МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Кваліфікаційна наукова праця на правах рукопису

**КОМАРНИЦЬКИЙ ОЛЕГ ОЛЕКСАНДРОВИЧ**

УДК 004.771

# ДИСЕРТАЦІЯ
# МЕТОДИ ТА МОДЕЛІ ВДОСКОНАЛЕННЯ ТРАНСПАРЕНТНОЇ ТЕХНОЛОГІЇ ТАЄМНОГО ІНТЕРНЕТ-ГОЛОСУВАННЯ

05.13.06 «Інформаційні технології»

Подається на здобуття наукового ступеня кандидата технічних наук

Дисертація містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ О.О. Комарницький

Науковий керівник - доктор технічних наук,
професор Мачалін Ігор Олексійович

Київ – 2021

# АНОТАЦІЯ

*Комарницький О.О.* Методи та моделі вдосконалення транспарентної технології таємного Інтернет-голосування. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня кандидата технічних наук (доктора філософії) за спеціальністю 05.13.06 «Інформаційні технології». – Національний авіаційний університет МОН України, Київ, 2021.

Дисертаційна робота присвячена проблемі забезпечення довіри громадян до систем дистанційного таємного голосування через Інтернет (ДТГ), а також скороченню витрат часу виборців на процедуру волевиявлення.

Удосконалення спрямовані на підвищення рівня автоматизації трудомістких процедур та надання виборцям можливості аудиту системи ДТГ з метою набуття впевненості у відсутності зловмисного втручання у роботу сервера голосування, яке може призвести до розкриття таємниці голосів та фальсифікації підрахунку, що усуває причини для недовіри до чесності роботи системи ДТГ.

Розроблено моделі, протоколи та технології інформаційної взаємодії елементів системи ДТГ, що забезпечують автоматизацію пошуку виборцями *IP* адрес їхніх виборчих дільниць та адаптивне управління пульсуючим потоком запитів виборців до серверного обладнання системи ДТГ з метою уникнення його перенавантаження.

Розроблено технологію дистанційної автентифікації особи виборця з урахуванням обмежень, що пов’язані із забезпеченням транспарентності системи ДТГ.

**Ключові слова:** дистанційне таємне голосування, довіра до систем голосування, транспарентність, адаптивне управління потоком запитів, контрольованість програмно-апаратного засобів, дистанційна автентифікація.

## Список публікацій здобувача

1. В.М.Чуприн, В.М.Вишняков, О.О.Комарницький, «Метод протидії атакам посередника у транспарентній системі інтернет голосування», Захист інформації, Ukrainian Information Security Research Journal. - К.: НАУ, 2019. – Т.20. - №2. – С.172-182. http://jrnl.nau.edu.ua/index.php/ZI/article/view/13079. *Здобувачу належить розробка основного принципу, що покладений в основу розробленого методу.*

2. І.О.Мачалін, В.М.Вишняков, О.О.Комарницький, «Технологія автентифікації виборців у відкритій системі інтернет-голосування», Науково-технічний журнал «РАДИОЭЛЕКТРОНИКА И ИНФОРМАТИКА». - № 2(81), апрель – июнь 2018. - С. 55-63. *Здобувачу належить розробка основних ідей, що лежать в основі розробленої технології автентифікації.*

3. І.О.Мачалін, О.О.Комарницький, В.О.Гнатюк, «Удосконалення технології доступу до ресурсів транспарентних систем Інтернет-голосування, Науковий журнал «Наукоємні технології». - № 4 (40), 2018. - С. 415 – 423. *Здобувач розробив схему та протокол інформаційної взаємодії елементів серверного обладнання.*

4. В.М.Чуприн, В.О.Антонов, О.О.Комарницький, «Метод розподілу навантаження між серверами системи інтернет-голосування», Захист інформації Ukrainian Information Security Research Jornal. – К.:НАУ, 2019. – Т. 21. - №1, - С. 25-34. *Здобувачу належить ідея застосування адаптивного регулювання потоком звернень виборців із застосування методу динамічного програмування.*

5. В.М.Вишняков, О.О.Комарницький, А.О.Жуковський, «Методи контролю керування системою Інтернет голосування», Управління розвитком складних систем. – 2019. - № 38 – С. 37-44. *Здобувачу належить розробка методу виявлення позаштатних проникнень до сервера виборчої дільниці під час його функціонування.*

6. С.В.Цюцюра, О.О.Комарницький, «Застосування новітніх інформаційних технологій в Україні», Сучасні наукові дослідження та розробки:

теоретична цінність та практичні результати: тези Міжнародної науково-практичної конференції (14-19 березня 2016 р., Братислава), 2016, С. 155-156.

7. О.О.Комарницький, Г.Б.Нестерук, «Захист систем дистанційного опитування від атак посередника», Матеріали 5-ої міжнародної науково-практичної конференції «Management of the development of technologies», Секція "Information technology development of education» ( Kyiv, 30 – 31 March, 2018), Київ, С.76.

8. П.В.Ворона, В.М.Вишняков, О.О.Комарницький, Д.Ю.Хлапонін, «Принципи побудови транспарентних систем таємного електронного голосування», Науково-практична конференція до Дня місцевого самоврядування "Форум прямої демократії", 4 грудня 2018 р.: тези доп. – К., 2018. – С.169-171.

9. О.О.Комарницький, Д.Ю.Хлапонін, «Системи таємного електронного голосування як елемент цифрової демократії», IX Міжнародна науково-практична конференція молодих вчених «Інформаційні технології: економіка, техніка, освіта», 14-15 листопада 2018 р.: тези доп. – К., 2018. – С.117-118.

10. Вышняков В.М., Комарницкий О.А. Транспарентные системы электронной демократии. Accent Graphics Communications & Publishing, Оттава, Канада, 2019, 96 с.

http://doi.org/10.29013/VyshnyakovVM.KomarnickiyOA.TSED.2019.228

11. О.О. Комарницький Особливості забезпечення безпеки інформації в системах електронної демократії. // Матеріали V науково-практичної конференції «Перспективні напрямки захисту інформації» ОНАЗ ім. О.С.Попова, тези доп. Одеса, 2019. - С. 23 – 25.

12. D.I.Bakhtiiarov, O.Y.Lavrynenko, N.O.Lishchynovska, O.O. Komarnytskyi, (2020) Methods of evaluation and forecasting of levels of electromagnetic radiation in urban environments (in Ukrainian) Actual issues of modern science. No. 1. Vol. 2, add. Collection of Scientific Articles, 06-2 (06), __-__. European Scientific e-Journal. Hlučín-Bobrovníky: “Anisiia Tomanek” OSVČ. http://tuculart.eu/articles/inn/

13. В.М. Вышняков, О.А. Комарницкий, И.А. Мачалин Разрешение проблемы доверия к системам электронного голосования. «Colloquium-journal»

Wydrukowano w «Chocimska 24, 00-001 Warszawa, Poland» №29 (81), 2020 Ч.1 С.44-50. http://www.colloquium-journal.org/

# ABSTRACT

**Komarnitskiy O.O. Methods and models for improving the transparent technology of secret Internet voting.** –Manuscript.

Dissertation on the receipt of scientific degree of candidate of engineering sciences after speciality 05.13.06 are "Information technologies". - National aviation university, Kyiv, 2021.

The dissertation is devoted to the problem of ensuring citizens' trust in remote secret ballot systems via the Internet (RSB), as well as reducing the time spent by voters on the procedure of expression of will .

Improvements are aimed at increasing the level of automation of time-consuming procedures and giving voters the opportunity to audit the RSB system to ensure that there is no malicious interference with the voting server, which could lead to disclosure of votes and falsification of counting, which eliminates reasons for distrust of the RSB system.

Models, protocols and technologies of information interaction of RSB system elements have been developed, which provide automation of voters 'search of IP addresses of their polling stations and adaptive control of pulsating flow of voters' requests to RSB system server equipment in order to avoid its overload.

The technology of remote authentication of the voter's person has been developed, taking into account the limitations related to ensuring the transparency of the RSB system.

**Keywords:** remote secret ballot, trust in voting systems, transparency, adaptive control of the flow of requests, controllability of software and hardware, remote authentication.

# ЗМІСТ

**ПЕРЕЛІК СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ …..…….** 10
**ВСТУП ……..**12
**РОЗДІЛ 1. АНАЛІЗ ХАРАКТЕРИСТИК ІСНУЮЧИХ ТЕХНОЛОГІЙ ДИСТАНЦІЙНОГО ТАЄМНОГО ГОЛОСУВАННЯ ……..** 20
1.1. Фундаментальні принципи функціонування систем дистанційного таємного голосування у демократичних країнах ……. 20
1.2. Аналіз існуючих систем дистанційного таємного голосування …… 22
1.3. Логічна модель транспарентної системи дистанційного голосування …. 24
1.4. Аналіз характеристик існуючого прототипу, що обраний для удосконалення транспарентної системи голосування …….. 33
1.4.1. Модель загроз інформації у прототипі……..…….. 33
1.4.2. Функціональний профіль захищеності інформації у прототипі……. 34
1.4.3. Характеристики середовища функціонування прототипу…….. 36
1.4.4. Характеристики ОС сервера виборчої дільниці …….. 36
1.4.5. Характеристики засобів програмування …….. 37
1.4.6. Характеристики засобів забезпечення контрольованості ОС …… 37
1.4.7. Забезпечення відкритості та контрольованості прикладного ПЗ …….. 38
1.4.8. Забезпечення відкритості технології обробки інформації …… 39
1.5. Недоліки прототипу відкритої досконало стійкої системи дистанційного таємного голосування, постановка завдань досліджень ……. 40
Висновки до першого розділу ……..…… 44
**РОЗДІЛ 2. УДОСКОНАЛЕННЯ ТЕХНОЛОГІЇ ДОСТУПУ ВИБОРЦІВ ДО РЕСУРСІВ СИСТЕМИ ІНТЕРНЕТ-ГОЛОСУВАННЯ ……** 46
2.1. Задачі удосконалення технології дистанційного доступу виборців до серверного обладнання системи інтернет-голосування …… 46
2.2. Прискорення доступу виборців до ресурсів системи ДТГ …….. 48

2.3. Задача поділу потоку звернень виборців між серверами пошуку адрес … 52
2.4. Характеристики процесу авторегулювання …….. 55
2.5. Характеристики процесу вирівнювання керованих змінних …….. 57
2.6. Синтез регулятора системи вирівнювання керованих змінних …….. 58
2.7. Схема рішення задачі побудови регулятора …….. 63
2.8. Варіант рішення з побудови регулятора ……. 67
Висновки до другого розділу ……..…… 70
**РОЗДІЛ 3. УДОСКОНАЛЕННЯ МЕТОДІВ ЗАХИСТУ СИСТЕМИ ДИСТАНЦІЙНОГО ТАЄМНОГО ІНТЕРНЕТ-ГОЛОСУВАННЯ ……. 73**
3.1. Характеристики атаки *MITM* та можливості протидії цій атаці …….. 73
3.2. Вдосконалення методу протидії атаці *MITM* ……. ..80
3.3. Особливості технології дистанційної автентифікації виборців ……..… 86
3.4. Вдосконалення технології дистанційної автентифікації виборців у системі ДТГ …..…….…… …….. 88
3.4.1. Визначення періоду уведення автентифікаційних даних……... 88
3.4.2. Технологія маніпулювання автентифікаційними даними у системі з очною перевіркою осіб виборців…… 90
3.4.3. Технологія маніпулювання автентифікаційними даними у системі без очної перевірки осіб виборців……..... 92
Висновки до третього розділу ……..…… 94
**РОЗДІЛ 4. ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ СИСТЕМИ ДИСТАНЦІЙНОГО ТАЄМНОГО ГОЛОСУВАННЯ ……..** 96
4.1. Завдання експериментального дослідження ……..…….. 96
4.2. Вибір програмно-апаратних засобів ……..……..…….. 97
4.3. Ініціалізація серверного обладнання ……..…….. 99
4.4. Перевірка справжності ОС ……..……...……..105
4.5. Перевірка справжності апаратних засобів ……. 108
4.6. Перевірка дій персоналу щодо адміністрування сервера …….. 110
4.7. Перевірка процесу функціонування сервера виборчої дільниці …….. 115
Висновки до четвертого розділу ……...…… 122

**ОСНОВНІ РЕЗУЛЬТАТИ ТА ВИСНОВКИ** ……..… 124

**СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ** ……. 128

**ДОДАТОК А. ВИХІДНІ ТЕКСТИ ПРОГРАМ ЕКСПЕРИМЕНТАЛЬНОГО ДОСЛІДЖЕННЯ СИСТЕМИ ДИСТАНЦІЙНОГО ТАЄМНОГО ГОЛОСУВАННЯ** ……..… 140

Д1.1. Текст головної програми для управління проведенням експериментального дослідження системи ……... 140

Д1.2. Текст програми для експериментального дослідження роботи сервера виборчої дільниці ……..… 142

**ДОДАТОК Б. АКТ ВПРОВАДЖЕННЯ РЕЗУЛЬТАТІВ РОБОТИ** …..... 157

# ПЕРЕЛІК СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ

ВД – виборча дільниця;

ДВП – дозвіл на введення паролю для голосування;

ДТГ – дистанційне таємне голосування через Інтернет;

ЗДТГ – загальнонаціональне дистанційне таємне голосування;

ДРВ – державний реєстр виборців;

НСД – загроза несанкціонованого доступу до інформації;

ПБ – послуга, що забезпечує безпеку інформації;

ПЗ – програмне забезпечення;

ППЗ – прикладне програмне забезпечення;

ППД – період підготовки даних про виборців;

ППС – період підготовки сервера виборчої дільниці;

ПВЕБ – період введення електронних бюлетенів;

ПЗВ – перенаправлення звернень виборців;

ОС – операційна система;

СДВ – система дистанційного волевиявлення;

ТЗІ – технічний захист інформації;

ФПЗ – функціональний профіль захисту;

*ID* – ідентифікатор виборця;

ЦВК – центральна виборча комісія;

*MITM* – атака посередника (*Man In The Middle*);

A – множина дій адміністратора сервера виборчої дільниці;

V – множина дій голосуючого виборця;

K – множина дій контролера сервера виборчої дільниці;

$N$ – вектор керованих змінних;

$n$ – кількість керованих змінних;

$P_0$ – імовірність обслуговування запиту без очікування у черзі;

$p$ – імовірність події у відсотках;

$T$ – час обслуговування виборця, с.

$v$ – кількість сеансів зв’язку під час обслуговування одного виборця;

$w$ – кількість пауз між сеансами під час обслуговування одного виборця;

$\lambda$ – середнє значення інтенсивності потоку запитів.

# ВСТУП

***Актуальність теми***. Згідно розпорядженню Кабінету Міністрів України від 8 листопада 2017 року № 797 «Про схвалення Концепції розвитку електронної демократії в Україні та плану заходів щодо її реалізації» передбачено впровадження електронного голосування, а також електронних референдумів та електронних плебісцитів, що повинні базуватися на принципі підвищення довіри громадян до інструментів електронної демократії.

Особливої актуальності набули технології електронного голосування у 2020 році через появу вірусу COVID-19. З метою збереження здоров'я і життя людей та забезпечення розвитку наукової діяльності в умовах карантину Кабінет Міністрів України 15 липня 2020 р. прийняв Постанову № 607 «Про внесення змін до Порядку присудження наукових ступенів», яка дозволила вченим радам проводити засідання у режимі on-line з використанням засобів таємного дистанційного голосування. У відповідності до цієї постанови методи, що запропоновані у даній роботі, були з успіхом використані в умовах карантину для дистанційного голосування на засіданнях Вченої Ради Київського національного університету будівництва і архітектури 16 жовтня та 30 листопада 2020 року, а також з безпосередньою участю автора 4 грудня 2020 року з 19 до 20 години було проведено дистанційне таємне Інтернет голосування для обрання керівних органів Товариства Червоного Хреста України, де виборці голосували з різних областей України не покидаючи своїх міст.

Технології дистанційного таємного голосування через Інтернет (надалі - ДТГ) мають ряд беззаперечних переваг у порівнянні із традиційними технологіями голосування, котрі потребують обов'язкової присутності виборців під час голосування на виборчих дільницях. Із публікацій витікає, що у сучасних умовах основною перешкодою на шляху впровадження ДТГ є недовіра виборців щодо відсутності шахрайства з боку персоналу, який обслуговує засоби дистанційного голосування. Єдиним ефективним методом досягнення довіри людей є надання їм можливості аудиту усіх тих програмно-апаратних засобів

ДТГ, які можуть викликати недовіру, а саме це є засоби, від яких залежить збереження таємниці голосів та чесність підрахунку. Вирішенню цих питань, а також скороченню витрат часу виборців на процедури дистанційного волевиявлення, присвячена дана робота.

**Мета роботи -** зменшення сумарних витрат часу виборців на процедури, що пов’язані з дистанційним голосуванням за рахунок автоматизації довготривалих процедур та підвищення довіри виборців щодо збереження таємниці їх голосів і відсутності шахрайства у підрахунку шляхом побудови автоматизованої системи безперервного аудиту виборцями програмно-апаратних засобів та процесів, які виконують розшифровку та підрахунок голосів.

Для досягнення цієї мети слід розробити методи удосконалення технологій ДТГ, що дозволяють виборцям не витрачати час на пошук *IP* адрес їхніх виборчих дільниць шляхом автоматизації процесу такого пошуку, усунути необхідність фізичної присутності виборців на виборчих дільницях в період уточнення списків шляхом створення відповідної технології дистанційної автентифікації, мінімізувати можливі затримки в обслуговуванні звернень виборців шляхом використання оптимізованого адаптивного управління непрогнозованим потоком цих звернень та нейтралізувати можливі атаки посередника шляхом створення відповідного методу захисту інформації. Розроблювані методи мають не порушувати прозорість функціонування системи ДТГ та досконалість захисту її інформаційних ресурсів. За кінцевим рахунком, досягнення поставленої мети дозволить створити на практиці транспарентну систему ДТГ і, таким чином, усунути проблему недовіри виборців щодо можливостей організації «чесного» волевиявлення.

**Задачі дослідження**

1. Здійснити аналіз існуючих технологій ДТГ з точки зору забезпечення довіри виборців до програмно-апаратних засобів та їх технічної підтримки, а також витрат часу виборців на процедуру волевиявлення. Визначити ті характеристики, які негативно впливають на транспарентність програмно-

апаратних засобів ДТГ та на якість обслуговування виборців. Визначити та обґрунтувати методи, що спрямовані на усунення виявлених недоліків.

2. Вирішити наукові завдання, що спрямовані на скорочення витрат часу на пошук *IP* адрес серверів дистанційного голосування шляхом розробки *структурно-функціональної моделі автоматизації пошуку цих адрес* та відповідного протоколу інформаційної взаємодії програмно-апаратних засобів, що реалізують цей метод. У рамках створеної моделі автоматизації *розробити метод балансування* (вирівнювання) навантаження на сервери пошуку *IP* адрес.

3. Вирішити наукові завдання щодо забезпечення транспарентності системи ДТГ шляхом розробки *моделі аудиту виборцями програмно-апаратних засобів розшифрування та підрахунку голосів виборців*, а також *розробити метод дистанційної автентифікації виборців* у системі ДТГ з тим, щоб усунути необхідність здійснення очних перевірок виборців перед кожним голосуванням.

4. Створити програмно-апаратне середовище для реалізації вдосконаленої технології ДТГ. Упевнитись, що воно здатне забезпечити вимоги транспарентності та задовольняє вимогам щодо часу обслуговування запитів виборців. *Оцінити показники якості функціонування* цього середовища за різних умов використання.

**Об’єктом дослідження** є процес дистанційного таємного голосування з використанням мережі Інтернет.

**Предметом дослідження** є методи, моделі та засоби, що спрямовані на забезпечення транспарентності технології ДТГ і зменшення витрат часу виборців на процедури, що пов’язані з дистанційним голосуванням.

**Методи дослідження.** Адаптивне управління потоками запитів до серверного обладнання розроблено з використанням результатів теорії аналітичного конструювання регуляторів з урахуванням необхідності забезпечення сталості та дотримання показників якості перехідних процесів регулювання потоками запитів. Статистичні параметри створеної системи ДТГ оцінювались з використанням результатів теорії інформації та телетрафіку. Виявлення «слабких місць» у захисті систем ДТГ здійснено з використанням

методів побудови комплексних систем захисту, що знайшли своє відображення у чинних нормативних документах з ТЗІ. Розробка методів забезпечення гарантованої конфіденційності та цілісності даних, що передаються каналами зв'язку, а також технології дистанційної автентифікації заснована на теорії криптографічних систем, у т.ч. теорії секретного зв'язку К. Шеннона. Програмне забезпечення удосконаленої системи ДТГ створено з використанням мови програмування *JavaScript*.

**Наукова новизна одержаних результатів**

1. Вперше розроблено *структурно-функціональну модель автоматизованого пошуку* виборцями *IP*-адрес серверів виборчих дільниць шляхом синтезу структури функціональних елементів цієї моделі та розробки протоколу інформаційної взаємодії між цими елементами, що дозволяє зменшити витрати часу на здійснення актів волевиявлення в умовах зростання кількості виборчих дільниць.

2. Вперше розроблено *метод балансування (вирівнювання) навантаження* на одночасно працюючі сервери, що входять до складу лінійки серверів пошуку *IP*-адрес. В основу методу покладено результати синтезу адаптивного регулятора розподілу потоку запитів виборців між серверами шляхом його зведення до відомої формально вирішеної крайової задачі аналітичного конструювання регуляторів на мінімізацію функціонала Р.Беллмана у класі неперервних динамічних систем регулювання щодо об'єктів, що описуються звичайними лінійними диференціальними рівняннями настроювання першого порядку, що забезпечує сталий режим вирівнювання значень коефіцієнтів завантаження серверів, тим самим запобігаючи можливим перенавантаженням в роботі серверів в умовах непередбачуваних пульсацій трафіку, за рахунок чого підвищується якість технічної підтримки процесу волевиявлення.

3. Дістав подальший розвиток *метод дистанційної автентифікації виборців* у транспарентній системі ДТГ, котрий за рахунок спеціалізованих серверів, що містять бази даних з біологічними або іншими унікальними

ознаками виборців, дозволяє уникнути обов'язкової очної перевірки перед кожним актом голосування.

4. Вперше розроблено *модель безперервного аудиту виборцями програмно-апаратних засобів сервера голосування* за рахунок використання відкритого для перевірки монтажу міні комп'ютерів та автоматизації процедур аудиту за допомогою спеціалізованого сервера, який підключено до сервера голосування через спільну локальну мережу, а доступ виборців до нього реалізовано через захищений канал, де центр сертифікації *HTTPS* обирають представники виборців, при цьому інсталяція та запуск серверів виконується під наглядом виборців або їх довірених осіб у період часу, коли на серверах ще немає ніякої критичної інформації, а після запуску серверів виборці продовжують аудит дистанційно без втрати інформації про наявність чи відсутність втручань у роботу серверів, бо усі спроби таких втручань виявляються та реєструються сервером аудиту, що забезпечується спеціально розробленим програмним забезпеченням та відкритими для виборців правилами адміністрування і реєстрацією кодів з'єднань з виборцями на сервері голосування, що дозволяє позбавити виборців підозри про те, що сервер голосування являє собою «чорний ящик» з імітатором який демонструє виборцям нібито чесне голосування, а насправді розкриває і підмінює їхні голоси, бо така підозра руйнує довіру виборців, а також завдяки розробленій моделі виборці можуть самостійно у будь-який момент часу виявляти атаку посередника, яка є найнебезпечнішою загрозою для транспарентних систем ДТГ.

**Практичне значення одержаних результатів**

1. Використання вдосконаленої транспарентної технології ДТГ, що реалізована на основі розроблених методів та моделей, надала можливість кожному виборцю під час голосування контролювати наявність загроз, які можуть призвести до порушення таємниці голосів та вірності результатів волевиявлення, що усуває причини для недовіри громадян до запропонованої системи ДТГ.

2. Створено та апробовано у реальних умовах програмно-апаратне середовище з відповідною технічною документацією (лістінги програм, специфікації апаратних засобів, інструкції користувачам та адміністраторам), яке може бути використано для побудови транспарентних систем ДТГ будь-якої розмірності у будь-яких сферах людської активності із заявленою в роботі функціональністю та якістю технічної підтримки процесу волевиявлення.

3. Розроблені специфікації протоколу взаємодії елементів системи ДТГ та відповідного програмного забезпечення (ПЗ) використано для створення інтерфейсів транспарентних систем ДТГ. Позитивною особливістю цих інтерфейсів у порівнянні з існуючими є те, що користування ними дозволяє виборцям впевнитись у відсутності загроз щодо порушення таємниці голосів та підробки результатів волевиявлення. Розроблені специфікації ПЗ підсистеми захисту інформації рекомендується застосовувати для виявлення атак посередника. Розроблені специфікації засобів автентифікації рекомендується використовувати для дистанційної автентифікації виборців з тим, щоб усунути необхідність проходження виборцями обов'язкової очної перевірки перед кожним голосуванням.

4. Результати роботи впровадженні у ДНДІАСБ, НАУ, НТУУ КПІ та КНУБА, де протягом останніх двох років регулярно проводяться вибори до органів студентського самоврядування. З жовтня 2020 року за допомогою запропонованої у даній роботі системи ДТГ проводяться голосування на засіданнях Вченої Ради КНУБА, а також з грудня 2020 року проводиться дистанційне таємне Інтернет голосування для обрання керівних органів Товариства Червоного Хреста України, де виборці голосують з різних областей України не покидаючи своїх міст.

***Особистий внесок здобувача***

Основні положення і результати дисертаційної роботи, отримані автором самостійно, обмежуються обсягом тих результатів наукової діяльності, які відображені у цій роботі. Із робіт, що опубліковані у співавторстві, у дисертаційній роботі використовуються результати, що отримані особисто

здобувачем. (Творчий вклад здобувача у роботах із співавторами відображено у розділі «ПУБЛІКАЦІЇ ЗА ТЕМОЮ ДИСЕРТАЦІЇ»).

***Зв'язок роботи з науковими та учбовими програмами, планами, темами***

Висвітлені в дисертації наукові результати отримано, здебільшого, в рамках науково-дослідної роботи, яка була виконана Київським національним університетом будівництва і архітектури (КНУБА) на замовлення Державного НДІ автоматизованих систем у будівництві (ДНДІАСБ), що здійснює діяльність у сфері створення комп'ютерних систем для потреб будівельної галузі, та „Укртелеком" (Договори про НДР №165-ХО4 (№ держ. реєстрації 0113U000093), №513-9-931, №514-9-931, №401/03). Автор дисертації був виконавцем цих робіт. Отримані результати використовуються у навчальному процесі КНУБА при викладанні навчальної дисципліни «Комп'ютерні мережі та захист даних», у НТУУ КПІ на основі монографії автора з вересня 2020 року розпочато вивчення дисципліни «Протоколи та алгоритми електронного голосування», а також ідеї автора враховані у НАУ при формуванні змісту навчальної програми «Стратегії обслуговування телекомунікаційних мереж».

***Апробація результатів дисертації***

Результати досліджень дисертаційної роботи доповідались, обговорювались і отримали позитивну оцінку на:

1. Міжнародна науково-технічна конференція «Сучасні наукові дослідження та розробки: теоретична цінність та практичні результати» (м.Братіслава,14-19 березня 2016 р.).

2. 5-та міжнародна науково-практична конференція «Management of the development of technologies», Секція "Information technology development of education» ( Kyiv, 30 – 31 March, 2018).

3. Науково-практична конференція до Дня місцевого самоврядування "Форум прямої демократії" (Київ, 4 грудня 2018 р.).

4. IX Міжнародна науково-практична конференція студентів, аспірантів та молодих вчених «Інформаційні технології: економіка, техніка, освіта» (Київ, 14-

15 листопада 2018 р.).

5. Науково-практична конференція КНУБА (м. Київ, 2019).

6. V Всеукраїнська науково-практична конференція «Перспективні напрямки захисту інформації» ОНАЗ ім. О.С.Попова (м. Одеса, 2019)

***Достовірність результатів.*** Для перевірки достовірності одержаних результатів проведено повне натурне моделювання і випробування усіх програмно-технічних засобів в умовах реальних опитувань серед студентів провідних вищих навчальних закладів України.

***Публікації.*** За результатами виконаних досліджень опубліковано 13 наукових робіт, з яких 6 робіт у фахових науково-технічних спеціалізованих виданнях, одна монографія та 6 тез доповідей на науково-технічних конференціях.

***Структура та обсяг роботи.*** Дисертаційна робота складається зі вступу, чотирьох розділів, висновків по кожному розділу та загальних висновків по роботі в цілому, списку використаних літературних джерел (122 найменувань), 2 додатки. Повний обсяг дисертації - 147 сторінок (без анотації), у тому числі 135 сторінки основного тексту, 29 рисунків, 12 таблиць.

# РОЗДІЛ 1

# ВИЯВЛЕННЯ НЕДОЛІКІВ ІСНУЮЧИХ ТЕХНОЛОГІЙ ДИСТАНЦІЙНОГО ТАЄМНОГО ГОЛОСУВАННЯ. ОБГРУНТУВАННЯ ЗАВДАНЬ ДОСЛІДЖЕНЬ

## 1.1. Принципи функціонування систем дистанційного голосування

Об’єктом даного дослідження є процеси дистанційного таємного голосування (волевиявлення) (ДТГ) з використанням мережі Інтернет, під час проведення державних виборів, референдумів та плебісцитів, а також і різного роду конкурсів, опитувань, експертних оцінювань тощо [1]. Під голосуванням розуміється акт прийняття рішення групою осіб шляхом вибору $k$ варіантів із $n$ можливих, де $k < n$, в умовах збереження таємниці вибору кожного із членів групи, а процес голосування може породжувати проблемну ситуацію – недовіру до результатів волевиявлення. ДТГ у даній роботі розглядається як процес дистанційного визначення колективної думки щодо конкретних питань з використанням мережі Інтернет для виконання акту волевиявлення. Технічно це забезпечується програмно-апаратними засобами ДТГ, зокрема у системах загальнонаціонального дистанційного таємного голосування (ЗДТГ), які застосовуються для вирішення політичних задач.

Дистанційне голосування – зручний для виборців спосіб волевиявлення з будь-якого місця, де є доступ до мережі Інтернет. Особливо за умов відсутності обмежень щодо типу термінального пристрою, а саме коли для здійснення волевиявлення може бути використаний комп’ютер довільного типу, смартфон, планшет або телевізор з приставкою *TV BOX* чи функцією *SmartTV*. Через зручність користування системи ДТГ (у т.ч., ЗДТГ) набувають усе більш широкого застосування.

Легальність функціонування систем ДТГ у демократичних країнах базується на принципах, які описані у Міжнародному пакті про політичні права [15]. Згідно з цим пактом *кожна людина має право брати участь в управлінні своєю країною* через представників, які обираються на засадах *загального і*

*рівного виборчого права при таємному голосуванні* [14]. Таємне голосування гарантує стаття 71 Конституції України і Закони України про вибори. Порушення службовою особою таємниці голосування є кримінальним злочином, за який передбачена відповідальність за статтею 159 Кримінального Кодексу України [15-17].

За умов вільного вибору не припускається будь-який примус щодо здійснення акту волевиявлення. Процедура голосування має проводитись таким чином, щоб було неможливо виявити, як проголосував той чи інший виборець і чи мав він намір голосувати. Забороняється примушувати виборців, навіть у суді, зізнаватись про те, як вони проголосували.

Під час проведення виборів має місце *принцип гласності* [51, 53]. Це означає, що організаторам виборів необхідно інформувати виборців про роботу системи ДТГ. Виборці мають право на ознайомлення з особливостями виборчої системи, що використовується, щоб прийняти для себе рішення щодо довіри або недовіри до коректності її функціонування. В демократичних країнах права людини, здебільшого, підтримуються державними інституціями і тому забезпечення коректності здійснення виборчих процедур не є нагальною проблемою. Проте, внаслідок усе більш широкого використання територіально розповсюджених інтернет-терміналів, з яких в деяких країнах виборці мають можливість проголосувати, контроль за процесом голосування спостерігачами від громадськості або преси істотно погіршується. Тому інформуванню про хід виборів в демократичних країнах почали приділяти більш значну увагу.

Опубліковано звіти про результати впровадження інтернет-технологій на виборах, зокрема у Норвегії [22], на виборах у США серед військових за кордоном [27] та на парламентських інтернет-виборах в Естонії [19]. Досліджено також користувацьку "зручність" інтерфейсів різних систем інтернет-голосування [29] та окремі вимоги до них [30-31]. Загальний висновок: системи ДТГ у країнах із сталими демократичними традиціями у повній мірі відповідають усім, зазначеним вище, принципам, а щодо «чесності» проведення

виборів, то отримані результати волевиявлення, за винятком деяких окремих форс-мажорних випадків, не мали нарікань від суспільства.

Інша справа – країни, що намагаються відповідати принципам демократії, у тому числі і Україна. На жаль, в таких країнах, внаслідок відомих причин [35, 37, 40, 44, 48, 49], відсутня довіра значної частини громадян до об'єктивності результатів голосування. У країнах з недорозвиненою демократією більшість потенційних виборців не упевнені, що організатори виборів забезпечать таємницю їхнього волевиявлення, що вони у змозі ефективно протидіяти спробам спотворити результати волевиявлення та запобігти проявам різного роду зловживань.

Дане дослідження являє наукову розробку, що спрямована на створення технології голосування, яка б не викликала сумнівів щодо коректності її функціонування.

### 1.2. Характеристики існуючих систем дистанційного голосування

Дистанційне голосування через Інтернет надає суттєві переваги виборцям щодо зручності, мобільності та економії часу, крім того, скорочуються немалі витрати на друк бюлетенів, а також, скоріш за все, і на інші витрати з організаційно-технічного забезпечення виборчої кампанії. Всеохоплюючий аналіз існуючих систем ДТГ наведено, зокрема, у [1], де розглянуто характеристики широкого ряду систем за різних умов їхнього функціонування [1-42].

Основний недолік існуючих нетранспарентних систем ДТГ, що використовуються на практиці, як показано у [1], полягає у відсутності дієвих механізмів всеохоплюючого контролю функціонування програмно-апаратних засобів системи голосування з боку громадськості у цілому, а також гарантій збереження таємниці голосів та забезпечення цілісності інформації у каналах обміну даними. Через це не може бути забезпечена впевненість виборців у тому, що в умовах адміністративного тиску, результати голосування не будуть спотворені, а голоси і персональні дані виборців не будуть розкриті.

За результатами цього аналізу встановлено, що суттєвим стримуючим фактором на шляху розвитку систем ДТГ є недовіра виборців щодо коректної роботи цих систем через неможливість упевнитись у тому, що у них не закладено можливостей для розкриття таємниці голосів та/або викривлення результатів волевиявлення. Показано, що для подолання недовіри треба надавати широкі можливості контролювання усіх об'єктів, суб'єктів і процесів, які викликають сумніви, оскільки від повноти можливостей контролю залежить рівень довіри. Саме такі можливості забезпечують так звані транспарентні (відкриті або прозорі) системи ДТГ.

*Під транспарентною системою ДТГ розуміємо* таку систему, у якій не тільки все без винятку програмне забезпечення (ПЗ), включаючи операційну систему (ОС), є заздалегідь відкритим для перевірок і експертиз, але є й можливість у режимі реального часу контролювати відсутність підміни або модифікації штатного програмного забезпечення, а також контролювати (з боку необмеженої кількості активістів з правом тільки на спостереження за роботою системи) точність і своєчасність виконання штатних дій персоналом щодо управління такою системою.

Принципи побудови транспарентних систем ДТГ детально описані в роботах [1-6, 41].

Зрозуміло, що для беззаперечної довіри необхідно надати усім бажаючим можливість контролювати усі складові системи протягом усього часу її функціонування. Саме такий підхід запропоновано в роботі [1], де детально описана транспарентна система таємного голосування, у якій надається можливість масового дистанційного контролю з боку необмеженої кількості будь-яких осіб щодо усіх програмних засобів та процесів в режимі реального часу функціонування цієї системи. У роботах [1-6, 41] розвинуто цей підхід і доведено, що після проведення такого контролю не залишається підстав для недовіри, бо всі елементи системи і дії обслуговуючого персоналу, які можуть бути потенційно небезпечними, є відкритими для масового спостереження. Іншими словами, будь-яка спроба вчинення зловмисної дії у такій системі

може бути виявлена та зафіксована контролюючими особами. При цьому забезпечується збереження таємниці голосів і неможливість викривлення результатів волевиявлення. Фактично тільки такі транспарентні системи можуть претендувати на беззаперечну довіру виборців, бо наявність хоч однієї закритої частини завжди буде породжувати підозри щодо фальсифікації.

### 1.3. Логічна модель транспарентної системи дистанційного голосування

В роботі [1] представлено логічну модель відкритої системи дистанційного голосування (див. рис. 1.1), де все, що знаходиться у середині зовнішнього кола, відповідає множині об’єктів сервера, котрий на стороні виборчої дільниці підтримує процеси інтернет-голосування. Надалі цей сервер будемо називати сервером виборчої дільниці (СВД).

У цій моделі передбачено, що операційна система СВД (після виконання процедур налаштування) дозволятиме виконувати тільки ті дії, які є елементами множини Q, де Q - об’єднання множин дій голосуючих виборців, адміністратора СВД та контролерів, які в сукупності складають повну групу можливих дій користувачів [1].

$$Q = V \cup A \cup K, \tag{1.1}$$

де $V$ – множина дій виборців під час їх звернення до СВД; $A$ – множина дій адміністратора сервера, до якої входять штатні і можливі позаштатні дії щодо управління сервером СВД ; $K$ – множина дій контролерів.

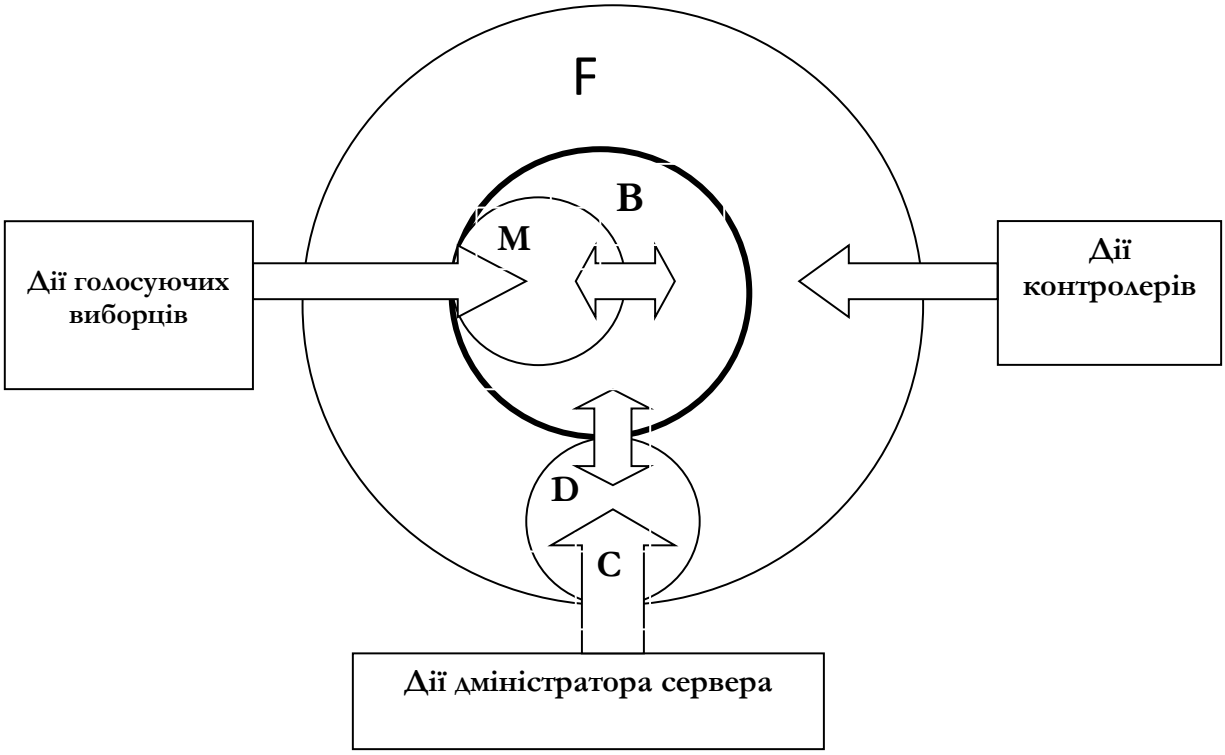



Рис. 1.1. Логічна модель відкритої системи дистанційного голосування [1]

Слід зауважити, що всі некоректні і помилкові дії тут не розглядаються, бо вони не сприймаються сервером.

Повна множина об’єктів, над якими виконуються дії користувачів складається з п’яти наступних множин: $F$ – множина даних у файлах, що розміщенні на сервері СВД; $C$ – множина відображень команд адміністратора сервера, при чому $C \subset F$, $f : C \to A$, де $f$ – функція відображення; $D$ – множина файлів у директорії, до якої має доступ адміністратор, при чому $D \subset F$; $B$ – множина даних в оперативній пам’яті прикладної програми, при чому $B \not\subset F$; $M$ – множина вихідних даних моніторингу звернень виборців, які прикладна програма використовує для авторизації голосуючих виборців, де $M \subset B$.

Множини дій користувачів над переліченими об’єктами описують наступні вирази [1]:

$$V = \{G_1, ..., G_i, ..., G_n\}, \qquad (1.2)$$

де $G_i$ – множина варіантів запитів виборця до СВД $i = \overline{1, n}$; $n$ – кількість варіантів запитів виборця до СВД; $f_м$ – функція моніторингу звернень голосуючих виборців до СВД [1].

$$A = W \cup E, \tag{1.3}$$

де $W$ – множина дій адміністратора для запису файлів (приєднання файлів до множини $D$); $E$ – множина дій адміністратора щодо запуску програмних файлів на виконання (з множини $D$);

$$K = R \cup P, \tag{1.4}$$

де $R$ – множина дій контролерів щодо ознайомлення з файлами множини $F$, при чому $C \subset F$, $D \subset F$; $P$ – множина дій контролера щодо отримання інформації від СВД за допомогою команд ОС з метою виявлення можливих порушень політики безпеки.

Даний аналіз показує, що єдиним користувачем, який має можливість для виконання небезпечних дій на сервері є адміністратор сервера, бо будь-які дії виборців і контролерів не в змозі створити загрозу штатній роботі сервера. Тому, для запобігання можливим несанкціонованим діям, адміністратору дозволено виконувати лише дві дії, а саме, заносити файли в його директорію і запускати на виконання програму лише з цієї директорії. При цьому, будь-яку нештатну дію адміністратора можуть виявити контролери. В адміністратора не існує таких дій щодо управління СВД, які можна було б приховати від контролерів.

Технологічний цикл функціонування транспарентної системи ДТГ з позначенням періодів часу щодо виконання окремих процедур та моментів, які слід враховувати при організації спостереження (контролювання), показано на рис. 1.2 [1].

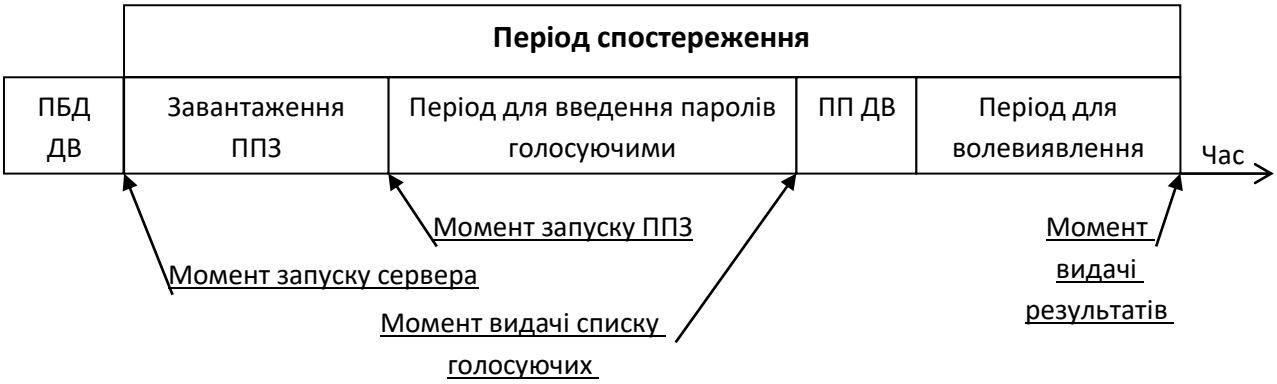



Рис. 1.2. Технологічний цикл функціонування системи ДТГ, де ПБД ДВ – період занесення даних про претендентів на дистанційне волевиявлення, а ПП ДВ – період підготовки до дистанційного волевиявлення (завантаження електронних бюлетенів після корегування) [1]

У [1] розглянуто наступні дії спостерігачів-контролерів у різні періоди часу функціонування сервера ВД та можливості отримання тих чи інших результатів контролю.

На початку функціонування сервера пропонується упевнитись у тому, що встановлена операційна система (ОС) є справжньою (без усякого роду підмін та модифікацій). Розглянуто ОС *OpenBSD* (*BBOS*), яка перевірена та сертифікована в Україні за критеріями ТЗІ. Цю ОС, а також її відкриті специфікації, можна вільно отримати з Інтернету або придбати на диску. Тому вірність вибору даної ОС не може викликати недовіру. Для перевірки ОС виділено значний інтервал часу для (3-4 тижні), так що будь-яка людина із базовими знаннями в галузі інформатики має можливість таку перевірку зробити, оприлюднити та обговорити результати на широкому форумі, а у разі виявлення порушень домогтися рішення щодо переустановлення ОС.

Після того, як отримано підтвердження вірності встановленої ОС пропонується протягом усього подальшого часу функціонування системи ДТГ постійно контролювати стан процесів на сервері для того, щоб уникнути модифікації або підміни зловмисниками штатної ОС. У [1] запропоновані методи здійснення процедур контролю роботи системи ДТГ, які зможе виконувати будь-який користувач мережі Інтернет. Користуючись цими методами кожен

бажаючий зможе впевнитись, що за весь час функціонування сервера не було змінено або модифіковано штатну ОС та інше штатне програмне забезпечення, а також усі дії персоналу щодо управління сервером ВД виконувались точно за встановленим графіком.

Зокрема, перший із відомих методів дозволяє виявити відсутність позаштатних втручань в роботу сервера протягом усіх проміжків часу, коли дії персоналу щодо керування сервером не передбачені регламентом, а також дозволяє виявити моменти початку та інтервали часу виконання дій, що пов'язані із керуванням системою ДТГ з боку її адміністраторів. Другий метод дозволяє упевнитись у тому, що дії персоналу точно відповідали затвердженому уповноваженим органом і опублікованому для суспільства регламенту.

У першому методі використано особливість ОС *OpenBSD*, яка полягає в тому, що ідентифікатори усіх процесів (*PID*), крім першого, для якого *PID*=1, обираються випадково з ряду чисел від 2 до 32767. При цьому 20 значень перших *PID* залишаються незмінними від моменту запуску ОС до завершення функціонування сервера. Неможливо замінити чи зробити перезапуск ОС так, щоб залишити незмінними ці 20 значень *PID*. Так що, процеси, які пов'язані з роботою контролерів, апріорі не можуть бути небезпечними для роботи сервера - вони мають характерні ознаки, а саме: значення *kontrol* у стовпчику *USER* або значення *sshd: kontrol* у стовпчику *COMMAND*. Все це можна побачити на рис.1.3, де наведено результат виконання команди ***ps -aux***.

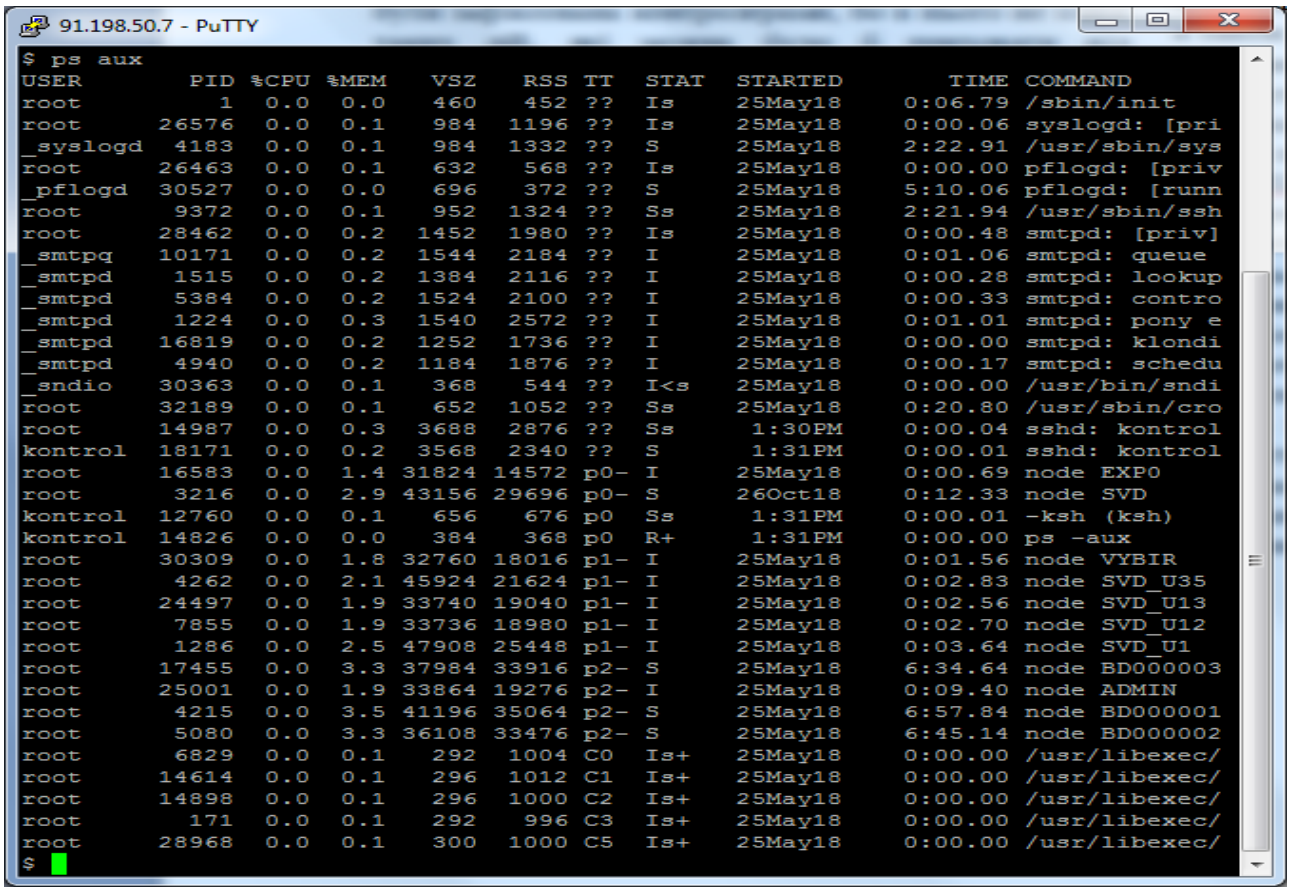



Рис.1.3. Роздруківка результату виконання команди ***ps -aux***

Для того, щоб будь-який користувач під ОС *Windows* міг реалізувати перший метод контролю, створено програму з відкритим кодом на мові C#. Цю програму разом з інструкцією можна скачати за посиланням *http://vybir.knuba.edu.ua/* та перевірити на сервері за *IP*-адресою 91.198.50.7.

На початку роботи програма завантажує з файлу *PIDIGNOR.txt* список *PID* постійно діючих в ОС процесів для ігнорування. Процеси контролерів програма ігнорує і не фіксує. У вікні програми друкуються, а також фіксуються у файлі *KRP.txt*, усі нештатні процеси та процеси адміністратора.

Таку програму найзручніше встановлювати на постійно діючому сервері із можливістю управління нею через *web*-інтерфейс і отриманням повідомлень від неї на електронну пошту або у вигляді *SMS*. У разі відсутності порушень штатної роботи сервера СВД перше, що буде зафіксовано програмою, це процес, який є початком дій адміністратора щодо занесення файлів ППЗ на сервер. Момент появи цього процесу повинен бути наперед відомим, бо він узгоджується у часі з

графіком проведення виборів. Використання даної програми, яка реалізує перший метод контролю, дозволяє упевнитись у відсутності порушень штатної роботи сервера до початку вказаних дій адміністратора. Після цього слід дочекатись завершення дій адміністратора, про що буде свідчити зникнення процесу із значенням *admin* у стовпчику *USER*, і задіяти другий метод контролю, який дозволяє перевірити точність дій адміністратора та відсутність у цих діях зловмисних намірів. Для цього можна скористатись програмою *WinSCP*, яка є у вільному доступі, і дозволяє виявити у директорії *home/admin* появу файлів ППЗ. Ретельної перевірки потребує тільки файл з програмою, яка є заздалегідь відкритою у вигляді тексту на мові *JavaScript* (розширення *js*). Ця програма призначена для управління сервером в автоматичному режимі від моменту її запуску до завершення роботи системи ДТГ. Найсерйознішою загрозою у нашому випадку є заміна або модифікація цієї програми з боку адміністраторів системи. Однак спостерігачам досить просто упевнитись у тому, що саме цю штатну програму підготовлено для запуску на сервері, а не якусь підробку, бо файл з цією програмою повинен бути розповсюдженим заздалегідь і потрібна лише елементарна перевірка на ідентичність.

Задача зловмисника, який хоче сфальсифікувати результат виборів або розкрити таємницю голосів, потребує заміни штатного файлу з цією програмою перед її запуском на підроблений, а після запуску повернення на місце штатного файлу. Все це треба зробити так швидко, щоб контролери не встигли зафіксувати. Інших можливостей для заподіяння подібної шкоди у персоналу, який керує системою ДТГ, у даній операційній системі не існує.

Щоб упевнитись у відсутності реалізації описаної загрози необхідно перевірити і зафіксувати у режимі реального часу такі події:

1. Початок процедури управління сервером СВД.
2. Запуск штатної програми на виконання.
3. Завершення сеансу управління сервером.
4. Відсутність будь-яких дій, які б мали змогу спричинити загрозу протягом цього сеансу.

За допомогою першого методу контролю із точністю до встановленого значення інтервалу між запитами (на рис.1.4 він дорівнює 10 секунд) можна без сумнівів забезпечити перевірку за першими трьома пунктами, але це не надає гарантій щодо виконання четвертого пункту. Для отримання таких гарантій необхідно, щоб адміністратор обов'язково через одну-дві хвилини після запуску програми виконував додаткову дію до завершення сеансу, а саме команду *history > history. У* результаті такої команди у файл *home/admin/history* будуть занесені усі команди, які були уведені під час даного сеансу управління сервером. У разі, якщо у цьому файлі буде виявлено тільки один рядок наступного змісту: *nohup node* <ім'я прикладної програми>, то можна після ще декількох простих перевірок упевнитись у тому, що ніяких небезпечних дій під час запуску програми не було виконано.

Розглянемо докладніше послідовність перевірок під час запуску серверної програми з використанням розглянутих вище методів.

Перш за все, слід перевірити, щоб момент запуску серверної програми з точністю до хвилини збігався із запланованим у регламенті, бо від цього моменту автоматично визначаються наступні інтервали часу, включаючи період голосування. Цей момент буде зафіксовано у рядку робочого вікна програми та у файлі *KRP.txt*. Цей рядок у полі *USER* буде мати значення *admin*, а у полі *COMMAND – node* <ім'я програми>. Процес, який супроводжує дії адміністратора, також буде мати значення *admin* у полі *USER*, а його поява на 1-5 хвилин випереджатиме момент запуску програми. Цей процес повинен зникнути за 1-2 хвилини після запуску програми. Залишається перевірити, щоб після запуску програми і виконання команди *history > history* адміністратор не виконував зловмисних дій, які були описані вище. Для цього достатньо перевірити значення *PID* для процесу серверної програми. Це значення повинно залишатись незмінним після завершення процедури управління. Далі у файлі *KRP.txt* буде фіксуватись тільки один процес виконання прикладної програми, який не повинен припинятись чи замінюватись до повного завершення циклу роботи системи ДТГ.

У разі позитивних результатів перевірок обома названими вище методами можна гарантувати точність виконання персоналом усіх потрібних штатних дій та неможливість порушення таємниці голосів або викривлення результатів підрахунку, бо в обраній операційній системі не існує шляхів для непоміченого контролюючими особами заподіяння шкоди під час керування роботою системи ДТГ.

Таким чином, можна констатувати, що представлена в [1] логічна модель (див. рис. 1.1) за умов повної відкритості ПЗ, включаючи ОС, створює умови для забезпечення досконалого захисту критичних даних при обміні через середовище Інтернет, і, якщо ці умови відповідним чином використані, гарантує збереження таємниці голосів та неможливість фальсифікації результатів волевиявлення, навіть, коли існує повна недовіра до усіх без винятку користувачів системи. Крім того, як показано в роботах [1, 4, 6] така модель дозволяє застосовування методу протидії незаконному впливу на виборців.

Слід зазначити, що головним досягненням транспарентних систем дистанційного голосування є можливість їх повноцінного всеохоплюючого контролю з боку необмеженої кількості будь-яких суспільно активних громадян. Тільки такий контроль не залишає жодних підстав для недовіри щодо дійсності результатів волевиявлення і збереження таємниці голосів.

З метою збереження достоїнств транспарентної системи ДТГ, не можна розміщувати на сервері СДВ файли, які не є відкритими для ознайомлення, бо це позбавляє систему прозорості, а встановлення на цьому сервері додаткового програмного забезпечення для автентифікації осіб виборців буде ускладнювати процедуру спостереження за роботою сервера та перешкоджатиме проведенню повноцінного контролю. Тому розміщення на серверах ВД додаткових засобів, зокрема для розпізнавання осіб виборців, є неприпустимим. Навпаки, слід максимально обмежувати функціональність цього сервера, видаляючи з нього усі зайві файли, з метою забезпечення прозорості для повноцінного контролю з боку спостерігачів. Це означає, що розміщення додаткових засобів для розпізнавання

осіб виборців, потребує додаткового сервера, який повинен сприймати ознаки виборців і обмінюватись даними із серверами ВД.

## 1.4. Аналіз характеристик існуючого прототипу, що обраний для удосконалення транспарентної системи голосування

### 1.4.1. Модель загроз інформації у прототипі

Проаналізуємо модель загроз, протидія котрим забезпечена програмно-технічними засобами захисту у прототипі [1], бо існують сумніви щодо повноти цієї моделі.

«Слабкі місця» прототипу щодо захисту інформації, через які можливе виникнення загроз, відсутність протидії котрим підриває довіру до того, що результати голосування будуть відображати дійсну волю виборців, представлено у табл. 1.1.

**Таблиця 1.1**

**Загрози, що ставлять під сумнів коректність роботи прототипу**

| № з.п. | Визначення загрози | Методи нейтралізації загроз |
|---|---|---|
| 1 | Реєстрація фіктивних виборців | 1. Використано **модель системи ДТГ**, що гарантує цілісність результатів волевиявлення та збереження таємниці голосів в умовах недовіри до всіх без винятку осіб, що приймають участь у розробці, створенні та обслуговуванні СДВ.<br>2. Використано **метод спостереження у** реальному часі за станом сервера і діями адміністратора системи ДТГ з боку необмеженого кола будь-яких користувачів Інтернету, що виключає можливість виникнення непомічених порушень прийнятої політики безпеки та усуває підстави для недовіри до коректної роботи системи ДТГ. |
| 2 | Заміна системного ПЗ сервера на нештатне | |
| 3 | Модифікація штатного системного ПЗ сервера | |
| 4 | Виконання позаштатної команди управління сервером | |
| 5 | Підробка прикладного ПЗ сервера | |
| 6 | Нелегальна фізична підміна сервера | |
| 7 | Доповнення серверного обладнання нештатними засобами з метою реалізації *MITM* (атаки посередника) | |
| 8 | Підміна результатів голосування у процесі розсилки | |
| 9 | Підробка результату підрахунку голосів | |

| | | |
|---|---|---|
| 10 | Примус виборців віддавати свій голос усупереч їх власному бажанню | Використано метод нейтралізації спроб здійснення будь-яких видів тиску на учасників процедур волевиявлення. |
| 11 | Порушення цілісності та (або) конфіденційності інформації при обміні даними через Інтернет | 1. Використано **метод досконало захищеного обміну даними**, що гарантує цілісність та конфіденційність інформації при обміні даними через Інтернет.<br>2. Використано **метод отримання чисто випадкових бітових послідовностей**, засобами виключно типового клієнтського обладнання масового виробництва. |
| 12 | Перехоплення автентифікаційних даних виборців з метою підміни голосуючої особи | |

Як бачимо, у цій моделі не враховано можливість здійснення атаки посередника. Окрім того, ця модель не передбачає можливість виникнення інших загроз, що пов’язані із можливими змінами в архітектурі системи ДТГ.

1.4.2. Функціональний профіль захищеності інформації у прототипі

Технологія дистанційного волевиявлення, що реалізована у прототипі, складається з ряду процедур із різними вимогами щодо функціональних послуг захисту інформації. Характеристики цих процедур з позицій ТЗІ, починаючи від формування електронних списків виборців у розрізі кожної виборчої дільниці і до отримання остаточного результату підрахунку голосів, надано у таблиці 1.2.

**Таблиця 1.2**

**Функціональний профіль захищеності інформації у системі ДТГ** [1]

| **Назва процесу** | **Назва дії, виконавець** | **Необхідність у послугах захисту** | **Специфікації послуг рівня Г7** |
|---|---|---|---|
| Очна перевірка і реєстрація претендентів на дистанційне волевиявлення | Занесення даних претендента у базу, реєстратор | Ідентифікація та автен-тифікація отримувача | НП-1<br>НИ-1 |
| | Введення пароля, претендент | Ідентифікація та автен-тифікація відправника | НА-1<br>НИ-1 |
| Підготовка сервера СДВ | Завантаження і запуск ОС, адміністратор | Не впливають на рівень довіри з боку суспільства | Набір послуг захисту від НСД до підсистеми керування СДВ |
| | Завантаження і запуск ППЗ, адміністратор | | |

| | | | |
|---|---|---|---|
| Очна перевірка особи голосуючого і введення пароля для голосування<br><br>Підготовка до волевиявлення | Перевірка особи, реєстратор | Однонаправлений достовірний канал<br>Одиночна ідентифікація та автентифікація<br><br>Не впливають на рівень довіри з боку суспільства | НК-1<br>НИ-2<br><br>Набір послуг захисту від НСД до підсистеми керування СДВ |
| | Введення пароля, голосуючий | | |
| | Завантаження бюлетенів, адміністратор | | |
| Дистанційне волевиявлення | Акт волевиявлення, голосуючий | Ідентифікація і автентифікація при обміні<br>Абсолютна конфіденційність при обміні | НВ-2<br>КВ-4 |
| | Отримання довідок про кількість голосуючих, голосуючий | Не впливають на рівень довіри з боку суспільства | Набір послуг захисту від НСД до підсистеми керування СДВ |
| | Підрахунок голосів, сервер | | |
| Вивід результатів волевиявлення | Вивід результату, сервер | | |
| Масове спостереження | Перевірка справжності ОС та ПЗ, спостерігач (будь-хто) | Послуги спостереженості:<br>1) Самотестування у реальному часі (за запитом спостерігача);<br>2) Розподіл обов'язків на підставі привілеїв;<br>3) Цілісність КЗЗ з функціями диспетчера доступу | НТ-3<br>НО-3<br>НЦ-3 |
| | Перевірка справжності сервера, спостерігач | | |
| | Перевірка дій адміністратора, спостерігач | | |

Коди функціональних послуг захисту, що наведені в таблиці 1.2, надано згідно специфікаціям НД ТЗІ [56-57].

Під послугою масового спостереження мається на увазі контроль, при якому не накладаються обмеження на кількість контролерів. У нашому випадку число виборців на дільниці не може перевищити 2,5 тисячі осіб, що у певній мірі обумовлює можливу кількість контролерів.

Функціональний профіль захищеності інформації у прототипі, як бачимо із табл.1.2, повністю відповідає моделі загроз, що представлена у табл.1.1.

1.4.3. Характеристики середовища функціонування прототипу

Зазначимо, що під поняттям абсолютно відкритого середовища системи ДТГ розуміється середовище функціонування глобальної комп'ютерної мережі, що базується на використанні ресурсів Інтернет і задовольняє наступним вимогам:

1) усе без винятку програмне забезпечення (ПЗ) є відкритим і доступним без будь-яких обмежень для перевірок і випробувань;

2) апаратні засоби є стандартними із відомою структурою побудови за архітектурними, топологічними та функціональними ознаками;

3) існує необмежена кількість реально функціонуючих точок дистанційного доступу через Інтернет з використанням відкритого стандартного мережного протоколу до усіх без виключення серверів виборчих дільниць для необмеженої кількості користувачів ресурсів системи ДТГ;

4) усі без винятку користувачі мають право контролю змісту усіх без винятку комп'ютерних файлів, що встановлені на серверах виборчих дільниць (можливо, крім тої обмеженої кількості закритих для читання файлів ядра операційної системи, зміст яких щодо можливостей модифікацій чи підмін не викликає сумнівів);

5) усі без винятку користувачі, зокрема громадські контролери, мають право контролю відсутності подій перезапуску ОС від моменту їхнього запуску, що встановлені на усіх активних макро-елементах системи, до моменту фізичного виключення серверів дистанційного волевиявлення (тобто, після закінчення виборчої кампанії);

6) усі без винятку користувачі мають право і можливість здійснювати активний контроль (за допомогою спеціальних команд) усіх процесів, які знаходяться у стадії виконання.

1.4.4. Характеристики ОС сервера виборчої дільниці

1) Операційна система (ОС) на сервері виборчої дільниці прототипу є відкритою і простою для забезпечення можливості її детальної перевірки будь-

якою зацікавленою особою, що має стандартний рівень знань спеціаліста з комп'ютерних технологій.

2) Специфікації ОС, у т.ч. вихідні коди комп'ютерних програм, є опублікованими та доступними в Інтернеті для встановлення на будь-яких хостах з метою забезпечення можливості порівняння файлів серверної ОС із файлами ОС, що встановлені на комп'ютерах користувачів Інтернет.

3) Функціональність ОС забезпечує гарантовано досконалий захист від несанкціонованого доступу (НСД) з тим, щоб адміністратор сервера не мав можливості перекладати свою відповідальність на формально невизначену особу.

4) ОС забезпечує доступ до сервера кожному виборцю для перевірки усіх файлів і процесів без можливості заподіяння будь-якої шкоди нормальній роботі сервера.

1.4.5. Характеристики засобів програмування

1) Забезпечено мінімальну кількість та максимальну простоту і популярність обраних мов програмування для створення прикладних програм (ПП), з метою спрощення перевірки ПП з боку громадських контролерів.

2) Забезпечена можливість функціонування ПП на різних програмно-апаратних платформах з метою зняття зайвих обмежень під час перевірок ПП будь-якими особами на будь-якому обладнанні.

3) Забезпечена повна відкритість інструментальних програмних засобів, які обрані для створення і забезпечення функціонування ПП на сервері виборчої дільниці.

1.4.6. Характеристики засобів забезпечення контрольованості ОС

Ці засоби забезпечують:

1) абсолютний захист від НСД в період виконання заданої прикладної задачі;

2) надання спеціальних прав користувачам для контролю усіх файлів, як самої ОС, так і всіх інших файлів у її середовищі;

3) унеможливлюють створення скритих файлів;

4) реалізовують функціональність команди, за допомогою якої користувачі мають можливість впевнитись, що робота прототипу не переривалась у часі на протязі тривалості виборчої кампанії (доступ до цієї команди є необмеженим);

5) реалізовують функціональність команди, за допомогою якої користувачі мають можливість безперешкодної перевірки у реальному часі стану виконання усіх без винятку процесів і команд управління (доступ до цієї команди є не обмеженим);

6) унеможливлюють виникнення будь-якого шкідливого впливу на роботу прототипу в умовах необмеженого доступу до команд контролю параметрів, що визначають стан роботи системи;

7) реалізовують можливість встановлення на загальнодоступних комп'ютерах виборців, що підключені до Інтернет, безпосередньо із мережі необхідної ОС та прикладного ПЗ системи ДТГ з функціями контролю середовища функціонування;

Система легко і швидко встановлюється і перевіряється.

1.4.7. Забезпечення відкритості та контрольованості прикладного ПЗ

1) Прикладне ПЗ, що встановлене на сервері виборчої дільниці, є відкритим і простим (з метою забезпечення можливості його детальної перевірки будь-якою зацікавленою особою, що має стандартний рівень знань спеціаліста з комп'ютерних технологій).

2) Специфікації прикладного ПЗ, у т.ч. вихідні коди комп'ютерних програм, опубліковані в Інтернеті; існує можливість безпосереднього встановлення цих програм на будь-яких хостах, що дає змогу порівняти файли серверного прикладного ПЗ із файлами прикладного ПЗ, що встановлено на комп'ютерах користувачів Інтернет.

3) Мови програмування для створення цього забезпечення є широко відомими і популярними.

4) Тексти програм є зрозумілими і не потребують спеціальних знань та вмінь .

5) Серверна програма розміщується у єдиному модулі (це спрощує спостереження за її роботою).

6) Файли даних мають просту структуру (це спрощує процедури їх перевірки).

7) Конфіденційні дані у файлах зашифровані таким чином, що їх дешифрування протягом визначеного періоду є практично неможливим.

8) Обробка конфіденційних даних відбувається виключно в оперативній пам’яті діючих програм, після чого конфіденційні дані знищуються.

9) Інтерфейси для обміну даними із зовнішніми програмно-апаратними засобами унеможливлюють проникнення до системи будь-якої інформації зловмисного характеру.

10) Створено умови для внесення пропозицій з метою вдосконалення роботи прикладного ПЗ. Надання таких можливостей, безумовно, сприяє досягненню головної мети створення відкритих комп’ютерних систем, а саме, – усуненню будь-яких підстав для недовіри цим системам.

1.4.8. Забезпечення відкритості технології обробки інформації

Технологія обробки інформації у системі ДТГ забезпечує контроль за роботою системи і виявлення можливих відхилень від штатного режиму функціонування системи.

1) Чітко визначено моменти або інтервали часу щодо дій, які мають виконуватись у системі, за умов відсутності відхилень від режиму штатної роботи. До таких дій слід віднести команди управління роботою системи, завантаження файлів та запуск програм. Для спрощення контролю за роботою системи у реальному часі контролюючим особам надано вільний доступ до розкладу цих дій.

2) Розроблено конкретні рекомендації щодо контролювання роботи системи у реальному часі із прив’язкою до розкладу штатних дій.

3) Створено спеціальний ресурс в мережі Інтернет, на якому розміщена вся необхідна інформація щодо контролю за роботою системи і надана можливість

оперативного звернення до відповідальних осіб для інформування про виявлені нештатні дії.

### 1.5. Недоліки прототипу відкритої досконало стійкої системи дистанційного таємного голосування, постановка завдань досліджень

Аналіз характеристик системи ДТГ, що запропонована у [1], показав, що вона має ряд суттєвих достоїнств у порівнянні з іншими відомими у світі системами аналогічного призначення. До таких достоїнств, перш за все, слід віднести наступне.

1) Прозорість здійснення усіх без винятку виборчих процедур, завдяки чому забезпечується абсолютна гарантованість захисту у межах прийнятої моделі захисту, що згідно НД з ТЗІ в Україні прийнято відображати у вигляді реалізованого профілю захищеності інформації. Такий профіль надано у [1].

2) Досконала формально доказана стійкість захисту інформації у межах прийнятої моделі загроз, що надана у [1] і є, на наш погляд, адекватною умовам використання систем ДТГ в Україні.

3) Абсолютна гарантованість захисту як таємниці, так і результатів голосування в умовах повної недовіри громадян до усіх без виключення суб'єктів, що мають або можуть мати хоч якесь відношення до процесу волевиявлення.

4) Відсутність необхідності у користуванні будь-якими спеціалізованими термінальними пристроями для здійснення акту волевиявлення. Для цього виборцю достатньо мати звичайний термінальний пристрій мобільного зв'язку або Інтернет.

Як було вже відмічено, основною перешкодою впровадженню систем ДТГ у практику суспільних відносин, як показано у [1], є відсутність довіри до об'єктивності результатів голосування. У випадку дистанційного волевиявлення (ДВ) через Інтернет виборець знаходиться поза меж контрольованої зони домену безпеки (тобто, поза меж виборчої дільниці), що суттєво збільшує можливості зловмисників щодо порушень свободи волевиявлення. Окрім того, обмін конфіденційними даними між виборцями та сервером ДВ здійснюється,

зазвичай, через відкриті канали зв'язку, що також збільшує можливості зловмисників щодо порушень конфіденційності, цілісності та доступності персональних даних. Слід підкреслити, що функціональність системи технічного захисту інформації (ТЗІ) , що запропонована до реалізації у найбільш розвиненій та адаптованій до реалій України розробці системи ДТГ [1], передбачає використання ефективних механізмів протидії зазначеним вище загрозам, що у значній мірі вирішує проблему недовіри виборців.

Названа вище система ДТГ [1] є транспарентною, забезпечує в умовах України досконалу (згідно К.Шеннону [52]) формально доказану стійкість захисту інформації і надає цей захист із 100% гарантією неможливості здійснення будь-яких не передбачених регламентом дій навіть за умов відсутності будь-якої довіри щодо коректної роботи системи з боку будь-якого суб'єкту, котрий має хоч якесь відношення до процесу волевиявлення.

Порівняння характеристик існуючих технологій з тими, що пропонуються у даній роботі представлено у таблиці 1. 3.

**Таблиця 1.3**

**Порівняння технологій дистанційного таємного Інтернет-голосування**

| Опис технології | Критерій порівняння якості обслуговування виборців | | | | |
|---|---|---|---|---|---|
| | Витрати часу на доступ до сервера | Витрати часу на процедуру голосування | Необхідність очної перевірки особи | Необхідність встановлення клієнтського програмного забезпечення | Прозорість апаратних засобів |
| Запропонована у роботі Пригари М.П. | Значні | Малі | Перед кожним голосуванням | Немає | Немає |
| Естонська система | Значні | Значні | Тільки один раз | Є | Немає |
| Запатентована у США US 2017/ 0109955 | Значні | Значні | Тільки один раз | Є | Немає |
| Пропонується у даній роботі | Мінімі-зовані | Малі | Тільки один раз | Немає | Є |

Вищенаведене дозволяє розглядати систему ДТГ, що запропонована у [1], як основний варіант технічних рішень (тобто, прототип) при вирішенні завдань реальної розробки такої системи для України.

Проте цій системі притаманні наступні недоліки.

По-перше, запропоноване у [1] не передбачає автоматизацію процесу пошуку виборцем *IP*-адреси своєї виборчої дільниці серед десятків тисяч інших дільниць. Виборець самостійно має здійснювати такий пошук, використовуючи методи, що висвітлені у [1]. Це створює незручність для виборця, що не сприяє зацікавленості у використанні дистанційного способу волевиявлення.

Автоматизація процесу пошуку має враховувати можливість виникнення непередбачуваних перешкод у мережі доступу до серверного обладнання системи ДТГ. Внаслідок суттєвих за величиною та швидких за динамікою пульсуючих змін інтенсивності потоку звернень на обслуговування, що надходять від термінальних пристроїв виборців до серверного обладнання у період голосування, що мають, до того ж, майже не прогнозований характер (про це свідчать результати багатьох досліджень пакетного трафіка [7, 61, 63]), в окремі випадкові проміжки часу можуть виникати перевантаження серверів, що обробляють запити виборців. Під час дистанційного голосування можуть утворюватися черги звернень виборців до серверів виборчих дільниць. На певних проміжках часу тривалість чекання у чергах може виявитись неприпустимо великою для широкого кола виборців. Як результат – надмірні затримки або, навіть, відмови в обслуговуванні цих запитів. Зрозуміло, що за цих умов існує потреба у розробці засобів підтримки певного рівня якості обслуговування запитів виборців з тим, щоб тривалість можливого чекання у чергах на обробку запитів не перевищувала припустимого рівня.

По-друге, дистанційне голосування, хоч і не потребує фізичної присутності голосуючих на виборчих дільницях під час здійснення акту волевиявлення, тим не менш, передбачає необхідність здійснення обтяжливих очних перевірок осіб виборців в період уточнення списків голосуючих перед кожним актом

волевиявлення. Така необхідність у [1] не усунута. Це в значній мірі обмежує можливості виборців щодо мобільності і потребує з точки зору сучасної людини невиправданих витрат часу. От же, за цих обставин є доцільним розробити технологію беззаперечної дистанційної автентифікації особи виборця в умовах повної відкритості (прозорості) системи ДТГ.

На кінець, по-третє, найбільш суттєве, технічне рішення, що запропоноване у [1], не передбачає аудиту апаратних засобів, які забезпечують збереження таємниці голосів та їх підрахунок. Методи технічного захисту інформації (ТЗІ), які пропонуються використати при реалізації технології [1] не надають упевненості у тому, що усі специфічні загрози інформації, які притаманні виборчим кампаніям, належним чином нейтралізовані. Перш за все, це стосується засобів нейтралізації так званої атаки посередника, яку називають ще *MITM* (*Man In The Middle*) і вважають однією із найбільш небезпечних загроз в системах ДТГ. Особливість реалізації атаки посередника у відкритій (прозорій) системі голосування полягає в тому, щоб показувати контролерам картину нормально функціонуючого СВД, а запити виборців непомітно для контролерів перехоплювати і обробляти іншим сервером, у якому закладені можливості для підробки та/або розкриття таємниці голосів виборців.

От же, для реального впровадження в умовах України доцільно за основу узяти систему ДТГ, що запропонована у [1] та усунути її недоліки, що розглянуто вище.

Таким чином, для реалізації мети даної роботи відповідно до висновків, зроблених за результатами наведеного вище аналізу, слід виконати наступні завдання:

1. Удосконалити технологію доступу виборців до серверного обладнання системи ДТГ, розширивши її функціональні можливості з тим, щоб спростити процес доступу та мінімізувати затримки в обслуговуванні. Удосконалення здійснити шляхом розробки відповідної моделі інформаційної взаємодії елементів системи ДТГ і методів, що забезпечують автоматизацію пошуку для виборців *IP*-адрес їхніх виборчих дільниць та реалізують оптимальне адаптивне

управління пульсуючим потоком звернень, що надходять від виборців під час голосування до серверного обладнання системи ДТГ.

2. Усунути необхідність здійснення очних перевірок виборців перед кожним актом волевиявлення та нейтралізувати можливі атаки посередника. Зокрема, створити відповідний метод протидії атаці посередника та розробити технологію дистанційної автентифікації виборців з урахуванням обмежень, що пов'язані із забезпеченням транспарентності системи ДТГ.

3. Створити програмне забезпечення для удосконаленого варіанту технології ДТГ. Оцінити показники якості функціонування цього програмного забезпечення.

Матеріали щодо виконання вищеназваних завдань містяться у наступних трьох розділах роботи.

**Висновки до першого розділу**

1. Системи дистанційного таємного голосування (ДТГ), що функціонують у демократичних країнах, в основному, відповідають фундаментальним принципам Міжнародного пакту про політичні права. Але вони не надають можливість аудиту апаратних засобів, які забезпечують збереження таємниці голосів та їх підрахунок, що є актуальним для громадян країн, що стали на шлях демократичних перетворень. У таких країнах (у т.ч., в Україні) більшість потенційних виборців не упевнені, що організатори виборів забезпечать таємницю їхнього волевиявлення, що вони у змозі ефективно протидіяти спробам спотворити результати волевиявлення та запобігти проявам різного роду зловживань. Тому у таких країнах доцільно впроваджувати так звані транспарентні системи ДТГ, котрі надають можливість в режимі реального часу контролювати відсутність підміни або модифікації штатного програмно-апаратного забезпечення, а також контролювати (з боку необмеженої кількості активістів) точність і своєчасність виконання штатних дій персоналом щодо управління такими системами.

2. Здійснено аналіз характеристик транспарентних систем ДТГ, зокрема детально розглянуто логічну модель та технологічний цикл функціонування таких систем. Показано, що представлена модель за умов повної відкритості програмно-апаратного забезпечення, включаючи операційну систему, дозволяє забезпечити досконалий захист критичних даних при обміні через середовище Інтернет, а також гарантує збереження таємниці голосів та неможливість фальсифікації результатів волевиявлення за умови повної недовіри до усіх без винятку користувачів системи. Крім того, показано, що така модель дозволяє застосовувати методи протидії незаконному впливу на виборців.

3. Здійснено аналіз характеристик існуючого прототипу, що обраний для удосконалення транспарентної системи ДТГ. Розглянуто модель загроз для інформації прототипу, протидія котрим забезпечена його програмно-технічними засобами захисту. Визначено функціональний профіль захищеності інформації у прототипі. Розглянуто характеристики середовища функціонування та усіх його складових елементів. Показано, що розглянутому прототипу у повній мірі притаманні властивості відкритої системи. От же, для розробки адекватного технічного рішення транспарентної системи ДТГ і її реального впровадження в умовах України доцільно за основу узяти розглянутий прототип.

4. Виявлено основні недоліки існуючих транспарентних систем. Зокрема, у них не передбачена автоматизація процесу пошуку виборцем *IP*-адреси своєї виборчої дільниці. Це створює незручність для виборця і не сприяє зацікавленості у використанні дистанційного способу волевиявлення. Не усунута необхідність здійснення обтяжливих очних перевірок осіб виборців в період уточнення списків голосуючих перед кожним актом волевиявлення. Відсутні засоби аудиту апаратних засобів, які забезпечують збереження таємниці голосів та їх підрахунок, а також не передбачені засоби виявлення виборцями під час голосування атаки посередника, яку називають ще *MITM* (*Man In The Middle*) і вважають однією із найбільш небезпечних загроз в системах ДТГ.

Визначено наукові завдання, вирішення котрих має забезпечити можливість усунення названих вище недоліків.

# РОЗДІЛ 2
# УДОСКОНАЛЕННЯ ТЕХНОЛОГІЇ ДОСТУПУ ВИБОРЦІВ ДО РЕСУРСІВ СИСТЕМИ ІНТЕРНЕТ-ГОЛОСУВАННЯ

## 2.1. Задачі удосконалення технології дистанційного доступу виборців до серверного обладнання системи інтернет-голосування

Основна перешкода, яка стримує процес впровадження систем ДТГ у практику суспільних відносин, - відсутність довіри з боку переважної більшості громадян до можливості забезпечення транспарентної (прозорої) роботи цих систем [1,8]. Через це виникає підозра щодо можливості фальсифікації результатів волевиявлення та/або порушення таємниці голосів адміністраторами системи ДТГ. У [1] запропоновано технологію, що забезпечує прозорість роботи системи ДТГ та досконалий захисту її інформаційних ресурсів. Іншими словами, зроблена спроба усунення підстав для недовіри щодо можливості створення «чесних» систем ДТГ. Однак, щоб якомога більше виборців були зацікавлені у користуванні засобами такої ДТГ, треба зробити дистанційне голосування не тільки захищеним від загроз, але і зручним для виборців. Технологія, що запропонована у [1], не є достатньо зручною у користуванні, бо, з метою прискорення обробки голосів та для забезпечення можливості масштабування на будь-яку кількість виборців, в ній передбачено використовувати окремі сервери для кожної виборчої дільниці, а це означає, що виборцю, наприклад, в Україні треба буде відшукати адресу потрібного сервера серед 33 тисяч, що функціонують на виборчих дільницях. В роботі [1] для цього пошуку запропоновано задіяти сервер формування загального результату голосування, на якому є дані про адреси усіх виборчих дільниць. Таке рішення у разі зростання кількості виборців може призвести до затримки процесу пошуку необхідної адреси через перевантаженість сервера у разі зростання інтенсивності потоку звернень виборців до цього серверу у непередбачувані моменти часу, що може негативно впливати на бажання громадян щодо користування системою дистанційного голосування [10, 12]. От же, ця технологія потребує

удосконалення у напрямі вирішення завдань із забезпечення доступу виборців до потрібних ресурсів без зайвих ускладнень. У даному розділі представлено результати розробки системи та відповідної технології автоматизованого пошуку за зверненнями виборців IP-адрес серверів ВД, до яких вони отримали легальний доступ. Така розробка здійснена з метою спрощення інтерфейсу та заощадження виборцями витрат часу на здійснення актів волевиявлення.

Критерії доступності з технічної точки зору передбачають таке [56-57, 63]:

- мінімізацію відмов в обслуговуванні;
- відновлення роботи після збоїв;
- гарячу заміну обладнання у разі виходу з ладу;
- забезпечення доступу користувачам до інформаційних ресурсів без зайвих ускладнень.

Розглянемо останній четвертий пункт наданого переліку, оскільки перші три стосуються сфери забезпечення надійності обладнання, яка не є об’єктом даного дослідження.

Єдине ускладнення при інтернет-голосуванні у порівнянні із традиційним методом голосування – необхідність користування паролями. Але користування паролями в Інтернеті – широко розповсюджена процедура, яка є звичайною для будь-якого користувача. Так що цей недолік слід вважати несуттєвим.

Основна незручність для виборця при дистанційному голосуванні - необхідність пошуку *IP*-адреси своєї виборчої дільниці серед десятків тисяч інших дільниць. У роботі [1] запропоновано метод, що полегшує її пошук, але суттєво таку незручність не усуває.

Інша незручність – можливі затримки в обробці запитів виборців. Оскільки потік звернень виборців до серверного обладнання під час інтернет-голосування має випадковий непрогнозований характер, то не виключена можливість виникнення сплесків у цьому потоці [7]. Внаслідок чого можливе перенавантаження серверного обладнання трафіком, що призводить до виникнення черг в обслуговуванні звернень або, навіть, до їхньої втрати. Затримки через очікування в чергах вкрай негативно сприймаються більшістю

виборців. Тому вони мають бути мінімізовані і не більшими певного порогового рівню, що обирається із інтуїтивних міркувань розробниками системи ДТГ.

### 2.2. Прискорення доступу виборців до ресурсів системи ДТГ

Якщо проаналізувати часову діаграму процесу обслуговування виборця сервером ВД за умов, коли цей сервер обслуговує не більше 2500 чоловік, а пошук *IP*-адреси своєї виборчої дільниці виконується виборцем самостійно перед початком здійснення процедури інтернет-голосування, то, як показано у роботі [5], час очікування відповіді від сервера ВД у середньому буде не перевищувати 2-3 секунди. Однак у реаліях України сервер визначення *IP*-адреси повинен бути потенціально здатним здійснювати обслуговування більше 10 мільйонів виборців. І хоч на практиці певна кількість виборців буде знати *IP*-адресу серверу своєї виборчої дільниці заздалегідь, тим не менш, потреба в визначенні цієї адреси безпосередньо в процесі голосування може виникнути у мільйонів виборців. При цьому слід враховувати сплески трафіка звернень, які можуть бути значними і, з очевидних міркувань, майже не прогнозованими. Такі сплески негативно впливають на час очікування відповіді від сервера.

Для мінімізації зусиль виборців з пошуку адрес своїх виборчих дільниць при ДТГ існує можливість скористатися відомим методом пошуку адрес на відповідних мапах місцевості [1,5]. У даному випадку маємо справу з великою кількістю адрес, що є об’єктами пошуку, тому доводиться здійснювати пошук у 2-4 етапи на мапах різного масштабу, що потребує значних витрат часу. У роботі [5] розглядається альтернативний варіант пошуку з використанням назв міст або селищ і вулиць, як це реалізовано у Державному реєстрі виборців на ресурсі *https://www.drv.gov.ua/portal/!cm_core.cm_index*. Однак і цей метод пов’язаний із значними витратами часу на пошук.

Обидва ці методи безпосереднього (ручного) пошуку виборцем *IP*-адреси сервера ВД своєї виборчої дільниці можливо застосовувати в інтерактивному режимі взаємодії виборця із спеціально виділеним сервером, на котрому містяться дані щодо *IP*-адрес усіх виборчих дільниць, але враховуючи значні

витрати часу виборцем при використанні як першого, так і другого методу, здійснення такого пошуку для мільйонів виборців потребує побудови більш складної схеми обслуговування, ніж окремий сервер.

Модель інформаційної взаємодії елементів системи ДТГ із забезпеченням автоматизованого пошуку *IP*-адрес серверів ВД, що пропонується у даній роботі, базується на використанні відомих методів безпосереднього пошуку виборцями цих адрес, які описані у попередньому розділі. Програми, які реалізують ці методи, встановлюються на сервери пошуку адрес (ПА), що виконують в інтерактивному режимі процедуру пошуку *IP*-адрес. Як було зазначено, пропускної спроможності одного сервера ПА буде недостатньо для того, щоб забезпечити задовільне значення середнього часу очікування у чергах на обробку звернень виборців. Тому у моделі, яку ми пропонуємо, відому технологічну схему обробки інформації, яка детально розглянута у [1], доповнено лінійкою серверів ПА та сервером перенаправлення звернень виборців (ПЗВ), який розподіляє потік звернень виборців між серверами лінійки з метою уникнення можливих перенавантажень у роботі цих серверів. При цьому загальна пропускна спроможність (продуктивність) одного сегменту пошуку *IP*-адрес у системі ДТГ щодо обробки звернень виборців фактично буде дорівнювати сумі продуктивностей усіх серверів ПА, що входять до складу цього сегменту.

В умовах України, де може голосувати дистанційно 4 млн виборців протягом 12 годин, на сервер ПЗВ за секунду може, в середньому, надходити 100 запитів. І якщо лінійка серверів ПА буде складатися із 10 - 50 серверів, то реального часу на обчислення є лише 2-3 ms. Так що продуктивності одного сервера ПЗВ в залежності від конкретних умов застосування може виявитися недостатньо. Тому у склад системи ДТГ може бути включено кілька сегментів пошуку *IP*-адрес в залежності від того, яка частка виборців визначила *IP*-адреси своїх виборчих дільниць заздалегідь до початку здійснення процедури голосування.

Алгоритм роботи сервера ПЗВ залежить від характеристик потоку звернень виборців, зокрема від характеристик пульсацій трафіка. Бо не виключено, що активність значної кількості виборців може співпадати у часовому вимірі.

Схематичне відображення протоколу інформаційної взаємодії засобів технічної підтримки прискореного пошуку *IP*-адрес серверів ВД при здійсненні виборцями процедури Інтернет-голосування представлено на рис.2.1.

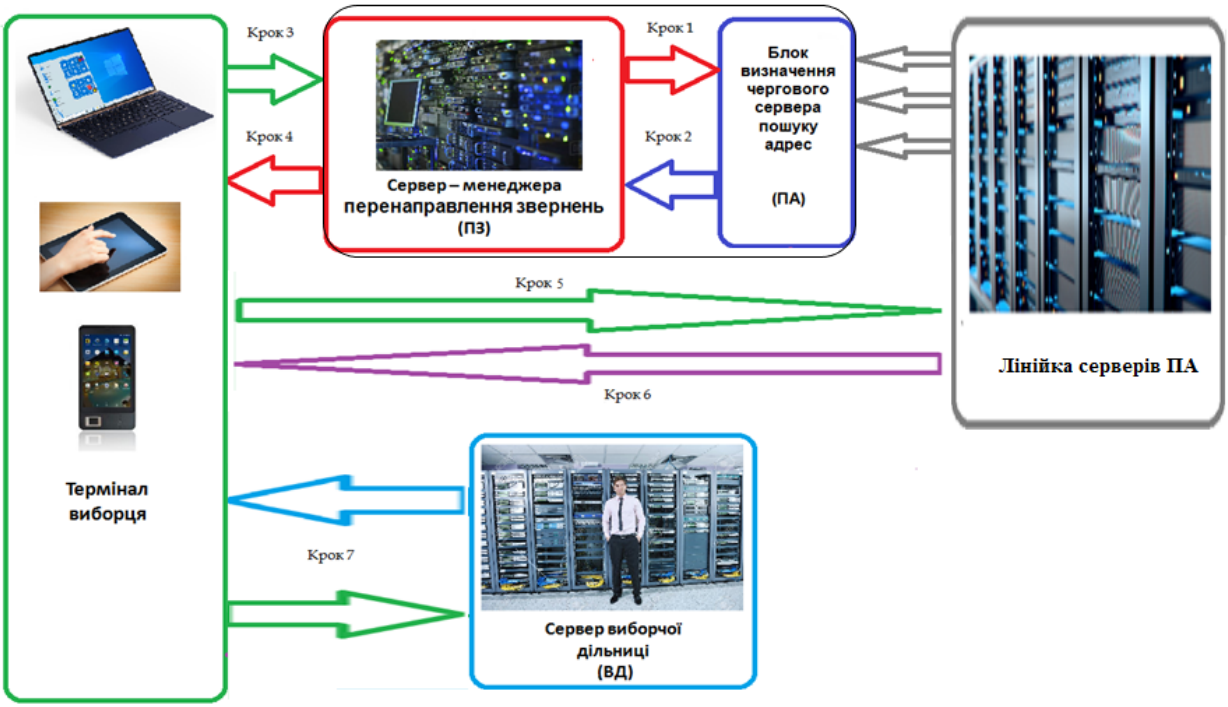



Рис. 2.1. Схематичне відображення протоколу інформаційної взаємодії засобів прискореного пошуку IP-адрес серверів ВД у сегменті системи ДТГ під час Інтернет- голосування

Як бачимо, перш ніж звернення виборця під час інтернет-голосування надійде на обробку до сервера відповідної виборчої дільниці (ВД), воно спочатку приймається сервером ПЗВ, який перенаправляє це звернення на обробку до одного із серверів ПА. Сервери ПА функціонують одночасно, незалежно один від одного. Кожен з них здійснює обробку отриманого звернення, а саме, здійснює пошук *IP*-адреси сервера ВД, до якої прикріплений виборець, що ініціював це звернення. При цьому виконується наступна послідовність дій:

1. Сервер ПЗВ звертається до блоку визначення чергового сервера ПА з метою отримання  ідентифікатора сервера ПА, котрий має обробити звернення

чергового виборця, що зробив запит на пошук IP-адреси сервера його виборчої дільниці у поточний момент часу. Блок визначення чергового сервера ПА фактично виконує функції механізму розподілу потоку звернень виборців між серверами ПА. Алгоритм його роботи має враховувати як пульсації трафіка звернень, так і випадковий характер часової координати процесу пошуку кожним із серверів ПА. В якості вихідних даних для такого урахування використовується інформація про зайнятість серверів ПА, яка постійно надходить на уводи цього блоку не протязі усього часу голосування. У нашому випадку, як буде показано далі, алгоритм роботи механізму розподілу розроблено з використанням теорії аналітичного конструювання регуляторів [122]. Результатом роботи даного блоку є ідентифікатор сервера ПА, котрий має обробити звернення чергового виборця. Даний блок можливо реалізувати у вигляді одної із програмних компонент сервера ПЗВ.

2. Сервер ПЗВ отримує від блоку визначення чергового сервера ПА ідентифікатор цього сервера, після чого сервер ПЗВ стає готовим для передачі *IP*-адреси сервера ПА на запит від термінала виборця. З цього моменту сервер ПЗВ буде знаходитись в режимі очікування запиту від виборця.

3. Сервер ПЗВ отримує очікуваний запит від чергового виборця.

4. Сервер ПЗВ відправляє *IP*-адресу чергового сервера ПА, яка визначена механізмом розподілу потоку звернень виборців, і одразу після цього готується до обслуговування нового запиту від чергового виборця, переходячи до виконання дії 1.

5. Отримавши *IP*-адресу сервера ПА, термінал виборця звертається за цією адресою до цього сервера ПА для отримання від нього в інтерактивному режимі *IP*-адреси сервера своєї виборчої дільниці. Визначений сервер ПА здійснює пошук *IP*-адреси сервера ВД, до якого прикріплений виборець, що здійснив звернення.

6. Термінал чергового виборця отримує від чергового сервера ПА *IP*-адресу сервера ВД, що обслуговує даного виборця.

7. Виборець розпочинає процедуру ДТГ в інтерактивному режимі на сервері своєї виборчої дільниці.

Кожен сегмент прискореного пошуку *IP*- адрес у системі ДТГ має один сервер ПЗВ та певним чином обрану кількість серверів ПА. Кількість цих серверів обирають виходячи із необхідності додержання вимог щодо припустимого рівня якості обслуговування запитів виборців під час голосування, а саме припустимого максимального значення часу очікування обробки запиту та очікуваної кількості втрачених запитів, що не пройшли процедуру обробки. Оскільки політика безпеки в домені ДТГ не дозволяє під час голосування змінювати склад програмно-апаратних засобів, то загальна пропускна спроможність сегменту системи ДТГ залежить від кількості серверів ПА. Тому актуальним є завдання раціонального вибору необхідної кількості серверів ПА у цьому сегменті. Однак вирішення цього завдання може бути вирішено лише за результатами експлуатації в реальних умовах використання системи.

## 2.3. Задача поділу потоку звернень виборців між серверами пошуку адрес

*Фізичні передумови доцільності створення технології розподілу потоку звернень*

Час пошуку *IP*-адреси кожним сервером, що входить до складу лінійки серверів ПА, залежить від багатьох непередбачуваних факторів і є випадковою величиною з невідомою щільністю ймовірності. Можуть виникнути ситуації, коли одні сервери перенавантажені обробкою звернень у той час, як інші є недозавантаженими.

Реалії виборчих кампаній показують, що інтенсивність потоку звернень виборців, що оброблюються серверами ПА, будуть швидко змінюватися у реальному часі непередбаченим чином. Під час сплесків потоку звернень навантаження на сервери адресації суттєво зростає. Якщо $s_1$, $s_2$, $s_3$, …, $s_n$ - це поточні значення інтенсивності пульсуючих потоків запитів від виборців, що

мають надходити на уводи серверів ПА в реальних умовах експлуатації, то має виконуватися умова

$$s_1 + s_2 + s + \ldots + s_n \leq \Delta F, \qquad (2.1)$$

де $\Delta F$ - загальна пропускна спроможність лінійки серверів ПА щодо обробки запитів від виборців.

$$\Delta F = f_1 + f_2 + f_3 + \ldots + f_n = const \ , \qquad (2.2)$$

де $f_1, f_2, f_3, \ldots, f_n$ - пропускні здатності серверів ПА.

Якщо умова (2.1) не виконується, то лінійка серверів ПА буде перенавантажена трафіком звернень, що призведе до неприпустимих рівнів втрат цих звернень. Тому має існувати механізм реагування на випадок порушення умови (2.1). Однак, навіть, якщо ця умова не порушується, але при цьому відсутня можливість швидкого вирівнювання навантаження між серверами ПА, то внаслідок швидких змін інтенсивності потока звернень завантаження серверів лінійки виявиться незбалансованим і окремі сервери будуть перенавантаженими з усіма витікаючими із цього негативними наслідками.

Щоб зменшити ймовірність перенавантаження серверів ПА при обраній їхній кількості, доцільно поставити та вирішити задачу раціонального розподілу потоку звернень виборців на обслуговування між серверами ПА за умови, що значення пропускної спроможності сегменту системи ДТГ, що розглядається, є сталою величиною, що обмежена умовою (2.1).

У зв'язку із вищезазначеним необхідно розробити технологію розподілу пропускної спроможності визначеного сегменту системи ДТГ між її серверами ПА, яка б враховувала як характер змін інтенсивності потоку звернень виборців, що надходять на вход (увід) сервера ПЗВ, так і ступінь завантаженості трафіком серверів ПА.

*Умови фізичного середовища використання розроблюваної технології*

Розроблювана технологія має реалізовувати механізм розподілу потоку звернень, що здатний функціонувати в умовах:

1) непрогнозованості сплесків (пульсацій) потоку звернень у період голосування;

2) невизначеності щодо тривалості обробки кожного окремого звернення кожним із серверів ПА, що призводить до швидко змінюваної неоднаковості ступеню їхнього завантаження роботою з пошуку IP-адрес.

*Загальний підхід до створення механізму розподілу*

Найбільш ефективною з точки зору балансування завантаженістю серверного обладнання є технологія адаптивного дискретно-аналогового управління.

Оскільки механізм розподілу має враховувати, з одного боку, непрогнозовану динаміку змін інтенсивності потоку звернень, а з другого боку, можливу неоднаковість навантаження на сервери ПА, що також є не прогнозованим фактором, то адаптивний механізм розподілу має, з одного боку, відслідковувати тенденції у змінах інтенсивності потоку звернень на вході сервера ПЗВ, а з іншого боку, підтримувати процес розподілу робочого навантаження між серверами ПА у напрямку вирівнювання поточних значень їхніх коефіцієнтів завантаження за зазначених вище умов. Процес відслідковування тенденцій у змінах інтенсивності потоком назвемо процесом авторегулювання, щоб відрізняти його від процесу вирівнювання коефіцієнтів завантаження серверів.

У нашому випадку під коефіцієнтом завантаження сервера ПА будемо розуміти відношення проміжку часу, коли цей сервер безпосередньо здійснює обробку звернень (а не простоює у чеканні наступного запиту на пошук IP-адреси), до тривалості часового інтервалу поточного кроку адаптації системи авторегулювання під нове виміряне значення інтенсивності потоку звернень, тобто коефіцієнт завантаження має відображатися значеннями безперервної безрозмірної величини у діапазоні від 0 до 1. Якщо тривалість кроку адаптації призначити як 1година, то тоді коефіцієнт завантаження сервера буде вимірюватися в *Ерлангах*. Проте у нашому випадку крок адаптації, як показує практика, доцільно призначати у діапазоні 2,5 – 5 хвилин (що залежить від поточної активності виборців).

### 2.4. Характеристики процесу авторегулювання

Для того, щоб відслідковувати тенденції у змінах інтенсивності потоку звернень, необхідно у реальному часі отримувати інформацію щодо змін у поточних значеннях інтенсивності цього потоку. Проте в умовах вільного волевиявлення інтенсивність пульсуючого потоку звернень не може бути визначена та виміряна миттєво. Існує теоретична можливість вимірювань характеристик цього потоку, якщо знати інформацію про його поведінку на попередніх відрізках часу. Але такі вимірювання мають сенс за умови, що на цих відрізках часу інтенсивність потоку запитів не змінюється. Окрім того, щоб зменшити похибки регулювання, бажано враховувати затримку на вимірювання. Однак на практиці такі умови не виконуються. Отже, при виборі методу реалізації процесу авторегулювання слід враховувати наведені вище міркування.

Трафік звернень у даній роботі доцільно розглядати як послідовність квазістаціонарних ділянок із реалізаціями дискретного випадкового процесу, що у певній мірі можуть бути згладженими за допомогою процедур усереднення [117]. У цьому випадку, як показано у [117], такий трафік буде придатним для обробки в системі адаптивного управління. Однак реальний потік звернень - це непередбачуваний нестаціонарний дискретний процес, оскільки на вхід сервера ПЗВ просувається майже не прогнозований швидко змінюваний нестаціонарний потік звернень. Тому технологія формування трафіка звернень має забезпечувати можливість перетворення цього нестаціонарного потоку на квазістаціонарні відрізки з проміжками стаціонарності, які узгоджені з інтервалами кроків регулювання тривалістю $\tau$. Тоді в якості моделі потоку на виході такого перетворювача можливо розглядати напівнескінчений відрізок стаціонарного випадкового процесу $X$ дискретного аргументу (часу) $t=0,1,\ldots,k,\ldots$, тобто часовий ряд $\{X_k;\ k=0;1;2;\ldots\}$, де $k$ - поточний номер часового інтервалу усереднення процесу $X$. Точкове значення $k$-го відліку часового ряду $\{X_k^{(\tau)};\ k=0;1;2;\ \ldots\}$ при моделюванні потоку інтерпретовано як кількість звернень $x_k^{\tau}$, що надійшли на сервер ПЗВ протягом *k-го* інтервалу часу тривалістю $\tau$. Тобто, у даному випадку $\tau$ - це інтервал усереднення звернень у

потоці. Якщо ряд $\{X_k^{(\tau)};\ k= 0;1;2;\ \ldots\}$ унормувати відносно τ, то отримаємо ряд $\{I_k^{(\tau)};\ k= 0;1;2;\ \ldots\}$, в якому $k$ -й компонент визначає поточну інтенсивність трафіка на $k$ -ому кроці його усереднення. Кількість звернень, що надійшли на сервер ПЗВ протягом *k-го* інтервалу часу тривалістю $\tau$, дорівнює максимально можливому значенню індексу $i_{max}$ , що задовольняє нерівності

$$i\Delta\tau_{k,i} = \tau$$
$$\tau \geq sum\ \Delta\tau_{k,i} = \Delta\tau_{k,1} + \Delta\tau_{k,2} + \ldots + \Delta\tau_{k,imax}, \qquad (2.3)$$
$$i=i+1$$

де $\Delta\tau_{k,i}$ - проміжок часу між сусідніми зверненнями у потоці, $i = 0,1,2, \ldots$ - поточний номер цього проміжку, а $k$ - поточний номер часового інтервалу усереднення процесу $\{X_k^{(\tau)};\ k= 0;1;2;\ \ldots\}$.

Отже, $k$-й компонент ряду $\{I_k^{(\tau)};\ k= 0;1;2;\ \ldots\}$, що визначає поточну інтенсивність трафіка на $k$-ому кроці його усереднення, визначено як

$$I_k^{(\tau)} = x_k^{\tau}/\tau. \qquad (2.4)$$

Звернемо увагу, якщо береться значення індексу $i_{max} \geq 2$, то маємо справу із усередненим процесом. Якщо ж розглядається послідовність моментів проходження одиничних звернень, то вважають, що $i_{max} = 1$.

Алгоритми згладжування трафіку та його перетворення у стаціонарні відрізки відповідних трендів, а також умови використання цих алгоритмів у задачах адаптивного керування (зокрема, похибки управління), що створені шляхом модифікацій відомого алгоритму «відра токенів» [117], детально висвітлені у [117] і у даній роботі не розглядаються. Підкреслимо тільки, що результати [117] дозволяють обґрунтувати вибір інтервалу усереднення трафіку звернень, а також і тривалості кроку авторегулювання, придатного для прийнятного відтворення процесу регулювання.

У нашому випадку доцільно обрати наступний показник ефективності процесу авторегулювання**:**

$$min\ \boldsymbol{Ц_i} = (s_1 - s^0_1)^2 + (s_2 - s^0_2)^2 + \ldots\ldots + (s_n - s^0_n)^2, \qquad (2.5)$$

де $Ц_i$ – **цільова функція** у задачі оптимізації, що враховує пульсації трафіку звернень; $s_1, s_2, s_3, \ldots, s_n$ - поточні значення інтенсивності пульсуючих потоків запитів від виборців, що мають буферізуватися у вхідних чергах серверів ПА на і-ому кроці авторегулювання в реальних умовах експлуатації; $s^0_1, s^0_2, s^0_3, \ldots, s^0_n$ - виміряні значення інтенсивності трафіка звернень на момент початку і-го кроку авторегулювання.

Розподіл потоку має здійснюватися за умови, коли кількість серверів ПА у складі сегменту ДТГ та значення їхньої пропускної здатності є заданими величинами. При цьому мають виконуватися умови (2.1) та (2.2).

Зрозуміло, що даний варіант реалізації процесу авторегулювання не є оптимальним, оскільки відслідковування тенденцій (траєкторії) щодо змін інтенсивності потоку звернень здійснюється без використання засобів прогнозування та не враховуються похибки, що пов'язані із затримками вимірювання.

### 2.5. Характеристики процесу вирівнювання керованих змінних

У даному випадку, окрім непередбачуваного процесу змін інтенсивності потоку звернень, маємо непередбачуваний швидко змінюваний потік різних щодо тривалості подій – сеансів пошуку серверами ПА IP-адрес серверів виборчих дільниць (ВД), до яких прикріплені виборці. Через це, якщо не застосовувати спеціально розроблених засобів, на кожному кроці авторегулювання $\tau$ рівні завантаженості серверів ПА будуть неоднаковими. І в діапазоні високих рівнів завантаженості усієї лінійки серверів ПА будуть підвищуватися ризики втрати звернень (через переповнення буферної пам'яти вхідних черг) на входах окремих серверів ПА, які на даному кроці регулювання були найбільш завантаженими. Так що в цих умовах на кожному кроці авторегулювання доцільно реалізувати процес вирівнювання поточних значень коефіцієнтів завантаження серверів з тим, щоб мати можливість функціонування системи розподілу звернень у діапазоні високих значень коефіцієнта навантаження на лінійку серверів ПА з мінімізованими ризиками

перенавантаження будь-якого із них. От же, щоб унеможливити ситуації перенавантаження серверів, доцільно у склад системи ДТГ увести додаткові програмні засоби, які б здійснювали функцію більш/менш рівномірного розподілу навантаження між серверами ПА. (Зрозуміло, що непередбачуваний характер швидко мінливого пульсуючого потоку звернень не дозволяє досягти на практиці повної однаковості завантаженості серверів ПА). Таку функцію у нашому випадку має виконувати спеціальний блок (програмний модуль) визначення чергового сервера ПА (див. рис.2.1), який прийнято називати регулятором системи вирівнювання значень керованих змінних – коефіцієнтів завантаження серверів ПА.

### 2.6.Синтез регулятора системи вирівнювання керованих змінних

У даній роботі здійснено оптимальний синтез регулятору системи вирівнювання шляхом його зведення до відомої формально вирішеної крайової задачі аналітичного конструювання регуляторів на мінімізацію функціонала Р.Беллмана у класі динамічних систем регулювання щодо об’єктів, що описуються звичайними лінійними диференціальними рівняннями настроювання першого порядку [102,118-122]. Під оптимальним синтезом у даному випадку розуміється конструювання регулятора, що здатний забезпечити сталу (і, от же, оптимальну) траєкторію змін стану об’єкту регулювання у фазовому просторі $C^2$ із заданими характеристиками якості перехідного процесу. (Простір $C^2$ – це простір неперервних гладких функцій, що двічі диференціюються). У разі роботи регулятора не по оптимальній траєкторії можливе виникнення неконтрольованого коливального процесу внаслідок утворення позитивного зворотнього зв’язку у контурі автоматичного регулювання (згідно Ляпунову). Синтезований регулятор відслідковує як динаміку змін інтенсивності вхідного потоку звернень (зокрема, використовуючи механізм, що реалізує відомий алгоритм відра токенів, згідно якого нестаціонарний потік перетворюється на послідовність часових відрізків сходинко подібного стаціонарного потоку із випадковими значеннями показника його інтенсивності), так і динаміку

перехідного процесу вирівнювання названих вище коефіцієнтів з метою мінімізації похибок регулювання та з урахуванням обмежень, що забезпечують сталість системи регулювання.

*Формальна постановка задачі синтезу регулятора*

У задачах адаптивного керування параметрами швидкозмінюваних непередбачуваних процесів керовані змінні та керуючі впливи прийнято представляти у векторній формі. Тим більш, у прийнятій постановці оптимізаційної задачі тенденції щодо змін інтенсивності оброблюваних потоків на серверах ПА слід відслідковувати як у бік збільшення, так і у бік зменшення.

Адаптивний механізм настроювання засобу розподілу потоків має забезпечувати можливість динамічного розподілу загального потоку запитів між серверами ПА з урахуванням як можливих трендів та пульсацій потоків запитів до цих серверів, так і непередбачуваних за величиною проміжків часу, необхідних цим серверам для пошуку IP-адрес серверів ВД. Так що, керовані змінні зручно представити у вигляді $n$ –мірного вектора-стовпця

$$\dot{k} = \begin{pmatrix} k_1 \\ k_2 \\ \cdot \\ \cdot \\ \cdot \\ k_n \end{pmatrix}, \qquad (2.6)$$

У свою чергу, керуючі впливи на регулятори системи адаптивного керування доцільно представити у вигляді $m$–мірного вектора керування $\acute{\boldsymbol{u}}$ :

$$\acute{\boldsymbol{u}} \in \{u_1, u_2, u_3, \ldots, u_m\}, \qquad (2.7)$$

де $u_1, u_2, u_3, \ldots, u_m$ - компоненти вектора керування.

У теорії адаптивного керування розглядаються також транспоновані вектори-стовпці, тобто вектори-рядки. $T$ – знак транспонування.

З урахуванням вище зазначеного поточний стан кожного сервера ПА характеризується його пропускною спроможністю з обробки запитів виборців, інтенсивністю потоку оброблених запитів і відповідними значеннями векторів прямого і зворотного коефіцієнтів завантаження цього сервера.

І якщо блок визначення чергового сервера ПА ( програма, що інстальована у складі сервера ПЗВ) діє у напрямку вирівнювання значень коефіцієнтів завантаження цих серверів, то процес авторегулювання адаптивним розподілом потокуу у загальному випадку можна відобразити у вигляді диференціального рівняння настроювання:

$$\dot{k} = \Phi_k , \qquad (2.8)$$

де $\Phi_k$ - функціонал управління процесом авторегулювання у неявному виді.

При цьому з урахуванням обмеження (2.2) сумарне значення компонентів вектора керованих змінних у процесі регулювання має підтримуватися на попередньо визначеному незмінному рівні.

Програмно-апаратний засіб адаптивного розподілу на кожному кроці регулювання здійснює інтегрування рівняння настроювання. У результаті поточні значення коефіцієнтів завантаження серверів ПА мають поступово вирівнюватися шляхом відповідного перерозподілу часток загального потоку запитів між серверами за умови неперевищення значення загальної незмінної пропускної спроможності системи ДТГ.

Таким чином, для вирішення задачі керування розподілом потоку запитів між серверами ПА на основі спостереження за рівнями їхнього завантаження у реальному часі необхідно вирішити задачу динамічного вирівнювання коефіцієнтів завантаження цих серверів. При цьому зручно використовувати один із різновидів методу динамічного програмування – метод аналітичного конструювання регуляторів, що запропонований у [101] і удосконалений у роботах [102, 104].

Формальна постановка задачі синтезу шуканого регулятора, зокрема визначення рівнянь вирівнювання та вектора керування у замкнутій векторно-

матричній формі, передбачає необхідність представлення функціоналу *I*, що зв'язує в одне ціле параметри процесу розподілу, умови та обмеження, що накладені на цей процес, виразом (2.6):

$$I = \int_0^{\infty} (N^T CPC^T N + \alpha N^T CQC^T N + u^T Ru)dt \text{ ,} \qquad (2.9)$$

де $\boldsymbol{N}$ – вектор керованих змінних, тобто вектор коефіцієнтів завантаження серверів ПА, оптимальну траєкторію змін котрого слід визначити у фазовому простору $C^2$; $\boldsymbol{C}$ – матриця регулюючих зв'язків між серверами ПА, значення котрої визначаються на кожному кроці чисельного інтегрування рівнянь вирівнювання; $\boldsymbol{P}$ – діагональна позитивно визначена матриця вагових коефіцієнтів при регулюючих зв'язках між серверами ПА (у даному випадку вони однакові та дорівнюють *1*); $\boldsymbol{Q}$ - $m \times m$- мірна позитивно визначена симетрична матриця квадратичної форми як складова функції Ляпунова/Беллмана, що впливає на швидкість та сталість перехідного процесу вирівнювання змінних; $\alpha$ - позитивна константа – показник загасання функції Беллмана (цей параметр підбирається експериментально з тим, щоб знайти компромісне співвідношення між швидкістю перехідного процесу та величиною помилок вирівнювання змінних); $\boldsymbol{u}$ - вектор керуючих впливів, що знаходиться як лінійна функція компонентів вектору $\boldsymbol{N}$; $\boldsymbol{R}$ - $m \times m$ - мірна симетрична позитивно визначена матриця вагових коефіцієнтів при керуваннях (у даному випадку вони однакові та дорівнюють *1*); $T$ - символ операції транспонування матриці; $[t_{0i}, t_{1i}]$ – часовий проміжок інтегрування, де $t_{0i}$ - момент початку процесу вирівнювання керованих змінних на і-му поточному часовому інтервалі усереднення потоку звернень, а $t_{1i}$ – момент закінчення цього інтервалу ($t_{1i} \to \infty$).

Проаналізуємо фізичний смисл складових підінтегрального виразу у функціоналі (2.9) стосовно нашої прикладної задачі.

Перший член підінтегрального виразу у функціоналі (2.9) являє собою зважену суму квадратів відмінків (через коефіцієнти матриці $\boldsymbol{P}$) вирівнювальних змінних. У [101] показано, що мінімізація значення функціоналу (2.9) приводить до вирівнювання змінних (з певними похибками), що є керованими. Другий член

функціоналу (2.9) є квадратична форма функції Ляпунова/Беллмана із показником загасання $\boldsymbol{\alpha}$, яка уведена у функціонал для визначення швидкості та обмеження процесу вирівнювання, що є важливим для забезпечення сталості цього процесу в умовах значних пульсацій трафіку звернень. Третій член функціоналу обмежує керування та одночасно сприяє формальному замиканню процесу вирівнювання. У даному випадку під вирівнюванням розуміється вибір траєкторії змін стану вектора коефіцієнтів завантаження лінійки серверів ПА, що забезпечує сталість системи вирівнювання.

Відповідно до відомих результатів теорії аналітичних регуляторів [100,101,122] у нашому випадку система настроювання має представлятися векторно-матричним диференційним рівнянням у вигляді:

$$\dot{N} = -CR^{-1}C^{T}CQC^{T}N\,, \qquad (2.10)$$

а вектор керування у вигляді (2.11):

$$u = -R^{-1}C^{T}CQC^{T}N \;;\quad u^{T} = -N^{T}CQC^{T}CR^{-1}. \qquad (2.11)$$

Як бачимо, вектор керованих змінних $\boldsymbol{N}$ пов'язаний з вектором керуючих впливів $\boldsymbol{u}$ через матрицю регулюючих зв’язків $\boldsymbol{C}$. Якщо дотримуватися визначених вище умов, та наведених нижче обмежень, то процес вирівнювання коефіцієнтів завантаження серверів буде здійснюватися по оптимальній траєкторії, що мінімізує значення функціоналу (2.9). На кожному часовому інтервалі усереднення потоку звернень $\boldsymbol{\tau}$ буде здійснюватися вирівнювання значень керованих змінних з похибками, що залежать як від тривалості цього інтервалу, так і від значення показника загасання $\boldsymbol{\alpha}$. Бажані значення часового інтервалу усереднення потоку та показника $\boldsymbol{\alpha}$ обираються експериментально.

Теорія аналітичних регуляторів [100,101,122] для нашої системи вирівнювання керованих змінних *визначає наступні обмеження*:

1. Функція Беллмана повинна мати наступний вигляд:

$$V = N^{T}CQC^{T}N \quad . \qquad (2.12)$$

2. Визначення матриці $Q$, що входить до складу функції Беллмана, має здійснюватися шляхом рішення рівняння Ріккаті (2.13), яке у даному випадку має виконуватися для будь-яких значень винесених за дужки множників:

$$N^T C \cdot (P + \alpha Q - QC^T CR^{-1} C^T CQ) \cdot C^T N = 0. \qquad (2.13)$$

3. Поточне значення інтенсивності потоку звернень у будь-який момент має не перевищувати величину загальної пропускної здатності серверів ПА.

В якості *початкових умов роботи механізму вирівнювання* задається кількість серверів ПА у лінійці та значення показника загасання $\alpha$.

У результаті «роботи» рівнянь вирівнювання (відповідно з будь-яким методом чисельного інтегрування) з фізичної точки зору значення коефіцієнтів завантаження серверів ПА у реальному часі крок за кроком будуть вирівнюватися шляхом відповідного перерозподілу часток загального потоку звернень виборців між серверами ПА.

Примітка. Зазвичай у прикладних задачах вирівнювання керованих змінних використовується векторно-матрична форма представлення системи вирівнювання. Проте такі системи можливо представити у формі Коші, тобто як систему лінійних диференціальних рівнянь першого порядку.

### 2.7. Схема рішення задачі побудови регулятора

Рішення даної задачі передбачає необхідність визначення конкретного вигляду рівняння настроювання системи вирівнювання керованих змінних, вектору керуючих впливів та взаємозв'язків між усіма змінними (і керованими, і керуючими) з урахуванням обмежень на їхні значення з тим, щоб система вирівнювання не втратила сталість у процесі своєї роботи. У відповідності з теорією аналітичних регуляторів [101] взаємозв'язки між керованими та керуючими змінними задаються за допомогою матриці регулюючих зв'язків, а сталість системи забезпечується шляхом дотримання у процесі регулювання обмежень, що накладаються функціоналом Беллмана, оскільки цей функціонал зв'язує в одне ціле параметри процесу розподілу потоку звернень з умовами та обмеженнями, що накладені на цей процес.

Матриця регулюючих зв’язків (МРЗ) з фізичної точки зору встановлює: скільки ресурсу на кожному кроці регулювання треба відняти від одного сервера та передати його іншому серверу. І таке встановлення зробити для кожної пари серверів, що складають лінійку серверів ПА. У даному випадку під ресурсом розуміється частка потоку звернень, що передається з вхідного буферу одного сервера до іншого. Згідно [101] МРЗ задається у вигляді

$$C = \begin{pmatrix} 1 & 0 & -1 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix}, \qquad (2.14)$$

для побудови якої обирають наступне правило. Кількість стовпців у матриці має дорівнювати кількості серверів ПА у лінійці, а кількість рядків - максимально можливому числу пар цих серверів за умови, що в кожній парі номерний індекс хоча б одного сервера відрізняється від номерних індексів серверів у будь-якій іншій парі. Усі пари серверів мають бути різними за номерними індексами, але містити сервери з однаковими номерами. Таким чином встановлюються регулюючі зв'язки між будь-якими двома серверами ПА. Напрямок передавання частки потоку звернень типу «від сервера» у матриці (2.14) відображається знаком мінус. Згідно наведеному правилу напрямок передавання типу «у сервер» відображається знаком плюс. Нуль означає, що перерозподіл потоку між серверами у парі не змінюється. Інтерпретація змісту першого рядка матриці (2.14) така: між третім і першим серверами реалізується зв'язок, що полягає у передаванні ресурсу (тобто, певної частки потоку запитів) із третього сервера у перший, що відбито знаками одиничних елементів у матриці (2.14). Другий рядок відповідно відображає передачу частки потоку запитів із першого сервера у другий. Третій рядок відповідає передачі ресурсу з другого сервера у третій. Між трьома серверами (оскільки у наведеному прикладі маємо три стовпці) можна попарно встановити три зв'язки (тобто, утворити три різних пари), що в конструкції матриці відображається трьома

рядками. Відзначимо, що для відображення взаємозв'язків між чотирма і більше серверами матриця перестає бути квадратною, оскільки кількість пар з, наприклад, чотирьох серверів дорівнює шести, а з п'яти серверів – десяти і т.д..

Оптимізуючий функціонал Беллмана *I* слід задати у вигляді (2.9).

У нашому випадку процес вирівнювання згідно (2.10) та (2.11) є процесом вирівнювання поточних значень коефіцієнтів завантаження серверів ПА. Цей процес детально розглянутий у [102], де для випадку трьох керованих змінних показано, що

$$\vec{N}^T C P C^T \vec{N} = (n_1 - n_2,\ n_2 - n_3,\ n_3 - n_1) \cdot \begin{pmatrix} p_{11} & 0 & 0 \\ 0 & p_{22} & 0 \\ 0 & 0 & p_{33} \end{pmatrix} \cdot \begin{pmatrix} n_1 - n_2 \\ n_2 - n_3 \\ n_3 - n_1 \end{pmatrix} = \tag{2.15}$$

$$= p_{11}(n_1 - n_2)^2 + p_{22}(n_2 - n_3)^2 + p_{33}(n_3 - n_1)^2.$$

У (2.1) вектор керованих змінних *N* представлений у вигляді вектора-стовпця:

$$\vec{N} = \begin{pmatrix} n_1 \\ n_2 \\ n_3 \end{pmatrix}, \tag{2.16}$$

*P* - діагональна позитивно визначена матриця вагових коефіцієнтів (у нашому випадку значення її ненульових компонентів дорівнюють одиниці) виду:

$$P = \begin{pmatrix} p_{11} & 0 & 0 \\ 0 & p_{22} & 0 \\ 0 & 0 & p_{33} \end{pmatrix}. \tag{2.17}$$

Аналіз отриманого добутку у (2.15) показує, що він являє собою зважену суму квадратів відмінків різних компонент вектора керованих перемінних. І ця сума дорівнює нулеві тільки у випадку, якщо всі перемінні дорівнюють нулю.

Це означає, що перший член підінтегрального виразу у функціоналі (2.9) відображає процес вирівнювання поточних значень коефіцієнтів завантаження серверів ПА (які є керованими змінними). Він являє згідно (2.15) зважену суму квадратів відмінків (через коефіцієнти матриці $P$) вирівнювальних змінних. Чим ця сума більша або чим довше за часом здійснюється процес вирівнювання, тим буде більше значення оптимізуючого функціоналу. Тому мінімізація значення функціоналу приводить до вирівнювання змінних, що є керованими.

Другий член функціоналу у виразі (2.9) відображає функцію Беллмана, яка уведена у функціонал у вигляді складової із заданим показником загасання α. Функція Беллмана $\boldsymbol{V}$ у виразі (2.12) представлена у вигляді квадратичної форми.

Звісно [99-102], що на оптимальних траєкторіях при мінімальних значеннях функціоналу $\boldsymbol{I}$ функція Беллмана $\boldsymbol{V}$ убуває із швидкістю підінтегрального виразу цього функціоналу. У нашому випадку функція Беллмана на оптимальних траєкторіях убуває із швидкістю не меншій за здобуток її самої та показника загасання $\alpha$ . Тому, шляхом вибору достатньо великого значення $\alpha$ є можливим забезпечити за умов замкнутої системи вирівнювання швидкодію не меншу за таку, що визначається показником загасання.

Третій член функціоналу відображає роботу сервера ПЗВ щодо формування керуючих впливів на перерозподіл потоку і повністю співпадає із відповідним членом при звичайній постановці задачі аналітичного конструювання регуляторів [102]. Його уведення дозволяє обмежити керування та одночасно сприяє формальному замиканню процедури визначення оптимального керування.

Розглянемо рівняння настроювання. Поводження компонентів системи адаптивного вирівнювання керованиз змінних, як відомо [102], має здійснюватися згідно наступного рівняння настроювання:

$$\dot{N} = Cu\,. \qquad (2.18)$$

У виразі (2.18) $\boldsymbol{u}$ - вектор керуючих впливів, що знаходиться як лінійна функція компонентів вектору керованих змінних $N$, а $C$ - матриця регулюючих зв'язків, зміст якої роз'яснено вище (див. вираз 2.14).

Рівняння настроювання (2.18) сформовано таким чином, що забезпечує аперіодичний не коливальний характер зміни керованих перемінних, тобто поточних значень коефіцієнтів навантаження серверів ПА трафіком запитів виборців під час здійснення актів дистанційного голосування. Згідно теореми про сталість (рос. – устойчивость) [52] рівняння настроювання (2.18) при будь-якому довільному векторному керуванні забезпечує під час настроювання збереження суми компонент вектору $N$.

Результати теорії побудови аналітичних регуляторів у динамічних системах адаптивного регулювання показують: якщо маємо рівняння настроювання у вигляді (2.18), то у наведеній вище постановці задачі вирівнювання вектор керування буде мати вигляд (2.11).

От же, підставляючи керування (2.11) у рівняння настроювання (2.18), одержимо замкнуту систему настроювання (систему вирівнювання) у вигляді (2.10).

Підкреслимо, що замкнута система адаптивного розподілу потоку, що функціонує згідно (2.10) та (2.11), за будь-яких початкових значень вектору керованих перемінних $\boldsymbol{N}$ у процесі настроювання завжди намагається досягти такого сталого значення керованого вектору, коли його компоненти є однаковими, а їхня сума протягом усього процесу настроювання є незмінною.

Відмітемо: наближення до режиму однаковості значень коефіцієнтів навантаження серверів створює ситуацію, коли однакові за відсотком пульсації потоків запитів на різних серверах ПА не призводять до втрат запитів, якщо величини цих пульсацій знаходяться в межах запасу пропускної здатності системи ДТГ.

### 2.8. Варіант рішення з побудови регулятора

В основу роботи механізму вирівнювання значень коефіцієнтів завантаження покладено покрокове інтегрування диференціального рівняння

настроювання, яке у реальному часі відображає траєкторію прийнятого алгоритму розподілу загальної пропускної спроможності системи ДТГ між серверами ПА. Так що, внаслідок інтегрування поточні зміни потоків впливають на величини коефіцієнтів завантаження серверів ПА. От же, диференціальне рівняння настроювання забезпечує відстеження тенденцій у змінах інтенсивності оброблених потоків. Слід сподіватися, що у рамках наведених вище обмежень сформульована задача буде змістовною, замкнутою і внутрішньо несуперечливою. У результаті її рішення отримаємо зменшення ймовірності перенавантаження серверів ПА трафіком, що надходить до їхніх портів, і, таким чином, зменшення величини коефіцієнту втрат необроблених запитів від виборців при заданій кількості серверів ПА у складі системи ДТГ.

Процес адаптації може здійснюватися шляхом застосування будь якого чисельного методу інтегрування диференціальних рівнянь. Чисельне інтегрування рівняння настроювання на одному кроці може бути виконано, наприклад, згідно з методом Ейлера [97]:

$$k(t+h) = k(t) + \Phi_k(t)h \qquad (2.19)$$

- для прямих коефіцієнтів завантаження цих серверів.

У виразах (2.19) $h$ – крок чисельного інтегрування. (Відзначимо, що якщо інтегрування виконується іншими чисельними методами, то співвідношення (2.19) буде приймати інший вигляд.

Результатом інтегрування на поточному кроці рішення рівняння (2.10) є новий набір коефіцієнтів завантаження, значення котрих відповідно до вибраного рівняння настроювання між собою будуть відрізнятися менше, ніж до виконання даного кроку інтегрування.

Зрозуміло, що новим значенням коефіцієнтів завантаження, отриманим на даному кроці інтегрування, відповідають нові рекомендовані значення часток від інтенсивності загального потоку запитів, що мають бути спрямовані на обробку відповідними серверами ПА. Ці частки обчислюються виходячи з нормування на сталість їхньої суми, яка за величиною дорівнює пропускній здатності системи ДТГ $\Delta F$.

Наприклад, у випадку, коли в системі ДТГ використовуються три сервери ПА, нові рекомендовані значення часток від інтенсивності загального потоку запитів задаються наступними виразами:

$$s_1 = \frac{f1 + f2 + f3}{F} k_1 f_1 \ ;$$

$$s_2 = \frac{f1 + f2 + f3}{F} k_2 f_2 ; \qquad (2.20)$$

$$s_3 = \frac{f1 + f2 + f3}{F} k_3 f_3 .$$

Структура виразів (2.20) відображає умову, що сума ширин смуг пропускання усіх трьох серверів дорівнює загальній пропускній здатності домену системи ДТГ, тобто виконується співвідношення (2.1), що є умовою збереження загальної пропускної здатності системи ДТГ під час роботи механізму настроювання.

От же, поставлена задача оптимального розподілу потоку звернень виборців на обслуговування між серверами адресації є вирішеною.

Таким чином, можливий варіант проектного рішення з технічної реалізації механізму оптимального розподілу полягає у наступному. На кожному кроці перерозподілу потоку звернень між серверами ПА шляхом інтегрування диференціального рівняння (2.10) визначається поточне значення матриці регулюючих зв'язків *C*. Для прикладу розглянемо випадок, коли лінійка серверів ПА містить три сервери. Припустимо, що значення матриці регулюючих зв'язків є таким, що показано у виразі (2.14). Тоді перший рядок цієї матриці означає, що на даному кроці інтегрування частка потоку звернень має передатися від третього сервера на перший, що відбито знаками одиничних елементів у матриці (2.14). При цьому частка потоку звернень, що надходить на другий сервер має не змінюватися. Програмно такий механізм можливо реалізувати наступним чином. Загальний потік звернень розбивається на триади. Триада на поточному кроці інтегрування розподіляється між серверами так, що на перший сервер ПА надійде два звернення, на другий сервер – одне звернення, а на третій

нічого не надійде. Другий рядок матриці  задає таке правило розподілу: на перший сервер ПА не надійде нічого, на другий сервер надійде два звернення, а на третій сервер – одне звернення. І т.д.

Шляхом комп'ютерного імітаційного моделювання подібної системи настроювання, що здійснено іншими дослідниками [116], показано, що якщо для некерованої системи розподілу звернень досяжним для лінійки серверів ПА є коефіцієнт завантаження 0,323, то для системи розподілу, замкнутої регулятором, можливий коефіцієнт завантаження може досягати значення 0,886. Названі кількісні показники отримано для випадку, коли пульсації трафіку моделювались з використанням відомої підпрограми генерації випадкових чисел, що рівномірно розподілені у довільно обраних межах.

**Висновки до другого розділу**

2.1. За результатами аналізу існуючих методів забезпечення доступу виборців до інформаційних ресурсів системи інтернет-голосування зроблено висновок про доцільність розробки засобу автоматизації пошуку *ip*-адрес серверів виборчих дільниць, котрі мають надаватися виборцям за їхніми запитами для здійснення актів дистанційного голосування. Автоматизація процедури пошуку суттєво підвищує зручність користування послугами системи інтернет-голосування і зацікавленість виборців здійснювати дистанційне голосування.

2.2. Для автоматизації пошуку *ip*-адрес серверів ВД запропоновано використати спеціалізований сервер ПА, програмні засоби котрого у відповідь на звернення виборців у реальному часі мають реалізовувати процедуру пошуку *ip*-адрес потрібних серверів ВД відповідно до будь-якого із відомих методів пошуку, а знайдені значення *ip*-адрес надсилати на термінальні вузли ініціаторів звернень.

2.3. Оскільки швидкість пошуку *ip*-адрес, що здійснюються ПА, є набагато нижчою у порівнянні із швидкістю обробки звернень серверами ВД, то з метою

уникнення довготривалих затримок у доступі виборців до ресурсів системи інтернет-голосування у склад її обладнання слід увести не один, а кілька серверів ПА, що мають функціонувати одночасно і незалежно один від одного. У цьому випадку загальна продуктивність системи голосування буде визначатися сумарною продуктивністю серверів ПА.

2.4. Під час сплесків трафіка виборців, що має випадковий і непередбачуваний характер, а також внаслідок випадкової тривалості часу пошуку *ip*-адрес серверами ПА, можуть виникати ситуації, коли частина серверів ПА перенавантажуються цим трафіком у той час, як інші ПА функціонують недовантаженими. Тому, щоб мінімізувати ймовірність перенавантаження серверів ПА при певним чином обраній їхній кількості, запропоновано нову модель доступу виборців до ресурсів серверного обладнання системи ДТГ, зокрема у склад обладнання уведено додатковий засіб – сервер-менеджер ПЗВ (перенаправлення звернень виборців), котрий у реальному часі має здійснювати адаптивний розподіл потоку звернень між серверами ПА з тим, щоб забезпечити рівномірне завантаження серверів в умовах непрогнозованих сплесків (пульсацій) цього потоку під час голосування. З урахуванням вище наведеного розроблено протокол взаємодії засобів технічної підтримки процедур інтернет-голосування з прискореним пошуком ір-адрес серверів виборчих дільниць як складова частина відомого протоколу процедури інтернет-голосування.

2.5. Розроблено структурно-функціональну модель та технологію розподілу потоку запитів виборців на обслуговування між серверами пошуку адрес ПА. Оскільки пульсації потоку звернень та час пошуку *ip*-адрес серверами ПА мають непередбачуваний характер, запропоновано використати адаптивний принцип роботи сервера ПЗВ, що має здійснювати оптимальний розподіл потоку. Цей засіб має у реальному часі відслідковувати динаміку змін інтенсивності оброблюваних потоків на ПА і діяти у напрямку вирівнювання значень коефіцієнтів завантаження цих серверів з урахуванням як можливих трендів та пульсацій потоку запитів виборців, так і непередбачуваних за величиною проміжків часу, необхідних серверам ПА для пошуку *ip*-адрес СВД.

2.6. Задача керування розподілом потоку запитів між серверами ПА представлена як задача динамічного вирівнювання коефіцієнтів завантаження цих серверів. При цьому використано один із різновидів методу динамічного програмування – метод аналітичного конструювання регуляторів. Цей метод враховує якість перехідних процесів у динамічно керованих системах.

2.7. Синтезовано диференціальне рівняння настроювання, що на формальному рівні відображає процес авторегулювання динамічним перерозподілом потоку запитів між серверами ПА. Запропоновано схему рішення цього рівняння, зокрема для чисельного інтегрування рівняння настроювання на кожному кроці використано метод Ейлера.

2.8. Під час аналітичного конструювання регуляторів побудована відповідна функція Беллмана. Це дало змогу конкретизувати рівняння настроювання, задати оптимізуючий функціонал і записати відповідне їм рівняння Беллмана.

2.9. Задачу аналітичного конструювання регуляторів зведено до рішення рівняння Ріккаті – матричного квадратного рівняння, необхідного для пошуку матриці функції Беллмана. Шляхом підстановки знайденої матриці у вираз для керування одержано остаточний вираз для шуканих регуляторів.

# РОЗДІЛ 3
# УДОСКОНАЛЕННЯ МЕТОДІВ ЗАХИСТУ СИСТЕМИ ДИСТАНЦІЙНОГО ТАЄМНОГО ІНТЕРНЕТ-ГОЛОСУВАННЯ

## 3.1. Характеристики атаки *MITM* та можливості протидії цій атаці

Атака посередника, котру називають *MITM* (*Man In The Middle*), є найбільш небезпечною серед загроз для транспарентних систем ДТГ. Реалізація такої атаки, за умов відсутності або недостатньої ефективності механізмів захисту, може призвести, до порушення таємниці голосів, а також до підміни результатів волевиявлення.

Особливість реалізації атаки посередника у транспарентних системах ДТГ полягає в тому, щоб надавати контролюючим особам сфальсифіковану картину нібито нормально працюючого сервера інтернет-голосування (зазвичай, це сервер виборчої дільниці (ВД) системи ДТГ), а запити виборців відправляти на сервер, на якому може бути встановлена програма, що розкриває конфіденційну інформацію про голоси та підміняє результати волевиявлення.

Враховуючи особливу значимість шкідливих наслідків від реалізації загрози *MITM*, вкрай бажано надати кожному виборцю можливість без зайвих зусиль проконтролювати той факт, що на шляху до сервера його виборчої дільниці не встановлені позаштатні засоби зловмисників. Така можливість спрямована на надання виборцям впевненості у тому, що таємниця їх голосу буде збережена, а результати не будуть викривлені. Реалізації саме такої можливості присвячено матеріал даного підрозділу.

Слід зазначити, що в останнє десятиліття досвід створення систем ДТГ активно обговорюється на міжнародних конференціях [6,23,28], а в деяких країнах, як, наприклад, в Естонії, системи ДТГ вже набувають популярності [19,21], але існує також і протидія їхньому впровадженню з боку громадян, які не вірять в можливість створення прозорих систем ДТГ [35,40]. Безумовно, голосування через Інтернет надає суттєві переваги виборцям щодо зручності, мобільності та економії часу, але недовіра буде існувати до того часу, поки

громадяни не зможуть впевнитись у тому, що в інформаційно-телекомунікаційних засобах для голосування не існує можливості для розкриття таємниці голосів та/або викривлення результатів волевиявлення. У попередньому розділі показано, що для подолання недовіри треба надати можливість широкому загалу контролювати усі об'єкти і процеси, які щодо рівню захищеності викликають сумніви. Для досягнення беззаперечної довіри необхідно надати усім бажаючим можливість контролювати усі складові системи ДТГ протягом усього часу її функціонування. Саме такий підхід запропоновано в роботах [1-6], де описані принципи побудови систем ДТГ, у яких надається можливість масового дистанційного контролю з боку необмеженої кількості будь-яких осіб щодо усіх програмних засобів та процесів в режимі реального часу. В роботі [1] показано, що після проведення такого контролю не залишається підстав для недовіри, бо всі елементи системи і дії обслуговуючого персоналу, які можуть бути потенційно небезпечними, є відкритими для масового спостереження. Іншими словами, будь-яка спроба вчинення зловмисної дії у такій системі може бути виявлена та зафіксована контролюючими особами. Скоріш за все, неможливо досягти справжніх успіхів щодо впровадження ДТГ без наявності такого контролю.

Наведений в роботі [1], повний перелік загроз, які можуть стати причиною порушення таємниці голосів виборців або вплинути на вірність підрахунку, представлено у табл. 3.1 і табл.3.2.

**Таблиця 3.1**

**Загрози, які можуть виникати поза сервером ВД**

| Опис загрози | Метод протидії |
|---|---|
| 1. Перехоплення даних у середовищі передавання | Створення досконало захищених каналів |
| 2. Заміна даних під час передавання | Використання протоколів, які досконало захищають цілісність даних |
| 3. Проникнення до серверу через засоби дистанційного доступу | Усунення можливості проникнення до серверу з правами повного доступу |
| 4. Заміна даних про результат | Порівняння даних з довідками, |

| | |
|---|---|
| голосування | отриманими через досконало захищений канал |

**Таблиця 3.2**

**Загрози, які може утворювати адміністратор сервера ВД**

| Опис загрози | Метод протидії |
|---|---|
| 1. Фальсифікація операційної системи | Порівняння файлів ОС зі штатними |
| 2. Виконання позаштатної команди управління | Контроль введення команд управління |
| 3. Фізична заміна сервера | Контроль параметрів процесів ОС |
| 4. Фальсифікація прикладного ПЗ | Порівняння текстів ПЗ зі штатними |
| 5. Несвоєчасне виконання штатних дій | Перевірка дій за регламентом |
| 6. Підключення позаштатних засобів з метою реалізації атаки посередника | Контроль характеристик трафіку |

Загрози з табл. 3.1 можуть бути усунуті криптографічними, програмними та апаратними засобами захисту інформації, а загрози з табл. 3.2, реалізація котрих пов’язана з помилками або зловмисними діями обслуговуючого персоналу, крім програмно-апаратних засобів вимагають ще й адміністративних заходів.

Методи, що наведені у табл. 3.2, дозволяють користувачам Інтернету виявляти і документувати загрози, але після цього треба приймати адміністративні рішення щодо порушників, бо інакше немає шансів на отримання бажаних чесних результатів волевиявлення. Іншими словами, якщо повідомлення про виявлені загрози будуть проігноровані, то виборцям залишається відмова від голосування і вихід на акцію протесту.

Надалі припустимо, що зроблено все необхідне для того, щоб після виявлення загрози приймались правильні рішення.

Проаналізуємо кожен з методів протидії на наявність «слабких місць» [56-57] і можливих ускладнень під час їх реалізації. Протидія фальсифікації ОС шляхом порівняння файлів вимагає від контролюючих наявності додаткових

комп'ютерів для встановлення такої ж ОС, як на сервері ВД (в нашому випадку це *Open BSD*). Порівняння файлів займає близько години, але для підтвердження справжності ОС достатньо двом-трьом незалежним групам активістів виконати таке порівняння і викласти в Інтернеті повідомлення про відсутність небезпечних розбіжностей, а також комусь із них опублікувати результат виконання команди ***ps -aux*** у вигляді, який показано на рис. 3.1.

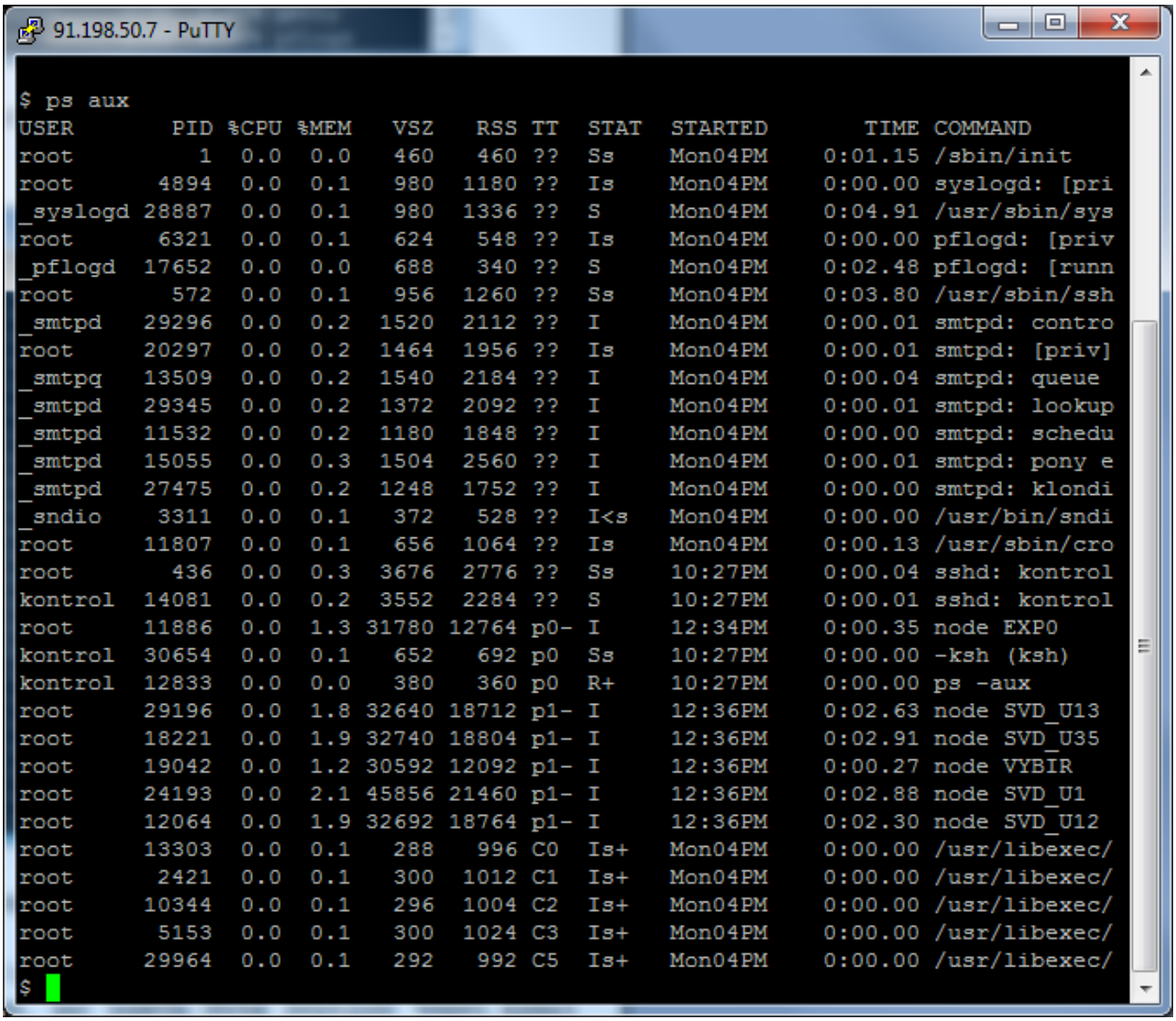



Рис. 3.1. Результат виконання команди ***ps -aux***

Виконання цієї команди дозволяє користувачам Інтернету упевнитись, що встановлена ОС на сервері ВД є штатною, а також виявляти загрози, що описані у рядках 2-5 табл. 3.2. Для цього треба увійти на сервер ВД з доступними для усіх правами контролера і виконати команду ***ps aux***, після чого слід порівняти значення 20 чисел у стовпчику ідентифікаторів процесів *PID* між своїм і

опублікованим результатами виконання даної команди. Ці 20 чисел знаходяться у рядках, які у стовпчику *STARTED* мають однакові значення (в нашому випадку *Mon04PM*). Протягом усього часу роботи сервера ці 20 цілих чисел не повинні змінюватись.

Методи протидії загрозам, що описані у рядках 2-5 табл. 3.2, не потребують значних витрат часу і можуть бути виконані широким колом користувачів Інтернету. Наприклад, для тих, хто користується ОС *Windows*, достатньо встановити на своєму комп'ютері безкоштовне ПЗ типу *PuTTY* та/або *WinSCP* і виконувати нескладні дії зі спостереження за процесами на сервері ВД у визначені моменти технологічного циклу виборчого процесу, який представлено на рис.3.2.

На рис. 3.2 сірим фоном виділено процеси на сервері ВД, а також прийнято наступні скорочення:

ППД – період підготовки даних про претендентів на роботу у режимі ДТГ;

ППС – період підготовки сервера ВД (встановлення ОС та програм загального користування);

ППЗ – прикладне програмне забезпечення для дистанційного голосування;

ПВЕБ – період введення електронних бюлетенів (в цей період запити виборців сервером не обслуговуються, а в списках виборців помічають тих, хто голосуватиме дистанційно, щоб не видавати їм паперові бюлетені).


Рис. 3.2. Технологічний цикл виборчого процесу у відкритій системі ДТГ

Контроль введення команд управління розпочинають у наперед визначений організаторами голосування момент в межах періоду встановлення ППЗ. В цей момент у трьох рядках стовпчика *USER* з'явиться слово *admin*, а також з'явиться рядок зі значенням *sshd: admin* у стовпчику *COMMAND*, що означає початок роботи адміністратора, який повинен занести в директорію *home*/*admin* такі три файли ППЗ:

*VDn.js* – серверна прикладна програма;

*VDn.DBT* – дані про виборців для зчитування серверною програмою;

*PWn.html* – клієнтська програма введення паролів для голосування.

Замість букви *n* у назвах файлів проставляється номер виборчої дільниці. Конфіденційні дані про виборців у файлі *VDn.DBT* надаються у зашифрованому вигляді.

Після запуску сервера до початку роботи адміністратора повинно пройти достатньо часу, щоб активісти мали змогу (через порівняння файлів) впевнитись у відсутності будь-яких підробок серверного ПЗ. Після занесення перелічених вище трьох файлів до моменту запуску ППЗ також повинно бути достатньо часу для перевірки активістами змісту цих файлів. Запуск ППЗ супроводжується появою процесу, який відображується рядком з такими значеннями параметрів:

*USER*: *admin*;

*PID*: x (x – ціле число, яке не повинне змінюватись до кінця роботи сервера);

*STARTED*: ГГ:ХХ (час запуску ППЗ, ГГ – години; ХХ – хвилини);

*COMMAND*: *node VDn* (*n* - номер виборчої дільниці).

Цей рядок повинен залишатись незмінним до кінця роботи сервера ВД.

Після запуску ППЗ адміністратор повинен завершити свою роботу командою ***exit***. Після цього управління сервером буде виконувати виключно прикладна програма. Задача активістів в цей період полягає у тому, щоб виявляти появу нештатних процесів і у разі їх появи засвідчувати цей факт. При цьому можуть виникати в будь-якій кількості процеси з параметрами:

*USER*: *kontrol*;

*COMMAND*: *sshd*: *kontrol.*

Ці процеси слід залишати поза увагою, бо вони пов’язані з початком роботи активістів, які через обмеженість прав доступу не можуть утворювати загрози.

У період введення електронних бюлетенів на час занесення в директорію *home*/*admin* файлу *AVn.html* (це клієнтська програма голосування електронними бюлетенями) також повинні з’явитись чотири процеси з параметрами:

*USER*: *admin* (3 рядки);

*COMMAND*: *sshd*: *admin* (1 рядок).

Ці процеси свідчать про виконання адміністратором своєї штатної дії. Файл *AVn.html* також підлягає перевірці. Ніяких інших процесів на сервері ВД за весь час спостереження не повинно з’являтись. Оскільки будь-яке зловмисне втручання в роботу сервера обов’язково потребує встановлення і запуск додаткової програми, а це не може бути невідображеним, як у файловій системі, так і в переліку активних процесів. Тому відсутність позаштатних процесів протягом усього періоду спостереження свідчить про те, що сервер працював виключно у штатному режимі і ніяких вдалих спроб втручання в його роботу не було. У разі виявлення позаштатного файлу або процесу до моменту запуску ППЗ існує можливість виправлення небезпечної ситуації шляхом повторного виконання усіх дій на сервері ВД. При цьому період введення паролів, який має тривалість близько двох тижнів (оскільки він збігається в часі з періодом уточнення списків голосуючих [12]), може бути скорочено на декілька годин (або навіть на добу) без суттєвого впливу на процес голосування у цілому. Також можливо подібним чином виправити ситуацію у разі виявлення порушень на початку періоду введення паролів. У разі виявлення порушень у більш пізні часи, залишається тільки відміняти результати волевиявлення щодо конкретної групи учасників ДТГ і проводити для цієї групи повторне голосування.

Слід зауважити, що складнощі у роботі активістів під час спостереження за роботою сервера ВД можуть бути усунені шляхом автоматизації. Дії контролерів не є складними, але потребують постійної уваги, бо через тимчасову неуважність активістів загроза може залишитись непоміченою. За допомогою автоматизації процесу спостереження цілком можливо створення таких засобів перевірки, коли

жодна із загроз, що описані у рядках 1-5 табл. 3.2, не буде мати шансів залишитись непоміченою.

Але відносно загрози *MITM*, яка описана у рядку 6 цієї таблиці, процедура виявлення не виглядає простою. Для реалізації цієї загрози зловмисники можуть скористатись тим, що запити контролерів і виборців відправляються у різні моменти часу. Тому необхідно, щоб кожен виборець мав би можливість виявити загрозу саме під час голосування.

Виявлення даної загрози пов'язане з необхідністю аналізу трафіка на сервері, який демонструє штатну роботу системи ДТГ, за допомогою команди ***netstat***, але такий аналіз залишає для зловмисників можливість розкриття таємниці окремих голосів, наприклад, з обраних *IP*-адрес. Це є недоліком метода протидії атаці *MITM*, що запропонований у [1]. Тому доцільно надати кожному виборцю можливість самостійно контролювати факт спілкування з реальним сервером ВД, а не з підробкою зловмисників, що усуває можливість створення непомічених загроз з використанням *MITM*. Розробка методу, реалізація котрого дозволить усім виборцям самостійно впевнитись в тому, що їх звернення потрапляють дійсно на штатний сервер ВД, і є одним із основних завдань даної роботи.

### 3.2. Вдосконалення методу протидії атаці *MITM*

**Постановка завдання:** вдосконалити метод протидії атакам посередника у відкритій системі ДТГ, який би дозволив кожному виборцю перед здійсненням акту волевиявлення без особливих ускладнень самостійно впевнитись у тому, що він дійсно спілкується зі штатним сервером ВД, а не з підробкою зловмисників.

Для виконання цього завдання запропоновано модель безперервного аудиту виборцями програмно-апаратних засобів сервера голосування. На цьому сервері і тільки на ньому під час голосування може зберігатись інформація, з якою пов'язані питання довіри виборців. Тому треба надати доступ виборцям під час голосування до інформації з цього сервера, яка свідчить про неможливість розкриття таємниці голосів, а також про відсутність фальсифікацій щодо їх

підрахунку. Інтерфейс для отримання такої інформації вже було запропоновано у роботі [1]. Це є інтерфейс для контролерів (див. рис. 1.1). Наша задача полягає у створенні захищеного каналу доставки інформації з цього інтерфейсу до виборців під час голосування. Крім того, слід обрати технічні засоби з відкритим монтажем, який би спрощував процедуру аудиту. Також для захисту каналу від електромагнітних випромінювань розроблено інформаційну технологію автоматизованого радіомоніторингу [116], який в спрощеному вигляді може бути представлений у вигляді сукупності наступних чотирьох етапів.

Перший етап передбачає аналіз поточного завантаження діапазону і накопичення даних про частоти, рівні і характер електромагнітних випромінювань в робочому діапазоні частот з прив'язкою даних до місця прийому. Під «відомими» випромінюваннями розуміється сукупність накопичених за певний інтервал часу даних про завантаження діапазону, отриманих за результатами проведення поточного контролю. При цьому передбачається, що небезпечні сигнали відсутні, що досягається, наприклад, поступовим накопиченням «відомих» випромінювань з ретельною перевіркою кожного з випромінювань ретельною перевіркою кожного з випромінювань.

На другому етапі до переліку «невідомих» включаються дані про випромінювання, сукупності параметрів яких задовольняють заданим критеріям пошуку. Використання «опорної» антени передбачає наявність у складі пошукової системи антенного комутатора, що забезпечує почергове підключення однієї з прийомних (в виділеному приміщенні) антен та «опорної» антени, що знаходиться поза контрольованою зоною приміщення, але забезпечує надійний прийом всіх зовнішніх сигналів.

Третій етап передбачає проведення тестування, що дає певний ефект як при виявленні радіомікрофонів без закриття (випромінювання в виділеному приміщенні спеціально синтезованих акустичних сигналів), так і при проведенні спецдосліджень на ПЕМВН шляхом відповідної модуляції інформативних параметрів випромінювань.

Для виконання четвертого етапу необхідно здійснити порівняння

максимальних (з виходів антен в контрольованій зоні) компонент спектра з рівнями відповідних компонент попередньо накопичених у виділеному приміщенні «відомих» електромагнітних випромінювань (при явній відсутності випромінювань ЗП) і граничним рівнем для відповідної частоти, а потім за результатами порівняння приймається рішення про наявність (відсутність) «невідомих» випромінювань в контрольованій зоні

Для розв'язання задачі аудиту сервер голосування реалізовано на відкритій платі міні комп'ютера, до якого через спільну локальну мережу *Ethernet* підключено спеціалізований сервер аудиту. Таке підключення виключає можливість реалізації атаки посередника між серверами, бо розірвання зв'язку фіксується як порушення. Принцип роботи контролюючого сервера полягає у наступному. Періодично кожні декілька секунд цей сервер за протоколом *SSH* звертається до сервера голосування за інформацією про поточні активні процеси (команда *ps -aux*). На процеси, що запущені контролерами і операційною системою цей сервер не реагує, а у разі появи будь-якого іншого процесу – протоколюються його параметри і відправляється сигнал тривоги на пристрої, які вказані контролерами. Необхідно, щоб встановлювали і підключали контролюючі сервери представники виборців, або щоб це відбувалось під їх наглядом. Підключення контролюючих серверів і їх запуск слід робити у той час коли на сервері голосування ще не має критичної інформації. Тому ніяких обмежень щодо доступу виборців та їх довірених осіб для перевірки апаратних засобів вводити не потрібно. Цим забезпечується довіра громадян до засобів розшифровки і підрахунку голосів, бо інакше може виникати підозра у тому, що сервер голосування являє собою «чорний ящик» з імітатором який демонструє виборцям нібито чесне голосування, а насправді розкриває і підмінює їхні голоси. Після запуску контролюючого сервера виборці можуть продовжувати безперервний контроль у дистанційному режимі без втрати інформації про можливі порушення. Слід зауважити, що будь-якій групі громадян можна дозволяти встановлення своїх контролюючих серверів (фізичних або логічних) у необмеженій кількості. Для доступу виборців до цих серверів слід

використовувати протокол *HTTPS* з вибором центру сертифікації на розсуд громадян. Програма контролюючого сервера виявляє і протоколює усі дії адміністратора щодо управління сервером голосування. Адміністратор повинен керуватись спеціальною відкритою для виборців інструкцією, яка зобов'язує після кожного сеансу управління сервером голосування перед завершальною командою ***exit*** вводити команду ***history > haabbccdd.txt*** , де замість букв ***aabbccdd*** слід вказати дату і час завершення сеансу роботи, а саме так: ***aa*** – номер місяцю, ***bb*** – число, ***cc*** – години, ***dd*** – хвилини. При цьому буде утворено файл з усіма командами, які були введені адміністратором у даному сеансі роботи. Це дозволяє виборцям контролювати роботу адміністратора шляхом порівняння змісту створених файлів з переліком штатних команд.

Також через контролюючий сервер кожен виборець може впевнитись у тому, що він спілкується зі штатним сервером голосування, а не з підробкою зловмисників.

Концептуальна модель вдосконаленої системи ДТГ представлена на рис.3.3.

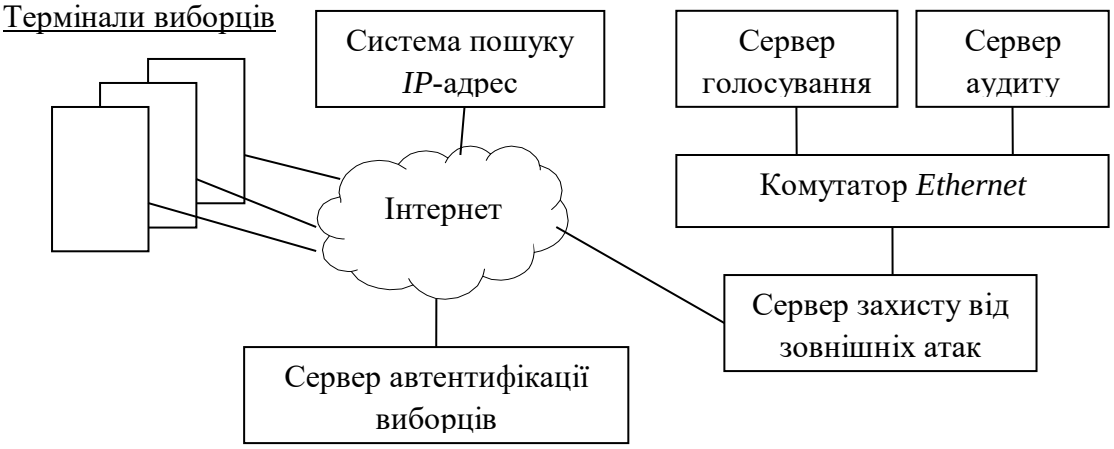



Рис. 3.3. Концептуальна модель вдосконаленої системи ДТГ

Для виявлення атаки посередника виборець через клавішу аудиту отримує доступ до журналу реєстрації з'єднань з сервером. Фрагмент цього журналу показано на рис. 3.4.

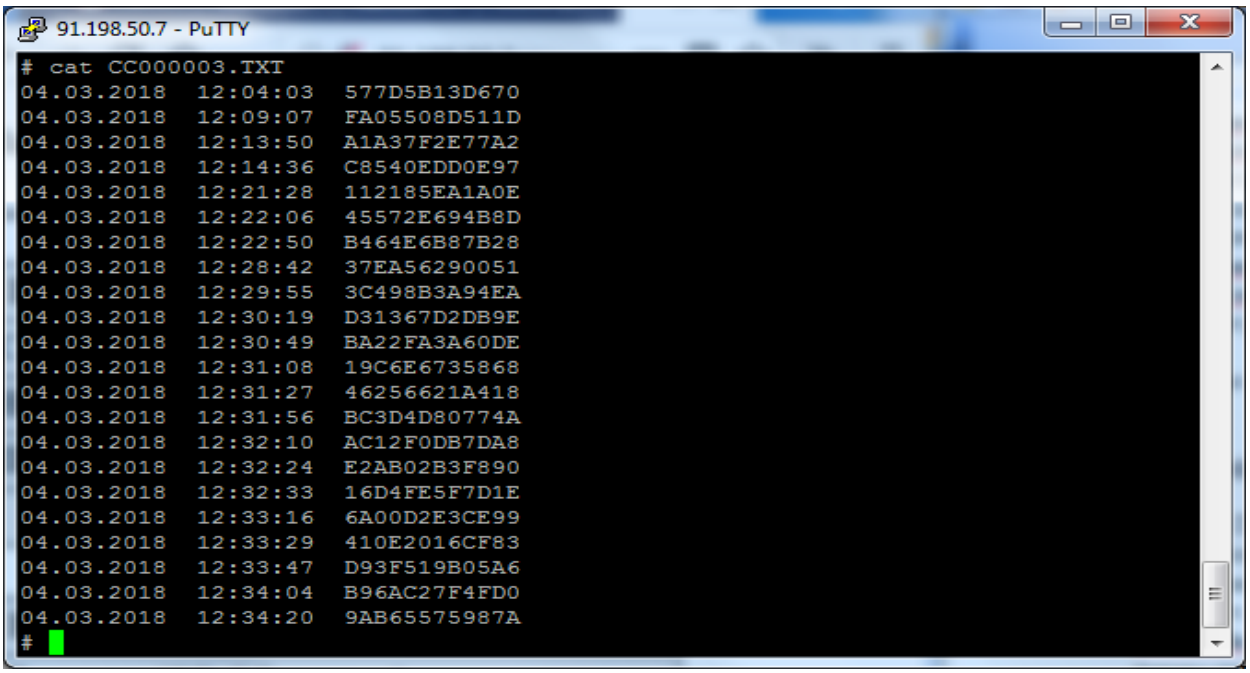



Рис. 3.4. Результат роздруківки сторінки журналу реєстрації з’єднань з сервером

У цьому журналі, крім дати і часу з’єднання роздруковуються коди, що являють собою випадкову степінь примітивного елементу поля Галуа, яка відправляється виборцю для обміну ключами за алгоритмом Діффі-Геллмана.

Представлення виборцю даних для перевірки з’єднання з сервером показано на рис. 3.5.

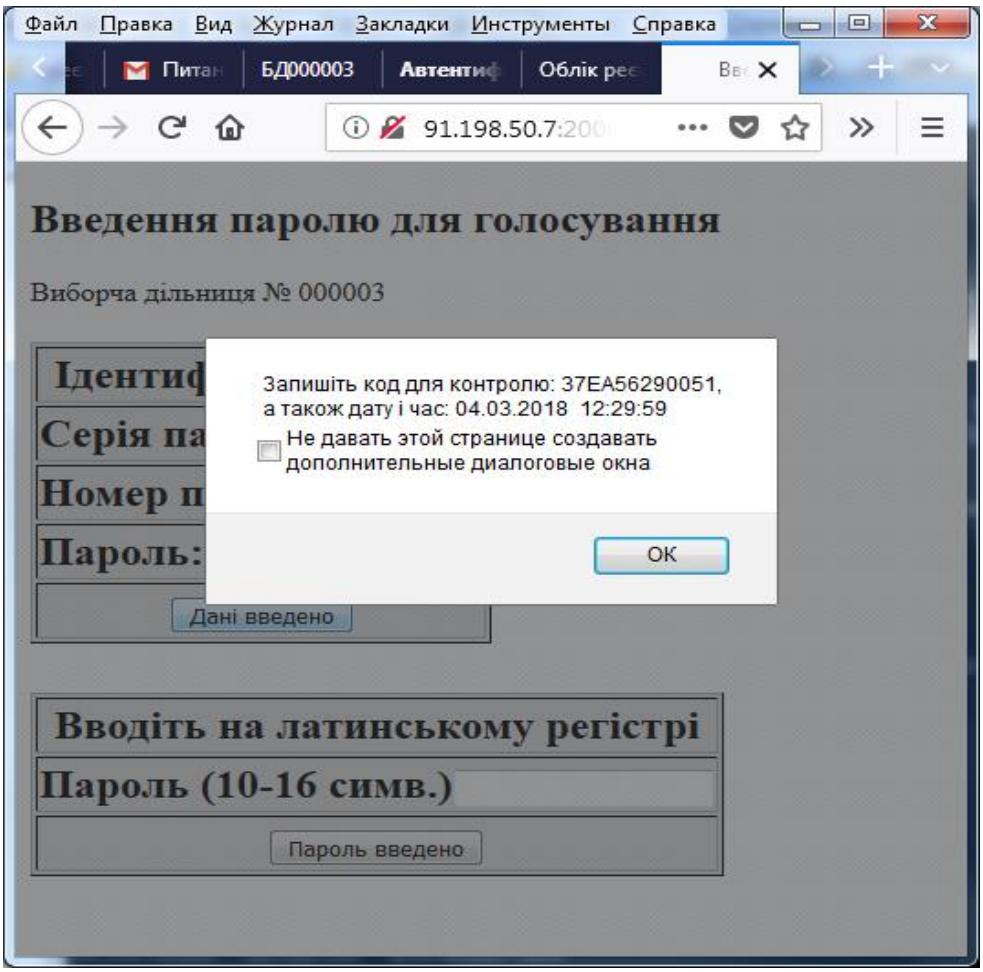



Рис. 3.5. Вигляд форми представлення виборцю даних для перевірки дійсності його звернення до сервера голосування

Шляхом порівняння кодів на момент часу свого запиту у журналі і в отриманому повідомленні може одразу впевнитись, що він дійсно спілкується зі штатним сервером для голосування.

У разі реалізації атаки посередника зловмисник повинен для обміну ключами з виборцем використовувати свою випадкову степінь, код якої не може співпадати з кодом, що занесений у журнал сервера для голосування. Звернення виборця через контролюючий сервер до журналу на сервері ВД відбувається за протоколом *HTTPS*, що виключає можливість атаки посередника на цьому з'єднанні, а контролюючий сервер підключено у єдину мережу *Ethernet* із сервером для голосування, де неможливо розірвати з'єднання для реалізації атаки, бо наявність зв'язку між цими серверами контролюється безперервно.

### 3.3. Особливості технологій дистанційної автентифікації виборців

Крім описаних у розділі 1 переваг відкритих систем ДТГ, слід відмітити, що залишається без відповіді питання дистанційної автентифікації особи виборця. Оскільки однією з відомих вимог до систем голосування є заборона передачі свого права голосу іншій особі, то можна вважати актуальною задачу дистанційного підтвердження особи виборця (або автентифікацію) в умовах повної відкритості (прозорості) системи ІГ з метою усунення очної перевірки особи в період уточнення списків голосуючих перед кожним актом волевиявлення. Це має особливе значення у випадках тривалих відряджень виборців.

Для дистанційного підтвердження особи може використовуватись низка відомих ознак. Наприклад, в Естонії, де було вперше впроваджено ДТГ на виборах державного масштабу [19-21], для підтвердження особи виборця використовують персональну електронну картку, яка є заміною паспорту. Одним з недоліків такого методу підтвердження є потреба у спеціальному пристрої для зчитування інформації з електронної картки. Крім того, як показано в роботі [4], такий метод автентифікації не захищає виборця від незаконного впливу. В Україні з 1 січня 2016 року також розпочато впровадження пластикових ID-карток замість паспортів. Але слід зауважити, що неможливо за допомогою будь-якої картки досягти беззаперечної гарантії того, що проголосувала саме та, а не якась інша особа. Це багаторазово продемонстровано голосуючими у ВР України. Тільки у тих випадках, коли ознаку неможливо відокремити від особи виборця, може бути досягнута беззаперечна гарантія того, що проголосувала саме та, а не якась інша особа. Такими властивостями у тій чи іншій мірі наділені біологічні ознаки людини. Серед цих ознак, крім широко відомих відбитків пальців, в останні десятиріччя використовують сітківку ока, райдужну оболонку ока, геометрію обличчя, термограму обличчя, геометрію руки, голос та динаміку почерку [89-90]. Найбільш придатною з перелічених ознак для дистанційної автентифікації можна вважати голос, бо майже все обладнання для доступу до мережі Інтернет має вбудований мікрофон, а у разі відсутності

вбудованого мікрофону можна скористатись окремим телефоном. В роботі [91] щодо розпізнавання по голосу вказано на високу імовірність помилок другого роду. Такі помилки у разі зміни голосу через хворобу або з інших причин можуть унеможливити здійснення виборцем акту волевиявлення, що неприпустимо для системи голосування. Слід зауважити, що вплив помилок першого роду щодо розпізнавання голосу можна компенсувати шляхом використання комбінації голосу з паролем. При цьому для того, щоб проголосувати за когось іншого, треба крім знання паролю ще й мати такий самий або у достатній мірі схожий голос. Результати експериментальних досліджень, що наведені в роботі [92] і представлені у таблиці 3.3, свідчать про існування суттєвої залежності кількості помилок розпізнавання особи від варіанту прочитаного тексту.

**Таблиця 3.3**

**Залежність помилок від варіантів прочитаного тексту**

| Варіант тексту | Процент помилок | Варіант тексту | Процент помилок | Варіант тексту | Процент помилок |
|---|---|---|---|---|---|
| «шиншилла шила шубу» | 18,7% | «Клара украла кораллы» | 1,2% | «шалость олигарха» | 3,8% |

Бачимо з наведених у табл. 3.3 значень, що шиплячі звуки негативно впливають на якість розпізнавання особи, а обираючи тексти з переважною більшістю дзвінких звуків, можна значно підвищити цю якість. Дослідження в напрямку поліпшення розпізнавання по голосу тривають, а існуючі результати свідчать про можливість за допомогою біологічних ознак отримувати дані для уточнення особи виборця. В роботі [93] для автентифікації особи виборців обрано електронний цифровий підпис (ЕЦП), який може використовуватись в інших цілях, що не пов’язані з виборами, і тому не може бути переданий іншій особі. Але для того, щоб використовувати будь-яке уточнення особи у відкритих системах дистанційного голосування, слід розробити технологію, яка б дозволяла підключати додаткові засоби розпізнавання не втрачаючи жодної з описаних вище переваг системи ДТГ, включаючи властивості прозорості і

контрольованості. Розробка саме такої технології і висвітлена у матеріалі даного підрозділу.

### 3.4. Вдосконалення технології дистанційної автентифікації виборців у системі ДТГ

3.4.1. *Визначення періоду уведення автентифікаційних даних*. Візьмемо за основу логічну модель відкритої системи дистанційного голосування, яку зображена у розділі 1 на рис. 1.1.

Ще раз розглянемо технологічний цикл функціонування системи дистанційного голосування, зображений на рис. 3.2, але вже з метою визначення часових інтервалів, у яких доцільно розпізнавання осіб виборців.

Нагадаємо, що на рис. 3.2 сірим тоном виділені процеси, які відбуваються на сервері ВД, і прийнято такі скорочення назв періодів технологічного циклу: ППД – період підготовки даних про претендентів на дистанційне голосування; ППС – період підготовки сервера ВД (встановлення ОС та програм загального користування); ППЗ – прикладне програмне забезпечення для дистанційного голосування; ПВЕБ – період введення електронних бюлетенів (в цей період запити виборців сервером не обслуговуються, а в дільничних списках виборців помічають тих, хто голосуватиме дистанційно, щоб не видавати їм паперові бюлетені).

Проаналізуємо кожний з періодів технологічного циклу з метою визначення тих періодів, де є потреба у додаткових засобах розпізнавання осіб виборців.

В період підготовки даних про претендентів на дистанційне голосування необхідна особиста присутність виборців з паспортами. В цей період відбувається очна перевірка осіб виборців і занесення в базу даних відомостей про них та ознак, що необхідні для ідентифікації та автентифікації. Для зберігання цих даних використовується окремий сервер, який працює незалежно від сервера ВД і може зберігати дані протягом багатьох голосувань.

В період підготовки сервера ВД, крім встановлення ОС *OpenBSD* і пакетів *Node.js*, які забезпечують виконання програми на мові *Java Script*, створюють

користувача *kontrol* з правами спостерігача (без права на внесення будь-яких змін на сервері), а також користувача *admin* з правами роботи виключно у директорії *home/admin* і блокують користувача *root* з повними правами. Після цього сервер буде працювати автоматично в режимі обмеженої функціональності, що дозволяє уникнути будь-яких спроб щодо зловмисного втручання в роботу сервера.

В період встановлення ППЗ адміністратор повинен занести в директорію *home/admin* наступні три файли:

- файл з серверною програмою на мові *Java Script*;
- файл з клієнтською програмою на мовах *HTML* та *Java Script*;
- файл з даними, який формується по запиту адміністратора на окремому сервері для кожного голосування. Конфіденційні дані у цьому файлі знаходяться у зашифрованому вигляді.

З початку цього періоду кожен користувач мережі може отримати права на спостереження за усіма файлами і параметрами процесів на сервері ВД. Це надає можливість впевнитись у тому, що все програмне забезпечення сервера ВД є штатним, а потрібні дії адміністратора виконуються точно за графіком. Слід зауважити, що цей графік, а також все програмне забезпечення заздалегідь відкриті для проведення будь-яких експертиз.

У період введення паролів для дистанційного голосування виборці повинні пройти процедуру автентифікації. Слід зауважити, що в цей період згідно виборчому законодавству [16,17] на основі Державного реєстру виборців (ДРВ) складаються, а потім уточнюються списки виборців. Тривалість підготовчого періоду зазвичай вкладається в 15 днів до виборів, бо раніше за законом можуть бути ще не створені дільничні комісії. Період введення паролів недоцільно розпочинати до створення виборчих дільниць, а оскільки збільшення цього періоду розширює інтервал часу для обрання виборцями зручного моменту проходження автентифікації, то зменшувати цей період теж недоцільно. В роботі [1] для введення паролю запропоновано приймати виборців у відділах ДРВ, де й проводити очну перевірку. Але маючи можливість дистанційної автентифікації,

не обов’язковою стає очна перевірка, що особливо доцільно у разі тривалих відряджень виборців. Таким чином, саме в період введення паролів для дистанційного голосування існує потреба у додаткових засобах розпізнавання осіб виборців, щоб уникнути можливої підміни особи голосуючого.

3.4.2. *Технологія маніпулювання автентифікаційними даними у системі з очною перевіркою осіб виборців*. В період введення електронних бюлетенів запити не обслуговуються, тому залишається проаналізувати тільки період голосування. В цей період виконуються найбільш відповідальні дії, але за допомогою нейтралізації незаконного впливу на виборців, як запропоновано в роботі [4], шансів примусити виборця голосувати всупереч власному розсуду не існує. Це може статись тільки тоді, коли сам виборець передасть свій вірний пароль для голосування іншій особі, при цьому виборець захищений тим, що має можливість передати зловмиснику помилковий пароль, бо система однаково реагує, як на вірний, так і на помилковий пароль. Оскільки цей пароль вводять у відкритому вигляді, то сам виборець завжди побачить і виправить помилку. Через відкрите введення паролю не виникає небезпеки, бо пароль діє лише один раз і ним неможна скористатись вдруге. У разі виникнення сумнівів у виборця щодо точності пароля можна багато разів голосувати з різними паролями, але зараховано буде тільки один голос з вірним паролем.

Технологію отримання пароля для голосування у спеціалізованому пункті представлено на рис. 3.6.

Ця технологія передбачає прибуття виборця до спеціалізованого пункту з паспортом, і, можливо, зі своїм мобільним терміналом. У принципі, замість свого терміналу виборець може скористатись будь-яким іншим, наприклад, тим, що призначений для загального користування, але, зазвичай, своєму він більше довірятиме.

Після очної перевірки особи виконується наступна послідовність дій:

1. Оператор відправляє на сервер ДВП запит для утворення захищеного з’єднання (через обмін ключами за алгоритмом Діффі-Геллмана).

2. Оператор авторизується через захищене з'єднання та отримує дозвіл на відправку ідентифікатора виборця.

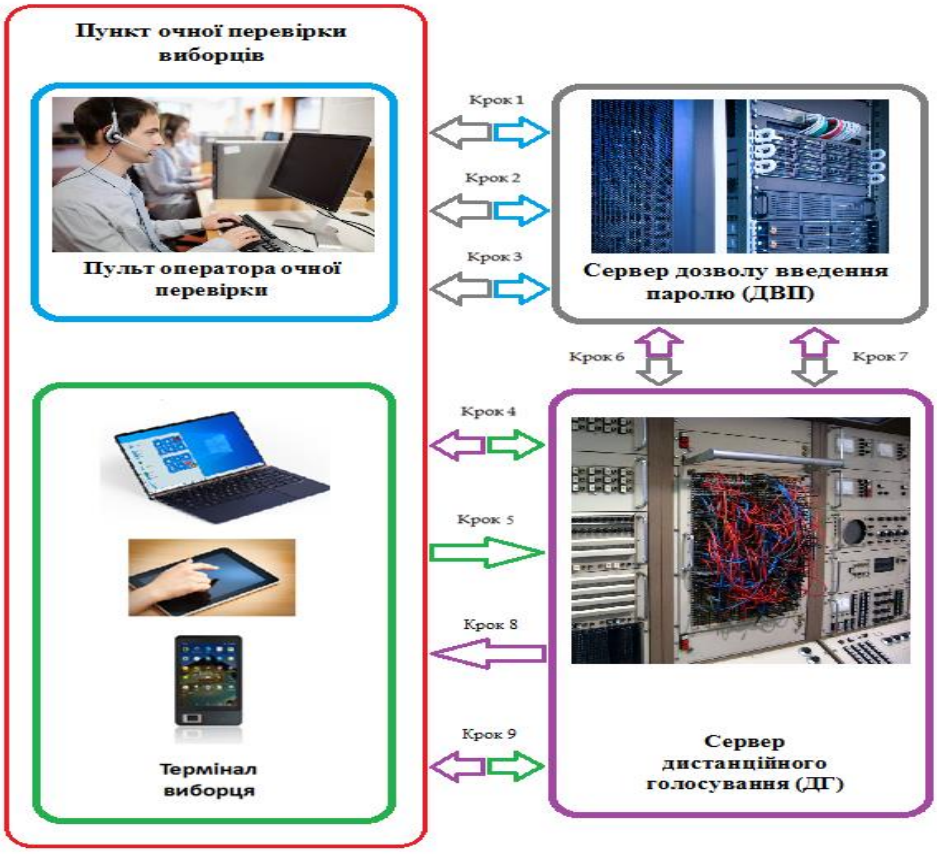



Рис. 3.6. Технологія отримання пароля для голосування у спеціалізованому пункті з очною перевіркою виборців, де Сервер ДВП – сервер дозволу введення паролю

3. Оператор відправляє на сервер ДВП ідентифікатор виборця (після очної перевірки) і отримує повідомлення про надання 10 хвилин для введення паролю.

4. Виборець відправляє на сервер дистанційного голосування (ДГ) запит на утворення захищеного з'єднання (через обмін ключами за алгоритмом Діффі-Геллмана).

5. Виборець авторизується через захищене з'єднання на сервері ДГ і очікує дозвіл на відправку пароля.

6. Сервер ДГ утворює захищене з'єднання з сервером ДВП (через обмін ключами за алгоритмом Діффі-Геллмана).

7. Сервер ДГ відправляє на сервер ДВП запит з ідентифікатором виборця і, якщо момент запиту вкладається у виділені 10 хвилин, отримує відповідь з цим самим ідентифікатором.

8. Сервер ДГ відправляє на термінал виборця дозвіл для введення паролю.

9. Виборець відправляє на сервер пароль для голосування і отримує відповідь про успішне завершення процедури.

У разі, коли момент запиту (див. дію 7) не вкладається у виділені 10 хвилин, відповідь сервера ДВП замість ідентифікатора заповнюється нулями. При цьому виборцю замість дозволу для введення паролю відправляється відмова.

Введення у склад обладнання системи ДТГ сервера ДВП дозволяє при повному збереженні прозорості сервера ВД доповнювати систему додатковими засобами дистанційного розпізнавання осіб виборців по голосу, по ЕЦП і т.ін. Ці засоби встановлюються на сервері ДВП, який не потребує повної контрольованості (прозорості) в режимі реального часу, бо в період голосування, а тільки в цей період на сервері ІГ з'являється інформація, яка потребує абсолютного захисту, ніякої взаємодії між серверами ІГ і ДВП не відбувається. Тому на сервері ДВП можуть використовуватись традиційні засоби захисту інформації. Оскільки між обома серверами для кожного сеансу обміну даними утворюється спеціальний захищений канал зв'язку, то це дозволяє розміщувати їх незалежно один від одного в довільному місці мережі Інтернет.

3.4.3. *Технологія маніпулювання автентифікаційними даними у системі без очної перевірки осіб виборців*. Запропонований в даній роботі розподіл дій між серверами ІГ і ДВП дозволяє виборцям отримувати пароль для голосування без обов'язкової очної перевірки. Кількість обраних виборцями додаткових ознак для автентифікації залежить тільки від можливостей придбання ними тих чи інших засобів для введення цих ознак.

Запропонована технологія представлена на рис. 3.7.

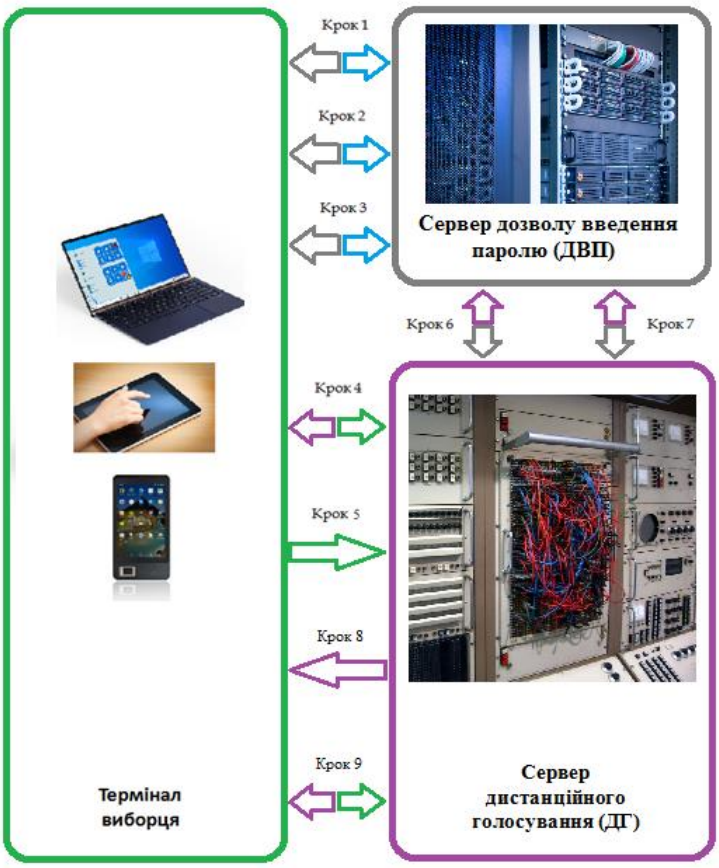



Рис. 3.7. Технологія отримання пароля для голосування без очної перевірки

Ця технологія передбачає наявність на сервері ДВП біологічних та/або інших ознак виборців, які заздалегідь повинні бути занесені в базу даних. Крім того на сервері ДВП повинно бути встановлено програмне забезпечення для розпізнавання осіб виборців по цих ознаках в дистанційному режимі. При цьому для отримання пароля виконується наступна послідовність дій:

1. Виборець відправляє на сервер ДВП запит для утворення захищеного з'єднання (через обмін ключами за алгоритмом Діффі-Геллмана).

2. Виборець авторизується через захищене з'єднання та отримує запит на введення біологічних або інших додаткових ознак своєї особи.

3. Виборець виконує запит сервера ДВП щодо введення додаткових ознак та отримує повідомлення про вдалу автентифікацію і надання 10 хвилин для введення паролю. У разі невдалої автентифікації виборець отримає запрошення на повторну спробу введення додаткових ознак. Слід зауважити, що кількість додаткових ознак може бути якою завгодно.

Дії 4 – 9 в точності співпадають з відповідними діями технології, розглянутої вище.

Таким чином, запропонована технологія автентифікації виборців у відкритій системі ДТГ, за рахунок доповнення системи сервером ДВП, у якому розміщені засоби розпізнавання осіб виборців за додатковими біологічними або іншими ознаками, дозволяє створити більш зручні умови для виборців, не вимагаючи від них проходження обов'язкової очної перевірки перед кожним голосуванням, що особливо доцільно у разі тривалих відряджень виборців.

## Висновки до третього розділу

1. Показано, що атака посередника – цілком реальна загроза інформаційним ресурсам систем ДТГ, яка за умов відсутності або недостатньої ефективності механізмів захисту, може призвести до порушення конфіденційності голосів виборців, а також до підміни результатів волевиявлення.

За результатами аналізу моделі атаки посередника в транспарентних системах ДТГ запропоновано модель безперервного аудиту виборцями програмно-апаратних засобів сервера голосування відповідний метод протидії цим атакам. А саме, запропоновано на сервері ВД вести журнал, у якому під час встановлення зв'язку між виборцем і сервером ВД реєструються дані про поточний момент часу разом з випадковим числом, що генерується сервером ВД і відправляється виборцю. При цьому виборець може порівняти дані про момент часу у журналі і значення випадкового числа з тими, що він отримав у повідомленні, та впевнитись у тому, що він спілкується дійсно зі штатним сервером ВД.

2. Удосконалено технологію дистанційної автентифікації виборців з урахуванням обмежень, що пов'язані із забезпеченням транспарентності системи ДТГ. Удосконалення здійснено за рахунок доповнення системи сервером дозволу введення паролю (у якому розміщені засоби розпізнавання осіб виборців за біологічними або іншими ознаками, які заздалегідь повинні бути занесені в базу даних цього серверу) та відповідного розподілу дій між основним сервером

(що безпосередньо оброблює звернення виборців під час здійснення актів волевиявлення) і додатковим сервером, внаслідок чого забезпечується можливість розпізнавання осіб виборців у дистанційному режимі. При цьому між обома цими серверами для кожного сеансу обміну даними утворюється спеціальний захищений канал зв'язку, що дозволяє розміщувати їх незалежно один від одного в довільному місці мережі Інтернет.

3. Показано, що дистанційну автентифікацію виборця доцільно проводити під час уведення ним парольної інформації безпосередньо перед здійсненням акту голосування, що позбавляє можливості  забороненої передачі права голосу іншим особам.

4. Удосконалена технологія автентифікації створює більш зручні умови голосування для виборців, зокрема надає можливість позбутися обов'язкової очної перевірки осіб виборців перед кожним актом волевиявлення, що особливо доцільно у разі тривалих відряджень виборців. При цьому збережено усі позитивні якості відкритої системи, включаючи повну контрольованість процесів на сервері інтернет-голосування в режимі реального часу, що усуває будь-які підстави для недовіри з боку виборців щодо збереження таємниці голосів або точності підрахунку.

5. Запропонована технологія надає можливість виборцям гнучкого вибору методів автентифікації, не позбавляючи їх можливості користуватись також і очною перевіркою. Обрання виборцями додаткових ознак для автентифікації залежить тільки від можливостей придбання ними тих чи інших засобів для введення обраних ознак.

# РОЗДІЛ 4
# ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ СИСТЕМИ ДИСТАНЦІЙНОГО ТАЄМНОГО ГОЛОСУВАННЯ

## 4.1. Завдання експериментального дослідження

Мета цього розділу полягає у практичному доведенні з підтвердженням на прикладі конкретної програмно-апаратної реалізації можливості побудови транспарентної (відкритої, повністю контрольованої) системи, де ключову роль при перевірці точності функціонування грають користувачі мережі Інтернет, включаючи всіх без винятку громадян, які мають бажання проконтролювати вірність функціонування системи. Це дозволяє усунути привід для підозр у тому, що хтось має можливість для прихованих від громадян фальсифікацій чи інших порушень штатного режиму роботи системи, бо кожному, хто має подібні підозри, надається можливість дистанційного контролю системи. У цьому експерименті систему слід побудувати так, щоб будь-які приховані порушення штатного режиму роботи унеможливити за рахунок вибору програмно-апаратних засобів та технології контролю в мережі Інтернет.

Вибір програмно-апаратних засобів будемо робити виходячи з того, щоб задовольнити вимогам проведення голосування на державному рівні через публічну мережу Інтернет в умовах України.

Оскільки системи електронного голосування на державному рівні завжди створюються як додаткова гілка до традиційних систем з метою полегшення процедури волевиявлення, то будемо враховувати особливості існуючої в Україні системи проведення виборів. При цьому, бажано не порушувати структуру виборчих дільниць, на кожній з яких підраховують і визначають суму голосів. Це змушує використовувати окремі сервери для підрахунку голосів по дільницях. Таке рішення не є технічно складним або витратним. Оскільки в Україні кількість виборців на дільниці не може перевищити 2500, то для серверів можуть підійди сучасні міні-комп'ютери [95], які за розміром трохи більше банківської карти, а за ціною близько $50 (за станом на 01.05.2019 р.),

включаючи необхідні до них додаткові пристрої. Використання окремих серверів для кожної виборчої дільниці зручно і з тієї точки зору, що на різних дільницях зазвичай відрізняються комплекти бюлетенів, тому що вибори в органи різних рівнів влади найчастіше збігаються в часі. Ще однією важливою перевагою використання окремих серверів для виборчих дільниць є неможливість перевантаження мережевого обладнання у разі розширення системи. Крім того, встановлений період часу для голосування не потрібно збільшувати у порівнянні з наявним в традиційній системі. З цією проблемою зіткнулись в Естонії, де прийшлось збільшувати період голосування до одного тижня через неспроможність обслуговувати усі дільниці на одному сервері за існуючий у старій системі період часу.

Програмно-апаратні засоби, з яких складатиметься експериментальна система, повинні бути стандартними, сучасними і добре відомими користувачам мережі Інтернет, щоб не виникало питань з приводу якогось нестандартного невідомого обладнання, що може негативно вплинути на процеси створення і впровадження системи. Наше завдання полягає в тому, щоб конкретизувати і обґрунтувати вибір тих чи інших засобів, з усіх відомих, для досягнення поставленої мети, що полягає у створенні контрольованої системи, у якій був би неможливим витік конфіденційної інформації і неможливі невиявлені спотворення тієї інформації, яка повинна зберігатися незмінною. При цьому будь-які спроби злому системи повинні виявлятися таким чином, щоб не допустити реалізації загроз для інформації, яка підлягає захисту.

У разі одержання експериментальних підтверджень можливості практичної побудови транспарентної системи ДТГ вкрай бажано отримати кількісні оцінки продуктивності її роботи за різних умов застосування.

### 4.2. Вибір програмно-апаратних засобів

З огляду на те, що перший крок щодо вибору серверного обладнання для виборчих дільниць відображено у попередньому розділі, залишається

конкретизувати цей вибір і підтвердити на практичному прикладі його спроможність.

Міні-комп'ютерів (*SBC - single-board computer*) налічувалося на ринку у 2018 році близько сотні типів. Наш вибір полегшується тим, що операційна система (ОС) *OpenBSD* може бути встановлена тільки на 13 з них. Прив'язка до *OpenBSD* пояснюється тим, що це єдина ОС, яка (під назвою *BBOS*) сертифікована в Україні за критеріями технічного захисту інформації [1]. Оскільки в нашій системі питання захисту інформації мають першорядну важливість, вибір ОС *OpenBSD* був єдино можливим. З цим вибором легко погодитися, оскільки ця ОС зарекомендувала себе у світовій практиці як найкраща з точки зору ТЗІ. Її розробники питання захисту інформації завжди вважали найбільш пріоритетними.

Наш вибір *SBC* після випробувань щодо можливості встановлення ОС *OpenBSD* зосередився на *Raspberry Pi* 3*B*. Саме цей *SBC* є найбільш відомим користувачам мережі Інтернет, бо обсяг його продажу перевищив 10 млн. Крім того, на *Raspberry Pi* 3*B* без будь-яких проблем встановлюється ОС *OpenBSD*.

Вибір клієнтського обладнання вкрай бажано залишати таким, як було запропоновано в роботі [1], бо інакше це буде створювати обмеження для виборців у доступі до мережі, що негативно вплине на залучення виборців до дистанційного голосування. Єдиною вимогою для клієнтів залишається наявність будь-якого браузера, включаючи телевізор з функцією *SmartTV* або з приставкою *SmartTV BOX*.

Крім серверів виборчих дільниць ми запропонували до складу системи ввести додаткові сервери автентифікації, що надає виборцям можливість позбутись обов’язкової очної перевірки перед кожним актом волевиявлення. Оскільки на серверах автентифікації, а також на серверах підсумкових результатів волевиявлення, не має інформації про те, як хто проголосував, то ніяких обмежень щодо їх транспарентності не існує. Тому до цих серверів ніяких особливих вимог, крім передбачених існуючими стандартами і нормативними документами щодо захисту інформації, висувати непотрібно. Інформація, яка

обробляється на цих серверах, не стосується довіри виборців, а все, що стосується поняття довіри виборців, зосереджено виключно на серверах виборчих дільниць, які потребують забезпечення транспарентності.

Єдиною мовою для написання клієнтських модулів є мова *HTML*, до якої не існує альтернатив, а для виконання криптографічних обчислень у цих модулях також безальтернативною є мова *JavaScript*. З метою мінімізації мов програмування, що полегшує процеси розробки і перевірки програмного забезпечення, для написання серверної програми обрано спеціалізовану мову *JavaScript* для серверів, яка вільно розповсюджується у вигляді програмної платформи *Node.js*. Завдяки такому вибору переважна більшість функцій для криптографічних перетворень у клієнтській і серверній програмах однакові, що спрощує перевірку і написання програм.

### 4.3. Ініціалізація серверного обладнання

Є досить багато інформації в мережі про те, як встановити ОС *OpenBSD* на *SBC Raspberry Pi* 3*B* за допомогою комп'ютера під ОС *Linux*, але ми вирішили зробити це під найбільш популярною серед користувачів ОС *Windows*.

Для того щоб встановити *OpenBSD* на обрану *SBC* в нашому випадку використовувалися наступні технічні засоби:

- *SBC Raspberry Pi* 3*B* з блоком живлення (5 В);
- перехідник на *FT*232*RL UART-USB* з кабелем *USB*-міні *USB*;
- пристрій для запису карт пам'яті *MicroSD*;
- карта пам'яті *MicroSD* 8ГБ (досить 1ГБ);
- флеш *USB* з ємністю 4 ГБ (достатньо 1 ГБ);
- перемички для макетування 3 шт. ;
- кабель для підключення до мережі *Ethernet*;
- комп'ютер з *USB* портом під *Windows* 7 із встановленими програмами *Rufus*, *PuTTY* і *Node.js*.

В нашій локальній мережі *Ethernet* надання *IP* адрес відбувалося автоматично за допомогою маршрутизатора *TP-LINK* типу *TL-R*860.

Поточна версія ОС *OpenBSD* 6.4 була отримана у вигляді 18 файлів через посилання https://ftp.openbsd.org/pub/OpenBSD/6.4/arm64/.

На рис. 4.1 показано з'єднання вказаних пристроїв, крім живлення до *Raspberry Pi* 3*B*.

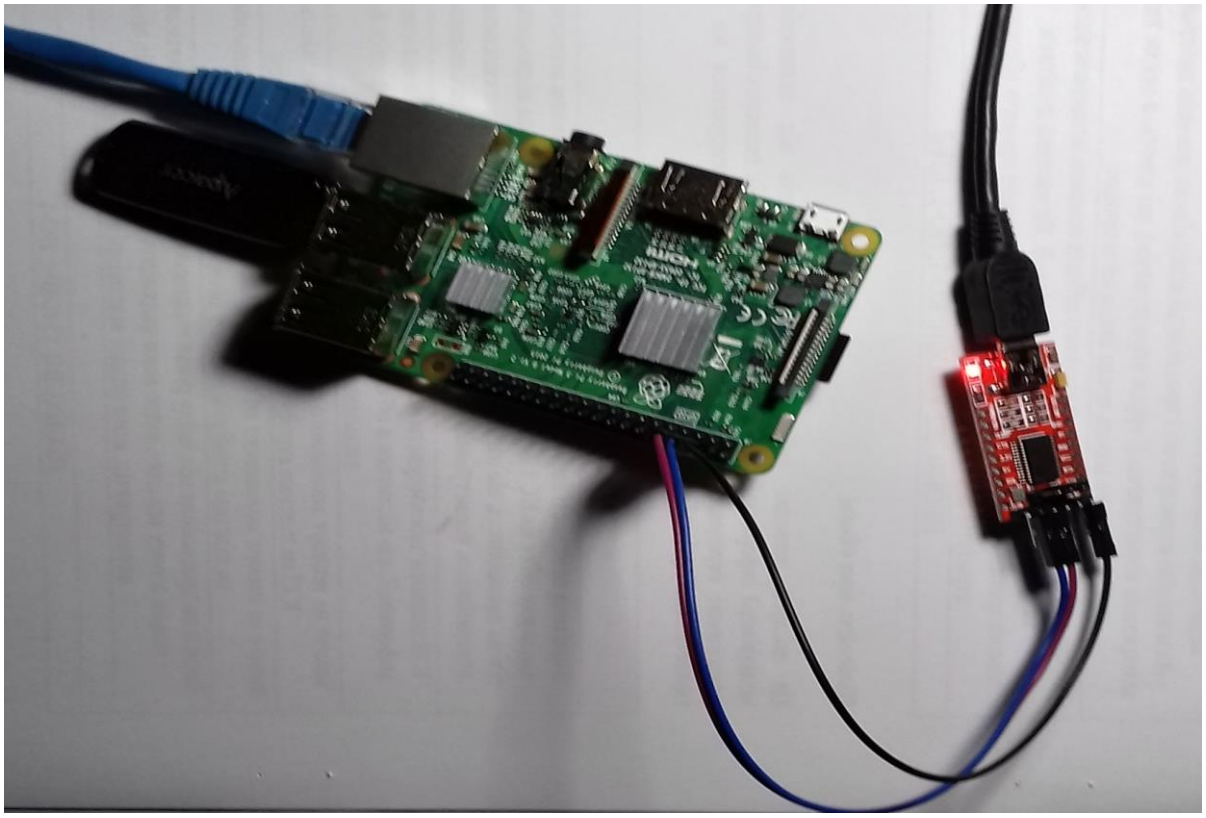


Рис. 4.1. Вигляд з'єднання пристроїв

Далі була виконана наступна послідовність дій.

1. Файл *miniroot64.fs* за допомогою програми *Rufus* був скопійований в режимі *DD* на карту пам'яті. Вибір режиму копіювання показаний на рис. 4.2.

2. Карту пам'яті після завершення копіювання було переставлено в *SBC Raspberry Pi* 3*B*. З'єднання пристроїв було виконано так, як показано на рис. 4.1. Кабель USB був підключений до комп'ютера, а живлення до пристрою Raspberry Pi 3B поки не підключалося.

3. Запущено програму *PuTTY* в режимі, який показаний на рис. 4.3.

4. Підключили живлення до пристрою *Raspberry Pi* 3*B* і дочекалися появи запрошення *Welcome to the OpenBSD / arm64*. Це показано на рис. 4.4.

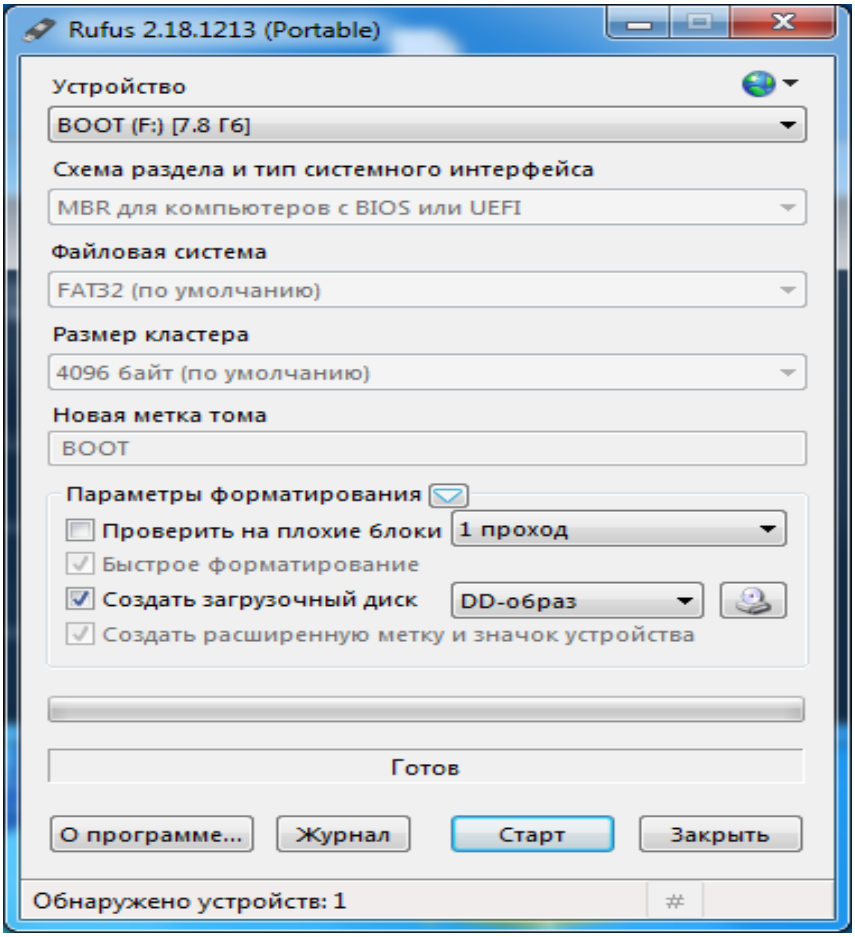



Рис. 4.2. Робоче вікно програми *Rufus*

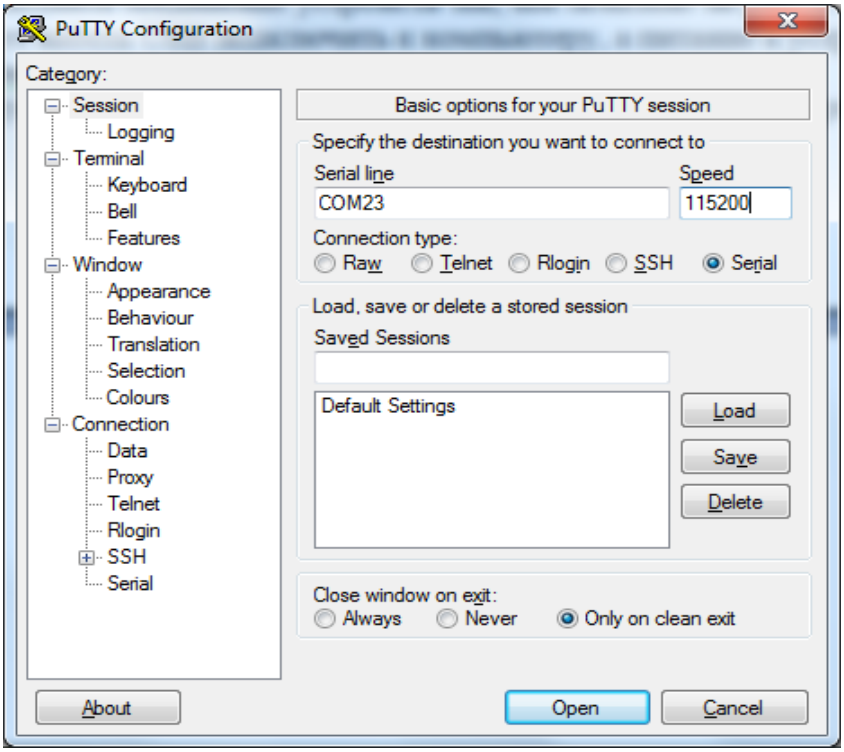



Рис. 4.3. Робоче вікно програми *PuTTY*

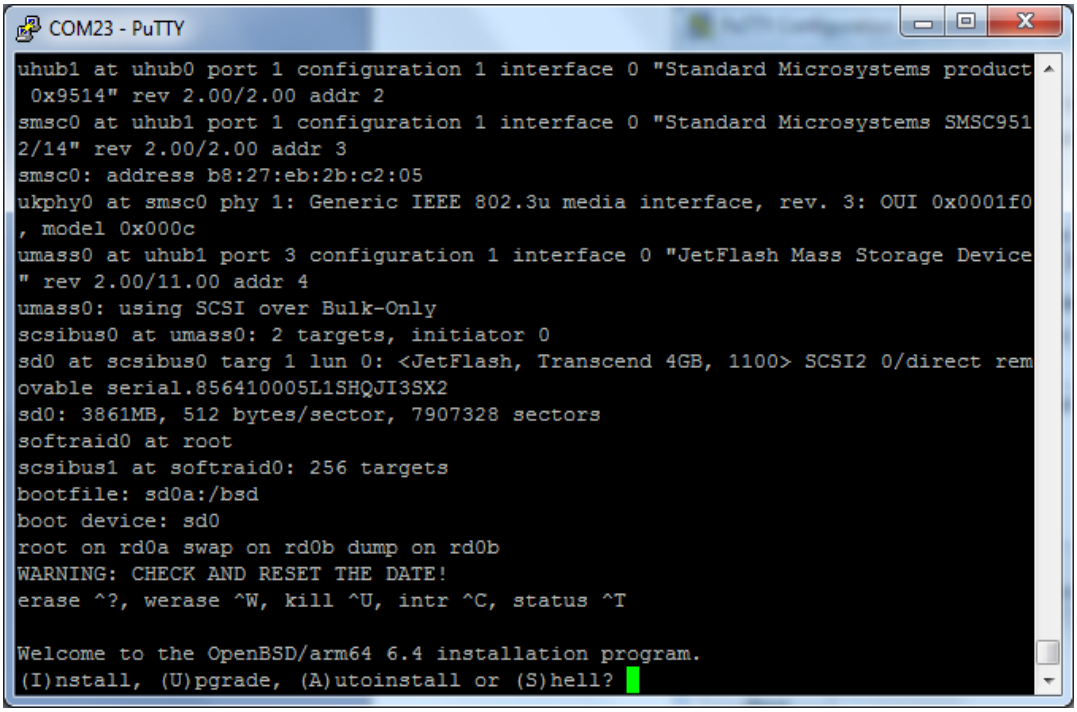



Рис. 4.4. Вікно терминалу програми *PuTTY*

5. За допомогою програми Node.js на своєму комп'ютері під *Windows* 7 створили і запустили *HTTP* сервер. Текст програми цього сервера показаний на рис. 4.5, а розміщені на ньому файли показані на рис. 4.6.

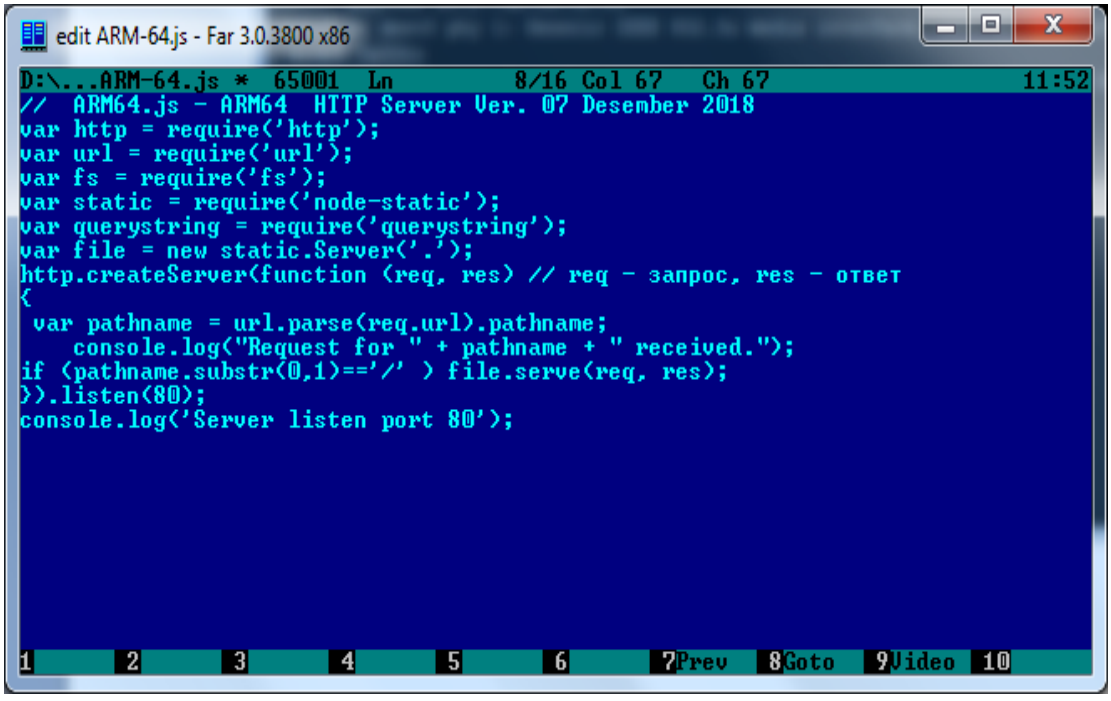



Рис. 4.5. Текст серверної програми

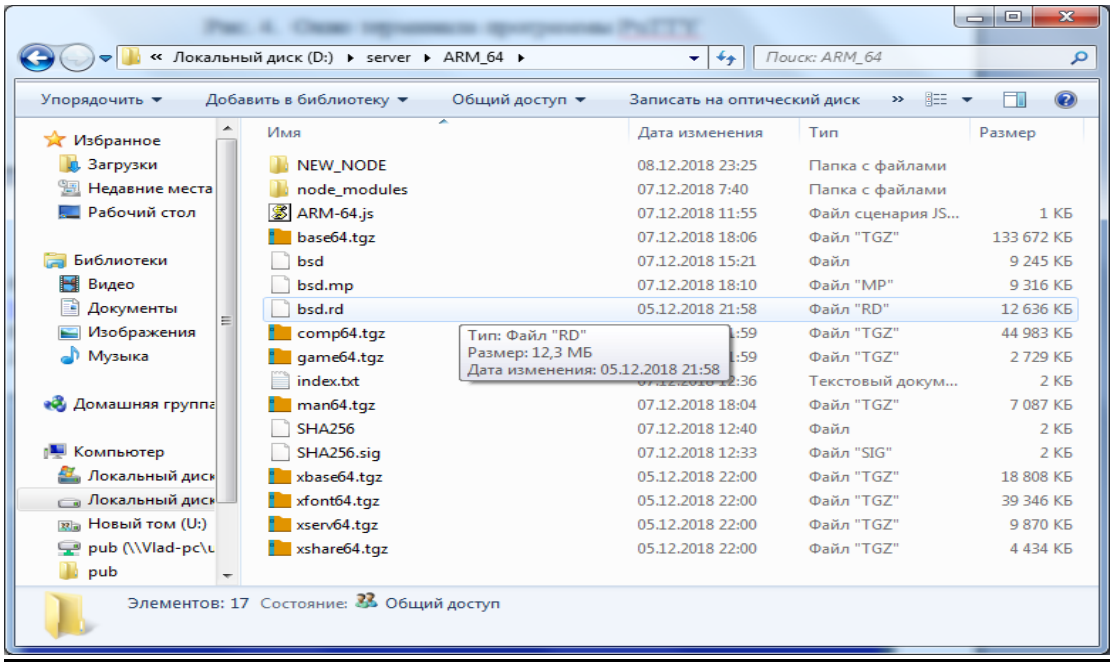



Рис. 4.6. Каталог файлів на сервері

6. Відповіли на питання через консоль згідно до інструкції на ресурсі *http://bijanebrahimi.github.io/blog/installing-openbsd-63-on-raspberry-pi-3.html* Вигляд консолі показаний на рис. 4.7.

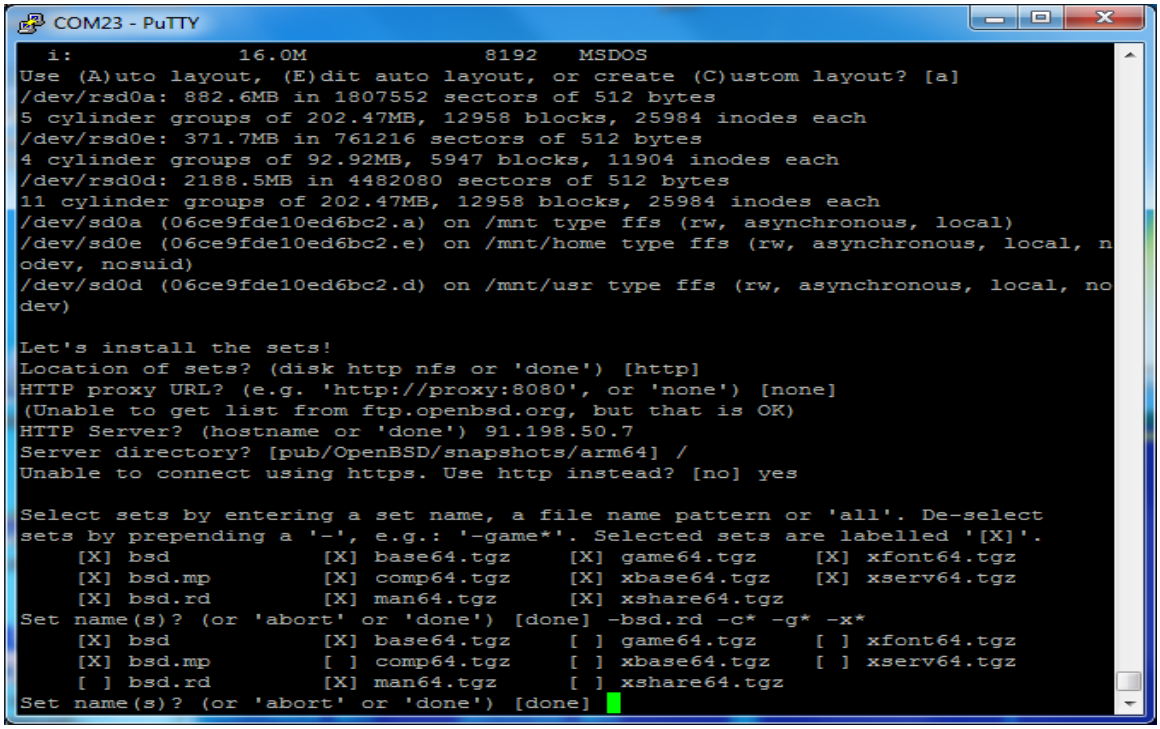



Рис. 4.7. Вигляд консолі під час завантаження файлів з сервера

7. На останнє запитання обрали відповідь за замовчуванням [*reboot*], після чого почалося завантаження ОС. Після завантаження отримали сервер під *OpenBSD*, образ якої зберігається на флеш *USB* з ім'ям диску *usb*0.

Першою дією адміністратора після встановлення ОС за технологією, якої ми дотримуємося, є створення двох користувачів, одному з яких надано ім’я *kontrol*, а другому – ім’я *admin*. Контролер з ім’ям *kontrol* може ознайомлюватись зі змістом файлів на сервері та виконувати усі безпечні команди, але не може заносити чи коригувати файли. Користувач *admin* може заносити файли у директорію /*home/admin* та виконувати програми з цієї директорії. Користувач-контролер безпечний для процесу функціонування системи, оскільки він нічого не може змінити в системі, але він може, виконуючи безпечні команди, перевірити справжність, встановленої на сервері ОС. Крім того, він може стежити за роботою системи, виявляючи всі загрози, які можуть привести до порушення політики безпеки системи.

### 4.4. Перевірка справжності ОС

Експеримент щодо перевірки справжності ОС проведено за допомогою програм *WinSCP* та *WinMerge*, які вільно розповсюджуються у мережі. Цю перевірку було проведено на комп'ютері під *Windows* 7, на якому створили дві папки, в одну, з яких, скопіювали за допомогою програми WinSCP всі директорії і файли з сервера виборчої дільниці, а в іншу - зі свого одноплатного комп'ютера, на якому встановили ОС *OpenBSD* точно так, як на сервері. Вміст однієї з цих папок показано на рис. 4.8. А результат порівняння папок на ідентичність за допомогою програми *WinMerge* для директорії *usr* показано на рис. 4.9.

Слід зауважити, що під час копіювання файлів з директорій *dev* і *tmp* виникали помилки у випадках, коли файли порожні або містять посилання на інший файл. Ці випадки ми ігнорували, оскільки порожні файли і посилання порівнювати не потрібно. В результаті порівняння були виявлені відмінності в деяких файлах, які представлені в табл. 4.1.

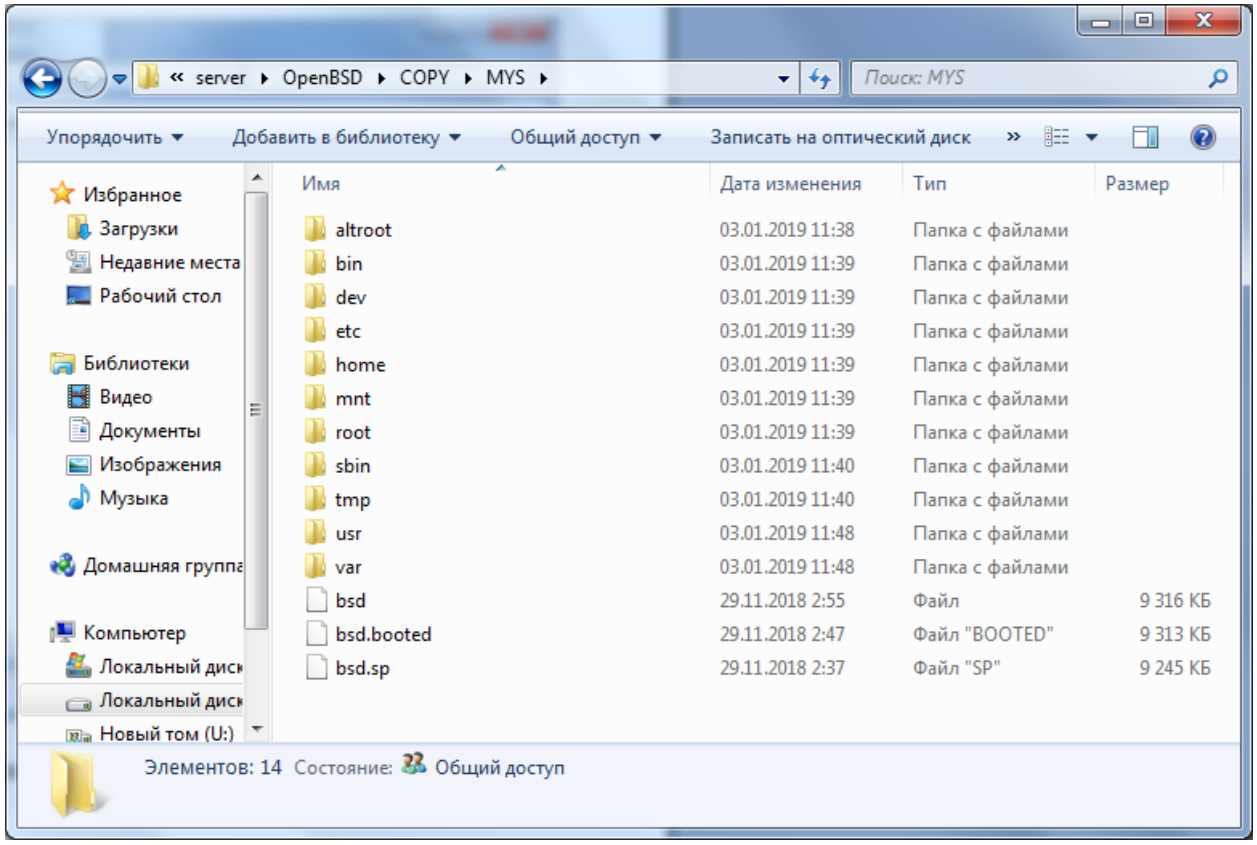



Рис. 4.8. Зміст папки з копіями файлів свого одноплатного комп’ютера

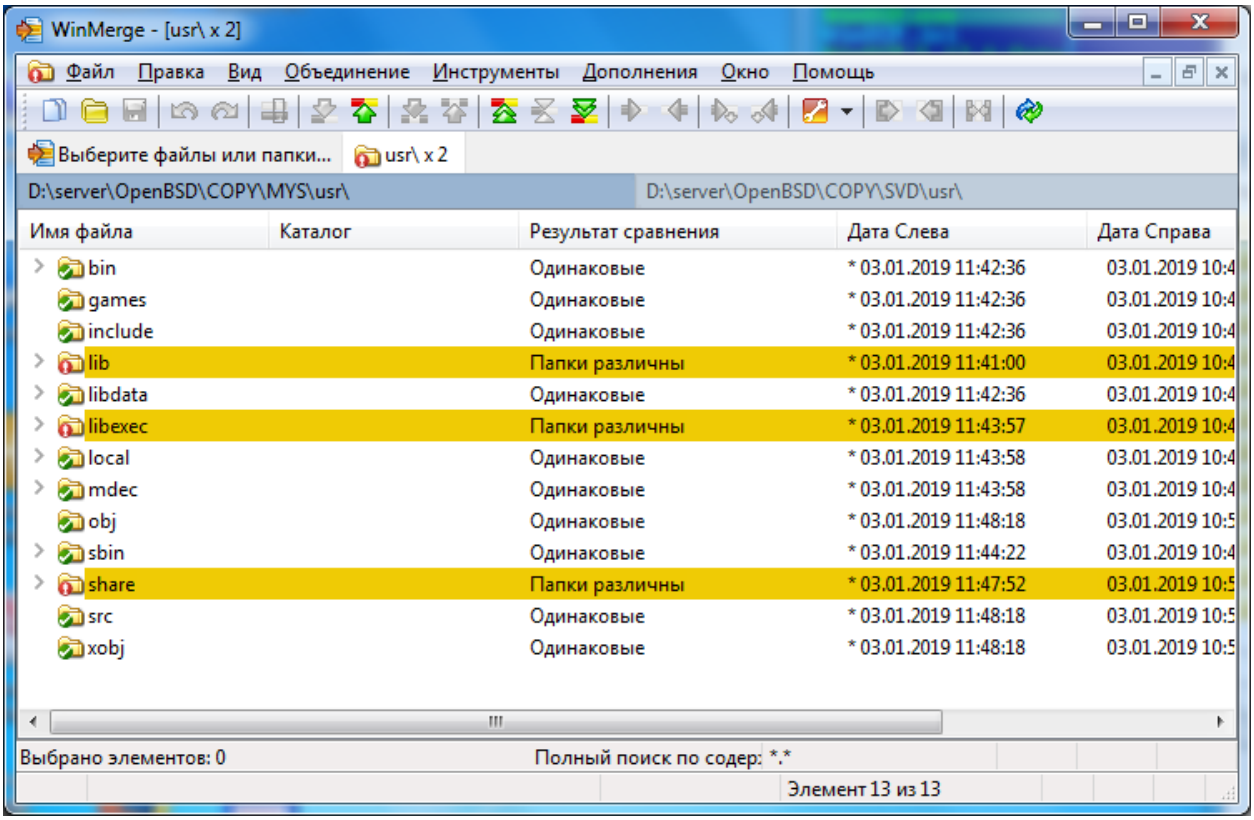



Рис. 4.9. Результат порівняння папок за допомогою програми *WinMerge*

Таблиця 4.1

**Файли, у яких виявлено розбіжності**

| Директорія | Файл | Примітка |
|---|---|---|
| /etc/iked/ private/ | local.key | Завжди різні |
| /etc/iked/ | local.pub | Завжди різні |
| /etc /isakmpd/ private/ | local.key | Завжди різні |
| /etc /isakmpd/ | local.pub | Завжди різні |
| /etc/ssh/ | ssh_host_dsa_key | Завжди різні |
| /etc/ssh/ | ssh_host_dsa_key.pub | Завжди різні |
| /etc/ssh/ | ssh_host_ecdsa_key | Завжди різні |
| /etc/ssh/ | ssh_host_ecdsa_key.pub | Завжди різні |
| /etc/ssh/ | ssh_host_ed25519_key | Завжди різні |
| /etc/ssh/ | ssh_host_ed25519_key.pub | Завжди різні |
| /etc/ssh/ | ssh_host_rsa_key | Завжди різні |
| /etc/ssh/ | ssh_host_rsa_key.pub | Завжди різні |
| /etc/ | fstab | Завжди різні |
| /etc/ | group, group.bak, installurl | Можуть співпасти |
| /etc/ | master.passwd | Завжди різні |

| | | |
|---|---|---|
| /etc/ | myname, passwd, pwd.db | Можуть співпасти |
| /etc/ | random.seed, soii.key, spwd.db | Завжди різні |
| /usr/lib/ | libc.so.92.5, libcrypto.so.44.1 | Завжди різні |
| /usr/libexec/ | ld.so, ld.so.save | Завжди різні |
| /usr/share/relink/kernel/ GENERIC.MP/ | bsd, gap.link, gap.o, lorder, newbsd.gdb, relink.log | Завжди різні |
| /usr/share/terminfo/ | Усі файли | Можуть співпасти |
| /var/backups | Усі файли | Можуть співпасти |
| /var/ cron/ | log | Завжди різні |
| /var/ db/ | dhclient.leases.smsc0, host.random, kernel.SHA256, kvm_bsd.db | Завжди різні |
| /var/ log/ | adduser, authlog, daemon, lastlog, maillog, maillog.0.gz, messages, wtmp | Завжди різні |
| /var/ mail/ | root | Завжди різні |
| /var/run/ | dmesg.boot, sshd.pid, syslog.pid, utmp | Завжди різні |
| /var/ spool/ smtpd/ purge/ | Різні назви директорій | Завжди різні |
| /var/ spool/ smtpd/queue/ | Різні назви директорій | Завжди різні |
| / | bsd, bsd.booted | Завжди різні |

Відмінності, які наведені у табл. 4.1, не можуть бути причиною будь-яких порушень справжності ОС *OpenBSD*, оскільки вони пов'язані виключно з точкою установки ОС в мережі Інтернет. При наявності ідентичності всіх інших файлів, можна отримати впевненість у справжності ОС, бо неможливо підмінити або модифікувати ОС, зберігши незмінними всі ті файли, які не відображені у табл. 4.1. Слід зазначити, що таку перевірку може виконати будь-хто, маючи лише базові знання з інформатики. Витрати часу на цю перевірку становлять близько двох годин при швидкості доступу до мережі 30 *Mb*/*s*, але без такої перевірки неможливо гарантувати автентичність ОС *OpenBSD*, без чого всі наступні перевірки втрачають ефективність.

Можуть виникнути підозри з приводу файлів, в яких є відмінності. Для того, щоб остаточно зняти можливі підозри з приводу підміни ОС можна перевірити вміст файлів, які представлено у табл. 4.1. Ці файли містять дані, які

ніяк не можуть становити небезпеку в сенсі можливості порушення роботи ОС. Для того, щоб примусити ОС працювати по-іншому потрібно замінити програмні файли, а не файли даних. На рис. 4.10 для прикладу показано вміст файлу *local.pub* з директорії */etc/iked/*.

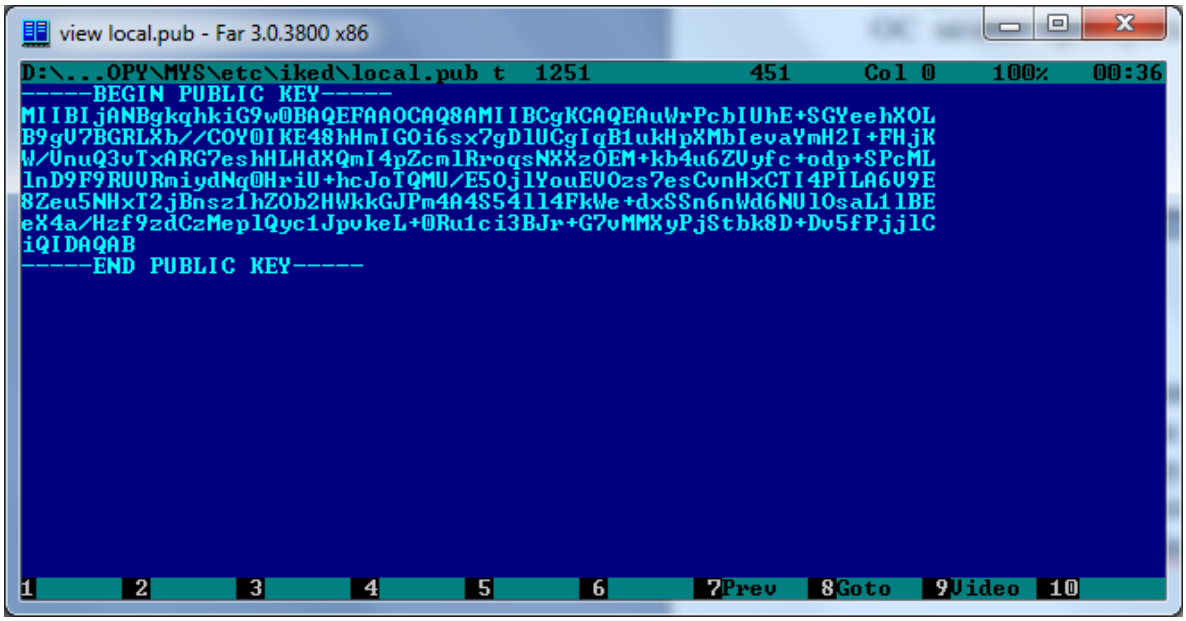



Рис. 4.10. Зміст файлу *local.pub* з директорії */etc/iked/*

Завдяки перевірці вмісту файлів, що мають відмінності, всі сумніви з приводу підробки ОС можуть бути зняті. Цю перевірку можна здійснювати дистанційно з будь-якої точки доступу до мережі.

### 4.5. Перевірка справжності апаратних засобів

Завдяки використанню міні-комп'ютерів широко відомих типів, які випускаються масовим тиражем і мають характерний зовнішній вигляд, ми отримуємо можливість протистояти спробам фальсифікації серверів виборчих дільниць. Ці сервери вимагають тільки два зовнішніх кабельних підключення, одним з яких є підключення до мережі Інтернет (з використанням типової технології Ethernet), а іншим - електроживлення (5V, 2A). Вигляд монтажу міні-комп'ютерів показано на рис. 4.11.

Рис. 4.11. Зовнішній вигляд монтажу міні-комп’ютерів

Чим більше відкритості у монтажі, тим простіше виявляти будь-які спроби фальсифікації апаратних засобів або позаштатних підключень. Завдання зловмисників значно ускладнюється тим, що недостатньо підключитися до якогось окремого пристрою, а потрібно зробити однакове підключення до усіх серверів, яких в масштабах держави може налічуватися десятки тисяч. На шляху зловмисників стоять досить складні перешкоди. Перш за все, це вхідний контроль при закупівлі та встановлення обладнання, а також можливість широкого доступу контролерів для візуальних перевірок. Важливо, що приховати фальсифікацію апаратних засобів від адміністратора неможливо, а це означає, що в усіх таких випадках є конкретна відповідальна особа. Слід зауважити, що в разі виборів державного значення спроби незаконного втручання у виборчий процес кримінально карані.

Справжність апаратних засобів допомагає контролювати сама ОС *OpenBSD*, в якій передбачена команда *sysctl hw*, що дозволяє перевіряти характеристики комп'ютера, на якому встановлена дана ОС. Результат виконання цієї команди показаний на рис. 4.12.

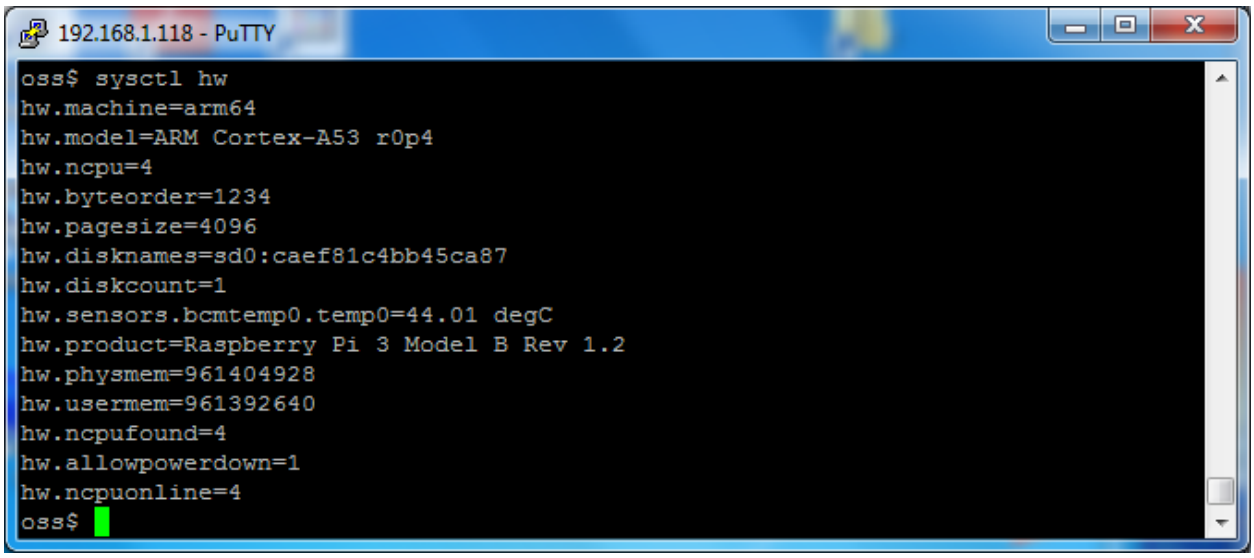



Рис. 4.12. Результат виконання команди *sysctl hw*

Порівнюючи результати виконання команди *sysctl hw* на сервері і на своєму міні-комп'ютері, бачимо, що в обох випадках це *Raspberry Pi* 3*B*.

### 4.6. Перевірка дій персоналу щодо адміністрування сервера

Щоб переконатися у відсутності порушень штатного режиму роботи адміністратором, необхідно цілодобово стежити за тим, щоб, крім відомих і заздалегідь опублікованих у вигляді графіка штатних процедур, на сервері нічого не виконувалося. В цьому відношенні надає суттєву допомогу ОС *OpenBSD*, у якій передбачено команди для контролю, що дозволяють виявити будь-яку спробу позаштатного втручання в роботу сервера. Для реалізації такого контролю в експерименті використано метод, який ми запропонували в роботі [5]. Цей метод дозволяє зафіксувати і опублікувати докази позаштатного втручання в роботу сервера або, в іншому випадку, можна переконатися в тому, що ніяких порушень політики безпеки з боку персоналу не було. Цей контроль базується на наявній у *OpenBSD* команді *ps –aux* (див. рис. 3.1), яка показує список активних процесів. Якщо цей список залишається без змін, то це гарантує відсутність будь-яких впливів на роботу ОС. Іншими словами, якщо постійно контролювати стан активних процесів на сервері, то всі потенційно небезпечні дії, можуть бути виявлені. Для зручності проведення контролю дій адміністратора треба мати графік з вказаними моментами часу початку і

завершення усіх штатних дій. Повний список цих дій зі значеннями витрат часу, які відповідають виборам державного масштабу в Україні, надано у табл. 4.2.

Таблиця 4.2

**Дії персоналу щодо адміністрування сервера виборчої дільниці**

| Найменування дії | Витрати часу для виконання | Інтервал часу до наступної дії |
|---|---|---|
| Підключення сервера до мережі Інтернет | Момент 1 | 2-3 тижні |
| Встановлення пакетів *Node.js* | 0,5-1 годин | 1 доба |
| Копіювання файлів прикладного ПО | 1-5 хвилин | 0,5-1 годин |
| Запуск прикладної програми | Момент 2 | 12-13 днів |
| Копіювання файлу з бюлетенями | 1-5 хвилин | Не визначено |
| Відключення сервера | Момент 3 | |

Важливо, щоб контролювалося не тільки виконання дій, але, щоб контроль не переривався також і в інтервалах між діями персоналу, тому що інакше неможливо буде засвідчити відсутність позаштатного втручання в роботу сервера. Таку перевірку в нашому експерименті було реалізовано за допомогою комп'ютерної програми, яку описано в роботі [105]. З програмою можна ознайомитися, а також завантажити і випробувати за посиланням http://www.asdev.com.ua/dndiasb/publications.html. Робоче вікно програми представлено на рис. 4.13.

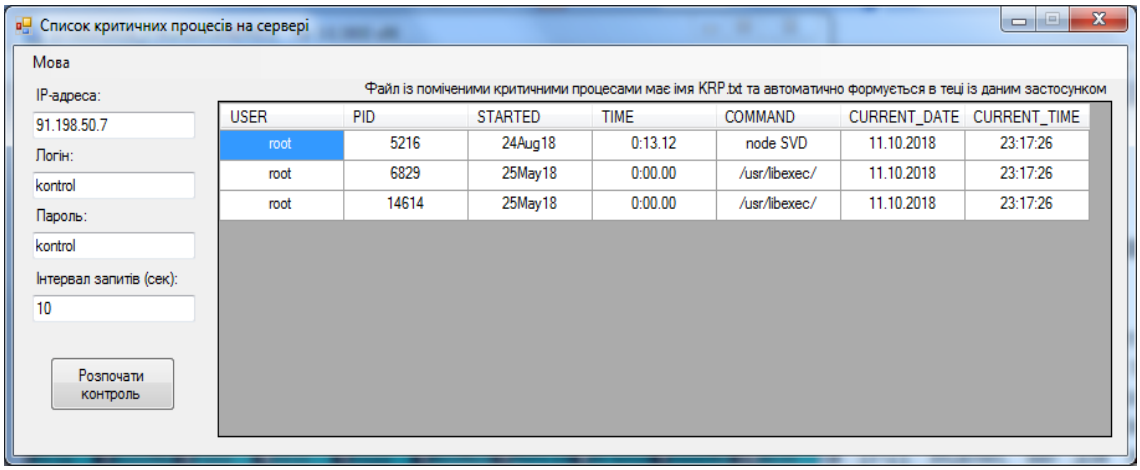



Рис. 4.13. Робоче вікно програми для контролю процесів на сервері

На початку роботи програма завантажує з файлу *PIDIGNOR.txt* список значень *PID* постійно діючих процесів ОС для ігнорування. Процеси контролерів програма також ігнорує і не фіксує. У вікні програми виводяться дані тільки про штатні і нештатні процеси, які можуть вплинути на роботу сервера. Ці ж дані заносяться у файл *KRP.txt* для документування результатів контролю. Подібну програму краще за все встановити на комп'ютері, який розміщено у тому самому сегменті мережі, що і сервер виборчої дільниці. У цьому випадку неможливо переривання контролю через перевантаження, а також неможлива реалізація атак посередника.

Опишемо повний цикл контролю дій персоналу щодо управління сервером. Після запуску програми контролю в Момент 1 (див. Табл. 4.2) і до початку установки пакетів *Node.js* ніяких процесів у файлі реєстрації *KRP.txt* не повинно з'явитись. Оскільки в цьому файлі не фіксуються процеси контролю, то єдине, що може стати причиною появи записів в цьому файлі, - введення правильного пароля адміністратора. Таке може статися тільки в трьох випадках: 1) був обраний ненадійний пароль і його розгадали; 2) адміністратор допустив розголошення пароля; 3) порушення адміністратором встановленого графіка роботи. Як бачимо, у всіх випадках до відповідальності слід залучати конкретну особу - адміністратора. У момент початку дії «Установка пакетів *Node.js*» (див. Табл. 4.2) у файлі *KRP.txt* повинен з'явитись рядок, в якому буде зафіксовано час початку виконання дії (в параметрах *STARTED* і *TIME*), а також в цьому рядку буде вказано значення *USER: root*. Далі такі рядки будуть повторюватися з інтервалом, який вибирає контролер, до завершення роботи адміністратора. Якщо інтервал обраний 10 секунд (див. Рис. 4.13), то за кожну хвилину роботи адміністратора буде з'являтися по 6 рядків. Всі параметри в цих рядках повинні бути однаковими, крім останнього, де відображається поточний час з точністю до секунди. Для того, щоб можна було проконтролювати дії адміністратора потрібно команди, які повинен ввести адміністратор, опублікувати на сайті. Ці команди представлені у табл. 4.3.

Таблиця 4.3

**Команди адміністратора щодо встановлення пакетів *Node.js***

| Команда | Метод перевірки |
|---|---|
| cd .. | Візуальне порівняння |
| cd home | Візуальне порівняння |
| cd admin | Візуальне порівняння |
| export PKG_PATH=ftp://openbsd.cs.toronto.edu/pub/OpenBSD/6.4/packages/aarch64/ | Візуальне порівняння |
| pkg_add -r node | Візуальне порівняння |
| npm install node-static | Візуальне порівняння |
| npm install nodemailer | Візуальне порівняння |
| history > haabbccdd.txt | Перевірка результату |
| exit | Перевірка в часі |

Оскільки контролер не може безпосередньо спостерігати введення кожної команди, то в обов'язок адміністратора перед завершальною командою ***exit*** введено виконання команди ***history > haabbccdd.txt*** , де замість букв ***aabbccdd*** слід вказати дату і час завершення сеансу роботи, а саме так: ***aa*** – номер місяцю, ***bb*** – число, ***cc*** – години, ***dd*** – хвилини. Таке рішення дозволяє контролеру перевірити всі команди адміністратора після завершення процесу установки пакетів. Всі команди, крім двох останніх будуть занесені командою ***history > haabbccdd.txt*** у файл ***haabbccdd.txt***, який буде розміщений у директорії */home/admin/*. Після виконання цієї дії з'явиться директорія */node_modules*, файли у якій можна не перевіряти, тому що досить перевірити команди адміністратора у файлі ***haabbccdd.txt*** . Інтервал до наступної дії (1 день) передбачений для того, щоб у разі аварії або помилкової команди можна було цю дії виконати повторно. Також слід порівняти час завершення дії з останнім записом у файлі *KRP.txt*, щоб виключити можливість виконання будь-яких команд перед завершенням роботи. Наступна перевірка дій адміністратора пов'язана з процедурою копіювання файлів прикладного ПЗ. При цьому у файлі *KRP.txt* під час роботи адміністратора з'являються записи з параметром *USER: admin*. Оскільки

користувач *admin* не може виконувати дії поза директорії *home/admin /*, то досить перевірити появу трьох нових файлів у директорії *home/admin*, вміст яких є заздалегідь відомим. Ці файли описані в перших трьох рядках табл. 4.4.

**Таблиця 4.4**

**Файли прикладного програмного забезпечення (ПЗ)**

| Им'я файлу | Опис |
|---|---|
| *VDnnnnnn.js* | Серверна програма, яка буде керувати сервером після її запуску монопольно до відключення сервера |
| *VDnnnnnn.DBT* | Вибірка даних про виборців, де усі конфіденційні дані зашифровані |
| *PWnnnnnn.html* | Клієнтська частина програми для занесення виборцями паролів для голосування |
| *AVnnnnnn.html* | Клієнтська частина програми з бюлетенями для голосування |

На місці символів nnnnnn розміщено номер виборчої дільниці.

Після завершення дій адміністратора щодо копіювання файлів прикладного ПЗ і до моменту запуску прикладної програми (Момент 2) необхідно переконатися, що у цих файлах немає відхилень від заздалегідь опублікованого на сайті варіанту. Від Моменту 2 починається автоматичний відлік інтервалів часу для всіх наступних процедур, включаючи голосування, тому запуск прикладної програми слід виконати точно в зазначений момент часу. Для того, щоб забезпечити можливість перевірки правильності запуску прикладної програми, адміністратор зобов'язаний виконати послідовність команд, яку наведено у табл. 4.5.

Таблиця 4.5

**Команди адміністратора щодо запуску прикладної програми**

| Команда | Метод перевірки |
|---|---|
| *nohup node VDnnnnnn &* | Візуальне порівняння |
| *history > h*05211215*.txt* | Перевірка результату |
| *exit* | Перевірка в часі |

Як і в разі копіювання файлів прикладного ПО, контролери можуть починати спостереження з появи у файлі *KRP.txt* записів з параметром *USER:*

*admin*. Після запуску прикладної програми в файлі *KRP.txt* ще додадуться нові рядки зі значенням *USER: admin*, але з іншим значенням *PID* і значенням *COMMAND:VDnnnnnn*. Це свідчить про початок роботи прикладної програми, яка буде керувати сервером до відключення (Момент 3). Для того, щоб впевнитись у тому, що дійсно запущено штатну прикладну програму, треба перевірити вміст файлу h05211215.txt, де має бути тільки один рядок з першою командою (див. табл. 4.5). Важливо, що описана послідовність дій контролерів надає можливість впевнитись у правильному функціонуванні усіх програмно-апаратних засобів сервера виборчої дільниці.

### 4.7. Перевірка процесу функціонування сервера виборчої дільниці

Описані перевірки дозволяють впевнитись, що з Моменту 2 управління сервером виборчої дільниці здійснюється виключно прикладною програмою.

Крім того, показано, що всі програмно-апаратні засоби, включаючи прикладну програму, яка з моменту її запуску монопольно керує сервером, є штатними. Завдяки безперервному контролю щодо позаштатних дій з боку персоналу або зловмисників, можемо виявити будь-який процес, якій міг би перешкоджати нормальному функціонуванню прикладної програми. У разі виявлення подібних процесів можуть бути прийняті оперативні рішення щодо усунення виявлених перешкод.

Експеримент щодо перевірки роботи прикладної програми проведено за участю студентів Факультету автоматизації і інформаційних технологій Київського національного університету будівництва і архітектури (КНУБА) через сайт виборів представників до органів студентського самоврядування, на який можна увійти через посилання http://vybir.knuba.edu.ua/. З метою створення умов подібних до державних виборів в Україні кожному студенту надано ідентифікатор, який за структурою відповідає номеру паспорта громадянина України, а номери виборчих дільниць мають форму таку, як прийнято в Україні.

Головна мета цієї перевірки полягає у вимірюванні експлуатаційних характеристик сервера виборчої дільниці та виявленні можливих слабких місць в процесі обслуговування виборців. Для цього у меню вибору дільниці додані два

спеціальних режими: «Пробний разовий пароль» та «Пробне голосування». Наявність цих режимів дозволила моделювати екстремальні умови, які можуть виникати під час реального голосування, і випробувати поведінку системи у разі підвищення інтенсивності потоку запитів виборців до сервера. Вигляд головної сторінки системи студентського голосування у мережі Інтернет представлено на рис. 4.14, а додаткові пункти у меню для вибору дільниці показані на рис. 4.15. Вигляд повідомлень діалогу виборця з сервером під час отримання одноразового паролю для голосування показано на рис. 4.16, а під час голосування – на рис. 4.17 та 4.18.

Вихідні тексти програм, що були задіяні під час експерименту, наведені у додатку А.

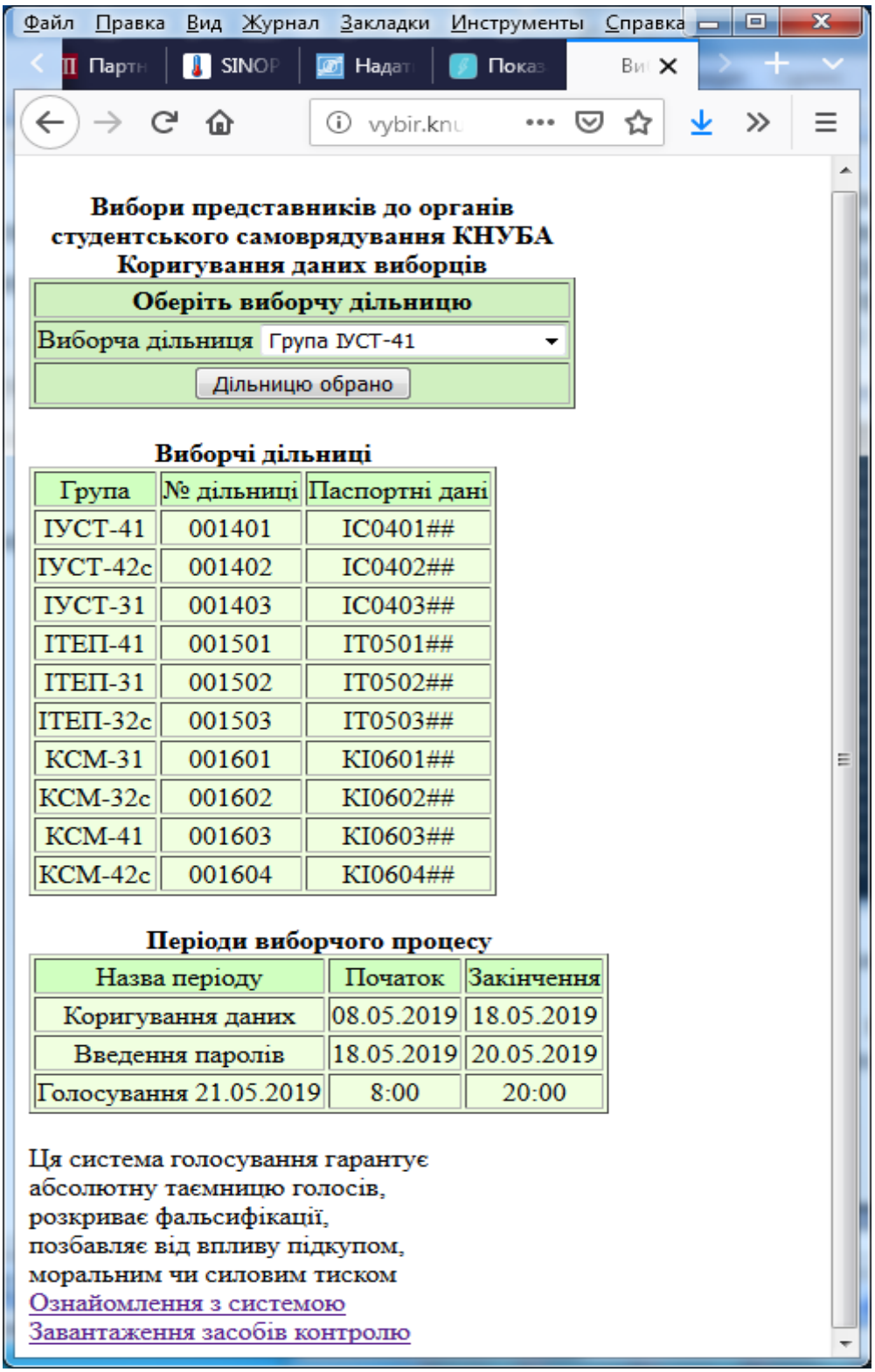



Рис. 4.14. Головна сторінка системи студентського голосування у мережі Інтернет

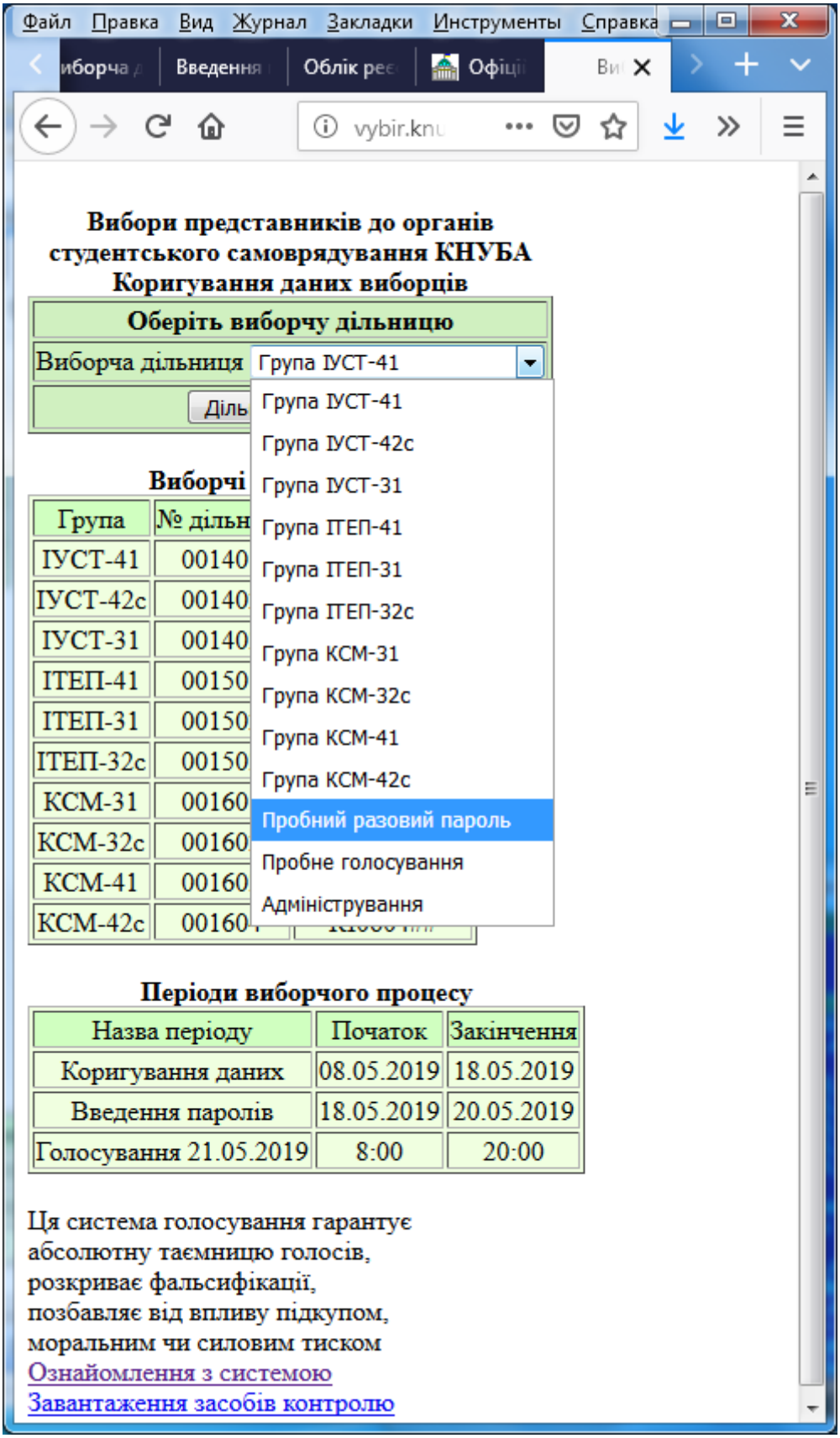
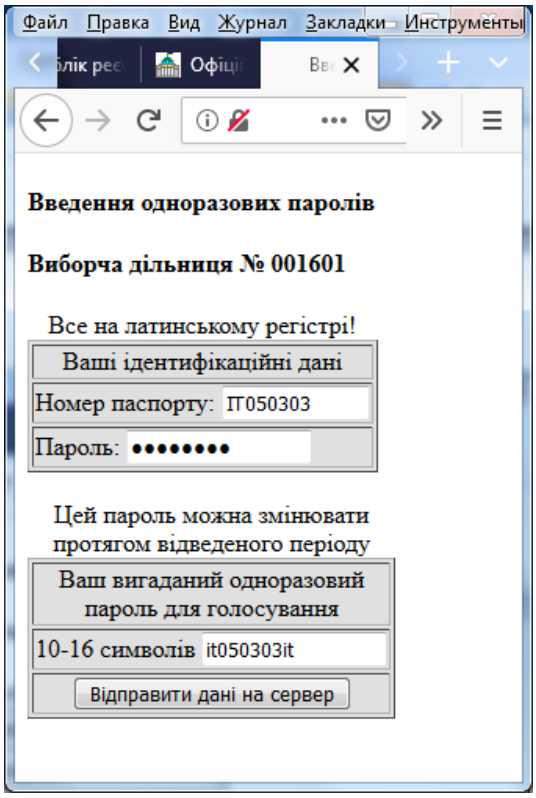
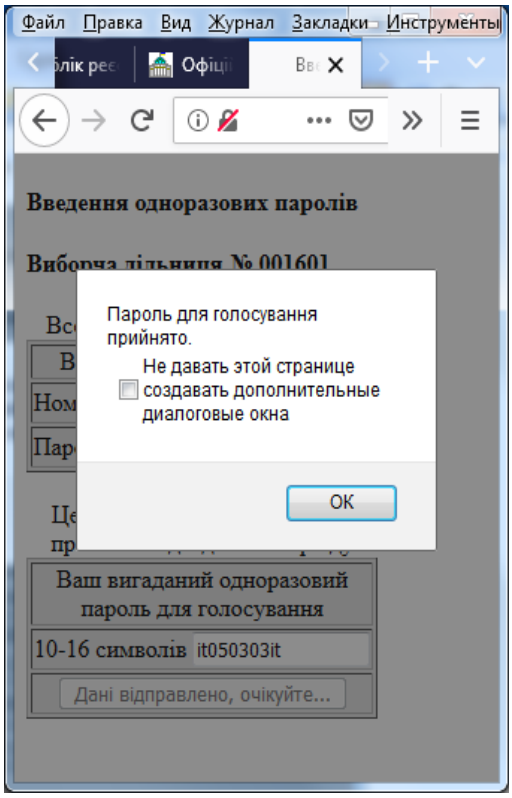



Рис. 4.15. Вигляд меню з додатковими пунктами для випробувань

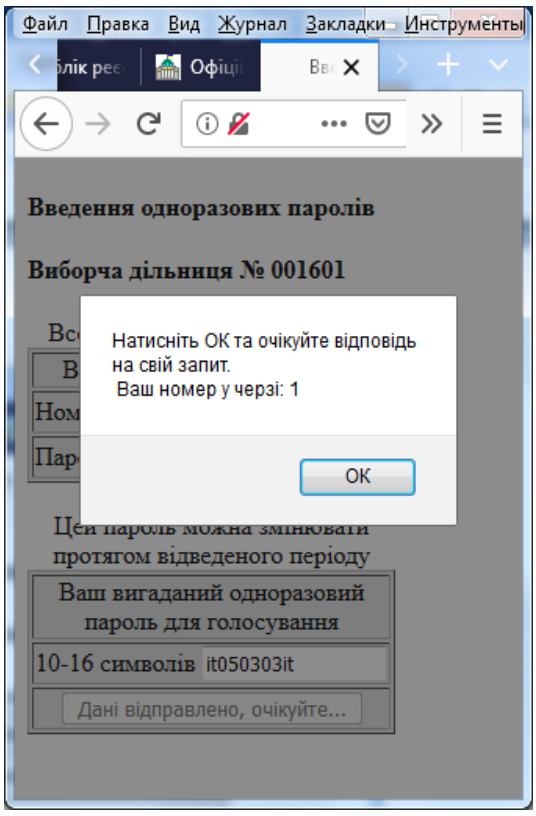
а)




в)




б)




г)

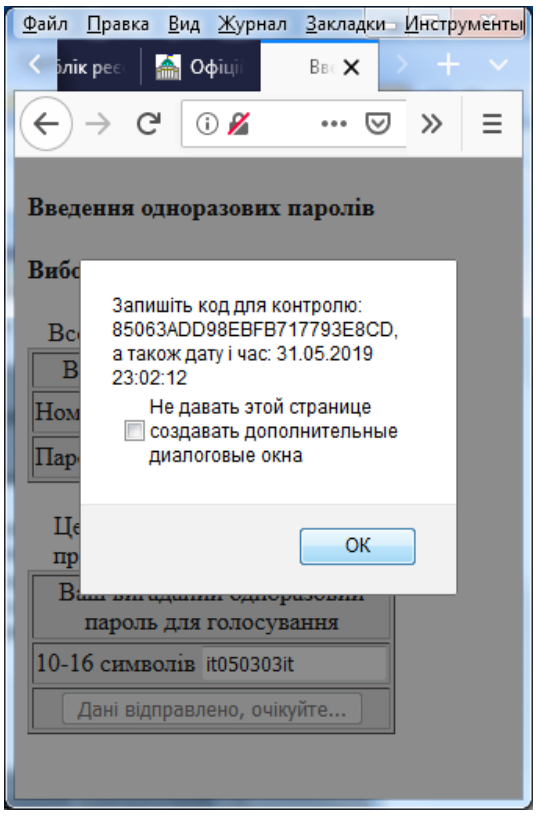



Рис. 4.16. Діалог з сервером під час отримання одноразового паролю

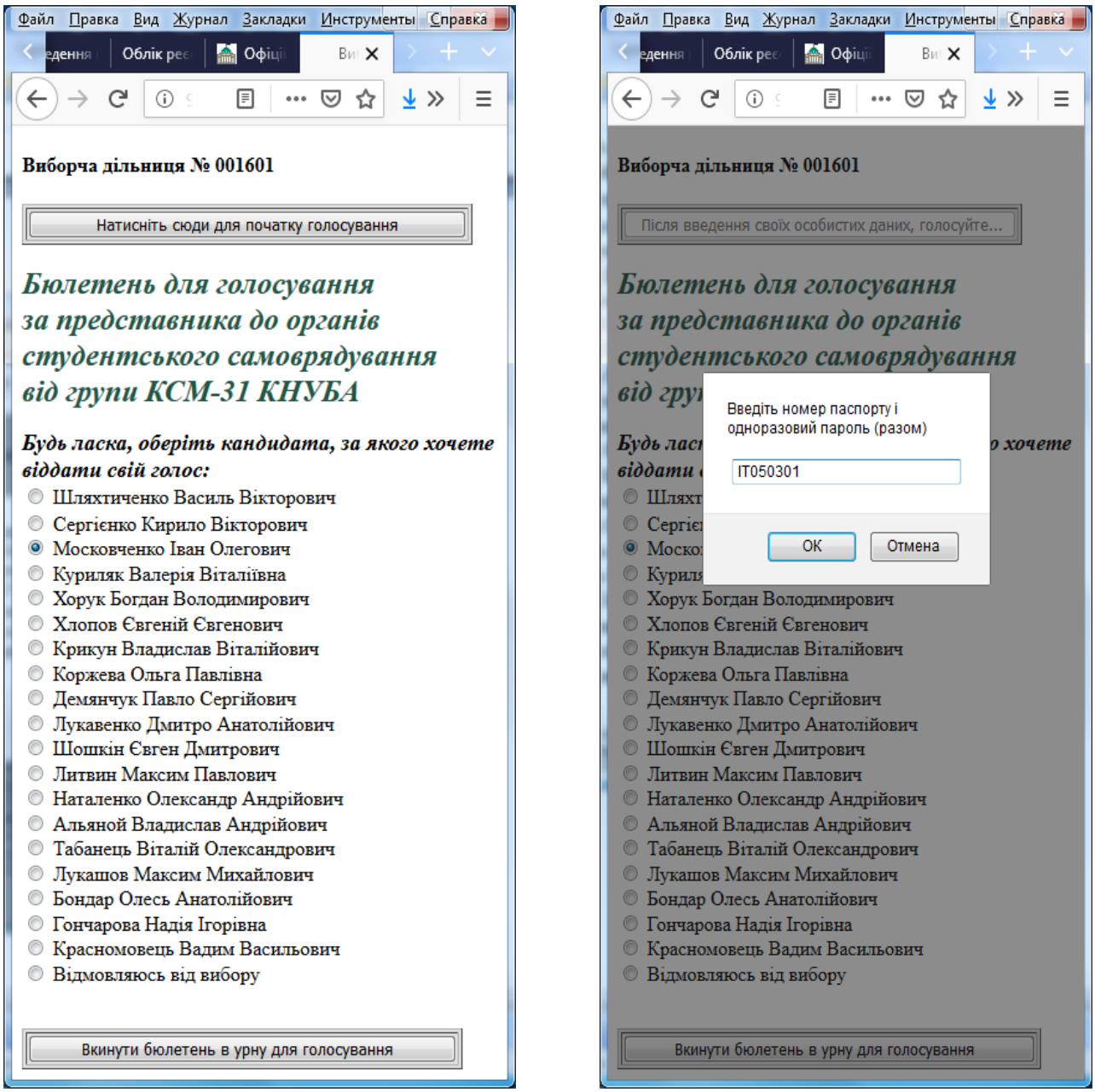



Рис. 4.17. Початкові повідомлення діалогу виборця з сервером виборчої дільниці під час голосування

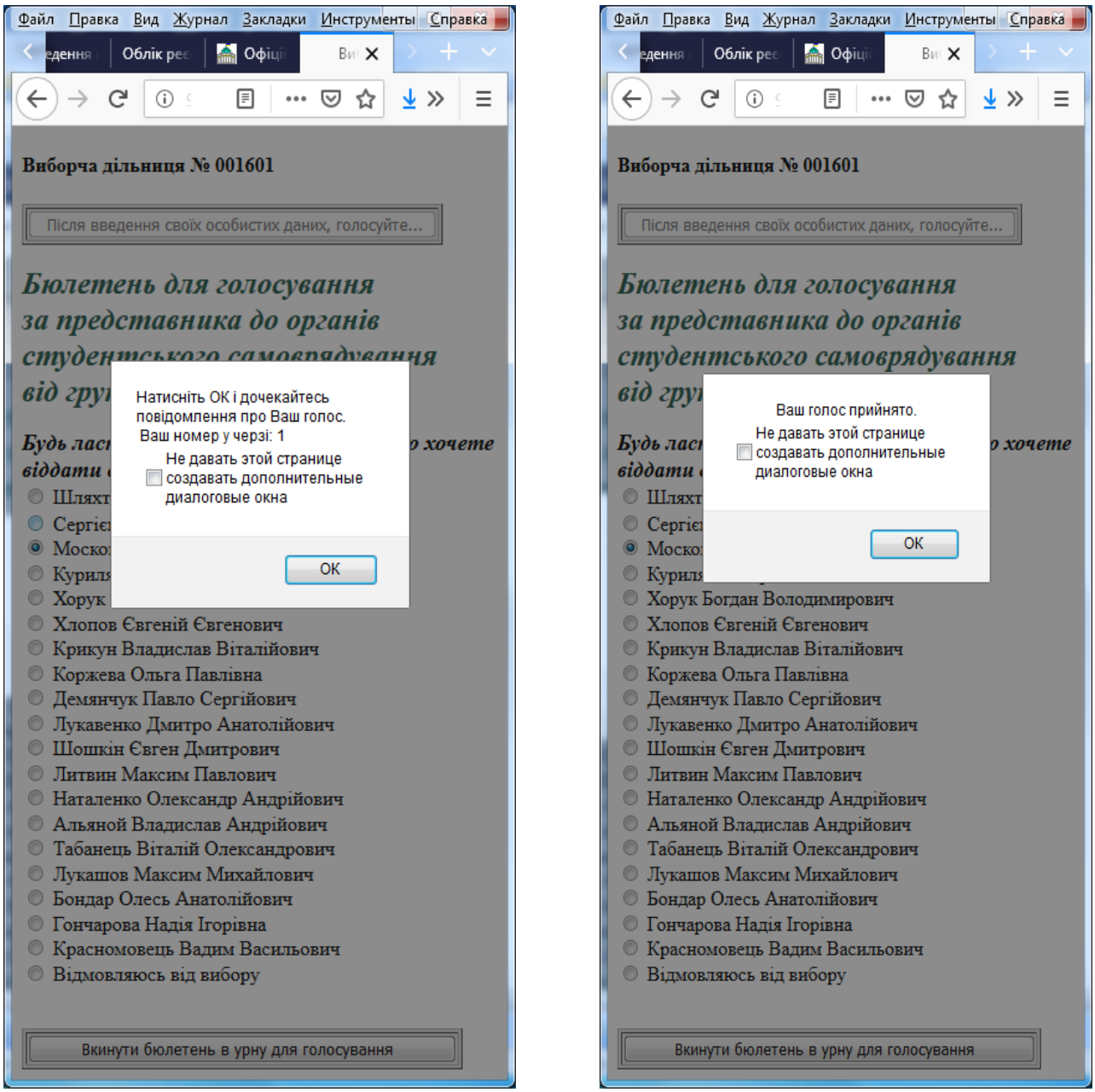



Рис. 4.18. Завершальні повідомлення діалогу виборця з сервером виборчої дільниці під час голосування

Результати вимірювання часу обробки запитів виборців сервером виборчої дільниці представлені у табл. 4.6.

Таблиця 4.6

**Результати вимірювання часу обробки запитів виборців сервером на міні-комп'ютері *Raspberry Pi* 3*B***

| Найменування запиту | Час обробки сервером, с | Час очікування відповіді клієнтом, с |
|---|---|---|
| Процедура введення одноразових паролів для голосування | | |
| Виклик *HTML* документу | <0.01 | 1-5 |
| Обмін ключами | <0.01 | <1 |
| Відправка даних з отриманням номеру в черзі на обробки | <0.1 | 1-5 |
| Отримання результату обробки | 2 (на кожний запит з черги) | 2*q (q – номер в черзі) |
| Отримання коду для перевірки | <0.01 | <1 |
| Процедура голосування | | |
| Виклик *HTML* документу | <0.01 | 1-5 |
| Обмін ключами | <0.01 | <1 |
| Відправка даних з отриманням номеру в черзі на обробки | <0.1 | 1-5 |
| Отримання результату обробки | 2 (на кожний запит з черги) | 2*q (q – номер в черзі) |
| Отримання коду для перевірки | <0.01 | <1 |

Отримані результати вимірювання часу обробки запитів сервером, а також часу очікування відповіді від сервера показують, що у разі одночасного звернення до сервера 30 виборців затримка обробки запитів сервером не перевищить двох хвилин, а під час голосування сервер здатен обслуговувати за годину більше ніж 1500 виборців. Це свідчить про те що, застосування даної системи на міні-комп'ютерах *Raspberry Pi* 3*B* цілком задовольняє вимогам щодо швидкодії обслуговування, бо кількість виборців на дільниці не може перевищити 2500.

## Висновки до четвертого розділу

1. Вибір програмно-апаратних засобів під час експериментального дослідження зроблено таким чином, щоб задовольнити вимогам проведення

голосування на державному рівні через публічну мережу Інтернет в умовах України.

2. Продемонстровано на прикладі конкретної програмно-апаратної платформи у складі міні комп'ютерів ***Raspberry Pi* 3*B*, операційної системи *OpenBSD*** та програмного забезпечення *Node.js* можливість реалізації усіх необхідних складових транспарентної системи дистанційного таємного голосування в мережі Інтернет.

3. Показана можливість перевірки клієнтами мережі Інтернет справжності операційної системи *OpenBSD*, яка встановлена на сервері виборчої дільниці, з використанням на клієнтському комп'ютері стандартних програм *WinSCP* та *WinMerge*, які вільно розповсюджуються у мережі.

4. Показана можливість перевірки справжності апаратних засобів серверів виборчих дільниць дистанційно за допомогою команд операційної системи та шляхом візуального порівняння встановленого обладнання зі стандартними аналогами.

5. Продемонстрована можливість контролю користувачами Інтернету дій персоналу щодо встановлення та обслуговування програмного забезпечення на серверах виборчих дільниць, що дозволяє впевнитись у відсутності будь-яких відхилень від штатного порядку.

6. Показана на прикладі конкретної програмно-апаратної реалізації можливість побудови траспарентної системи, де ключову роль при перевірці точності функціонування грають користувачі мережі Інтернет, включаючи всіх без винятку громадян, які мають бажання проконтролювати вірність функціонування системи. Це дозволяє усунути привід для підозр у тому, що хтось має можливість для прихованих від громадян фальсифікацій штатного режиму роботи системи, бо кожному надається можливість контролю.

7. Отримані результати вимірювання часу обробки запитів сервером, а також часу очікування відповіді від сервера показують, що застосування системи ДТГ на міні-комп'ютерах є доцільним впроваджувати вже сьогодні.

# ОСНОВНІ РЕЗУЛЬТАТИ ТА ВИСНОВКИ

У дисертаційній роботі, відповідно до поставленої мети, розв’язано актуальну науково-технічну задачу гарантування неможливості виникнення порушень цілісності результатів та конфіденційності персональних даних в системах дистанційного таємного голосування (ДТГ), що усуває будь-які підстави для недовіри з боку голосуючих щодо можливості реалізації вказаних порушень.

У процесі виконання дисертаційної роботи отримані такі основні результати:

1. Здійснено аналіз характеристик існуючих технологій ДТГ на предмет наявності у них дієвих механізмів контролю коректності їхнього функціонування з боку суспільства, оскільки вони забезпечують гарантії збереження таємниці голосів та істинності результатів волевиявлення за умов, коли відсутня довіра до всіх без винятку суб’єктів, що можуть бути зацікавлені у результатах волевиявлення. Проаналізовано також можливості існуючих технологій щодо якості надання послуг виборцям та технічної підтримки процесу волевиявлення. Виявлено наступні недоліки існуючих технологій ДТГ:

1) відсутність під час голосування виконувати аудит дій адміністратора щодо управління сервером голосування та виявляти атаки посередника, що негативно впливає на рівень довіри громадян до процесу волевиявлення;

2) відсутність автоматизованого пошуку виборцями *IP* адрес своїх виборчих дільниць, що створює незручності і потребує зайвих витрат часу;

3) відсутність автоматизації процесу автентифікації виборців, що обумовлює необхідність здійснення обтяжливих очних перевірок осіб виборців в період уточнення списків голосуючих перед кожним актом волевиявлення.

Тому були визначені як актуальні наукові завдання розробки методів забезпечення транспарентної роботи засобів ДТГ і автоматизації процесів пошуку виборцями *IP* адрес серверів їхніх виборчих дільниць та автоматизації процесів, які ускладнюють процедуру волевиявлення, потребуючи зайвих витрат часу.

2. Розроблено структурно-функціональну модель автоматизації пошуку виборцями *IP* адрес серверів їхніх виборчих дільниць, а також протокол інформаційної взаємодії елементів цієї моделі. Оскільки пошук *IP* адрес є довготривалою процедурою, то передбачено застосування лінійки одночасно працюючих серверів для цього пошуку. Згідно розробленого протоколу, запити виборців приймаються сервером, який розподіляє весь потік запитів між серверами пошуку *IP* адрес. Застосування даної моделі скорочує витрати часу на здійснення актів волевиявлення.

3. Розроблено метод балансування (вирівнювання) навантаження на одночасно працюючі сервери, що входять до складу лінійки серверів автоматизованого пошуку *IP* адрес. Оскільки потік запитів характеризується непередбачуваними пульсаціями, то для вирівнювання навантаження на ці сервери обрано адаптивний принцип роботи, що здійснює розподіл потоку звернень виборців з урахуванням динаміки змін як інтенсивності потоку звернень, так і тривалості пошуку адрес кожним із серверів. Регулятор діє у напрямку вирівнювання значень коефіцієнтів завантаження серверів, тим самим запобігаючи можливим перенавантаженням в роботі лінійки серверів і, от же, можливим затримкам в обслуговуванні виборців.

4. Дістав подальший розвиток метод автентифікації виборців у системі ДТГ, котрий за рахунок введення спеціалізованих серверів дозволу введення паролю, що містять бази даних з біологічними або іншими ознаками виборців, та створення захищених з’єднань між цими серверами з сервером голосування, забезпечує можливість дистанційної додаткової автентифікації осіб виборців. Реалізація цього методу надає виборцям можливість позбутися обов’язкової очної перевірки перед кожним актом волевиявлення.

5. Запропоновано модель безперервного автоматизованого аудиту виборцями програмно-апаратних засобів сервера голосування за рахунок використання відкритого для перевірки монтажу міні комп’ютерів та автоматизації процедур аудиту за допомогою спеціалізованого сервера, який підключено до сервера голосування через спільну локальну мережу, а доступ

виборців до нього реалізовано через захищений канал, де центр сертифікації обирають представники виборців, при цьому інсталяція та запуск серверів виконується під наглядом виборців або їх довірених осіб у період часу, коли на серверах ще немає ніякої критичної інформації, а після запуску серверів виборці продовжують аудит дистанційно без втрати інформації про наявність чи відсутність втручань у роботу серверів, бо усі спроби таких втручань виявляються та реєструються сервером аудиту, що забезпечується спеціально розробленим програмним забезпеченням та відкритими для виборців правилами адміністрування і реєстрацією кодів з'єднань з виборцями на сервері голосування, що дозволяє усунути можливість підозри про те, що сервер голосування являє собою «чорний ящик» з імітатором який демонструє виборцям нібито чесне голосування, а насправді розкриває і підмінює їхні голоси, бо така підозра руйнує довіру виборців.

6. Розроблена модель безперервного автоматизованого аудиту програмно-апаратних засобів сервера голосування дозволяє виборцям під час голосування самостійно виявляти атаки посередника шляхом порівняння кодів з'єднання, що надаються у повідомленнях під час голосування зі значеннями, які реєструються у журналі з'єднань на сервері голосування, до якого виборці отримують доступ через захищений протоколом *HTTPS* канал зв'язку з сервером аудиту.

7. Продемонстровано на прикладі конкретної програмно-апаратної платформи у складі міні-комп'ютерів *Raspberry Pi* 3*B*, операційної системи *OpenBSD* та програмного забезпечення *Node.js* можливість коректної реалізації усіх необхідних складових транспарентної системи ДТГ. Все це підтверджено під час проведення реальних виборів керівних органів Товариства Червоного Хреста України 4 грудня 2020 року, де виборці голосували з різних областей України не покидаючи своїх міст.

8. Отримані результати вимірювання часу обробки запитів сервером, а також часу очікування відповіді від сервера показують, що у разі одночасного звернення до сервера 30 виборців затримка обробки запитів сервером не перевищить двох хвилин, а під час голосування сервер здатен обслуговувати за

годину більше ніж 1500 виборців. Це свідчить про те що, застосування даної системи на міні-комп’ютерах *Raspberry Pi* 3*B* цілком задовольняє вимогам щодо швидкодії обслуговування, бо кількість виборців на дільниці не може перевищити 2500.

9. В умовах організації можливості альтеративного вибору порядку голосування (дистанційного через Інтернет або з безпосередньою фізичною присутністю на виборчій дільниці) використання міні-комп’ютерного обладнання є доцільним вже у теперішній час.

10. У разі прийняття відповідного виборчого законодавства впровадження транспарентних систем ДТГ, побудованих з використанням міні-комп’ютерів, є можливим і доцільним вже сьогодні.

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Пригара М.П. Захищена система технічної підтримки процесів дистанційного волевиявлення. – Рукопис. Дисертація на здобуття наукового ступеня кандидата технічних наук за спеціальністю 05.13.21 – системи захисту інформації. –К.: НАУ, 2018.

2. Вишняков В.М., Пригара М.П., Воронін О.В. Відкрита система таємного голосування. Управління розвитком складних систем, 2014, №20.- С. 110 – 115.

3. Генерування випадкових чисел штатними засобами хостів мережі Інтернет./ В.М. Чуприн, В.М.Вишняков, М.П. Пригара // Захист інформації. – 2016. – Т. 18, №4. – С. 323-335.

4. Метод протидії незаконному впливу на виборців у системі Інтернет голосування./ В.М. Чуприн, В.М. Вишняков, М.П. Пригара - Безпека інформації. – 2017. – Т. 23, №1. – С. 7-14.

5. Захист операційного середовища систем Інтернет голосування / М. Чуприн, В.М.Вишняков, М.П. Пригара // Захист інформації. – Т. 19, №1. – С. 56 – 66.

6. Вишняков В.М., Пригара М.П. Забезпечення свободи волевиявлення в системі Інтернет-голосування (ІГ). Матеріали 4-ї Міжнародної наукової конферен- ції ICS-2015 «Інформація, комунікація, суспільство 2015» , С. 124 – 125.

7. Альомар Мхамад. Удосконалення технології управління розподілом ресурсів пакетних мереж. Дисертація на здобуття наукового ступеня кандидата технічних наук за спеціальністю 05.13.05 – комп'ютерні системи та компоненти. –К.: НАУ, 2016.

8. Сабанов А. Г. Вопросы доверия при построении электронного правительства. Информационно-методический журнал «Защита информации. Инсайд» № 2 2010. [Online]. Available: http://www.inside-zi.ru/pages/2_2010/66.html

9. Acemyan C.Z., Kortum P., Byrne M.D., Wallach D.S. Usability of Voter Verifiable, End-to-endVoting Systems: Base line Data for Helios, Prêt à Voter, and Scantegrity II. USENIX Journal of ElectionTechnology and Systems (JETS), 2014, Vol. 2, No 3, pp. 26 – 56.

10. Lombardi E. Electronic Vote & Democracy. [Online]. Available: http://www.electronic-vote.org (дата звернення: 25.11.2018).

11. Schneier B. Unusual Electronic Voting Machine Threat Model. [Online]. Available: https://www.schneier.com/blog/archives/2014/05/unusual_electro.html (дата звернення: 10.11.2018).

12. Lessons from the EVOTE 2014 International Conference, November 27, 2014. [Online]. Available: http://e-lected.blogspot.com (дата звернення: 15.11.2018).

13. Schneier B. What's Wrong With Electronic Voting Machines? [Online]. Available: https://www.schneier.com/essays/archives/2004/11/whats_wrong_with_ele.html (дата звернення: 10.21.2018).

14. Міжнародний пакт про громадянські і політичні права (ратифіковано Указом Президії Верховної Ради Української РСР від 19.10.73 № 2148-VIII).

15. Закон України «Про вибори Президента України». Редакція від 28.02.2019, https://zakon.rada.gov.ua/laws/show/474-14 (дата звернення: 10.05.2019).

16. Закон України «Про вибори народних депутатів України». Редакція від 28.02.2019, https://zakon.rada.gov.ua/laws/show/4061-17 .

17. Закон України «Про місцеві вибори». *Відомості Верховної Ради України.* 2015. № 37-38. Ст.366.

18. Henman P., Ackland R., Graham T. Community Structure in e-Government Hyperlink Networks. *14th European Conference on e-Government (ECEG)*. Brasov (Romania), 2014. Pp. 135 – 143.

19. Kalvet1 T. Innovation: a factor explaining e-government success in Estonia. *Electronic Government*: International Journal, 2012, Vol. 9, No. 2, pp. 142 – 157.

20. Margolis M., Moreno-Riaño G. The Prospect of Internet Democracy. *Surrey, UK*: Ashgate Publishing Company Brookfield (USA), 2009. 200 p.

21. Springall D., Finkenauer T., Durumeric Z. Security Analysis of the Estonian Internet Voting System. *Proceedings of the ACM SIGSAC* Conference on Computer and Communications Security (CCS '14), 2014, pp. 703 – 715.

22. Segaard S.B., Christensen D.A., Jo Saglie B.F. Internettvalg: Hva gjor og mener velgerne? Rapport 2014:07. Oslo, Institutt for samfunnsforskning, 2014. 146 s.

23. The future of Internet voting in the US, October 22, 2015. [Online]. Available: http://e-lected.blogspot.com (дата звернення: 15.11.2018).

24. Ernest et al. Blockchain electronic voting system and method. Patent US 2017/0109955 Pub. Date: Apr. 20, 2017.

25. Ali Mohammed A., Timour R.A. Efficient E-voting Android Based System. International Journal of Advanced Research in Computer Science and Software Engineering, 2013, Vol. 3, Issue 11. Pp. 43 – 48.

26. Acemyan C.Z., Kortum P., Byrne M.D., Wallach D.S. Usability of Voter Verifiable, End-to-end Voting Systems: Baseline Data for Helios, Prêt à Voter, and Scantegrity II. USENIX Journal of Election Technology and Systems (JETS), 2014, Vol. 2, No 3, pp. 26 – 56.

27. Smith C.M. Convenience Voting and Technology: The Case of Military and Overseas Voters. New York, Palgrave Macmillan, 2014. 240 p.

28. Springall D., Finkenauer T., Durumeric Z. Security Analysis of the Estonian Internet Voting System. Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS '14), 2014, pp. 703 – 715.

29. Виктор Мусияка: Кому и зачем внедрять электронное голосование? [Online]. Available: http://www.3republic.org.ua/ru/ideas/11613 (дата звернення: 15.01.2019).

30. Buchmann J., Neumann S., Volkamer M. Tauglichkeit von Common Criteria-Schutzprofilen für Internetwahlen in Deutschland. Datenschutz und Datensicherheit-DuD. Springer Fachmedien Wiesbaden, 2014, Vol. 38, Issue 2, pp. 98 – 102.

31. Ali Mohammed A., Timour R.A. Efficient E-voting Android Based System. International Journal of Advanced Research in Computer Science and Software Engineering, 2013, Vol. 3, Issue 11, Pp. 43 – 48.

32. Модуль інтернет-голосування. [Online]. Available: https://arma.gov.ua/modul-inet-vote (дата звернення: 05.10.2018).

33. Спосіб електронного голосування: пат. 39582 Україна: МПК G07C 13/00. № 200900174; заявл. 10.01.2009; опубл. 25.02.2009, Бюл. № 4, 4 с.

34. Вибори до Верховної Ради: 5 інтернет-проектів у допомогу виборцям. URL: http://basicgroup.ua/vybory-do-verkhovnoi-rady-5-internet-proektiv-u-dopomogu-vybortsiam (дата звернення: 05.10.2018).

35. Інтерактивна карта виборчих порушень – статистика і узагальнення тенденцій. [Online]. Available: http://maidanua.org/vybory2012/stats/ (дата звернення: 15.11.2018).

36. У Рівному презентовано новий проект Регламенту міської ради [Online]. Available: http://opora.rv.ua/novini/1399-u-rivnomu-prezentuvaly-proekt-rehlamentu-miskoi-rady-2.html. (дата звернення: 05.10.2018).

37. Моніторинг виборів до місцевих рад [Online]. Available: http://tusovka.kr.ua/news/2015/09/23/monitoring-viboriv-do-mistsevih-rad-gromada-na-storozhi-zakonu (дата звернення: 05.10.2018).

38. eCast – супроводження дня голосування [Online]. Available: http://basicgroup.ua/ecast (дата звернення: 05.10.2018).

39. Електронні петиції. Офіційне Інтернет-представництво Президента України [Online]. Available: https://petition.preside (дата звернення: 05.10.2018).

40. Інтернет-вибори Президента України зірвано [Online]. Available: http://politiko.ua/blogpost17340 (дата звернення: 05.10.2018).

41. Спосіб електронного голосування: пат. 91920 Україна: МПК G07C 13/00. № 200900175; заявл. 10.01.2009; опубл. 10.09.2010, Бюл. № 17, 4 с.

42. Гірник Д.А., Сергієнко О.В. Створення АС комп'ютерної бази даних єдиного реєстру документів дозвільного та декларативного характеру в будівництві. *Новітні комп'ютерні технології.* Випуск XI. ДВНЗ "Криворізький національний університет", 2013. С. 155 -158.

43. Вовк А.І., Гірник Д.А., Підлужня В.А. Концепція автоматизованої системи аудиту та моніторингу енергоефективності будівель. *Інформаційно-комп'ютерні технології:* зб. тез допов. VII Міжнародна науково-технічна конференція. Житомир, МОН України, 2014. С. 13-14.

44. Neumann P.G. Security Criteria for Electronic Voting. Proceedings of National Computer Security Conference, 1993 (16th), Baltimore (USA). DIANE Publishing, 1995, p.p. 478 – 482.

45. Buchmann J., Neumann S., Volkamer M. Tauglichkeit von Common Criteria-Schutzprofilen für Internetwahlen in Deutschland. Datenschutz und Datensicherheit-DuD. Springer Fachmedien Wiesbaden, 2014, Vol. 38, Issue 2, pp. 98 – 102.

46. Cranor L.F., Cytron R.K. Sensus: a security-conscious electronic polling system for the Internet. Proceedings of the Thirtieth Hawaii International Conference on System Sciences. Wailea (USA), 1997, vol.3, pp. 561 – 570.

47. Clarkson M.R., Chong S., Myers A.C. Civitas: Toward a Secure Voting System. IEEE Symposium on Security and Privacy. Oakland (USA), 2008, pp. 354 – 368.

48. EVOTE 2014. Review on the EVOTE 2014 conference at which POLYAS has been represented by Kai Reinhard [Online]. Available: http://www.micromata.de/en/news/latest-news/news/?tx_ttnews[year]=2014&tx_ttnews[month]=11&tx_ttnews[tt_news]=609&cHash=de8528e29dfb0aca7fe1265134bf0e0a (дата звернення: 06.10.2018).

49. Electronic vote and democracy [Online]. Available: http://www.electronic-vote.org/INTRO/index.php (дата звернення: 05.10.2017).

50. Вишняков В.М., Пригара М.П., Воронін О.В. Відкрита система таємного голосування. *Управління розвитком складних систем*. 2014. №20. С. 110 – 115.

51. Электронное голосование в Эстонии [Online]. Available: https://ru.wikipedia.org/wiki/%D0%AD%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D0%BD%D0%BE%D0%B5_%D0%B3%D0%BE%D0%BB%D0%BE%D1%81%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_%D0%B2_%D0%AD%D1%81%D1%82%D0%BE%D0%BD%D0%B8%D0%B8 (дата звернення: 06.10.2018).

52. Shannon C. Communication Theory of Secrecy Systems. *Bell System Technical Journal*. 1949. 28 (4). Pp. 656–715.

53. E-Voting denied in Switzerland over hacking fears. October 28, 2015 [Online]. Available: http://e-lected.blogspot.com/search?updated-min=2015-01-01T00:00:00-

08:00&updated-max=2016-01-01T00:00:00-08:00&max-results=49 (дата звернення: 08.10.2018).

54. ДСТУ 4145-2002. Інформаційні технології. Криптографічний захист інформації. Цифровий підпис, що ґрунтується на еліптичних кривих. Формування та перевіряння. [Чинний від 2003-07-01] Вид. офіц. Київ, 2003. 22 с.

55. W.Diffie, M.E.Hellman. New Direction in Cryptography. *IEEE Transactions on Information Theory*. 1976. v.IT-22, n.6. Pp. 644-654.

56. НД ТЗІ 1.1-002-99. Загальні положення щодо захисту інформації в комп'ютерних системах від несанкціонованого доступу. [Чинний від 1999-04-28]. Вид. офіц. Київ: ДСТСЗІ СБ України, 1999. 14 с.

57. НД ТЗІ 2.5-004-99. Критерії оцінки захищеності інформації в комп'ютерних системах від несанкціонованого доступу. [Чинний від 1999-04-28]. Вид. офіц. Київ: ДСТСЗІ СБ України, 1999. 53 с.

58. Бройдо В.Л., Ильина О.П. Архитектура ЭВМ и систем: Учебник для вузов. 2-е изд., СПб.: Питер, 2009. 720 с.

59. Гордеев А.В. Операционные системы: Учебник для вузов. 2-е изд., СПб.: Питер, 2004. 416 с.

60. Гмурман В. Е. Теория вероятностей и математическая статистика: Учебное пособие для вузов. 9-е изд., М.: Высшая школа, 2003. 479 с.

61. Вишняков В.М. Захист даних в інформаційних системах**:** Навчальний посібник. К.: КНУБА, 2009. 128 с.

62. Fenollosa C. OpenBSD from a veteran Linux user perspective. June 26, 2015. [Online]. Available: http://cfenollosa.com/blog/openbsd-from-a-veteran-linux-user-perspective.html (дата звернення: 08.09.2018).

63. Конахович Г.Ф., Климчук В.П., Паук С.М., Потапов В.Г. Защита информации в телекоммуникационных системах: Навч. посіб. К. : МК-Пресс, 2005. 279 с.

64. Сушко С.О., Кузнецов Г.В., Фомичова Л.Я., Корабльов А.В. Математичні основи криптоаналізу: Навч. посіб. Д.: Національний гірничий ун- т, 2010. 465 с.

65. Вентцель Е.С., Овчаров Л.А. Теория вероятностей и ее инженерные приложения. 3 изд., перераб. и доп. М.: Издательский центр «Академия», 2003. 464 с.

66. Хачатурова С.М. Электронный учебник по дисциплине "Математические модели системного анализа" [Online]. Available: http://ermak.cs.nstu.ru/mmsa/main/Proba.htm (дата звернення: 09.09.2018).

67. Core Technology Services. North Dakota University System, Grand Forks, ND 58201 [Online]. Available: https://listserv.nodak.edu (дата звернення: 05.09.2018).

68. Discrete logarithm records [Online]. Available: https://en.wikipedia.org/wiki/Discrete_logarithm_records (дата звернення: 09.09.2018).

69. Thorsten Kleinjung, 2014 October 17, "Discrete Logarithms in GF(2^1279)". [Online]. Available: https://listserv.nodak.edu/cgi-bin/wa.exe?A2=NMBRTHRY;256db68e.1410 (дата звернення: 09.09.2018).

70. Antoine Joux, "Discrete logarithms in GF($2^{4080}$)", Mar 22, 2013, [Online]. Available: https://listserv.nodak.edu/cgi-bin/wa.exe?A2=ind1303&L=NMBRTHRY&F=&S=&P=13682 (дата звернення: 09.09.2018).

71. Faruk Gologlu et al., On the Function Field Sieve and the Impact of Higher Splitting Probabilities: Application to Discrete Logarithms in GF($2^{1971}$), 2013. [Online]. Available: http://eprint.iacr.org/2013/074 (дата звернення: 09.09.2018).

72. Antoine Joux, "Discrete logarithms in GF($2^{6168}$) [=GF(($2^{257}$)$^{24}$)]", May 21, 2013. [Online]. Available: https://listserv.nodak.edu/cgi-bin/wa.exe?A2=ind1305&L=NMBRTHRY&F=&S=&P=3034 (дата звернення: 09.09.2018).

73. Erich Wenger and Paul Wolfger, “Harder, Better, Faster, Stronger - Elliptic Curve Discrete Logarithm Computations on FPGAs”. [Online]. Available: http://eprint.iacr.org/2015/143/ (дата звернення: 09.09.2018).

74. Jens Zumbrägel, "Discrete Logarithms in GF(2^9234)", 31 January 2014. [Online]. Available: https://listserv.nodak.edu/cgi-bin/wa.exe?A2=NMBRTHRY;9aa2b043.1401 (дата звернення: 09.09.2017).

75. Programming languages used on the Internet and the World Wide Web (WWW) [Online]. Available: http://www.webdevelopersnotes.com/basics/languages_on_the_internet.php3 (дата звернення: 09.09.2018).

76. Bangladesh exploring e-voting technology for future elections. [Online]. Available: http://e-lected.blogspot.com/search?updated-min=2014-01-01 T0 0:00:00-08:00&updated-max=2015-01-01T00:00:00-08:00& max-results=50 .

77. НД ТЗІ 2.5-010-03 Вимоги до захисту інформації WEB – сторінки від несанкціонованого доступу. [Чинний від 2003-04-15] Вид. офіц. Київ: ДСТСЗІ СБ України, 2003. 53 с.

78. Electronic vote & democracy. [Online]. Available: http://www.electronic-vote.org (дата звернення: 09.09.2018).

79. Каргаполов М. И., Мерзляков Ю. И. Основы теории групп: 3 изд. М. : Изд-во «Наука», 1982. 288 с.

80. Essex A., Clark J., Hengartner U. Cobra: Toward Concurrent Ballot Authorization for Internet Voting. Electronic Voting Technology Workshop/Workshop on Trustworthy Elections (EVT/WOTE'12). Bellevue (USA), 2012, pp. 1 – 13.

81. НД ТЗІ 2.5-005-99 Класифікація автоматизованих систем і стандартні функціональні профілі захищеності оброблюваної інформації від несанкціонованого доступу. [Чинний від 1999-07-01] Вид. офіц. Київ: ДСТСЗІ СБ України, 1999. 53 с..

82. Корченко О.Г., Козлюк І.О., Муратов О.Є. Модель складної орієнтованої інформаційної мережі ЗВДТ. *Захист інформації*. 2011. №2(52). С. 87-94.

83. Козлюк І.О., Конахович Г.Ф., Антонов В.В. Особливості передавання захищеного мовного трафіка через стандартний радіоканал авіаційних систем зв'язку. *Захист інформації*. 2011. №3(52). С. 72-77.

84. Козлюк І.О., Бахтіяров Д.І. Аналіз ефективності комплексного застосування заходів захищеності для підвищення стійкості функціонування засобів керування БПЛА. *Проблеми розвитку глобальної системи зв'язку, навігації та оргалізації повітряного руху CNS/ATM*: зб. тез. доп. науково-технічна конференція, 17-19 листопада 2014 р.: К., 2014. С. 25-26.

85. Смагин А. Г., Ярославский М. И. Пьезоэлектричество кварца и кварцевые резонаторы. М.: «Энергия», 1970. 488 с.

86. Чуприн В.М., Вишняков В.М., Пригара М.П. Генерування випадкових чисел штатними засобами хостів мережі Інтернет. *Захист інформації*. 2016. Том 18, №4. С. 323-335.

87. Чуприн В.М. Метод протидії незаконному впливу на виборців у системі Інтернет голосування./ В.М. Чуприн, В.М.Вишняков, М.П. Пригара // Безпека інформації. – 2017. – Т. 19, №1 – С. 7 – 14.

88. Чуприн В.М. Захист операційного середовища систем Інтернет голосування./ В.М. Чуприн, В.М.Вишняков, М.П. Пригара // Захист інформації. – 2017. – Т. 19, №1 – С. 56-66.

89. Брагина Е. К., Соколов С. С. Современные методы биометрической аутентификации: обзор, анализ и определение перспектив развития. // Вестник АГТУ. — 2016. — № 61 — С. 40 –45.

90. Daugman J.Information Theory and the Iris-Code. IEEE Trans. Info.Foren.Sec 11(2), 2015. – P. 400-409.

91. Тассов К. Л., Дятлов Р. А. Метод идентификации человека по голосу. Инженерный журнал: наука и инновации, 2013, вып. 6.

92. Матвеев Ю. Н . Технологии биометрической идентификации личности по голосу и другим модальностям — ISSN 0236-3933. Вестник МГТУ им. Н. Э. Баумана. Сер. «Приборостроение». 2012.

93. Назарук В.Д. Технології обміну даними дистанційних електронних виборів./ В.Д. Назарук, О.А. Хоменчук// Захист інформації. – 2016. – Т. 18, №4 – С. 10-15.

94. Конахович Г.Ф., Чуприн В.М., Мачалін І.О. Експлуатація телекомунікаційних систем. Підручник для ВУЗів. –К.:Центр учбової літератури, 2016. – 657С.

95. Одноплатный компьютер [Online]. Available: https://ru.wikipedia.org/wiki/%D0%9E%D0%B4%D0%BD%D0%BE%D0%BF%D0%BB%D0%B0%D1%82%D0%BD%D1%8B%D0%B9_%D0%BA%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80.

96. Рид М., Саймон Б. Функциональный анализ.-М: Мир, 1977. – 357 с.

97. Эльсгольц Л.Э. Дифференциальные уравнения. – М: ГИТТЛ, 1957. –271 с.

98. Стратонович Р.Л. Принципы адаптивного приёма. - М: Сов. Радио, 1973. – 141 с.

99. Репин В.Г., Тартаковский Г.П. Статистический синтез при априорной неопределённости и адаптация информационных систем. –М: Сов.Радио, 1977. – 429 с.

100. Летов А.М. Динамика полета и управление.-М.: Наука, 1969. – 359 с.

101. Антонов В.К. Метод построения качественных регуляторов // Кибернетика и вычислительная техника, вып. 126, 2000. – С.40-48.

102. Антонов В.К. Побудова регуляторів із заданою якістю руху за допомогою обмеження зміни функції Ляпунова-Беллмана // Вісник НАУ №4(11), 2001. – С.129 – 132.

103. Валеев К.Г., Финин Г.С. Построение функций Ляпунова. – К.: Наукова думка, 1981. – 412 с.

104. Антонов В.К. Теорема об устойчивости // Теория и методы исследования авиационных автоматических систем и тренажеров. Изд-во КИИГА, 1993. - С. 14-19.

105. Вишняков В.М., Комарницький О.О, Жуковський А.О. Методи контролю керування системою Інтернет голосування. // Управління розвитком складних систем. – 2019. - № 38 – С. 37-44.

106. Чуприн В.М., Вишняков В.М., Комарницький О.О. Метод протидії атакам посередника у транспарентній системі інтернет голосування. // Захист

інформації, Ukrainian Information Security Research Journal. - К.: НАУ, 2019. – Т.20. - №2. – С.172-182.

107. Чуприн В.М., Антонов В.О., Комарницький О.О. Метод розподілу навантаження між серверами системи інтернет-голосування. // Захист інформації Ukrainian Information Security Research Jornal. – К.:НАУ, 2019. – Т. 21. - №1, - С. 25-34.

108. Цюцюра С.В., Комарницький О.О. Застосування новітніх інформаційних технологій в Україні. // Сучасні наукові дослідження та розробки: теоретична цінність та практичні результати: тези Міжнародної науково-практичної конференції (14-19 березня 2016 р., Братислава), 2016, С. 155-156.

109. Комарницький О.О., Нестерук Г.Б. Захист систем дистанційного опитування від атак посередника. // Матеріали 5-ої міжнародної науково-практичної конференції «Management of the development of technologies», Секція "Information technology development of education» ( Kyiv, 30 – 31 March, 2018), Київ, С.76.

110. Ворона П.В., Вишняков В.М., Комарницький О.О., Хлапонін Д.Ю. Принципи побудови транспарентних систем таємного електронного голосування. // Науково-практична конференція до Дня місцевого самоврядування "Форум прямої демократії", 4 грудня 2018 р.: тези доп. – К., 2018. – С.169-171.

111. Комарницький О.О., Хлапонін Д.Ю. Системи таємного електронного голосування як елемент цифрової демократії. // IX Міжнародна науково-практична конференція молодих вчених «Інформаційні технології: економіка, техніка, освіта», 14-15 листопада 2018 р.: тези доп. – К., 2018. – С.117-118.

112. Мачалін І.О., Вишняков В.М., Комарницький О.О. Технологія автентифікації виборців у відкритій системі інтернет-голосування. // Науково-технічний журнал «РАДИОЭЛЕКТРОНИКА И ИНФОРМАТИКА». - № 2(81), апрель – июнь 2018. - С. 55-63.

113. Мачалін І.О., Комарницький О.О., Гнатюк В.О. Удосконалення технології доступу до ресурсів транспарентних систем Інтернет-голосування. // Науковий журнал «Наукоємні технології». - № 4 (40), 2018. - С. 415 – 423.

114. Вышняков В.М., Комарницкий О.А. Транспарентные системы электронной демократии. Accent Graphics Communications & Publishing, Оттава, Канада, 2019, 96 с.

115. Комарницький О.О. Особливості забезпечення безпеки інформації в системах електронної демократії. // Матеріали V науково-практичної конференції «Перспективні напрямки захисту інформації» ОНАЗ ім. О.С.Попова, тези доп. Одеса, 2019. - С. 23 – 25.

116. Бахтіяров, Д.І., Лавриненко, О.Ю., Ліщиновська, Н.О., Комарницький, О.О. Методи оцінювання та прогнозування рівнів електромагнітних випромінювань в урбанізованих середовищах // European Scientific e-Journal. – 2020. – Режим доступу до ресурсу: DOI: 10.47451/inn2020-12-001.

117. Мхамад Ібрагім Ахмад Альомар. Удосконалення технології управління розподілом ресурсів пакетних мереж. Кандидатська дисертаційна робота. –К.: Національний авіаційний університет, 2015. -140 с.

118. Заболотнов Ю.М. Оптимальное управление непрерывными динамическими системами // Изд-во Самарского гос. аэрокосмического университета. – Самара, 2005. -129 с.

119. Башняков О.М., Пічкур В.В. Задача синтезу в теорії керування. –К.: Вид-во «Сталь», 2012. -116 с.

120. Громов Ю.Ю. и др. Специальные разделы теории управления. Оптимальное управление динамическими системами// Изд-во Тамбовского гос. технического университета. – Тамбов, 2007. -168 с.

121. Ким Д.П. Теория автоматического управления. Том 2. Многомерные, нелинейные, оптимальные и адаптивные системы. –М.: Физматлит, 2007. -235 с.

122. Зацепилова Ж.В., Честнов В.Н. Синтез регуляторов многомерных систем заданной точности по среднеквадратичному критерию//Авоматика и телемеханика, №11, 2011.

# ДОДАТОК А

# ВИХІДНІ ТЕКСТИ ПРОГРАМ ДЛЯ ЕКСПЕРИМЕНТАЛЬНОГО ДОСЛІДЖЕННЯ СИСТЕМИ ДИСТАНЦІЙНОГО ТАЄМНОГО ГОЛОСУВАННЯ

## Д1.1. Текст головної програми для управління проведенням експериментального дослідження системи

## (файл index.html)

```
<!DOCTYPE html><html>
<head><meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>  Вибір КНУБА </title>
</head>
<body onload="document.getElementById('submit').disabled=false; return false;">
<table border="1"   bgcolor="#D0F0C0">
  <caption><b>  Вибори  представників  до  органів<br>студентського
самоврядування КНУБА <br>
  Коригування даних виборців</b></caption>
<tr><th>Оберіть виборчу дільницю</th></tr>
<tr><td>
<div>Виборча дільниця
<select size="1" id="answer">
<option value="http://91.198.50.7:11401/BD001401.html">Група ІУСТ-41</option>
<option value="http://91.198.50.7:11402/BD001402.html">Група ІУСТ-42с</option>
<option value="http://91.198.50.7:11403/BD001403.html">Група ІУСТ-31</option>
<option value="http://91.198.50.7:11501/BD001501.html">Група ІТЕП-41</option>
<option value="http://91.198.50.7:11502/BD001502.html">Група ІТЕП-31</option>
<option value="http://91.198.50.7:11503/BD001503.html">Група ІТЕП-32с</option>
<option value="http://91.198.50.7:11601/BD001601.html">Група КСМ-31</option>
<option value="http://91.198.50.7:11602/BD001602.html">Група КСМ-32с</option>
<option value="http://91.198.50.7:11603/BD001603.html">Група КСМ-41</option>
<option value="http://91.198.50.7:11604/BD001604.html">Група КСМ-42с</option>
<option value="http://91.198.50.7:21601/PW001601.html">Пробний разовий
пароль</option>
<option value="http://91.198.50.7:21601/AV001601.html">Пробне
голосування</option>
<option value="http://91.198.50.7:8082/ADMIN.html">Адміністрування</option>
</select>
</div>
 </td></tr>
<br>
  <tr><td align="center">
  <input id="submit" type="button" class="submit" value="Дільницю обрано"
  onclick="this.disabled=true; this.value='Оброляємо запит...'; window.location =
document.getElementById('answer').value; return false;"> </td></tr>
</table><br>
<table border="1" bgcolor="#F0FFE0">
   <caption><b>Виборчі дільниці  </b></caption>
<tr bgcolor="#D0FFC0">
      <td align="center"> Група</td>
      <td align="center"> № дільниці</td>
      <td align="center"> Паспортні дані</td>
</tr>
<tr><td align="center"> ІУСТ-41</td>
      <td align="center">001401</td>
      <td align="center"> IC0401##</td>
</tr>
```

```
<tr><td align="center"> ІУСТ-42с</td>
      <td align="center">001402</td>
      <td align="center"> IC0402##</td>
</tr>
<tr><td align="center"> ІУСТ-31</td>
      <td align="center">001403</td>
      <td align="center"> IC0403##</td>
</tr>
<tr><td align="center"> ІТЕП-41</td>
      <td align="center">001501</td>
      <td align="center"> IT0501##</td>
</tr>
<tr><td align="center"> ІТЕП-31</td>
      <td align="center">001502</td>
      <td align="center"> IT0502##</td>
</tr>
<tr><td align="center"> ІТЕП-32с</td>
      <td align="center">001503</td>
      <td align="center"> IT0503##</td>
</tr>
<tr><td align="center"> КСМ-31</td>
      <td align="center">001601</td>
      <td align="center"> KI0601##</td>
</tr>
<tr><td align="center"> КСМ-32с</td>
      <td align="center">001602</td>
      <td align="center"> KI0602##</td>
</tr>
<tr><td align="center"> КСМ-41</td>
      <td align="center">001603</td>
      <td align="center"> KI0603##</td>
</tr>
<tr><td align="center"> КСМ-42с</td>
      <td align="center">001604</td>
      <td align="center"> KI0604##</td>
</tr>
</table>
 </table>
  <br>
   <table border="1"   bgcolor="#F0FFE0">
   <caption><b>Періоди виборчого процесу </b></caption>
<tr bgcolor="#D0FFC0">
      <td align="center"> Назва періоду </td>
      <td align="center"> Початок</td>
      <td align="center"> Закінчення</td>
</tr>
<tr><td align="center"> Коригування даних</td>
      <td align="center">08.05.2019</td>
      <td align="center"> 18.05.2019</td>
</tr>
<tr><td align="center"> Введення паролів</td>
      <td align="center">18.05.2019</td>
      <td align="center"> 20.05.2019</td>
</tr>
<tr><td align="center"> Голосування 21.05.2019</td>
      <td align="center">8:00</td>
      <td align="center"> 20:00</td>
</tr>
</table>
<br>
<a>Ця система голосування гарантує</a><br>
<a>абсолютну таємницю голосів,  </a><br>
<a>розкриває фальсифікації, </a><br>
<a>позбавляє від впливу підкупом, </a><br>
```

```
<a>моральним чи силовим тиском </a><br>
<a
href="http://www.asdev.com.ua/dndiasb/news/lates_news/e_voting.html">Ознайомлення
з системою </a><br>
<a
href="http://asdev.com.ua/dndiasb/assets/files/Vyshnyakov/SSH.rar">Завантаження
засобів контролю </a>
 </body>
</html>
```

## Д1.2. Текст програми для експериментального дослідження роботи сервера виборчої дільниці

```
///////<><><><><><><><> Начало блока настраиваемых параметров
<><><><><><><><>////
var NVD='001601'; // Номер избирательного участка (ИУ)
var PORT='21601'; // Номер TCP порта
var TCPW=0.15;  // Период ввода одноразовых паролей для голосования (часов)
var TPAUSE=0.01; // Пауза (часов) для печати списка голосующих и загрузки
бюллетней
var TGOL=0.15;   // Период голосования (часов)
var PORTU = 61526; // UDP-порт сервера избирательного участка
var HOSTA = '91.198.50.7';// Нужно проставить адрес сервера аутентификации
//'91.198.50.7'; '192.168.1.106';
var PORTA = 61524; // UDP-порт сервера аутентификации
///////<><><><><><><><> Конец блока настраиваемых параметров
<><><><><><><><>////////
var TRS='';  // Строка для отправки данных на UDP-сервер аутентификации
var TRSP=''; // Строка для упаковки данных для UDP-сервера
var TRSG=''; // Строка для подготовки данных на UDP-сервер
var RTS='';  // Строка для приема данных от UDP-сервера
var RTU='';  // Строка для распаковки данных от UDP-сервера
var KODPZ='AABBCCDDEEFF'; // Код для проверки связи с сервером
var NPERIOD=1; // № периода (1-ввод разовых паролей, 2-пауза, 3-голосование, 4-
завершено)
var NAMEHTML="/PW"+NVD+".html";  // Имя клиентской программы (это для 1-го
периода)
var buffer = require('buffer'); // Подключаем библиотеки node.js
var udp = require('dgram');    // Объявления для UDP клиента
var client = udp.createSocket('udp4');
var OLD_LEN=0;  // Длина последнего принятого UDP пакета
var url= require('url');   // Подключили библиотеку url
var http= require('http'); // Подключили библиотеку http
var fs= require('fs');     // Подключили библиотеку fs
var static= require('node-static');      // Подключили библиотеку node-static
var querystring= require('querystring'); // Подключили библиотеку querystring
var file= new static.Server('.');
var PW='';// Пароль для доступа на сервер (один на все выборы)
var i,j,I,J;  // Переменные для циклов
var KBUL=0; //Количество видов бюллетей
////////////// Оперативные даные избирателей (до 2500 на участке)
//////////////////////////
var PASSVYB = [2500]; // Одноразовые пароли для голосования
 for(var i=0; i<2500; i++) PASSVYB[i]=' '; // Заполняем пробелами
var PASSINP = [2500]; // Зашифрованные пароли избирателей с паспортными данными
 for(var i=0; i<2500; i++) PASSINP[i]=' '; // Заполняем пробелами
var GOLOSF = [2500]; // Признак голосования с фиктивным или верным паролем
 for(var j=0; j<2500; j++) GOLOSF[j]=0; // Заполняем нулями
var GOLOS = [2500]; // Признак голосования с верным паролем (1-проголосовал, 0-
нет)
 for(var j=0; j<2500; j++) GOLOS[j]=0; // Заполняем нулями
```

```
var PASPBIT = [2500];  //  Серия и номер паспорта в битовом виде
////////////////////// Конец оперативных даных избирателей
////////////////////////////////
var GOL=new Array(10);  // Двумерный массив GOL[][] для подсчёта голосов
 for(j=0; j<10; j++) GOL[j] = new Array(100);//GOL[номер бюллетня][номер пункта в
бюллетне]
 for(j=0; j<10; j++) for (i=0;i<100;i++) GOL[j][i]=0; // Заполняем нулями GOL[][]
////////////////// Я Ч Е Й К И  Д Л Я  К Р И П Т О Г Р А Ф И И
////////////////////////////
// Криптография основана на возведении в степень N примитивного элемента X поля
Галуа
var N = [504];// Массив бит показателя степени N для функции STEPX(N)
// Младдший бит в N[1], старший - в N[503]. Нулевые элементы не используем
var A = [505];// Массив бит результата вычисления степени
// В A[] добавлен лишний бит для кратности преобразования пароля в 72 байта
(7*72=504)
var B = [504];// Массив бит для формирования кода проверки для борьбы с атакой
посредника
var CU = [504];   // Массив CU[] для битов шифрования UDP сообщений
var M= new Array(504);// Создаем массив M[][] для степеней примит. элемента
  for(var j=0; j<504; j++) M[j] = new Array(504);
var MA= new Array(504);// Создаем массив MA[][] для степеней A[]
  for(var j=0; j<504; j++) MA[j] = new Array(504);
var M1= [504]; // Массивы для function MULT()
var M2= [504]; // M1, M2 - сомножители  R - результат
var R= [504];
var RN= [2500]; // Создаем двумерный массив RN[2500][504] для хранения случайных
битов
  for(var i=0; i<2500; i++) RN[i] = new Array(504);
var RNB = [2500]; // Создаем двумерный массив RNB[2500][504] для хранения
степеней X^RN
  for(var i=0; i<2500; i++) RNB[i] = new Array(504);
var IRN=-1; // Индекс для максимального элемента RN[IRN][] и RNB[IRN][]
var CRN=-1; // Индекс текущего элемента RN[IRN][] и RNB[IRN][]
var DATA=[20]; // Массив байт для расшифровки пароля
/////////////// Переменные для работы с данными избирателей
//////////////////////
var SFILE='';// Строка для чтения файла данных
var Kstr=0; // К-во строк данных об избирателях (до 2500)
var Lstr=0; // Длина каждой строки, включая первый байт-признак
var Nstr=0; // Номер обрабатываемой строки, начиная с нулевого
var EEE=0;  // Признак наличия избирателя ( 0-нет, 1-есть)
// Переменные в строке данных (первый байт в строке - пробел)
// за пробелом байт OZNAKA
var OZNAKA='0'; // 0-данные валидны, 1-избиратель выбыл, 2-данные заменены,
                // 3- данные ошибочны, 4-отказ избирателя
////////////////////////////////////////////////////////////////////////////////
///
SFILE= fs.readFileSync('VD'+NVD+'.DBT');   // Чтение файла  в строку SFILE
           Lstr= 74;  // Длина строки (включая первый пробел)
           Kstr= SFILE.length/Lstr; /// К-во строк
           Nstr= 0;  // Начальный номер строки
console.log("Kstr=" + Kstr); //-/-/-/-/-/-/-/-/-/-/-/-/-
// Создаём файл для записи строк кода для борьбы с атакой посредника
if (fs.existsSync('CC'+NVD+'.TXT')==true) fs.unlinkSync('CC'+NVD+'.TXT');
else fs.writeFileSync('CC'+NVD+'.TXT','');
////////////////////////////////////////////////////////////////////////////////
////////
var T1= new Date(); // Метка времени для отсчета периодов
var TBG= T1.getTime(); // Момент старта в миллисекундах (от 01.01.1970)
var TEND= TBG+3600000*TCPW; // Момент окончания периода ввода одноразового пароля
////// Резервируем 100 строк (ячеек памяти) для одновременного обслуживания 100
клиентов
var CYF='0123456789'; // Строка цифр для преобразования чисел в строки символов
```

```
var NU2 = [100]; // Массив для преобразования чисел 0-99 в строки "00", "01",
..., "99"
 for(var j=0; j<10; j++){for(var i=0; i<10; i++) NU2[j*10+i]=CYF[j]+CYF[i];} //
Заполнили массив
var NL=0; // Номер строки данных очередного клиента
var ONL=0; // Номер строки клиента, которого начали обслуживать (обмен по
алгоритму DH)
var SNL=0; // Номер строки клиента, которого продолжаем обслуживать
var QNL=0; // Номер строки клиента, обрабатываемой в порядке очереди
var QQ=0;  // К-во запросов в очереди на обслуживание сервером
var FQ=0;  // Флаг обработки очередного запроса из очереди
// В каждую из 100 строк (зарезервированных ячеек памяти) входят: PWG[NL],
PZ[NL], PAU[NL],
// NQ[NL], ICRN[NL], NLA[NL][], NLC[NL][], NLO[NL][], TIO[NL], NstrK[NL],
NLU[NL], QRT[NL]
//
var PWG= [100];         // Пароль для голосования
var PZ= [100];      // Признак занятости строки
var PAU= [100];     // Признак прохождения аутентификации
var FW1=0;              // Флаг необходимости ожидания на этапе аутентификации
for(i=0; i<100; i++) PZ[i]=PAU[i]=0; // Заполнили нулями
var NQ= [100];      // Номер ожидаемого запроса клиента
var ICRN= [100];    // Индекс используемых элементов RN[ICRN[NL]][] и
RNB[ICRN[NL]][]
for(var i=0; i<100; i++) NQ[i]=0;
var NLA = new Array(100);     // Создаем двумерный массив NLA[NL][] для
полученных битов
  for(var i=0; i<100; i++) NLA[i] = new Array(504);
var NLC = new Array(100);     // Создаем двумерный массив NLC[NL][] для битов
шифрования
  for(var i=0; i<100; i++) NLC[i] = new Array(504);
var NLO = new Array(100);     // Создаем двумерный массив NLO[NL][] для
расшифрованного
  for(var i=0; i<100; i++) NLO[i] = new Array(504);
var NLU = new Array(100);// Создаем двумерный массив NLU[NL][] для UDP
  for(var i=0; i<100; i++) NLU[i] = new Array(504);
 var TIO=[100];   // Таймауты
 var NstrK=[100]; // Номер Nstr строки в файле данных клиента
 INIT_M(); // Инициализация массива M[][] + вычисление RN[0-99][] и RNB[0-99][]
var RT='/';  // Строка для приема данных от клиента
var QRT= [100];   // Массив строк для занесения принятых данных от клиента в
очередь
var TR='N';  // Строка для отправки данных клиенту
var QTR= [100];   // Массив строк для отправки данных клиенту через очередь
for(i=0; i<100; i++) QRT[i]=QTR[i]=0; // Заполнили нулями
////////////////////////////////////////////////////////////////////////////
var timeOut1=setInterval(function()
{ // Каждую секунду будем опрашивать очередь
 if (QQ>0 && FQ==0) QQo();
},1000);
////////////////////////////////////////////////////////////////////////////
var timeOut=setInterval(function()
{ // Каждые 10 секунд будем проверять время завершения выделенных интервалов
времени
var T1 = new Date(); // Берем метку времени
var TM = T1.getTime(); // Момент проверки в миллисекундах от 01.01.1970
if (NPERIOD<4 && TEND<TM) ENDPER(); // если период закончился
else {for(var i=0; i<100; i++) if (PZ[i]==1) if (TM>TIO[i]) CLSTRD(i);}
// Иначе, если время вышло, то освобождаем строку для нового клиента

},10000);
////////////////////// U D P -client
////////////////////////////////////////////////
/*      ПОКА ОТКЛЮЧЕН
```

```
client.on('message',function(message, remote)
{ // console.log('From '+remote.address + ':' + remote.port +' - ' + message);
//-/-/-/-/
 RTS=''; // Строка для приема данных от UDP сервера аутентификации
      RTS=RTS+message;
      SNL=0; S2=RTS.substr(0,2);
      // Подготовили SNL для поиска номера строки данных
      while (NU2[SNL]!=S2) SNL++; // Ищем номера строки  клиента
      if (RTS.length==128 && OLD_LEN!=128 )
      {
      OLD_LEN=128;
 UNPACK_(RTS); // Распаковываем данные из RTS в RTU для вычисления ключа
 COUNTU(SNL); //////// Вычисляем ключ и заносим в CU[] //////////////
// Цикл шифрования номера паспорта из NLO[SNL][i] ключом CU[i] в строку TRS
      TRS=NU2[SNL]; // Занесли начальные 2 байта (номер строки клиента)
      for (var i=1;i<=32;i++) {CU[i]=NLO[SNL][i]+CU[i];
      if (CU[i]==1) TRS=TRS+'1'; else TRS=TRS+'0';} FLAG_UDP=1;
// Отправляем зашифрованные паспортные данные через Buffer.from(TRS)
var data = Buffer.from(TRS);
client.send(data,PORTA,HOSTA,function(error){ if(error) throw error;});
      }
      if (RTS.length==34 && OLD_LEN!=34)
      {
      for (var i=1;i<=32;i++)
      {var j=0; if (RTS[i+1]=='1') j=1; CU[i]=j+CU[32+i]; if (CU[i]>1) CU[i]=0;

      }// Расшифровку занесли в первые 32 элемента CU[]
var PASP_S='';    // Отправленные данные паспорта (в битах)
for (var i=1;i<=32;i++) {if (NLO[SNL][i]==1) PASP_S=PASP_S+'1'; else
PASP_S=PASP_S+'0';}
var PASP_R='';    // Принятые данные паспорта (в битах)
for (var i=1;i<=32;i++) {if (CU[i]==1) PASP_R=PASP_R+'1'; else
PASP_R=PASP_R+'0';}
console.log('PASP_S='+PASP_S);//-/-/-/-/-/-/-/-/-/-/-/-/-//--/--/-///-/-/
console.log('PASP_R='+PASP_R);//-/-/-/-/-/-/-/-/-/-/-/-/-//--/--/-///-/-/
if (PASP_S==PASP_R)
{
var T1 = new Date(); // Метка времени для шифрования и установки таймаута
var TB = T1.getTime(); // Момент начала преобразований
TIO[SNL]=TB+90000;     // Установили таймаут для ввода пароля (90 sec)
NQ[SNL]=2; // Номер ожидаемого запроса клиента
PAU[SNL]=1; // Признак прохождения аутентификации
NstrK[SNL]=Nstr; // Номер Nstr строки в файле данных облуживаемого клиента
} OLD_LEN=FW1=0;
      }
});
*////////////////////////////////////////////////////////////////////////////
//////////
http.createServer (function (req, res) // req-запрос, res-ответ
{
var pathname= url.parse(req.url).pathname;
var T1= new Date(); // Метка времени -/-/-/-/-/-/-/-/-/-/
var T= T1.getTime(); //-/-/-/-/-/-/-//--/-/-/--/-/-
     console.log("Request for " + pathname + " received. T="+T); //-/-/-/-/-/-/-
/-/-/-/-/-/-/
var ADDR=req.connection.remoteAddress+':'+req.connection.remotePort;
      console.log("ADDR=" + ADDR); //-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-
/-/-/-/-/
if (pathname==NAMEHTML) file.serve(req, res);
else {RT=pathname;
      if (RT[1]=='Q')
      {
if (NPERIOD==1) /////// Начало работы в первом периоде (ввод одноразовых паролей)
///////
```

```
{
switch (RT[2])
   {
case '0': // Отправим новому клиенту номер строки и значение B[NL][]
      if (RT.length !=506) break; // Проверка валидности запроса
var NL=0; while (PZ[NL]==1 && NL<100) NL++; // Поиск пустой строки     для
клиента
      if (NL==100) TR='W'; // Если все 100 строк заняты, то отправим 'W'
(ожидайте)
      else
      {
      PZ[NL]=1; // Заняли строку для клиента
      ICRN[NL]=CRN; CRN++; // Присвоили текущий индекс CRN для строки NL
      NQ[NL]=1; // Установили номер следующего запроса клиента
      for (var i=1;i<=503;i++) NLA[NL][i]=RT[i+2]; // Занесли принятые 503 бита в
NLA[NL]
      var T1= new Date(); // Берем метку времени для установки таймаута
      var TB= T1.getTime(); // Момент начала преобразований в миллисекундах
      TIO[NL]= TB+1200000;     // Установили таймаут для обслуживания (20 минут)
      TR='N'+NU2[NL];  // Начали формировать строку для отправки данных клиенту
            for (var i=1;i<=503;i++) // Цикл переноса значения RNB[ICRN[NL]][] в
TR
            {if (RNB[ICRN[NL]][i]==1) TR=TR+'1'; else TR=TR+'0';}
      } // Производим обмен ключами по алгоритму Диффи-Хеллмана
var T1 = new Date(); // Контрольная метка времени -/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/
var T= T1.getTime();
console.log("Q0 TR="+TR+' T='+T);//////////////////////////////////////-/-/-/-/-/-
/-/-/-/-/-/-/-/-/-/
      res.end(TR);            // Отправили данные клиенту
break;
case '1': // Будем принимать и ставить в очередь запрос клиента
      if (RT.length !=262) break; // Проверка валидности запроса
      for (var i=6;i<262;i++) if (RT[i]>"1" || RT[i]<'0') break;
var S2='' // Строка поиска номера строки в массиве NU2
      S2=RT[4]+RT[5]; // Занесли номер строки клиента в S2
var ONL=Number(S2); // В ONL занесли номер строки с данными обратившегося клиента

      if (NQ[ONL]!=1 ) break;
      NQ[ONL]=2;
      if (PZ[ONL]!=1 )  //  отправляем код сообщения "Время истекло"
      TR='E4';
      else
      {      // Постановка запроса клиента в очередь на выполнение
      QRT[ONL]=RT; QQ++; // Занесли принятый запрос в очередь
      TR='S'+QQ.toFixed(0); // Сообщение о том, что запрос занесен в очередь
      }
var T1 = new Date(); // Берем метку времени -/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/
var T= T1.getTime();
      res.end(TR);      // Отправили данные клиенту
break;
case '2': // Будем проверять очередь
      if (RT.length != 6) break;    // Проверка валидности запроса
var S2=''; // Строка поиска номера строки в массиве NU2
            S2=RT[4]+RT[5]; // Занесли номер строки обратившегося клиента в S2

var SNL=Number(S2); // В SNL занесли номер строки с данными обратившегося клиента
            if (PZ[SNL]!=1 ) TR='E4';  // отправляем код "Время истекло"
            else  // Опрос результата выполнения запроса из очереди
            {
            if (QRT[SNL]!=0 && QTR[SNL]==0) TR='W'; // Ожидайте
            else
            {// Отправляем сообщение из очереди
                  if (QRT[SNL]!=0 && QTR[SNL]!=0)
                  {
```

```
                  TR=QTR[SNL];  QQ--; if (QQ<0) QQ=0; QRT[SNL]=0;
                  FKODPZ(SNL); // Формирование и запись кода для борьбы с атакой
посредника
                  }
                  else break;
            }
            }
var T1 = new Date(); // Контрольная метка времени -/-/-/-/-/-/-/-/-/-/-/-/-/-/-/
var T= T1.getTime(); //-/-/-/-/-/-/-/-//--/-/-/--/-/-
res.end(TR);            // Отправили ответ клиенту
break;
   } // switch (RT[2])
} //>>>>>>> Конец обработки периода ввода одноразовых паролей (NPERIOD==1)
<<<<<<<<
if (NPERIOD==2 || NPERIOD==4)
      {  TR='E4';       // Устанавливаем код "Время истекло"
      res.end(TR);      // и отправляем код ошибки
      }
////////////////// Н А Ч А Л О   Г О Л О С О В А Н И Я
//////////////////////////
if (NPERIOD==3)
  {
switch (RT[2])
{
case '0': // Отправим клиенту номер строки и значение B[NL][]
      if (RT.length !=506) break; // Проверка валидности запроса
var NL=0; while (PZ[NL]==1 && NL<100) NL++; // Поиск пустой строки     для
клиента
      if (NL==100) TR='W'; // Если все 100 строк заняты, то отправим 'W'
(ожидайте)
      else
      {
      PZ[NL]=1; // Заняли строку для клиента
      ICRN[NL]=CRN; CRN++; // Присвоили текущий индекс CRN для строки NL
      NQ[NL]=1; // Установили номер следующего запроса клиента
      for (var i=1;i<=503;i++) NLA[NL][i]=RT[i+2]; // Занесли принятые 503 бита в
NLA[NL]
      var T1 = new Date(); // Метка времени для установки таймаута
      var TB = T1.getTime(); // Момент начала преобразований
      TIO[NL]=TB+1200000;     // Установили таймаут для обслуживания (20 минут)
      TR='N'+NU2[NL];  // Начали формировать строку для отправки данных клиенту
            for (var i=1;i<=503;i++) // Цикл переноса значения RNB[ICRN[NL]][] в
TR
            {if (RNB[ICRN[NL]][i]==0) TR=TR+'0'; else TR=TR+'1';}
      } // Производим обмен ключами по алгоритму Диффи-Хеллмана
var T1 = new Date(); // Контрольная метка времени -/-/-/-/-/-/-/-/-/-/-/-/-/-/-/
var T= T1.getTime(); // в миллисекундах -/-/-/-/-/-/-/-/-/-//-/-//-
      res.end(TR);            // Отправляем данные клиенту
break;
case '1': // Будем принимать и ставить в очередь запрос клиента
      if (RT.length < 158) break; // Проверка валидности запроса
      for (var i=6;i<158;i++) if (RT[i]>"1" || RT[i]<'0') break;
var S2='' // Строка поиска номера строки в массиве NU2
      S2=RT[4]+RT[5]; // Занесли номер строки клиента в S2
var ONL=Number(S2); // В ONL занесли номер строки с данными обратившегося клиента
      if (NQ[ONL]!=1 ) break;
      NQ[ONL]=2;
      if (PZ[ONL]!=1 )  //  отправляем код сообщения "Время истекло"
      TR='E4';
      else
      {      // Постановка запроса клиента в очередь на выполнение
      QRT[ONL]=RT; QQ++; // Занесли принятый запрос в очередь
      TR='S'+QQ.toFixed(0); // Сообщение о том, что запрос занесен в очередь
      }
```

```
      res.end(TR); // Отправили ответ клиенту
break;
case '2': // Будем проверять очередь
      if (RT.length != 6) break; // Проверка валидности запроса
var S2=''; // Строка поиска номера строки в массиве NU2
            S2=RT[4]+RT[5]; // Занесли номер строки клиента в S2
var SNL=Number(S2); // В SNL занесли номер строки с данными обратившегося клиента
            if (PZ[SNL]!=1 ) TR='E4'; //  отправляем код сообщения "Время
истекло"
            else  // Опрос результата выполнения запроса из очереди
            {
            if (QRT[SNL]!=0 && QTR[SNL]==0) TR='W'; // Ожидайте
            else
            {// Отправляем сообщение из очереди
                  if (QRT[SNL]!=0 && QTR[SNL]!=0)
                  {
                  TR=QTR[SNL];  QQ--; if (QQ<0) QQ=0; QRT[SNL]=0;
                  FKODPZ(SNL); // Формирование и запись кода для борьбы с атакой
посредника
                  }
            }
            }
      res.end(TR);            // Отправили ответ клиенту
break;
   }  // switch (RT[2])
  }  // Анализируем случай NPERIOD==3
 } // Отсутствует символ "Q" в начале запроса
 } // Анализируем RT=pathname в случае, когда это не вызов .html - файла
}).listen(PORT);
console.log('Server listen port '+PORT);
////////////////////////////// Ф У Н К Ц И И ////////////////////////////
function INIT_M() // Инициализация M[][] + вычисление 0-99 элементов RN[][],
RNB[][]
{ var T1 = new Date();   // Метка времени
  var TB = T1.getTime(); // Момент в милисекундах (от 01.01.1970)
  var TN=TB;      // Будем определять чётность к-ва миллисекунд
  var TD = TN%2;  // и в зависимости от четности заполнять случайный бит RN[0][1]
  IRN=0;  if (TD>0) RN[IRN][1]=1; else  RN[IRN][1]=0;
// В первую строку массива M[1][] заносим примитивный элемент поля Галуа (0100
... 0000),
// а каждая следующая строка массива M[][] будет равна квадрату предыдущей строки
    for (var j=1;j<=503;j++) M[1][j]= 0; // Обнулили M[1][]
    M[1][2]=1; // Занесли 1 во второй бит и получили примитивный элемент поля
    for (var I=2;I<=503;I++) // Начало цикла заполнения M[][] степенями элемента
    {for (var J=1;J<=503;J++)  M1[J]=M2[J]=M[I-1][J]; // Сомножители равны
предыдущей строке
    MULT(); // Умножение элементов поля Галуа GF(2^503) по правилу полиномов
      for (var j=1;j<=503;j++) M[I][j]=R[j]; // Перенесли результат в M[I][j]
      var T1 = new Date();      // Вычисляем очередной случайный бит
      var TN = T1.getTime();  // из четности к-ва миллисекунд таймера
      var TD = TN%2; //  это признак четности
      if (TD>0) RN[IRN][I]=1; else RN[IRN][I]=0; // Учитываем четность
      var NN = Math.random();                    // и стандартное случайное число
      if (NN>0.5) RN[IRN][I]=RN[IRN][I]+1; if (RN[IRN][I]>1) RN[IRN][I]=0; //
Заполнили бит
    } // Конец цикла заполнения массива M[][] константами (степенями примитивного
элемента)
  for (var i=1;i<=503;i++) N[i]=RN[IRN][i]; // Перенесли случайные биты в массив
N[] для STEPX()

      STEPX(N); // Вычисление случайной степени элемента X в строке IRN=0
// Вычислили A[]=X[]^N[] (X[]- примитивный элемент, N[]- случайные биты
for (var I=1;I<=503;I++) RNB[IRN][I]=A[I]; //Результат в RNB[IRN][]
       // Строка с номером IRN=0 в массивах RN[][] и RNB[][] подготовлена
```

```
for (IRN=1;IRN<=9;IRN++) // Заполняем в цикле ещё 99 строк этих массивов
{
      for (var I=1;I<=503;I++) // Начинаем цикл заполнения массива RN[IRN]
случайными битами
      {
      var T1 = new Date();     // Вычисляем очередной случайный бит
      var TN = T1.getTime();  // из чётности к-ва миллисекунд таймера
      var TD = TN%2; //  это признак четности
      if (TD>0) RN[IRN][I]=1; else RN[IRN][I]=0;
      var NN = Math.random(); // бит зависит от четности и стандартного
случайного числа
      if (NN>0.5) RN[IRN][I]=RN[IRN][I]+1; if (RN[IRN][I]>1) RN[IRN][I]=0; //
Заполнили бит
      N[I]=RN[IRN][I]; // Перенесли случайный бит в N[] для STEPX()
      }
      STEPX(N); // Вычислили A[]=X[]^N[], где X[]- примитивный элемент поля Галуа
      for (var i=1;i<=503;i++) RNB[IRN][i]=A[i]; //Результат из A[] в RNB[IRN][]

} // Конец цикла первоначального заполнения массивов RN[][] и RNB[][]
 IRN=9; // Индекс максимального готового элемента RN[IRN][] и RNB[IRN][]
 CRN=0;  // Индекс текущего элемента RN[IRN][] и RNB[IRN][]
} // End of function INIT_M()
////////////////////////////////////////////////////////////////////////////////
///////////
function POWA(ONL) // Вычисление степени A[] через MA[][] + вычисление RN[IRN][]
и RNB[IRN][]
{//---- Вычисление 503 битов для шифрования сообщений
      IRN++; // Будем заполнять очередную пару элементов RN[IRN][] и RNB[IRN][]
      var T1 = new Date(); // Метка времени для нахождения чётности
      var TD = T1.getTime()%2; // четность таймера занесли в TD
      if (TD>0) RN[IRN][1]=1; else  RN[IRN][1]=0;// В зависимости от четности
заполнили бит

    for (var j=1;j<=503;j++) MA[1][j]= A[j]; // занесли значение A[] в первую
строку массива
      for (var I=2;I<=503;I++) // Заполняем MA[][] степенями A[],а RN[IRN][]
      {                        // случайными битами
      for (var J=1; J<=503; J++) M1[J]= M2[J]= MA[I-1][J]; // M1 и M2 равны пред.
строке
      MULT(); // Каждая строка MA[][] равна квадрату предыдущей строки
      for (var i=1;i<=503;i++) MA[I][i]=R[i]; // результат занесли в MA[I][]
            var T1= new Date(); // Метка времени для получения случайного бита
            var TD= T1.getTime()%2; // Находим четность к-во миллисекунд
            if (TD>0) RN[IRN][I]=1; else RN[IRN][I]=0;// Случайный бит зависит от
чётности
            var NN= Math.random();  // и стандартного случайного числа
            if (NN>0.5) RN[IRN][I]=RN[IRN][I]+1; if (RN[IRN][I]>1)
RN[IRN][I]=0;// Есть бит
            } // Конец заполнения массива MA[][] степенями A[] и массива
RN[IRN][]
            for (var j=1; j<=503; j++) A[j]=0; A[1]=1; // Занесли 1 в A[]
      for (var J=1;J<=503;J++) // Находим A в степени RN
        if (RN[ICRN[ONL]][J]==1)
        {for (var I=1;I<=503;I++){M1[I]= MA[J][I]; M2[I]=A[I];}
         MULT();
         for (var I=1;I<=503;I++) A[I]=R[I];
        } // --- В A[] получена случайная послеовательность для шифрования
сообщений
            for (var i=1;i<=503;i++) {N[i]=NLU[ONL][i]=RN[ICRN[ONL]][i];
NLC[ONL][i]=A[i];}
            // Перенесли A[] в NLC[ONL][], а RN[ICRN[ONL]] в N[] для STEPX(N) и в
NLU[]
            STEPX(N); // Вычисление случайной степени элемента X для  RNB[IRN][]
            for (var I=1;I<=503;I++) RNB[IRN][I]=R[I];
```

```
            // Вычислены RN[IRN][] и RNB[IRN][]
} // End of function POWA(ONL)
////////////////////////////////////////////////////////////////////////////////
/////////
function POWA0(ONL) // Вычисление степени A[] через MA[][]
{//---- Вычисление 503 битов для шифрования сообщений
    for (var i=1;i<=503;i++) MA[1][i]= A[i]; // Занесли значение A[] в строку 1
      for (var I=2;I<=503;I++) // Заполняем массив MA[][] степенями A[] в цикле
      {
      for (var J=1;J<=503;J++) M1[J]= M2[J]= MA[I-1][J]; // M1 и M2 равны
предыдущей строке
      MULT(); // Каждая строка массива MA[][] равна квадрату пред. строки
      for (var j=1;j<=503;j++) MA[I][j]=R[j]; // Результат занесли в MA[I][]
      } // Конец заполнения массива MA[][] степенями A[]
      for (var i=1;i<=503;i++) A[i]=0; A[1]=1; // 1 -> A[]
    for (var J=1;J<=503;J++) // Находим A в степени RN[ICRN[ONL]][]
       if (RN[ICRN[ONL]][J]==1)
       {for (var I=1;I<=503;I++){M1[I]= MA[J][I]; M2[I]=A[I];}
        MULT();
        for (var I=1;I<=503;I++) A[I]=R[I];
       } // -- В A[] получена послеовательность для шифрования сообщений
      for (var i=1;i<=503;i++) NLC[ONL][i]=A[i]; // Перенесли A[] в NLC[ONL][]

} // End of function POWA0(ONL)
////////////////////////////////////////////////////////////////////////////////
/////
function CLSTRD(CNL)     // Очистка строки данных клиента
{
      PZ[CNL]=PAU[CNL]=0; // Признаки занятости строки и прохождения
аутентификации
      QRT[CNL]=QTR[CNL]=0; // Заполнили нулями данные очереди
      NQ[CNL]=0; // Номер ожидаемого запроса клиента
} // End of function CLSTRD(CNL)
////////////////////////////////////////////////////////////////////////////////
//////
function ENDPER()     // Завершение периода
{
console.log('ENDPER: '+NPERIOD); ////////////////////////////////-/-/-/-/-/-/-/

switch (NPERIOD)
{
case 1: NPERIOD=2; // Завершён период ввода одноразовых паролей
      var T1 = new Date(); // Берем метку времени для расчета конца очередного
периода
      var TM = T1.getTime(); // текущий момент в миллисекундах от 01.01.1970
      TEND=TM+3600000*TPAUSE; // Момент завершения паузы (периода 2)
      var REGISTR=''; // Готовим строку для регистрации (ввод пароля)
      for (var Nstr=0; Nstr<Kstr; Nstr++)
      {
      if (PASSVYB[Nstr]==' ') REGISTR=REGISTR+'0';
      else REGISTR=REGISTR+'1';
      }
      fs.writeFileSync('RE'+NVD+'.TXT', REGISTR); // Вывод данных о регистрации
      NL=0; // Заполняем нулями признак занятости строки и номер ожидаемого
запроса
      for(var i=0; i<100; i++) PZ[i]=NQ[i]=0;
      for(var i=0; i<100; i++) QRT[i]=QTR[i]=0; // Заполнили нулями данные
очереди
      QQ=0; QNL=0;
break;
case 2: NPERIOD=3;  // Голосование начинается
      CRN=0; // Устанавливаем начальное значение индекса RN[CRN][] и RNB[CRN][]
      var T1 = new Date(); // Метка времени для расчета конца очередного периода
      var TM = T1.getTime(); // текущий момент в миллисекундах от 01.01.1970
```

```
      TEND=TM+3600000*TGOL;   // Установили момент окончания голосования
      NAMEHTML="/AV"+NVD+".html";  // Название программы для 3-го периода
(голосование)
break;
case 3: NPERIOD=4;  // Голосование окончено
      clearInterval(timeOut); clearInterval(timeOut1);
      var RESULT=''; // Строка для формирования результата голосования (К-во
голосов)
      var GOLOSA=''; // Строка для признака участия в голосовании (Наличие
голоса)
      var S1; // Ячейка для вывода номера бюллетня
for (var b=0; b<KBUL; b++) // Вывод результатов по бюллетням в цикле
      {S1=CYF[b+1]; RESULT=RESULT+'Bulletin #'+S1+'\r\n';
       for (var i=0;i<100;i++)
       if (GOL[b][i]!=0)
       RESULT=RESULT+'#'+i+': '+GOL[b][i]+'\r\n';
      }

      fs.writeFileSync('KG'+NVD+'.TXT', RESULT); // Вывод кол-ва голосов
      for (var Nstr=0; Nstr<Kstr; Nstr++) // Цикл по строкам
      if (GOLOSF[Nstr]==0) // Определяем строки тех, кто не голосовал
      GOLOSA=GOLOSA+'0';
      else GOLOSA=GOLOSA+'1'; // Выводим 1 для тех, кто проголосовал
      fs.writeFileSync('NG'+NVD+'.TXT', GOLOSA); // Вывели данные о наличии
голоса
      break;
}
} // End of function ENDPER()
////////////////////////////////////////////////////////////////////////
function FKODPZ(SNL) // Формирование и запись кодовой строки против атак
посредника
{
if (PZ[SNL]==0) return;
for (var i=1; i<503; i++) B[i]=RNB[ICRN[SNL]][i];
KODPZ='';
for (var i=0; i<24; i++) // Цикл формирования 24 байт кода
{ k=0;
if (B[4*i+1]=='1') k++;
if (B[4*i+2]=='1') k=k+2;
if (B[4*i+3]=='1') k=k+4;
if (B[4*i+4]=='1') k=k+8;
switch (k)
            {
            case 0: KODPZ=KODPZ+'0'; break;
            case 1: KODPZ=KODPZ+'1'; break;
            case 2: KODPZ=KODPZ+'2'; break;
            case 3: KODPZ=KODPZ+'3'; break;
            case 4: KODPZ=KODPZ+'4'; break;
            case 5: KODPZ=KODPZ+'5'; break;
            case 6: KODPZ=KODPZ+'6'; break;
            case 7: KODPZ=KODPZ+'7'; break;
            case 8: KODPZ=KODPZ+'8'; break;
            case 9: KODPZ=KODPZ+'9'; break;
            case 10: KODPZ=KODPZ+'A'; break;
            case 11: KODPZ=KODPZ+'B'; break;
            case 12: KODPZ=KODPZ+'C'; break;
            case 13: KODPZ=KODPZ+'D'; break;
            case 14: KODPZ=KODPZ+'E'; break;
            case 15: KODPZ=KODPZ+'F'; break;
            }
}
            var T1 = new Date(); // Метка времени
            var S=''; // Ячейка для формирования строки с кодом
            S=S+NU2[T1.getDate()]+'.';
```

```
            S=S+NU2[T1.getMonth()+1]+'.';
            S=S+'20'+NU2[T1.getFullYear()-2000]+'  ';
            S=S+NU2[T1.getHours()]+':';
            S=S+NU2[T1.getMinutes()]+':';
            S=S+NU2[T1.getSeconds()]+'  '+KODPZ+'\r\n';
// Сформировали строку по формату ЧЧ.ММ.ГГГГ  ЧЧ:ММ:СС  AABBCCDDEEFF
            fs.appendFileSync('CC'+NVD+'.TXT',S); // Дописали строку в файл
} // End of function FKODPZ(SNL)
////////////////////////////////////////////////////////////////////
function MULT()     // Умножение элементов поля Галуа GF(2^503) по правилу
полиномов
{  // Младдший бит в [1], старший - в [503]. Нулевые элементы не используем
 var i,j,r,r1,r2,r3; // Подготовили переменные
    for (i=1;i<=503;i++) R[i]=0; // Заполнили нулями результат умножения
    for (i=1;i<=503;i++) // Умножение по правилам полиномов в двойном цикле
    if (M1[i]==1) // Выбираем единичные элементы M1[], т.к. нулевые дают нули
    {for (j=1;j<=503;j++)
     if (M2[j]==1) // Выбираем единичные элементы M2[]
        {r=i+j-1;
            if (r>503) // Если номер за пределами массива,
            { r=r-503; // то вычитаем 503
             if (r>=501) // Если после этого результат >=501,
                {r=r-501; // то вычитаем 501
                r1=1+r; r2=4+r; r3=501+r; // Нашли r1,r2,r3 для случая большого r
            // К этим элементам нужно добавить 1 по модулю 2
                if (R[r3]==0) R[r3]=1; else R[r3]=0; // Добавили 1 к R[r3]
                }
            else {r1=r; r2=r+3;} // Случай, когда r<=503
            if (R[r1]==0) R[r1]= 1; else R[r1]= 0; // Добавили 1 к R[r1]
            if (R[r2]==0) R[r2]= 1; else R[r2]= 0; // Добавили 1 к R[r2]
            } // Иначе добавим 1 к R[r]
            else {if (R[r]==0) R[r]= 1; else R[r]= 0;}
        }
    }
} // End of MULT()
////////////////////////////////////////////////////////////////////////////
///////
function STEPX(N)  // Нахождение степени примитивного элемента поля Галуа
{ // A[]=X[]^N[]  X- примитивный элемент, N- показатель степени
var i,I,J;
    for (i=2; i<=503; i++) A[i]=0; A[1]=1; // 1 -> A[]
    for (J=1;J<=503;J++) // Цикл возведения в степень
        if (N[J]==1)
        {for (I=1;I<=503;I++){M1[I]= M[J][I]; M2[I]=A[I];}
         MULT();
         for (I=1;I<=503;I++) A[I]=R[I];
        } // Конец цикла возведения в степень.
} // End of STEPX()
////////////////////////////////////////////////////////////////////////////
/////
function COUNTU(SNL) // Вычисление последовательности битов для шифрования UDP
сообщений
{ var i,I,J;
for (i=1;i<=503;i++) {if (RTU[i-1]=='1') A[i]=1; else A[i]=0;}
// Занесли, принятые от клиента 503 бита в A
for (i=1;i<=503;i++) MA[1][i] = A[i]; // Занесли A[] в MA[1][]
for (I=2;I<=503;I++) // Заполняем массив MA[][] степенями A[]
      {for (J=1; J<=503; J++) M1[J]= M2[J]= MA[I-1][J]; // сомножители равны
пред. строке
    MULT(); // Каждая строка массива равняется квадрату пред. строки
      for (j=1;j<=503;j++) MA[I][j]=R[j]; // Результат умножения  -> MA[I][]
      } // Заполнили массив MA[][] степенями A[]
for (i=2;i<=503;i++) CU[i]=0; CU[1]=1; // Занесли 1 в CU[]
for (J=1;J<=503;J++) // Цикл возведения A в степень NLU[SNL][]
```

```
    if (NLU[SNL][J]==1)
    {for (I=1;I<=503;I++){M1[I]= MA[J][I]; M2[I]=CU[I];}
      MULT();
            for (I=1;I<=503;I++) CU[I]=R[I];
    }
//--- В CU[] получена случайная последовательность для шифрования UDP сообщений

} // End of function COUNTU(SNL)
////////////////////////////////////////////////////////////////////////////
function PACK_(TRSG)     // Упаковка 4 битов в 16-ичный символ (0-F)
{
TRSG=TRSG+'0'; // Строку из 503 символов "0", "1" дополнили символом "0" для
кратности 4
TRSP='';
for (var i=0; i<126; i++) // Цикл формирования 16-ичных символов
{ k=0;
if (TRSG[4*i+0]=='1') k++;
if (TRSG[4*i+1]=='1') k=k+2;
if (TRSG[4*i+2]=='1') k=k+4;
if (TRSG[4*i+3]=='1') k=k+8;
switch (k)
            {
            case 0: TRSP=TRSP+'0'; break;
            case 1: TRSP=TRSP+'1'; break;
            case 2: TRSP=TRSP+'2'; break;
            case 3: TRSP=TRSP+'3'; break;
            case 4: TRSP=TRSP+'4'; break;
            case 5: TRSP=TRSP+'5'; break;
            case 6: TRSP=TRSP+'6'; break;
            case 7: TRSP=TRSP+'7'; break;
            case 8: TRSP=TRSP+'8'; break;
            case 9: TRSP=TRSP+'9'; break;
            case 10: TRSP=TRSP+'A'; break;
            case 11: TRSP=TRSP+'B'; break;
            case 12: TRSP=TRSP+'C'; break;
            case 13: TRSP=TRSP+'D'; break;
            case 14: TRSP=TRSP+'E'; break;
            case 15: TRSP=TRSP+'F'; break;
            }
}
} // End of function PACK_()
////////////////////////////////////////////////////////////////////////////
function UNPACK_(RTS) // Распаковка данных из RTS в RTU для вычисления ключа
{ var i,U='';
      for (i=2; i<128; i++) // Цикл распаковки 16-ичных символов в двоичные
      switch (RTS[i])
            {
            case '0': U=U+'0000'; break;
            case '1': U=U+'1000'; break;
            case '2': U=U+'0100'; break;
            case '3': U=U+'1100'; break;
            case '4': U=U+'0010'; break;
            case '5': U=U+'1010'; break;
            case '6': U=U+'0110'; break;
            case '7': U=U+'1110'; break;
            case '8': U=U+'0001'; break;
            case '9': U=U+'1001'; break;
            case 'A': U=U+'0101'; break;
            case 'B': U=U+'1101'; break;
            case 'C': U=U+'0011'; break;
            case 'D': U=U+'1011'; break;
            case 'E': U=U+'0111'; break;
            case 'F': U=U+'1111'; break;
            }
```

```
            RTU='';
            for (i=0; i<503; i++) RTU=RTU+U[i];
} // End of function UNPACK_()
////////////////////////////////////////////////////////////////////
function QQo() // Обслуживание очереди запросов
{ // Ищем строку от номера QNL до 99, а затем QNL=0 и QNL++
FQ=1;
console.log("QQo QNL="+QNL);/////////-/-/-/-/-/-/-/-/-/-/--/-/-/-/-/-/-/-/-/-/-/-
/-
      while(!(PZ[QNL]==1 && QRT[QNL]!=0 && NQ[QNL]==2 && QTR[QNL]==0) && QNL<100)
QNL++;
      if (QNL==100) {QNL=0;
      while(!(PZ[QNL]==1 && QRT[QNL]!=0 && NQ[QNL]==2 && QTR[QNL]==0) && QNL<100)
QNL++;}
      if (QNL==100) {QNL=0; QQ=0; FQ=0; return;}
      if (NPERIOD==1) QQPW(QNL);
      else if (NPERIOD!=3) {QNL=0; QQ=0; FQ=0; return;} else QQGO(QNL);
FQ=0;
} // End of function QQo()
////////////////////////////////////////////////////////////////////
function QQPW(SNL) // Обслуживание ввода одноразовых паролей через очередь
{     var RT=QRT[SNL];
      var TR,i,j;
      NQ[SNL]=0;
console.log("QQPW SNL="+SNL);/////////-/-/-/-/-/-/-/-/-/-/--/-/-/-/-/-/-/-/-/-/-
/-/-
      for (i=1;i<=503;i++) A[i]=NLA[SNL][i]; // Занесли принятый от клиента
массив битов в A[]
      POWA(SNL); // Вычисление степени A[] через MA[][] + вычисление RN[IRN][] и
RNB[IRN][]
      for (i=1;i<=256;i++) // Расшифровка по Вернаму
            {j=0; if (RT[i+5]=='1') j=1;
                  NLO[SNL][i]=j+NLC[SNL][i]; if (NLO[SNL][i]>1) NLO[SNL][i]=0;

            } // Результат расшифровки занесли в NLO[SNL][]
      // Преобразуем пароль с паспортными данные к виду хранения (в 72 байта)

            for (i=1;i<=144;i++) N[i]=NLO[SNL][i];
            for (i=145;i<=503;i++) N[i]=0;      // Перенесли результат расшифровки
в N[]
      STEPX(N); // Вычисление A[]=X[]^N[], где X[]- примитивный элемент, N[]-
биты пароля
PW='';
// Закодированный пароль с данными паспорта занесен в A[]. Переносим его в PW по
7 бит
            for (j=0;j<72;j++) // Формируем байты строки PW
                  {
                  var e=1+j*7;// Вычислили номер начального бита для текущего
байта PW
                  var w=0; // Числовое значение байта
                  if (A[e]  == 1) w= w+1;
                  if (A[e+1]== 1) w= w+2;
                  if (A[e+2]== 1) w= w+4;
                  if (A[e+3]== 1) w= w+8;
                  if (A[e+4]== 1) w= w+16;
                  if (A[e+5]== 1) w= w+32;
                  if (A[e+6]== 1) w= w+64;
                  PW= PW+String.fromCharCode(w);
                  } // Будем проверять наличие такого избирателя
            EEE=0; // Признак наличия избирателя (0-нет)
            if (Kstr>0) // Это защита для случая отсутствия избирателей
            {
      var SERNUM=''; // Строка для серии и номера паспорта
```

```
            for (i=0; i<Kstr; i++) // Поиска строки с зашифрованными данными
пасп.+пароля
            {SERNUM='';
            for (j=1; j<74; j++)
SERNUM=SERNUM+String.fromCharCode(SFILE[i*Lstr+j]);
            if (SERNUM=='0'+PW) {EEE=1; Nstr=i; break;} // Нашли номера паспорта
            }
            }
            if (EEE!=1) TR="E0"; // "Помилкові паспортні дані або помилковий
пароль."
            else
            {     // Запоминаем паспортные данные в битовом виде
            PASPBIT[Nstr]='';
NstrK[SNL]=Nstr; // Запомнили номер строки с данными избирателя
            for (j=1; j<33; j++) PASPBIT[Nstr]=PASPBIT[Nstr]+NLO[SNL][j];
            // Принимаем одноразовый пароль для голосования

var PWR=''; for (i=145;i<257;i++) if (NLO[SNL][i]==1) PWR=PWR+'1'; else
PWR=PWR+'0';
            PASSVYB[NstrK[SNL]]=PWR; // Запомнили одноразовый пароль в виде 112
"1" или "0"
            TR="S"; // Проверки прошли успешно
//    !!Следующие две строки нужно убрать при подключении UDP-сервера //-/-/-/-/-
/-/-/-/-!!!!!!!!!!!!!
PAU[SNL]=1; // Признак прохождения аутентификации
NstrK[SNL]=Nstr; // Запомнили номер строки с данными избирателя
            } // TR='E0'  "Помилкові паспортні дані або помилковий пароль."
            QTR[SNL]=TR;
var T1 = new Date(); // Берем метку времени -/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/
var T= T1.getTime();
console.log("QQ  TR="+TR+' T='+T);////////////////////////////////////-/-/-/-/-
/-/-/-/-/-/
//////////  А У Т Е Н Т И Ф И К А Ц И Я  /////////////// ПОКА ОТКЛЮЧЕНА
/*    TRSG=''; // Строка для подготовки данных на сервер аутентификации
      for (var i=1;i<=503;i++) // Переносли значение NLB[SNL][] в строку TRSG
      {if (NLB[SNL][i]==0) TRSG=TRSG+'0'; else TRSG=TRSG+'1';}
      PACK_(TRSG); // Уплотняем данные в 4 раза из TRSG в TRSP для передачи
      OLD_LEN=34;
      TRS=NU2[SNL]+TRSP;  // Будем отправлять 128 байт через буфер
                                    //   на сервер аутентификации (в AU.js)

var data = Buffer.from(TRS);
client.send(data,PORTA,HOSTA,function(err){
if (err) throw err;
console.log('To ' + HOSTA +':'+ PORTA+' L='+data.length+' SNL='+SNL);
});
*/
//    }   Временно отключено /-/--/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/
} // End of function QQPW()
////////////////////////////////////////////////////////////////////////
function QQGO(SNL) // Обслуживание голосования через очередь
{
      var RT=QRT[SNL];
      var TR,i,j;
            KBUL=(RT.length-150)/8; // Вычислили к-во бюллетней
            for (i=1;i<=503;i++) A[i]=NLA[SNL][i]; // Занесли принятый от клиента
массив в A[]
            POWA0(SNL); // Вычисление степени A[] через MA[][]
            for (i=1;i<=144+KBUL*8;i++) // Расшифровка по Вернаму
              {j=0; if (RT[i+5]=='1') j=1;
              NLO[SNL][i]=j+NLC[SNL][i]; if (NLO[SNL][i]>1) NLO[SNL][i]=0;}

            // Результат расшифровки занесли в NLO[SNL][]
      var PASP_R='';    // Принятые данные паспорта (в битах)
```

```
            for (i=1;i<=32;i++) {if (NLO[SNL][i]==1) PASP_R=PASP_R+'1'; else
PASP_R=PASP_R+'0';}
            PWG[SNL]=''; for (i=33;i<145;i++)
                              if (NLO[SNL][i]==1) PWG[SNL]=PWG[SNL]+'1'; else
PWG[SNL]=PWG[SNL]+'0';
      var EEE=0; // Признак наличия паспорта избирателя в PASPBIT[] ( 0-нет, 1-
есть)
      if (Kstr>0)
      { // Цикл поиска паспорта избирателя
      for (j=0; j<Kstr; j++)
      if (PASP_R==PASPBIT[j]) {EEE=1; Nstr=NstrK[SNL]=j;}
      } // Если нашли паспорт, то EEE=1, Nstr=j
      if (EEE!=1) TR="E0"; // "Помилкові паспортні дані"
      else
      {     // Сверяем только первые два байта пароля
      if (PASSVYB[Nstr].substr(0,14)!=PWG[SNL].substr(0,14)) TR="E3"; //
"Помилковий пароль."
      else
      { // Проверки прошли успешно. Будем принимать голос.
      GOLOSF[NstrK[SNL]]=1; // Установили признак голосования, включая фиктивное
      if (GOLOS[NstrK[SNL]]==1 ||  PASSVYB[NstrK[SNL]] != PWG[SNL])
      TR="S"; //  Отправляем сообщение "Ваш голос прийнято"
      else
      {
      var r=145; // установили номер в r для начала цикла по бюллетням
            for (var b=0; b<KBUL; b++)
            {
            var VYB=0; // Ячейка для номера, выбранного избирателем в бюллетне
            for (i=0; i<2; i++) // Определяем выбранный номер
                  {
                  j=0;
                  if (NLO[SNL][r]==1) j=j+1;
                  if (NLO[SNL][r+1]==1)j=j+2;
                  if (NLO[SNL][r+2]==1) j=j+4;
                  if (NLO[SNL][r+3]==1) j=j+8;
                  if (i==0) VYB=10*j; else VYB=VYB+j;
                  r=r+4;
                  } // Получили в VYB выбранный номер
            GOL[b][VYB]++; // Засчитали голос
            GOLOS[NstrK[SNL]]=1; // Пометили использование избирателем права
голоса
            TR="S"; // Отправляем сообщение "Ваш голос прийнято"
            }
      } // TR="S"; "Ваш голос прийнято"
      } // TR="E3" "Помилковий пароль."
      } // TR="E0" "Помилкові паспортні дані"
      QTR[SNL]=TR;
} // End of function QQGO()
////////////////////////////////////////////////////////////////////////
```

# ДОДАТОК Б

# АКТ ВПРОВАДЖЕННЯ РЕЗУЛЬТАТІВ РОБОТИ

**З А Т В Е Р Д Ж У Ю**

В.о. директора ДП Державний науково-дослідний інститут автоматизованих систем в будівництві (ДНДІАСБ)

_____ С.В. Басько

«_____» _____ 2019 р.

## АКТ ВПРОВАДЖЕННЯ

**результатів дисертаційної роботи здобувача наукового ступеня кандидата технічних наук Комарницького Олега Олександровича у комп'ютерній мережі ДНДІАСБ**

Я, що нижче підписався, *радник директора ДНДІАСБ Слободян Я.О.,* склав цей акт про те, що результати наукових досліджень *за темою дисертаційної роботи на здобуття наукового ступеня кандидата технічних наук Комарницького Олега Олександровича «Методи вдосконалення технологій дистанційного таємного Інтернет-голосування»* використовуються в комп'ютерній мережі ДНДІАСБ.

| **Найменування впровадженого результату** | **Форма впровадження і досягнутий фактичний ефект** |
|---|---|
| *Транспарентна система* | *Діюча транспарентна системи* |

| | |
|---|---|
| *дистанційного таємного голосування, у якій є можливість повного контролю з боку будь-якої особи усіх програмно-апаратних засобів, які забезпечують таємницю та точність підрахунку голосів, що надає впевненості у відсутності зловживань під час проведення голосування.* | *дистанційного таємного голосування, програмно-апаратні засоби якої побудовано на основі результатів даної дисертаційної роботи і встановлено у Інтернет-вузлі ДНДІАСБ на сервері за адресою http://91.198.50.8:29901/VD999901.html. Система відкрита для експериментального голосування і для контролю програмно-апаратних засобів з метою демонстрації усіх її характеристик у робочому стані.* |
| *Метод протидії атакам посередника, який дозволяє кожному виборцю перед здійсненням акту волевиявлення за допомогою штатних термінальних засобів мережі Інтернет впевнитись у тому, що він дійсно спілкується зі штатним сервером своєї виборчої дільниці, а не його фальшивим аналогом.* | *Діюче програмне забезпечення для контролю запитів виборців розроблено за методом, який запропонований у даній дисертаційній роботі, що дозволяє виборцям одразу після встановлення зв'язку з сервером шляхом порівняння числових кодів впевнитись у відсутності атаки посередника. Цей метод реалізовано на усіх серверах для голосування за адресами:*<br>*http://91.198.50.8:29901/VD999901.html*<br>*http://91.198.50.130:29900/VD999900.html*<br>*http://91.198.50.132:29902/VD999902.html* |

Радник директора ДП

ДНДІАСБ д.т.н., професор _____ Я.О. Слободян