розробкою та впровадженням IoT систем, для аспірантів університетів, які навчаються за напрямами IoT, кібербезпеки в мережах, комп'ютерних наук, комп'ютерної та програмної інженерії, а також для викладачів відповідних курсів.

Бібл. – 106, рисунків – 48, таблиць – 20.

Затверджено Вченою радою Національного аерокосмічного університету «Харківський авіаційний інститут» (запис № 4, грудень 19, 2018).

УДК 004.415/.416.052.056(076.5)=111

ISBN 978-617-7361-88-5

© В.В. Скляр., В.В. Яцків, Н.Г. Яцків, В.С. Харченко

Ця робота захищена авторським правом. Всі права зарезервовані авторами, незалежно від того, чи стосується це всього матеріалу або його частини, зокрема права на переклади на інші мови, перевидання, повторне використання ілюстрацій, декламацію, трансляцію, відтворення на мікрофільмах або будь-яким іншим фізичним способом, а також передачу, зберігання та електронну адаптацію за допомогою комп'ютерного програмного забезпечення в будь-якому вигляді, або ж аналогічним або іншим відомим способом, або ж таким, який буде розроблений в майбутньому.

## ABBREVIATIONS

C&C – Command and Control  
CPU – Central Process Unit  
DC – Diagnostic Coverage  
EUC – Equipment Under Control  
GSN – Goal Structuring Notation  
FPGA – Field-Programmable Gate Array  
FMECA – Failure Mode, Effect and Criticality Analysis  
IEC – International Electrotechnical Commission  
IoT – Internet of Things  
NIST – National Institute of Standards and Technology  
OT – Operational Technology  
PCB – Printed Circuit Board  
PTC – Proof Test Coverage  
RBD – Reliability Block Diagram;  
RFID – Radio Frequency Identifier  
SAD – System Architecture Design  
SHA – Secure Hash Algorithm  
SFF – Safe Failure Fraction  
SRS – Safety Requirements Specification  
SSLC – Safety & Security Life Cycle  
SSMP – Safety and Security Management Plan  
TCP – Transmission Control Protocol  
UDP – User Datagram Protocol  
UML – Unified Modeling Language

## INTRODUCTION

The materials of the practical part of the study course PC3 “Dependability and Security of IoT”, developed in the framework of the ERASMUS+ ALIOT project “Internet of Things: Emerging Curriculum for Industry and Human Applications” (573818-EPP-1-2016-1-UK-EPPKA2-CBHE-JP)<sup>1</sup>.

The structure of work on verification of residual knowledge in the discipline, the corresponding practical material, examples of tasks and criteria of evaluation are given. In the learning process, the theoretical aspects of dependability and security of IoT-based systems are presented. The basic concepts and approaches to development and implementation of dependable, safe and secure IoT systems, models and methods for dependability and security assurance and assessment of IoT-based systems are examined.

The module PCM3.1 “Dependability and security models of IoT” includes 1 seminars and 2 laboratory works.

The seminar is dedicated to hardware-software products certified against functional safety requirements.

The first laboratory work is aimed on assessment of safety indicators.

The second laboratory work covers analysis of IoT threats and attacks scenarios.

The module PCM3.2 “Safety and security management of IoT” contains 1 seminar and 2 laboratory works.

The first laboratory work introduces development of safety and security management plan for IoT systems.

The second laboratory work covers development of IoT safety and security life cycle.

The seminar is dedicated to study of IoT verification and validation methods.

The module PCM3.3 “Assurance Case for IoT” contains 1 seminar and 2 laboratory works.

The first laboratory work introduces software tools for IoT

---

<sup>1</sup> *The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.*

assurance case development.

The second laboratory work provides safety and security techniques and measures for IoT systems.

The seminar is about green (energy efficient) assurance case for IoT.

The module PCM3.4 “Security of IoT Based Blockchain Technology” contains 1 seminar and 2 laboratory works.

The first laboratory work introduces generation and research of Hash Function for Blockchain Technology safety and security techniques and measures for IoT systems.

The seminar is about consensus algorithms in IoT systems,

The second laboratory work covers blockchain-based protection of video files integrity.

Practicum is prepared by Professor, DrS. Sklyar V.V. (Computer Systems, Networks and Cybersecurity Department, National Aerospace University “KhAI), DrS. Yatskiv V.V. (Cybersecurity Department, Ternopil National Economic University), and Associated Professor, Dr. Yatskiv N.G. (Cybersecurity Department, Ternopil National Economic University).

The authors are grateful to the reviewers, project colleagues, staff of the departments of academic universities, industrial partners for valuable information, methodological assistance and constructive suggestions that were made during the course program discussion and assistance materials.



## **PCM3.1. Dependability and security models of IoT**

**Prof., DrS. V. V. Sklyar (KhAI)**

### **Seminar 24.1**

#### **STUDY OF HARDWARE-SOFTWARE PRODUCTS CERTIFIED AGAINST FUNCTIONAL SAFETY REQUIREMENTS**

**The purpose and tasks of the seminar:** acquisition of knowledge and practical skills on the work with technical description of safety and security critical programmable components.

**Training tasks:**

- study of products data sheets;
- study of products safety assessment reports.

**Practical tasks:**

- conducting analysis of safety critical functions and parameters;
- conducting analysis of security critical functions and parameters.

**Preparation for the seminar**

Preparation for the seminar includes the following steps.

1) Obtaining (defining) the topic of the paper.

Themes of the paper may be formulated by students independently based on technical descriptions of the following products:

1. Vector Informatik GmbH MICROSAR SafeRTE.
2. SEC RTMSafety.
3. National Instruments 9350 C-Series Logic Module.
4. RadICS Controller.
5. Emerson Rosemount 3051 Pressure Transmitter.
6. General Electric RA Hydraulic Actuator.
7. Honeywell Safety Control Controller.

2) Search for information on the paper of the abstract and its preliminary analysis.

3) Development of the paper plan and presentation.

The plan of a typical paper includes:

- purpose and main tasks;

- analysis of the issue;
- the main part (the principle of the algorithm, modeling, research results, advantages and disadvantages of the algorithm, examples of application);
- conclusions;
- list of references;
- applications.

#### 4) Writing the paper.

The paper should be 15-20 pages of A4 format (font 14.1,5 in., Field 2 cm), including title page, content, main text, references, applications.

An obligatory appendix to the paper is presentation slides.

#### 5) Preparation of the presentation.

The presentation should be done in PowerPoint and corresponds to the paper plan (8-10 slides for 10 minute talk).

The presentation should include the following slides:

- title slide (university, department, discipline, topic of the report, author, date of presentation);
- content (structure) of the report;
- relevance of the considered issues, the purpose and tasks of the report;
- slides with the disclosure of the main content;
- conclusions;
- references.

The content of the slides should not contain parts of the text from the paper, but includes the keywords, figures, and formulas.

### **The presentation**

The presentation (report) is carried out at the seminar, it takes 15 minutes and includes the report (10 minutes) and discussion (5 minutes). Languages of presentations and papers are Ukrainian or English.

### **Evaluation**

Assessment of this work includes:

- the quality of the text of the paper (form and content),
- presentation quality (content and design);
- the quality of the report (content, logical structure, conclusions);
- distribution of time by sections;

– completeness and correctness of answers.

### **Questions**

1. What types of risks arise in IoT systems, and what is the nature of each of these risks?
2. How are dependability, safety and security interrelated?
3. What are the challenges in terms of safety and security those are created by the rapid increase in the number of devices connected to the Internet?
4. What are the groups of requirements for safety and security?
5. How do safety attributes constitute a common system with attributes of security and dependability?
6. What is the difference between reliability, dependability, availability and safety?
7. What is risk?
8. How can risk be assessed qualitatively and quantitatively?

### **References**

1. Rausand M. Reliability of safety-critical systems : theory and application. – John Wiley & Sons, Inc., Hoboken, New Jersey, USA, 2014.
2. Avižienis, A., Laprie, J.-C., Randell, B. and Landwehr, C. Basic Concepts and Taxonomy of Dependable and Secure Computing. IEEE Transactions on Dependable and Secure Computing (2004), 1(1): 11-33.
3. Leveson N. Engineering a Safer World: Systems Thinking Applied to Safety. – The MIT Press, 2011.

## Laboratory work 24.2

### ASSESSMENT OF SAFETY INDICATORS

**Goal and objectives:** The purpose of the laboratory work is to study and research dependability indicators of IoT systems.

Learning objectives:

- study the properties of safety indicators;
- study different functions and algorithms of self-diagnostics.

Practical tasks:

- develop Reliability Block Diagram (RBD);
- calculate safety indicators;
- implement Failure Mode, Effect and Criticality Analysis (FMECA).

#### Reliability Block Diagram

RBD is diagrammatic method demonstrated how components reliability contributes to failure modes of complex systems.

The typical Industrial IoT system includes (Fig. 2.1):

- power supply components;
- field equipment (sensors and actuators);
- programmable logic controllers, including input and output modules and control modules;
- network equipment, servers, and human-machine interface components.

Redundant architectures with “2-out-of-3” and “2-out-of-4” voting logic are also used in Industrial IoT systems important for safety and security.

In this laboratory work we will focus our research on the main control chain such as “sensor – input module – logic module – output module – actuator”. It should be taking into account during practical task performance and in the report.

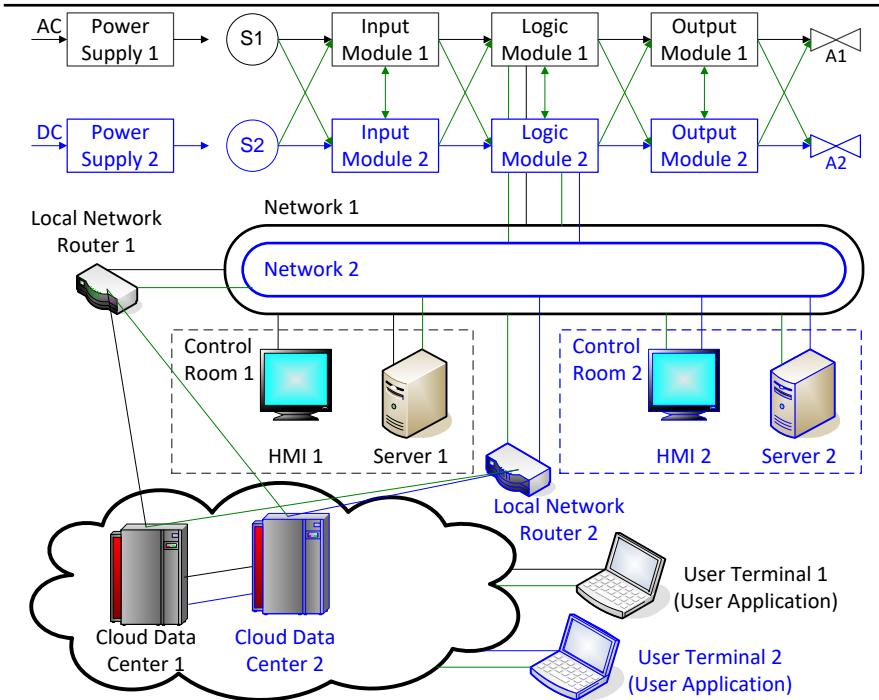


Fig. 2.1 – Redundant Industrial IoT system

### Self-diagnostics

Self-diagnostics of digital devices (input module, logic module, output module) can be described as on Fig. 2.2. Along with the main algorithms of digital control, in parallel, the system implements the processing of diagnostic data and watchdog functions. All these three processes are performed independently of each other, and independent clock sources, different chips, etc. can be used.

Watchdog monitors the simplest response (heart bit) from the chips that perform data processing, and when a problem is detected (the response is stopped), it turns off the power and puts the system into a safe state. In addition, the watchdog timer can monitor the power level and produce a similar shutdown command if there is a dangerous power deviation from the specified level. Safe state for safety systems, as a rule, consists in removing power from the output analogue and discrete outputs. If necessary, the safety system can supply power to the actuators, but then the output requires additional signal converters.

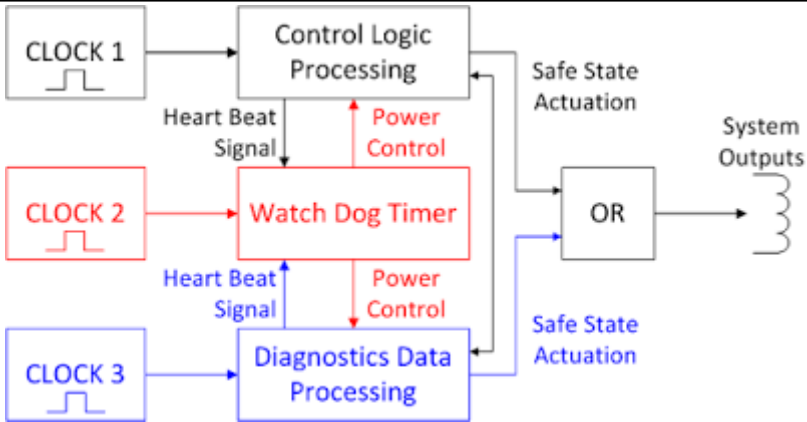


Fig. 2.2 – Self-diagnostics of IoT

### Dependability indicators

The basic concept of functional safety assessment is dividing a common failure rate  $\Lambda$  (let us begin with the exponential distribution with a constant failure rate) into dangerous and safe failures as well as into detected and undetected failures. This is a main difference of functional safety from reliability. From this point of view we have four failures sets:

- Safe Detected failures with a failure rate  $\lambda S_d$  – failures which put the equipment under control (EUC) to a safe state and are discovered by self-diagnostics;

- Safe Undetected failures with a failure rate  $\lambda S_u$  – failures which put the EUC to the a state and are not discovered by self-diagnostics;

- Dangerous Detected failures with a failure rate  $\lambda D_d$  – failures which put the EUC to a potentially dangerous state and are discovered by self-diagnostics;

- Dangerous Undetected failures with a failure rate  $\lambda D_u$  – failures which put the EUC to a potentially dangerous state and are not discovered by self-diagnostics.

The most important safety indicators are the following:

- Safe Failure Fraction (SFF) in accordance with IEC 61508 is  $SFF = (\lambda S + \lambda D_d) / \Lambda$ ;

- Diagnostic Coverage (DC) for dangerous failures in accordance with IEC 61508 is  $DC_D = \lambda D_d / \lambda D$ ;

- Proof Test Coverage (PTC) should be calculated from the total

failure rates for the using the formula  $PTC = 1 - \lambda_{DuaPT} / \lambda_{Du}$ , where  $\lambda_{DuaPT}$  is  $\lambda_{Du}$  after Proof Test.

### **Failure Mode, Effect and Criticality Analysis**

FMECA differs from other methods of dependability and safety analysis in that it puts together all the tasks of calculating safety indicators. The standard IEC 60812:2006 “Analysis techniques for system reliability – Procedure for failure mode and effects analysis (FMEA)” has been developed is to describe this method.

At the initial stages of the FMECA application, it is recommended to apply a hierarchical decomposition of the system, for example, using Reliability Block Diagrams (RBD).

Different levels of details may be applied during FMECA. Usually for safety systems, the analysis takes into account all electronic components, such as resistors, capacitors, diodes, etc.

FMECA is performed for the identified safety and security functions from the point of view of the software and hardware involved in the execution of the function. For these functions, the states of dangerous failures have to be defined and described. The results of the analysis are recorded in the form of FMECA tables.

### **Report**

The report should contain: title page with the name of the laboratory work; aim of the work; problem statement according to the task; the progress and results of the study in graphical form; analysis of the results and conclusions.

### **Tasks**

1. Choice you task variant (Table 2.1).
2. Develop functional diagram of the investigated system:
  - analysis of the structure and functions of the system;
  - division of the system into its parts and elements, based on the influence of element failures on system failures and the level of detail.
3. Develop RBD of the investigated system:
  - drawing and analysis of RBDs for system decomposition.
4. Analyze self-diagnostics functions of the investigated system:
  - determination of methods for detection and compensation of failures; for this self-diagnostic approach is analysed, both for hardware and software, as well as the diagnostic coverage.

5. Perform FMECA of the investigated system:
- determination of types of failures and operating modes of the system;
  - determination of the effects of failures and their criticality;
  - determination of root causes of failures.
6. Define dependability indicators of the investigated system
- determination of failures rate;
  - calculation and analysis of dependability and security indicators;
- bottom-up analysis has to be performed, i.e. elements are assembled in units, parts and the system as a whole; the obtained indicators are compared with the specified requirements.

Table 2.1 – Task variants

#	Device type	Failure rate	Redundancy
1	IoT platform 1	$\lambda_{IM} = 10^{-6}$ 1/hour (Input Module) $\lambda_{LM} = 10^{-5}$ 1/hour (Logic Module) $\lambda_{OM} = 5 \cdot 10^{-6}$ 1/hour (Output Module)	1oo2
2	IoT platform 1	$\lambda_{IM} = 10^{-6}$ 1/hour $\lambda_{LM} = 10^{-5}$ 1/hour $\lambda_{OM} = 5 \cdot 10^{-6}$ 1/hour	2oo3
3	IoT platform 1	$\lambda_{IM} = 10^{-6}$ 1/hour $\lambda_{LM} = 10^{-5}$ 1/hour $\lambda_{OM} = 5 \cdot 10^{-6}$ 1/hour	2oo4
4	IoT platform 2	$\lambda_{IM} = 5 \cdot 10^{-6}$ 1/hour $\lambda_{LM} = 2 \cdot 10^{-5}$ 1/hour $\lambda_{OM} = 10^{-5}$ 1/hour	1oo2
5	IoT platform 2	$\lambda_{IM} = 5 \cdot 10^{-6}$ 1/hour $\lambda_{LM} = 2 \cdot 10^{-5}$ 1/hour $\lambda_{OM} = 10^{-5}$ 1/hour	2oo3
6	IoT platform 2	$\lambda_{IM} = 5 \cdot 10^{-6}$ 1/hour $\lambda_{LM} = 2 \cdot 10^{-5}$ 1/hour $\lambda_{OM} = 10^{-5}$ 1/hour	2oo4



#	Device type	Failure rate	Redundancy
...	....	....	...
...	IoT platform N	....	....

### Questions

1. What are the main indicators of dependability and formulas for their calculations?
2. What are the main indicators of safety and formulas for their calculations?
3. Why is it necessary to analyze simultaneously indicators of dependability, safety and security?
4. How is the method of Reliability Block Diagrams applied?
5. How is the method of Failure Mode, Effect and Criticality Analysis (FMECA) applied?

### References

1. Rausand M. Reliability of safety-critical systems : theory and application. – John Wiley & Sons, Inc., Hoboken, New Jersey, USA, 2014.
2. Avizienis, A., Laprie, J.-C., Randell, B. and Landwehr, C. Basic Concepts and Taxonomy of Dependable and Secure Computing. IEEE Transactions on Dependable and Secure Computing (2004), 1(1): 11-33.
3. IEC 61508:2010 “Functional safety of electrical/ electronic/ programmable electronic safety-related systems”.
4. IEC 60812:2006 “Analysis techniques for system reliability – Procedure for failure mode and effects analysis (FMEA)”.

## Laboratory work 24.3

### ANALYSIS OF IOT ATTACKS SCENARIOS

**Goal and objectives:** The purpose of the laboratory work is to study and research possible IoT attacks scenarios.

Learning objectives:

- study the IoT threats;
- study different cyber attacks impacts to IoT systems.

Practical tasks:

- define vulnerable components of IoT systems;
- consider attack scenarios;
- define attack impact and related threats.

#### **IoT systems threats**

The root causes of IoT threats are grouped in 8 categories including the following:

- Nefarious activity / Abuse;
- Eavesdropping / Interception / Hijacking;
- Physical attack;
- Unintentional damages (accidental);
- Failures / Malfunctions;
- Outages;
- Legal;
- Disaster.

#### **A kill chain**

A kill chain is a systematic process to target and engage an adversary to create desired effects. This is an integrated, end-to-end process described as a “chain” because any decency will interrupt the entire process. With respect to computer network attack or computer network espionage, the definitions for these kill chain phases are as follows:

1. Reconnaissance – Research, identification and selection of targets, often represented as crawling Internet websites such as conference proceedings and mailing lists for email addresses, social relationships, or information on specific technologies.

2. Weaponization – Coupling a remote access trojan with an exploit into a deliverable payload, typically by means of an automated tool (weaponizer). Increasingly, client application data files such as Adobe Portable Document Format (PDF) or Microsoft Office documents serve as the weaponized deliverable.

3. Delivery – Transmission of the weapon to the targeted environment. The three most prevalent delivery vectors for weaponized payloads by advanced persistent threats actors, as observed by the Lockheed Martin Computer Incident Response Team (LM-CIRT) for the years 2004-2010, are email attachments, websites, and USB removable media.

4. Exploitation – After the weapon is delivered to victim host, exploitation triggers intruders' code. Most often, exploitation targets an application or operating system vulnerability, but it could also more simply exploit the users themselves or leverage an operating system feature that auto-executes code.

5. Installation – Installation of a remote access trojan or backdoor on the victim system allows the adversary to maintain persistence inside the environment.

6. Command and Control (C&C) – Typically, compromised hosts must beacon outbound to an Internet controller server to establish a C&C channel. Advanced malware especially requires manual interaction rather than conduct activity automatically. Once the C&C channel establishes, intruders have “hands on the keyboard” access inside the target environment.

7. Actions on Objectives – Only now, after progressing through the first six phases, can intruders take actions to achieve their original objectives. Typically, this objective is data exfiltration which involves collecting, encrypting and extracting information from the victim environment; violations of data integrity or availability are potential objectives as well. Alternatively, the intruders may only desire access to the initial victim box for use as a hop point to compromise additional systems and move laterally inside the network.

### **Attack modeling**

An attack modeling is based on the kill chain details consideration for some specific scenario. Consider, for example command injection case targeted to organize a botnet network. This attack scenario is based on the Mirai botnet, which has conducted several of the most forceful

DDoS attacks in recent history, and has proven capable of attacking varied kinds of targets. Therefore, with potential targets such as a hazardous energy infrastructure, the impact of a Mirai's attack can reach extremely critical levels.

This attack entails the exploitation of some vulnerability inside a device to inject commands and obtain administrator privileges, with the purpose of creating a botnet made up of those vulnerable IoT devices. A botnet is a network of automatic devices that interact to accomplish some distributed task. Due to the characteristic interconnection of IoT devices and their poor configuration, carrying out such an attack is simple. There are the following attack steps (these steps can also be represented in a view of a diagram):

1. The attacker scans open ports in devices belonging to an IoT network.

2. If there are any open ports, the attacker tries to gain access to the device using weaknesses such as weak or default passwords, or through exploiting the test/debug modes.

3. Once inside, the attacker injects commands in order to obtain administrator privileges.

4. With these permissions, the attacker tries to connect the device to the Command and Control of the botnet.

5. The attacker downloads and executes a malicious script,

6. The script deletes itself and runs in-memory.

7. Then, it will begin to spread, attacking other vulnerable devices in the same way, in order to gather an IoT device army, conscripting them into a botnet.

8. The attacker can now control the botnet from a Command and Control (C&C) centre, from where he or she will launch distributed attacks conducted by the botnet.

### **Report**

The report should contain: title page with the name of the laboratory work; aim of the work; problem statement according to the task; the progress and results of the study in graphical form; analysis of the results and conclusions.

### **Tasks**

1. Choice you task variant (Table 3.1).
2. Analyze impact of the attack.

3. Analyze related treats of the attack.
4. Describe attack steps in accordance with a kill chain concept.
5. Design a diagram representing the attack scenario.

Table 3.1 – Task variants

#	Attack type
1	Against the network link between controller and actuator
2	Against sensors, modifying the values read by them or their threshold values and settings
3	Against actuators, modifying or sabotaging their normal settings
4	Against the information transmitted via the network
5	Against gateways
6	Manipulation of remote controller devices (e.g. operating panels, smartphones)
7	Against the Safety Instrumented Systems (SIS)
8	Attack using malware
9	DDoS attack using IoT botnets
10	Stepping stones attacks (e.g. against the Cloud)
11	Human error-based and social engineering attacks
12	Highly personalised attacks using Artificial Intelligence Technologies
13	Against the administration systems of IoT
14	Exploit Protocol vulnerabilities
15	Against devices by injecting commands into the system console
16	Power source manipulation and exploitation of vulnerabilities in data readings
17	Ransomware

### Questions

1. What are the main threats of IoT systems?
2. Which security measures are recommended to be implemented for IoT systems?
3. What are the most serious cyber-attacks of IoT systems?
4. How threats and attacks can be modeled for IoT systems?

## References

1. Good Practices for Security of Internet of Things in the context of Smart Manufacturing. – The European Union Agency for Network and Information Security, 2018.
2. Baseline Security Recommendations for IoT in the context of Critical Information Infrastructures. – The European Union Agency for Network and Information Security, 2017.
3. Hutchins E., Cloppert M., Amin R. Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains. – Lockheed Martin Corporation, 2017.
4. NISTIR 8200, Interagency Report on the Status of International Cybersecurity Standardization for the Internet of Things (IoT). – National Institute of Standards and Technologies, 2018.

## **PCM3.2. Safety and security management of IoT**

**Prof., DrS. V. V. Sklyar (KhAI)**

### **Laboratory work 25.1**

#### **DEVELOPMENT OF IOT SAFETY AND SECURITY MANAGEMENT PLAN**

**Goal and objectives:** The purpose of the laboratory work is to study and research structure and content of the Safety and Security Management Plan (SSMP) for IoT systems.

Learning objectives:

- study regulatory requirements to the SSMP;
- study the structure of the SSMP.

Practical tasks:

– develop parts of the SSMP in accordance with management directions.

#### **Safety and Security Management Plan**

The umbrella part of requirement requirements to safety and security is related with safety and security management. The SSMP is the document, which states the main safety and security issues for specific IoT system or systems development and operation project. The SSMP covers a set of processes which can be developed in a view of separated document. There are the following safety and security processes which have to be reflected in the SSMP:

- Human Resource Management;
- Configuration Management;
- Tools Selection and Evaluation;
- Verification and Validation;
- Requirements Tracing;
- Documentation Management;
- Safety and Security Assessment.

Also SSMP has to cover the following issues (see Fig. 1.1):

– Project Policy and Strategy is a declarative description of how and why the goals of the project will be achieved;

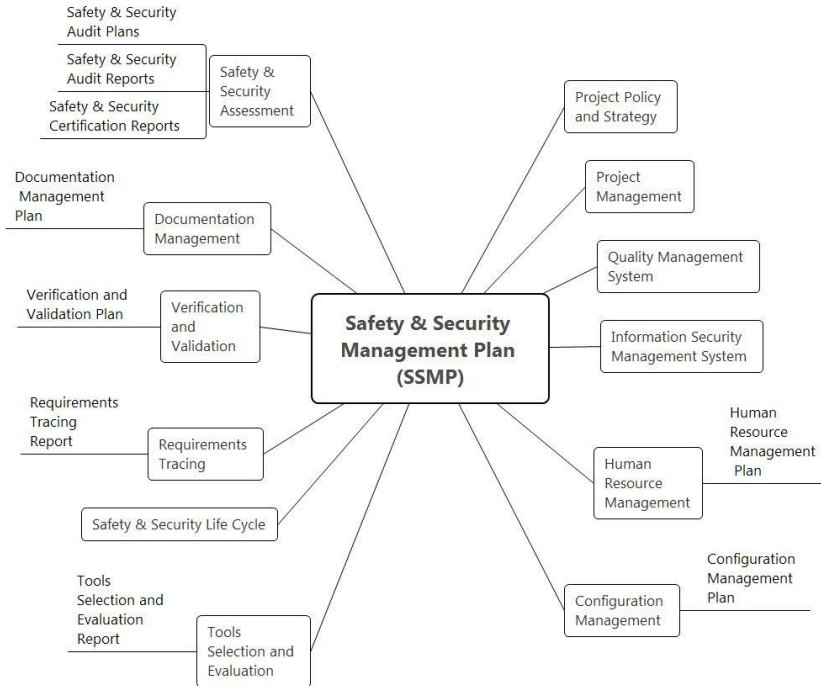


Fig. 1.1 – Structure of the Safety and Security Management Plan (SSMP)

- Project Management is reasonable applicable to project performance since, for example, the IEC 61508-2 (Annex B) requires applying this method to protect the product against systematic failures;
- Quality Management System it important to implement quality for all products and processes; special attention is paid to interaction with suppliers of products and services that affect safety and security;
- Information Security Management System (ISMS) has to cover activities in accordance with requirements of ISO/IEC 27000 “Information technology – Security techniques – Information security management systems” or any other relevant ISMS framework;
- Safety & Security Life Cycle has to be described in SSMP stage by stage.

All the above activities cover both safety and security issues. Additionally ISMS has to cover activities like the following: asset management, identification and authentication, access control, system



perimeter protection, work stations, servers, and other devices protection, network and communications protection, cloud infrastructure protection, database protection, cryptography, monitoring and recovery, incidents response and investigation. All appropriate measures and activities have to be implemented for the considered IoT system.

### **Report**

The report should contain: title page with the name of the laboratory work; aim of the work; problem statement according to the task; the progress and results of the study in graphical form; analysis of the results and conclusions.

### **Tasks**

1. Choice you task variant (Table 1.1).
2. Develop a structure of the SSMP.
3. Develop general sections of the SSMP which are related with policy, strategy, and project management.
4. Depending on task variant, develop sections of the SSMP which are related to:
  - 4.1 Human Resource Management:
    - Organizational chart of the project with a description of project roles;
    - A list of project participants indicating project roles and responsibilities for planning and performing work at various stages of the life cycle;
    - The competence matrix and the conclusions on the adequacy or lack of competencies of the appointed performers, i.e. what knowledge and skills are required for a particular project role and to what extent a particular employee corresponds to them;
    - Personnel training activities aimed at achieving and maintaining the above mentioned competences that are critical for the implementation of the project; training plans and reports should be documented;
    - Communication plan for the project participants;
    - A list of the signatures of personnel, indicating the familiarization with this plan.
  - 4.2. Configuration Management:
    - The roles and responsibilities of project participants in the configuration management process; the Configuration Management &

Change Control Board of the key project participants should be organized with all those, whose opinions are important to consider when making changes;

- An approach to planning and maintaining the configuration management process;

- Resources of the configuration management process, first of all, the applied tools of electronic document management (SVN, Git, etc.);

- The procedure for the identification of the configuration items and the formation of baselines (basic versions);

- The procedure for applying tools to control the versions of software and hardware components of the product and to account for their status;

- The procedure for accessing configuration components and backup storage;

- The procedure and periodicity for configuration audits;

- The procedure for analyzing and eliminating the detected defects and bugs including those found during operation;

- The procedure for change control, including impact analysis and validation of changes.

#### 4.3. Tools Selection and Evaluation:

- A description of the used stack of tools (both software and hardware, both commercially available and in-house) used for product development, testing, and supporting processes (configuration management, documents processing, project management, etc. .) for each of the tools you should specify: type (to support which process is used), name, version number, supplier name, class (T1, T2 or T3), as well as generated outputs in terms of Configuration Items;

- Results of evaluation (analysis) of tools according to a set of predetermined criteria, such as, for example: the functions performed and their applicability in this project, experience of use, available documentation, information about the supplier (market reputation, quality management system, approach to configuration management and etc.), the impact on the safety of the product, the errors found and eliminated, the possible risks of use in terms of failures and the strategy for managing these risks, the availability of compatible products on the market programmable chips (for software development and electronic projects);

- The results of the analysis for compliance with the requirements for the tools specified in IEC 61508-3.

#### 4.4. Documentation Management:

- Requirements to identification, development, execution, coordination and approval of documents;
- Review procedures and criteria for evaluating documents (for example, in the form of checklists);
- A list of project documents and allocation of responsibility for the development, review and approval;
- The procedure for access to documents and access rights of project participants;
- The procedure for making changes to documents, accounting policy and version changes;
- Requirements to use of electronic document management system;
- A structure of the project repository.

#### 4.5. Safety and Security Assessment:

- Periodicity of audits (for example, at the completion of each of the development stages);
- Areas of assessment in terms of the structure of products and processes;
- Involved participants, organizations and other required resources (temporary, financial, required tools, etc.);
- The level of independence of auditors; as noted above, audits can be internal and external; in general, the issue of independence in evaluating safety has its traditions in various industries and countries;
- Competencies of the employers performing the audit;
- Expected results;
- Corrective actions performance;
- An approach to document audit results and requirements for the content of audit report, which shall be issued based on the results of audits;
- Checklists, including a specific set of requirements (issues), compliance with which should be evaluated during the audit; the initial data for compiling an audit checklist are the requirements of SSMP and other plans related to ensuring of safety and security.

Table 1.1 – Task variants

#	Product name	Process name
1	IoT platform 1	Human Resource Management
2	IoT platform 1	Configuration Management
3	IoT platform 1	Tools Selection and Evaluation
4	IoT platform 1	Documentation Management
5	IoT platform 1	Safety and Security Assessment
6	IoT platform 2	Human Resource Management
7	IoT platform 2	Configuration Management
8	IoT platform 2	Tools Selection and Evaluation
9	IoT platform 2	Documentation Management
10	IoT platform 2	Safety and Security Assessment
...	....	....
...	IoT platform N	....

**Questions**

6. What structure of SSMP has to be implemented?

7. What documents can be developed to supplement the functional safety management plan, and in what cases it is advisable to develop such documents?

8. What structure should have Human Resource Management Plan?
9. What part of the Human Resource Management Plan should be developed during the preparatory work for the certification project?
10. List the components of the IoT system configuration.
11. What structure should have Configuration Management Plan?
12. What structure should have Tools Selection and Evaluation Report?
13. What are the criteria for tools selection and evaluation?
14. What structure should have Documentation Plan?
15. What structure should have Safety and Security Audit Plan?

### **References**

1. Medoff M., Faller R. Functional Safety – An IEC 61508 SIL 3 Compatible Development Process. exida L.L.C., Sellersville, PA, USA, 2010.
2. Smith D., Simpson K. Functional Safety. A Straightforward Guide to applying IEC 61508 and Related Standards. Elsevier Butterworth–Heinemann, Oxford, UK, 2004.
3. NIST SP 800-53 Revision 4, Security and Privacy Controls for Federal Information Systems and Organizations. National Institute of Standards and Technologies, 2015.
4. IEC 61508:2010 “Functional safety of electrical/ electronic/ programmable electronic safety-related systems”.

## Laboratory work 25.2

### DEVELOPMENT OF IOT SAFETY AND SECURITY LIFE CYCLE

**Goal and objectives:** The purpose of the laboratory work is to study and research the structure of the Safety and Security Life Cycle (SSLC).

Learning objectives:

- study the structure of the SSLC;
- study content of stages of the SSSC;

Practical tasks:

- develop the SSMP structure;
- develop content of stages of the SSSC.

#### Safety and Security Life Cycle

Existing standards do not describe a life cycle for IoT systems. Thus, we propose interpretation of Safety & Security Life Cycle (SSLC) based on requirements to critical programmable systems (see Fig. 2.1).

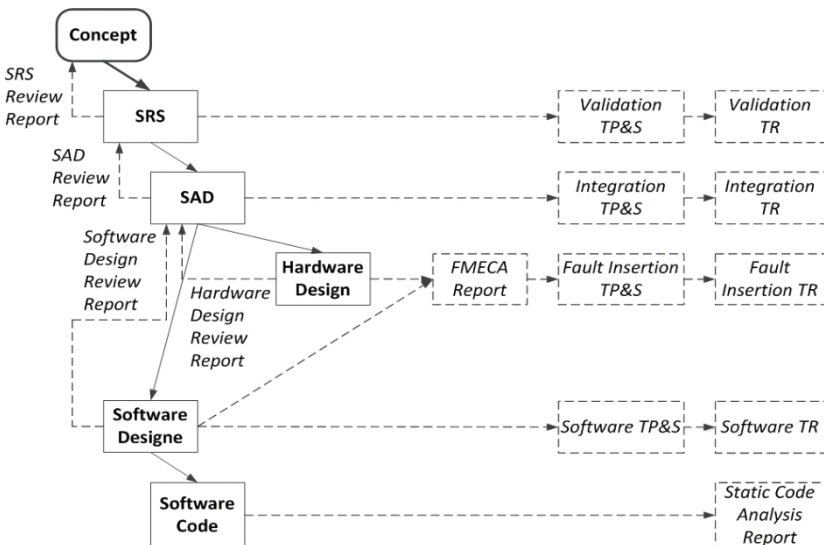


Fig. 2.1 – V-shape Safety & Security Life Cycle

The life cycle model includes the sequentially performed steps (in the diagram, the steps are indicated by the names of the final documents). For top-down branch details of design is performed from system level to hardware and software parts. For down-top branch staged integration with appropriate testing is performed before for software and hardware parts, and after for programmed component and for system as whole.

### **Requirements tracing**

Requirements tracing is one of the processes of a wider area of knowledge called Requirements Engineering. Requirements tracing is a method for managing changing requirements and related artifacts. Requirements tracing solves three main tasks:

- To ensure the implementation at the lower level of all the requirements of the upper level,
- To prevent from undocumented functions appearing on the lower level,
- To support testing of all requirements.

In the considered life cycle (see Fig. 2.1), requirements are traced between design documents as follows. First, direct tracing of requirements from SRS to SAD is performed. Then backtracking of the requirements from SAD to the SRS is performed in order to make sure that the SAD does not include extra functionality that is not documented in the SRS. After SAD, the design process is divided into two streams, which are Hardware Design & Software Design. Forward and backward tracing is performed for both documents. Hardware Design includes mainly drawings in which it is problematic to place tags, so the Hardware Design Review Report is laid out under the tracing.

During testing, requirements tracing is done by extracting requirements from project documents. For complicated projects, the development of test documents can take place in two stages. First, a test plan is developed, containing a list of test requirements, and then test cases are developed for each requirement in the test specification. During testing, direct tracing of requirements is carried out from the project document to the test plans and specifications, and then to the testing report. For testing by the method of seeding defects, the list of tests is extracted from the FMECA report by analyzing self-diagnosed failures. For a reasonable set of failures, a set of tests is made, on which diagnostic functions are checked. Backward tracking is not critical here,

because if additional tests are performed that are not due to project documents, this will not affect safety and security.

### Report

The report should contain: title page with the name of the laboratory work; aim of the work; problem statement according to the task; the progress and results of the study in graphical form; analysis of the results and conclusions.

### Tasks

1. Choice you task variant (Table 2.1).
2. Develop a structure of the SSLC.
3. Depending on task variant, develop stages of the SSMP which are related to:
  - Safety Requirements Specification (SRS);
  - Review of the SRS;
  - System Architecture Design (SAD);
  - Review of the SAD;
  - Hardware Design;
  - Review of the Hardware Design;
  - Failure Mode, Effect and Criticality Analysis (FMECA);
  - Software Design;
  - Review of the Software Design;
  - Software Coding;
  - Static Code Analysis (SCA);
  - Software Testing;
  - Fault Insertion Testing;
  - Integration Testing;
  - Validation Testing.
4. Depending on task variant, perform requirements tracing between relevant SSLC stages.

Table 2.1 – Task variants

#	Product name	Process name
1	IoT platform 1	Safety Requirements Specification (SRS); Review of the SRS; System Architecture Design (SAD);



PCM3.2. Safety and security management of IoT

#	Product name	Process name
		Validation Testing
2	IoT platform 1	System Architecture Design (SAD); Review of the SAD; Hardware Design; Review of the Hardware Design; Failure Mode, Effect and Criticality Analysis (FMECA); Software Design; Fault Insertion Testing; Integration Testing
3	IoT platform 1	Software Design; Review of the Software Design; Software Coding; Static Code Analysis (SCA); Software Testing.
4	IoT platform 2	Safety Requirements Specification (SRS); Review of the SRS; System Architecture Design (SAD); Validation Testing
5	IoT platform 2	System Architecture Design (SAD); Review of the SAD; Hardware Design; Review of the Hardware Design; Failure Mode, Effect and Criticality Analysis (FMECA); Software Design; Fault Insertion Testing; Integration Testing
6	IoT platform 2	Software Design; Review of the Software Design; Software Coding; Static Code Analysis (SCA);

#	Product name	Process name
		Software Testing.
...	....	....
...	IoT platform N	....

**Questions**

1. Describe a structure of V-shape life cycle.
2. What is a difference between software life cycle and IoT system life cycle?
3. What is a purpose of requirements tracing?
4. Which design and test documents have to be covered with requirements tracing?

**References**

1. Medoff M., Faller R. Functional Safety – An IEC 61508 SIL 3 Compatible Development Process. exida L.L.C., Sellersville, PA, USA, 2010.
2. IEC 61508:2010 “Functional safety of electrical/ electronic/ programmable electronic safety-related systems”.
3. Standard glossary of terms used in Requirements Engineering, Version 1.3. Requirements Engineering Qualification Board, 2014.
4. Hanssen G., Stålhane T, Myklebust T. SafeScrum® – Agile Development of Safety-Critical Software. Springer, 2018.
5. NIST SP 1500-201, Framework for Cyber-Physical Systems. National Institute of Standards and Technologies, 2017.

## Seminar 25.3

### STUDY OF IOT VERIFICATION AND VALIDATION METHODS

**The purpose and tasks of the seminar:** acquisition of knowledge and practical skills on the work with review, analysis and testing technics for IoT systems.

**Training tasks:**

- study of review and analysis methods;
- study of testing methods.

**Practical tasks:**

- conducting review and analysis of IoT hardware and software components;
- conducting testing of IoT hardware and software components.

**Preparation for the seminar**

Preparation for the seminar includes the following steps.

1) Obtaining (defining) the topic of the paper.

Themes of the paper may be formulated by students independently based on descriptions of the following verification and validation methods:

1. Review of the SRS.
2. Review of the SAD.
3. Review of the Hardware Design.
4. Failure Mode, Effect and Criticality Analysis (FMECA).
5. Review of the Software Design.
6. Static Code Analysis.
7. Software Functional Testing.
8. Software Structural Testing.
9. Fault Insertion Testing.
10. Integration Testing.
11. Validation Testing.
12. Penetration testing.

2) Search for information on the paper of the abstract and its preliminary analysis.

3) Development of the paper plan and presentation.

The plan of a typical paper includes:

- purpose and main tasks;
- analysis of the issue;
- the main part (the principle of the algorithm, modeling, research results, advantages and disadvantages of the algorithm, examples of application);
- conclusions;
- list of references;
- applications.

4) Writing the paper.

The paper should be 15-20 pages of A4 format (font 14.1,5 in., Field 2 cm), including title page, content, main text, references, applications.

An obligatory appendix to the paper is presentation slides.

5) Preparation of the presentation.

The presentation should be done in PowerPoint and corresponds to the paper plan (8-10 slides for 10 minute talk).

The presentation should include the following slides:

- title slide (university, department, discipline, topic of the report, author, date of presentation);
- content (structure) of the report;
- relevance of the considered issues, the purpose and tasks of the report;
- slides with the disclosure of the main content;
- conclusions;
- references.

The content of the slides should not contain parts of the text from the paper, but includes the keywords, figures, and formulas.

### **The presentation**

The presentation (report) is carried out at the seminar, it takes 15 minutes and includes the report (10 minutes) and discussion (5 minutes). Languages of presentations and papers are Ukrainian or English.

## **Evaluation**

Assessment of this work includes:

- the quality of the text of the paper (form and content),
- presentation quality (content and design);
- the quality of the report (content, logical structure, conclusions);
- distribution of time by sections;
- completeness and correctness of answers.

## **Questions**

1. Describe documents review method.
2. Describe static code analysis method.
3. Describe functional testing method.
4. Describe structural testing method.

## **References**

1. Standard glossary of terms used in Software Testing, Version 2.3. International Software Testing Qualifications Board, 2014.
2. Hanssen G., Stålhane T, Myklebust T. SafeScrum® – Agile Development of Safety-Critical Software. Springler, 2018.
3. NISTIR 8200, Interagency Report on the Status of International Cybersecurity Standardization for the Internet of Things (IoT). – National Institute of Standards and Technologies, 2018.
4. NIST SP 1500-201, Framework for Cyber-Physical Systems. National Institute of Standards and Technologies, 2017.
5. Martins B., Laranjeiro N., Vieira M. INTENSE: INteroperability TEstiNg as a Service // Proceedings of 2017 IEEE International Conference on Web Services (ICWS 2017).

## **PCM3.3. Assurance Case for IoT**

**Prof., DrS. V. V. Sklyar (KhAI)**

### **Laboratory work 26.1**

#### **SOFTWARE TOOLS FOR IOT ASSURANCE CASE DEVELOPMENT**

**Goal and objectives:** The purpose of the laboratory work is to study and research structure and content of the Assurance Case for IoT systems.

Learning objectives:

- study notation (GSN) for the Assurance Case;
- study the structure of the Assurance Case.

Practical tasks:

- develop parts of the Assurance Case using software tools.

#### **Assurance Case concept**

Final safety and security assessment is running after completion of all development, verification and validation stages. In this section we discuss how can all project artifacts be represented for safety and security assessment, and what is the way to most effectively confirm compliance with the safety and security requirements? The answer to these questions is provided by the Assurance Case methodology, which is widely used in the practice of safety and security assessment.

The Assurance Case is a structured set of arguments and documentary evidence that justify the compliance of a system or service with specified requirements.

Licensing and certification authorities check the Assurance Case, as an integral document proving compliance with the entire set of requirements to safety and security. The Assurance Case can be either compiled by the project team or outsourced.

In justifying safety and security, we need to confirm the compliance of a certain system or software with the requirements set. At the same time compliance with a particular requirement is the goal of the Assurance Case. In addition, there is a set of documented evidence that requirements are met. To associate evidence with goals and

requirements, an argumentation system is used, which is given special attention in the Assurance Case (see Fig. 1.1). The lack of arguments or evidence indicates a failure to comply with safety and security requirements.

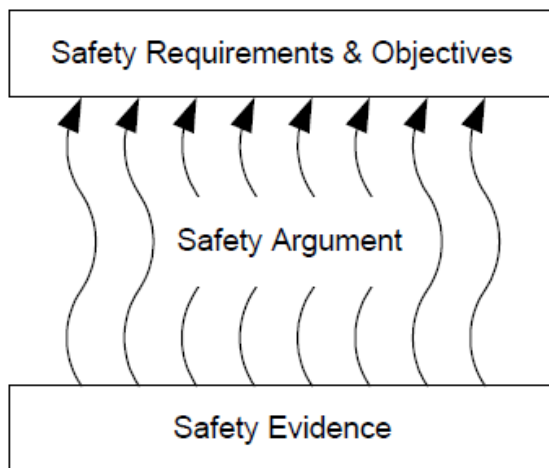


Fig. 1.1 – Objectives, arguments and evidence of safety

The Assurance Case should be developed in stages throughout the life cycle, starting from the first stage of the concept and contract. Then, over the course of development, deviations from requirements can be quickly identified and corrected with less cost. At the same time, assessment of the implementation of both product requirements and requirements for safety and security management processes is supported.

### Goal Structuring Notation (GSN)

GSN (Goal Structuring Notation), like CAE (Claim – Argument – Evidence) notation, operates with entities such as goal (indicated by a rectangle and is analogous to a claim), argumentation strategy (indicated by a parallelogram and is analogous to argument), and a solution (indicated by a circle and is analogous to evidence) (see Fig. 1.2).

The context is used for informational support of goal setting. Assumptions and justifications can be used to support argumentation. The goal structure is also hierarchical. It should be noted that the GSN is

described in the GSN Community Standard, and Structured Assurance Case Metamodel is developed by Object Management Group.

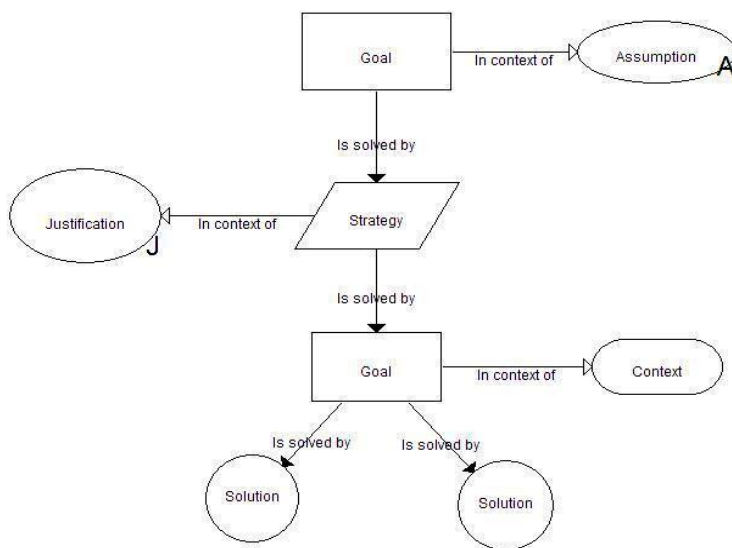


Fig. 1.2 – Goal Structuring Notation (GSN) notation: main components

### Software tool Astah GSN

Astah GSN developed by Change Vision company from Japan. The company was created in 2006. Astah GSN was developed as a part of the Astah Professional toolkit, which is a media for complex systems modeling.

As the name suggests, this program supports only GSN. In addition, it can create Mind Map diagrams. In the graphical editor, you can attach text and hyperlinks to graphic symbols. Charts are saved in the internal format of the program (\*.agml). It supports the export of diagrams in the form of figures, as well as in the XMI format (XML Metadata Interchange).

You can download a trial version of the software from the Astah GSN website. Supported operating systems are Windows, MacOS, and Linux. The trial version will work 50 days. User manuals and video demonstrations are also available on the site (see Fig. 1.3).



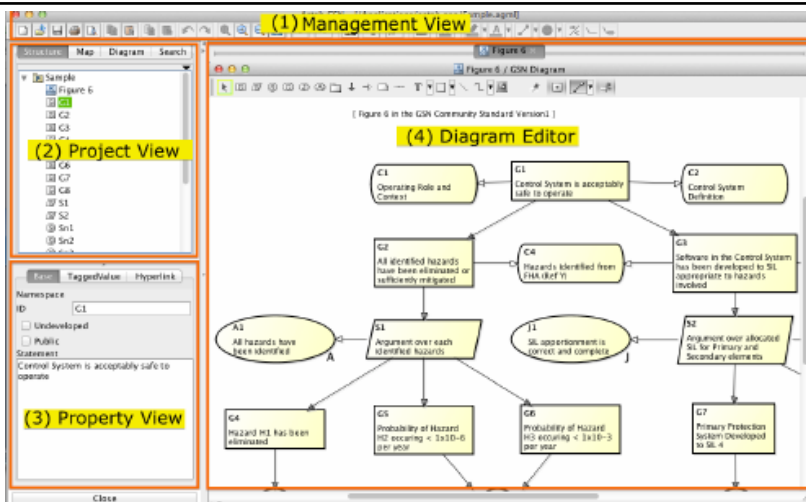


Fig. 1.3 – Astah GSN program interface

Fig. 1.4 represents a GSN diagram compliant with the top level of functional safety requirements in accordance with IEC 61508. The diagram is done in the Astah GSN environment. This diagram will be used as initial for performance of the actual laboratory work.

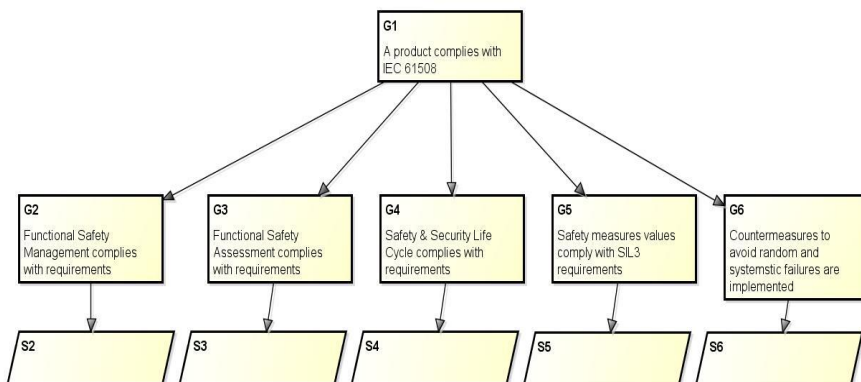


Fig. 1.4 – Template of an initial GSN diagram used for Assurance Case development

The goals G2-G6 (Fig. 1.4) are related with five the main groups of functional safety requirements:

- G2 – requirements to functional safety management;
- G3 – requirements to functional safety assessment;
- G4 – requirements to functional safety life cycle;
- G5 – requirements to values of functional safety indicators;
- G6 – requirements to measures and techniques of functional safety assurance.

### Report

The report should contain: title page with the name of the laboratory work; aim of the work; problem statement according to the task; the progress and results of the study in graphical form; analysis of the results and conclusions.

### Tasks

1. Choice you task variant (Table 1.1).
2. Load and install the program Astah GSN.
3. Launch Astah GSN.
4. Develop an initial GSN diagram in accordance with Fig. 1.4.
5. Depending on task variant, develop solutions of GSN diagram for IoT safety assurance.
6. Prove IoT system safety assurance and IoT system compliance with IEC 61508 requirements on the base of the developed Assurance Case.

Table 1.1 – Task variants

#	Product name	Solution name
1	IoT platform 1	G2 – requirements to functional safety management
2	IoT platform 1	G3 – requirements to functional safety assessment
3	IoT platform 1	G4 – requirements to functional safety life cycle

#	Product name	Solution name
4	IoT platform 1	G5 – requirements to values of functional safety indicators
5	IoT platform 1	G6 – requirements to measures and techniques of functional safety assurance
6	IoT platform 2	G2 – requirements to functional safety management
7	IoT platform 2	G3 – requirements to functional safety assessment
8	IoT platform 2	G4 – requirements to functional safety life cycle
9	IoT platform 2	G5 – requirements to values of functional safety indicators
10	IoT platform 2	G6 – requirements to measures and techniques of functional safety assurance
...	....	....
...	IoT platform N	....

### Questions

1. Define the Assurance Case methodology.
2. What is the history of the development of the Assurance Case methodology?
3. What types of notations can be used to represent the Assurance Case?
4. Give a description of the GSN notation.
5. What regulatory documents govern the GSN notation?
6. What software tools are used to support the Assurance Case methodology?
7. What are the main disadvantages and advantages of applying the Assurance Case methodology?

8. What the basic structure of the Assurance Case for IoT systems?

### **References**

1. Medoff M., Faller R. Functional Safety – An IEC 61508 SIL 3 Compatible Development Process. exida L.L.C., Sellersville, PA, USA, 2010.

2. IEC 61508:2010 “Functional safety of electrical/ electronic/ programmable electronic safety-related systems”.

3. Kelly T. Arguing Safety: A Systematic Approach to Managing Safety Cases. PhD thesis. Univ. of York, 1998.

4. Structured Assurance Case Metamodel, v2.0. Object Management Group, 2016.

5. GSN Community Standard, Version 1. Origin Consulting (York) Limited, 2011.

6. Weinstock C., Goodenough J. Towards an Assurance Case Practice for Medical Devices, Technical Note CMU/SEI-2009-TN-018. SEI, 2009.

## Laboratory work 26.2

### SAFETY AND SECURITY TECHNIQUES AND MEASURES FOR IOT SYSTEMS

**Goal and objectives:** The purpose of the laboratory work is to study and research the safety and security techniques and measures for IoT systems.

Learning objectives:

- study the organizational safety and security techniques and measures;

- study the technical safety and security techniques and measures;

- study the techniques and measures for cyber attacks avoidance.

Practical tasks:

- implement the organizational safety and security techniques and measures for IoT system;

- implement the technical safety and security techniques and measures for IoT system.

- implement the techniques and measures for IoT systems cyber attacks avoidance.

### Safety and security techniques and measures Safety and Security Life Cycle

Organizational safety and security techniques and measures implemented for IoT systems include the following:

- project management;

- documents management;

- Safety and Security Life Cycle (SSLC) implementation;

- using best practices and standards for programming and system and engineering;

- certified translators and compilers, code libraries and certified software components

- quality control in the production of hardware components

- formal and semiformal notations to develop specifications and design.

- asset management;

- risks, threats and vulnerabilities management;

- incident handling;

- training and awareness;

– third party management.

Technical safety and security techniques and measures implemented for IoT systems include the following:

- redundancy;
- diversity;
- separation and independency;
- self-diagnostics;
- internal and external hazards protection;
- human factor engineering.

Special techniques and measures against cyber attacks avoidance include the following:

- trust and integrity management;
- cloud security;
- business continuity and recovery;
- machine-to-machine security;
- data protection;
- software/firmware updates;
- access control;
- networks, protocols and encryption;
- monitoring and auditing;
- configuration management.

### **Tasks**

1. Choose your task variant (Table 2.1).

2. Depending on task variant, analyses IEC 61508 and study methods of random failures, systematic failures and software failures avoidance:

- IEC 61508-2 (Annex A);
- IEC 61508-7 (Annex A);
- IEC 61508-2 (Annex B);
- IEC 61508-7 (Annex B);
- IEC 61508-3 (Annex A);
- IEC 61508-7 (Annex C).

3. Depending on task variant, analyses the ENISA report “Good Practices for Security of Internet of Things in the context of Smart Manufacturing” (Annex B) and study methods of cyber attacks avoidance.

4. Describe implementation of the organizational safety and security techniques and measures for IoT system.

5. Describe implementation of the technical safety and security techniques and measures for IoT system.

5. Describe implementation of the techniques and measures for IoT systems cyber attacks avoidance.

Table 2.1 – Task variants

#	Product name	Measures name
1	IoT platform 1	<p>Organizational measures:</p> <ul style="list-style-type: none"> <li>– project management;</li> <li>– documents management;</li> <li>– SSLC implementation;</li> <li>– using best practices and standards for programming and system and engineering.</li> </ul> <p>Technical measures:</p> <ul style="list-style-type: none"> <li>– redundancy;</li> <li>– diversity.</li> </ul> <p>Cyber attacks avoidance measures:</p> <ul style="list-style-type: none"> <li>– trust and integrity management;</li> <li>– cloud security;</li> <li>– business continuity and recovery.</li> </ul>
2	IoT platform 1	<p>Organizational measures:</p> <ul style="list-style-type: none"> <li>– certified translators and compilers, code libraries and certified software components</li> <li>– quality control in the production of hardware components</li> <li>– formal and semiformal notations to develop specifications and design.</li> <li>– asset management;</li> </ul> <p>Technical measures:</p> <ul style="list-style-type: none"> <li>– separation and independency;</li> <li>– self-diagnostics.</li> </ul> <p>Cyber attacks avoidance measures:</p> <ul style="list-style-type: none"> <li>– machine-to-machine security;</li> <li>– data protection;</li> <li>– software/firmware updates.</li> </ul>

#	Product name	Measures name
3	IoT platform 1	<p>Organizational measures:</p> <ul style="list-style-type: none"> <li>– project management;</li> <li>– documents management;</li> <li>– SSLC implementation;</li> <li>– using best practices and standards for programming and system and engineering.</li> </ul> <p>Technical measures:</p> <ul style="list-style-type: none"> <li>– redundancy;</li> <li>– diversity.</li> </ul> <p>Cyber attacks avoidance measures:</p> <ul style="list-style-type: none"> <li>– trust and integrity management;</li> <li>– cloud security;</li> <li>– business continuity and recovery.</li> </ul>
4	IoT platform 2	<p>Organizational measures:</p> <ul style="list-style-type: none"> <li>– certified translators and compilers, code libraries and certified software components</li> <li>– quality control in the production of hardware components</li> <li>– formal and semiformal notations to develop specifications and design.</li> <li>– asset management;</li> </ul> <p>Technical measures:</p> <ul style="list-style-type: none"> <li>– separation and independency;</li> <li>– self-diagnostics.</li> </ul> <p>Cyber attacks avoidance measures:</p> <ul style="list-style-type: none"> <li>– machine-to-machine security;</li> <li>– data protection;</li> <li>– software/firmware updates.</li> </ul>
5	IoT platform 2	<p>Organizational measures:</p> <ul style="list-style-type: none"> <li>– risks, threats and vulnerabilities management;</li> <li>– incident handling;</li> <li>– training and awareness;</li> <li>– third party management.</li> </ul> <p>Technical measures:</p>



#	Product name	Measures name
		<ul style="list-style-type: none"> <li>– internal and external hazards protection;</li> <li>– human factor engineering.</li> </ul> Cyber attacks avoidance measures: <ul style="list-style-type: none"> <li>– access control;</li> <li>– networks, protocols and encryption;</li> <li>– monitoring and auditing;</li> <li>– configuration management.</li> </ul>
6	IoT platform 2	Organizational measures: <ul style="list-style-type: none"> <li>– project management;</li> <li>– documents management;</li> <li>– SSLC implementation;</li> <li>– using best practices and standards for programming and system and engineering.</li> </ul> Technical measures: <ul style="list-style-type: none"> <li>– redundancy;</li> <li>– diversity.</li> </ul> Cyber attacks avoidance measures: <ul style="list-style-type: none"> <li>– trust and integrity management;</li> <li>– cloud security;</li> <li>– business continuity and recovery.</li> </ul>
...	....	....
...	IoT platform N	....

### Questions

1. Describe the organizational safety and security techniques and measures for IoT system.
2. Describe the technical safety and security techniques and measures for IoT system.
3. Describe the techniques and measures for IoT systems cyber attacks avoidance.

## References

1. Medoff M., Faller R. Functional Safety – An IEC 61508 SIL 3 Compatible Development Process. exida L.L.C., Sellersville, PA, USA, 2010.
2. IEC 61508:2010 “Functional safety of electrical/ electronic/ programmable electronic safety-related systems”.
3. NIST SP 1500-201, Framework for Cyber-Physical Systems. National Institute of Standards and Technologies, 2017.
4. Good Practices for Security of Internet of Things in the context of Smart Manufacturing. – The European Union Agency for Network and Information Security, 2018.
5. Baseline Security Recommendations for IoT in the context of Critical Information Infrastructures. – The European Union Agency for Network and Information Security, 2017.

## Seminar 26.3

### STUDY OF GREEN ASSURANCE CASE FOR IOT

**The purpose and tasks of the seminar:** acquisition of knowledge and practical skills on the work with Green Assurance Case for IoT systems.

**Training tasks:**

- study of methods for energy consumption reducing;
- study of Assurance Case development methods.

**Practical tasks:**

- conducting energy consumption analysis of IoT hardware and software components;
- conducting testing of energy consumption for IoT hardware and software components.

**Preparation for the seminar**

Preparation for the seminar includes the following steps.

1) Obtaining (defining) the topic of the paper.

Themes of the paper may be formulated by students independently based on descriptions of the Green Assurance Case issues:

1. Green ITs for sustainable development.
2. Green IT principles.
3. Temperature control techniques.
4. Dynamic Voltage and Frequency Scaling (DVFS).
5. Wireless Sensor Networks (WSN).
6. Sensor Network as a Service (SNaaS).
7. Energy consumption of IoT devices.
8. Energy consumption of a data center.
9. GREENSOFT model.

2) Search for information on the paper of the abstract and its preliminary analysis.

3) Development of the paper plan and presentation.

The plan of a typical paper includes:

- purpose and main tasks;
- analysis of the issue;

- the main part (the principle of the algorithm, modeling, research results, advantages and disadvantages of the algorithm, examples of application);
- conclusions;
- list of references;
- applications.

#### 4) Writing the paper.

The paper should be 15-20 pages of A4 format (font 14,1,5 in., Field 2 cm), including title page, content, main text, references, applications.

An obligatory appendix to the paper is presentation slides.

#### 5) Preparation of the presentation.

The presentation should be done in PowerPoint and corresponds to the paper plan (8-10 slides for 10 minute talk).

The presentation should include the following slides:

- title slide (university, department, discipline, topic of the report, author, date of presentation);
- content (structure) of the report;
- relevance of the considered issues, the purpose and tasks of the report;
- slides with the disclosure of the main content;
- conclusions;
- references.

The content of the slides should not contain parts of the text from the paper, but includes the keywords, figures, and formulas.

### **The presentation**

The presentation (report) is carried out at the seminar, it takes 15 minutes and includes the report (10 minutes) and discussion (5 minutes). Languages of presentations and papers are Ukrainian or English.

### **Evaluation**

Assessment of this work includes:

- the quality of the text of the paper (form and content),
- presentation quality (content and design);
- the quality of the report (content, logical structure, conclusions);
- distribution of time by sections;
- completeness and correctness of answers.

### **Questions**

1. What are the main disadvantages and advantages of applying the Assurance Case methodology?
2. What role does the human factor play in applying the Assurance Case methodology?
3. What the basic structure of the Assurance Case for IoT systems?
4. Which are the approaches to implement green technologies for IoT systems?

### **References**

1. Kelly T. *Arguing Safety: A Systematic Approach to Managing Safety Cases*. PhD thesis. Univ. of York, 1998.
2. Sheng, Z., Mahapatra, C., Zhu, C., Leung, V. Recent Advances in Industrial Wireless Sensor Networks Toward Efficient Management in IoT // *IEEE Access*, Volume 3, 2015, pp. 622-637.
3. Ardito, L., Morisio, M. Green IT – Available data and guidelines for reducing energy consumption in IT systems // *Sustainable Computer Information Systems*, Volume 4, 2014, pp. 24–32.
4. Chen, F., Schneider, J.-G., Yang, Y., Grundy, J., He, Q. An Energy Consumption Model and Analysis Tool for Cloud Computing Environments. In: *Proceedings of the First International Workshop on Green and Sustainable Software*. – Piscataway, NJ, USA, pp. 45–50, June 2012.
5. Naumann, S., Dick, M., Kern, E., Johann, T. The GREENSOFT Model: A reference model for green and sustainable software and its engineering // *Sustainable Computer Information Systems*, Volume 1, 2011, pp. 294-300.
6. Sklyar V., Kharchenko V. *Green Assurance Case: Applications for Internet of Things*. *Green IT Engineering: Social, Business and Industrial Applications*. Studies in Systems, Decision and Control, vol. 171. Springer, Cham, 2019.

## PCM3.4. Security of IoT Based Blockchain Technology

DrS. V. V. Yatskiv, Ass. Prof., Dr. N. G. Yatskiv

### Laboratory work 27.1

#### THE GENERATION AND RESEARCH OF HASH FUNCTION FOR BLOCKCHAIN TECHNOLOGY

**Goal and objectives:** The purpose of the laboratory work is to study and research the hash functions and Merkle trees.

Learning objectives:

- study the properties of hash functions;
- study different algorithms for hash functions calculation;
- study the Merkle tree structure.

Practical tasks:

- acquire practical skills in hash functions calculation using Python libraries;
- acquire practical skills in using hash functions for data integrity protection:
- create the Merkle tree and research of its effectiveness.

#### 1.1 Hash Function

Hashing is the process of converting of arbitrary length input data array into a (output) bit string of fixed length. For example, a hash function can take a string with any number of characters (one letter or a whole literary work), and at the output we get a string with a strictly defined number of characters (digest).

Hashing is an irreversible mathematical function which relatively easy calculated but it is much more difficult to find an argument using the value of hash function (Figure 1.1).

Properties of the cryptographic hash function:

- input data can be of arbitrary size;
- output data has a fixed size;
- hash function is a [one-way function](#), that is, a function which is [infeasible](#) to invert;
- it is almost impossible to get the same hash values from two different input value sets.

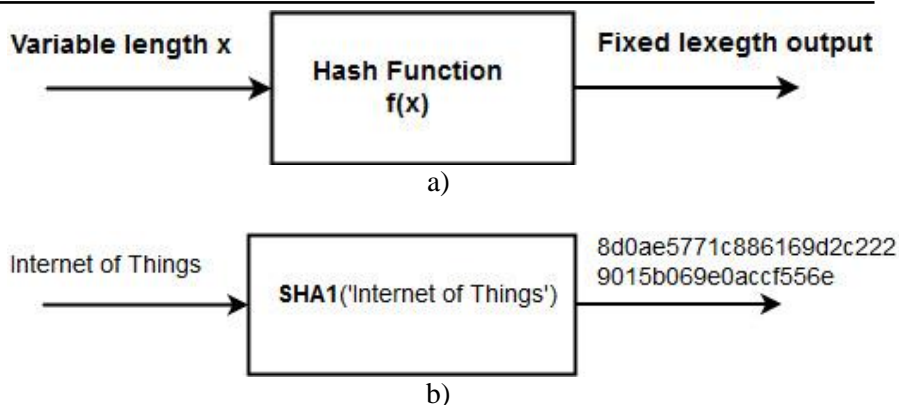


Figure 1.1 – Hashing example

Practically all programming languages support hash functions (often implemented via libraries). For example, they are used for the implementation of hash tables and sets (HashMap / HashSet в Java, dict and set in Python, Map, Set and objects in JavaScript, etc).

**Secure Hash Algorithm (SHA).** The National Institute of Standards and Technology (NIST) has developed a SHA algorithm that is included in the Secure Hash Standard (SHS) standard. NIST has published SHA-1 in 1994. The SHA-2 algorithm family replaced SHA-1 by adding four additional hash functions to create the SHA family: SHA-224 (224 bits); SHA-256 (256 bits); SHA-384 (384 bits); SHA-512 (512 bits).

SHA-2 is a more stable algorithm and it is increasingly being used instead of MD5. SHA-256, SHA-384 and SHA-512 are the next generation algorithms.

Cryptographic hash functions are developed and thoroughly checked by researchers all over the world.

The Python Standard library includes a hashlib module that contains an interface for the most popular hashing algorithms.

It is necessary to execute the following commands to view available hashing algorithms in the hashlib module:

```
import hashlib
print(hashlib.algorithms_available)
print(hashlib.algorithms_guaranteed)
```

The code works in Python 3.2 and newer versions. To start in

Python 2.x, remove `algorithms_available` and `algorithms_guaranteed` calls.

Results:

```
print(hashlib.algorithms_available)
{'sha1', 'shake_256', 'DSA', 'MD4', 'SHA384', 'blake2b', 'SHA512',
'SHA224', 'dsaWithSHA', 'md5', 'dsaEncryption', 'whirlpool', 'DSA-
SHA', 'sha384', 'sha3_224', 'SHA1', 'blake2s', 'md4', 'ecdsa-with-SHA1',
'sha', 'SHA256', 'sha3_256', 'RIPEMD160', 'MD5', 'sha256', 'shake_128',
'ripemd160', 'SHA', 'sha3_512', 'sha224', 'sha512', 'sha3_384'}
```

We can use `algorithms_available` function to get the list of all the algorithms available in the system, including those available through OpenSSL. Duplicate algorithm names can be seen also.

By using `algorithms_guaranteed` function you can see the algorithms present in the module.

```
print(hashlib.algorithms_guaranteed)
{'blake2s', 'sha1', 'shake_256', 'sha256', 'md5', 'shake_128',
'sha3_512', 'blake2b', 'sha224', 'sha384', 'sha3_256', 'sha3_224', 'sha512',
'sha3_384'}
```

Source code for hash function calculation on Python [2]:

```
import hashlib
def hash_hex(message):
    return hashlib.sha256(message.encode()).hexdigest()

hash_hex('Internet of Things')
'49ea9bbbcc65f1f558ab947737be1c471ff67312b5a3f1b52270d674
fccfeafb'
```

The `hash_hex ()` function calculates the hash value for a string in hexadecimal form. In the example below, the SHA-256 function is used.

The cryptographic hash function provides protection against collisions (it is impossible to obtain two identical hashes at different initial data) and has the avalanche effect, wherein if an input is changed slightly the output changes significantly.

The avalanche effect in the SHA-256 hash function looks like this:

```
>>> hash_hex('Internet of Things')
```



```
'49ea9bbbcc65f1f558ab947737be1c471ff67312b5a3f1b52270d674
fccfeafb'
>>> hash_hex('Internet Of Things')
'4039809521288eb0b424ba491464e7a3d2c780fd92693722959bbc
531e4560df'
>>> hash_hex('Internet of things')
'd1fd0774e43dd7abf891c724fe3bbc5a9eec8a7f1ec421a81677d9ef
ed9b63c2'
```

Hash functions in the blockchain provide the "irreversibility" of the whole transaction chain. The thing is, that each new transaction block uses the hash of the previous block. The hash of the block depends on all the transactions in the block, but instead sequentially transferring of transaction hashes, these hashes are combined in one hash-value using a Merkle tree. Thus, hashes are used to replace pointers in common data structures: linked lists and binary trees.

Due to the use of hashes, the general state of Blockchain - all the transactions that were ever performed and their sequence - can be presented by single number: the hash of the newest block. Hash function ensures the integrity of one block and the whole blockchain.

Salts are used to safeguard passwords in storage. A new salt is randomly generated for each password. The salt value is appended to the plaintext password and then the result is hashed, this is referred to as the hashed value. Since salts are different in each case, they also protect commonly used passwords, or those users who use the same password on several sites, by making all salted hash instances for the same password different from each other. The different hash values for two identical passwords is shown in Figure 2. Both the salt value and hashed value are stored in a database.

Let's consider an example.

Source code for the computation of hash value of password:

```
import uuid
import hashlib

def hash_password(password):
    # uuid is used to generate a random number
    salt = uuid.uuid4().hex
```

```
return hashlib.sha256(salt.encode()+
password.encode()).hexdigest() + ':' + salt
```

```
pass_new = input('Enter a password: ')
hashed_password = hash_password(pass_new)
print(hashed_password)
```

```
Enter a password: 123
8bb65c415c1a8ec4f3daba0adca13ee012361a4ec633c4db7da40bb9
e510440b:01a3bd9ea8624087a730e47c734be011
```

```
Enter a password: 123
f1a1243bf03f099d3f6b747b21317f3738b802eac9211f56a9aaeec3
94d77110:b0f777b999044b69b69e066cee1e2817
```

In the example above, the hash values of the same passwords are completely different, because the "salt" in each case is different.

The determined conditions for hash functions are often used in Blockchain, for instance, hash sum must be less then defined value or it must start from defined number of zeros: 0000...009d2c44a674ca95.....

Let us consider an example of hash sum computation that starts from two zeros:

Source code for computation of the hash sum under specified condition:

```
import hashlib
import time

#selecting hashing algorithm
hash256 = hashlib.sha256()

#searching for hash starting with '00' in hex
time_current = time.time()
mystring = "Internet of Things"
nonce=0
while True:
    hash256.update(str(mystring+'nonce').encode())
    hashed_bytes = hash256.digest()
```

```
if (hashed_bytes[0] == 0):
    time_end = time.time()
    time_diff = time_end - time_current
    print ('Nonce: '+str(nonce))
    print ('Hash(Nonce): '+hash256.hexdigest())
    print ('Seconds: '+str(time_diff))
    #print (str(nonce))
    #print (hashed_bytes)

    break

nonce+=1
```

The result:

Nonce: 190

Hash(Nonce):

00ccf80ce32418ea3b6b0ef8c6898a6bf0a24912560ef0c4efa7f24e1b5290  
17

Seconds: 0.0009989738464355469

To compute a hash function that starts from four zeros "0000" you need to change the condition as follows:

```
if (hashed_bytes[0] ==0) & (hashed_bytes[1] ==0)
```

## 1.2 Merkle tree

A Merkle tree, also known as a binary hash tree, is a data structure used for efficiently summarizing and verifying the integrity of large sets of data. Merkle trees are binary trees containing cryptographic hashes. The term “tree” is used in computer science to describe a branching data structure, but these trees are usually displayed upside down with the “root” at the top and the “leaves” at the bottom of a diagram, as you will see in the examples that follow.

Merkle trees are used in bitcoin to summarize all the transactions in a block, producing an overall digital fingerprint of the entire set of transactions, providing a very efficient process to verify whether a transaction is included in a block (Figure 1.2). A Merkle tree is constructed by recursively hashing pairs of nodes until there is only one hash, called the root, or Merkle root.

The cryptographic hash algorithm used in bitcoin’s Merkle trees is SHA256 applied twice, also known as double-SHA256.

When N data elements are hashed and summarized in a Merkle tree, you can check to see if any one data element is included in the tree with at most  $2 \cdot \log_2(N)$  calculations, making this a very efficient data structure.

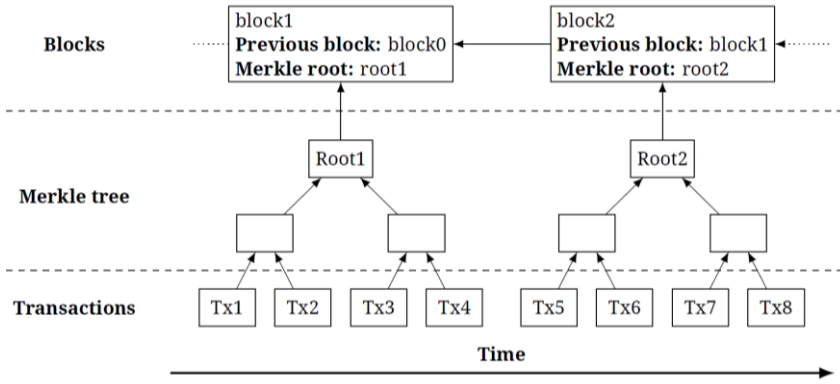


Figure 1.2 – Merkle trees in Bitcoin.

The Merkle tree is constructed bottom-up. In the following example, we start with four transactions, T1, T2, T3 and T4, which form the leaves of the Merkle tree, as shown in Figure 1.3 The transactions are not stored in the Merkle tree; rather, their data is hashed and the resulting hash is stored in each leaf node as  $H_{T1}$ ,  $H_{T2}$ ,  $H_{T3}$  and  $H_{T4}$ :

$$H_{T1} = \text{SHA256}(\text{SHA256}(\text{Transaction T1}))$$

The Merkle tree is constructed as follows:

1. The hash function of included in block transactions is calculated: hash (T1), hash (T2), hash (T3) and so on.

The hash function of the transactions included in the block is calculated: hash (T1), hash (T2), hash (T3) and so on.

2. Consecutive pairs of leaf nodes are then summarized in a parent node, by concatenating the two hashes and hashing them together: hash (hash (T1) + hash (T2)). Because the Merkle tree is a binary tree, it needs an even number of leaf nodes. If there is an odd number of transactions to summarize, the last transaction hash will be duplicated to create an even number of leaf nodes, also known as a *balanced tree*. This is shown in Figure1.4, where transaction T3 is duplicated: hash (hash (T3) + hash (T3)).

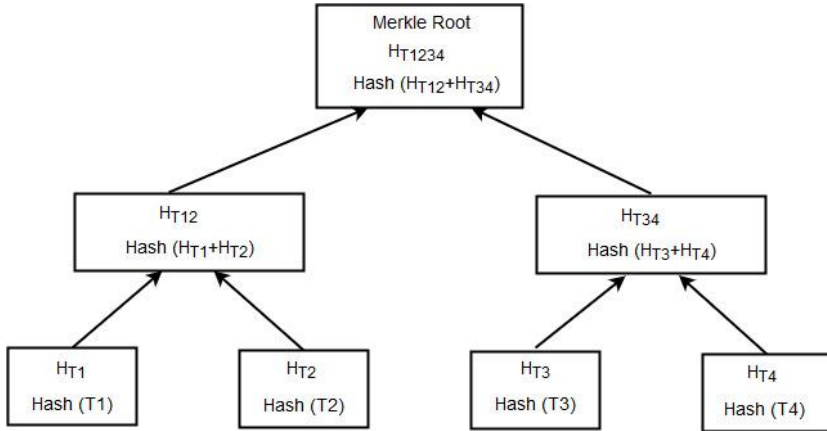


Figure 1.3 – Merkle tree

3. The hash function of the sum of hashes is calculated. The process continues until there is only one node at the top, the node known as the Merkle root. That 32-byte hash is stored in the block header and summarizes all the data in all four transactions.

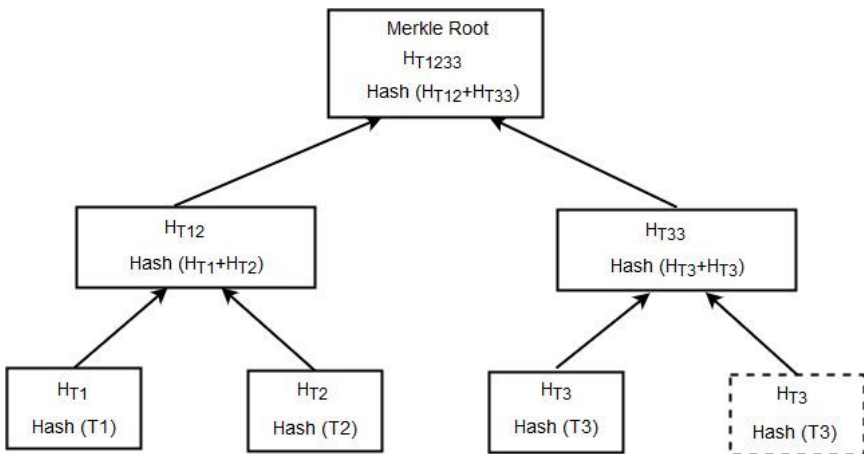


Figure 1.4 – Duplicating one data element achieves an even number of data elements.

To prove that a specific transaction is included in a block, a node only needs to produce  $\log_2(N)$  32-byte hashes, constituting an authentication path or Merkle path connecting the specific transaction to the root of the tree. This is especially important as the number of transactions increases, because the base-2 logarithm of the number of transactions increases much more slowly. This allows bitcoin nodes to efficiently produce paths of 10 or 12 hashes (320–384 bytes), which can provide proof of a single transaction out of more than a thousand transactions in a megabyte-size block.

The efficiency of Merkle trees becomes obvious as the scale increases. Table 1.1 shows the amount of data that needs to be exchanged as a Merkle path to prove that a transaction is part of a block.

Table 1.1 – Merkle tree efficiency

Number of transactions	Approx. size of block	Path size (hashes)	Path size (bytes)
16 transactions	4 kilobytes	4 hashes	128 bytes
512 transactions	128 kilobytes	9 hashes	288 bytes
2048 transactions	512 kilobytes	11 hashes	352 bytes
65535 transactions	16 megabytes	16 hashes	512 bytes

The computational complexity of operations using the Merkle tree is given in Table 2.

Table 1.2 – For a Binary Merkel tree

<b>Operation</b>	<b>Complexity</b>
Space	$O(n)$
Searching	$O(\log n)$
Traversal	$O(n)$
Insertion	$O(\log n)$
Deletion	$O(\log n)$
Synchronization	$O(\log n)$

The recursive implementation of the Merkle tree used in bitcoin in Python is described below [1, 3].

The input of the function is a list of transaction hashes. At each stage of the calculation, consecutive pair of hashes are summarized by concatenating the two hashes and hashing them together with a hash function; if there is an odd number of transactions to summarize, the last transaction hash will be duplicated. The result is only one hash, called the Merkle root.

```
import hashlib
def merkle_root(lst):
#
```

Bitcoin applies twice SHA-256 for hash concatenation and changes the bytes order.

```
    sha256d = lambda x:
hashlib.sha256(hashlib.sha256(x).digest()).digest()
    hash_pair = lambda x, y: sha256d(x[::-1] + y[::-1])[::-1]
    if len(lst) == 1: return lst[0]
    if len(lst) % 2 == 1:
        lst.append(lst[-1])
    return merkle_root([ hash_pair(x, y)
        for x, y in zip(*[iter(lst)] * 2) ])
```

Furthermore, the hash trees are used for verifying the integrity of data in file systems of distributed databases that allows the fast copy synchronization and keys management. Git uses a generalization of hash trees - directed acyclic graphs based on hashes. The hash trees in Blockchain are used for performance increasing since they make it possible the functioning of "light clients" that help users access and interact with a blockchain in a secure and decentralized manner without having to synchronize the full blockchain.

### **Report**

The report should contain: title page with the name of the laboratory work; aim of the work; problem statement according to the task; the progress and results of the study in graphical form; analysis of the results and conclusions.

### **Tasks**

1. Investigate the time complexity of hash functions computing for different IoT platforms (Table 1.3).

2. Construct a Merkle tree with a specified number of transactions, considering the number of transactions is  $(K + 20)$ , where  $K$  is the variant number.

2.1. Generate a specified number of transactions (random characters sequence) for hashing and constructing a tree.

2.2. Compute the hash sum for each transaction.

2.3. Compute the hash sum for all concatenated hashes.

2.4. Investigate an influence of changing of one transaction on Merkle tree.

2.5. Show the minimum number of hashes that need to be checked to prove the presence of the specified transaction.

Table 1.3 – Tasks

<b>№</b>	<b>Device type</b>	<b>Hash function</b>	<b>*k</b>
1	IoT platform 1	SHA1, SHA224, HA256, SHA384, SHA512	4
2	IoT platform 2	SHA1, SHA224, HA256, SHA384, SHA512	6
3	IoT platform 3	SHA1, SHA224, HA256, SHA384, SHA512	8
4	IoT platform 1	SHA1, SHA224, HA256, SHA384, SHA512	1 0
...	....	....	...
...	IoT platform	SHA1, SHA224, HA256, SHA384	n

\*where k-is a number of leading zeroes in hash sum.

### Questions

1. What are advantages of hash functions in the blockchain technology?

2. What is "salt" addition?

3. Explain the "salt" adding for passwords storage.

4. Build a Merkle trees for odd and even number of transactions.

5. Explain how the number of transactions effects on Merkle tree efficiency.



### References

1. A.M. Antonopoulos. Mastering Bitcoin: Unlocking Digital Crypto-Currencies. California, Sebastopol: O'Reilly Media, Inc., 2014.
2. Hashing Strings with Python. <https://www.pythoncentral.io/hashing-strings-with-python/> [July. 12, 2019]
3. Jimmy Song. Programming Bitcoin: Learn How to Program Bitcoin from Scratch. O'Reilly Media: 2019, 322 p.

## Seminar 27.2

### STUDY OF CONSENSUS ALGORITHMS IN IOT SYSTEMS

**The purpose and tasks of the seminar:** acquisition of knowledge and practical skills on the work with consensus algorithms for Internet of Things.

**Training tasks:**

- study of consensus algorithms;
- study of the basic properties of consensus algorithms;
- study of the consensus algorithms comparison criteria.

**Practical tasks:**

- implementation of consensus algorithms in an arbitrary programming language;
- research of the characteristics of consensus algorithms;
- conducting analysis of consensus algorithms.

#### 2.1. Preparation for the seminar:

Preparation for the seminar includes the following steps.

1) *Obtaining (defining) the topic of the paper.*

Themes of the paper may be formulated by students independently based on the following sequence of keywords:

- methods, algorithms, technologies, (blockchain, Internet of things, cryptocurrency);
- characteristics (Consensus method, Accessibility, Mode of operation, Decentralization, Compute-intensive, Network-intensive, Scalability, Throughput, Latency, Immutability, Faulty Replicas Computing Power Stakes Computing Power, Privacy, Smart contract, Currency, Tokens possible);
- design, development, implementation, testing,... ;
- application (cryptocurrency, Smart contract, Internet of things, ).

#### 2.2 Examples of the paper themes:

1. Algorithm Proof-of-work (PoW)
2. Algorithm Proof-of-Stake (PoS)
3. Algorithm Delegated Proof-of-Stake (DPoS)
4. Algorithm Leased Proof-of-Stake (LPoS)

5. Algorithm Proof-of-Capacity (PoC)
6. Algorithm Proof-of-Importance (PoI)
7. Algorithm Proof-of-Activity (PoA)
8. Algorithm Proof-of-Authority (PoAuthority)
9. Algorithm Proof-of-Burn (PoB)
10. Algorithm Ouroboros Proof-of-stake (PoS)
11. Algorithm Proof of Elapsed Time (PoET)
12. Algorithm Practical Byzantine Fault Tolerance (pBFT)
13. Algorithm Delegated Byzantine Fault Tolerance (dBFT)
13. Algorithm Proof-of-Value alternative (PoV)
14. Algorithm Stellar Consensus Protocol (SCP)
15. Tangle protocol
16. Blockchain types. Private Blockchains: Hyperledger Fabric, Hyperledger Sawtooth, Hyperledger Indy
17. Blockchain types. Private Blockchains: Hyperledger Iroha, Hyperledger Burrow
18. Blockchain types. Private Blockchains: IBM Watson IoT
19. Blockchain types. Public Blockchains: Ethereum
20. Blockchain types. Blockchain Alternatives: Corda
21. Blockchain Alternatives: Iota

2) *Search for information on the paper of the abstract and its preliminary analysis:*

Search Engines: <https://scholar.google.com.ua> and etc., electronic libraries: IEEE *Xplore* Digital Library (<https://ieeexplore.ieee.org/Xplore/home.jsp>), Science Direct (<https://www.sciencedirect.com/>), scientific journals, conference materials and official developer sites.

3) *Development of the paper plan and presentation*

The plan of a typical paper includes:

- purpose and main tasks;
- analysis of the issue;
- the main part (the principle of the algorithm, modeling, research results, advantages and disadvantages of the algorithm, examples of application);
- conclusions;
- list of references;
- applications.

#### *4) Writing the paper*

The paper should be 15-20 pages of A4 format (font 14.1,5 in., Field 2 cm), including title page, content, main text, references, applications.

An obligatory appendix to the paper is presentation slides.

#### *5) Preparation of the presentation*

The presentation should be done in PowerPoint and corresponds to the paper plan (8-10 slides for 10 minute talk).

The presentation should include the following slides:

- title slide (university, department, discipline, topic of the report, author, date of presentation);
- content (structure) of the report;
- relevance of the considered issues, the purpose and tasks of the report;
- slides with the disclosure of the main content;
- conclusions;
- references.

The content of the slides should not contain parts of the text from the paper, but includes the keywords, figures, and formulas.

### **The presentation**

The presentation (report) is carried out at the seminar, it takes 15 minutes and includes the report (10 minutes) and discussion (5 minutes). Languages of presentations and papers are Ukrainian or English.

### **Evaluation**

Assessment of this work includes:

- the quality of the text of the paper (form and content),
- presentation quality (content and design);
- the quality of the report (content, logical structure, conclusions);
- distribution of time by sections;
- completeness and correctness of answers.

### **Questions**

1. What is a consensus algorithm?
2. Explain the main consensus algorithms.
3. What are the consensus algorithms requirements for Internet of

thing?

4. Explain the blockchain types, their advantages and disadvantages.
5. The alternatives of blockchain.
6. What are the peculiarities of cryptocurrency for Internet of things - IOTA?
7. What are the main characteristics of consensus algorithms comparison?

### References

1. M.Salimitari, M.Chatterjee. “A Survey on Consensus Protocols in Blockchain for IoT Networks” arXivpreprint <https://arxiv.org/abs/1809.05613v3> [July 29, 2019].
2. “Proof of Stake versus Proof of Work”. White Paper, Internet:<https://bitfury.com/content/downloads/pos-vs-pow-1.0.2.pdf> [Jan. 22, 2019].
3. Implement your first IoT and blockchain project <https://www.ibm.com/internet-of-things/trending/blockchain> [July. 29, 2019]
4. What is IOTA? <https://www.iota.org/> [July. 29, 2019]
5. Jimmy Song. Programming Bitcoin: Learn How to Program Bitcoin from Scratch. O'Reilly Media: 2019, 322 p.

## Laboratory work 27.3

### BLOCKCHAIN-BASED PROTECTION OF VIDEO FILES INTEGRITY

**The aim of the laboratory work** is to develop practical skills for setting up and conducting experiments on the integrity of video files protection using the Blockchain technology and Internet of Things.

Study tasks:

- exploring the structure of the blocks and the principles of their formation;
- study of the basics of the blockchain technology in data integrity protection systems.

Practical tasks:

- skills in work with video files and images using Internet of Things devices;
- skills in building of the blockchain based systems;
- interpretation of the obtained results and recommendations for choosing the most optimal operating modes of the video surveillance system and storage options for the blockchain.

#### **Preparation for laboratory work:**

During the preparation for laboratory work it is necessary:

- to understand the goals and tasks of the work;
- to study the theoretical material given in this manual, as well as in the works [1 - 4];
- to get video files using the Raspberry Pi camera (or another device);
- to make conclusions about the computational complexity of video files processing with different extensions.

#### **1. Theoretical information**

Connect and configure Raspberry Pi.

Before you start, you need to update the system:

```
sudo apt-get update  
sudo apt-get dist-upgrade
```

The camera is connected to Raspberry Pi through a special connector on the board and exchanges data via the CSI interface. The

CSI interface, unlike USB-cameras, requires less computational complexity and allows to use the camera as efficiently as possible.

After connecting the camera to Raspberry Pi, enable camera usage in the Raspberry OS Settings (Preferences ->RP Config -> Interfaces tab -> enable Camera) (Figure 3.1).

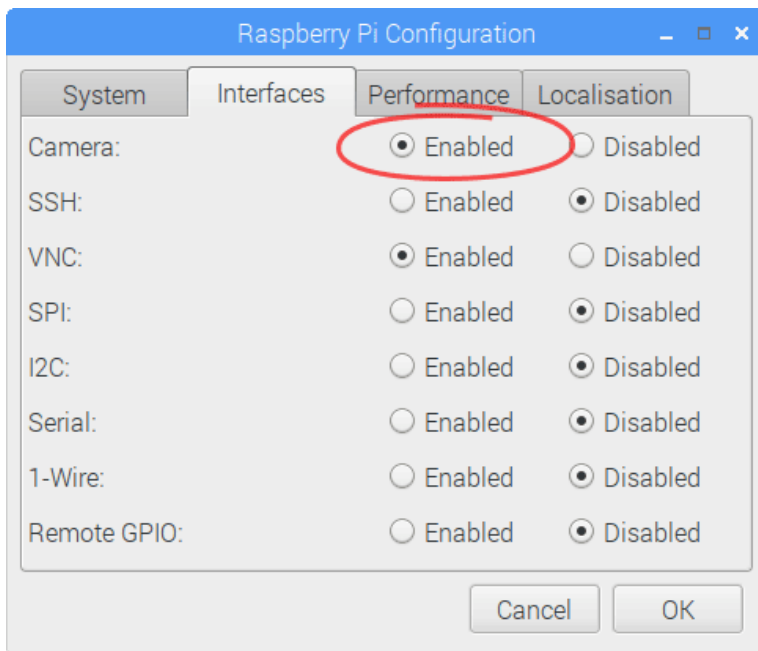


Figure 3.1 – Raspberry Pi Configuration

Once rebooted, the camera is ready to use.

There are following Raspberry Pi camera applications driven from the command line:

raspidvid - is the command line tool for capturing video with the camera module;

raspistill - is the command line tool for capturing photographs with the camera module;

raspiyuv - has the same set of features as raspistill, but instead of outputting standard image files such as .jpg, it generates raw unprocessed image files from the camera.

To create a time-lapse video, you simply configure the Raspberry Pi to take a picture at a regular interval, such as every minute, then use an application to stitch the pictures together into a video.

All applications should run with specified parameters.

For a full list of possible options, run application with no arguments:

```
raspistill
```

or run application with parameter --help:

```
raspistill --help
```

The examples of the use of standard applications:

By default, captures are done at the highest resolution supported by the sensor. This can be changed using the -w and -h command line options. Taking a default capture after two seconds (-t 2000, note times are specified in milliseconds) at resolution 640 × 480 (-w 640 -h 480) on viewfinder, saving in image.jpg:

```
raspistill -t 2000 -o image.jpg -w 640 -h 480 -v
```

Commands -hf and -vfflips the preview and saved image horizontally or vertically correspondingly:

```
raspistill -t 2000 -o image16.jpg -w 640 -h 480 -hf -vf -v
```

Record a 10s clip with default settings (1080p30) and save file video.h264 (-o video.h264):

```
raspivid -t 10000 -o video.h264
```

Time-lapse Video.

The process of Time-lapse video creating consists of 2 stages. At the first stage, a set of images is created with a given interval using the raspistill command at the -o, -t and -tl parameters:

--output -o - output filename <filename>. Specify the output filename. If not specified, no file is saved.

--timeout, -t- time before capture and shut down.

The program will run for this length of time, then take the capture (if output is specified). If not specified, this is set to 5 seconds.

--timelapse, -tl - timelapse mode.

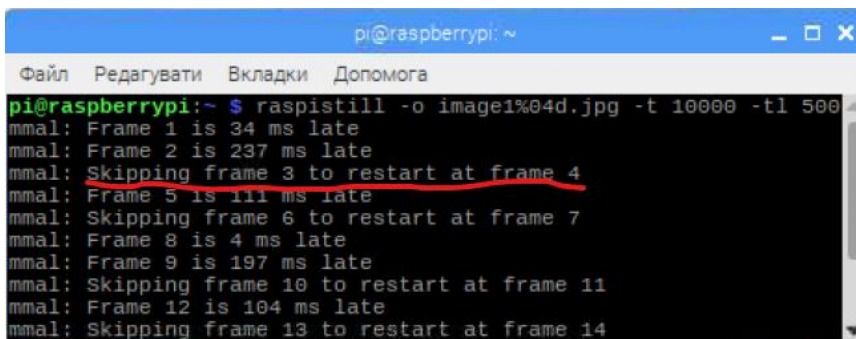


The specific value is the time between shots in milliseconds. Note you should specify %04d at the point in the filename where you want a frame count number to appear. For example:

```
-t 30000 -tl 2000 -o image%04d.jpg
```

will produce a capture every 2 seconds over a total period of 30s, named image1.jpg, image0002.jpg...image0015.jpg. Note that the %04d indicates a four-digit number with leading zero added to pad to the required number of digits. So, for example, %08d would result in an eight-digit number.

If the camera does not capture a shot, it will be skipped, then message about the skipping frame will be generated: "Skipping frame X to restart at frame Y" (Figure 3.2).



```
pi@raspberrypi:~ $ raspistill -o image1%04d.jpg -t 10000 -tl 500
mmal: Frame 1 is 34 ms late
mmal: Frame 2 is 237 ms late
mmal: Skipping frame 3 to restart at frame 4
mmal: Frame 5 is 111 ms late
mmal: Skipping frame 6 to restart at frame 7
mmal: Frame 8 is 4 ms late
mmal: Frame 9 is 197 ms late
mmal: Skipping frame 10 to restart at frame 11
mmal: Frame 12 is 104 ms late
mmal: Skipping frame 13 to restart at frame 14
```

Figure 3.2 – Skipping frame X

In this case it is needed to:

- increase time between shots (frames) (-tl parameter);
- use the -md 1 command that sets a specified sensor mode 1, disabling the automatic selection (possible values depend on the version of the Camera Module being used [5]);
- use the -bm command, that enables burst capture mode;

For creating a video from many images, you will need to use a Linux application called AVConv.

To install AVConv, run the following command:

```
Sudo apt-get install libav-tools
```

AVConv application demands (needs) defined numbering of the files.

In the case when you create a video from images the numbering does not start with 0000 it is necessary to use the in dicessui table for processing by avconv application.

Options:

-y -overwrites output files without asking;

-r n - where n is the number of frames per second, for example -r 10 - 10frames per second.

-i name - is the file that we want to convert, for example- iimage%04d.jpg, where % 4dspecifies to use a decimal number composed of four digits padded with zeroes to express the sequence number.

-vcodec-chooses a video codec to use while processing the conversion, for the codec H.264 it is "libx264": -vcodec libx264 (to show all codecs, you need to use option-codecs).

-q: v uses the option for quality scale '-qscale n', where n - is between 1 (excellent quality) and 31 (worst quality).

-start\_number start - specifies the number of the start frame for starting, for example, image\_0115.jpg, image\_0116.jpg, image\_0117.jpg and etc., then start = 115: -start\_number115

More information you can find on the official page.

avconv -start\_number115 -i image\_% 5d.jpg -r 30 -vcodec libx264 -q: v 3 output.mp4

The usage of ffmpeg.

ffmpeg - is cross-platform open-source project consisting of a vast software suite of libraries and programs for handling video and audio.

Options:

1) To display information about a video file, we can use the command:

ffmpeg -i video.avi

2) Creating video from multiple images

ffmpeg -fimage2 -iimage% d.jpgvideo.mpg

The images of current directory (image1.jpg, image2.jpg, etc.) will be encoded to the video file video.mpg.

```
ffmpeg -r 12 -y -i "image_% 010d.png" output.mpg
```

To set the video frame rate 12 we use the `-r` option before the output file, the number after the `%` sign must be the same as the number of digits in the image filenames, for example `<image_% 010d.png>` provides the same filename length `image_0000000001.png`, `image_0000000002.png` etc.....

3) Video conversion to images  
`ffmpeg -ivideo.mpgimage% d.jpg`

The output directory will contain the files named `image1.jpg`, `image2.jpg`, etc.

The PGM, PPM, PAM, YUV, JPEG, GIF, PNG, TIFF, SGI formats are also supported.

We will build a simple model to protect the integrity of images using javascript to study the basic principles of Blockchain technology. The "Naivechain" will be used as the basis for the blockchain. [2].

We will simulate the project to study and demonstrate the work of Blockchain [3]. You can run the above mentioned project on Windows, Linux, MacOS.

Firstly, you have to install 'node.js', 'npm' and 'curl' on your computer.

## **2. Blockchain simulation includes the following steps:**

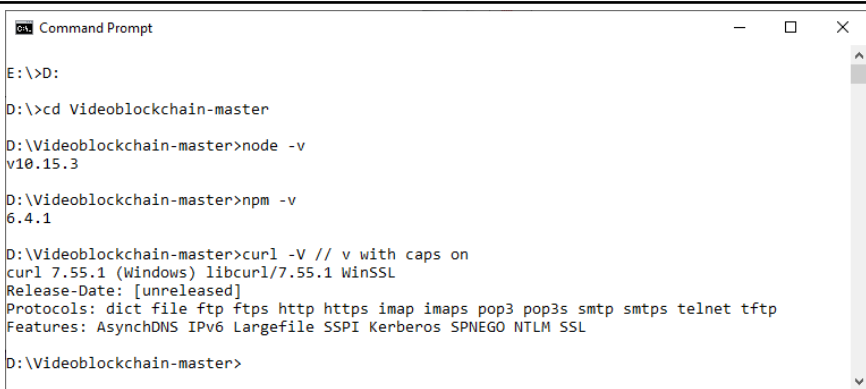
1. Check that git, npm, curl, node are installed. To do this, run the following commands in your terminal or command line to verify that all of the above mentioned components are installed on your computer.

```
node -v
```

```
npm -v
```

```
curl -V // v withcapson.
```

Example:



```

Command Prompt
E:\>D:
D:\>cd Videoblockchain-master
D:\Videoblockchain-master>node -v
v10.15.3
D:\Videoblockchain-master>npm -v
6.4.1
D:\Videoblockchain-master>curl -V // v with caps on
curl 7.55.1 (Windows) libcurl/7.55.1 WinSSL
Release-Date: [unreleased]
Protocols: dict file ftp ftps http https imap imaps pop3 pop3s smtp smtps telnet tftp
Features: AsynchDNS IPv6 Largefile SSPI Kerberos SPNEGO NTLM SSL
D:\Videoblockchain-master>

```

2. Clone the "videoblockchain" from github [3].

2.1 Opengithub profile and select 'Videoblockchain'.

2.2 Click the green button 'Clone or download'.

3. Install the project on a local machine.

3.1 Create a directory for the project 'Videoblockchain' or use any other name.

3.2 Dearchive the downloaded file into created directory.

3.3 Open the terminal.

3.4 Open the project directory: Videoblockchain-master

Enter the command

`npm install`

to get npm packages.

Using the standard editor, open the 'main.js' file and view the code. The description of the code is given in [4].

The node opens two web servers. One is for the user control of the node (HTTP server) and one is for Peer-Peer communication between the nodes (websocket HTTP server).

4. Configure the connected nodes and main unit.

4.1 Set up the first node.

4.2 Return to the terminal, open the videochain directory and enter:

npm install

Save the image file in the directory: .. \ videochain \ myvideo

File name: myvideo.mp4

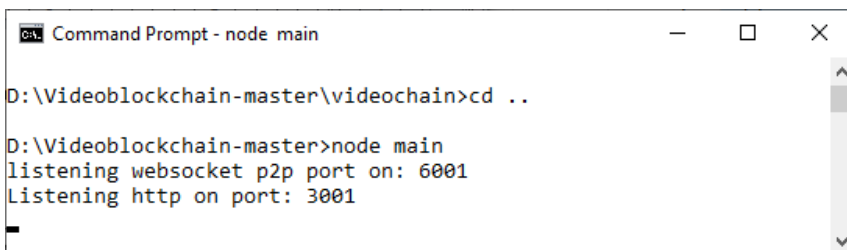
Or specify the file name in the index.js file

Let file Name = 'myvideo.mp4'

In the terminal navigate to the project directory and enter:

node main

Example:



```

Command Prompt - node main
D:\Videoblockchain-master\videochain>cd ..
D:\Videoblockchain-master>node main
listening websocket p2p port on: 6001
Listening http on port: 3001

```

Save in the 'videochain' directory ffprobe.exe, ffmpeg.exe, fnode.exe (for windows OS).

For MacOS, you need to install ffmpeg using the package manager brew [6].

*Launch* a new terminal window and navigate to the 'videochain' directory and enter:

node index

Example:

```

Administrator: Command Prompt
Screenshot is created: 17
Hash function for image has been generated #17: 1c553b59ce38789b6b264685bc182d3c7990ae8fd741ac93838be9454514f55c
Screenshot is created: 18
Hash function for image has been generated #18: f2d2e7b46054f72a0881872c9e6e75af425c281ce0e90c89bf0026dd735
Screenshot is created: 19
Hash function for image has been generated #19: b964d65d632203d1508df084721f658f974d6c47ffb703843d24e2eda88b802a
*All screenshots have been successfully created
=====
*The process of constructing the Merckle tree has started
=====
[ 'a7c606e89a452fc04633c38c1dad0262831034389a0d7f5ed8b9d37b20773663',
  'e5d906d0fb630c10be51b250517933ec60c104e53f365fbd038c7ea652fc6d4f5',
  '96fd33d130cc2af0c163518f6203c30e0712dd35a2c70c4f6d45a2126268ffc5',
  '6cc18c983f94fd28ab289448812d998b757e080be54a1ad966621a9bd8fd13f6',
  'd680fabf7b93a12b9ac2db4d273f3305c93294197e0573d6563dd2f027a027dc',
  '098204d9d18ee552896f2fe12b3865bfda7b11b48f8c8cef938d903e47aaefa',
  '67287cddf5e5042284be8c488ba823e508b90ac47835204296bd4d4e2a7f537b',
  'ebfe76ff077ca4162d2b8b6954111605202cf338e1e867008502dc652bc68e2b',
  '755de432fa2d323a6c2f0d09374a28e4aa62387a15dee2ac78d41ca2be803aa1',
  'b4dcc5b63726917eea72c5d0e92effca99254790ee6de6291a3b4b6b428cf7a0' ]
[ '35a9c1962e8a0e882567fceb35fbbc48d571c52651bec2922e3f113be2543d',
  '00773f8ff4b1499c4847b2f163082699b0c2c8e39ae3fb451c5704a823c0b004',
  'd6ae51c1fcd087ae99e7ef90a9d1a38e9a0afB403e7dd9e7f4427fb4a00ac0b6',
  'c5fb72672e6f90c621f645443Feb966325096cFabc2192168017f68235b7c36f',
  'fb61d1ded0113c19da07f2bf75b0d5fdfe8ffcf055fdcad2fa848f8b20bf' ]
[ '69f2389f3636639ca49cdc92f5ee1f427fb1962e53848d34f9170fc1b676c',
  '5dfe957022dfb907cf3857327a92ff05cc5585eb55ac54f28ea504d483c73c',
  '210850d770db1804f1d5fd8e2ed9937889c18275f2cb7a8536c85acd2acdaf60' ]
[ 'ed2545c70103d73c6d08e447aa59534a0ff5a3dca0a58d2756c430ac79a6724e9f7',
  'c253c865945a36dfc653eb52e649a5f6bacb1da6d7fdcc3cd08de624ea6af',
  '454b4a3e6e2f125bd24664e3242b70ea5d2e98893ff41a1629f3b0e0149daf1c' ]
=====
*The process of storing image
-The '454b4a3e6e2f125bd24664e3242b70ea5d2e98893ff41a1629f3b0e0149daf1c.txt' file is generated
=====
*Recording of the video file root hash in blockchain
-Block is created
=====
*The process of removing screenshots has been started
-All files have been deleted
=====
*The module for video interaction has completed
=====
Start:- 1564388523612 End:- 1564388530094
D:\Videoblockchain-master\videochain>
    
```

After launching the program, the first terminal window will look like.

Example:

```

Command Prompt - node main
Listening http on port: 3001
Received information 454b4a3e6e2f125bd24664e3242b70ea5d2e98893ff41a1629f3b0e0149daf1c
block added: {"index":1,"previousHash":"816534932c2b7154836da6afc367695e6337db8a921823784c14378abed4f7d7","timestamp":1564388530.187,"data":{"454b4a3e6e2f125bd24664e3242b70ea5d2e98893ff41a1629f3b0e0149daf1c","hash":"1749a59c6a28bd4e6f9eb1dfd36d7a74d0073c5cb09b8ea21565b116776771"}}
    
```

## 5. Adding a new node.

Open a new terminal window, navigate to the project folder and enter:

```

set HTTP_PORT=3002
set P2P_PORT=6002
set PEERS=ws://localhost:6001
    
```

npm start

The result of the work is shown in the figure:

Example:

```

D:\Videoblockchain-master>set HTTP_PORT=3002
D:\Videoblockchain-master>set P2P_PORT=6002
D:\Videoblockchain-master>set PEERS=ws://localhost:6001
D:\Videoblockchain-master>npm start

> naivechain@1.0.0 start D:\Videoblockchain-master
> node main.js

listening websocket p2p port on: 6002
Listening http on port: 3002
Received message{"type":0}
Received message{"type":2,"data":[{"index":1,"previousHash":"816534932c2b7154836da6afc3
67695e6337db8a921823784c14378abed4f7d7","timestamp":1564388530.187,"data":{"454b4a3e6e2f
125bd24664e3242b70ea5d2e98893ff41a1629f3b0e0149daf1c","hash":"174a9a59c6a28bed4e6f9eb1dfd
36d7a74d0037c5cb09b8ea21565b116776771"}]}
blockchain possibly behind. We got: 0 Peer got: 1
We can append the received block to our chain
    
```

The second node will listen to signals from other nodes on port 6002 and will listen to commands via the HTTP interface on port 3002.

This node will receive information from the first node through a P2P connection through port 6001.

### 5.1 Adding a new node for MacOS

HTTP\_PORT = 3002 P2P\_PORT = 6002 PEERS = ws://localhost: 6001 npm start

### 6. Adding a new block.

Open the terminal and enter:

curl -H "Content-type: application/json" --data '{"data": "Internet of Things"}' <http://localhost:3001/mineBlock>

Example:

```

D:\Videoblockchain-master>curl -H "Content-type: application/json" --data '{"data": "Internet of Things"}' http://localhost:3001/mineBlock
D:\Videoblockchain-master>
D:\Videoblockchain-master>
D:\Videoblockchain-master>
    
```

The result of adding a block.

## Main node:

### Example:

```

Command Prompt - node main
Received information Internet of Things
block added: {"index":2,"previousHash":"174a9a59c6a28bed4e6f9eb1dfd36d7a74d0037c5cb09b8ea21565b116776771","timestamp":1564389201.385,"data":"Internet of Things ","hash":"dab6782e417c21f9011edf865f9e86e604a72acfd5d290d09a9d79567dcb4d39"}
Received message{"type":2,"data":[{"\index\":2,\previousHash\":"174a9a59c6a28bed4e6f9eb1dfd36d7a74d0037c5cb09b8ea21565b116776771","\timestamp\":1564389201.385,\data\":"Internet of Things \","\hash\":"dab6782e417c21f9011edf865f9e86e604a72acfd5d290d09a9d79567dcb4d39"}]}
received blockchain is not longer than current blockchain. Do nothing
    
```

## Second node.

### Example:

```

npm
listening websocket p2p port on: 6002
Listening http on port: 3002
Received message{"type":0}
Received message{"type":2,"data":[{"\index\":1,\previousHash\":"816534932c2b7154836da6afc367695e6337db8a921823784c14378abed4f7d7","\timestamp\":1564388530.187,\data\":"454b4a3e6e2f125bd24664e3242b70ea5d2e98893ff41a1629f3b0e0149daf1c","\hash\":"174a9a59c6a28bed4e6f9eb1dfd36d7a74d0037c5cb09b8ea21565b116776771"}]}
blockchain possibly behind. We got: 0 Peer got: 1
We can append the received block to our chain
Received message{"type":2,"data":[{"\index\":2,\previousHash\":"174a9a59c6a28bed4e6f9eb1dfd36d7a74d0037c5cb09b8ea21565b116776771","\timestamp\":1564389201.385,\data\":"Internet of Things \","\hash\":"dab6782e417c21f9011edf865f9e86e604a72acfd5d290d09a9d79567dcb4d39"}]}
blockchain possibly behind. We got: 1 Peer got: 2
We can append the received block to our chain
    
```

You can also view the results of the execution in the web browser:

### Example:

```

[{"index":0,"previousHash":"0","timestamp":1465154705,"data":"my genesis block!!","hash":"816534932c2b7154836da6afc367695e6337db8a921823784c14378abed4f7d7"}, {"index":1,"previousHash":"816534932c2b7154836da6afc367695e6337db8a921823784c14378abed4f7d7"}, {"index":2,"previousHash":"174a9a59c6a28bed4e6f9eb1dfd36d7a74d0037c5cb09b8ea21565b116776771"}, {"index":3,"previousHash":"dab6782e417c21f9011edf865f9e86e604a72acfd5d290d09a9d79567dcb4d39"}]
    
```



## Report

The report should contain: title page with the name of the laboratory work; aim of the work; problem statement according to the task; the progress and results of the study in graphical form; analysis of the results and conclusions.

## Tasks

1. Record a video file with camera and Raspberry Pi (or get a video file from the teacher).
2. Convert the video file into the frames (images) with FFmpeg or another video converter.
3. Calculate the hash of each frame using SHA-256.
4. Construct a Merkle tree of calculated hashes.
5. Include the Merkle root in current block of the blockchain.
6. Modify any random frame.
7. Repeat paragraphs 3-5.
8. Compare the headers of the first and second Merkle trees.
9. Find a modified frame.
10. Carry out a simulation of the project of video files protection using the project [4] in accordance with the recommendations contained in the theoretical basics.

## Questions

1. Explain the methods for the file integrity protection, their advantages and disadvantages.
2. What are the benefits of the blockchain technology use for the files integrity protection?
3. Explain the elements of the block header in blockchain.
4. How to calculate previous Hash?
5. Explain the command-line utilities for work with the camera on Raspberry Pi.
6. What is ffmpeg?

## References

1. A.M. Antonopoulos. Mastering Bitcoin: Unlocking Digital Crypto-Currencies. California, Sebastopol: O'Reilly Media, Inc., 2014.
2. A blockchain implementation in 200 lines of code <https://github.com/lhartikk/naivechain>. [July. 29, 2019]

3. Proof of Video Integrity Based on Blockchain <https://github.com/vy22/Videoblockchain>. [July. 29, 2019]
4. A-blockchain-in-200-lines-of-code-963cc1cc0e54 <https://medium.com/@lhartikk/> [July. 29, 2019]
5. Raspberry Pi Camera Module <https://www.raspberrypi.org/documentation/raspbian/applications/camera.md> [July. 29, 2019]
6. Homebrew. The missing package manager for macOS (or Linux) <https://brew.sh> [July. 29, 2019]
7. V. Yatskiv, N. Yatskiv, O. Bandrivskyi. “Proof of Video Integrity Based on Blockchain”, in Proc. Advanced Computer Information Technologies (ACIT), 2019 IEEE9th International Conference on, 2019, pp. 431-434.

## APPENDIX A

### TEACHING PROGRAMME OF THE COURSE PC3 “DEPENDABILITY AND SECURITY OF IOT”

#### DESCRIPTION OF THE COURSE

TITLE OF THE COURSE	Code
<b>Dependability and Security of IoT</b>	<b>PC3</b>

Teacher(s)	Department
<b>Coordinating:</b> Prof., DrS. V. V. Sklyar <b>Others:</b> Modules PCM3.4: DrS. V. V. Yatskiv, Ass. Prof., Dr. N. G. Yatskiv (TNEU)	Computer Systems, Networks and Cybersecurity Department (KhAI) Cybersecurity Department (TNEU)

Study cycle	Level of the course	Type of the course
PhD	A	Bounden

Form of delivery	Duration	Language(s)
Full-time tuition	One semester	English

Prerequisites	
<b>Prerequisites:</b> Foundation of Modelling; Computer Systems and System Analysis, Computer Networks, Reliability Theory	<b>Co-requisites (if necessary):</b>

Credits of the course	Total student workload	Contact hours	Individual work hours
4	120	64	56

Aim of the course: competences foreseen by the study program
The aim of the course is to give understanding principles of operation the dependable, safe and secure platforms and tools for IoT application. Relevant knowledge is based on a study of the features and methods of information models formation for IoT-based devices and technologies for safety and security. Obtained knowledge will allow choosing the means and technologies for the development and implementation of dependable and secure IoT-based systems.

Appendix A. Teaching programme of the course PC4

<b>Learning outcomes of course</b>	<b>Teaching/learning methods</b>	<b>Assessment methods</b>
At the end of course, the successful student will be able to: 1. perform modelling of IoT systems dependability, safety and security	Interactive lectures, Learning in laboratories, Just-in-Time Teaching	Course Evaluation Questionnaire
2. perform quantitative and qualitative analysis of measure of IoT systems dependability, safety and security	Interactive lectures, Learning in laboratories	Course Evaluation Questionnaire
3. apply methods of assurance of IoT systems dependability, safety and security	Interactive lectures, Learning in laboratories	Course Evaluation Questionnaire
4. develop safety and security management plan	Interactive lectures, Learning in laboratories	Course Evaluation Questionnaire
5. develop structure of safety and security life cycle	Interactive lectures, Just-in-Time Teaching	Course Evaluation Questionnaire
6. perform forward and backward requirements tracing	Interactive lectures, Learning in laboratories	Course Evaluation Questionnaire
7. recognize review, analysis and testing techniques	Interactive lectures, Learning in laboratories	Course Evaluation Questionnaire
8. analyse safety and security management documents	Interactive lectures, Learning in laboratories, Just-in-Time Teaching	Course Evaluation Questionnaire
9. define Assurance Case methodology for IoT safety and security assessment	Interactive lectures, Learning in laboratories	Course Evaluation Questionnaire
10. use software tools for development of Assurance Case	Interactive lectures, Just-in-Time Teaching	Testing based on alternative method of assessment
11. apply methods of assurance of IoT systems safety and security	Interactive lectures, Just-in-Time Teaching	Course Evaluation Questionnaire

Appendix A. Teaching programme of the course PC4

12. understand how Blockchain technology can solve business problems	Interactive lectures, Just-in-Time Teaching	Course Evaluation Questionnaire
13. Blockchain usage in the IoT projects	Interactive lectures, Just-in-Time Teaching	Testing based on alternative method of assessment
14. development of blockchain-based project for IoT protection.	Interactive lectures, Just-in-Time Teaching	Course Evaluation Questionnaire
15. Blockchain-based video file integrity protection.	Interactive lectures, Just-in-Time Teaching	Course Evaluation Questionnaire

Themes	Contact work hours							Time and tasks for individual work	
	Lectures	Consultations	Seminars	Practical work (tutorials)	Laboratory work	Placements	Total contact work	Individual work	Tasks
1. Dependability and security concepts for IoT. 1.1. Taxonomy of safety and security requirements. 1.2. Dependability, safety and security attributes taxonomy. 1.3. Risk analysis fundamentals.	2		2				4	6	1.5 Hardware and software of Arduino platform. 1.6 Control logic implementation on the base of Arduino platform.
2. Dependability and safety models of IoT. 2.1. Reference architectures of Industrial IoT. 2.2. Dependability and safety measures. 2.3. Failure Mode, Effect and Criticality Analysis	2				4		6	4	2.5 IoT systems for safety and security critical applications.

Appendix A. Teaching programme of the course PC4

(FMECA) of IoT systems.									
3. Security models for IoT. 3.1. IoT systems architectures from security outlook. 3.2. Security measures. 3.3. Threats and attacks modeling for IoT systems.	2				4		6	4	3.5. Tools for dependability and security modelling.
4. Safety management requirements to IoT. 4.1. Safety & security management plan. 4.2. Human resource management. 4.3. Configuration management. 4.4. Tools selection and evaluation. 4.5. Documentation management. 4.6. Safety & security assessment.	2				4		6	4	4.7 Project management.
5. Safety and security life cycle for IoT. 5.1. Overall life cycle. 5.2. Safety & security life cycle: design top-down brunch. 5.3. Safety & security life cycle: integration down-top brunch. 5.4. Requirements tracing.	2				4		6	4	5.5. Terms used in requirements engineering.
6. Review, analysis and testing techniques for IoT. 6.1. Documents review. 6.2. Static code analysis. 6.3. Functional testing. 6.4. Code structural testing.	2		2				4	6	6.7. Penetration testing. 6.8. Terms used in software testing.
7. Assurance Case fundamentals. 7.1. Assurance Case	2				4		6	4	1.5. Software tools for Assurance

Appendix A. Teaching programme of the course PC4

concept and history. 7.2. Standards for Assurance Case.								Case development.
8. Safety and security techniques and measures for IoT. 8.1. Claims, Arguments and Evidence (CAE) notation. 8.2. Update and application of Claims, Arguments and Evidence (CAE) notation 8.3. Goal Structuring Notation (GSN).	2			4		6	4	8.4. Techniques and measures overview. 8.5. Techniques directed to attacks avoidance.
9. Security informed and energy efficiency informed Assurance Case for IoT 9.1. Tools for development of Assurance Case. 9.2. Assurance Case structure for IoT systems.	2		2			4	6	9.3. Measurement of energy consumption for IoT device layer.
10. Bases of blockchain technology and examples of application 10.1 The principle of the blockchain technology 10.2 Block structure and Merkle tree 10.3 Blockchain cryptography	2			4		6	4	10.4. Hash Function 10.5. Generating Hash function with specified parameters and study the time complexity
11. Consensus algorithms in blockchain technology 11.1 Proof-of-work algorithm 11.2 Proof of Stake algorithms 11.3 Blockchain technology for the IoT security	2		2			4	6	11.4. Research of consensus algorithm for IoT. 11.5. Comparison of consensus algorithm usage for IoT systems

Appendix A. Teaching programme of the course PC4

12. Blockchain technology for the IoT security 12.1 Blockchain and the IoT 12.2 Benefits of Integrating Blockchain with IoT 12.3 Main challenges of blockchain in IoT 12.4 Blockchain-based the IoT security solutions	2				4		6	4	12.4. Blockchain based IoT security. 12.5. Blockchain development and simulation for data integrity protection
<b>On the whole</b>	<b>24</b>		<b>8</b>		<b>32</b>			<b>56</b>	

Assessment strategy	Weight in %	Deadlines	Assessment criteria
Lecture activity, including fulfilling special self-tasks	10	7,14	85% – 100% Outstanding work, showing a full grasp of all the questions answered. 70% – 84% Perfect or near perfect answers to a high proportion of the questions answered. There should be a thorough understanding and appreciation of the material. 60% – 69% A very good knowledge of much of the important material, possibly excellent in places, but with a limited account of some significant topics. 50% – 59% There should be a good grasp of several important topics, but with only a limited understanding or ability in places. There may be significant omissions. 45% – 49% Students will show some relevant knowledge of some of the issues involved, but with a good grasp of only a minority of the material. Some topics may be answered well, but others will be either omitted or incorrect. 40% – 44% There should be some work of some merit. There may be a few topics answered partly or there may be scattered or perfunctory knowledge across a larger range.



Appendix A. Teaching programme of the course PC4

			<p>20% – 39% There should be substantial deficiencies, or no answers, across large parts of the topics set, but with a little relevant and correct material in places.</p> <p>0% – 19% Very little or nothing that is correct and relevant.</p>
Learning in laboratories	30	7,14	<p>85% – 100% An outstanding piece of work, superbly organized and presented, excellent achievement of the objectives, evidence of original thought.</p> <p>70% – 84% Students will show a thorough understanding and appreciation of the material, producing work without significant error or omission. Objectives achieved well. Excellent organization and presentation.</p> <p>60% – 69% Students will show a clear understanding of the issues involved and the work should be well written and well organized. Good work towards the objectives.</p> <p>The exercise should show evidence that the student has thought about the topic and has not simply reproduced standard solutions or arguments.</p> <p>50% – 59% The work should show evidence that the student has a reasonable understanding of the basic material. There may be some signs of weakness, but overall the grasp of the topic should be sound. The presentation and organization should be reasonably clear, and the objectives should at least be partially achieved.</p> <p>45% – 49% Students will show some appreciation of the issues involved. The exercise will indicate a basic understanding of the topic, but will not have gone beyond this, and there may well be signs of confusion about more complex material. There should be fair work towards the laboratory work objectives.</p>

Appendix A. Teaching programme of the course PC4

			<p>40% – 44% There should be some work towards the laboratory work objectives, but significant issues are likely to be neglected, and there will be little or no appreciation of the complexity of the problem.</p> <p>20% – 39% The work may contain some correct and relevant material, but most issues are neglected or are covered incorrectly. There should be some signs of appreciation of the laboratory work requirements.</p> <p>0% – 19% Very little or nothing that is correct and relevant and no real appreciation of the laboratory work requirements.</p>
Course Evaluation Quest	60	8,16	The score corresponds to the percentage of correct answers to the test questions

Author	Year of issue	Title	No of periodical or volume	Place of printing. Printing house or internet link
<b>Compulsory literature</b>				
N. Leveson	2011	Engineering a Safer World: Systems Thinking Applied to Safety		The MIT Press
N. Mead, J. Allen, S. Barnum, R. Ellison, G. McGraw	2008	Software Security Engineering		Addison-Wesley Professional
A. Sajid, H. Abbas, K. Saleem	2016	Cloud-Assisted IoT-Based SCADA Systems Security: A Review of the State of the Art and Future Challenges	Volume 4	IEEE Access

Appendix A. Teaching programme of the course PC4

R. Natella, D. Cotroneo, H. Madeira	2016	Assessing Dependability with Software Fault Injection: A Survey	Volume 48, Issue 3	ACM Computing Surveys
P. Kuber, B. Russell, S. Sundaram	2016	Future-proofing the Connected World: 13 Steps to Developing Secure IoT Products		<a href="http://www.researchandmarkets.com/blog/csas">http://www.researchandmarkets.com/blog/csas</a>
	2015	ISA/IEC 62443, Security for Industrial Automation and Control Systems		International Society of Automation (ISA)
	2013	ABB Safety Handbook. Machine Safety – Jokab Safety products		<a href="https://library.e.abb.com/public/6a43569b705c4d8f94e1809f4a620088/ABB_Jokab%20Safety_katEN13.pdf">https://library.e.abb.com/public/6a43569b705c4d8f94e1809f4a620088/ABB_Jokab%20Safety_katEN13.pdf</a>
	2013	A Guide to the Project Management Body of Knowledge (PMBOK Guide), Fifth Edition		Project Management Institute
	2014	Syllabus: REQB Certified Professional for Requirements Engineering. Foundation Level, Version 2.1		<a href="http://reqb.org/index.php/download/syllabi">http://reqb.org/index.php/download/syllabi</a>
	2014	Standard		<a href="http://reqb.org/i">http://reqb.org/i</a>

Appendix A. Teaching programme of the course PC4

		glossary of terms used in Requirements Engineering, Version 1.3		<a href="http://www.istqb.org/download/glossary">ndex.php/download/glossary</a>
	2011	Certified Tester Foundation Level Syllabus		<a href="http://www.istqb.org/download/syllabi.html">http://www.istqb.org/download/syllabi.html</a>
	2014	Standard glossary of terms used in Software Testing, Version 2.3		<a href="http://www.istqb.org/download/s/glossary.html">http://www.istqb.org/download/s/glossary.html</a>
	2010	IEC 61508, Functional safety of electrical/electronic/programmable electronic safety-related systems		International Electrotechnical Commission
	2011	Goal Structuring Notation (GSN) Community Standard		<a href="http://www.goalstructuringnotation.info/">http://www.goalstructuringnotation.info/</a>
	2015	Structured Assurance Case Metamodel (SACM)		<a href="http://www.omg.org/spec/SACM/">http://www.omg.org/spec/SACM/</a>
Christidis Konstantinos, Michael Devetsikiotis.	2016	Blockchains and Smart Contracts for the Internet of Things	<i>IEEE Access</i> 4 (2016): 2292-2303.	<a href="http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7467408">http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7467408</a>
Dorri Ali, Salil S. Kanhere, Raja Jurdak	2016	Blockchain in internet of things: Challenges and Solutions.	<i>arXiv preprint arXiv:1608.05187</i> (2016).	<a href="https://arxiv.org/ftp/arxiv/papers/1608/1608.05187.pdf">https://arxiv.org/ftp/arxiv/papers/1608/1608.05187.pdf</a>

Appendix A. Teaching programme of the course PC4

Bahga, Arshdeep, Vijay K. Madiseti	2016	Blockchain Platform for Industrial Internet of Things.	<i>Journal of Software Engineering and Applications</i> 9.10 (2016): 533.	<a href="http://file.scirp.org/pdf/JSEA_2016102814012798.pdf">http://file.scirp.org/pdf/JSEA_2016102814012798.pdf</a>
Lee Boohyung, Jong-Hyouk Lee	2016	Blockchain-based secure firmware update for embedded devices in an Internet of Things environment."	<i>The Journal of Supercomputing</i> (2016): 1-16.	DOI: 10.1007/s11227-016-1870-0
Antonopoulos Andreas M.	2014	Mastering Bitcoin: unlocking digital cryptocurrencies.	O'Reilly Media, Inc.", 2014.	<a href="https://books.google.com.ua/books?hl=uk&amp;lr=&amp;id=IXmrBQAAQBAJ&amp;oi=fnd&amp;pg=PR4&amp;dq=Mastering+Bitcoin&amp;ots=9AgWjqJpPW&amp;sig=zn4j0Hp-K14_3IL9msTgMDiPGQc&amp;redir_esc=y#v=onepage&amp;q=Mastering%20Bitcoin&amp;f=false">https://books.google.com.ua/books?hl=uk&amp;lr=&amp;id=IXmrBQAAQBAJ&amp;oi=fnd&amp;pg=PR4&amp;dq=Mastering+Bitcoin&amp;ots=9AgWjqJpPW&amp;sig=zn4j0Hp-K14_3IL9msTgMDiPGQc&amp;redir_esc=y#v=onepage&amp;q=Mastering%20Bitcoin&amp;f=false</a>
N.G.Yatskiv, S.V.Yatskiv	2016	Perspectives of the Usage of Blockchain Technology in the Internet of Things	The Scientific Bulletin of UNFU, vol. 26, n.8, pp. 381-387, 2016. (In Ukrainian)	<a href="http://nltu.edu.ua/nv/Archive/2016/26_8/59.pdf">http://nltu.edu.ua/nv/Archive/2016/26_8/59.pdf</a>
V.Yatskiv, N.Yatskiv, O. Bandrivskiyi.		Proof of Video Integrity Based on Blockchain	<i>Proc. Advanced Computer</i>	<a href="https://ieeexplore.ieee.org/document/8780097">https://ieeexplore.ieee.org/document/8780097</a>

Appendix A. Teaching programme of the course PC4

			<i>Information Technologies (ACIT), 2019 IEEE 9th International Conference on, 2019, pp. 431-434.</i>	
V. Sklyar	2017	Functional Safety of Computer Control Systems (series of paper)		<a href="http://vvslyar.blogspot.in/">http://vvslyar.blogspot.in/</a>
<b>Additional literature</b>				
Ю.Н. Федоров	2008	Справочник инженера по АСУ ТП: Проектирование и разработка		Инфра-Инженерия, Москва
В.В. Склад	2018	Обеспечение безопасности АСУТП в соответствии с современными стандартами		Инфра-Инженерия, Москва
K. Lueth, C. Patsioura, Z. Williams, J. Rickert	2017	IoT Analytics (electronic resource of IoT market analysts)		<a href="https://iot-analytics.com/">https://iot-analytics.com/</a>
D. Serodon	2017	Eurosmart (electronic resource of European IoT engineers)		<a href="http://www.eurosmart.com/">http://www.eurosmart.com/</a>
	2017	Habrahabr (electronic resource of IT-engineers)		<a href="https://habrahabr.ru/hub/iot_dev/">https://habrahabr.ru/hub/iot_dev/</a>
	2017	Association of Ukrainian Enterprises of		<a href="http://appau.org.ua/">http://appau.org.ua/</a>

Appendix A. Teaching programme of the course PC4

		Industrial Automation (electronic resource of Ukrainian automation engineers)		
Li Shancang, Li Da Xu, Shanshan Zhao	2015	The internet of things: a survey.	<i>Information Systems Frontiers</i> 17.2 (2015): 243-259.	<a href="http://www.istp.ethz.ch/content/dam/ethz/special-interest/gess/cis/international-relations-dam/Teaching/bridging/Shancang%202014.pdf">http://www.istp.ethz.ch/content/dam/ethz/special-interest/gess/cis/international-relations-dam/Teaching/bridging/Shancang%202014.pdf</a>
Zheng Zibin et al.	2016	"Blockchain Challenges and Opportunities: A Survey." (2016).		<a href="http://inpluslab.sysu.edu.cn/files/blockchain/blockchain.pdf">http://inpluslab.sysu.edu.cn/files/blockchain/blockchain.pdf</a>
Ouaddah A., Abou Elkalam A., Ait Ouahman A.	2016	FairAccess: a new Blockchain-based access control framework for the Internet of Things.	Security Comm. Networks, 9: 5943–5964.	doi: 10.1002/sec.1748
Dorri Ali et al.	2017	Blockchain for IoT Security and Privacy: The Case Study of a Smart Home.	IEEE Percom workshop on security privacy and trust in the internet of thing. IEEE. 2017.	<a href="https://www.researchgate.net/profile/Ali_Dorri5/publication/312218574_Blockchain_for_IoT_Security_and_Privacy_The_Case_Study_of_a_Smart_Home/links/5877309508ae329d6226e96f/Blockchain-for-">https://www.researchgate.net/profile/Ali_Dorri5/publication/312218574_Blockchain_for_IoT_Security_and_Privacy_The_Case_Study_of_a_Smart_Home/links/5877309508ae329d6226e96f/Blockchain-for-</a>

Appendix A. Teaching programme of the course PC4

---

---

				<a href="#"><u>IoT-Security-and-Privacy-The-Case-Study-of-a-Smart-Home.pdf</u></a>
--	--	--	--	--



АНОТАЦІЯ

УДК 004.415/.416.052.056(076.5)=111

Скляр В.В., Яцків В.В., Яцків Н.Г. Гарантоздатність та безпека систем інтернету речей: Практикум / За ред. Харченка В.С. та Скляра В.В. – МОН України, Національний аерокосмічний університет ім. М. Є. Жуковського «ХАІ». – 98 с.

Викладено матеріали практичної частини курсу РС4 “Розробка та впровадження IoT систем”, підготовленого в рамках проекту ERASMUS+ ALIOT “ Internet of Things: Emerging Curriculum for Industry and Human Applications” (573818-EPP-1-2016-1-UK-EPPKA2-SBHE-JP).

Наведена структура робіт з перевірки знань з курсу, відповідний практичний матеріал, приклади виконання завдань та критерії оцінювання. В процесі навчання наводяться теоретичні аспекти розробки та впровадження надійних та безпечних IoT систем. Вивчаються основні концепції та підходи до розробки і імплементації IoT систем, моделі для надійності та безпеки IoT пристроїв та систем.

Призначено для інженерів, розробників та науковців, які займаються розробкою та впровадженням IoT систем, для аспірантів університетів, які навчаються за напрямками IoT, кібербезпеки в мережах, комп'ютерних наук, комп'ютерної та програмної інженерії, а також для викладачів відповідних курсів.

Бібл. – 106, рисунків – 48, таблиць – 20.

---

ABSTRACT

UDC 004.415/.416.052.056(076.5)=111

Sklyar V.V., Yatskiv V.V., Yatskiv N.G. Dependability and Security of IoT: Practicum / Kharchenko V.S. and Sklyar V.V. (Eds.) – Ministry of Education and Science of Ukraine, National Aerospace University “KhAI”, Ternopil National Economic University, 2019. – 98 p.

The materials of the practical part of the study course PC3 “Dependability and Security of IoT”, developed in the framework of the ERASMUS+ ALIOT project “Internet of Things: Emerging Curriculum for Industry and Human Applications” (573818-EPP-1-2016-1-UK-EPPKA2-CBHE-JP).

The structure of work on verification of residual knowledge in the discipline, the corresponding practical material, examples of tasks and criteria of evaluation are given. In the learning process, the theoretical aspects of dependability and security of IoT-based systems are presented. The basic concepts and approaches to development and implementation of dependable, safe and secure IoT systems, models and methods for dependability and security assurance and assessment of IoT-based systems are examined.

It is intended for engineers, developers and scientists engaged in the development and implementation of IoT-based systems, for postgraduate students of universities studying in areas of IoT, cybersecurity in networks, computer science, computer and software engineering, as well as for teachers of relevant courses.

Ref. – 33 items, figures – 14, tables – 7.

## ЗМІСТ

СКОРОЧЕННЯ	3
ВСТУП	4
1. КОНЦЕПЦІЇ ТА ПІДХОДИ ДО РОЗРОБКИ ТА ІМПЛЕМЕНТАЦІЇ ІОТ СИСТЕМ	6
1.1. СЕМІНАР 1. ВИВЧЕННЯ ПРОГРАМНО-АПАРАТНИХ ПРОДУКТІВ, ЯКІ СЕРТИФІКОВАНО ЗА ВИМОГАМИ ДО ФУНКЦІОНАЛЬНОЇ БЕЗПЕКИ	6
1.2. ЛАБОРАТОРНА РОБОТА 1. ОЦІНЮВАННЯ ПОКАЗНИКІВ БЕЗПЕКИ	9
1.3. ЛАБОРАТОРНА РОБОТА 2. АНАЛІЗ СЦЕНАРІЇВ АТАК НА ІОТ СИСТЕМИ	15
2. УПРАВЛІННЯ ФУНКЦІОНАЛЬНОЮ ТА ІНФОРМАЦІЙНОЮ БЕЗПЕКОЮ ІОТ СИСТЕМ	20
2.1. ЛАБОРАТОРНА РОБОТА 1. РОЗРОБКА ПЛАНУ УПРАВЛІННЯ ФУНКЦІОНАЛЬНОЮ ТА ІНФОРМАЦІЙНОЮ БЕЗПЕКОЮ ІОТ СИСТЕМ	20
2.2. ЛАБОРАТОРНА РОБОТА 1. РОЗРОБКА ЖИТТЄВОГО ЦИКЛУ ФУНКЦІОНАЛЬНОЇ ТА ІНФОРМАЦІЙНОЇ БЕЗПЕКИ ІОТ СИСТЕМ	27
2.3. СЕМІНАР 1. ВИВЧЕННЯ МЕТОДІВ ВЕРИФІКАЦІЇ ТА ВАЛІДАЦІЇ ІОТ СИСТЕМ	32
3. ASSURANCE CASE ДЛЯ ІОТ СИСТЕМ	35
3.1. ЛАБОРАТОРНА РОБОТА 1. ПРОГРАМНІ ЗАСОБИ ДЛЯ РОЗРОБКИ ASSURANCE CASE	35
3.2. ЛАБОРАТОРНА РОБОТА 2. МЕТОДИ ТА ЗАСОБИ ЗАБЕЗПЕЧЕННЯ ФУНКЦІОНАЛЬНОЇ ТА ІНФОРМАЦІЙНОЇ БЕЗПЕКИ ІОТ СИСТЕМ	42
3.3. СЕМІНАР 1. ВИВЧЕННЯ «ЗЕЛЕНОГО» ASSURANCE CASE ДЛЯ ІОТ СИСТЕМ	48
4. ІНФОРМАЦІЙНА БЕЗПЕКА ТЕХНОЛОГІЙ БЛОКЧЕЙН, ЯКІ БАЗУЮТЬСЯ НА ІОТ	51
4.1. ЛАБОРАТОРНА РОБОТА 1. ГЕНЕРУВАННЯ ТА ВИВЧЕННЯ ХЕШ-ФУНКЦІЙ ДЛЯ ТЕХНОЛОГІЙ БЛОКЧЕЙН	51
4.2. СЕМІНАР 1. ВИВЧЕННЯ АЛГОРИТМІВ УЗГОДЖЕННЯ ДЛЯ ІОТ СИСТЕМ	63
4.3. ЛАБОРАТОРНА РОБОТА 2. ЗАХИСТ ІНТЕГРОВАНОСТІ ВІДЕО ФАЙЛІВ З ВИКОРИСТАННЯМ ТЕХНОЛОГІЙ БЛОКЧЕЙН	67
ДОДАТОК А. НАВЧАЛЬНА ПРОГРАМА КУРСУ РС4	80
АНОТАЦІЯ ТА ЗМІСТ	94

---

CONTENTS

ABBREVIATIONS	3
INTRODUCTION	4
1. DEPENDABILITY AND SECURITY MODELS OF IOT	6
1.1. SEMINAR 1. STUDY OF HARDWARE-SOFTWARE PRODUCTS CERTIFIED AGAINST FUNCTIONAL SAFETY REQUIREMENTS	6
1.2. LABORATORY WORK 1. ASSESSMENT OF SAFETY INDICATORS	9
1.3. LABORATORY WORK 2. ANALYSIS OF IOT ATTACKS SCENARIOS	15
2. SAFETY AND SECURITY MANAGEMENT OF IOT	20
2.1. LABORATORY WORK 1. DEVELOPMENT OF IOT SAFETY AND SECURITY MANAGEMENT PLAN	20
2.2. LABORATORY WORK 1. DEVELOPMENT OF IOT SAFETY AND SECURITY LIFE CYCLE	27
2.3. SEMINAR 1. STUDY OF IOT VERIFICATION AND VALIDATION METHODS	32
3. ASSURANCE CASE FOR IOT	35
3.1. LABORATORY WORK 1. SOFTWARE TOOLS FOR IOT ASSURANCE CASE DEVELOPMENT	35
3.2. LABORATORY WORK 2. SAFETY AND SECURITY TECHNIQUES AND MEASURES FOR IOT SYSTEMS	42
3.3. SEMINAR 1. STUDY OF GREEN ASSURANCE CASE FOR IOT	48
4. SECURITY OF IOT BASED BLOCKCHAIN TECHNOLOGY	51
4.1. LABORATORY WORK 1. THE GENERATION AND RESEARCH OF HASH FUNCTION FOR BLOCKCHAIN TECHNOLOGY	51
4.2. SEMINAR 1. STUDY OF CONSENSUS ALGORITHMS IN INTERNET OF THINGS SYSTEMS	63
4.3. LABORATORY WORK 2. SAFETY AND SECURITY TECHNIQUES AND MEASURES FOR IOT SYSTEMS	67
APPENDIX A. TEACHING PROGRAMME OF THE COURSE PC3	80
ABSTRACT AND CONTENTS	94

Скляр Володимир Володимирович  
Яцків Василь Васильович  
Яцків Наталія Георгіївна

# ГАРАНТОЗДАТНІСТЬ ТА БЕЗПЕКА СИСТЕМ ІНТЕРНЕТУ РЕЧЕЙ

**Практикум**  
(англійською мовою)

Редактори Харченко В.С. та Скляр В.В.

Комп'ютерна верстка  
В.С. Харченко,  
О.О. Ілляшенко

Зв. план, 2019  
Підписаний до друку 22.08.2019  
Формат 60x84 1/16. Папір офс. №2. Офс. друк.  
Умов. друк. арк. 5,69. Уч.-вид. л. 6,12. Наклад 150 прим.  
Замовлення 220819-5.

---

Національний аерокосмічний університет ім. М. Є. Жуковського  
"Харківський авіаційний інститут"  
61070, Харків-70, вул. Чкалова, 17  
<http://www.khai.edu>

Випускаючий редактор: ФОП Голембовська О.О.  
03049, Київ, Повітрофлотський пр-кт, б. 3, к. 32.  
Свідоцтво про внесення суб'єкта видавничої справи до державного реєстру видавців,  
виготовлювачів і розповсюджувачів видавничої продукції  
серія ДК № 5120 від 08.06.2016 р.

Видавець: ТОВ «Видавництво «Юстон»  
01034, м. Київ, вул.. О. Гончара, 36-а, тел.: +38 044 360 22 66  
[www.yuston.com.ua](http://www.yuston.com.ua)  
Свідоцтво про внесення суб'єкта видавничої справи до державного реєстру видавців,  
виготовлювачів і розповсюджувачів видавничої продукції  
серія ДК № 497 від 09.09.2015 р.