МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЦЕНТРАЛЬНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ
ТЕХНІЧНИЙ УНІВЕРСИТЕТ

УДК 004.4

Тези доповідей

V Міжнародної науково-практичної конференції

"Інформаційна безпека та комп'ютерні технології"

19–20 травня 2022 р.

Кропивницький 2022

УДК 004.4

Матеріали V Міжнародної науково-практичної конференції "Інформаційна безпека та комп'ютерні технології": тези доповідей, 19–20 травня 2022 р. – Кропивницький: ЦНТУ, 2022. – 72 с.

Наведені тези пленарних та секційних доповідей за теоретичними та практичними результатами наукових досліджень і розробок. Представлені результати теоретичних досліджень в галузях проектування інформаційних систем, технологій захисту інформації, використання сучасних інформаційних технологій в управлінні системами за різними галузями народного господарства.

Матеріали публікуються в авторській редакції.

За достовірність викладених фактів, цитат та інших відомостей відповідальність несуть автори.

СЕКЦІЯ 1. ІНФОРМАЦІЙНА БЕЗПЕКА ДЕРЖАВИ, СУСПІЛЬСТВА ТА ОСОБИСТОСТІ

УДК: 004.932

О.М. Дрєєв , Є.В. Мелешко
drey_sanya@ukr.net., elismeleshko@gmail.com
Центральноукраїнський національний технічний університет, м. Кропивницький

ВИКОРИСТАННЯ НЕЙРОННОЇ МЕРЕЖІ ДЛЯ ПЕРЕДБАЧЕННЯ ТИСНЯВИ В ГРОМАДСЬКИХ МІСЦЯХ

Авторами розглянуто проблему спостереження за скупченнями людських мас з метою запобігання тисняви та супутніх негативних явищ. В якості вхідних даних визначено чотирьохканальний відеоряд з компонентами червоного, зеленого, синього та теплового інфрачервоного частин спектру. З причини того, що відеоряд отриманий з малих літальних апаратів, зображення містить змінне масштабування об'єктів зацікавленості (людські маси) та присутнє переміщення фону.

За існуючими роботами розмітки присутності людських мас [1, 2], було вирішено вдосконалити наявні технології комбінуванням видимого та теплового спектру. Важливим фактором, який дозволив спростити архітектуру нейронної мережі стало те, що зображення, які підлягали аналізу, мали розміри людських фігур наближено одного розміру, бо кут відеоспостереження був близьким до перпендикуляру до площі землі або спостереження велося зі значної відстані.

Навчальні данні готувалися шляхом встановлення плям на місці розташування тіла людини на теплових кадрах. Проте, навчання нейронної мережі проводилося малими фрагментами зображення 32x32 пікселі з урівноваженням кількості фрагментів з наявними зображеннями людини та без. Фрагменти утворюються за допомогою проходу вікна 32x32 пікселі по повному зображенню з кроком в кілька пікселів. Такий метод навчання дозволив отримати працездатну нейронну мережу лише на кількох десятках кадрів для навчання. Дані для навчання нейронної мережі та апробації роботи системи було взято з відкритих джерел [3]. Розмічення наявності людських мас дозволило створити маску для відокремлення фонового зображення та людських потоків. Відокремлені зображення аналізуються на наявність оптичних потоків, методом Лукаса-Канаде, що містить бібліотека OpenCV [4]. Оптичні потоки фону використовуються для компенсації виявлених переміщень людських мас. Це дозволило отримати переміщення людських мас незалежними від переміщення камери і використати систему не лише для стаціонарних камер спостереження. Повна схема функціонування системи передбачення скупчення людських мас показана на діаграмі на рис. 1:

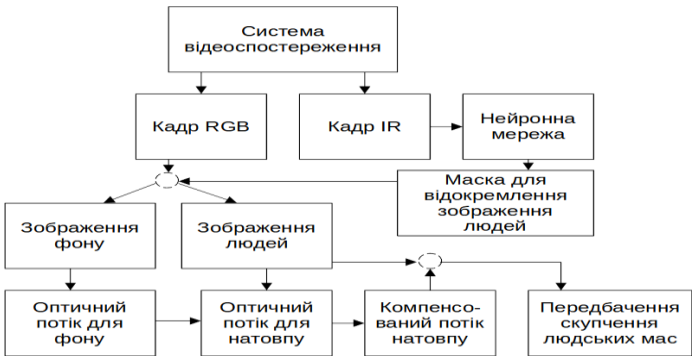

Рис.1. Схема функціонування системи передбачення скупчення людських мас

В результаті роботи отримано систему, яка в реальному часі дозволяє відмічати зони, для яких є тенденція збільшення кількості людей, що допоможе завчасно вжити запобіжних заходів щодо утворення тисняви.

Список літератури

1. Lei, Tao & Zhang, Dong & Wang, Risheng & Li, Shuying & Zhang, Weijiang & Nandi, Asoke. (2021). IET Image Processing MFP-Net: Multi-scale feature pyramid network for crowd counting. IET Image Processing. 15. 10.1049/ipr2.12230.

2. Li, Baiping & Han, Xinyi & Wu, Dongmei. (2018). Real-Time Crowd Density Estimation Based on Convolutional Neural Networks. 690-694. 10.1109/ICITBS.2018.00179.

3. Протести в Балтіморі, штат Меріленд, 2015, відео з повітряного спостереження. URL: https://vault.fbi.gov/protests-in-baltimore-maryland-2015/unedited-versions-of-video-surveillance-footage

УДК 004(056.53+413.4):001.51

В.В. Мохор[1], О.О. Бакалинський[1], В.В. Цуркан[1,2]
v.mokhor@gmail.com, baov@meta.ua, v.v.tsurkan@gmail.com
[1]*Інститут проблем моделювання в енергетиці ім. Г.Є. Пухова Національної академії наук України, м. Київ*
[2]*Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», м. Київ*

СПЕЦИФІКАЦІЯ ЗАХОДІВ ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ

Розроблення систем управління інформаційною безпекою зводиться до обирання відповідних заходів. Вона визначається їхньою сукупністю, відношеннями між ними та впроваджується для збереження властивостей інформаційних активів організацій [1]. Серед основних виокремлюється тріада: конфіденційність, цілісність і доступність. Їхня збереженість досягається завдяки обробленню ризиків інформаційної безпеки. Потреба у виконанні даного етапу встановлюється за результатами оцінювання. Це дозволяє, по-перше, прийняти рішення про необхідність переходу до оброблення ризиків інформаційної безпеки. По-друге, обрати варіанти оброблення (наприклад [2], змінення, утримання, уникнення, розподілення) і, як наслідок, по-третє, заходи в межах обраного або обраних варіантів. Тож специфікування заходів забезпечення інформаційної безпеки є актуальним завданням.

Заходами забезпечення інформаційної безпеки визначаються дії, що направлені на змінення, утримання, уникнення та/або розподілення ризиків інформаційної безпеки. Загальноприйняті сукупності дій зведено в [3], а саме: організаційні, кадрові, фізичні, технологічні. Водночас передбачається адаптування як загальноприйнятних, так і розроблення власних заходів забезпечення інформаційної безпеки відповідно до потреб організацій з урахуванням особливостей їхніх діяльностей [4]. Узгодженість і одноманітність використання кожного зі заходів досягається шляхом структурованого представлення інформації про них (табл. 1 [3]). Вони характеризується назвою, належністю до однієї з чотирьох категорій (організаційна, кадрова, фізична, технологічна) та набором атрибутів. Атрибутами враховуються вплив на ризик (запобігання, виявлення, коригування); збереження властивостей інформаційних активів (конфіденційність, цілісність, доступність); діяльності зі забезпечення кібербезпеки (ідентифікувати, захищати, виявляти, відповісти, відновити); можливості забезпечення інформаційної безпеки (наприклад, управління_подіями_забезпечення інформаційної_безпеки); домени забезпечення безпеки (управління та екосистема, захист, оборона, стійкість). Водночас даний перелік може змінюватися, зокрема, розширюватися шляхом задання нових атрибутів з урахуванням особливостей діяльностей організацій.

Таблиця 1
Звітування про події забезпечення інформаційної безпеки

Тип заходу	Властивість інформаційного активу	Концепція забезпечення кібербезпеки	Операційні можливості	Домен забезпечення безпеки
#Виявлення	#Конфіденційність #Цілісність #Доступність	#Виявляти	#Управління_подіями_забезпечення інформаційної_безпеки	#Оборона

Отже, специфікування заходів забезпечення інформаційної безпеки дозволить структурувати представлення інформації про них. Завдяки цьому стане можливим, з одного боку, узгоджене та одноманітне їх використання відповідно до настанов [3]. Тоді як з іншого – адаптування до потреб організацій з урахуванням особливостей їхніх діяльностей при розробленні і впровадженні систем управління інформаційної безпекою.

Список літератури

1. ISO/IEC 27001:2013. Information technology. Security techniques. Information security management systems. Requirements. [Valid from 2013-09-25; revised 2018-12-04]. URL: https://www.iso.org/standard/ 54534.html (accessed on: 14.05.2022).

2. ISO/IEC 27005:2018. Information technology. Security techniques. Information security risk management. [Valid from 2018-06-10]. URL: https://www.iso.org/standard/75281.html (accessed on: 14.05.2022).

3. ISO/IEC 27002:2022. Information technology. Information security, cybersecurity and privacy protection. Information security controls. [Valid from 2022-02-15]. URL: https://www.iso.org/standard/75652.html (accessed on: 14.05.2022).

4. ISO/IEC 27011:2016. Information technology. Security techniques. Code of practice for Information security controls based on ISO/IEC 27002 for telecommunications organizations. [Valid from 2016-11-23; revised 2020-01-27]. URL: https://www.iso.org/standard/64143.html (accessed on: 14.05.2022).

УДК 004.056.5:004.77]:355.1

Я.І. Шестак
shestack@knute.edu.ua
директор Інформаційно-обчислювального центру
Головного центру інформаційних технологій,
старший викладач кафедри програмної інженерії та кібербезпеки
Державний торговельно-економічний університет, м. Київ

КІБЕРГІГІЄНА У ІНФОРМАЦІЙНОМУ ПРОСТОРІ В УМОВАХ ВОЄННОГО СТАНУ

Стратегія побудови супер інтелектуального суспільства в умовах розвитку промислової революції не обходить інформаційний простір. У Суспільстві 5.0 штучний інтелект, робототехніка, робота з BigData, інтернет речей (IoT) стрімкими темпами вносять корективи у роботу цифрових технологій в усіх сферах життя суспільства. Так, під єдиним інформаційним простором країни слід розуміти сукупність інформаційних ресурсів та інформаційної інфраструктури, що дозволяє на основі єдиних принципів та за загальними правилами забезпечувати безпечну інформаційну взаємодію держави, організацій та громадян за рахунок доступу до відкритих інформаційних ресурсів.

Сучасний світ давно зробив перший крок до принципово новітньої технологічної, економічної та соціальної реальності – нової епохи цифрової глобалізації. Забезпечення кібербезпеки є одним з вагомих пріоритетів в загальній системи національної безпеки України. XXI ст. характеризується активним формуванням ризиків з якими зіштовхується цивілізація впроваджуючи новітні технології [1]. Питома вага кіберзагроз зростає і ця тенденція в міру розвитку інформаційних технологій та їх конвергенції з технологіями штучного інтелекту в найближче десятиліття посилюватиметься. Зростання такого впливу на функціонування структур управління як національних, так і транснаціональних формує нову безпекову ситуацію. Між світовими центрами сили відбувається поділ сфер впливу у кіберпросторі, посилюється їх прагнення за рахунок такого поділу забезпечити реалізацію власних геополітичних інтересів.

Кіберпростір разом з іншими фізичними просторами визнано одним з можливих театрів воєнних дій. Набирає сили тенденція зі створення кібервійськ, до завдань яких належить не лише забезпечення захисту критичної інформаційної інфраструктури від кібератак, а й проведення превентивних наступальних операцій у кіберпросторі, що включає виведення з ладу критично важливих об'єктів інфраструктури противника шляхом руйнування інформаційних систем, які управляють такими об'єктами [1]. Фундаментом дієвої системи кібербезпеки, безумовно, є ефективна нормативно-правова база а тому слід зазначити про актуальність Указу Президента України «Про рішення Ради національної безпеки і оборони України «Про Стратегію кібербезпеки України» від 26.08.2021 р. [1]. Відповідно до зазначеної стратегії передбачено перспективи створення максимально вільного та безпечного, відкритого і стабільного кіберпростору в інтересах забезпечення прав людини.

Сьогодні, коли країна знаходиться в умовах воєнного стану важливим є не лише для IT-фахівців, але й для будь якого громадянина навчання застосування цифрової грамотності, дотримання цифрового етикету та правил кібергігієни. З початком війни IT-фахівці з усієї країни долучилися до кіберполіції та зуміли дати відсіч агресору. В результаті злагоджених дій були виведені з ладу критично важливі інформаційні системи окупанта. Проте і українцям необхідно дотримуватися вимог кібергігієни оскільки інформаційний простір це джерело поширення фейків, діпфейків, підробки сайтів, фішингових атак, заволодіння акаунтами та ін. Розглянемо детальніше можливі загрози та правила кібергігієни у боротьбі з ними.

Фейки. В умовах воєнного стану, українці щохвилини оновлюють стрічку новин, з метою дізнатися про ситуацію на фронті та в дипломатичному полі. Тим часом ворог масово розповсюджує фейки про начебто захоплення міст, капітуляцію України чи евакуацію місцевих [2].

Правило кібергігієни у боротьбі з фейками: читати лише офіційні та перевірені джерела, а не сумнівні телеграм-канали та пости в соцмережах. Однак потрібно пам'ятати, що в умовах воєнного часу навіть надійні медіа та офіційні особи можуть помилятися. Прочитавши важливу новину, дочекайтеся її спростування чи підтвердження.

Діпфейки. Дещо складніша ситуація оскільки діпфейк це підроблене відео, на якому можна побачити публічну особу з виступом та відповідно чути її голос. Так, у Центрі інформаційної безпеки повідомляли, що у мережі може з'явитися відеозвернення Президента Володимира Зеленського про начебто капітуляцію. Однак це технологія машинного навчання, яку можуть використати, щоб заплутати вас і зламати бойовий дух [1].

Правило кібергігієни у боротьбі з діпфейками: можна помітити неприродній тон і дивну текстуру шкіри, тіні на обличчі, які неправильно падають, забагато чи замало кліпання тощо, а основне – дивитися звернення лише на офіційних сторінках.

Підробки сайтів та акаунтів офіційних структур. Для дезорієнтації населення з'являється безліч аналогів сторінок офіційних структур. Так, можна натрапити на фейковий акаунт Верховної Ради України в соціальних мережах або підроблений сайт Служби безпеки України.

Правило кібергігієни у боротьбі з підробкою сайтів та акаунтів офіційних структур: якщо йдеться про сторінки в соціальних мережах, доречно звернути увагу на те, чи верифікований аккаунт (синя галочка поруч із назвою). Друга особливість – невелика кількість підписників та дописів. Що стосується веб-сайтів, то треба звернути увагу чи в адресному рядку браузера відображається символ замочку. Це означає, що сторінка пройшла детальну перевірку та отримала сертифікат безпеки. Додатково сайт можна перевірити через сервіс Whois та дізнатися дату реєстрації/створення, приналежність компанії та інші юридичні дані.

Фішингові посилання. Найпоширеніший і найбільш дієвий спосіб інтернет-шахрайства під час війни використовують і на кіберфронті. Зловмисники розсилають файли різних форматів – від посилання на сайт до архівованого документу чи медіафайлу. Можна отримати несподіваного листа як на електронну пошту, так і в особисті повідомлення в соцмережах чи месенджерах. Ще один, удосконалений метод фішингу – підробка посилань на підпис електронних петицій [1]. Тому важливо навчитися розрізняти достовірні посилання від підроблених.

Правило кібергігієни у боротьбі з фішинговими посиланнями: перше і головне правило – не відкривати несподівані посилання та файли. Якщо їх надіслали знайомі контакти, необхідно уточнити, що саме знаходиться в повідомленні, адже сторінки ваших друзів можуть зламати. Також необхідно уважно читати адресу сайту оскільки часто зловмисник підмінює всього одну літеру чи символ.

Заволодіння акаунтами у Telegram. У перші дні війни чимало користувачів Telegram скаржилися, що в налаштуваннях сторінки їм відображало власне місцезнаходження у російських містах.

Правило кібергігієни у боротьбі з заволодінням акаунтів: перевірити, чи аккаунт у Telegram не використовують зловмисники у налаштуваннях де вказано з якого регіону підключений пристрій. Щоб знизити ризик злому акаунту варто підключити двофакторну аутентифікацію. Такі правила стосуються і акаунтів у всіх соціальних мережах.

Швидко змінюваний цифровий світ потребує формування більш збалансованої та ефективної національної системи кібербезпеки, яка зможе гнучко адаптуватися до змін безпекового середовища, гарантуючи громадянам України безпечне функціонування національного сегмента кіберпростору, передбачивши нові можливості для цифровізації всіх сфер суспільного життя.

Україна має бути здатною забезпечити свій соціально-економічний розвиток у цифровому світі, що вимагає набуття спроможності ефективно стримувати деструктивні дії в кіберпросторі, досягнення кіберстійкості на всіх рівнях та взаємодії всіх суб'єктів забезпечення кібербезпеки, яка ґрунтується на довірі. Отже, встановлено, що нова ера кібербезпеки вимагає цілком нових підходів до управління ресурсами, зокрема інформаційними. Успіх таких змін значною мірою залежить від того, як гнучко організовані процеси на рівні держави, а також як імлементуються нові моделі та методи роботи у боротьбі з можливими загрозами.

Список літератури

1. Про рішення РНБО України від 14.05.2021 р. «Про Стратегію кібербезпеки України» від 26.08.2021 р. № 447/2021.: Указ Президента України. URL: https://www.president.gov.ua/documents/4472021-40013

2. Кібербезпека в умовах війни: як встояти на інформаційному фронті URL: https://pingvin.pro/gadgets/news-gadgets/kiberbezpeka-v-umovah-vijny-yak-vstoyaty-na-informaczijnomu-fronti.html

УДК 004.031 42

Т.Х. Фаталієв, Ш.А. Мехтієв
tfataliyev@gmail.com, shakir.mehtieff@gmail.com
Інститут інформаційних технологій НАН Азербайджану, м. Баку

КІБЕРСТІЙКІСТЬ КРИТИЧНИХ ІНФРАСТРУКТУР НАУКИ

В останні роки перспективи додатків Industry 4.0 під час проведення наукових досліджень та експериментів актуалізували питання забезпечення кіберстійкості систем, що застосовуються у цьому середовищі. Складність і невизначеність, властиві їхньому операційному середовищу, мали значний вплив на стійкість, як ключового критерію, який слід враховувати у доповненні до традиційних, таких, як гнучкість, швидкість реагування, надійність тощо, які використовуються при проектуванні, управлінні та експлуатації цих систем. У міру ускладнення, так звані кіберсистеми набувають актуальних системних властивостей: кіберстійкість, керованість, самоорганізація, проактивна кібербезпека та адаптивність [1]. Таким чином, кіберстійкість є найважливішою особливістю критичних інфраструктур науки, схильною до впливу базових технологій Industry 4.0, таких як Інтернет речей (IP), кіберфізичні системи (КФС), штучний інтелект, хмарні обчислення, аналітика великих даних і т. д. [2]. Слід зазначити, що кіберстійкість є важливим показником для кожної з цих технологій, що робить необхідним забезпечення повної стійкості всього середовища. При цьому формування умов сталої роботи кожної критичної інфраструктури та одночасно їх взаємодія одна з одною є основою процесу проектування подібних систем. На відміну від кібербезпеки, яка орієнтована на оцінку ймовірності інцидентів та запобігання можливим загрозам безпеці, кіберстійкість в основному спрямована на збереження цільової поведінки та продуктивності цих інфраструктур в умовах виникнення кіберінцидентів.

Сучасну науку в контексті трансформації в Science 4.0 можна розглядати як складну корпоративну систему, засновану на взаємозв'язаних розумних підсистемах: будівлі, інфраструктура, ресурси, дослідницька середа, КФС, менеджмент, комплексна безпека та ін. [3]. Однак серед множества проблем, що виникають при подібній трансформації, особлива значущість безпеки забезпечує забезпечення комплексної та кіберостійкості як її підсистеми, так і всієї системи.

Science 4.0 характеризується такими особливостями:

- Високий рівень взаємодії смарт-об'єктів, що входять до її складу.
- Генерація великих даних та інтеграція смарт-об'єктів із аналітикою великих даних.
- Динамічні зміни структури смарт-об'єктів для досягнення більшої гнучкості.
- Глобальний зворотний зв'язок та координація для підвищення ефективності наукових досліджень.

Однак ці особливості припускають, що Science 4.0 має складнішу структуру і може зіткнутися з різними типами загроз. Це можуть бути загрози, пов'язані з інформаційною безпекою, конфіденційністю даних та факторами, пов'язаними з кібербезпекою. Хоча кібербезпека стає невід'ємною частиною стратегії безпеки в науковому середовищі, проте не завжди забезпечується захист від складних кібератак, коли несанкціонований доступ до інформації може призвести до небажаних наслідків. Таким чином, гарантія кіберстійкості критичних інфраструктур Science 4.0, тобто забезпечення цілеспрямованої роботи системи при впливі атак з кіберпростору, є актуальною проблемою, що потребує превентивних рішень. Кіберстійкість - це здатність системи передбачати, витримувати, відновлюватися і адаптуватися до несприятливих умов, навантажень, атак, які використовують або активуються кіберресурсами [4].

На відміну від кібербезпеки кіберстійкість це ширше поняття, яке включає забезпечення безперервності функціонування; безпека критично важливих процесів; виявлення потенційних загроз; управління ризиками; мінімізацію атак та впровадження процедур проти інцидентів кібербезпеки. Вона дозволяє продовжити свою звичайну роботу без перерви під час та після деструктивних подій, таких як кібератаки або технічні збої.

Кібератаки є цілком реальною загрозою для Science 4.0, оскільки вона заснована на застосуванні інноваційних технологій і, особливо, Інтернету. Для такого середовища "ландшафт" кібербезпеки складний і постійно змінюється. Відомо, що кібербезпека пов'язана із захистом інформаційних активів та систем з урахуванням наступних метрик: конфіденційність, цілісність та доступність. Оцінювання ризиків є важливою частиною визначення того, які заходи кібербезпеки необхідно реалізувати і якою мірою, щоб захист був рентабельним і стійким. Більшість будь-якої оцінки ризиків повинна полягати у виявленні загроз і вразливостей. Насамперед, щоб оцінити ризики кібербезпеки, необхідно визначити, які активи потребують захисту. Після цього, відповідно, виявляються потенційні вразливості та загрози для цих активів. Отже, необхідно враховувати як зазначені метрики безпеки, а й групи ризику, куди має спиратися захист: співробітники, ресурси, процеси і технології.

Для забезпечення комплексної безпеки і тим самим кіберстійкості критичних інфраструктур Science 4.0 слід вирішити такі проблеми:

- Управління безпекою, захистом та конфіденційністю;
- Охорона будівлі, безперебійне електропостачання, клімат-контроль, відеоспостереження, контроль доступу;

- Кібербезпека мереж, ЦОД, АСУ науковими експериментами, ресурсами, інструментами та обладнанням;
- Конфіденційність даних, захист особистої інформації, дотримання Загального регламенту захисту даних (GDPR);
- Безпека інфраструктури інформаційного забезпечення науки;
- Політика та процедури безпеки, аудит безпеки та звітність;
- Аналітичний моніторинг, моделювання атак, контрольоване тестування DDoS тощо;
- Автоматизація процесів із широким застосуванням ІВ, КФС, ІІ, аналітики великих даних та інших передових технологій у забезпеченні комплексної безпеки;
- Розробка смарт та кіберімунних ІТ-рішень для забезпечення кіберстійкості;
- Поінформованість співробітників про кібербезпеку та навчання;
- Співпраця з вітчизняними та міжнародними організаціями тощо.

Розглянемо деякі концептуальні питання забезпечення кіберстійкості.

Заходи щодо запобігання ризикам можна поділити на три категорії: а) превентивні, б) у реальному часі та в) після інциденту. Для ефективної реалізації заходів щодо усунення ризиків найрезультативнішим вважається використання динамічного підходу, при якому окремі заходи безпеки ефективно працюють разом і компенсують недоліки один одного. Застосування глибокоешелонованого захисту може стати одним з найефективніших інструментів запобігання атакам на різних рівнях. При розгляді варіантів атак щодо ешелонованого захисту важливо враховувати, як можна уникнути засобів контролю. Для цього потрібно знайти найслабшу ланку в системі безпеки і далі для зниження ризику слід зміцнити цю ланку. Зазначимо, що останніми роками розроблена прогресивна концепція кіберімунності для безпечного цифрового середовища, яка відкриває нові перспективи та для безпеки Science 4.0 [5]. Аналогічно імунній системі живих організмів кіберімунні ІТ-рішення створюються вихідно захищеними від цілого класу кіберзагроз, і складаються з елементів, які постійно стежать за незвичною поведінкою системи та попереджають про підозрілу активність. Наприклад, архітектура KasperskyOS заснована на розподілі об'єктів на безліч ізольованих модулів. Усі взаємодії з-поміж них контролюються лише на рівні мікроядра і внутрішньої системи безпеки: вони дозволяють лише ті події, що були позначені ще етапі розробки системи.

На закінчення слід зазначити, що можна виділити технічні, організаційні, економічні та соціальні аспекти кіберстійкості критичних інфраструктур середовища Science 4.0. Досліджувана проблема багатоаспектна, складна і тому потребує поетапного комплексного вирішення.

Список літератури

1. С.А. Петренко, Киберустойчивость Индустрии 4.0, Санкт-Петербург, Издательский Дом "Афина", 2020, 256 с.

2. Т.Х. Фаталиев, Ш.А. Мехтиев, О возможностях применения решений Industry 4.0 в науке. IV Международная научно-практическая конференция "Информационная безопасность и компьютерные технологии", Украина, г. Кропивницкий, 15-16 апрель 2021, сс. 67-68.

3. T. Kh. Fataliyev, Sh. A. Mehdiyev, Integration of Cyber-Physical Systems in E-Science Environment: State-of-the-Art, Problems and Effective Solutions, I.J. Modern Education and Computer Science, № 9, pp. 35-43, 2019.

4. R.S. Ross, R. Graubart, D. Bodeau, R. McQuaid, Systems Security Engineering: Cyber Resiliency Considerations for the Engineering of Trustworthy Secure Systems. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) 800-160, 2019, Vol. 2. https://doi.org/10.6028/NIST.SP.800-160v2.

5. B. Pang,, etc., The cyber-physical immune system: work-in-progress, Proceedings of the 2021 International Conference on Embedded Software, September 2021, pp 43–44.

УДК 004.414

аспірант М.О. Кобець, к.т.н., доц. А.С. Коваленко, д.т.н., доц. О.В. Коваленко
nicko9298@gmail.com, annasun911@gmail.com, dr.kovalenkoov@gmail.com
Центральноукраїнський національний технічний університет, м. Кропивницький

РОЗРОБКА КОМПОНЕНТІВ ТЕХНІЧНОЇ ДІАГНОСТИКИ ІНТЕГРОВАНИХ ІНФОРМАЦІЙНИХ СИСТЕМ З ВРАХУВАННЯМ ПАРАМЕТРІВ БЕЗПЕКИ

Проектне управління та розробку компонентів технічної діагностики інтегрованих інформаційних систем в IT та інших сферах можна розглядати за допомогою вивчення та використання ефективних підходів, методик та способів. Всі розроблені методи управління проектами відрізняються деталізованістю, формалізованістю, областями застосування та ін. Такі методи управління проектами є дуже різноманітними, системними підходами до управління. Отже в якості основи до компонентів технічної діагностики інтегрованих інформаційних систем можуть бути застосовані різноманітні відмінні концепції. Більшість сучасних підприємств схильна до постійної динамічної зміни у зв'язку з конкурентною модифікацією ринків збуту незважаючи на напрям роботи, у зв'язку з чим проходе розширення та оновлення продукції (програмні та апаратні рішення), створення та модифікації версій продукту та відповідно компонентів технічної діагностики. Крім того, в процесі розробки відбуваються кадрові зміни, а з плином часу змінюється і політика безпеки та технології діагностики безпеки автоматизованих систем керування. Ці зміни викликають нові ризики, а ризики, які раніше були передбачені, можуть знову стати проблемою. Таким чином, процес розробки та модернізації діагностичних систем та засобів тестування безпеки автоматизованих систем керування знаходиться в постійному розвитку.

У результаті проведених досліджень було сформовано ряд практичних рекомендацій з управління силами та засобами під час розробки компонентів технічної діагностики в умовах використання гнучкої методології розробки з нестандартним розподілом ролей в команді і унікальною організацією ітерацій SCRUM [1,2]. Успішна програма управління розробкою компонентів технічної діагностики та тестування безпеки неможлива без спільної підтримки та участі керівництва, команди та експертів безпеки.

Аналіз літератури [3-5], а також проведені дослідження показали, що більшість існуючих методів та засобів управління розробкою компонентів технічної діагностики та засоби тестування безпеки автоматизованих систем керування враховують можливості вже застарілої методології розробки, а саме – каскадної системи розробки. Однак в даний час більшість фірм використовує гнучкі методології розробки й управління SCRUM чи її гібридів, це гнучка методологія розробки з нестандартним розподілом ролей в команді і унікальною організацією ітерацій.

Виконання рекомендацій дозволяє: проводити перевірку розробки компонентів технічної діагностики, документів та питань безпеки; організацію роботи з ключовими елементами безпеки; проводити аналіз та оцінку змін технічної моделі, середовища обробки даних, політику безпеки; зрозуміти необхідність постійного збільшення частки автоматизованих підходів технологій тестування безпеки та управління процесами на всіх етапах розробки та реалізації продукту; визначати та встановлювати параметри якості системи; забезпечити необхідний заданий замовником рівень якості моделі без залучення значних ресурсів та сил (експертів безпеки).

Список літератури

1. Putu Adi Guna Permana Scrum Method Implementation in a Software Development Project Management/ Putu Adi Guna Permana // International Journal of Advanced Computer Science and Applications. – 2015. – Vol. 6. № 9. – P. 199-205.

2. Krishnan M. Soumya Software Development Risk Aspects and Success Frequency on Spiral and Agile Model / M. Soumya Krishnan // International Journal of Innovative Research in Computer and Communication Engineering (An ISO 3297: 2007 Certified Organization). – 2015. –Vol.3. – №1.

3. J. Highsmith Agile Software Development Ecosystems. – Boston: AddisonWesley, 2006. – 448p.

4. Soumya Krishnan M. Software Development Risk Aspects and Success Frequency on Spiral and Agile Model. / M. Soumya Krishnan // International Journal of Innovative Research in Computer and Communication Engineering (An ISO 3297: 2007 Certified Organization) Vol. 3, Issue 1, January 2015 pp. 301-310.

5. Zeng Y. Risk Management For Enterprise Resource Planning System Implementations in Project-Based / Y. Zeng // Firms : dis. for the degree of PHD, Maryland, 2010. – P. 210.

6. Коваленко О.В. Моделі та методи розроблення програмного забезпечення комп'ютерних систем для підвищення безпеки даних: монографія / О.В. Коваленко // К.: Вид. «КОД» – 2019. – 295 с.

7. Demarco T, Lister T Waltzing with bears: Managing risk on software projects. New York, 2003. 342 p.

УДК 004.056.53

О.С. Сосна, О.К. Конопліцька-Слободенюк
litetab387@gmail.com, ksuha80@gmail.com
Центральноукраїнський національний технічний університет, м. Кропивницький

ПРИВАТНІСТЬ ТА ЗАХИСТ ПЕРСОНАЛЬНИХ ДАНИХ

Персональні дані – це відомості чи сукупність відомостей про фізичну особу, яка ідентифікована або може бути ідентифікована.

Ідентифікована особа – це та особа, яку можна безпомилково ідентифікувати серед інших людей, за допомогою персональних даних (ім'я, прізвище, номер паспорту, ідентифікаційний номер).

За Законом про "захист персональних даних" персональні дані поділяються на дві категорії:

- загальні(звичайні);
- особливі(чутливі).

Рис. 1 Категорії персональних даних

Налаштування конфіденційності має велике значення для захисту персональних даних, але важливо також докласти максимум зусиль для збереження даних. Якщо приватність стосується прав людини на керування її особистою інформацією, то безпека – це сукупність методів захисту.

Існує декілька простих, але важливих методів захисту:

- подвійна автентифікація це подвійний захист при вході в якусь систему. Після вводу логіна і пароля на телефон надходить SMS з одноразовим кодом. Цей код вводиться на сайті для входу до персонального кабінету.
- генерація важких паролів. Створення надійного пароля - це застосування в паролі цифр та букв (самий простий метод захисту) , або можна сгенерувати пароль, але тоді слід його десь записати, тому що він буде досить здоровий.

Список літератури

1. Ю. Родичев "Нормативная база и стандарты в области информационной безопасности" (2017).
2. Е. Баранова, А. Бабаш "Информационная безопасность и защита информации" 3-е изд. (2016)
3. В. Бондарев "Введение в информационную безопасность автоматизированных систем".

УДК 004.9

В.В. Прокопов, Є.В. Мелешко, М.С. Якименко, С.В. Шимко
elismeleshko@gmail.com
Центральноукраїнський національний технічний університет, м. Кропивницький

ДОСЛІДЖЕННЯ МОЖЛИВОСТЕЙ НАБОРУ ДАНИХ CSE-CIC-IDS ДЛЯ ТЕСТУВАННЯ СИСТЕМ ВИЯВЛЕННЯ КІБЕРАТАК

Забезпечення комп'ютерної безпеки у різних сферах людської життєдіяльності є задачею, яка має всі підстави розглядатися, як одна із найбільш важливих проблем сучасного суспільства. Комп'ютери та Інтернет-мережа стають все більш потрібними людям у роботі та проведенні дозвілля. Зростає і значимість наявності вразливостей в комп'ютерних системах, що привертають увагу зловмисників, які використовують їх як можливість отримати гроші або заподіяти шкоду [1, 2]. Також вражливості комп'ютерних систем можуть використовуватися протиборчими сторонами для здобуття переваги під час інформаційних протиборств та війн [2]. Тому вкрай важливо захищати комп'ютерні системи та веб-ресурси від кібератак, а також розпізнавати успішні атаки для їх своєчасного усунення.

Метою даної роботи є дослідження можливостей набору даних CSE-CIC-IDS для тестування систем виявлення кібератак на веб-сайти.

Датасет CSE-CIC-IDS [3] був створений за результатами аналізу мережевого трафіку в ізольованому середовищі, в якому моделювалися дії звичайних користувачів, а також шкідливі дії порушників. Набір даних CSE-CIC-IDS містить розмічений мережевий трафік з наявністю поширених кібератак та відповідає вигляду реального трафіку у форматі PCAP. У ньому представлені наступні атаки [3]: DoS Hulk, PortScan, DDoS, DoS GoldenEye, FTP-Patator, SSHPatator, DoS slowloris, DoS Slowhttptest, Bot, Infiltration, Heartbleed, Web Attack – Brute Force, Web Attack – XSS, Web Attack – SQL Injection.

Датасет CSE-CIC-IDS також включає результати аналізу мережевого трафіку за допомогою CICFlowMeter, інструменту генерації та аналізу мережевого трафіку, з позначеними потоками на основі відмітки часу, IP-адрес джерела і призначення, портів джерела та призначення, протоколів та атак (файли CSV). Створення реалістичного трафіку було головним пріоритетом у створенні цього набору даних. Було використано систему B-profile для профілювання абстрактної поведінки людських взаємодій і створення натуралістичного трафіку. Для цього набору даних було побудовано абстрактну поведінку 25 користувачів на основі протоколів HTTP, HTTPS, FTP, SSH та електронної пошти.

Сам набір даних в загальній сукупності містить у собі понад два мільйони чотириста тисяч зразків даних, кожен з яких розмічений як сутність, що належить до певного класу, який описує належність до нормального трафіку (benign) чи шкідливого (наприклад, PortScan, DDoS, Bot, і т.д.). Всього в датасеті виділяється близько п'ятнадцяти різних класів, та слід зазначити, що кількість зразків в кожному із них розподілена вкрай нерівномірно, та ті сильно відрізняється від класу до класу. Наприклад, кількість зразків у класі heartbleed складає усього одинадцять, тоді як клас goldeneye налічує близько десяти тисяч зразків. Якщо взяти усю сукупність даних то частка класу benign складає близько 84%, а решта 16% припадає на усі інші класи атак. Тож, беручи до уваги ці фактори, аби уникнути проблеми із збалансованістю даних необхідним є проведення наступних дій: об'єднання усіх класів атак в один єдиний (це також зводить проблему класифікації з мультикласової до бінарної); забезпечення міжкласового балансу. Однією із методик вирішення проблеми дизбалансованості між класами є методика субдискретизації. Сутність даного методу полягає у вибірці елементів із домінуючих класів із метою скорочення їх кількості. Стратегії субдискретизації можуть бути простими, як, наприклад, випадковий вибір групи елементів, але при цьому можливі втрати інформації у певних наборах даних. У таких випадках стратегія вибірки має передбачати в першу чергу видалення елементів, які дуже схожі на інші елементи, що залишаються в наборі даних. Для відкидання надлишкових даних було застосовано техніку субдискретизації мажоритарного (домінуючого) класу на основі центроїдів кластерів. Ця техніка полягає у створені за допомогою алгоритму кластеризації (у даному випадку метод k-середніх) кластеру домінуючого класу та у подальшому відкиданню зразків, керуючись відстанню від центроїда до положення зразку у просторі, розрахованою за евклідовою метрикою.

Увесь датасет представлений у вигляді кількох файлів у форматі .csv:

- Monday-WorkingHours.pcap_ISCX;
- Tuesday-WorkingHours.pcap_ISCX;
- Wednesday-workingHours.pcap_ISCX;
- Thursday-WorkingHours-Morning-WebAttacks.pcap_ISCX;
- Thursday-WorkingHours-Afternoon-Infilteration.pcap_ISCX;
- Friday-WorkingHours-Morning.pcap_ISCX;
- Friday-WorkingHours-Afternoon-PortScan.pcap_ISCX;
- Friday-WorkingHours-Afternoon-DDos.pcap_ISCX.

Кожен файл частково чи повністю містить дані лише про певні види мережевих атак, тож у даній роботі їх було об'єднано у один єдиний файл задля забезпечення простоти у подальших маніпуляціях, змінах та

перетвореннях даних.

Загальна кількість атрибутів, які описують кожний окремий зразок даних, становить близько 80-ти. Така велика кількість характеризуючих ознак, хоча і послугує для якнайбільш якіснішого відділення зразків між собою та класів, може виявитися надлишковою оскільки не кожна ознака може слугувати для виявлення унікальності, що буде виявляти відмінність одного класу від іншого; деякі ознаки взагалі можуть не нести ніякої корисної інформації, яка б описувала дані. Беручи до уваги велику кількість зразків даних та розмір ознакового простору буде доцільним провести відбір ознак. Тобто буде доречно створити таку підмножини ознак, яка буде значно меншою порівняно з наявною, але зведе до мінімуму втрату вагомої інформації. Зниження ознакового простору набору даних дозволить також отримати низку вагомих переваг, як наприклад: підвищення ступеню інтерпретації моделі; збільшення швидкості навчання; зменшення ймовірності приймання рішень моделлю на основі «шумів», що буде мати позитивний вплив на якість навчання.

Було розроблено програмне забезпечення на мові програмування Python для виявлення кібератак мережевого рівня моделі OSI у мережі Інтернет. Атаки виявлялися на основі їх ознак з використанням методів машинного навчання. Програмне забезпечення обробки та аналізу даних складається з декількох модулів: модуль попередньої обробки даних датасету, модуль дослідження ознакового простору мережевого трафіку та модуль використання алгоритмів машинного навчання для пошуку кібератак.

Для тренування моделі були обрані наступні алгоритми машинного навчання: наївний баєсів класифікатор, k-найближчих сусідів, дерева рішень, метод опорних векторів (SVM) з використанням гауссівського ядра, адаптивний бустинг, дерева рішень з прискоренням (градієнтний бустинг). Разом з тренуванням одразу виконувалася перехресна перевірка (з контролем) по семи блоках, для отримання більш точної оцінки узагальнюючої здатності моделі.

Після проведення тренування моделі, було здійснено тестування, якість роботи різних алгоритмів виявлення кібератак оцінювалася за декількома метриками, зокрема, точністю (1) та повнотою (2), а також F-мірою (3):

$$Precision = \frac{tp}{tp + fp}, \tag{1}$$

$$Recall = \frac{tp}{tp + fn}, \tag{2}$$

$$F = 2 \cdot \frac{precision \cdot recall}{precision + recall}. \tag{3}$$

Найбільш ефективні результати (за f-мірою) показали градієнтний бустинг (97,8%) та адаптивний бустинг (97,6%), потім усі інші: k-найближчих сусідів (96%), ядерний метод опорних векторів (95%), дерево рішень (95%) та баєсів класифікатор (77%).

Таким чином завдяки набору даних CSE-CIC-IDS можна здійснювати тестування різних алгоритмів виявлення кібератак, порівнювати якість їх роботи. Цей набір даних корисний тим, що містить різні сучасні кібератаки, а також надає достатньо великий набір даних для тренування та тестування методів, якість роботи яких треба визначити.

Список літератури

1. Chang J. (2021) "10 Cybersecurity Trends for 2022/2023: Latest Predictions You Should Know", URL: https://financesonline.com/cybersecurity-trends/

2. Branch J. (2021). "What's in a Name? Metaphors and Cybersecurity", International Organization, vol. 75, no. 1, pp. 39-70. doi:10.1017/S002081832000051X URL: https://www.cambridge.org/core/journals/international-organization/article/abs/whats-in-a-name-metaphors-and-cybersecurity/563998100A2FAF1E5DFDB5C52EC68569

3. Canadian Institute for Cybersecurity (2017) "Intrusion Detection Evaluation Dataset (CSE-CIC-IDS2017)", URL: https://www.unb.ca/cic/datasets/ids-2017.html

УДК 004.056.53

О.С. Кривда, О.К. Конопліцька-Слободенюк
krivdaolse@gmail.com, ksuha80@gmail.com
Центральноукраїнський національний технічний університет,м. Кропивницький

КУКІ ФАЙЛИ ТА БЕЗПЕКА

Кукі – кукіс – cookies - це слово ми бачимо практично щоразу, коли потрапляємо на новий сайт на нашому пристрою. Кукі — інформація про користувача, яку сервер передає браузеру під час відвідування сайту. Ці дані дозволяють ідентифікувати відвідувача без повторної авторизації.

Види куки:

- сесійні (або тимчасові) cookies - існують, поки користувач перебуває на сайті;
- постійні cookies – це файли, що зберігають інформацію про користувача та його дії, навіть якщо відвідувач покинув сайт.
- захищені cookies — передаються лише через шифроване з'єднання https.

Навіщо потрібні cookies?

Кукі потрібні, щоб зробити використання інтернету зручним. Наприклад, користувач може випадково закрити сторінку інтернет-магазину із заповненим кошиком, а потім повернутися на сайт і продовжити покупки - наново складати товари в кошик не доведеться.

Можливо, ви зараз думаєте: "А чи потрібно мені чистити кукі-файли?"

По суті, cookie безпечні, але можуть використовуватися зловмисниками для перехоплення даних при запиті браузера до сервера. З іншого боку, при накопиченні даних про попередні відвідування забивається пам'ять браузера та місце на жорсткому диску. У цьому випадку різні версії сторінок накладаються одна на одну, а заповнена пам'ять уповільнює швидкість завантаження сторінок. Тому cookie потрібно час від часу чистити.

Ви можете очистити кукі відразу для всіх ресурсів або окремих сайтів. У багатьох браузерах видалення cookie можна виконати за допомогою комбінації клавіш Ctrl+Shift+DEL.

Інший спосіб – очистити кукі через десктопну програму. Наприклад, Ccleaner.

У 2011 році Євросоюз ввів закон про cookies. Згідно з ним, кожен сайт спочатку повинен вимагати дозвіл від відвідувачів перед тим, як вносити cookies на їх пристрої.

Підсумки

Cookie у браузері – це фрагменти даних, які зберігаються на комп'ютері користувача. Вони містять дані про відвідування різних сайтів і використовуються для прискорення завантаження сторінок і збереження налаштувань користувача. При підключених кукі-файлах вам не доведеться вводити логін при кожному заході на сайт, що дуже зручно.

Список літератури

1. Що таке cookies, як це працює, і чому ми бачимо плашку про cookies на кожному сайті? [Електронний ресурс]: https://gol.ru/materials/17505-why-are-cookies-so-important

2. Что такое cookies | Словарь маркетолога Roistat[Електронний ресурс]: https://roistat.com/rublog/cookies/

3. Що таке файли cookie, як і навіщо їх чистити – Блог Netpeak Software [Електронний ресурс]: https://netpeaksoftware.com/ru/blog/chto-takoe-fayly-cookie-kak-i-zachem-ih-chistit

УДК 004.4

П.А. Семенюк, Н.В. Гунько
p.semeniuk@lutsk-ntu.com.ua, n.hunko@lutsk-ntu.com.ua
Науковий керівник: доцент кафедри комп'ютерних наук Н. В. Здолбіцька
ninazdolb@gmail.com,
Луцький національний технічний університет, м. Луцьк

ВІДТВОРЕННЯ ШИФРУВАЛЬНОЇ МАШИНИ ЕНІГМИ ЗАСОБАМИ DHTML

Засоби захисту інформації є вкрай актуальною проблемою сьогодення як для звичайних користувачів, так і промислових та державних установ. Важливо зберегти конфіденційну персональну, корпоративну або державну інформацію. Відомо безліч способів захисту передачі / збереження цінних даних [1,2]. Ці засоби постійно вдосконалюються, проходять процес модернізації [3].

Енігма – це шифрувальна машина для передачі секретних повідомлень. Ця машина має симетричну шифровку, тобто вона може шифрувати і розшифровувати повідомлення. Вона складається з клавіатури з 26 літер, також з 26 лампочок які містять літери і трьох роторів. Кожен ротор містить 26 перемішаних букв. Додатково військовий варіант Енігми має комутаційну панель, яка дозволяє з'єднати дві букви, щоб при шифруванні замість натиснутої букви передавалася її пара.

Енігма шифрує кожну букву окремо і одна й та ж буква на виході буде іншою (рис. 1). При натисканні клавіші сигнал букви проходить через комутаційну схему, потім повертається ротор і сигнал букви пропускається через ротори. Комутаційна схема міняє букву на іншу, якщо вони з'єднані між собою. У машині є три ротори, які пропускають через себе електричні сигнали. Спочатку сигнал проходить з правого ротора до рефлектора потім назад і йде до комутаційної схеми. Сигнал міняє букву на кожному кроці. Потім він йде до лампочки яка відповідає зміненій букві. Тобто проходячи через ротори і рефлектор буква змінюється 7 разів і ще 2 рази, якщо букви з'єднані на комутаційній схемі. При натисканні клавіші перший ротор повертається, якщо перший ротор зробив повне коло, тоді повертається другий. Якщо другий ротор зробив повне коло то повертається третій. Щоб розшифрувати повідомлення треба знати початковий стан роторів і комутаційної схеми і ввести зашифроване повідомлення.

Для відтворення Енігми в браузері було використано HTML, CSS, JavaScript. Ротори в коді відображені як рядки тексту на рис. 2. При шифруванні букви спочатку визначається її індекс в змінній алфавіту і по тому ж індексу переходять в перший ротор. Потім визначається індекс в змінній алфавіту нової букви і переходимо в ротор 2. Аналогічно для ротор 3. Рефлектор і комутаційна схеми змінюють букву на її пару. У рефлекторі кожна буква має собі пару, а комутаційна схема міняє букви за вибором користувача. Також в коді є окремі функції, які відповідають за з'єднання/роз'єднання букв на комутаційній схемі, підсвічування лампочки, обертання ротору при натисканні клавіші і саме натискання клавіші.

На рис. 3 відповідно зображено машину Енігми в браузері і в реальному житті. У верхній частині рисунку 3 є три ротори. Нижче від них – лампочки з буквами, нижче лампочок клавіатура і далі комутаційна схема. При натисканні клавіші повертається ротор і засвічується лампочка. Також в браузері виводиться текст, де зліва введені дані а справа вивід Енігми, що подано на рис. 4.

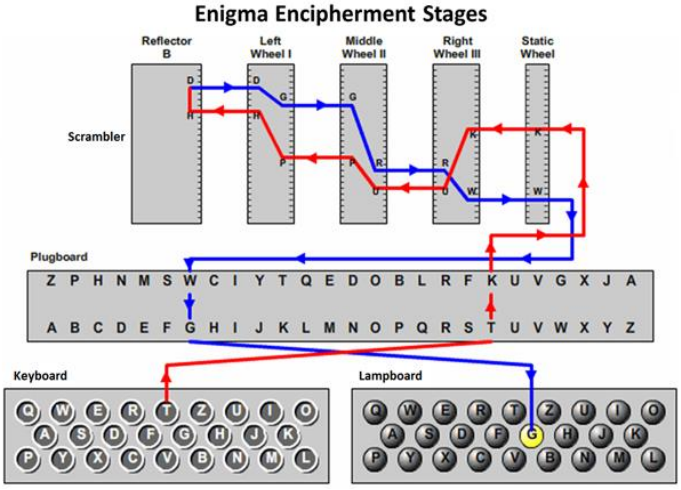

Рис. 1 Кроки шифрування Енігми

```
var alphabet = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";

var rotor1 = "EKMFLGDQVZNTOWYHXUSPAIBRCJ";
var rotor2 = "AJDKSIRUXBLHWTMCQGZNPYFVOE";
var rotor3 = "BDFHJLCPRTXVZNYEIWGAKMUSQO";
```

Рис. 2. Ротори в коді

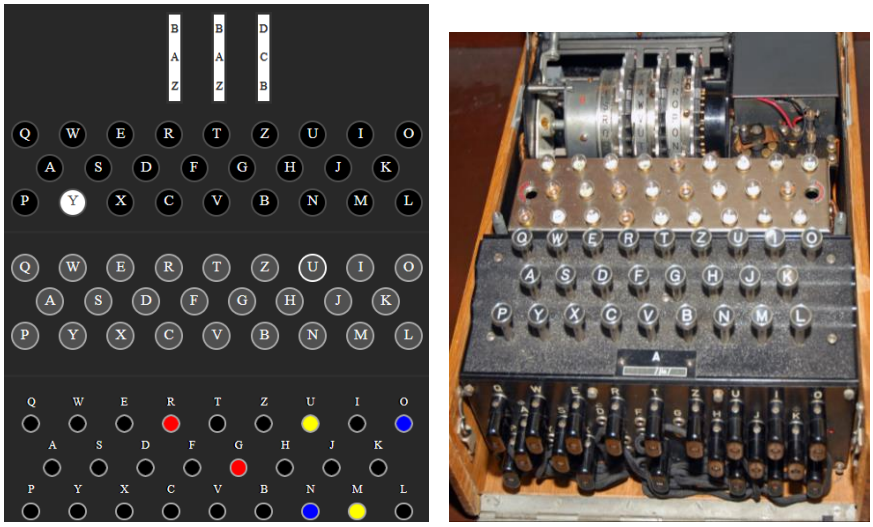

Рис. 3. Машина Енігми в браузері та в реальному житті

```
HELLO THERE ILBDA          ILBDA RTYDC HELLO
RTYDC                      THERE
```

Рис. 4. Вивід шифрованого тексту в браузері

Список літератури

1. А.О. Лукіянчук, Сучасний стан сфери інформаційної безпеки України, Матеріали IV Міжнародної науково-практичної конференції "Інформаційна безпека та комп'ютерні технології": тези доповідей, 15–16 квітня 2021 р., Кропивницький: ЦНТУ, 27-28, 2021.

2. Р.С. Дробиш, М.В. Люта Аналіз існуючих методів захисту та шифрування інформації, Інноватика в освіті, науці та бізнесі: виклики та можливості: матеріали II Всеукраїнської конференції здобувачів вищої освіти і молодих учених, м. Київ, 18 листопада 2021 року, Т. 1, Київ: КНУТД, с. 250-253, 2021.

3. О.В. Гресь, В.М. Косован, М.І., Скрипський, Г.М. Розорінов, П.М. Шпатар Аналіз алгоритмів шифрування інформації на основі хаотичних відображень, Сучасний захист інформації №3, с.49-58, 2015.

УДК 004.7.056.5(477)(047)

Білявська Ю.В.
доцент кафедри менеджменту, кандидат економічних наук, доцент
y.biliavska@knute.edu.ua
Державний торговельно-економічний університет, м. Київ

ІНФОРМАЦІЙНА БЕЗПЕКА У СФЕРІ УПРАВЛІННЯ БІЗНЕСОМ

З кожним роком підприємства все більше долучаються до сучасного інформаційного простору, імплементуючи такі диджитал-інструменти, як CRM-системи, ERP-системи, CAD-системи та системи з WEB-доступом. Інноваційні технології допомагають підприємствам бути на зв'язку зі своїми клієнтами в режимі 24/7, оптимізувати свої бізнес-процеси, вивільнити час від оперативних задач на користь стратегічних та більш креативних. Водночас цифрова ера готує і низку випробувань для систем захисту інформації та кібербезпеки сучасних підприємств. За українським законодавством, кібербезпека представляє собою захищеність життєво важливих інтересів людини і громадянина, суспільства та держави під час використання кіберпростору, за якої забезпечуються сталий розвиток інформаційного суспільства та цифрового комунікативного середовища, своєчасне виявлення, запобігання і нейтралізація реальних і потенційних загроз національній безпеці України у кіберпросторі [1]. Під час пандемії COVID-19 практично в усьому світі кібербезпека опинилася під загрозою: підприємства відчувають постійне посилення цільових атак, які стають більш витонченими та прихованими, часто з елементами фінансової мотивації.

Сьогодні впровадження режимів віддаленої роботи, дистанційного навчання, практик міжособистісного спілкування і відеоконференцій докорінно змінює світовий кіберпростір. В Україні політика щодо кібербезпеки покладається на низку державних органів: Державну службу спеціального зв'язку та захисту інформації України, Національну поліцію України, Службу безпеки України, Міністерство оборони України, Генеральний штаб Збройних Сил України, розвідувальні органи, Національний банк України. В кожному із зазначених органів діють відповідні підрозділи.

Нинішня ситуація вимагає абсолютно нових підходів до управління підприємством та його ресурсами. Успіх цих змін, а відтак, і виживання підприємства за сучасних умов, значною мірою залежить від того, наскільки гнучко організовуються бізнес-процеси та як швидко відбувається перехід до нових методів роботи. Через постійні зміни та неможливість точно спланувати цей процес, підприємства відчули наслідки кібер-ризиків більшою мірою, ніж це відбувалося раніше. Ці аргументи обумовлюють необхідність дотримання нової моделі кібербезпеки підприємства за умов COVID-реальності (табл. 1).

Таблиця 1

Кібербезпека в управлінні бізнесом за умов COVID-реальності та воєнного стану

Проблема	Характеристика	Рекомендації щодо вирішення	Платформи для роботи
Незахищене віддалене підключення до підприємства (офісу)	Не всі підприємства технічно готові до впровадження масової віддаленої роботи. ІТ-персонал під тиском часу може придбати і запропонувати не найбільш безпечні рішення	Доречно впровадити багатофакторну аутентифікацію для доступу до даних підприємства і використовувати безпечні і надійні хмарні рішення для співпраці по бізнес-процесах	Microsoft Teams, Google Hangouts Meet, LogMeIn Emergency Remote Work Kit, Cisco Webex
Розширене використання офісної техніки в особистих цілях	Підвищення ризику зараження робочої техніки вірусами або зловмисним програмним забезпеченням під час відвідування менш захищених веб-сайтів (пов'язаних із особистими інтересами)	Доречно оновлювати офісне обладнання автоматично, дотримуючись порад постачальника програмного забезпечення	Особливо важливо оновлювати веб-переглядачі та відповідне програмне забезпечення сторонніх виробників (наприклад, програми для перегляду PDF-файлів, Flash-програвачі та Java)
Спроби фішингових атак, які експлуатують тематику COVID-19 та воєнного стану	Багато злочинних кібер-угруповань змінили свою тактику на використання матеріалів COVID-19 в якості приманки. Створено велике число підроблених веб-сайтів.	Консультування співробітників щодо моніторингу новин про COVID-19 та воєнного стану в Україні та світі з використанням суто офіційних джерел інформації	Веб-сайт Міністерства охорони здоров'я України, Центру громадського здоров'я або внутрішні корпоративні ресурси підприємства

Джерело: сформовано автором

Таким чином, сучасні підприємства мають бути обізнані з підвищеними кібер-ризиками при застосуванні віддаленої праці в умовах пандемії COVID-19, про які потрібно замислитись керівникам підприємств. Пандемія змусила переглянути інфраструктуру підприємств і переорієнтуватися на актуальні інформаційні потреби користувачів та співробітників. Проведене дослідження показало, що сучасні зміни обумовлюють необхідність переосмислення стратегії та інвестування значних коштів у забезпечення надійного захисту корпоративної інформації і даних співробітників підприємств.

Гіганти світової IT-індустрії (Google, Microsoft, Amazon, Intel, Intella, IBM, Infenion, NXP, Lenovo, RSA), відомі фінансові компанії (PayPal, MasterCard, VISA, Goldman Cachs, ING) і безліч компаній меншого масштабу пропонують безпечний спосіб реєстрації в хмарі і забезпечують захист транзакцій за допомогою перевірки відбитків пальців. В цьому випадку паролі більше не спрямовуються на сервери баз даних. Шифрування виконується за допомогою відкритих ключів, при цьому ключі залишаються на пристроях. При такій схемі захисту, яка називається FIDO, неможливо відстежити взаємозв'язок між сервісами.

До переваг багатофакторної ідентифікації відноситься її здатність захистити інформацію як від внутрішніх загроз, так і від зовнішніх вторгнень. Вона заснована на спільному використанні ряду факторів аутентифікації, що значно підвищує інформаційну безпеку.

Всім відомий приклад – аутентифікація за допомогою SMS, заснована на використанні одноразового пароля. Перевага такого підходу, в порівнянні з постійним паролем, полягає в тому, що цей пароль не можна використовувати повторно. Приклад застосування біометричних пристроїв і методів аутентифікації – використання сканера відбитка пальця з підтвердженням повноважень паролем. Аналогічним чином можуть бути використані й інші біометричні аутентифікатори: обриси і розміри особи; характеристики голосу; візерунок райдужної оболонки і сітківки очей.

Існують також програмно-апаратні рішення, такі як автономні ключі для генерації одноразових паролів, зчитувачі RFID-міток, програмні та апаратні токени, Mobile ID, електронні ключі різних типів.

Як інший фактор аутентифікації може використовуватися і біометрія. Наприклад, «Match-On-Card», «Match-On-Chip» і подібні технології дозволяють замінити введення PIN-коду аналізом відбитка пальця, що додає зручності використання, оскільки не потрібно запам'ятовувати і вводити PIN-код. Також набувають популярності «платформи безпеки» або ThinkShield, які були розроблені для захисту пристроїв, особистих або конфіденційних даних від крадіжки в інтернеті.

У зв'язку із зростаючою роллю інформаційних ресурсів, а також через реальність численних загроз надзвичайно актуальні й потребують поглибленого вивчення проблеми інформаційної безпеки підприємств і організацій України. Без належного захисту інформаційного середовища підприємства неможливо забезпечити його економічну безпеку. Збільшення коштів і заходів захисту інформації, спільно з існуючими недоліками типової реалізації системи захисту інформації, збільшують навантаження на персонал підприємства й час на прийняття управлінських рішень. Водночас, менеджери підприємств сфери економіки та бізнесу відчувають певний дефіцит спеціальної літератури з питань забезпечення інформаційної безпеки як складової загальної системи економічної безпеки господарюючого суб'єкта. У зв'язку з викладеним тема дослідження представляється актуальною.

Таким чином, захист інформаційних даних необхідний будь якому підприємстві: від невеликих фірм до великих корпорацій. Захисту підлягають не тільки комп'ютерні пристрої, а й усі технічні засоби, які мають контакт з інформаційними відомостями. Будь-яка інформація, що потрапила в руки злочинців, небезпечна. У зв'язку із цим, потрібно докласти максимум зусиль для того, щоб забезпечити надійний рівень конфіденційності IT-систем.

Список літератури

1. Про основні засади забезпечення кібербезпеки України : Закон України від 5 жовтня 2017 року № 2163-VIII (В редакції Закону України від 24.10.2020 р. № 2163-VIII). URL: https://zakon.rada.gov.ua.

УДК 519.876.5, 004.942

М.О. Ларченко
urlinka2006@gmail.com
Національний університет «Чернігівська політехніка», м. Чернігів

ПРОГНОЗУВАННЯ ЗЛОЧИННОЇ ПОВЕДІНКИ ЗА ДОПОМОГОЮ ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ

Побудовані математичні моделі дають можливість вирішити питання, чи впливає певний фактор на процес прийняття рішення про скоєння злочину. Математична модель дає точну кількісну міру цього впливу.

Фахівці в галузі комп'ютерного моделювання зазначають, що злочинність є складною динамічною системою, при моделюванні та прогнозуванні якої виникає ряд проблем: 1) суттєва нестаціонарність часових рядів показників злочинності, яка обумовлена не лише нестаціонарністю параметрів, але і нестаціонарністю структури явища (нестаціонарність структури породжується якісними змінами злочинності); 2) вихідні ряди є короткими, що не дозволяє застосувати існуючі математичні методи, засновані на апараті класичної математичної статистики; 3) оскільки різні види злочинності взаємопов'язані, методи прогнозування ізольованих рядів відповідних показників не завжди є ефективними, а тому виникає необхідність розробки методів прогнозування, які засновані на застосуванні узагальнених моделей взаємопов'язаних показників злочинності [1, с. 4].

Коли потрібно вирішувати задачі прогнозування, класифікації чи управління, ефективно використовуються нейронні мережі, оскільки вони можуть бути застосовані практично в будь-якій ситуації, коли наявний зв'язок між змінними-предикторами (вхідними) та прогнозованими змінними (вихідними), навіть якщо цей зв'язок має складну природу та її важко виразити у звичайних термінах кореляцій та відмінностей між групами. Методи нейронних мереж можуть використовуватись незалежно або слугувати доповненням до традиційних методів аналізу даних. Більшість статистичних методів аналізу пов'язані з побудовою моделей, що засновані на тих чи інших припущеннях та теоретичних висновках. Нейромережний же підхід є вільним від модельних обмежень, він однаково годиться для лінійних та складних нелінійних залежностей та особливо ефективний у розвідувальному аналізі даних, коли необхідно з'ясувати, чи наявні взагалі залежності між змінними. Сила нейронних мереж полягає в їх здатності до самонавчання. Поточний стан нейрона визначається формулою:

$$u_i = \sum_{i=1}^{N} w(i,\ j)x(\ j) + b(i), \qquad (1)$$

де *x(j), j=1,2…N* – вхідні сигнали. Коефіцієнти *w(i,j)* називаються вагами синаптичних зв'язків, додатні значення яких відповідають збуджуючим синапсам, від'ємні значення – гальмуючим синапсам. Якщо *w(i,j)=0*, то говорять, що зв'язок між нейроном *i* та нейроном *j* відсутній. Величина *w(i,j)* називається пороговим значенням. Отриманий нейроном сигнал перетворюється за допомогою функції активації або передаточної функції f у вихідний сигнал

$$y_i = f(u_i) \qquad (2)$$

З математичної точки зору в моделі нейрону ми маємо нелінійне перетворення вектору *x(1), x(2)…x(N)* у вихідний сигнал y_i. Функція активації або передаточна функція f у формулі (2) – це деяка нелінійна функція, що моделює процес передавання збудження.

Таким чином, вхідний шар слугує для введення значень вхідних змінних, вихідний шар – для виведення результатів. Приховані вихідні нейрони поєднані з усіма елементами попереднього шару. Послідовність шарів та їх з'єднань називається архітектурою мережі. Перетворення сигналу проводиться наступним чином. Послідовно для кожного нейрона в мережі вираховується значення активації, береться зважена сума виходів елементів попереднього шару та віднімається порогове значення. Потім значення активації перетворюються за допомогою передаточної функції, та в результаті отримується вихід нейрону, що поступає на вхід нейронів, з якими від з'єднаний.

Нейронні мережі володіють якостями узагальнення, абстрагування та піддаються навчанню. При цьому, на думку фахівців, якість узагальнення дає можливість нейронній мережі знижувати чутливість до незначних флуктуацій вхідних сигналів. Ця особливість дуже важлива для об'єктів, які існують у реальному середовищі. Унікальність нейромережного підходу в даному випадку визначає те, що узагальнення – це результат особливостей структури, а не роботи спеціальної програми [2, с. 13].

Для формування масиву даних нами був проведений збір відомостей, який полягав у анкетному опитуванні та психологічному тестуванні («Тест Роршаха») 408 осіб, які були засуджені за скоєння різних злочинів, а також 408 осіб контрольної групи, які ніколи не були засуджені. Для цієї та іншої групи характерний один і той самий регіон переважного проживання, що дозволяє нейтралізувати дію ряду економічних показників. У обох групах однакова кількість жінок та чоловіків, а також середній вік опитаних. Зібрані таким чином відомості були нами спеціальним чином оцифровані та внесені до бази даних.

У результаті ми отримали 20 початкових змінних-предикторів, які враховувались програмою: психофізіологічні ознаки (показники психологічного тестування), демографічні показники та показники середовища, макроекономічні показники та індикатор злочинної активності (загальні статистичні показники по країні в рік скоєння останнього злочину, за який особа була засуджена). Для контрольної групи визначені ті ж самі індекси як імітація названих показників з експериментальною метою. Одна змінна-ідентифікатор визначена як номінальна вихідна та приймає два значення: засуджені, контрольна група.

Для аналізу ми використовували комп'ютерну програму STATISTICA Neural Networks (SNN).

Задача експерименту – визначити, наскільки адекватно нейромережний імітатор може відтворити ймовірнісну належність при розпізнаванні показників установленого зразка, що характеризують будь-якого індивідуума, на основі показників конкретних осіб, які раніше вчинили злочини, та даних про соціально-економічну ситуацію на той час.

Таким чином, маючи деякий набір даних, які представляють собою результати вимірювання певних характеристик об'єкта, необхідно вирішити задачу класифікації, тобто визначення, до якого з двох заданих класів належить кожний об'єкт. У пакеті SNN автоматично враховується розподіл даних на навчальну, контрольну та тестову підмножини.

Загальна ідея полягає в наступному: спочатку на вхід мережі подається навчальна вибірка з відомими результатами X та спостерігаються відгуки $Y=F(X)$. Змінюючи ваги $w(i,j)$ та значення порогу активації для кожного нейрону можливо налаштувати мережу, іншими словами, знайти якомога більш точне наближення функції F. Далі на контрольній вибірці побудована мережа екзаменується. Якщо вона пройшла іспит, її можна використовувати для проведення класифікації, що підтверджується з використанням тестової підмножини.

Очевидно, що існує певний ризик, у зв'язку із застосуванням реальних сирих даних, а не абсолютних чисел. Але цей же ризик має місце при застосуванні будь-яких математичних методів на практиці.

У результаті запуску процедури підрахунку та деякої кількості експериментів програма визначила задану кількість мереж, що найкращим чином описують представлений набір даних. Ми бачимо, що більшість ефективних мереж це багатошарові персептрони (MLP). Під персептроном розуміють одношарову чи багатошарову мережу з прямою передачею сигналу та з певними функціями активації шарів. Частіше за все використовується одношаровий персептрон. Багатошарові персептрони застосовуються для наближеної класифікації лінійно невідокремлених систем векторів. Система векторів називається лінійно відокремленою, якщо множинність точок двовимірного (багатовимірного) простору, що відповідають векторам системи, можна розбити прямою (гіперплощиною) на дві підмножини, що неперетинаються. В іншому випадку – вектори лінійно невідокремлені.

Лінійні нейронні мережі, як і персептрони, вирішують задачі класифікації лінійно відокремлених систем векторів, а також задачі лінійної апроксимації функцій [3, с. 44-46].

Ряд проведених експериментів з аналізу наявної інформації у відношенні засуджених осіб призвели до висновку що використання нейромережного методу дозволяє розпізнати образ майбутнього злочинця, тобто передбачити ймовірність скоєння особою злочину. 1. Навчена описаним або подібним чином нейронна мережа здатна спрогнозувати схильність до скоєння суспільно-небезпечних діянь у 85% спостережень. 2. Для збирання інформації, яка складає психологічний профіль досліджуваного, цілком підходить класична методика проведення «тесту Роршаха» або ж іншого психологічного експерименту, який дозволяє оцінити ряд наведених показників для конкретної особи. 3. Додавання інших соціокультурних, економічних та демографічних характеристик дозволить удосконалити якість мережі. У той же час, важливою перевагою описаного методу є його адаптивність до зміни соціокультурних груп. 4. За допомогою можливостей нейронних мереж можна також виконувати дослідження параметрів кримінологічних методик прогнозування та уточнювати їх структуру.

Список літератури

1. Гнусов Ю.В. Розробка інформаційно-аналітичного забезпечення аналізу стану та динаміки розвитку злочинності: автореф. дис. … наук. ступеня канд. техн. наук. Харків, 1998. – 18 с.

2. Новотарський М.А., Нестеренко Б.Б. Штучні нейронні мережі: обчислення. *Праці Інституту математики НАН України.* Т.51, Київ: Ін-т математики НАН України. 2004. – 408 с.

3. Кизим Н.А., Ястремская Е.Н., Сенчуков В.Ф. Нейронные сети: теория и практика применения: Монография, Х.: ИД «ИНЖЭК», 2006. – 240 с.

УДК 004.056.5:005.93]:355.1

В.С. Варава, 1 курс
Науковий керівник
доцент кафедри менеджменту, кандидат економічних наук, доцент Ю.В. Білявська
v.varava_fmtp_8_21_b_d@knute.edu.ua., biliavska@knute.edu.ua
Державний торговельно-економічний університет, м. Київ

ІМПЛЕМЕНТАЦІЯ ЗАСОБІВ КІБЕРГІГІЄНИ НА ПІДПРИЄМСТВІ ЗА УМОВ ВОЄННОГО СТАНУ

Диджиталізація, роботизація та ера інтелектуальної економіки спонукають світ переходити на новий рівень життєдіяльності, за якого керівними чинниками виробництва стають інновації та творчі досягнення людей. Суспільство увійшло в еру інновацій, де промислові роботи, 3D-друк, хмарні джерела інформації, 4G- та 5G-зв'язок, геноміка, VR-технології, розумні міста стають звичайною справою. Проте, незважаючи на умови масштабної глобалізації та стрімкого розвитку Індустрії 4.0, воєнний стан в Україні змінює звичне буденне життя: населення відчуває стурбованість і занепокоєння, відчуття безпеки та прагнення до визначеності. У той же час організовані злочинні групи безцеремонно спекулюють на невпевненості, страхах і сумнівах, пов'язаних з війною, роблячи деяких людей та підприємства вразливими до їх втручання.

Враховуючи особливості еволюційної теорії кібербезпеки, останнім часом почастішало використання терміну «кібергігієна». Цей термін немає офіційного трактування, оскільки не закріплений на законодавчому рівні. Проте саме під ним розуміється прищеплення і застосування навичок особистої інформаційної безпеки користувачам інформаційно-комунікаційної мережі Інтернет. Ключовою тематикою наукових, науково-технічних та інноваційних конференцій, симпозіумів та форумів як державного, так і міжнародного рівня є тематика інформаційної безпеки особистості, підприємств та держави. Таким чином, встановлено, що нова ера кібербезпеки вимагає цілком нових підходів до управління підприємством та його ресурсами, зокрема інформаційними. Успіх таких змін значною мірою залежить від того, як гнучко організовані бізнес-процеси на підприємстві, а також як імлементуються нові моделі та методи роботи. Через недоліки в організаційній роботі та невмінні управляти змінами підприємства відчувають наслідки кібер-ризиків, а тому вважаємо доречним дотримання кібергігієни за умов воєнного стану (табл. 1).

Таблиця 1

Засоби дотримання кібергігієни за умов воєнного стану

Напрямок/Характеристика	Рекомендації з дотримання кібергігієни
Мережа	
Конфігурація мережі Wi-Fi та налаштування безпеки. Постійно зростає потреба у формуванні безпеки та правильній конфігурації пристроїв, які підключені до мережі Wi-Fi через зростання підключених пристроїв.	Зміна паролю на Wi-Fi – роутер замість встановленого за замовчуванням; створення паролю для Wi-Fi із символів та літер; обмеження кількості пристроїв, які можуть бути підключені до мережі Wi-Fi; вибір унікального імені для мережі Wi-Fi; увімкнення брандмауеру; здійснювати регулярне оновлення операційної системи Wi-Fi-роутера; налагоджувати шифрування даних у мережі Wi-Fi (WPA2 та WPA3).
Ризики під час використання публічних мереж. За безкоштовних та відкритих (публічних) умов використання мереж Wi-Fi для хакерів спрощується можливість перебування між точкою доступу Wi-Fi та пристроями.	Не використовувати відкриті (публічні) мережі Wi-Fi для бізнесу, а також обмежити їх використання з метою розваг; перевірка www-адрес та роботи HTTPS; при можливості застосовувати VPN з'єднання, що захистить від перехоплення даних.
Віддалена робота та доступ до мережі	
Аутентифікація, паролі та дані користувача. Хакери посилюють можливості доступу до різноманітних систем, атому необхідно дотримуватися захисту даних.	Регулярно змінювати паролі до особистих облікових записів, а також до програм у робочому середовищі; створювати правильний пароль для Wi-Fi із символами та цифрами, а також не менше 10 знаків; використовувати багатофакторну автентифікацію за допомогою програми, SMS або дзвінка.
Віддалені сесії. До таких сесій належать VPN та робота вдома.	Під час віддалених сеансів зв'язку з IT-інфраструктурою бізнесу доречно застосовувати VPN з'єднання; дотримуватися використання багатофакторної автентифікації; без додаткового захисту не користуватися віддаленим підключенням (Team Viewer, Remote Desktop Protocol).
Електронна пошта та комунікація	

Безпека. Фішинг електронної пошти. Електронна пошта найлегший спосіб наживи для зловмисників.	Уважно читати вміст електронних листів, а підозрілі файли та листи не відкривати; відкривати листи перевірених та відомих відправників; застосовувати антивірусні програми за допомогою яких можна перевірити електронну пошту на наявність вірусів.
Устаткування для роботи та особистого використання	
Розмежування надмірної експлуатації пристроїв. В умовах онлайн складно розмежувати пристрої для особистого використання та для бізнесу, що провокує кібер-ризики.	Не використовувати робочі пристрої для особистих цілей і навпаки, особисті пристрої для роботи; дотримуватися політики конфіденційності підприємства; встановлювати ліцензійне програмне забезпечення, а також погоджувати це з IT-фахівцем; на робочому пристрої не зберігати особисту інформацію (документи, файли, результати аналізів чи інші конфіденційні дані); не синхронізувати дані браузера між особистими та робочими пристроями.
Мобільні та смарт-пристрої	
Смарт-пристрої та дані користувача. Віруси, підозрілі програми, ПЗ без оновлень створюють загрозу безпеці та роблять дані доступними для кіберзлочинців.	Посилено використовувати антивірусні програми для захисту; перевіряти та контролювати наданий доступ до програм (наприклад, до контактів, камери чи геолокації); уважно аналізувати нетипові запити у соціальних мережах; не встановлювати програми з невідомих джерел; оновлювати програмне забезпечення; використовувати надійні паролі та біометричний захист
Підключення смарт-пристроїв до Інтернету, Інтернет речей (IoT). Близько 80% IoT-пристроїв не захищено від кібератак.	Обирати правильний пристрій; вивчати ліцензію та життєвий цикл пристрою в цілому; використовувати IoT-пристрої лише у закритій (приватній) мережі; регулярно оновлювати пристрій та паролі.
Робота з даними	
Зберігання даних. Хоча хмара є одним з найефективніших і найсучасніших видів сховищ даних, необхідно переконатися в її кібербезпеці.	Доступ до хмари дозволяти лише відомим користувачам та пристроям; не застосовувати відкриту мережу Wi-Fi для роботи у хмарі; використовувати багатофакторну автентифікацію для доступу до хмарних ресурсів даних; використовувати шифрування даних як для їх передачі, так і для зберігання.
Кіберзнання проти кібершахрайства	
Активність кіберзлочинців розширюється. Нинішня ситуація, викликана COVID-19 і умовами війни, та віддалена робота створюють сприятливі умови для всіх видів кібершахрайства та атак.	Створювати резервні копії; не зберігати конфіденційну інформацію в електронній пошті, а також у незахищеній хмарі або на пристрої, який належним чином не оснащено кіберзахистом; слідкувати за періодичним оновленням паролів та антивірусних програм; бути уважним до експлуатації електронної пошти; удосконалювати знання із кібербезпеки та кібергігієни.

Джерело: сформовано автором

Таким чином, розвиток цифровізації усіх сфер життя сприяє тому, що підприємства та звичайні громадяни усе більше будуть потерпати від зростання кіберзлочинності, наприклад, у процесі придбання товарів чи банківських операцій в мережі Інтернет. В процесі аналізу стану та тенденцій цифрових технологій, як нової ери кібербезпеки, сформовано ключові напрями захисту підприємства від кіберзагроз за умов воєнного стану. Щодня набуває актуальності зміна стереотипів у суспільстві, що особисті дані чи особа нікому не цікаві, доречно провести навчання фахівців користуватися захищеними протоколами передачі інформації, використовувати захищені інформаційні системи для роботи.

УДК 004.056.53

В.Є. Черновол, О.К. Конопліцька-Слободенюк
blackshox.mlp@gmail.com, ksuha80@gmail.com
Центральноукраїнський національний технічний університет, м. Кропивницький

БЕЗПЕКА КОМП'ЮТЕРНИХ МЕРЕЖ ТА ІНТЕРНЕТ

На сьогоднішній день, вже майже будь-яке підприємство не може існувати без Інтернету. Через це виникають загрози, які здатні завдати шкоди підприємству. В інтернеті є багато загроз, для яких є ціллю завдати шкоди - ураження комп'ютерів троянами, вірусами, спам-рекламою, і тому подібним загрозливим ПЗ, які здатні привести до викрадення конфіденційної інформації, руйнування цінної інформації, виведення з ладу комп'ютерних систем, стрімкого росту витрат на Інтернет.

Основні загрози

Зараз, з'явилася LTE мережа, яка стала ключовою технологією, що дозволяє Інтернет-юзерам побачити дані майже без затримки, але неможливо закривати очі на те, що в LTE існує багато недоліків, завдяки тому що це високошвидкісний стандарт, він швидше розповсюджує загрози та є вразливим до них.

Перша загроза - це DDoS на мережу.

Друга загроза - вірусні атаки.

Третя загроза - атаки на сторонні сервіси.

Під кіберінцидентами визначають інциденти, які полягатимуть в реалізації певної небезпеки та порушенні визначеного рівня безпечності стільникових мереж.

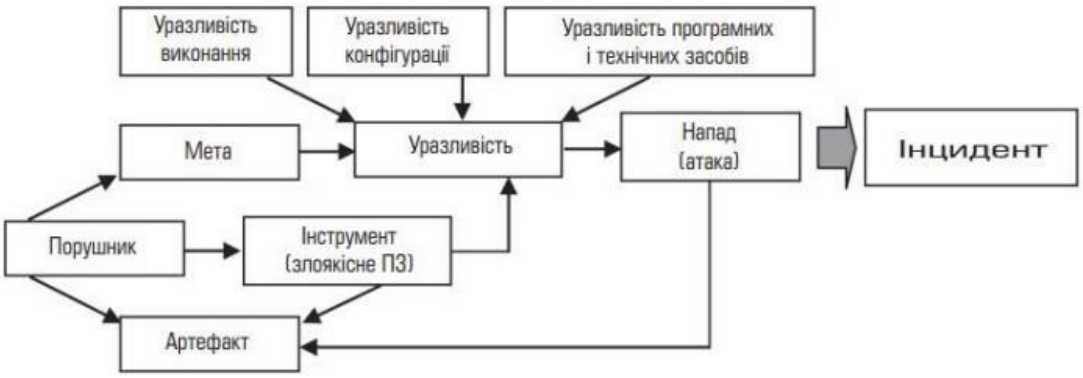

Рис. 1. Таблиця інцедентів

Побудова системи відклику на кіберінциденти - існує єдина служба для Інтернет-юзерів інформаційних систем CERT. Ця служба забезпечує пошук та обробку інформації про інтернет інциденти, дає консультації та технічну допомогу користувачам з ціллю уникнути вірусів. Профіль CERT знижує рівень загроз для юзерів Інтернету.

До компетенції служби входить обробка наступних інцедентів з метою їх виявлення і нейтралізації:

- DDoS сервера, з ціллю пошкодження їх та знищення даних записаних на них;
- незаконне проникнення до інформаційних ресурсів;
- розповсюдження небезпечного ПЗ, та спаму;
- вивчення національних інформаційних мереж.

Кожна людина або підприємство має загрозу схопити вірус, троян, або стати жертвою хакерів. Через те що LTE збільшує швидкість поширення шкідливих програм (оскільки цей стандарт сам по собі є високошвидкісним), виникає необхідність створення системи управління кіберінцидентами.

Список літератури

1. В. Гнатюк. «Методи обробки кіберінцидентів в інформаційно-телекомунікаційних системах». 2017.URL:http://dspace.nau.edu.ua/bitstream/NAU/2 8984/3/diss.pdf.

2.Служба реагування на комп'ютерні інциденти. URL: http://sts.kz/ru/kzcert.

3. A Distributed Architecture Delivering Scalability and Performance. URL: http://www.netfor en sics. com/architecture.html

УДК 004.652.8

Є.Р. Ковтун, С.В. Науменко, І.О. Розломій
kovtun.yelyzaveta1119@vu.cdu.edu.ua, sergey.naumenko@gmail.com, inna-roz@ukr.net
Черкаський національний університет імені Богдана Хмельницького, м.Черкаси

БЕЗПЕЧНЕ ПРОЄКТУВАННЯ БАЗ ДАНИХ З ВИКОРИСТАННЯМ ORM

Безпека програмного забезпечення складається з багатьох компонент. Для створення безпечного продукту важливо забезпечити захист не тільки від мережевих атак або можливості витоку логінів користувачів, що робить можливим підбір пароля перебором, але й украй важливо спроєктувати і реалізувати надійне і відмовостійке рішення, яке буде вести себе коректно і передбачувано в будь-якій ситуації. Правильно спроектована база даних (БД) є важливою складовою надійного рішення. Існують практики, використання яких звільняє від необхідності проектування бази даних. Наприклад, все частіше застосовується ORM (Object Related Mapping) підхід.

ORM дозволяє розробникам, не залежати від особливостей пристрої використовуваної системи керування базою даних (СКБД), робота с БД зводиться до роботи з даними об'єктно-орієнтованих класів, тобто з класами та їх екземплярами. Такий підхід здійснює перетворення даних класів, в дані придатні для зберігання в системі керування базою даних і навпаки, зникає необхідність для написання SQL-коду для взаємодії з СКБД.

Завдання проєктування бази даних зводиться до задачі виділення класів в відповідності з принципами об'єктно-орієнтованого програмування, відносин між ними і основних методів. Операції над цими класами будуть надалі за допомогою засобів ORM транслюватись на базу даних, саме тому дуже важливо правильно виділити основні класи та методи, щоб виключити можливість уразливості БД. Розглянемо, як на етапі виділення класів забезпечити побудову найбільш безпечного результату.

Основними загрозами під час етапу виділення класів є допущення так званих антипаттернів, що надають негативний вплив на ефективність, коректність роботи та відмовостійкість системи.

Антипаттерни представляють собою часто використовувані розробниками шаблони, застосування яких має негативний вплив на роботу програмного продукту.

Для виключення можливості появи антипаттернів на етапі проектування класів пропонується створити програмне рішення, яке б автоматизувало процес пошуку помилок. Такий засіб дозволить значно, скоротити час проектування і виключити людський фактор, що призводить до пропускання помилок.

При розгляді діаграми класів, що зображує загальну структуру ієрархії класів з обліком методів, полів та взаємозв'язків між ними, на перший погляд, здається логічним використання теорії графів для задачі пошуку антишаблонів. Справді, діаграма класів – це орієнтований граф, де вершинами – є класи, а ребрами – зв'язку між ними.

Однак, використання графів для формалізації класів, обмежує коло антипаттернів, які можна ідентифікувати. Це пов'язано з тим, що графи не дозволяють формалізувати поля і методи. Таким чином застосування теорії графів для формалізації класів підходить для випадку коли автоматизація пошуку використовується в основним для виявлення структурних помилок (наприклад, пошук зациклювання, ромбовидного успадкування, наявності та незв'язних графів).

Таким чином, необхідно знайти спосіб формалізації класів, найбільш повно відбиваючий всі аспекти класів і відношень між ними. Онтологія – це детальна і всеосяжна формалізація деякій предметної області з допомогою концептуальної схеми, чим і є схема бази даних [1].

В даному випадку доцільно застосувати описову або дескрипційну логіку для цілей формалізації діаграми класів. Дескрипційні логіки широко використовуються для опису онтологій та розроблялися в основному для цих цілей.

Дескрипційні логіки оперують поняттями «концепт» і «роль», відповідними в інших розділах математичної логіки, поняттям, «одномісний предикат» і «двомісний предикат». Інтуїтивно концепти використовуються для опису класів деяких об'єктів. Ролі використовуються для опису двомісних відношень між об'єктами. За допомогою мови дескрипційної логіки можна формулювати твердження загального вигляду – про класи взагалі і приватного вигляду – про конкретні об'єкти [2].

Дескрипційна логіка – це ціле сімейство мов представлення знань, описуючі поняття предметної області в недвозначному, формалізованому вигляді. Логіка ALC (Attributive Language with Complement) є однією з базових і має два необхідні розширення Q і I [3]. Розширення Q дозволяє визначити якісні обмеження кардинальності ролей, а I – зворотні ролі. Дані розширення необхідні для формалізації діаграми класів, тому буде використовуватись логіка ALCQI.

Для того, щоб задати якусь логіку, необхідно задати її синтаксис та семантику. Синтаксис визначає, які вирази (концепти, ролі, аксіоми тощо) вважаються правильно побудованими в даній логіці. Семантика вказує, як інтерпретувати ці висловлювання, тобто надає їм формального сенсу.

Семантика дескрипційної логіки задається шляхом інтерпретації її атомарних концептів як множини об'єктів, які вибираються з фіксованого множини (домена), і атомарних ролей у вигляді множини пар об'єктів, тобто. бінарні відносини на домені.

Інтерпретація – це непорожня множина(домен) та функція, що інтерпретує. Ця функція зіставляє кожному атомарному концепту деяке підмножину, а для кожної атомарної ролі – деяку підмножину. Якщо пара індивідів належить інтерпретації деякій ролі, то говорять, що індивід є послідовником індивіда.

Розширення відповідає за зворотні ролі: якщо роль, то також є роллю, що означає звернення бінарного відношення. Інтерпретація ролі – зворотної R, визначається аналогічно до інтерпретації R, тільки пара індивідів змінюється місцями.

Зазначимо, що логіку ALCQI можна перевести на логіку предикатів першого порядку з двома змінними. Дана логіка можна розв'язати, це дозволяє переносити результати про дозвіл, обчислювальну складність і вирішальні алгоритми з області логіки предикатів в область дескрипційних логік. Для діаграм класів ми маємо справу тільки з TBox, тому що об'єкти у явному вигляді у діаграмах класів не фігурують.

Таким чином, задавши синтаксис та семантику дескрипційної логіки, можна формалізувати будь-яку діаграму класів [4 –7].

Кожен антипаттерн проєктування є деяким шаблоном. Подання діаграми класів у вигляді набору тверджень дескрипційної логіки є текстовим рядком. Таким чином, висловивши антишаблон у вигляді деякого регулярного виразу мовою дескрипційної логіки ALCQI, можна звести пошук шаблону до зіставлення текстового рядка на наявність регулярного виразу.

Множинне успадкування полягає в тому, що певний клас має кілька предків. Ця помилка є критичною для мов програмування, які не підтримують подібну структуру, наприклад, Java. Також при множинному наслідуванні може виникнути перекриття імен методів класів-предків [7]. Виникнення такої ситуації призводить до помилки компілятора програми, оскільки він не дозволяє визначити, який метод із класів-предків необхідно використовувати. Тобто, множинне успадкування безпосередньо впливає на коректність роботи програмного засобу, а отже, на його безпеку.

Для знаходження випадку множинного наслідування в загальному випадку для деякого класу необхідно перевірити кількість рядків, що кодують ставлення наслідування. І якщо кількість рядків, що задовольняють подібному регулярному виразу більше одного, то можна зробити висновок про наявність кількох предків класу.

Аналогічно, можна представити інші антипаттерни, які допускаються при проектуванні класів, у вигляді регулярних виразів та реалізувати алгоритми їх знаходження у рядку, що кодує класи та їх взаємодію у вигляді набору тверджень дескрипційної логіки ALCQI.

Список літератури

1. Токарчук А.М. (2010) Применение средств OR для разработки безопасных веб-приложений. Безопасность информационных технологий. №1(17). 113–115.
2. Halpin. (2015) Object-Role Modeling Fundamentals: A Practical Guide to Data Modeling with ORM. Technics Publications LLC. 192 p.
3. Oleynik P.P., Salibekyan S.M, Kuznetsov N.V. (2015) Implementation patterns of object static models for database applications: classical ORM-patterns and object-attribute approach. International journal of applied engineering research. №24(10), 2015. 413–417.
4. Colin J. Neill, Philip A. (2005) Antipatterns: Identification, Refactoring and Management. CRC 336 p.
5. Neill, Colin J. (2012) Antipatterns in Systems Engineering: An Opening Trio. INCOSE International Symposium. №1(22). 1233–1245.
6. D. Berardi, D, Calvanese, G. D. Giacomo. (2005) Reasoning on UML Class Diagram. Artificial Intelligence. №2(168). 70–118.
7. M. Sergievskiy. (2017) Description Logic Application for UML Class Diagrams Optimization. International Journal of Advanced Computer Science and Applications. №1(8). 268–272.

СЕКЦІЯ 2. ПРОГРАМУВАННЯ ТА ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНІ ТЕХНОЛОГІЇ

УДК 004.8+614.8+331.452

К.М. Марченко, О.В. Оришака, А.К. Марченко
e-mail: k_marchenko@i.ua, oryhsaka@gmail.com
Центральноукраїнський національний технічний університет, м. Кропивницький

ПРОБЛЕМИ ВИКОРИСТАННЯ ШТУЧНОГО ІНТЕЛЕКТУ В КОМП'ЮТЕРНИХ СИСТЕМАХ

В інформаційному суспільстві спостерігається експонентне зростання інтенсивності процесів інформаційного обміну та обробки даних, що викликає необхідність використання потужних комп'ютерних систем. До таких систем пред'являють такі вимоги, як висока швидкодія, великий обсяг пам'яті, здатність обробляти велику кількість транзакцій одночасно, підвищена надійність.

Надійність, яка є однією з головних вимог до комп'ютерних систем, адже від рівня надійності системи залежить, наскільки відповідальні інформаційні процеси їй можна довірити. Оскільки абсолютна надійність комп'ютерних систем та результатів інформаційних процесів, які у них виконуються, не може бути забезпечена, задачею досліджень є визначення критичних областей, де такі помилки та збої в роботі не допустимі.

Розробникам складних комп'ютерних систем та програмного забезпечення важко уникнути помилок. Іноді помилки у програмному забезпеченні викликають справжні катастрофи [1].

З кінця XX сторіччя інформаційні та комп'ютерні системи розвивалися в напрямку застосування підтримки прийняття рішень, експертних систем і систем штучного інтелекту. Головними проблемами впровадження штучного інтелекту в комп'ютерні системи є неможливість передбачити всі реальні ситуації та запрограмувати поведінку машини адекватно до них, недостатня надійність та помилки у програмному забезпеченні. Вхідні дані, на основі яких навчається штучний інтелект можуть бути хибними. Ці недоліки за час використання систем штучного інтелекту привели до безлічі інцендентів, в тому числі летального характеру. Аналіз вибірки повідомлень про помилки штучного інтелекту дозволив визначити, до яких сфер відносяться критичні помилки, тобто де застосування систем штучного інтелекту пов'язане із значним ризиком. Це такі галузі як медицина, військова справа, транспорт, виробництво, де співпрацюють люди та роботизовані системи, небезпечні виробництва, ядерна енергетика, соціальне управління, судові процеси тощо.

Дослідники стверджують, що штучний інтелект є не що інше, як програма, заснована на статистиці. Точність роботи таких програм не перевищує 95%. Отже, при такому рівні похибок не варто довіряти таким системам людські життя [2]. Обов'язком розробників комп'ютерних систем із штучним інтелектом є попередити закладення в алгоритми процесів, які можуть зашкодити людині [3]. Алгоритми вчаться більш адекватно моделювати реальні ситуації але вони ніколи не стануть досконалими та безпомилковими. Питання про допустимий поріг помилок, ціни за помилку та економію від заміни людини штучним інтелектом стоятиме завжди. У найближчому майбутньому людина, як і раніше, прийматиме критичні рішення, хоч би якою розумною не була система штучного інтелекту [3].

Не вирішеним юридично залишається питання відповідальності за помилки штучного інтелекту: хто і в яких випадках має за них відповідати – розробник чи особа, яка ним користується, чи сам штучний інтелект. На сьогодні не існує жодних законодавчих норм, які б регулювали саме штучний інтелект [4].

Штучний інтелект, що використовується в критично важливих інфраструктурах, у галузях, пов'язаних із здоров'ям та життям людей, відноситься до категорії високого ризику. Засновуючись на проведеному аналізі автори не рекомендують використовувати штучний інтелект у сферах, пов'язаних з безпекою, здоров'ям та життям людини, особливо великих людських колективів.

Список літератури

1. ИИ посоветовал пациенту умереть: самые крупные ошибки машинного обучения. Стаття на сайті https://hightech.fm/2021/09/02/ai-failures
2. «Плохо обученный искусственный интеллект опаснее восстания машин». Стаття на сайті https://www.hse.ru/news/expertise/506082229.html
3. Жуков Л. Почему люди в ближайшем будущем не смогут полностью довериться ИИ. Стаття на сайті https://trends.rbc.ru/trends/industry/5fb52daf9a7947234c4d28d3
4. Андраш Ю. Кто несет ответственность за преступления искусственного интеллекта? Стаття на сайті https://www.lansky.at/ru/newsroom/news-media/zhurnal-lgp-news-022021/kto-neset-otvetstvennost-za-prestuplenija/#
.

УДК 004.056

Т.В. Смірнова, О.А. Смірнов, С.А. Смірнов
sm.tetyana@gmail.com., dr.SmirnovOA@gmail.com, smirnov.ser.81@gmail.com
.Центральноукраїнський національний технічний університет, м. Кропивницький

ДОСЛІДЖЕННЯ СТАТИСТИЧНОЇ СТІЙКОСТІ ТА ШВИДКІСНИХ ХАРАКТЕРИСТИК ЗАПРОПОНОВАНОЇ ФУНКЦІЇ ГЕШУВАННЯ

На сучасному етапі розвитку хмарних технологій, існує завдання захисту даних, які зберігаються у відповідних інформаційно-комунікаційних системах. Це підтверджується подіями початку 2022 року в України, коли було реалізовано ряд кібератак на хмарні ресурси державних установ. Під час масованої кібератаки, яка почалася у ніч з 13-го на 14 січня 2022 року, постраждали 22 сайти органів державної влади. Шести сайтам було завдано значної шкоди, 70 – відключено за вказівкою Держспецзв'язку та Служби безпеки України [1]. Починаючи з другої половини дня 15 лютого 2022 року спостерігалась потужна DDoS-атака на низку інформаційних ресурсів України. Зокрема, було зафіксовано перебої в роботі веб-сервісів Приватбанку та Ощадбанку. Також атаки зазнали сайти Міністерства оборони та Збройних Сил України [2]. Аналіз останніх подій показав, що хмарні сервіси потребують розроблення нових або удосконалення існуючих механізмів захисту інформації. Одними з таких механізмів є програмні модулі криптографічного захисту даних, у яких необхідно реалізовувати вибір стійких методів шифрування та гешування, а також синхронізацію секретного ключа. У якості зазначених процедур можуть використовуватись відомі криптографічні методи і засоби, стійкі до лінійного, диференціального, алгебраїчного, квантового та інших відомих видів криптоаналізу.

Сьогодні серед множини методів захисту інформації особливе місце займають криптографічні методи [3]. З огляду на результати проведеного аналізу, прототипом було обрано розглянутий модуль MTProto Mobile Protocol v.1.0 [4], порівняно з яким було змінено наступне [5]:

1. Змінені вхідні та вихідні дані. На вході приймаються і обробляються наступні дані: повідомлення M, інформацію про ідентифікатор користувача та ідентифікатор сесії S, інформацію про час відправлення і довжину повідомлення ID та порядковий номер повідомлення PD. На виході тільки отримуємо $mHash$ – геш значення DB ($DB = (S, ID, M)$) та $EncP$ – зашифроване повідомлення P [5].

2. Замість використання геш функції SHA-1 введено використання певної криптостійкої геш функції F_{hash}. Слід зауважити, що у якості F_{hash} може бути використана функція гешування, що побудована на основі одного із методів [7-9, 16].

3. Замість використання блокового шифру AES введено використання функції F_{enc}. Слід зауважити, що у якості F_{enc} може бути використаний певний криптостійкий алгоритм шифрування, побудований на основі блокових, потокових шифрів чи геш функцій тощо [5].

4. У якості *authKey*, введено використання заздалегідь узгодженого секретного ключа користувачів, наприклад за допомогою протоколів асиметричної криптографії [5].

Для використання цього модуля на практиці потрібно визначитись з функціями гешування F_{hash} та шифрування F_{enc}.

Статистичним портретом генератора є матриця розмірністю $m \times q$, де m – кількість двійкових послідовностей, що перевіряються, а q – кількість статистичних тестів, використовуваних для тестування кожної послідовності. Елементи матриці $P_{ij} \in [0,1]$, де $i = \overline{1, m}$ і $j = \overline{1, q}$ є значеннями ймовірності, яка отримана в результаті тестування i-ої послідовності j-им тестом.

1. Згідно з отриманим статистичним портретом визначають долю послідовностей, які пройшли кожен статистичний тест. Для цього задають рівень значущості $\alpha \in [0.001,\ 0.01]$ і виконують підрахунок значень ймовірності, що перевищують встановлений рівень значущості α для кожного з q тестів, тобто визначають коефіцієнт:

$$r_j = \frac{\#\left\{P_{ij} \geq \alpha \middle| i = \overline{1, m}\right\}}{m}$$

В результаті формується вектор коефіцієнтів $\mathbf{R} = \{r_1, r_2, r_q\}$, елементи якого характеризують, у відсотках, проходження послідовності S_i всіх статистичних тестів.

Правило 1. Передбачається, що генератор G пройшов тестування по j-у тесту, якщо значення коефіцієнта r_j знаходиться в межах довірчого інтервалу $[r_{\max}, r_{\min}]$. Кордони довірчого інтервалу визначаються відповідно до вираження:

$$r_{\max(\min)} = \hat{p} \pm 3\sqrt{\frac{\hat{p}(1-\hat{p})}{m}}\text{, де } \hat{p} = 1-\alpha\text{.}$$

2. Виробляється статистичний аналіз статистичного портрета. Набутих значень ймовірності P_{ij} підкоряються рівноймовірному закону розподілу на інтервалі $[0, 1]$. Для вектора-стовпця статистичного портрета будується гістограма частот F_k попадання значень P_{ij} в кожен з $k = \overline{1,10}$ підінтервалів, на які розбитий інтервал $[0, 1]$. Рівноймовірність розподілу значень ймовірності P_{ij}, перевіряється з використанням критерію χ^2. Для цього розраховується статистика вигляду:

$$\chi_j^2 = \sum_{k=1}^{10} \frac{\left(F_k - m/10\right)^2}{m/10},$$

яка підкоряється розподілу χ^2 з дев'ятьма мірами свободи.

Правило 2. Передбачається, що генератор G пройшов тестування по j-у тесту, якщо виконується умова $P\left(\chi_j^2\right) > 0.0001$.

3. Передостаннє рішення приймають відповідно до правила: передбачається, що генератор G пройшов статистичне тестування пакетом NIST STS, якщо значення коефіцієнтів r_j для всіх $j = \overline{1,q}$ знаходяться в межах довірчого інтервалу $[r_{\min}, r_{\max}]$ і виконується умова $P\left(\chi_j^2\right) > 0.0001$ для всіх $j = \overline{1,q}$.

Висновки

Удосконалено модуль криптографічного захисту інформації, який за рахунок фіксування інформації про ідентифікатор користувача, ідентифікатор сесії, час відправлення, довжину повідомлення та його порядковий номер, а також використання нової процедури формування сеансового ключа для шифрування, дозволяє забезпечити конфіденційність і цілісність даних в ІКСМ управління технологічними процесами. Для ефективного використання цього модуля важливим є вибір криптостійких методів шифрування F_{enc} та гешування F_{hash}, а також синхронізація секретного ключа *authKey*. У якості функцій F_{enc} та F_{hash} можуть бути використані криптоалгоритми, стійкі до лінійного, диференціального, алгебраїчного, квантового та інших відомих видів криптоаналізу. Проведено експериментальне дослідження, що підтвердило криптостійкість удосконаленого алгоритму до лінійного та диференціального криптоаналізу та дозволило визначити, що швидкість криптографічної обробки даних удосконаленим методом з використанням розробленої функції шифрування є у 1,98 разів вищою у порівнянні з аналогами, а швидкість генерування ключів є вищою відповідно у 1,17 разів в порівнянні з генераторами, що використовуються зокрема в обраному прототипі.

Список літератури

1. https://www.kmu.gov.ua/news/vid-kiberataki-14-sichnya-postrazhdali-22-derzhavnih-organi-derzhspeczvyazku

2. https://www.kmu.gov.ua/news/shchodo-kiberataki-na-sajti-vijskovih-struktur-ta-derzhavnih-bankiv

3. R. Oppliger, Cryptography 101: From Theory to Practice, Artech, 2021.

4. Job J, Naresh V and K. Chandrasekaran, "A modified secure version of the Telegram protocol (MTProto)", 2015 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), 2015, pp. 1-6,

5. Смірнова Т.В., Гнатюк С.О., Бердибаєв Р.Ш., Бурмак Ю.А., Оспанова Д.М., «Удосконалений модуль криптографічного захисту інформації в сучасних інформаційно-комунікаційних системах та мережах». Кібербезпека: освіта, наука, техніка. № 2(14). С. 176-185. 2021.

УДК 629.7.02+004.056.5

О.О. Майданик, Є.В. Мелешко, С.В. Шимко, О.Г. Собінов
maidanyksmail@gmail.com, elismeleshko@gmail.com, shymko.sv@meta.ua, sagcob14@gmail.com
Центральноукраїнський національний технічний університет, м. Кропивницький

РОЗРОБКА АПАРАТНО-ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОБМІНУ ДАНИМИ МІЖ КВАДРОКОПТЕРОМ ТА УПРАВЛЯЮЧИМ ПРИСТРОЄМ

В даній роботі було створено апаратні пристрої з бездротовим каналом зв'язку та програмне забезпечення для шифрування даних на мікроконтролерах.

Для реалізації програмного забезпечення, яке повинне виконуватися на мікроконтролері, було обрано середовище розробки STM32CubeIDE від компанії STMicroelectronics – вдосконалену платформу розробки C/C++ [1] з периферійною конфігурацією, генерацією коду, складанням коду та налагодженням для мікроконтролерів та мікропроцесорів STM32.

Для побудови розроблюваної системи створено робочі макети пристроїв. Для створення макету обрано модуль на основі мікроконтролера STM32F103C8T6 [2] (рис. 1). Модуль має:

- Виводи портів A0-A12, B0-B1, B3-B15, C13-C15.
- Micro-USB через який можна живити плату. На платі присутній стабілізатор напруги на 3.3В. Живлення 3.3В або 5В можна подавати на відповідні виводи на платі.
- Кнопку RESET.
- Два перемички BOOT0 і BOOT1. Використовувати під час прошивки через UART.
- Два кварци 8МГц і 32768Гц. У мікроконтролера є множник частоти, тому на кварці 8 МГц можливо досягти максимальної частоти контролера 72Мгц.
- Два світлодіоди. PWR - сигналізує про подачу живлення. PC13 - підключений до виходу C13.

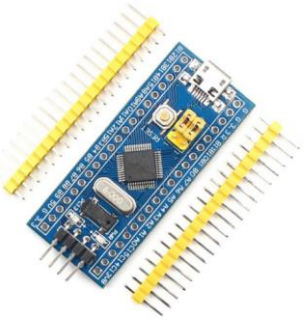

Рис. 1. Модуль на основі мікроконтролера STM32F103C8T6

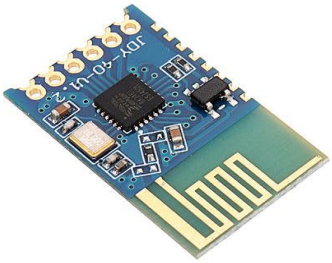

Рис. 2. Радіомодуль JDY-40

Важливою частиною пристрою є радіомодуль. Вирішено використовувати радіомодуль JDY-40. Так як цей модуль має низьку ціну та відносно просте керування.

Модуль JDY-40 [3] дуже компактний і його можна живити від будь-якого літій-іонного акумулятора на 3,7 В, що дозволяє вбудувати його в будь-які моделі. Радіус дії радіомодуля до 120 метрів в прямій видимості, що для більшості пристроїв цілком достатньо, якщо потрібна велика дальність то можна застосувати зовнішню антену.

Для обміну даними з мікроконтролером модуля JDY-40 здійснюється по UART інтерфейсу з максимальною швидкістю 19200 біт. Для підключення периферії до модулю є 8 портів введення-виведення. На рис. 2 зображено модуль зв'язку JDY-40.

На основі модулів JDY-40 та макетної плати розроблено робочі макети пристроїв. На рис. 3 зображено макети пристроїв для передачі даних по радіоканалу.

На рис. 3 присутні програматори та логічний аналізатор, за допомогою яких здійснювалася наладка схеми. Обидва пристрої підключені до ПК, на якому відкрито два термінали. В кожному із терміналів необхідно обрати відповідний номер COM порту та відкрити його. Після відкриття COM порту можна передавати данні від ПК до мікроконтролера, який в свою чергу передасть їх на радіомодуль.

Для шифрування даних було використано шифр Вернама. А у якості ключа шифрування псевдовипадкову послідовність, що генерувалася за допомогою математичного більярду Сіная.

При використанні запропонованого методу шифрування через канал зв'язку не передається ніякої інформації про ключі. Початкові параметри ключа записуються на квадрокоптер перед стартом, а ключі генеруються на кожному із пристроїв самостійно на основі початкових параметрів.

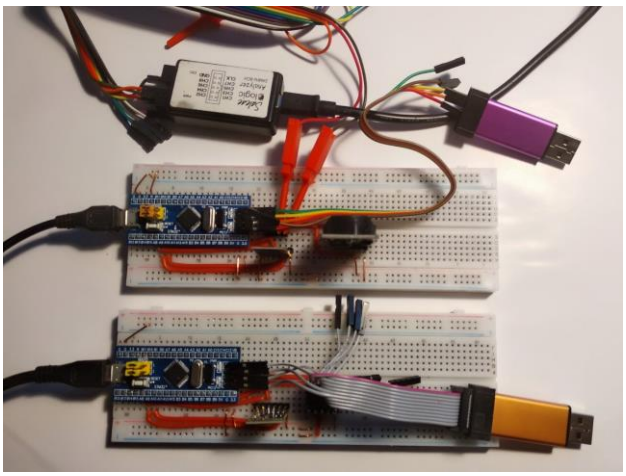

Рис. 3. Робочі макети пристроїв для передачі даних по радіоканалу

Всі програми для мікроконтролерів починаються з ініціалізації. Спочатку треба ініціалізувати тактовий генератор мікроконтролера з вказаними джерелами тактового сигналу, множенням тактового сигналу для ядра та діленням для периферії та системних шин. Після цього ініціалізується потрібна для роботи периферія мікроконтролера. А саме: GPIO – порти вводу виводу, UART – послідовний асинхронний порт та USB – універсальна системна шина.

Зазвичай першими ініціалізують порти вводу виводу (GPIO) [4]. GPIO - general-purpose input output pin або ж порт вводу виводу. Порт вводу виводу – це контакт загального призначення. Може працювати як на вхід так і на вихід. Тобто має можливість або читати логічний стан контакта або навпаки задавати логічний рівень (рівень напруги). Також деякі групи контактів мають можливість працювати в аналоговому режимі при вимірах аналого-цифрового перетворювача (АЦП). Потім треба послідовну шину UART.

UART – це універсальна асинхронна послідовна шина [5]. Шина UART дуже гнучка та дозволяє підключати багато різних пристроїв (мікросхем модулів). Сама шина є повністю дуплексною. Тобто дозволяд передавати данні в обидві сторони як від ведучого пристрою так і від веденого пристрою.

Ініціалізація UART починається з тактування. Після ініціалізації тактування UART необхідно налаштувати контролер NVIC який відповідає за події та вектори переривання.

NVIC (Nested vectored interrupt controller) – модуль контролю переривань [6]. Він виконує наступні функції:

- Дозволяє або забороняє переривання.
- Назначає пріоритет переривань (від 0 до 15. 0 - максимальній пріоритет, 15 - мінімальний пріоритет).
- Автоматично зберігає дані при виконанні одиноких чи вкладених переривань.

При використанні промислового зразка розробленої системи потрібно мати комп'ютерний пристрій, який може передавати цифровий сигнал. Пристрій повинен бути оснащений засобами Bluetooth, Wi-Fi або радіомодулем.

Список літератури

1. STM32CubeIDE [Електронний ресурс]. Режим доступу: https://www.st.com/en/development-tools/stm32cubeide.html.
2. STM32F103C8 [Електронний ресурс]. Режим доступу: https://www.st.com/en/microcontrollers-microprocessors/stm32f103c8.html.
3. JDY-40 2.4G wireless serial port transmission transceiver and remote communication module [Електронний ресурс]. Режим доступу: https://sunhokey.cn/collections/wifi-module/products/jdy-40-2-4g-wireless-serial-port-transmission-transceiver-and-remote-communication-module.
4. GPIO internal peripheral [Електронний ресурс]. Режим доступу: https://wiki.st.com/stm32mpu/wiki/GPIO_internal_peripheral.
5. AN2582 Application note [Електронний ресурс]. Режим доступу: http://read.pudn.com/downloads106/sourcecode/embed/437624/stm32/STM32F Documents/Application Note/STM32F10xxx USART application examples.pdf.
6. Nested Vectored Interrupt Controller (NVIC) [Електронний ресурс]. Режим доступу: http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.dai0179b/ar01s01s01.html

УДК 004.056.5:343.326 (045)

О.О. Маліновська, студ. гр. МКБ-211
Науковий керівник: Ю.М Ткач, *к.т.н., професор*
Національний університет «Чернігівська політехніка», м. Чернігів

ПРИНЦИП РОБОТИ ЕЛЕКТРОННОГО ЦИФРОВОГО ПІДПИСУ

Електронний цифровий підпис (ЕЦП) – реквізит електронного документа, призначений для захисту даного електронного документа від підробки, отриманий внаслідок криптографічного перетворення інформації з використанням закритого ключа електронного цифрового підпису та що дозволяє ідентифікувати власника сертифіката ключа підпису, а також встановити відсутність спотворень інформації в електронному документі. Електронний цифровий підпис в електронному документі рівнозначний власноручному підпису в документі на паперовому носії.

При цьому електронний документ з електронним цифровим підписом має юридичне значення під час здійснення відносин, зазначених у сертифікаті ключа підпису.

Загальна суть електронного підпису ось у чому. За допомогою криптографічної хеш функції обчислюється відносно короткий рядок символів фіксованої довжини (хеш). Потім цей хеш шифрується закритим ключем власника – результатом є підпис документа. Підпис додається до документа, таким чином виходить підписаний документ. Особа, яка бажає встановити справжність документа, розшифровує підпис відкритим ключем власника та обчислює хеш документа. Документ вважається справжнім, якщо обчислений за документом хеш збігається з розшифрованим з підпису, інакше документ є підробленим. (див. рис.1).

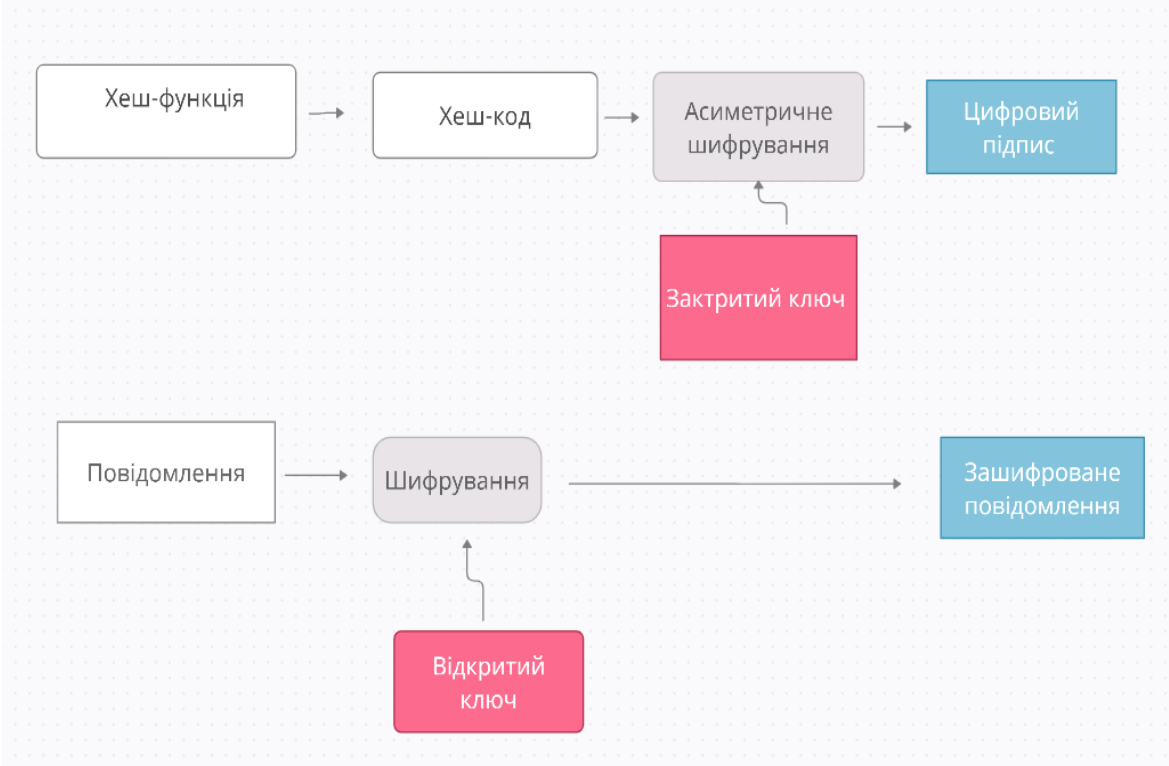

Рис. 1. Принцип роботи ЕЦП

Під час ділового листування, під час укладання контрактів підпис відповідальної особи є неодмінним атрибутом документа, який має кілька цілей:

– гарантування істинності листа шляхом звірення підпису з наявним зразком;

– гарантування авторства документа (з юридичної точки зору).

Виконання цих вимог ґрунтується на таких властивостях підпису:

– підпис автентичний, тобто з його допомогою одержувачу документа можна довести, що він належить підписувачу;

– підпис непідробний; тобто є доказом, що тільки та людина, чий автограф стоїть на документі, могла підписати цей документ, і ніхто інший;

– підпис непереносимий, тобто є частиною документа і тому перенести його на інший документ неможливо;

– документ з підписом є незмінним;

– підпис незаперечний;

– будь-яка особа, яка володіє зразком підпису, може переконатися, що документ підписаний власником підпису.

Розвиток сучасних засобів безпаперового документообігу, засобів електронних платежів немислимо без розвитку засобів доказу справжності та цілісності документа. Таким засобом є електронно-цифровий підпис (ЕЦП), який зберіг основні властивості звичайного підпису.

Існує кілька методів побудови ЕЦП, а саме:

– Шифрування електронного документа (ЕД) на основі симетричних алгоритмів. Ця схема передбачає наявність у системі третьої особи – арбітра, котрий користується довірою обох сторін. Авторизацією документа в даній схемі є сам факт шифрування ЕД секретним ключем та передачі його арбітру.

– Використання асиметричних алгоритмів шифрування. Фактом підписання документа є його шифрування на секретному ключі відправника.

– Розвитком попередньої ідеї стала найбільш поширена схема ЕЦП - шифрування остаточного результату обробки ЕД хеш-функції їй з допомогою асиметричного алгоритму.

Поява цих різновидів зумовлено різноманітністю завдань, розв'язуваних за допомогою електронних технологій передачі та обробки електронних документів.

При генерації ЕЦП використовуються параметри трьох груп:

– Загальні параметри

– Секретний ключ

– Відкритий ключ

Вітчизняним стандартом на процедури вироблення та перевірки ЕЦП є ГОСТ Р 34.10-94. [1]

Список літератури

1. Рябко Б. Я., Фіонов А. Н. Основи сучасної криптографії для фахівців в інформаційних технологіях - Науковий світ, 2004. - 173 с. - ISBN 978-5-89176-233-6

2. Статистика електронного підпису 2020/2021 [Електронний ресурс] - https://financesonline.com/25-essential-e-signature-statistics-analysis-of-trends-data-and-market-share/.

УДК 004.652

О.В. Губенко, І.О. Розломій
gubenko.oleksandr@vu.cdu.edu.ua, inna-roz@ukr.net
Черкаський національний університет імені Богдана Хмельницького, м. Черкаси

DYNAMO DB – NOSQL СИСТЕМА КЕРУВАННЯ БАЗАМИ ДАНИХ

Amazon Dynamo DB – це повністю керована служба NoSQL баз даних, яка забезпечує швидкий доступ до даних за рахунок постійного масштабування. Dynamo DB дозволяє зменшити кількість роботи, що відноситься до масштабування розподіленої бази даних. Таким чином, щоб не потрібно було турбуватися про програмне забезпечення обладнання, налаштування та конфігурацію, реплікацію, виправлення програмного забезпечення або масштабування кластера.

За допомогою Dynamo DB можна створювати таблиці бази даних, які можуть зберігати та отримувати будь-яку кількість даних і обслуговувати будь-який рівень трафіку запитів. Ви можете збільшити або зменшити пропускну здатність ваших таблиць без простоїв або зниження продуктивності. Ви можете використовувати AWS Management Console для моніторингу використання ресурсів і показників продуктивності [1].

Dynamo DB автоматично розподіляє дані та трафік для ваших таблиць на достатню кількість серверів, щоб задовольнити ваші вимоги до пропускної здатності та зберігання, підтримуючи при цьому стабільну та швидку продуктивність. Усі ваші дані зберігаються на твердо тільних дисках (SSD) і автоматично реплікуються в кількох зонах доступності в регіоні AWS, забезпечуючи вбудовану високу доступність і довговічність даних.

У Dynamo DB таблиці, елементи та атрибути є основними компонентами, з якими ви працюєте. Таблиця – це набір елементів, а кожен елемент — це набір атрибутів. Dynamo DB використовує первинні ключі для унікальної ідентифікації кожного елемента в таблиці та вторинні індекси, щоб забезпечити більшу гнучкість запитів.

Розглянемо сновні компоненти СКБД Dynamo DB.

Таблиці – Подібно до інших систем баз даних, Dynamo DB зберігає дані в таблицях. Таблиця – це набір даних.

Елементи – кожна таблиця містить нуль або більше елементів. Елемент – це група атрибутів, яку можна однозначно ідентифікувати серед усіх інших елементів. Елементи в Dynamo DB багато в чому подібні до рядків, записів або кортежів в інших системах баз даних. У Dynamo DB немає обмежень на кількість елементів, які ви можете зберігати в таблиці.

Атрибути – кожен елемент складається з одного або кількох атрибутів. Атрибут – це фундаментальний елемент даних, який не потребує подальшого розбиття. Атрибути в Dynamo DB багато в чому схожі на поля або стовпці в інших системах баз даних. Приклад даних які зберігаються в СКБД показано на рис 1.

Рис. 1. Приклад таблиці "People" в СКБД Dynamo DB

Dynamo DB підтримує два різні типи первинних ключів.

Ключ розділу(Partition key) – простий первинний ключ, що складається з одного атрибута, відомого як ключ розділу. БД використовує значення ключа розділу як вхід для внутрішньої хеш-функції. Вихід з хеш-функції визначає розділ (фізичне сховище, внутрішнє для DynamoDB), в якому буде зберігатися елемент. У таблиці, яка містить лише ключ розділу, два елементи не можуть мати однакове значення ключа розділу.

Ключ розділу та ключ сортування(Partition key and sort key). Цей тип ключа, який називається складеним первинним ключем, складається з двох атрибутів. Перший атрибут – це ключ розділу, а другий – ключ сортування.

DynamoDB використовує значення ключа розділу як вхід до внутрішньої хеш-функції. Вихід з хеш-функції визначає розділ (фізичне сховище, внутрішнє для DynamoDB), в якому буде зберігатися елемент. Усі елементи з однаковим значенням ключа розділу зберігаються разом у порядку сортування за значенням ключа сортування.

У таблиці, яка має ключ розділу та ключ сортування, кілька елементів можуть мати однакове значення ключа розділу. Однак ці елементи повинні мати різні значення ключа сортування.

Ключ розділу елемента також називають хеш-атрибутом. Термін хеш-атрибут походить від використання внутрішньої хеш-функції в DynamoDB, яка рівномірно розподіляє елементи даних між розділами на основі їх значень ключа розділу.

Ключ сортування елемента або ж атрибут діапазону – походить від того, як DynamoDB зберігає елементи з однаковим ключем розділу фізично близько один до одного, у порядку сортування за значенням ключа сортування.

Різниця принципу розподілу даних при використанні «Ключа розділу» та «Ключа розділу та ключа сортування» зображено на рис. 2(а) та рис. 2(б) відповідно.

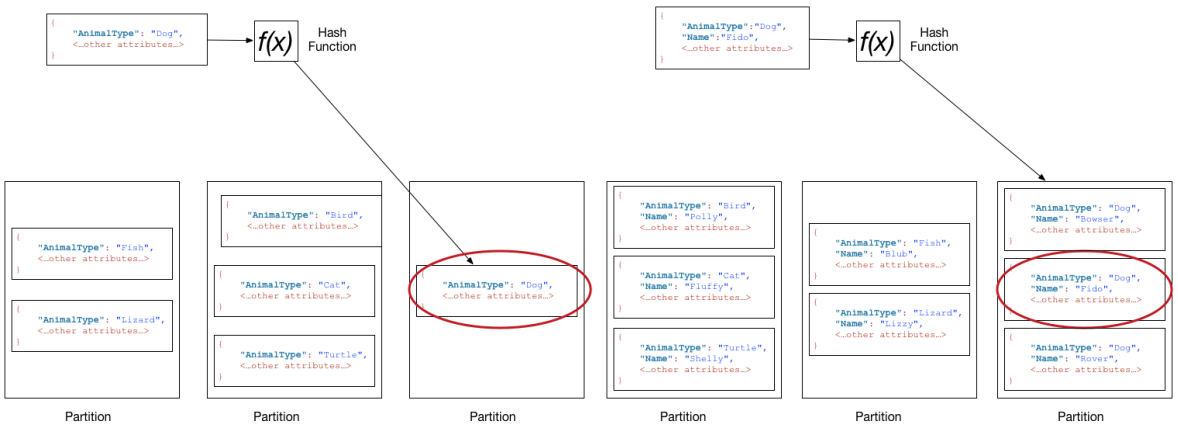

Рис. 2. а) – принцип розподілу даних Partition key; б) – принцип розподілу даних Partition Key and sort key

В Dynamo DB можна створити один або кілька вторинних індексів у таблиці. БД не вимагає використання індексів, але вони надають вашим програмам більшу гнучкість під час запитів ваших даних. Після створення вторинного індексу для таблиці ви можете читати дані з індексу приблизно так само, як і з таблиці.

DynamoDB підтримує два види індексів. Глобальний вторинний індекс – індекс із ключем розділу та ключем сортування, які можуть відрізнятися від тих, що містяться в таблиці.

Локальний вторинний індекс – індекс, який має той самий ключ розділу, що й таблиця, але інший ключ сортування. Кожна таблиця в DynamoDB має квоту з 20 глобальних вторинних індексів (квота за замовчуванням) і 5 локальних вторинних індексів.

Однією з особливостей Dynamo DB також є підтримка DynamoDB Streams – це додаткова функція, яка фіксує події зміни даних у таблицях DynamoDB. Дані про ці події відображаються в потоці майже в реальному часі і в тому порядку, в якому відбувалися події.

Кожна подія представлена записом потоку. При активації DynamoDB Streams, система буде здійснювати запис потоку щоразу, коли відбувається одна з таких подій:

– до таблиці додається новий елемент;

– елемент оновлено;

– елемент видалено з таблиці.

Кожен запис потоку також містить назву таблиці, мітку часу події та інші метадані. Потокові записи мають термін служби 24 години, після цього вони автоматично видаляються з потоку.

Список літератури

1. Core components DynamoDB. – [Електронний ресурс] – Режим доступу до ресурсу: docs.aws.amazon.com/amazondynamodb/latest/developerguide/HowItWorks.CoreComponents (Дата звернення: 09.05.2022).

.

УДК 004.056

Е.Ю. Яриніч, Р.М. Минайленко
aron70@ukr.net
Центральноукраїнський національний технічний університет, м. Кропивницький

ПРОБЛЕМИ ТА ПЕРСПЕКТИВИ ТЕХНОЛОГІЇ IoT

Інтернет речей — це нова і перспективна технологія, що має на меті глобальну зміну нашого світу шляхом об'єднання фізичних об'єктів («речей») і кіберпростору. Концепція Інтернету речей включає багато технологій, наприклад Інтернет, розподілені обчислення, машинне навчання, комунікації, великі дані, сенсорні технології, взаємодія машина-людина, машина-машина.

Інтернет речей, аналітика великих даних і машинне навчання — це ті області науки і техніки, які бурхливо розвиваються та формують нове покоління комп'ютерних систем з використанням штучного інтелекту. Варто відмітити, що ці області є міждисциплінарними за своєю природою. Це дозволяє їм нагромаджувати як і теоретичні основи, так і розвиватися в практичній площині.

Термін «Інтернет речей» (Internet of things, IoT) був запропонований в 1999 році Кевіном Ештоном під час його роботи над створенням Procter&Gamble, який припустив, що можливо зв'язати кілька фізичних об'єктів («речей») на виробництві для обміну інформацією і взаємодії між собою і із зовнішнім оточенням. У 2010 році в результаті стрімкого поширення смартфонів і планшетних комп'ютерів під поняттям «Інтернет речей» стали розуміти непросто автоматизацію процесів на локальному виробництві, але й глобальніше поняття, коли не тільки комп'ютер або смартфон, а й інші прилади, починаючи з кавомашини в офісі і закінчуючи холодильником у будинках, підключені до інтернету. У звичайних споживачів з використанням таких технологій життя стає комфортніше. У народному господарстві це спосіб економії ресурсів і оптимізації виробництва. IoT дозволяє створювати динамічні мережі, що складаються з мільярдів елементів, що можуть взаємодіяти між собою. Таким чином забезпечується зв'язок між накопиченим обсягом даних і реальними об'єктами, для яких додатки, сервіси, самі пристрої — це джерела даних.

По суті, Інтернет речей — це одна величезна хмара. Сам по собі крихітний чіп примітивний за своєю архітектурою: він нічого не може сказати своєму власнику, лише приймати потрібну йому інформацію. Як тільки він зв'язується через Wi-Fi з відповідним комп'ютером або системою обробки цієї інформації, цей чіп прирівнюється до суперкомп'ютера, що обробив отриману інформацію.

Передбачається, що в майбутньому IoT стануть активними учасниками бізнесу, інформаційних і соціальних процесів, де вони зможуть взаємодіяти і спілкуватися між собою, обмінюючись інформацією про навколишнє середовище, реагуючи і впливаючи на процеси, що відбуваються в навколишньому світі, без втручання людини.

Ряд вчених, таких як Роб Ван Краненбург, стверджують, що IoT можна уявити як «чотирьох-шаровий пиріг» (рис. 1). Це класифікація за масштабністю використання IoT:

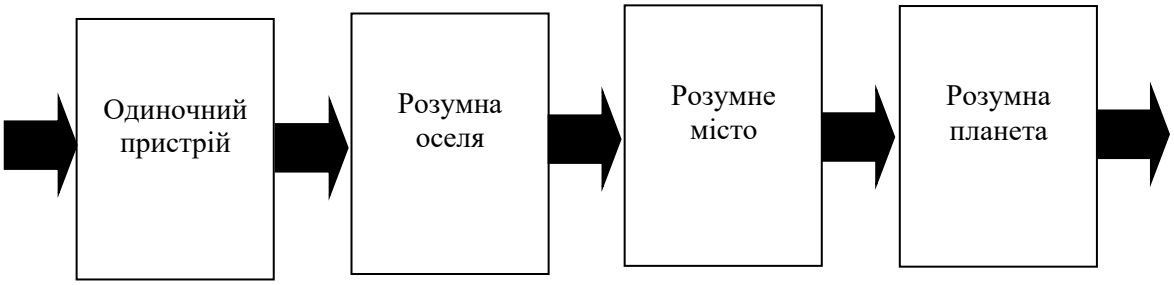

Рис. 1. Класифікація IoT за Робом Ван Краненбургом

Прогнозується, що до кінця 2022 кількість пристроїв які охоплюватиме технологія IoT сягатиме !00 млрд. Однак ці прогнози не враховують прискореного розвитку технологій і пристроїв. Комусь кількість підключених пристроїв може здатися заниженою. Це пов'язане з тим, що в розрахунках враховується все населення нашої планети, але більшість людей до сих пір не має доступу в Інтернет. Якщо ж враховувати тільки тих, хто користується Інтернетом, то кількість підключених пристроїв на одного користувача виявиться набагато вище.

На теперішній час Інтернет речей підходить до етапу, на якому різнорідні мережі і безліч датчиків потрібно об'єднати для взаємодії під керуванням єдиних стандартів. Цю мету поставлять перед собою комерційні організації, державні установи, органи, що займаються розробкою стандартів, і навчальні заклади.

УДК 004.056.

О.М. Полевод, М.О. Трощилов студ гр. МКБ-211
miketroshh@gmail.com, alexsnake97@gmail.com
Науковий керівник: Ю.М. Ткач., *професор*
Чернігівський Національний Технологічний Університет, м. Чернігів

OPEN SOURCE INTELLIGENCE ЯК ПРОВІДНИЙ НАПРЯМ КОНКУРЕНТНОЇ РОЗВІДКИ

Вступ

На сьогоднішній день, інформатизація та діджиталізація суспільства стає майже всеохоплюючою. Крім звичних наукових статей, відомостей про життя суспільства та держави, маючи певні навички, в мережі можна знайти повну інформацію про особу або підприємство, при цьому не порушуючи закон. Це стало можливим через те, що люди з різних причин демонструють особисте життя у соціальних мережах, не замислюючись над тим, як і ким ці дані можуть бути використані. Саме тому, на нашу думку, тема пошуку по відкритим джерелам є актуальною і потребує дослідження.

Поняття OSINT

OSINT(Open Source Intelligence) – процес пошуку, збору та аналізу інформації, зібраної із відкритих джерел. Наприклад, отримання інформації про особу чи підприємство із соціальних мереж, особистих сайтів, форумів та пошукових систем.

Виділяють шість основних джерел для *OSINT*

- ЗМІ (друковані газети, журнали, радіо та телебачення з різних країн.)
- Інтернет (онлайн-публікації, блоги, дискусійні групи, медіа громадян (наприклад, відео з мобільних телефонів, вміст, створений користувачами), YouTube та інші відео-хостинги, вікі-довідники та інші веб-сайти соціальних медіа (наприклад, Facebook, Twitter, Instagram та ін.). Державні дані (публічні урядові звіти, бюджети, слухання, телефонні довідники, прес-конференції, веб-сайти та виступи.
- Професійні та академічні публікації (інформація, отримана з журналів, конференцій, симпозіумів, наукових праць, дисертацій та дисертацій.)
- Комерційні дані (комерційні зображення, фінансові та промислові оцінки, бази даних.)
- Сіра література (технічні звіти, препринти, патенти, робочі документи, ділові документи, неопубліковані роботи та інформаційні бюлетені.)

Також іноді використовують метод спостереження - радіомоніторинг, використання загальнодоступних даних дистанційного зондування землі та аерофотозйомок (наприклад, Google Earth).

Методи збору інформації

1. Автоматизовані засоби. Розглянемо їх на прикладі ПЗ «Spectrum» від компанії «Інфозахист». Дане ПЗ має такі функціональні можливості:

- Автоматичний збір і аналіз активності об'єкту, що цікавить, у соціальних мережах або мережі інтернет. Аналіз взаємозв'язків, за якими була виявлена найбільша активність (лайки, коментарі тощо)
- Робота з соціальними мережами: Facebook, VK, Одноклассники, Instagram, Twitter та інші
- Робота з месенджерами
- Робота з відкритими базами даних, наприклад, «Миротворець»
- Потужний механізм аналізу даних, включаючи розпізнавання фото та повний цикл аналізу текстової інформації з автореферуванням, що працює на нейронних мережах. Модуль тестового аналізу дозволяє находити та визначити сутності (ПІБ, посади, організації, телефони, географічні назви та інші), визначати тексти, автоматично складати тематичні словники, виявляти тональність тексту. Цей модуль може використовуватись як окреме рішення для роботи з великими масивами текстової інформації.

Аналогами є наступні програми:

- Spiderfoot
- theHarvester
- Recon-ng

2. Сервіси. Вони надають можливість отримати важливі дані про веб-сайт підприємства або персональні сайти. Сервіси можуть надати наступну інформацію:

- Перелік активних мережевих вузлів та портів
- Перелік імен співробітників
- Перелік поштових адрес
- Перелік субдоменів
- Перелік зовнішніх Java-скриптів
- API-ключі
- Акаунти у соціальних мережах

- Пошук за іншими зовнішніми базами

Найбільш популярними та функціональними сервісами такого роду є Shodan, Threatcrowd, DnsDumpster.

3. SOCMINT. Даний підрозділ OSINT спрямований на роботу із соціальними мережами. Вже існує велика кількість онлайн-інструментів для збору інформації у найпопулярніших соцмережах(LinkedIn, Facebook, Twitter, Instagram). Далі наведено перелік найбільш потужних із них:

- Stalkscan - показує всю публічно доступну інформацію про людину.
- Foller - показує інформацію про будь-який відкритий акаунт, включаючи число твітів і підписників, списки, хештеги і згадки.
- Tinfoleak - показує девайси, операційні системи і соціальні мережі, які використовуються користувачем. Також показує локації, співвідносячи твіти користувача з місцями в Google Earth.
- InSpy - Python-програма, яка вміє знаходити працівників тієї чи іншої компанії. Також знаходить технології, що застосовуються в компанії, за заданими ключовими словами.
- www.picodash.com - надає статистику по підписниках конкретного користувача або обраного хештега в форматі CSV. Також відображає лайки і коментарі.

4. Мануальний пошук. Даний метод включає в себе моніторинг популярних веб-ресурсів на наявність інформації про особу/підприємство.

Якщо з усього перерахованого вище потрібно провести тільки декілька перевірок, тоді можна обійтися без встановлення додаткових програм. А якщо проводиться ретельний і тривалий пошук, краще скористатися спеціальним ПЗ, яке допоможе заощадити час.

Далі розглянемо як можна застосувати дану методику на практиці: перш за все, висока обізнаність зловмисника про особу/підприємство значно спрощує застосування одного із найефективніших методів соціальної інженерії:

Таргетований фішинг – можливість розповсюджувати фішингові повідомлення, складені так, що жертва не виявить обману с більшою вірогідністю.

Іншим застосуванням зібраної інформації може бути планування масштабної кібер-атаки на веб-ресурс особи чи підприємства. З іншого боку дану методику можна використати і для попередження дій зловмисників. Пошук за відкритими джерелами – ефективний спосіб зрозуміти, як виглядає організація з точки зору зовнішнього потенційного зловмисника. Цей набір заходів дозволяє оперативно оцінити потенційні точки входу до інфраструктури і почати опрацювання заходів для превентивної боротьби із злочинною активністю.

Під час дослідження всіх аспектів даної теми, нами було розроблено концепт програмного модулю для автоматизованого збору інформації про особу у соціальних мережах шляхом розпізнавання обличчя. Далі наведено приблизний алгоритм роботи модулю:

1. Навчена нейромережа аналізує обличчя особи на фото, визначаючи його характерні риси.
2. Шляхом порівняння отриманих даних обличчя із даними обличь акаунтів у соціальних мережах(Facebook та Instagram) програма створює звіт, який містить усі можливі збіги. Для оптимізації даного процесу рекомендується задавати регіон пошуку.
3. Користувач із наведеного переліку обирає найбільш вірогідний збіг за даними. Програма формує досьє по даній особі, використовуючи інформацію із акаунтів.

Перевагою розробленого нами алгоритму є можливість отримати досьє, не порушуючи чинне законодавство України. Подальшим розвитком даного дослідження є програмна реалізація алгоритму у вигляді пошукового модуля.

Провівши дослідження методики пошуку у відкритих джерелах, визначивши основні джерела та способи збору інформації, ми можемо зробити висновок, що OSINT є потужним інструментом для конкурентної розвідки, збирання досьє на осіб, тощо. На основі зібраних даних можна проводити як малі так і масштабні кібер-атаки, застосовувати інформацію для складання таргетованої фішинг-розсилки, яка буде ефективніша від звичайного спаму. Ще одним можливим застосуванням даної методики пошуку є попередження описаних вище загроз особі чи підприємству.

На нашу думку, більш глибоке вивчення OSINT сприятиме розвитку галузі конкурентної розвідки та підвищить ефективність роботи агентів.

Анотація:

У даних тезах проведено дослідження одного із найперспективніших напрямків конкурентної розвідки – OSINT. Було розглянуто основні джерела та методи пошуку. Другим кроком у дослідженні став опис можливих шляхів застосування зібраних даних. Отримані результати показали, що даний напрям є потужним інструментом як і для зловмисників, так і фахівців із інформаційної безпеки.

Список літератури

1. Сбор информации из открытых источников [Електронний ресурс] – Режим доступу до ресурсу:https://www.antimalware.ru/analytics/Threats_Analysis/Gathering-information-the-way-cybercrooks-see-you

2. Возрастающая роль OSINT [Електронний ресурс] – Режим доступу до ресурсу: https://www.securitylab.ru/blog/personal/Business_without_danger/344893.php

3. Целенаправленные атаки: разведка на основе открытых источников (OSINT) [Електронний ресурс] – Режим доступу до ресурсу:https://xakep.ru/2018/05/14/osint/.

УДК 339.138:004.658.6: 004.896

Я.С. Швець, І.О. Розломій
yaroslav.shvets.3707@gmail.com, inna-roz@ukr.net
Черкаський національний університет імені Богдана Хмельницького, м. Черкаси

ЗАСТОСУВАННЯ ТЕХНОЛОГІЙ BIG DATA ДЛЯ ОПТИМІЗАЦІЇ ЛОГІСТИЧНИХ ПРОЦЕСІВ

Big Data набирають своєї популярності в сучасному економічному світі. Необхідність використання інструментів Big Data в логістичних мережах великих компаній, військових та урядових організацій виникла після переходу до сучасних технологій, що реалізують збір та обробку даних із міток RFID, встановлених на кожній транспортній упаковці.

Big Data – це серія підходів, інструментів і методів обробки структурованих і неструктурованих даних величезних обсягів і значного різноманіття для отримання результатів, що сприймаються людиною, ефективних в умовах безперервного приросту, розподілу по численним вузлам обчислювальної мережі, що сформувалися в кінці 2000-х років, альтернативних традиційним системам керування базами даних [1].

Використовуючи дані інвентаризації та продажу, роздрібні компанії можуть адекватно оцінити ситуацію на полицях. Дані з продажу окремо також не гарантують, що ви можете зробити остаточні висновки про поточну ситуацію. Якщо десять одиниць конкретного товару завжди продаються у певний день тижня, то є ймовірність того, що наступного дня він закінчиться. З одного боку, асортимент товарів розширюється, і диференціація товару стає дедалі більшою. Крім того, є агресивні конкуренти та маркетингові акції. З іншого боку, клієнти стають все більш непередбачуваними споживачами. Навіть задоволений клієнт не обов'язково вибере той самий магазин для своєї наступної покупки. Все це збільшує непередбачуваність попиту та ускладнює продажі. Інтернет-магазини та його логістичні партнери повинні доставити дуже багато вантажів безпосередньо кінцевому споживачу.

Обробка всієї сукупності даних про історію продажів, обсяги запасів, ціни, а також інші додаткові дані, наприклад, про постійних клієнтів, що мають дисконтні картки, про конкурентів, інформацію про фактори, що впливають на обсяги продажів, з метою сформувати конкурентні ціни та проводити ефективні маркетингові компанії [2].

Узагальнюючи можливі застосування інструментів Big Data, перерахуємо типові завдання, які вирішуються за їх допомогою:

1) аналітика з клієнтів/об'єктів;
2) операційна та поведінкова аналітика;
3) побудова сховищ даних, економічно ефективних з погляду витрат за одиницю обсягу даних, що зберігаються;
4) боротьба з шахрайством та контроль за дотриманням норм [3].

На сьогоднішній день інформація розширюється з геометричною прогресією за рахунок розвитку технологій у цій галузі та появи нових пристроїв. Проте обробка цих даних ставити дедалі нові завдання перед дослідниками, оскільки в переважній більшості є неструктурованими і хаотичними. Це призвело до появи нового напряму Big Data, у якого формуються підходи для дослідження великих обсягів даних виявлення шаблонів і розкриття нової інформації.

У рамках бізнес-процесів найбільш успішним та перспективним напрямком є використання даних технологій у рамках логістистичних процесів компаній.

Переважна більшість новостворених даних надходить із зображень камери відеоспостереження, запису в блогах, форумів, дискусій та каталогів електронної комерції. Ці дані є неструктурованими, хаотичними та не завжди достовірними. Всі ці тенденції призвели до виникнення нового напряму – технологій обробки Big Data. У широкому значенні слова – це сукупність підходів, інструментів і методів обробки як структурованих, і неструктурованих даних величезних обсягів масивів інформації різного типу за умов її неперервного приросту та актуалізації [4].

Найефективніше ці технології зараз реалізуються в логістиці. Сьогодні логістичні провайдери управляють масовим потоком товарів, і в процесі цього управління, своєю чергою, створюються величезні набори даних. І чим масштабніша логістична операція, тим вищі вимоги до ефективної роботи даних.

Безумовним лідером за кількістю найпопулярніших рішень у цій галузі є маршрутизація товарів та транспортних засобів. Це пов'язано з тим, що ці напрямки є ключовими факторами, що впливають на ефективність роботи логістичної системи будь-якої компанії, оскільки стосуються не тільки часу доставки, витрат топлива, але навіть оподаткування, оскільки в деяких країнах воно пов'язане з викидами CO_2 у атмосферу. Проте нині, як ніколи раніше, авіакомпанії, автотранспортні компанії та залізниці повинні враховувати велику кількість зовнішніх факторів при вирішенні цього завдання – розуміти наслідки зміни погодних умов, можливих пробок, обмежень часу роботи транспорту, графіків технічного обслуговування та багатьох інших факторів.

Іншим, але не менш перспективним напрямком є планування оперативної потужності. Це напрямок включає оптимізацію планування завантажувальних потужностей для вантажних автомобілів, поїздів та повітряних суден, а також планування перерозподілу персоналу на складах. Часто при традиційному вирішенні цього завдання використовують історичні середні або особистий досвід співробітників, що нерідко призводить до неефективності прийнятих рішень. Натомість, використовуючи можливості розширеної аналітики, технології Big Data дозволяють оцінити динаміку не лише всередині, а й поза розподільчою мережею. Моделюється та оцінюється вплив на потреби в потужності на підставі інформації про доставку в режимі реальної години. Ці дані автоматично надходять з управління складом, систем та даних датчиків вдоль транспортного ланцюга. Крім того, агрегується інформація про можливі зміни спросу на підставі зовнішніх даних клієнтів. Наприклад, дані про випуск продуктів у конкурентів, нові відкриття, які можуть вплинути на попит, або інформація про сховане банкрутство. Додатково виявляються та класифікуються випадкові інциденти (наприклад, аварії чи стихійні лиха), які можуть вплинути на попит. Точний прогноз на основі використання технології Big Data дозволяє передбачати збої ланцюга постачання та пом'якшувати ефект непередбачених інцидентів. Наприклад, такі рішення ефективно передбачають майбутні скупчення на маршрутах або пунктах транзиту, які не можуть бути усунені компанією, але можуть бути пом'якшені їх наслідки шляхом перемаршрутизації або прискорення інших процесів.

Логістичні компанії часто збирають відгуки клієнтів, оскільки це забезпечує не тільки розуміння якості обслуговування, а й очікувань, вимог клієнтів. Цей зворотний зв'язок є основним джерелом інформації для безперервного покращення якості обслуговування, а також є джерелом нових ідей у сфері пошуку напрямків можливих інновацій.

У минулому єдиним джерелом таких даних виступали система CRM (система управління взаємодією з клієнтом) та дані опитувань клієнтів. Але сьогодні рішення Big Data забезпечують доступ до великих обсягів корисних даних, що зберігаються на громадських інтернет/сайтах [2]. У соціальних мережах та на форумах люди відкрито та анонімно діляться досвідом обслуговування та спілкування з компанією, але вилучення таких даних вручну – це досить тривалий пошук. Лише поява складних методів обробки Big Data, таких як інтелектуальна обробка тексту та семантична аналітика, дозволяють автоматично отримати потрібну інформацію з величезних сховищ «текстів та звуків».

На практиці існують чисельні перешкоди у сфері реалізації рішень у Big Data. Насамперед, до них можна віднести низьку якість даних, конфіденційність деяких з них та складну технічну здійсненність. Проте, у довгостроковій перспективі ці перешкоди відійдуть на інший план, оскільки перед очима завжди буде саме досвід їх успішного застосування, який перетворив Google, Amazon, Facebook та eBay на лідерів інформаційного ринку.

Наразі проривні рішення в області Big Data зосереджені саме у логістиці, оскільки там виникла найбільша потреба. Досягнення у галузі технологічних та методологічних аспектів великих даних забезпечують великі переваги у секторі логістики. Існує великий невикористаний потенціал для покращення операційної ефективності, набуття досвіду роботи з клієнтами та створення нових бізнес-моделей. Потенціал рішень у логістичній сфері доводить, що Big Data вже стала рушійною конкурентною силою у висококонкурентних галузях, але крім технологічних змін неминучим спутником ефективного впровадження Big Data мають бути й організаційні зміни.

Список літератури

1. Фоміченко І. П., Баркова С. О. (2020) Смарт-логістика: концептуальні засади та перспективи розвитку в Україні. Економічний вісник Донбасу № 1(59). 63–71.
2. Kersten W., Seiter M., B. von See, Hackius N., Maurer T. (2017) Trends and Strategies in Logistics and Supply Chain Management. Digital Transformation Opportunities. DVV Media Group GmbH, Bremen. 71 p.
3. Gandhi P. The Synergy between Big Data and the Internet of Things. URL:https://opensourceforu.com/2017/07/synergy-big-data-internet-things/.
4. Carter C., Easton P. (2011) Sustainable supply chain management: evolution and future directions. International Journal of Physical Distribution & Logistics Management. №1(41). 46–67.

УДК 004.056

В.В. Буза, С.В. Михайлов, Р.М. Минайленко
aron70@ukr.net
Центральноукраїнський національний технічний університет, м.Кропивницький

ВЛАСТИВОСТІ КОНТРОЛЕРА ПЕРЕРИВАНЬ

З кожним переривання зв'язана та чи інша подія. Система повинна розпізнати, яке переривання за яким номером відбулося і яку відповідну підпрограму треба виконати.

Відомо два види переривань: ***апаратне*** і ***програмне***.

Програмне переривання зручно використовувати для організації доступу для окремих спільних для всіх програм модуля. Прикладні програми можуть самі встановлювати свої обробники переривань для їх послідуючого використовування іншими програмами (рис.1).

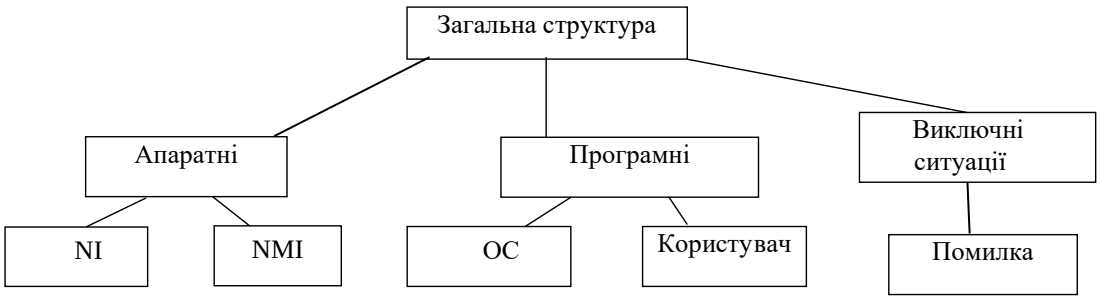

Рис.1. Види переривань

Помилка - являє собою виключні ситуації, які виявляються і обслужуються після вибірки до виконання команди, яка містить помилку.

Пастки – являють собою виключні ситуації, про які повідомляють одразу після виконання команди, яка привела до даної ситуації. CS і IP будуть вказувати на іншу команду, в разі порушення норм виконання команд безпосередньо на команду, яка порушила цей порядок.

Переривання, які визначаються користувачем є прикладом пастки.

Аварійне завершення **–** це виключна ситуація, що виникає при неможливості точно локалізувати джерело помилки і використати при виявленні помилок.

Апаратне переривання IRQ – використовує масковані переривання MI і немасковані NMI ((переривання інтерфейсів і пам'яті комп'ютера, які завжди проходять через PIC, який має регістр маски, що маскує необхідні переривання.

Програмні переривання – зручно використовувати для організації доступу до окремих і спільних для всіх програмних модулів.

Прикладні програми можуть встановлювати власні обробники переривань. Для цього переривання повинні бути резедентними у пам'яті. Використовуючи переривання з повільними пристроями дозволяю поєднати/О інформації з обробкою даних у ЦП.

Деякі переривання (перші 5) зарезервовані для використання самим процесором для виконання деяких подій. Складання особистих програм обробки переривання і заміна стандартних обробників є відповідальною і складною роботою. Необхідно враховувати усі тонкості роботи апаратури і взаємодії програмного і апаратного забезпечення.

Маскування переривання

Для виконання необхідної послідовності команд при наявності декількох переривань необхідно переривання з вищим пріоритетом замаскувати. Це можна зробити за допомогою команди CLI. Ця команда забороняє тільки масковані переривання, а немасковані завжди оброблюються процесором. Якщо використовується заборона переривань за допомогою CLI, то в кінці обов'язково треба поставити команду STI.

Зміна таблиці векторів переривань

Якщо вашій програмі потрібно змінити обробку деяких переривань, то програма повинна переназначити вектор переривання на свій обробник. Одним із шляхів вирішення є зміна в таблиці відповідного з векторів переривань.

Послідовність дій для нерезедентних програм обробки переривань:

- прочитати зміст елемента таблиці векторів переривань для вектора з потрібним вам номером;
- запам'ятати цей зміст (адреса старого обробника) у область даних програми;
- встановити нову адресу у таблиці векторів переривань так, щоб вона відповідала початку вашої;
- перед завершенням роботи програми прочитати із області даних адресу старого обробника переривання і записати у таблицю обробника переривання.

УДК 004.42

О.С. Савельєв, І.О. Розломій
saveliev.oleksandr1119@vu.cdu.edu.ua, inna-roz@ukr.net
Черкаський національний університет імені Богдана Хмельницького, м. Черкаси

СУЧАСНІ ФРЕЙМВОРКИ ДЛЯ РОЗРОБКИ ANDROID-ДОДАТКІВ

На сьогоднішній день мобільні програми знаходяться на піку своєї популярності. Кількість розробників мобільних додатків та доступних програм стрімко зростає. У зв'язку з цим збільшується кількість кросплатформних мобільних фреймворків для успішного створення програм. Повсюдно мобільні програми використовують дані із супутників, при цьому фреймворк дозволяє зробити спілкування з модулями супутникового зв'язку набагато зручнішим, оскільки не потрібно писати окремі нативні програми для кожної платформи [1].

Повний спектр обчислювальних сервісів і багаті функціональні можливості Android дозволяють створювати додатки, які далеко виходять за межі лише сегменту мобільних телефонів. Операційна система Android стає все більш популярною та в інших платформах та додатках.

Розробка програм під ОС Android є найбільш актуальним процесом у розробці мобільних програм. Тому актуальною є і тема роботи, пов'язана з аналізом та використанням інструментів програмування для ОС Android. Станом на січень 2022 року 69,74% від усіх мобільних операційних систем займає Android [2]. Сучасний розвиток тенденцій спонукує використовувати все новіші фреймворки для розробки Android додатків. Розглянемо деякі найкорисніші фреймворки для розробки додатків.

1. Sencha Touch – це фреймворк для розробки, який використовує JavaScript і HTML5, що дозволяє створювати динамічні та комплексні програми. Sencha Touch забезпечує високу продуктивність, оскільки включає в себе методи апаратного прискорення. Даний фреймворк допомагає розробляти захоплюючі та привабливі мобільні додатки з плавною анімацією та прокручуванням. Він також підтримує вбудований інтерфейс користувача та тестування в реальному часі [3].

2. TheAppsBuilder – ще один унікальний новий фреймворк Android, що базується на HTML і підтримує безкодовий інтерфейс користувача. TheAppBuilder має вбудовані блоки, які включають такі функції, як зворотний зв'язок, оновлення вмісту, опитування, push-повідомлення та багато іншого.

На даний момент існує сім популярних кросплатформових мобільних фреймворків, таких як Appcelerator Titanium, Kony Platform, Adobe PhoneGap, IBM Worklight, Telerik Platform, Verivo Akula та Xamarin [2]. Але тільки Xamarin створює програми за допомогою мови C #.

3. Xamarin використовується для розробки додатків Android, використовуючи «.NET». Він відносно старіший за більшість інших кросплатформних фреймворків. Xamarin має розширений набір інструментів, за допомогою якого розробники можуть працювати з кодом, написаним на C# або XAML. Ця функція спільного використання коду скорочує час кодування для розробників і кількість помилок під час процесу розробки.

Xamarin – це фреймворк для кросплатформної розробки мобільних додатків (iOS, Android, Windows Phone) з використанням мови C#. Xamarin заснований на open-source реалізації платформи .NET – Mono. Код програми пишеться широко використовуваною мовою програмування C# із застосуванням всіх звичних мовних особливостей, наприклад, LINQ, лямбда-виразів, Generic і Async. При цьому є повний доступ до всіх можливостей SDK платформи та механізму створення UI (user interface), отримуючи на виході додаток, який нічим не відрізняється від native-додатків і не поступається ним у продуктивності [4-5].

Фреймворк Xamarin складається з кількох основних частин:

- Xamarin. IOS – бібліотека класів, що надає розробнику доступ до iOS SDK;
- Xamarin. Android – бібліотека класів, що надає доступ до Android SDK;
- компілятори для iOS та Android;
- IDE Xamarin Studio;
- плагін для Visual Studio.

Для кожної платформи Xamarin надає можливість використовувати нативні засоби розробки UI та нативні елементи інтерфейсу користувача. Для Android створення UI може відбуватися безпосередньо в коді або за допомогою декларативного підходу з описом інтерфейсу в XML. Для iOS це також код, або використання нативних засобів проектування інтерфейсу.

Для кожної з платформи потрібно реалізувати власний шар UI, тобто код, який відповідає за зовнішній вигляд програми, доведеться написати для кожної платформи окремо. Якщо розбивати програму на шари, то виходить така схема:

- Data Layer (DL) – сховище даних, наприклад, база SqlLite або xml-файли;
- Data Access Layer (DAL);
- Business Layer (BL) – шар , що містить бізнес-логіку додатку;
- Service Access Layer (SAL) – шар, що відповідає за взаємодію Космосу з віддаленими сервісами;
- Application Layer (AL) – шар, що містить платформозалежний код;
- User Interface Layer (UI) – шар інтерфейсу користувача.

Кросплатформні є всі шари, розташовані вище Application Layer.

Розробники Xamarin як середовище розробки пропонують використовувати або власну IDE – Xamarin Studio або Visual Studio. Xamarin Studio – кросплатформова IDE , яка працює як на Mac OS X , і Windows. Також Xamarin пропонує можливість вести розробку у Visual Studio після встановлення спеціального плагіна [4-5].

Розглядаючи способи створення програм для iOS і Android, багато хто вважає, що єдиним варіантом є машинні мови Objective – C, Swift і Java. Однак протягом кількох останніх років з'явилася нова екосистема платформ для розробки мобільних додатків.

Xamarin є унікальним у цьому просторі, оскільки пропонує одну мову C#, бібліотеку класів та середовище виконання, що працює на всіх трьох платформах для мобільних пристроїв – iOS, Android та Windows Phone і підтримує компіляцію власних (без інтерпретації) додатків, що досить швидко діють навіть для ресурсомістких ігор.

Кожна з цих платформ має окремий набір функцій і пропонує різні можливості створення власних програм, які компілюються в машинний код і узгоджено взаємодіють з базовою підсистемою Java. Наприклад, деякі платформи допускають розробку програм тільки на HTML і JavaScript, тоді як інші працюють на дуже низькому рівні і підтримують лише код на C або C++. Деякі платформи взагалі не використовують власний набір засобів управління.

4. Google Flutter передбачений для створення кросплатформних програм. Фреймворк написаний мовою Dart і використовує єдину кодову базу. Цей провідний фреймворк Android має новітній підхід до розробки додатків. Він спрощує багатоплатформний процес розробки для створення високоякісних інтерфейсів для Android та iOS. Фреймворк Flutter має значну перевагу від механізму 2D рендеринга під назвою Skia [4]. Flutter є надійною платформою тестування для виконання UI, модульних і функціональних тестів. Має багато переваг, як швидке відтворення, зчитування з екрана, велика кількість тем та багато іншого.

5. Appcelerator Titanium SDK дозволяє розробникам створювати власні програми за допомогою єдиної кодової бази JavaScript. В результаті Appcelerator довелося потрапити до списку нових фреймворків Android для розробки високоефективних та надійних додатків. Завдяки фреймворку можна створювати багатоплатформні програми. Крім того, він надає доступ до API для операційних систем, включаючи Android, iOS, Universal Windows, HTML5 і Blackberry.

6. Ionic допомагає розробникам створювати сучасні гібридні додатки з використанням HTML5, CSS3 і JavaScript на кількох платформах. Ionic включає в себе безліч програмних інструментів та анімацій. Ці ресурси сприяють розробці сучасних мобільних додатків. Фреймворк містить простий CLI, що надає доступ до емуляторів, перезавантаження в реальному часі, журналювання тощо. Ionic досить легко інтегрується з іншими бібліотеками, такими як Cordova та AngularJS [6].

7. React Native використовує React, величезну бібліотеку JS. Міжплатформна характеристика дозволяє розробникам писати код лише один раз, а потім виконувати його в іншому місці. React Native має вбудовані компоненти інтерфейсу користувача та доступ до рідного API, що дозволяє додаткам Android мати гарний дизайн та високу продуктивність.

В статті було розглянуто сучасні фреймворки Android, які використовуються для розробки мобільних додатків. Найбільша увага зосереджена на розгляд кросплатформного фреймворку Xamarin, оскільки він є одним з найбільш використовуваних інструментів.

Список літератури

2. Ескендир М.А. (2019) Введение в разработку мобильных приложений. Вестник магистратуры. №6(93). 33–36.

3. 15 Apps for Programming on Android – [Електронний ресурс] – Режим доступу до ресурсу: http://android.appstorm.net/roundups/developer/15-apps-forprogramming-on-android/ (Дата звернення: 25.04.2022).

4. Sencha Touch 2 Developer Preview [Електронний ресурс]. – Режим доступу до ресурсу: https://www.sencha.com/blog/sencha-touch-2-developer-preview/ (Дата звернення: 25.04.2022).

5. Robert C. Martin (2017) Clean Architecture. A Craftsman's Guide to Software Structure and Design, НьюДжерси: Прентис Холл. 432 с.

6. Шматко А.В., Федорченко В.Н. (2016) Обзор и анализ инструментов разработки мобильных приложений для ОС Android. Инновации в науке. № 5(54). 59–73.

7. Ionic Docs Core Concepts [Електронний ресурс]. – Режим доступу до ресурсу: https://ionicframework.com/docs/core-concepts/fundamentals (Дата звернення: 25.04.2022).

УДК 004.9

І.О. Розломій, С.В. Науменко
sergey.naumenko94@gmail.com, inna-roz@ukr.net
Черкаський національний університет імені Богдана Хмельницького, м. Черкаси

ФРЕЙМВОРКИ ДЛЯ РОЗРОБКИ СЕРВЕРНОЇ ЧАСТИНИ WEB-ДОДАТКІВ

Фреймворк – це бібліотека мови програмування, що динамічно поповнюється, в якій зібрані її базові модулі. Фреймворки створюються задля спрощення процесів розробки додатків, сайтів, сервісів. Щоб не писати модуль у додатку з нуля, набагато простіше звернутися до готових шаблонів фреймворків, які формують робоче середовище розробника.

Архітектура багатьох фреймворків заснована на декомпозиції декількох окремих шарів (додатки, модулі) проєкту. Це означає, що можна розширювати функціональність програми, виходячи з потреб і використовувати змінену версію разом з кодом фреймворку або задіяти сторонні програми.

Така гнучкість є однією з ключових переваг використання фреймворків [1].

MVC (Model-View-Controller – «Модель-Представлення-Контролер») – концепція програмування, що розділяє класи на три групи, що широко застосовується і при використанні фреймворків. Структура концепції представлена рисунку 1.

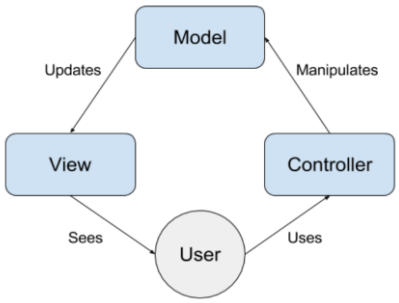

Рис. 1. Концепція MVC

Модель містить усі дані та рівні бізнес-логіки, її правила та функції. Подання відповідає за візуальне відображення даних, наприклад: діаграми, графіки тощо. Контролер легко трансформує дані для команд попередніх двох складових [1]. У фреймворків є дві основні функції: робота на серверній стороні (backend) та робота на клієнтській стороні (frontend).

Правила та архітектура серверних фреймворків не дозволяють створити веб-додаток з багатим інтерфейсом. Вони обмежені у своїй функціональності, проте все одно можна створювати прості сторінки та різні форми. Також вони можуть формувати вихідні дані та відповідати за безпеку у разі атак. Усе це безперечно може спростити процес розробки. Серверні фреймворки в основному відповідають за окремі, але критично важливі частини програми, без яких воно не зможе нормально працювати. Ось кілька самих популярних фреймворків та мови, з якими вони працюють: Django – Python, Zend – PHP.

Django – високорівневий фреймворк, який є не тільки швидким рішенням у веб-розробці, що включає все необхідне для якісного коду та прозорого написання, але також і відмінною платформою для роботи з клієнтурою того чи іншого бізнесу . Водночас він зручний для розробників.

У Django реалізований принцип DRY – Don't Repeat Yourself (не повторюйся). Тобто при використанні Django не потрібно кілька разів переписувати один і той же код. Фреймворк дозволяє створювати сайт з компонентів . Завдяки цьому скорочується час створення сайтів.

Фреймворк справляється з великим кількістю завдань та підвищеними навантаженнями. Його застосовують для створення:

- CRM-систем (Customer Relationship Management) відносинами з клієнтами);
- CMS-систем (Content Management System) – систем управління вмістом (контентом);
- комунікаційних платформ;
- сервісів бронювання номерів;
- платформ управління документообігом [4].

Також Django підходить для створення алгоритмічних генераторів, платформ для електронних розсилок, систем верифікації, систем фільтрації з динамічними правилами та складними параметрами, платформ для аналізу даних та складних обчислень, машинного навчання.

Реалізовано об'єктно-реляційне відображення (ObjectRelational Mapping (ORM), яке забезпечує взаємодія програми з базами даних (БД). ORM автоматично передає дані з БД, наприклад PostgreSQL або MySQL, в об'єкти, які використовуються в коді програми. включає механізми запобігання поширених атак на кшталт SQL-ін'єкцій (XSS, CrossSite Scripting – «міжсайтовий скриптинг») та підробки міжсайтових запитів (CSRF або XSRF) [2].

Django REST Framework, який часто скорочують до DRF, є бібліотекою для побудови програмного інтерфейсу програми, API (Application Programming Interface). Він має модульну та настроювану архітектуру, яка добре працює для створення як простих , і складних API. Він поставляється з базовими класами для CRUD-операцій та вбудованою утилітою для тестування розроблюваного API [2].

Таблиця 1

Переваги та недоліки фреймворку Django

№ п/п	Переваги	Недоліки
1	Маса бібліотек, що дозволяє не писати базову функціональність, а тільки лише імпортувати необхідне.	Django не підтримує WebSockets, тому він погано підходить для роботи в реальному часу.
2	Докладна документація та доброзичливе спільнота, тому завжди набагато простіше знайти вже готові рішення або отримати допомогу від спільноти.	Готові бібліотеки часто знижують гнучкість , Django є великим та монолітним фреймворком, та його компоненти розгортаються спільно.
3	Django дозволяє почати з малого і масштабуватися в міру необхідності.	

Zend Framework – це об'єктно-орієнтована середовище веб-додатків з відкритим вихідним кодом, що представляє собою набір професійних пакетів на основі PHP, розроблена та підтримувана компанією Zend. Це набір класів яких не прив'язані один до одного, а значить, їх можна, можливо використовувати як по запчастинам в інших проектах, так і повністю. У цьому каркасі є все, що потрібно в сучасному webпроекті. Цей фреймворк, як правило, більше всього використовують при розробці великих комерційних проєктів. Такі компанії, як Google, Microsoft і StrikeIron, співпрацюють з Zend, надаючи інтерфейси для веб-сервісів та інших технологій, які вони хочуть зробити доступними розробникам Zend. Компанія бере участь у розробці ядра мови програмування PHP [3].

Фреймворк використовує різні пакети, використовуючи Composer як частина своїх менеджерів залежностей пакетів. Деякі з них: PHPUnit для тестування всіх пакетів, Travis CI для служб безперервної інтеграції. Zend Framework надає користувачам підтримку MVC у поєднанні з рішенням Front Controller. Zend Framework слідує стандартам PHP-FIG і включає реалізацію PSR-7 для інтерфейсів HTTP- повідомлень. Підтримка декількох систем баз даних та постачальників, включаючи MariaDB, MySQL, Oracle, IBM DB2, Microsoft SQL Server, PostgreSQL , SQLite та Informix Dynamic Server [3].

За допомогою служб віддаленого виклику процедур (Remote Procedure Call, RPC) та REST (Representational State Transfer – «передача стану уявлення») Zend Apigility допомагає розробникам створювати та документувати API. Zend Server забезпечує покращену продуктивність для PHP, і особливо програм Zend Framework, за рахунок прискорення коду операції та кількох можливостей кешування, а також включає кошти моніторингу та діагностики додатків. Zend Studio є інтегрованої середовищем розробки програм (Integrated Development Environment, IDE), яка включає функції, спеціально призначені для спрощення роботи з Zend Framework. Вона забезпечує MVC-представлення, генерацію коду MVC на основі ZendTool (компонент Zend Framework), надає кошти форматування, аналізу та виправлення коду, допомога за параметрами та багато інше [3].

Таблиця 2

Переваги та недоліки фреймворку Zend

№ п/п	Переваги	Недоліки
1	Чудово підходить для розробки комерційних веб- додатків.	Великоваговий та ресурсомісткий , оскільки надає розробнику масу можливостей та функцій.
2	Об'єктно-орієнтований підхід до розробки.	Не підходить для швидкої розробки проектів.
3	Непов'язані компоненти для повторного використання у проектах.	Недостатньо матеріалів для російськомовного сегмента розробників.

Список літератури

1. Фрэйн, Б. (2017) HTML5 и CSS3. Разработка сайтов для любых браузеров и устройств. Второе издание = Responsive Web Desigh with HTML5 and CSS3. Second Edition. Санкт-Петербург: Питер. 272 с.

2. Django: The Web framework for perfectionists with deadlines. – [Електронний ресурс] – Режим доступу до ресурсу: http://www.djangoproject.com (Дата звернення: 30.05.2022).

3. Васвани, В. (2012) Zend Framework. Разработка веб-приложений на PHP = Zend Framework: A Beginner's Guide. Санкт-Петербург: Питер. 432 с.

УДК 004.94

А.М. Мельник, Д.О. Берестенко
mselnikanna@gmail.com,berestenkodaniilki212@gmail.com
Науковий керівник – Мелешко Є.В., *д.т.н., проф.,*
elismeleshko@gmail.com
Центральноукраїнський національний технічний університет, м. Кропивницький

ДОСЛІДЖЕННЯ ТЕХНОЛОГІЙ РЕАЛІЗАЦІЇ БЕЗДРОТОВИХ КАНАЛІВ ЗВ'ЯЗКУ

Термін "Бездротовий" використовується для позначення класу технологій зв'язку та технологій передачі даних без використання дротів або провідників. Зараз стрімко розвиваються різні бездротової комунікації, наприклад, Bluetooth-периферія, точки доступу WIFI, NFS (безконтактна оплата), навіть зарядні пристрої мають можливість бездротового з'єднання. Розглянемо типи бездротової передачі даних.

Супутникові канали. Цей спосіб передачі інформації полягає у використанні супутника, на якому встановлена антена зі спеціальним обладнанням. Сигнал надходить від абонента на найближчу наземну станцію, потім здійснюється переадресація сигналу на супутник. Звідти інформація відправляється на приймач, іншу наземну станцію. Супутниковий зв'язок використовується для забезпечення телебачення і радіомовлення. Супутниковим телефоном можна користатися в будь-якій віддаленій від станцій точці.

Інфрачервоні канали. Зв'язок встановлюється між приймачем і передавачем, які знаходяться на близькій відстані один від одного. Такий канал для бездротової передачі даних працює за допомогою світлодіодного випромінювання. Зв'язок може бути двостороннім або широкомовним.

Лазерні канали. Принцип дії такий же, як у попередньому варіанті, але замість світлодіодів використовують лазерний промінь. Об'єкти повинні знаходитися в безпосередній близькості один від одного.

Радіоканали стільникового зв'язку. Передача даних здійснюється бездротовим шляхом від передавача до приймача. Передавач формує радіоімпульс певної частоти й амплітуди, коливання випромінюється в простір. Приймач фільтрує і обробляє сигнал, після цього відбувається вилучення потрібної інформації. Мобільний зв'язок працює саме на основі радіохвильових стандартів.

Bluetooth. Це інтерфейсна безпроводова технологія. Діаметр мережі 10-30 м (у перспективі - 100 м). Працює в багатопунктовому режимі, не обов'язково в зоні прямої видимості. Головне призначення - створення побутових мереж, приєднання мультимедійної периферії, пральних машин, холодильників тощо. В 1997 р. створено перші приймачі-передавачі. У 1998 р. сформовано групу SIG, у яку ввійшли Ericsson, IBM, Intel, Nokia, Toshiba. У 1999 р. випущено специфікації на обладнання. Детальніше про технологію Bluetooth. Нові технології безпроводового передавання (Ultra Wideband (UWB)) пропонують швидкості передавання, які перевищують 100 Мбіт/с, та потребують мінімальних витрат енергії.

Технології та стандарти бездротової передачі даних

- Персональні мережі (WPAN). За допомогою цих стандартів підключається периферійне обладнання. Використовувати бездротові комп'ютерні миші і клавіатури набагато зручніше порівняно з дротовими аналогами. Швидкість бездротової передачі даних досить висока. Персональні мережі дозволяють обладнати системи розумних будинків, синхронізувати бездротові аксесуари з гаджетами. Прикладами технологій, що працюють за допомогою персональних мереж, є Bluetooth і ZigBee.
- Локальні мережі (WLAN) базуються на продуктах стандартів 802.11. Здатні створювати більший робочий радіус порівняно з WPAN, підвищився і рівень захисту.
- Мережі міського масштабу (WMAN). Такі мережі працюють за тим же принципом, що і Wi-Fi. Відмінною особливістю даної системи бездротової передачі даних є більш широкий обхват територій, підключитися до даної мережі може більше число приймачів.
- Глобальні мережі (WWAN) – GPRS, EDGE, HSPA, LTE. Мережі цього типу можуть працювати на основі пакетної передачі даних або за допомогою комутації каналів.

Висновки. Бездротові технології надали можливість повсюдного впровадження телекомунікаційного обладнання, яке масово використовується у всіх країнах світу. Постійні доопрацювання і нові відкриття в області бездротових комунікацій дають нам все більший рівень комфорту, а облаштування побуту за допомогою інноваційних приладів стає все більш доступним для більшості людей.

Список літератури

1. Бездротові технології. URL: https://www.wiki.uk-ua.nina.az/Бездротові_технології.html
2. Бездротова передача даних: типи, технологія та пристрої. URL: https://presa.com.ua/navchannia/bezdrotova-peredacha-danikh-tipi-tekhnologiya-ta-pristroji.html
3. Використання бездротових мереж у системах опрацювання біомедичних сигналів - Олексій Азаров, Сергій Богомолов, Леонід Крупельницький, Володимир Гончарук, Віталій Тищенко кафедра обчислювальної техніки, Вінницький національний технічний університет Вінниця, Україна

УДК 004.045

Д.О. Загребельна, І.О. Розломій, С.В. Науменко
zahrebelna.daria1119@vu.cdu.edu.ua, inna-roz@ukr.net, sergey.naumenko@gmail.com
Черкаський національний університет імені Богдана Хмельницького, м. Черкаси

СУЧАСНІ ПІДХОДИ ДО СЕМАНТИЧНОГО МОДЕЛЮВАННЯ ORM І ERMM

Своїм виникненням семантичне моделювання завдячує П. Чену та його моделі «Entity-Relationship». Ця методика доповнює класичну методику нормалізації, знімаючи такі проблеми останньої, як нетрадиційний для більшості людей і нетривіальний спосіб сприйняття та формалізації предметної області, практична незастосовність для складних предметних областей, неоднозначність вирішення проблеми проєктування, що призводить до прямого перебору багаточисельних варіантів схем в пошуках найбільш підходящої. Успіхи застосування семантичної методики проєктування баз даних (БД), крім знань і умінь людини, визначають два фактори: потужність виразних засобів семантичної моделі, що використовується, і детальність аналізу семантичної схеми в застосовуваних правилах її трансляції в модель СКБД.

Ідеальна семантична схема предметної області повинна містити повний її формальний опис для того, щоб усі подальші міжмодельні перетворення схеми мали суто синтаксичний характер і не вимагали повторного аналізу семантики предметної області та роботи з текстами природною мовою, в яких попередньо зафіксовані виявлені бізнес-правила. Можливість побудови таких семантичних схем відкриє перспективу повної автоматизації процесу проектування БД, включаючи створення таких елементів реалізації, як представлення, тригери та інше.

Далі розглянемо дві найбільш відомі семантичні моделі – OR-моделі та ERM-моделі.

Історія ORM почалася на початку 1970-х років, як семантичний підхід до моделювання, який сприймає світ просто як об'єкти, що грають ролі. Елементарний факт – це просте твердження чи атомарне судження у тому, що конкретний об'єкт грає конкретну роль. Слово «факт» вказує, що твердження прийняте дійсними користувачами цієї предметної. Роль, яку грає об'єкт, іноді називають властивістю об'єкта. Тоді елементарний факт стверджує, що певний об'єкт має певну властивість [1–2].

Світ, що моделюється, складають об'єкти. Основною інформаційною одиницею є судження, яке може бути виражене у мовній формі – висловлюванні. Причому головну роль (принаймні, при проектуванні та формуванні БД) відіграють поодинокі атрибутивні судження та поодинокі судження про відносини, що стверджують наявність у конкретного об'єкта певної якості або ставлення до інших об'єктів.

Характерним для одиничних суджень є використання як логічних висловлювань поодиноких імен, що підлягають їх, предметними значеннями яких є окремі предмети або об'єкти (емпіричні або ідеальні залежно від реальності або ідеальності модельованого світу). Таким чином, маємо одну з основних семантичних концепцій ERM-моделі – об'єкт.

Логічний присудок у таких висловлюваннях може бути задано з використанням одиночного загального імені, предикатора або предметного функтора, або складнішого логічного виразу, що включає ці знаки та логічні терміни. Усі зазначені семантичні категорії може бути виражені через предметні функції. Останні і представляють другу основну семантичну концепцію ERM-моделі, яку ми називатимемо більш відповідним терміном «відображення». Він, по-перше, несе математичного, кількісного сенсу образів і прообразів, традиційного поняття «функція». По-друге, з функцією, як правило, асоціюється вимога єдиності образу, чого нам також хотілося б уникнути.

Отже, в ERM-моделі для вираження будь-яких суджень про світ, що моделюється, достатньо двох семантичних концепцій – об'єкт і відображення. Об'єкт предметної області є унікальна цілісність, яку людина у процесі світосприйняття і мислення здатна відрізнити від того, що є даної цілісністю. Відображення предметонї – це певний закон предметної області, яким кожному об'єкту моделируемого світу (прообразу) то, можливо поставлений у відповідність один чи більше об'єктів (образів).

Зазначимо, що поки що йдеться про конкретні поняття «об'єкт» та «відображення», елементами обсягу яких є емпіричні чи ідеальні явища предметної області. Синтез на їх основі формальних понять ERM-моделі ще попереду.

Базові неформальні семантичні концепції грають настільки важливу роль, що вони часто становлять назву моделі. Так і з аналізованими моделями (ERM-модель спочатку називалася OM-моделью – модель «Object-Mapping»). Ці ж концепції знайшли однозначне відображення у базових формальних поняттях обох моделей. Зазначимо, що базових понять OR-моделі «об'єкт» та «роль» недостатньо для представлення більшої частини інформації.

На відміну від моделі ER, ORM не використовує атрибути. Усі факти представляються у термінах об'єктів (сутностей чи значень), які грають ролі. Хоча це часто призводить до великих діаграм, підхід без використання атрибутів має переваги для концептуального аналізу, що включають простоту, семантичну стабільність та легкість зміни.

Інша особливість ORM – її гнучка підтримка спеціалізації, включаючи множину успадкування, що базується на формальних визначеннях підтипів [2].

Основні базові поняття ERM-моделі – об'єкт, клас та відображення, що взаємно-однозначно відповідають нашим семантичним концепціям, забезпечують більшу частину виразної потужності моделі. Не дивно, що саме ці поняття представлені у синтаксисі та аксіоматиці формальної системи теорії семантично значимих відображень, що є математичною основою ERM-моделі. Проте людина завжди використовує цей дуже абстрактний рівень мислення. Для простоти роботи зі схемою виділяються приватні види об'єктів, класів та відображень, що утворюють безліч похідних понять моделі.

Об'єкти, мислимі у висловлюваннях про предметну як предмети, є сутності, а класи таких об'єктів не що інше, як безлічі сутностей. Ідеальні об'єкти, такі як числа, дати, рядки символів є значеннями. Вони не мають властивостей, характеристик і не вступають у відносини з іншими об'єктами, крім того, що є значеннями характеристик об'єктів. Їх об'єднують у безлічі значень із синтаксичними особливостями.

Відображення, що визначаються множинами зв'язків, які як області визначення та значень мають одиночні множини сутностей або їх декартові добутки, називаються реляційними. Ролі об'єктів-прообразів та об'єктів-образів цих відображень такі самі, як і ролі сутностей у зв'язках. Загальна кількість реляційних відображень, що визначаються однією множиною зв'язків ступеня n, дорівнює 2n-2.

Відображення, що ставить у відповідність об'єкту істинне значення, називається відображенням-властивістю. Якщо як область значень у відображенні використовується довільна безліч значень, таке відображення називатимемо відображенням-характеристикою. Відображення характеристики є не чим іншим, як атрибутними відображеннями або просто атрибутами.

Таким чином, нам вдалося пов'язати основні базові поняття ERM-моделі зі структурними поняттями моделі «Сутність – Зв'язок». Всі разом вони становлять понятєвий базис ERM-моделі.

Відповідно, поняття «клас» є узагальненням понять «безліч сутностей», «безліч зв'язків» і «безліч значень». А поняття «реляційне відображення» та «атрибутне відображення» є спеціальні види відображень. Вочевидь, на відміну перших двох остання спеціалізація неповна.

Слід зазначити, що поряд з основними структурними базовими поняттями – «клас» та «відображення» – для більш повного вираження семантики предметної області використовуються додаткові базові поняття – спеціалізація та категоризація, ролі об'єктів-прообразів та об'єктів-образів в екземплярах відображень, а також операції та відносини (непересічності, включення та рівності – для класів, несумісності, наслідування та еквівалентності – для відображень), визначені на множинах класів та відображень.

Пропонований підхід до вибору структурних понять моделі (абстрактних базових і простіших похідних) дозволяє при формалізації предметної області оперувати в основному знайомими поняттями («сутність», «зв'язок»), вдаючись до використання нових форм («відображення») лише у разі недостатньої виразності перших. У деяких випадках проєктування схем даних використання понять «клас» та «відображення» вдається уникнути зовсім.

ER-нотація менш виразна, ніж ORM, щодо охоплення обмежень та бізнес-правил. Відмова ER- та UML-діаграм включати стандартні нотації для багатьох важливих ORM-обмежень ускладнює створення всебічної моделі або виконання перетворень моделі [2].

Відмінною особливістю апарату обмежень цілісності ERM-моделі є те, що на відміну від OR-моделі, в ній пропонується обмежений універсальний набір конструкцій, за допомогою яких можна представити багато (якщо не всі) закони взаємини даних. Принаймні всі типи обмежень цілісності OR-моделі можна виразити запропонованими в ERM-моделі засобами.

Проведений аналіз показує, що, розуміючи важливість синтезу потужної семантичної моделі і розглядаючи, як зразок виразності природну мову, автори ORM- і ERMM-моделей йдуть різними шляхами. Багаті виразні можливості ERM-моделі – не результат механістичного винаходу спеціальних конструкцій для кожної характерної риси даних. Основу ERM-моделі становлять універсальні поняття логіки та математики, поєднуючи які, як і у природній мові, досягається вся її виразна міць.

Список літератури

1. Halpin T., Morgan T. (2008) Information Modeling and Relational Databases, Second Edition. Morgan Kaufman. 943 p.
2. Halpin T. (2010) Object-Role Modeling: Principles and Benefits. International Journal of Information Systems Modeling and Design. No. 1(1). 32–54.
3. Halpin T. (2009) Object-Role Modeling. Encyclopedia of Database Systems. Springer. 1941–1946.
4. Cuyler D., Halpin T. (2005) Two Meta-Models for Object-Role Modeling. Information Modeling Methods and Methodologies. Idea Publishing Group, Hershey PA, USA. 17–42.

УДК 004.42

О.С. Савельєв, І.О. Розломій
saveliev.oleksandr1119@vu.cdu.edu.ua, inna-roz@ukr.net
Черкаський національний університет імені Богдана Хмельницького, м. Черкаси

СУЧАСНІ ФРЕЙМВОРКИ ДЛЯ РОЗРОБКИ ANDROID-ДОДАТКІВ

На сьогоднішній день мобільні програми знаходяться на піку своєї популярності. Кількість розробників мобільних додатків та доступних програм стрімко зростає. У зв'язку з цим збільшується кількість кросплатформних мобільних фреймворків для успішного створення програм. Повсюдно мобільні програми використовують дані із супутників, при цьому фреймворк дозволяє зробити спілкування з модулями супутникового зв'язку набагато зручнішим, оскільки не потрібно писати окремі нативні програми для кожної платформи [1].

Повний спектр обчислювальних сервісів і багаті функціональні можливості Android дозволяють створювати додатки, які далеко виходять за межі лише сегменту мобільних телефонів. Операційна система Android стає все більш популярною та в інших платформах та додатках.

Розробка програм під ОС Android є найбільш актуальним процесом у розробці мобільних програм. Тому актуальною є і тема роботи, пов'язана з аналізом та використанням інструментів програмування для ОС Android. Станом на січень 2022 року 69,74% від усіх мобільних операційних систем займає Android [2]. Сучасний розвиток тенденцій спонукує використовувати все новіші фреймворки для розробки Android додатків. Розглянемо деякі найкорисніші фреймворки для розробки додатків.

1. Sencha Touch – це фреймворк для розробки, який використовує JavaScript і HTML5, що дозволяє створювати динамічні та комплексні програми. Sencha Touch забезпечує високу продуктивність, оскільки включає в себе методи апаратного прискорення. Даний фреймворк допомагає розробляти захоплюючі та привабливі мобільні додатки з плавною анімацією та прокручуванням. Він також підтримує вбудований інтерфейс користувача та тестування в реальному часі [3].

2. TheAppsBuilder – ще один унікальний новий фреймворк Android, що базується на HTML і підтримує безкодовий інтерфейс користувача. TheAppBuilder має вбудовані блоки, які включають такі функції, як зворотний зв'язок, оновлення вмісту, опитування, push-повідомлення та багато іншого.

На даний момент існує сім популярних кросплатформових мобільних фреймворків, таких як Appcelerator Titanium, Kony Platform, Adobe PhoneGap, IBM Worklight, Telerik Platform, Verivo Akula та Xamarin [2]. Але тільки Xamarin створює програми за допомогою мови C #.

3. Xamarin використовується для розробки додатків Android, використовуючи «.NET». Він відносно старіший за більшість інших кросплатформних фреймворків. Xamarin має розширений набір інструментів, за допомогою якого розробники можуть працювати з кодом, написаним на C# або XAML. Ця функція спільного використання коду скорочує час кодування для розробників і кількість помилок під час процесу розробки.

Xamarin – це фреймворк для кросплатформної розробки мобільних додатків (iOS, Android, Windows Phone) з використанням мови C#. Xamarin заснований на open-source реалізації платформи .NET – Mono. Код програми пишеться широко використовуваною мовою програмування C# із застосуванням всіх звичних мовних особливостей, наприклад, LINQ, лямбда-виразів, Generic і Async. При цьому є повний доступ до всіх можливостей SDK платформи та механізму створення UI (user interface), отримуючи на виході додаток, який нічим не відрізняється від native-додатків і не поступається ним у продуктивності [4-5].

Фреймворк Xamarin складається з кількох основних частин:

- Xamarin. IOS – бібліотека класів, що надає розробнику доступ до iOS SDK;
- Xamarin. Android – бібліотека класів, що надає доступ до Android SDK;
- компілятори для iOS та Android;
- IDE Xamarin Studio;
- плагін для Visual Studio.

Для кожної платформи Xamarin надає можливість використовувати нативні засоби розробки UI та нативні елементи інтерфейсу користувача. Для Android створення UI може відбуватися безпосередньо в коді або за допомогою декларативного підходу з описом інтерфейсу в XML. Для iOS це також код, або використання нативних засобів проектування інтерфейсу.

Для кожної з платформи потрібно реалізувати власний шар UI, тобто код, який відповідає за зовнішній вигляд програми, доведеться написати для кожної платформи окремо. Якщо розбивати програму на шари, то виходить така схема:

- Data Layer (DL) – сховище даних, наприклад, база SqlLite або xml-файли;
- Data Access Layer (DAL);
- Business Layer (BL) – шар , що містить бізнес-логіку додатку;
- Service Access Layer (SAL) – шар, що відповідає за взаємодію Космосу з віддаленими сервісами;
- Application Layer (AL) – шар, що містить платформозалежний код;
- User Interface Layer (UI) – шар інтерфейсу користувача.

Кросплатформні є всі шари, розташовані вище Application Layer.

Розробники Xamarin як середовище розробки пропонують використовувати або власну IDE – Xamarin Studio або Visual Studio. Xamarin Studio – кросплатформова IDE , яка працює як на Mac OS X , і Windows. Також Xamarin пропонує можливість вести розробку у Visual Studio після встановлення спеціального плагіна [4-5].

Розглядаючи способи створення програм для iOS і Android, багато хто вважає, що єдиним варіантом є машинні мови Objective – C, Swift і Java. Однак протягом кількох останніх років з'явилася нова екосистема платформ для розробки мобільних додатків.

Xamarin є унікальним у цьому просторі, оскільки пропонує одну мову C#, бібліотеку класів та середовище виконання, що працює на всіх трьох платформах для мобільних пристроїв – iOS, Android та Windows Phone і підтримує компіляцію власних (без інтерпретації) додатків, що досить швидко діють навіть для ресурсомістких ігор.

Кожна з цих платформ має окремий набір функцій і пропонує різні можливості створення власних програм, які компілюються в машинний код і узгоджено взаємодіють з базовою підсистемою Java. Наприклад, деякі платформи допускають розробку програм тільки на HTML і JavaScript, тоді як інші працюють на дуже низькому рівні і підтримують лише код на C або C++. Деякі платформи взагалі не використовують власний набір засобів управління.

4. Google Flutter передбачений для створення кросплатформних програм. Фреймворк написаний мовою Dart і використовує єдину кодову базу. Цей провідний фреймворк Android має новітній підхід до розробки додатків. Він спрощує багатоплатформний процес розробки для створення високоякісних інтерфейсів для Android та iOS. Фреймворк Flutter має значну перевагу від механізму 2D рендеринга під назвою Skia [4]. Flutter є надійною платформою тестування для виконання UI, модульних і функціональних тестів. Має багато переваг, як швидке відтворення, зчитування з екрана, велика кількість тем та багато іншого.

5. Appcelerator Titanium SDK дозволяє розробникам створювати власні програми за допомогою єдиної кодової бази JavaScript. В результаті Appcelerator довелося потрапити до списку нових фреймворків Android для розробки високоефективних та надійних додатків. Завдяки фреймворку можна створювати багатоплатформні програми. Крім того, він надає доступ до API для операційних систем, включаючи Android, iOS, Universal Windows, HTML5 і Blackberry.

6. Ionic допомагає розробникам створювати сучасні гібридні додатки з використанням HTML5, CSS3 і JavaScript на кількох платформах. Ionic включає в себе безліч програмних інструментів та анімацій. Ці ресурси сприяють розробці сучасних мобільних додатків. Фреймворк містить простий CLI, що надає доступ до емуляторів, перезавантаження в реальному часі, журналювання тощо. Ionic досить легко інтегрується з іншими бібліотеками, такими як Cordova та AngularJS [6].

7. React Native використовує React, величезну бібліотеку JS. Міжплатформна характеристика дозволяє розробникам писати код лише один раз, а потім виконувати його в іншому місці. React Native має вбудовані компоненти інтерфейсу користувача та доступ до рідного API, що дозволяє додаткам Android мати гарний дизайн та високу продуктивність.

В статті було розглянуто сучасні фреймворки Android, які використовуються для розробки мобільних додатків. Найбільша увага зосереджена на розгляд кросплатформного фреймворку Xamarin, оскільки він є одним з найбільш використовуваних інструментів.

Список літератури

1. Ескендир М.А. (2019) Введение в разработку мобильных приложений. Вестник магистратуры. №6(93). 33–36.

2. 15 Apps for Programming on Android – [Електронний ресурс] – Режим доступу до ресурсу: http://android.appstorm.net/roundups/developer/15-apps-forprogramming-on-android/ (Дата звернення: 25.04.2022).

3. Sencha Touch 2 Developer Preview [Електронний ресурс]. – Режим доступу до ресурсу: https://www.sencha.com/blog/sencha-touch-2-developer-preview/ (Дата звернення: 25.04.2022).

4. Robert C. Martin (2017) Clean Architecture. A Craftsman's Guide to Software Structure and Design, НьюДжерси: Прентис Холл. 432 с.

5. Шматко А.В., Федорченко В.Н. (2016) Обзор и анализ инструментов разработки мобильных приложений для ОС Android. Инновации в науке. № 5(54). 59–73.

6. Ionic Docs Core Concepts [Електронний ресурс]. – Режим доступу до ресурсу: https://ionicframework.com/docs/core-concepts/fundamentals (Дата звернення: 25.04.2022).

УДК 004.05

аспірант М.О., Кобець, к.т.н., доц. А.С Коваленко. д.т.н., доц. О.В. Коваленко
nicko9298@gmail.com, annasun911@gmail.com, dr.kovalenkoov@gmail.com
Центральноукраїнський національний технічний університет, м. Кропивницький

МЕТОД РОЗРАХУНКУ ЧИСТОЇ ПОТОЧНОЇ ВАРТОСТІ ІНТЕГРОВАНИХ ІНФОРМАЦІЙНИХ СИСТЕМ З ВРАХУВАННЯМ ПОТРЕБ СТАНДАРТУ ISO/IEC 29110

На теперішній час у більшості організацій різних форм власності все більше приділяють увагу питанням аналізу, розрахунку та оцінки вартості програмно-апаратних компонентів інтегрованих інформаційних систем.

Існуючі підходи розрахунків у більшості ґрунтуються на стандарті ISO/IEC/IEEE 12207 [1] який розроблено для середніх та великих систем з відповідними початковими константами – підприємство, організація чи відділ, у якому працює понад 25 осіб персоналу.

Більшість малих організацій не можуть дозволити собі такі ресурси з точки зору кількості співробітників, досвіду, бюджету та часу, а також не бачать чистої вигоди від створення надскладних програмно-апаратних систем чи процесів життєвого циклу. Для вирішення деяких із цих труднощів було розроблено набір стандартів ISO/IEC 29110 [2], заснований на наборі характеристик дуже малих організацій (Very Small Entities, VSEs).

Стандарт ISO/IEC 29110 заснований на підмножині відповідних стандартних процесів, дій, завдань та результатів, які називаються профілями з урахуванням використання з будь-якими типами життєвих циклів розробки, таких як каскадний, ітеративний, інкрементний, еволюційний або гнучкий.

На основі методу оцінки показника чистої приведеної вартості проекту розроблення безпечного ПЗ [3-5] було розроблено метод розрахунку чистої поточної вартості інтегрованих інформаційних систем та компонент з врахуванням потреб стандарту ISO/IEC 29110. Приклад використання зображено на рисунку 1, з якого можна побачити знайдені оптимальні значення чистої поточної вартості на основі положення нечітко-множинної теорії.

Розроблений удосконалений метод дозволяє використовувати стандарти серії ISO/IEC 29110 у організаціях чий відділ розробників не перевищує значення 20-25 осіб персоналу з урахуванням часткової невизначеності складових проекту та витрат різного призначення від технологічних етапів проекту, на основі теорії нечітких множин.

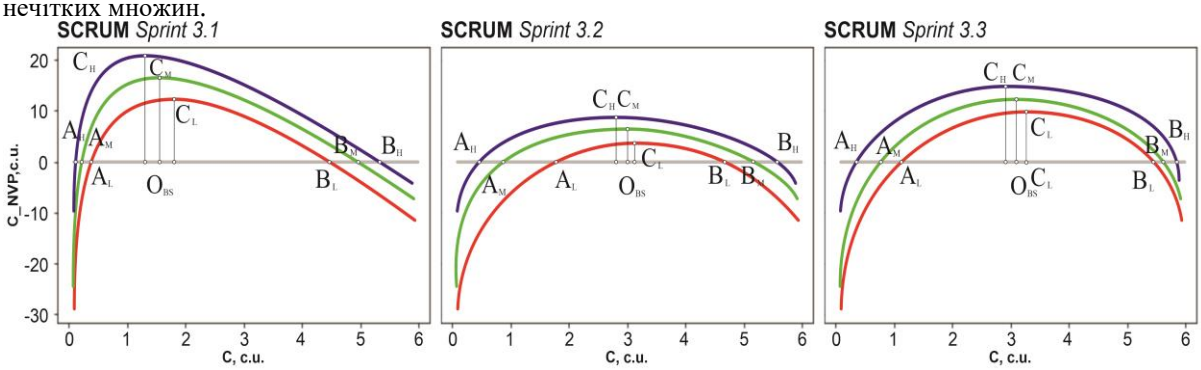

Рис. 1. Приклад застосування розробленого методу розрахунку чистої поточної вартості інтегрованих інформаційних систем з врахуванням потреб стандарту ISO/IEC 29110 з використанням гнучкого типу життєвого циклу

Список літератури

1. ISO/IEC/IEEE 12207:2017. Systems and software engineering - Software life cycle processes. URL: https://www.iso.org/standard/63712.html

2. ISO/IEC CD 29110-1. Systems and software engineering – Lifecycle profiles for Very Small Entities (VSEs) – Part 1: Overview. URL: https://www.iso.org/standard/83636.html

3. Коваленко О.В. Моделі та методи розроблення програмного забезпечення комп'ютерних систем для підвищення безпеки даних: монографія / О.В. Коваленко // К.: Вид. «КОД» – 2019. – 295 с.

4. Коваленко А.С., Коваленко О.В., Смірнов О.А., Доренський О.П. Удосконалення методу технічного обслуговування об'єктів інтегрованої інформаційної системи. Системи озброєння і військова техніка. – Випуск 2(46) – Х.: ХУПС – 2016. – С. 106-111. Режим доступу: http://nbuv.gov.ua/UJRN/soivt_2016_2_22 (Фахове видання. Категорія «Б»)

5. Коваленко О.В. Методи якісного аналізу та кількісної оцінки ризиків розробки програмного забезпечення. Системи управління, навігації та зв'язку. – Випуск 3 (49). – Полтава: ПолтНТУ. – 2018. – С. 116-125. Режим доступу: http://journals.nupp.edu.ua/sunz/article/view/1146 (Фахове видання. Категорія «Б»)

УДК 004.44

В.В.Кіш, Н.І.Йовбак
kish.viktor@student.uzhnu.edu.ua, yovbak.nika@student.uzhnu.edu.ua
ДВНЗ »Ужгородський національний університет», м.Ужгород

VUE ROUTER – МАРШРУТИЗАЦІЯ VUE

Vue router підтримує зв'язування вкладенних шляхів зі вкладеними компонентами і пропонує деталізований контроль над переходами. Vue дозволяє створення додатків за допомогою компонентів.

Якщо додати Vue router до цього, все що потрібно зробити – це зв'язати ваші компоненти з роутами і дозвольте Vue router вирішувати, де їх рендерити.

Розглянемо приклад реалізації роутеру на прикладі роутингу в додатку зі станціями поїздів (Home, About, StationForm):

```
import { createRouter, createWebHashHistory } from 'vue-router'
import Home from '../views/Home.vue'
import About from '../views/About.vue'
import StationForm from '../views/StationForm.vue'


const routes = [
//route to home page
  {
    path: '/',
    name: 'Home',
    component: Home
  },
//route to about page
  {
    path: '/about',
    name: 'About',
    component: About
  },
//route to station info form
  {
    path: '/station-form',
    name: 'StationForm',
    component: StationForm
  },
{
    path: '/login',
    name: 'LoginPage',
    component: 'LoginPage'
  },

{
    path: '/about',
    name: 'AboutPage',
    component: ' AboutPage '
  },

{
    path: '/contact',
    name: 'ContactPage',
    component: ' ContactPage '
  },

]

//creating router
const router = createRouter({
```

```
  history: createWebHashHistory(),
  routes
})

export default router
```

Розглянемо ще один приклад маршрутизації додатку з мультфільмами:

```
import { createRouter, createWebHashHistory } from 'vue-router'
import Home from '../views/Home.vue'
import AnimeDetail from '../views/AnimeDetail.vue'

const routes = [
  {
    path: '/',
    name: 'Home',
    component: Home
  },
  {
    path: '/anime/:id',
    name: 'Anime Details',
    component: AnimeDetail
  },

{
    path: '/login',
    name: 'LoginPage',
    component: 'LoginPage'
  },

{
    path: '/about',
    name: 'AboutPage',
    component: ' AboutPage '
  },

{
    path: '/contact',
    name: 'ContactPage',
    component: ' ContactPage '
  },


]

const router = createRouter({
  history: createWebHashHistory(),
  routes
})

export default router
```

Отже, у нас є приклад реалізації роутеру, який містить три шляхи.

За допомогою Vue router ми можемо переходити між різними сторінками без перезавантаження сторінки. Це відповідає головному принципу SPA, до якого належить Vue та Vue router в часності.

Список літератури

1. Документація Vue router [Електронний ресурс] – Режим доступу: https://vuejs.org/guide/scaling-up/routing.html

УДК 004.67

М.Л. Іванов, ст. 1 курсу
ivml@gmail.com
Науковий керівник В.С. Гермак*, викладач*
Центральноукраїнський національний технічний університет, м. Кропивницький

ПОРІВНЯЛЬНИЙ АНАЛІЗ МЕТОДІВ ЧИСЕЛЬНОГО ДИФЕРЕНЦІЮВАННЯ ФУНКЦІЙ

Диференціювання – це одна з найважливіших математичних операцій. Без неї неможливо обійтися при дослідженні функцій, при розв'язанні задач на екстремуми. Чисельне диференціювання функцій широко застосовується у тих випадках, коли функцію важко або навіть неможливо продиференціювати аналітично. Наприклад, необхідність у чисельному диференціюванні виникає у тому випадку, коли функція задана не аналітично, тобто формулою, а таблично або за допомогою алгоритму обчислення функції у довільній точці. Крім того, формули чисельного диференціювання широко використовуються при розробці обчислювальних методів розв'язання багатьох задач, наприклад, під час розв'язання диференціальних рівнянь, пошуку розв'язків систем нелінійних рівнянь, пошуку точок екстремуму функцій цій і т.д.

В залежності від особливостей поставленої задачі та від вимог до її розв'язання можуть бути використані різні методи чисельного диференціювання.

У випадку, коли функція має складне аналітичне вираження, зазвичай застосовують методи, засновані на використанні інтерполяційних поліномів. Для цього достатньо замінити початкову функцію її інтерполяційним поліномом, а потім продиференціювати його. Окрім інтерполювання в чисельному диференціюванні застосовуються ще сітчасті методи. При цьому обчислюються значення похідних у вузлах сітки на скінченому відрізку.

1. Інтерполяційні методи чисельного диференціювання. Ці методи відрізняються інтерполяційною функцією, закладеною в основу метода. В такому випадку в основі методу часто лежить інтерполяційний поліном Ньютона або інтерполяційний поліном Лагранжа. При цьому вузли таблично заданої функції можуть бути рівновіддаленими або не рівновіддаленими. В залежності від цього можна використовувати поліном Ньютона для рівновіддалених або нерівновіддалених вузлів.

a. Різницеві методи. При використанні інтерполяційного полінома Ньютона прирости аргумента (різниці) можуть відкладатися вправо, вліво або в обидва боки від досліджуваної точки. Недоліком цього методу, який ще має назву метод скінчених різниць, є досить низька точність. При цьому метод двосторонніх (або центральних) різниць має більш високу точність. Ще одним недоліком різницевих методів чисельного диференціювання є те, що класичні наближення скінченими різницями містять неусувну похибку і є погано обумовленими, тобто дуже чутливими до коливань вхідних змінних.

b. Окрім різницевих методів чисельного диференціювання виділяють і безрізницеві методи. Вони використовуються тоді, коли зручніше мати справу не з скінченими різницями функції, а з її значеннями в певних точках. В такому випадку зручно користуватись інтерполяційним поліномом Лагранжа.

2. Метод невизначених коефіцієнтів. Цей метод часто використовують у випадку довільного розташування вузлів. Також він використовується в багатовимірному випадку, коли побудувати інтерполяційний многочлен досить складно. Він дозволяє уникнути громіздких виразів і обчислити значення похідних будь якого порядку у вузлах інтерполяції.

3. Сітчастий метод диференціювання. Цей метод дозволяє наближено обчислити похідну будь-якого порядку у вузлах сітки з заданою точністю. Використовується цей метод для достатньо гладких функцій.

4. Чисельне диференціювання сплайнами. Цей метод полягає у диференціюванні сплайну з мінімальною нормою похідної, яка апроксимує цю функцію. При використанні даного методу треба враховувати необхідність задання крайових умов. Також треба враховувати, що сплайн володіє властивістю появи коливань, амплітуда яких росте по мірі віддалення від границі. Чим з більшою похибкою задана похідна на границі, тим раніше виникають коливання в сплайн-функції. Щоб зменшити похибку, яка виникає при цьому, має сенс розбити область на декілька підобластей і на кожній з них будувати свій сплайн з мінімальною нормою похідної.

Таким чином, можна зробити висновок, що, незважаючи на велике різноманіття методів чисельного диференціювання функцій не можна однозначно зупинитись на якомусь одному з методів. Універсальних методів диференціювання не існує. Для того, щоб коректно та швидко розв'язати поставлену задачу, треба її всебічно проаналізувати, виділити ті вимоги до розв'язку, які є ключовими. І лише після цього можна приступити до вибору оптимального для заданих умов методу.

УДК 004.67

Е.В. Ісаченков, ст. 1 курсу
isev@gmail.com
Науковий керівник В.С. Гермак*., викладач*
Центральноукраїнський національний технічний університет, м. Кропивницький

КЛАСИФІКАЦІЯ НАБЛИЖЕНИХ МЕТОДІВ РОЗВ'ЯЗАННЯ ДИФЕРЕНЦІАЛЬНИХ РІВНЯНЬ

Наряду з багатьма іншими розділами математики, диференціальні рівняння мають широке практичне застосування. Вони використовуються для математичного опису різноманітних природних явищ, формулювання фізичних законів, дослідження закономірностей розвитку популяцій в біології та генетиці, моделюванні економічних та соціальних процесів та інше.

Різноманітність реальних процесів породжує значну кількість прикладних задач, які зводяться до диференціальних рівнянь в звичайних і в часткових похідних. Їх точний розв'язок можливий лише у виняткових випадках. Цей факт спричиняє необхідність наближеного розв'язку таких задач.

В наш час створено і відпрацьовано значну кількість чисельних методів розв'язання диференціальних рівнянь. Однією з класичних задач є задача Коші для диференціальних рівнянь і звичайних похідних. На її прикладі можна розглянути різноманіття чисельних методів розв'язання диференціальних рівнянь (табл. 1).

Наближені методи розв'язання диференціальних рівнянь можна розділити на два класи: аналітичні та чисельні. Аналітичні методи дозволяють отримати наближений розв'язок в аналітичному вигляді, чисельні – у вигляді значень функції в зарані обраних вузлах.

Таблиця 1

Наближені методи розв'язання диференціальних рівнянь

Аналітичні методи		
№ п/п	Назва методу	Особливості методу
1.	Метод послідовних наближень (метод Пікара)	Ґрунтується на інтегруванні обох частин рівняння. Має два суттєвих недоліки, що обмежують його застосування на практиці: - потрібно встановлювати його збіжність та оцінювати швидкість збіжності; - вимагає проведення інтегрування
2.	Метод степеневих рядів	Ґрунтується на розкладанні розв'язку в ряд Тейлора за умови, що права частина є досить гладкою функцією. Рідко застосовується на практиці, так як при значному порядку є досить громіздким і може містити доволі значну похибку.
Чисельні методи		
1.	Метод Рунге-Кута	Однокроковий метод. Полягає в послідовному обчисленні шуканої функції за певною розрахунковою формулою. Похибка цього методу на порядок нижча похибки розрахункової формули.
2.	Метод Адамса	Багатокроковий метод. Обчислення шуканої функції в певній точці залежить відзначення функції в попередніх точках. Розрізняють екстраполяційний та інтерполяційний методи Адамса. Похибка розрахункової формули інтерполяційного метода на порядок вище, ніж екстраполяційного.
3.	Побудова обчислювальних схем на основі принципу послідовного уточнення результату.	Для застосування цього методу функція повинна бути досить гладкою. Дозволяє отримати розв'язок з будь-якою потрібною точністю.

На сучасному етапі жоден серйозний практичний проект чи теоретичне дослідження не обходиться без математичного та імітаційного моделювання, які, в свою чергу, не можна реалізувати без складання та розв'язання диференціальних рівнянь. З огляду на таке широке поле застосування диференціальних рівнянь при виборі методів розв'язання треба спиратись на специфіку задачі, яка розв'язується.

УДК 004.67

Я.О. Козлов, ст. 1 курсу
kozya@gmail.com
Науковий керівник В.С. Гермак. *викладач*
Центральноукраїнський національний технічний університет, м. Кропивницький

ПОРІВНЯЛЬНИЙ АНАЛІЗ МЕТОДІВ ЧИСЕЛЬНОГО ІНТЕГРУВАННЯ ФУНКЦІЙ

Інтегральне числення відіграє значну роль у процесах моделювання різноманітних реальних систем, в тому числі соціальних та економічних. Також інтегральне числення надає широкий математичний апарат для дослідження та прогнозування цих процесів. В зв'язку з тим, що математичні моделі реальних процесів є досить складними, аналітичне інтегрування, яке зводиться до знаходження первісної функції, часто є надто ускладненим або взагалі неможливим. В цьому випадку на допомогу приходять чисельні методи, що дозволяють знайти наближене значення інтегралу. Чисельні методи інтегрування в сукупності з сучасними програмними засобами та обчислювальною технікою роблять інтегрування найскладніших функцій тривіальною задачею. Також чисельні методи інтегрування стають в нагоді, коли функція задана неявно, набором своїх значень на дискретній множині точок.

Наразі існує велика кількість алгоритмів чисельного інтегрування (табл. 1). Кожен з цих методів має свої переваги та недоліки а також свою сферу застосування.

Таблиця 1
Методи наближеного обчислення інтегралів

№ п/п	Назва методу	Характеристичні особливості методу
1.	Методи Ньютона-Котеса	Методи засновані на апроксимації функції *f(x)* поліномом степеня *n*. Алгоритми цього класу відрізняються тільки степенем полінома. Як правило, вузли апроксимуючого полінома – рівновіддалені. Дані методи дають високу точність обчислень, коли степінь апроксимуючого полінома спів розмірна з кількістю інтервалів розбиття.
1.1.	Метод прямокутників	Функція апроксимується на інтервалах розбиття поліномами нульового степеня. Розрізняють три варіанти наближення прямокутниками: наближення лівими прямокутниками; наближення правими прямокутниками; наближення середніми прямокутниками. Ці методи фактично заміняють інтеграл верхніми і нижніми сумами Дарбу і дають значну похибку.
1.2.	Метод трапецій	Функція апроксимується на інтервалах розбиття поліномами першого степеня.
1.3.	Метод парабол (Симпсона)	Функція апроксимується на інтервалах розбиття поліномами другого степеня. Метод дає змогу знайти точне значення інтеграла у всіх випадках, коли функція *f(x)* многочлен, степінь якого менше або дорівнює трьом.
2.	Методи сплайн-інтегрування.	Методи сплайн-інтегрування базуються на апроксимації функції *f(x)* сплайном. Методи цього класу відрізняються за типом вибраних сплайнів і використовуються в основному в тих випадках, коли алгоритми аппроксимації сплайнами застосовуються такж і для обробки даних.
2.1.	Сплайн-квадратура	На кожному з інтервалів розбиття підінтегральна функція заміняється кубічним сплайном.
3.	Методи Монте-Карло.	Методи Монте-Карло використовують найчастіше при обчисленні кратних інтегралів, вузли вибирають випадковим чином, розв'язок носить імовірнісний характер.

Таким чином при здійсненні операції чисельного інтегрування при виборі методу інтегрування слід враховувати умови задачі, обчислювальну складність методу і поставлені вимоги до точності розв'язку та часового проміжку, відведеного на пошук цього розв'язку.

УДК 004.67

Д.А. Ходаковський, ст. 1 курсу
khodda@gmail.com
Науковий керівник В.С. Гермак *викладач*
Центральноукраїнський національний технічний університет, м. Кропивницький

ПОРІВНЯЛЬНИЙ АНАЛІЗ МЕТОДІВ РОЗВ'ЯЗАННЯ СИСТЕМ ЛІНІЙНИХ РІВНЯНЬ

Розв'язання систем лінійних алгебраїчних рівнянь - одне з основних завдань обчислювальної лінійної алгебри. Багато задач управлінського, економічного, фізичного, електротехнічного, технологічного характеру моделюються за допомогою систем лінійних алгебраїчних рівнянь або зводяться до них. Хоча завдання розв'язання саме системи лінійних рівнянь порівняно рідко представляє самостійний інтерес для прикладних задач, але від уміння ефективно розв'язувати дані системи часто залежить сама можливість математичного моделювання найрізноманітніших процесів із застосуванням ЕОМ. Значна частина чисельних методів розв'язання різних (особливо - нелінійних) задач включає в себе розв'язання систем лінійних рівнянь як елементарний крок відповідного алгоритму.

На даний час існує багато алгоритмів розв'язання систем лінійних алгебраїчних рівнянь, які дозволяють знайти як аналітичні (точні) розв'язки, так і чисельні (наближені) (табл. 1).

Таблиця 1

Методи розв'язання систем лінійних алгебраїчних рівнянь

№ п/п	Назва методу	Принцип реалізації	Особливості застосування
1.	Метод виключення Гауса	Метод виключення Гауса для розв'язування лінійних систем рівнянь є основою багатьох обчислювальних схем і зводиться до перетворення початкової системи до рівносильної системи з верхньотрикутною матрицею.	Необхідною й достатньою умовою застосовності методу є не рівність нулю всіх «головних» елементів»..
2.	Метод Гауса-Жордана	Суть методу – приведення матриці початкової системи до діагонального вигляду шляхом перетворення коефіцієнтів рівнянь, розташованих вище й нижче головного рівняння.	Застосування методу ускладнюється, якщо в будь-якому рівнянні «головний» елемент дорівнює нулю. Однак ці труднощі можна обійти, змінивши порядок розташування рівнянь системи. Найбільша точність досягається тоді, коли «головний» елемент має найбільше значення. Тому рядок з нульовим або малим головним елементом потрібно замінити на той з нижніх рядків, у якому у тому ж стовпці розміщений елемент, який має найбільше значення.
3.	Метод квадратного кореня	Процес знаходження розв'язку можна розбити на три етапи: Знайти матрицю S таку, що $S^TS=A$; Знайти вектор y, що відповідає умові:$S^Ty=b$; Знайти вектор x з умови Sx=y.	Метод застосовується тільки при виконанні певних умов: матриця системи повинна бути не виродженою; матриця системи повинна бути симетричною; щоб не виконувати обчислень із комплексними числами, матриця повинна бути додатньо визначена, тобто всі її головні мінори повинні бути додатніми.
4.	Метод прогонки	Складається з прямої і зворотної прогонки.	Застосовується для систем з розрідженою матрицею, яка містить багато нульових елементів.
5.	Метод простої ітерації (метод Якобі)	Дозволяє одержати розв'язок у вигляді границі послідовності векторів.	Застосовується при великій кількості невідомих системи, коли обчислити точний розв'язок стає досить складно. Може застосовуватись лише за умови збіжності ітераційного процесу.
6.	Метод Гауса-Зейделя	Є різновидом методу простої ітерації.	Застосовується при тих же умовах, що і метод простої ітерації, але має менший час виконання.

Отже, незважаючи на удавану простоту задачі розв'язання системи лінійних алгебраїчних рівнянь, треба обґрунтовано підходити до вибору методу розв'язання. Лише в цьому випадку буде отримано оптимальний розв'язок за мінімальний можливий час.

УДК 004.67

В.С. Гермак, К.О. Буравченко
germak_vs@ukr.net, buravchenkok@gmail.com
Центральноукраїнський національний технічний університет, м. Кропивницький

ПОРІВНЯЛЬНИЙ АНАЛІЗ МЕТОДІВ РОЗВ'ЯЗАННЯ СИСТЕМ НЕЛІНІЙНИХ РІВНЯНЬ

Задачі розв'язання систем нелінійних рівнянь часто зустрічаються на практиці. Наприклад, розв'язання нелінійних крайових завдань (як звичайних диференціальних рівнянь, так рівнянь з частковими похідними) методом скінчених різниць, і навіть мінімізація функцій багатьох змінних чисельними методами зводяться до розв'язання систем нелінійних рівнянь. Більшість нелінійних рівнянь та систем нелінійних рівнянь, які зустрічаються при вирішенні практичних задач, неможливо розв'язати аналітично, в явному вигляді. У таких випадках застосовуються чисельні методи. Більш того, багато з тих задач, які можна вирішити аналітичними методами нерідко набагато швидше і ефективніше вирішуються чисельними методами з необхідною точністю.

В зв'язку з цим до нашого часу було розроблено значну кількість чисельних методів розв'язання систем нелінійних рівнянь. Розглянемо основні з них(табл. 1).

Таблиця 1

Методи розв'язання систем нелінійних рівнянь

№ п/п	Назва методу	Особливості застосування
1.	Метод простої ітерації (метод Якобі)	Для реалізації цього методу стосовно заданої нелінійної системи рівнянь треба виконати наступну послідовність кроків: шляхом алгебраїчних перетворень виокремити з кожного рівняння по одній змінній; обрати вектор початкового наближення; підставити вектор початкового наближення в систему; з першого рівняння обчислити нове наближення до першої змінної, з другого – до другої змінної тощо; обчислені уточнені значення змінних знову підставляти у рівняння; перевірити умову завершення ітераційного процесу. Важливо забезпечити збіжність ітераційного процесу, збіжність методу залежить від вигляду функцій, які входять в систему, а точніше від матриці, що складена з часткових похідних. Ітераційний процес сходиться, якщо сума модулів кожного її рядка менша одиниці в деякому околі кореня.
2.	Метод Зейделя	Метод Зейделя для систем нелінійних рівнянь, як і для систем лінійних рівнянь, полягає у використанні уточнених значень змінних уже на поточному ітераційному кроці. Умова завершення процесу розв'язання системи нелінійних рівнянь за методом Зейделя збігається з умовою для методу простих ітерацій.
3.	Метод Ньютона	Основна ідея методу Ньютона полягає у виділенні з рівнянь системи лінійних частин, які є головними при малих приростах аргументів. Це дозволяє звести початкову задачу до розв'язання послідовності систем лінійних рівнянь. Математичним підґрунтям методу є лінеаризація функцій шляхом розкладання в ряд Тейлора в околі точки початкового наближення до розв'язку системи рівнянь й нехтування всіма членами ряду, окрім лінійних щодо приростів змінних. В цьому методі для забезпечення збіжності також важливий правильний вибір початкового наближення.
4.	Метод якнайшвидшого спуску	Сутність методу якнайшвидшого спуску полягає в тому, що шукане рішення системи розглядається як мінімум деякої функції в n - мірному просторі, і цей мінімум шукається в напрямку, протилежному напрямку градієнта функції, тобто в напрямку якнайшвидшого зменшення цієї функції.
5/	Метод поділу навпіл (метод дихотомії)	Розв'язання системи рівнянь складається з двох етапів: відділення коренів, тобто відшукання досить малих областей, у кожній з яких міститься рівно один корінь системи; обчислення кожного відокремленого кореня із заданою точністю. Відділення коренів можна виконати графічно з супутнім аналізом на монотонність, зміну знака, опуклість функції.

Кожен з розглянутих методів має свої переваги та недоліки. Задача розв'язання системи нелінійних рівнянь є нетривіальною, вимагає досить значних обчислювальних ресурсів і процес та результат розв'язання в значній мірі залежить від практичної задачі, для вирішення якої була застосована система нелінійних рівнянь. В зв'язку з цим вибір методу розв'язання вимагає обдуманого, виваженого підходу, що дозволить отримати оптимальний результат.

УДК 004.4

В.В.Кіш, Н.І.Йовбак
kish.viktor@student.uzhnu.edu.ua, yovbak.nika@student.uzhnu.edu.ua
ДВНЗ »Ужгородський національний університет», м.Ужгород

VUE STORE НА ПРИКЛАДАХ

Vuex – централізоване сховище станів усіх маршрутів та компонентів, що використовуються у web-додатку. Кожен із них буде запитувати дані у «State Management», а потім повертати назад змінені. Завдяки цьому зменшується кількість помилок та підвищується захищеність системи.

Розглянемо використання Vuex на прикладі створенні сховища з інформацією про поїздки та додавання нових поїздок:

```
import { createStore } from 'vuex'

export default createStore({
  state: {
    tripInfo: [
      {
        id: 1,
        tripStart: "Uzh",
        tripEnd: "Muk",
        tripDuration: "1 hour",
        tripPrice: 500,
        vehicle: "Bus"
      },
    ],

    favoriteTrips: [],
    searchT: "",
    sortInput: ""
  },

  mutations: {
    addTrip(state, trip) {
      state.tripInfo.push({
        tripStart: trip.start,
        tripEnd: trip.end,
        tripDuration: trip.dur,
        tripPrice: trip.price,
        vehicle: trip.veh
      })
    },

    updateTripInfo(state, list) {
      state.tripInfo = list
    },
    searchTitle(state, search) {
      state.searchT = search
    },
    sortValue(state, srt) {
      state.sortInput = srt
    }
  },
  actions: {
    addToFavorites(context, trip) {
      let fav = {
        id: trip.id,
        tripStart: trip.start,
        tripEnd: trip.end,
        tripDuration: trip.dur,
```

```
      tripPrice: trip.price,
      vehicle: trip.veh
     }
     context.commit("updateFavorites", fav)
     //console.log(context.state.favoriteTrips)
    },
  },
  getters: {
    AllTrips(state) {
     return state.tripInfo
    },
})
```

Розглянемо ще один приклад використання глобального сховища на прикладі додатка з пошуком мультфільмів:

```
import { createStore } from 'vuex'

export default createStore({
  state: {
    animeInfo: {},
    animeList: {},
    search_query: ''
  },

  mutations: {
    updateAnime(state, anime) {
      state.animeInfo = anime
    },
  },
  actions: {
    fetchAnimeInfo(context, id) {
      fetch(`https://api.jikan.moe/v3/anime/` + id)
        .then((response) => response.json())
        .then((data) => {
          context.commit('updateAnime', data)
        });

    },

  getters: {
    anime(state) {
      return state.animeInfo
    },
    animelist(state) {
      return state.animeList
    }
  }
})
```

Отже, можемо зробити висновок, що Vuex є чудовим state менеджером, який здорово допомагає в роботі з даними та підтриманням визначеної структури зберігання даних. Для роботи у Vuex є визначені об'єкти для роботи з інформацією:

- state
- mutations
- getters
- actions

Завдяки такому розподілу опрацювання та зберігання інформації виходить на новий рівень.

Список літератури

1. Документація Vuex [Електронний ресурс] – Режим доступу: https://vuex.vuejs.org/

УДК 004.021

М.В. Панченко
panmv@gmail.com
Науковий керівник Гермак В.С., викладач
Центральноукраїнський національний технічний університет, м. Кропивницький

ПОРІВНЯЛЬНИЙ АНАЛІЗ АЛГОРИТМІВ ГЕНЕРУВАННЯ ПСЕВДОВИПАДКОВИХ ЧИСЕЛ

Випадкові числа в наш час мають широке практичне застосування. Ось лише невелика кількість сфер їх застосування: при комп'ютерному моделювання природних явищ чи систем випадкові числа дають змогу наблизити модель до реальності; випадкова вибірка варіантів для перевірки у випадку коли повна перевірка практично не здійсненна; при розв'язанні ряду складних завдань обчислювальної математики; в програмуванні випадкові значення є хорошим джерелом даних для перевірки ефективності алгоритмів; ігрові алгоритми; створення рандомізованого дизайну; прийняття рішень. Наразі також однією з основних сфер застосування випадкових чисел є криптографія. Випадкові числа застосовуються для генерування криптографічних ключів і генераторів сеансів.

До появи мережі Інтернет випадкові числа не використовувалися в критичних з погляду безпеки застосунках. Тому у генераторах використовувалася досить проста математична формула, що породжує послідовність елементів з початкового значення. .Така генерація в принципі передбачала повторюваність послідовності. Це були так звані некриптографічні генератори псевдовипадкових чисел. Коли ж постало питання забезпечення криптографічної безпеки, вимоги до генераторів псевдовипадкових чисел зросли. Постало питання унеможливити передбачення наступного згенерованого числа незважаючи на тривалість спостереження за роботою генератора. Це призвело до створення генераторів псевдовипадкових чисел криптографічної якості. Але іноді і цього класу надійності виявляється мало. Найбільш безпечними є ентропійні генератори або генератори істинно випадкових чисел. Їх робота ґрунтується на істинно випадкових процесах. Джерела справжніх випадкових чисел знайти дуже складно. Прикладами джерел випадкових чисел можуть бути такі фізичні шуми, як детектори іонізуючої радіації або космічне випромінювання, хоча практичне застосування таких джерел вкрай мале.

На практиці для генерації так званих псевдовипадкових чисел найчастіше використовують спеціальні алгоритми. Їхні властивості заздалегідь відомі і вони генерують послідовність чисел, яка з теоретичної точки зору не може бути випадковою. Основною характеристикою послідовності псевдовипадкових чисел є період повторення, який повинен бути більшим за інтервал, з якого беруться числа. Наразі на основі математичних досліджень та експериментів розроблена велика кількість алгоритмів, які дозволяють генерувати псевдовипадкові числа (табл. 1).

Таблиця 1
Алгоритми генерації псевдовипадкових чисел

п/п	Назва алгоритму	Особливості алгоритму
.	Лінійний конгруентний алгоритм	Швидко працюють і потребують мало операцій на біт послідовності. Основний недолік - передбачуваність таких послідовностей. Продовжують широко використовуватися для некриптографічних прикладних задач.
.	Серединиквадратів	Недолік полягає в тому, що послідовності, побудовані на основі цього методу мають тенденцію перетворюватися в короткі цикли повторюваних елементів.
.	Дробові генератори	Не знайшли широкого застосування. Для представлення ірраціональних чисел в пам'яті комп'ютера необхідна нескінченна кількість розрядів, а у випадку використання раціональних чисел дуже великою є ймовірність отримання циклів з малими періодами.
.	Генератор М - послідовностей	Проста програмна га апаратна реалізація і рівноймовірність згенерованих чисел. Не задовольняють вимогу щодо непередбачуваності послідовно згенерованих чисел, для усунення цього недоліку числа на виході додатково обробляються хеш-функцією або кодуються DES алгоритмом.
.	Генератори на основі різних математичних алгоритмів	Характеристики залежать від обраного алгоритму.

Генератори відрізняються один від одного оперативністю, доступністю еріодичністю та рівномірністю розподілу. Кожен з них має свої переваги та недоліки і свою сферу застосування. Відповідальний підхід до вибору алгоритму генерації псевдовипадкових чисел гарантує, що отримана послідовність буде задовольняти більшості тестів на випадковість.

СЕКЦІЯ 3. ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ В ЕКОНОМІЦІ, МЕДИЦИНІ ТА ОСВІТІ

УДК 004.45, 004.08

Ю.А.Лук'янчук [1], Н.В. Здолбіцька[1], Ю.С. Повстяна[3]
iuriilukianchuk87@gmail.com, , ninazdolb@gmail.com, yuliapovstyana@ukr.net
[1]к.т.н., ст.викладач кафедри «Комп'ютерні науки»,
[1]к.т.н., доцент кафедри «Комп'ютерні науки»,
[3]к.т.н., доцент кафедри «Інженерія програмного забезпечення»,
Луцький національний технічний університет, м.Луцьк

ЗАСТОСУВАННЯ ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ ДЛЯ РЕДАГУВАННЯ ВІЗУАЛЬНИХ ЗОБРАЖЕНЬ

Сьогодні компанії будь-якого профілю стикаються з необхідністю обробки великої кількості даних та зображень. Від якості і швидкості їх аналізу залежить ефективність прийняття рішень і рівень підтримки клієнтів. Впоратися зі зростанням інформаційних потоків допомагають технології штучного інтелекту. В їх основі лежать алгоритми глибокого машинного навчання, що поєднують в собі нейромережі певних типів.

Згідно із дослідженнями [1, 2] вважається, що штучні нейронні мережі (ANN) - це проривна технологія, однак на рівні алгоритмів вони були описані ще в другій половині минулого століття. Наразі з'явилася технічна можливість реалізувати їх в кінцевих продуктах - людство накопичило достатні обсяги інформації і створило засоби для її швидкого аналізу. До теперішнього часу найбільшого поширення набули такі види нейромереж, як CNN, що імітують роботу зорової кори головного мозку і частково виконують функцію абстрактного мислення. Вони прекрасно справляються із завданням розпізнавання зображень, а їх обчислення можна розподілити на графічних процесорах, що дозволяє створювати відносно дешеві апаратні платформи з елементами ШІ.

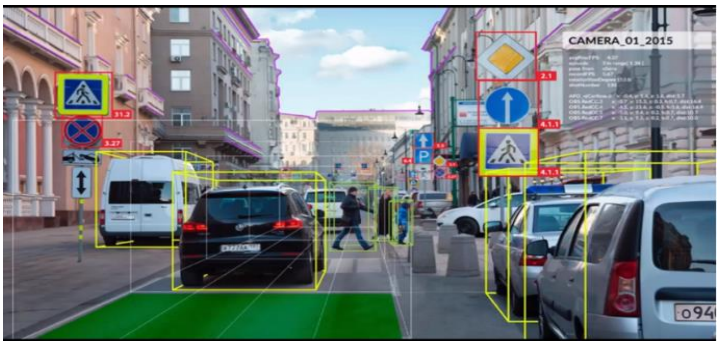

Рис. 1. Розпізнавання об'єктів за допомогою CNN

CNN застосовуються в системах машинного зору безпілотних автомобілів, комерційних дронів, роботів, а також в охоронному відеоспостереженні. Людина щодня використовує CNN, якщо ввімкнено в налаштуваннях смартфона розблокування за допомогою розпізнавання особи.

Google використовує нейронні мережі в усіх своїх сервісах - від онлайн-пошуку і фільтрації пошти до підбору роликів на YouTube і перекладу текстів. LinkedIn використовує нейронні мережі спільно з лінійними текстовими класифікаторами для аналізу всіх записів, які розповсюджуються через свою платформу. Facebook застосовує DeepText, який переважно використовується для таргетування рекламних оголошень. Додатково Facebook використовує нейронні мережі для розпізнавання осіб і обробки зображень. Tesla використовує набір DNN у фірмовій системі автопілота. Вони розпізнають всі об'єкти навколо машини в реальному часі, класифікують їх і визначають характеристики. Дев'ята версія автопілота стала в чотири рази складніше восьмої за рахунок збільшення потоку даних. Одних камер стало вісім штук, причому їх роздільна здатність теж збільшилася [3]. Антивірусні розробники застосовують різні нейронні мережі для пошуку нових загроз в галузі інформаційної безпеки. Вони допомагають розпізнавати невідомі модифікації шкідливих файлів і нові види мережевих атак. Із презентацією системи ICSP Neural від Symantec можна ознайомитись за посиланням [4].

На практиці, досить часто виникає необхідність обробки зображень, тому розробка та використання програмного забезпечення на основі штучного інтелекту є актуальним та доцільним завданням, що потребує глибокого аналізу.

В дизайнерів із різних галузей виникає проблема, коли необхідне зображення із інтернету має низьку роздільну здатність і погану якість, а замовник продукту потребує саме цю картинку. Тому покращення відбувається за допомогою засобів ретушування, наприклад Photoshop, однак немає можливості редагування

або додавання пікселів. Та й не всі фахівці володіють відповідними навичками. В цьому випадку доцільно використовувати програмне забезпечення на основі штучного інтелекту.

Нейромережі вже давно і цілком успішно справляються із ретушуванням зображень. Також існує велика кількість програм алгоритмічного редагування. Для збільшення роздільної здатності фотографій є програма під назвою Gigapixel AI від компанії Topaz Labs. Вона не тільки збільшує роздільну здатність, а й реально покращує якість самого зображення і тому є необхідність вивчення даного продукту.

В основі алгоритмів програми лежить цілий комплекс нейромереж. Програма розбирає фото на пікселі і сортує їх за кольором, потім формується маска в потрібній роздільній здатності і далі система схематично додає пікселі потрібних кольорів на наявне зображення, таким чином буквально «домальовуючи» його до необхідного покращення [5].

Topaz Gigapixel AI буде корисна для дизайнерів у різних сферах, де використовують роботу із зображеннями. Наприклад, якщо:

- кадр зроблений на старий фотоапарат, телефон або фотоплівку;
- сталася невелика помилка з фокусом;
- фото було дуже урізано, через що постраждала деталізація;
- знімок потрібно роздрукувати великим форматом (більше 1м^2).

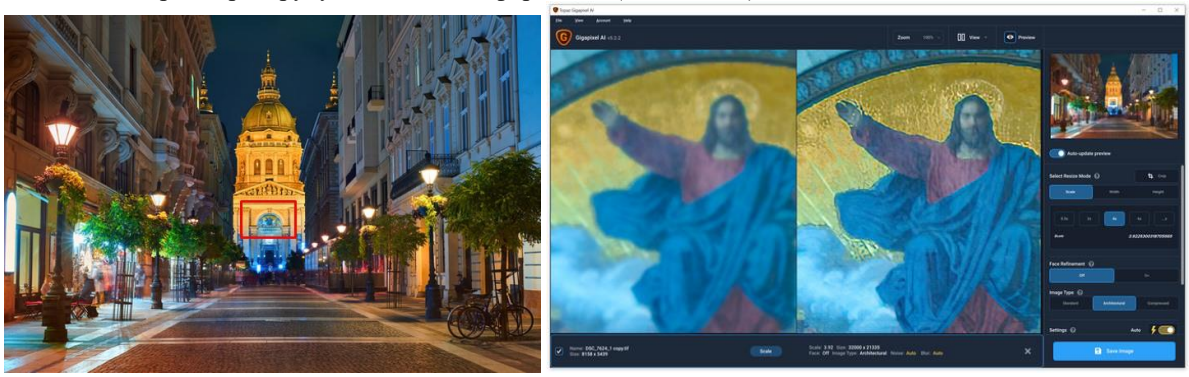

Рис. 2. Приклад обробки зображення за допомогою Topaz Gigapixel AI

Topaz Gigapixel AI розпізнає деталі і текстури, роблячи їх більш чіткими. Так знімок можна збільшити, не втративши в деталізації. Однак завантажувати відверто нерізкі, зіпсовані знімки сенсу немає - на них не вийде розпізнати достатню кількість деталей, результат буде посереднім.

Програма обробляє зображення та показує прев'ю. Можна вибрати кратність збільшення знімка. Краще ставити x2 або x4, цього достатньо, щоб виявити дрібні деталі і фактури на фото. Чим більше кратність, тим довше програма буде працювати. У Topaz Gigapixel AI дуже лаконічний інтерфейс, програма працює практично повністю автоматично.

Рекомендується проводити подальшу обробку зображень за допомогою інструментів Photoshop, щоб провести ресайз та зменшити розмір файлу зображення приблизно в 2-3 рази, не втративши в деталізації. Також цей інструмент дозволяє сховати дефекти роботи Topaz Gigapixel AI (вони іноді з'являються, якщо збільшувати фотографії в 4 або 6 разів).

На сьогоднішній день створено багато нових інструментів для редагування зображень. При ручній обробці в Photoshop на картинку витрачається багато часу. А це неприпустимо в сучасному світі: поки один дизайнер робить один кадр, конкуренти зроблять сто. Тому необхідно вивчати та розробляти програмне забезпечення на основі штучного інтелекту, що пришвидшує обробку зображень. Особливість програм на основі ШІ в тому, що не завжди контролюється результат від початку та до кінця, однак в 95% він виходить відмінним. А для решти 5% випадків важливо вміти працювати з класичними інструментами Adobe Photoshop. Така синергія класичних методів обробки та сучасних технологій дозволяють робити обробку не тільки швидко, але і якісно.

Нейронні мережі можуть допомогти компаніям розвиватись відразу по декількох напрямках: оптимізувати бізнес-логістику, покращити прогнозування та взаємодію з клієнтами. В кінцевому рахунку вони дають можливість підвищити конкурентоспроможність за рахунок більш глибокого аналізу всіх доступних даних. Очікування клієнтів постійно зростають, і вони вибирають сервіси на базі ШІ, оскільки вони швидше і повніше задовольняють їхні запити.

Список літератури

1. Artificial neural networks and applications: textbook. allowance / F. Gafarov, A. Galimyanov. - 2018. - 121 p.
2. Gonzalez R., Woods R. Digital image processing. - M.: Technosfera, 2002. - 1072 p.
3. https://youtu.be/7ztK5AhShqU
4. https://youtu.be/rung43Zsbpw
5. https://www.topazlabs.com/gigapixel-ai

УДК 004.9

І.О. Розломій, С.В. Науменко
sergey.naumenko94@gmail.com, inna-roz@ukr.net
Черкаський національний університет імені Богдана Хмельницького,м. Черкаси

ФРЕЙМВОРКИ ДЛЯ РОЗРОБКИ СЕРВЕРНОЇ ЧАСТИНИ WEB-ДОДАТКІВ

Фреймворк – це бібліотека мови програмування, що динамічно поповнюється, в якій зібрані її базові модулі. Фреймворки створюються задля спрощення процесів розробки додатків, сайтів, сервісів. Щоб не писати модуль у додатку з нуля, набагато простіше звернутися до готових шаблонів фреймворків, які формують робоче середовище розробника.

Архітектура багатьох фреймворків заснована на декомпозиції декількох окремих шарів (додатки, модулі) проєкту. Це означає, що можна розширювати функціональність програми, виходячи з потреб і використовувати змінену версію разом з кодом фреймворку або задіяти сторонні програми.

Така гнучкість є однією з ключових переваг використання фреймворків [1].

MVC (Model-View-Controller – «Модель-Представлення-Контролер») – концепція програмування, що розділяє класи на три групи, що широко застосовується і при використанні фреймворків. Структура концепції представлена рисунку 1.

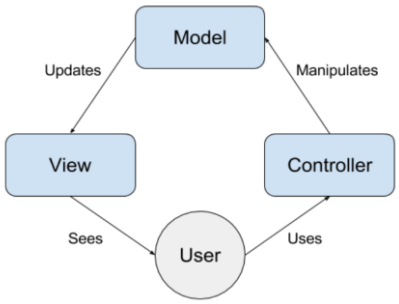

Рис. 1. Концепція MVC

Модель містить усі дані та рівні бізнес-логіки, її правила та функції. Подання відповідає за візуальне відображення даних, наприклад: діаграми, графіки тощо. Контролер легко трансформує дані для команд попередніх двох складових [1]. У фреймворків є дві основні функції: робота на серверній стороні (backend) та робота на клієнтській стороні (frontend).

Правила та архітектура серверних фреймворків не дозволяють створити веб-додаток з багатим інтерфейсом. Вони обмежені у своїй функціональності, проте все одно можна створювати прості сторінки та різні форми. Також вони можуть формувати вихідні дані та відповідати за безпеку у разі атак. Усе це безперечно може спростити процес розробки. Серверні фреймворки в основному відповідають за окремі, але критично важливі частини програми, без яких воно не зможе нормально працювати. Ось кілька самих популярних фреймворків та мови, з якими вони працюють: Django – Python, Zend – PHP.

Django – високорівневий фреймворк, який є не тільки швидким рішенням у веб-розробці, що включає все необхідне для якісного коду та прозорого написання, але також і відмінною платформою для роботи з клієнтурою того чи іншого бізнесу . Водночас він зручний для розробників.

У Django реалізований принцип DRY – Don't Repeat Yourself (не повторюйся). Тобто при використанні Django не потрібно кілька разів переписувати один і той же код. Фреймворк дозволяє створювати сайт з компонентів . Завдяки цьому скорочується час створення сайтів.

Фреймворк справляється з великим кількістю завдань та підвищеними навантаженнями. Його застосовують для створення:

- CRM-систем (Customer Relationship Management) відносинами з клієнтами);
- CMS-систем (Content Management System) – систем управління вмістом (контентом);
- комунікаційних платформ;
- сервісів бронювання номерів;
- платформ управління документообігом [4].

Також Django підходить для створення алгоритмічних генераторів, платформ для електронних розсилок, систем верифікації, систем фільтрації з динамічними правилами та складними параметрами, платформ для аналізу даних та складних обчислень, машинного навчання.

Реалізовано об'єктно-реляційне відображення (ObjectRelational Mapping (ORM), яке забезпечує взаємодія програми з базами даних (БД). ORM автоматично передає дані з БД, наприклад PostgreSQL або MySQL, в об'єкти, які використовуються в коді програми. включає механізми запобігання поширених атак на кшталт

SQL-ін'єкцій (XSS, CrossSite Scripting – «міжсайтовий скриптинг») та підробки міжсайтових запитів (CSRF або XSRF) [2].

Django REST Framework, який часто скорочують до DRF, є бібліотекою для побудови програмного інтерфейсу програми, API (Application Programming Interface). Він має модульну та настроювану архітектуру, яка добре працює для створення як простих , і складних API. Він поставляється з базовими класами для CRUD-операцій та вбудованою утилітою для тестування розроблюваного API [2].

Таблиця 1

Переваги та недоліки фреймворку Django

№ п/п	Переваги	Недоліки
1	Маса бібліотек, що дозволяє не писати базову функціональність, а тільки лише імпортувати необхідне.	Django не підтримує WebSockets, тому він погано підходить для роботи в реальному часу.
2	Докладна документація та доброзичливе спільнота, тому завжди набагато простіше знайти вже готові рішення або отримати допомогу від спільноти.	Готові бібліотеки часто знижують гнучкість , Django є великим та монолітним фреймворком, та його компоненти розгортаються спільно.
3	Django дозволяє почати з малого і масштабуватися в міру необхідності.	

Zend Framework – це об'єктно-орієнтована середовище веб-додатків з відкритим вихідним кодом, що представляє собою набір професійних пакетів на основі PHP, розроблена та підтримувана компанією Zend. Це набір класів яких не прив'язані один до одного, а значить, їх можна, можливо використовувати як по запчастинам в інших проектах, так і повністю. У цьому каркасі є все, що потрібно в сучасному webпроекті. Цей фреймворк, як правило, більше всього використовують при розробці великих комерційних проєктів. Такі компанії, як Google, Microsoft і StrikeIron, співпрацюють з Zend, надаючи інтерфейси для веб-сервісів та інших технологій, які вони хочуть зробити доступними розробникам Zend. Компанія бере участь у розробці ядра мови програмування PHP [3].

Фреймворк використовує різні пакети, використовуючи Composer як частина своїх менеджерів залежностей пакетів. Деякі з них: PHPUnit для тестування всіх пакетів, Travis CI для служб безперервної інтеграції. Zend Framework надає користувачам підтримку MVC у поєднанні з рішенням Front Controller. Zend Framework слідує стандартам PHP-FIG і включає реалізацію PSR-7 для інтерфейсів HTTP- повідомлень. Підтримка декількох систем баз даних та постачальників, включаючи MariaDB, MySQL, Oracle, IBM DB2, Microsoft SQL Server, PostgreSQL , SQLite та Informix Dynamic Server [3].

За допомогою служб віддаленого виклику процедур (Remote Procedure Call, RPC) та REST (Representational State Transfer – «передача стану уявлення») Zend Apigility допомагає розробникам створювати та документувати API. Zend Server забезпечує покращену продуктивність для PHP, і особливо програм Zend Framework, за рахунок прискорення коду операції та кількох можливостей кешування, а також включає кошти моніторингу та діагностики додатків. Zend Studio є інтегрованої середовищем розробки програм (Integrated Development Environment, IDE), яка включає функції, спеціально призначені для спрощення роботи з Zend Framework. Вона забезпечує MVC-представлення, генерацію коду MVC на основі ZendTool (компонент Zend Framework), надає кошти форматування, аналізу та виправлення коду, допомога за параметрами та багато інше [3].

Таблиця 2

Переваги та недоліки фреймворку Zend

№ п/п	Переваги	Недоліки
1	Чудово підходить для розробки комерційних веб- додатків.	Великоваговий та ресурсомісткий , оскільки надає розробнику масу можливостей та функцій.
2	Об'єктно-орієнтований підхід до розробки.	Не підходить для швидкої розробки проектів.
3	Непов'язані компоненти для повторного використання у проектах.	Недостатньо матеріалів для російськомовного сегмента розробників.

Список літератури

1. Фрэйн, Б. (2017) HTML5 и CSS3. Разработка сайтов для любых браузеров и устройств. Второе издание = Responsive Web Desigh with HTML5 and CSS3. Second Edition. Санкт-Петербург: Питер. 272 с.

2. Django: The Web framework for perfectionists with deadlines. – [Електронний ресурс] – Режим доступу до ресурсу: http://www.djangoproject.com (Дата звернення: 30.05.2022).

3. Васвани, В. (2012) Zend Framework. Разработка веб-приложений на PHP = Zend Framework: A Beginner's Guide. Санкт-Петербург: Питер. 432 с.

УДК 004.056.5:343.326 (045)

А.С. Батицька *студ. гр КБ-171*
Науковий керівник: Ю.М. Ткач, *к.т.н., професор*
Національний університет «Чернігівська Політехніка», м. Чернігів

COVID-19 ЯК ІМПУЛЬС ВПРОВАДЖЕННЯ ЦИФРОВИХ ТЕХНОЛОГІЙ В НАШЕ ЖИТТЯ

Під час пандемії COVID-19 злочинці активно використовують можливості, які їм надала криза, вміло адаптуючи свої методи. Загроза кіберзлочинності під час кризи носить динамічний характер і має потенціал для подальшого розростання. Дивлячись на небувалу кількість людей, які постійно знаходяться вдома і користуються онлайн-послугами, можливості злочинців значно зросли.

Ситуація ускладнюється тим, що багато питань взаємодії в цифровому середовищі ще не до кінця врегульовані в правовому полі. Це створює злочинцям простір для винаходу нових способів шахрайства.

Пандемія виявила вразливості в застарілих системах безпеки як в комп'ютерах громадського користування, так і в системах підприємств і установ і вимагала від урядів прийняття узгоджених заходів на державному, а також на міжнародному рівнях.

Пандемія коронавірусу дала потужний стимул масовому впровадженню цифрових технологій в повсякденне життя. Вже зараз очевидно, що зміни, які ця тенденція надасть в суспільно-економічний порядок, будуть носити безпрецедентний характер.

Тривалі в більшості країн світу заходи щодо соціальної ізоляції змусили перейти в онлайн істотну частину світової торгівлі товарами і послугами. Ймовірно, незабаром світ буде спостерігати подальше вибухове зростання капіталізації постачальників онлайн-послуг на тлі падіння позицій компаній сировинних галузей. Кардинально зміняться моделі споживання, помітна частка роботи і освіти також піде в дистанційний формат.

З одного боку, ці зміни зроблять життя людини ще більш зручною.

Широкі горизонти для розвитку людства відкриває можливість, не виходячи з дому, забезпечувати себе необхідними потребами, отримувати необхідну інформацію про основні соціально-економічні тенденції в форматі відкритих даних, лікувати захворювання і протидіяти їх поширенню за допомогою технологій дистанційного взаємодії, використання штучного інтелекту і аналізу великих даних.

Проте існує величезний набір ризиків і питань, на які поки немає однозначної відповіді.

- Як забезпечити приватність і захист персональних даних в умовах цифро-активізації життя?
- Як забезпечити права і свободу, підтримки?
- Як вирішити комплекс проблем, пов'язаних з кібербезпекою при тому, що дедалі більшої частини нашого повсякденного життя буде переведена в онлайн-режим?

Відповідь на ці питання дасть лише подальший розвиток подій. Але вже зараз є підстави вважати, що поточна криза в зв'язку з COVID-19 стане передвісником одного з найбільших переформатувань політичного і соціально-економічного укладу в сучасній історії. Провідну роль в ньому гратимуть саме цифрові технології, а в стороні від нього, ймовірно, не залишиться практично жодна держава світу.

Список літератури

1. COVID-19 та цифрова трансформація [Електронний ресурс] https://www.ua.undp.org/content/ukraine/uk/home/blog/2020/how-COVID-19-is-nudging-Ukraine-towards-digital-transformation.html.

2. Цифровий розвиток [Електронний ресурс] http://academy.gov.ua/infpol/pages/dop/2/files/225d8485-29b3-4826-9e3c-f7119b21dcd6.pdf

3. Інформаційні технології після COVID-19 [Електронний ресурс] https://www.dataart.com.ua/news/informafiini-tekhnologiji-pislya-covid-19.

УДК 004.9

К.О. Бойко, І.О. Розломій
boiko.kirill1119@vu.cdu.edu.ua, inna-roz@ukr.net
Черкаський національний університет імені Богдана Хмельницького, м. Черкаси

СУЧАСНІ КЛІЄНТСЬКІ ФРЕЙМВОРКИ МОВИ JAVA SCRIPT ДЛЯ РОЗРОБКИ WEB-ДОДАТКІВ

У сучасному світі складно уявити велику компанію та організацію без власного web-додатку або сайту. Разом з інтересом організацій зростає кількість бібліотек та фреймворків, спрямованих на полегшення розробки високоякісних веб-додатків. Фреймворки помітно змінили світ програмування і стали невід'ємною частиною веб-розробки, оскільки стандарти програм постійно модифікуються і, як наслідок, зростає складність технології. Використання готових інструментів та бібліотек, схвалених тисячами розробників по всьому світу, є розумним підходом для створення багатофункціональних та інтерактивних веб-додатків. Вибір фреймворку – завдання не найскладніше, але при виборі слід враховувати їх переваги та недоліки.

Існують клієнтські та серверні фреймворки. Клієнтська частина – це в основному HTML, CSS і JavaScript і розглядається, як спосіб подання інформації користувачам. Серверна частина, як правило, є найбільш складною та описує всю бізнес-логіку програми. Веб-розробку серверної частини можна виконати на багатьох мовах, таких як Java, Python, PHP.

На відміну від серверних, клієнтські фреймворки ніяк не пов'язані з логікою програми. Цей тип фреймворків працює у браузері. З їх допомогою можна покращити та впровадити нові користувацькі інтерфейси. Фронтенд-фреймворки дозволяють створювати різні анімації та односторінкові програми. Всі клієнтські фреймворки відрізняються за функціональністю та використанням. Розглянемо докладно деякі з них, які використовують мову JavaScript .

1. Angular – фреймворк з відкритим вихідним кодом, розроблений та підтримуваний Google. Інструмент дає все необхідне для створення та управління динамічними front-end сторінками для веб- програми.

У Angular є служба http, яка забезпечує взаємодію з віддаленими HTTP-серверами за допомогою XMLHttpRequest або JSONP [1]. Під час передачі об'єкта JavaScript на сервер він буде автоматично перетворений на рядок JSON. Після отримання відповіді служба також спробує перетворити отриманий рядок JSON в JavaScript. Використовуючи службу http можна створити власну службу з повним контролем над обробкою URL та даних.

Angular створений для спрощення складних процесів створення та управління JS-додатками. Бібліотека заснована на звичайному JS та HTML, тому Angular автоматично дбає про маніпуляції з DOM (Document Object Model – об'єктна модель документа) і AJAX-запити, які в іншому випадку розробникам довелося б писати самим. Інструмент надає модульні будівельні блоки коду JS, які можна поєднувати та тестувати. AngularJS можна швидко додати на будь-яку HTML-сторінку за допомогою простого тега. Завдяки підтримці TypeScript Angular є хорошим варіантом для розробки громіздких веб-програм [1].

У Angular застосовується двостороннє зв'язування: будь-які зміни у користувацькому інтерфейсі відразу ж позначаються на об'єктах додатки та навпаки. Фреймворк сам стежить за подіями браузера, змінами моделі та діями користувача на сторінці відразу оновлювати потрібні шаблони. При цьому код JavaScript не вимагає зберігати посилання на DOM-елементи та явно ними маніпулювати: просто описується необхідний результат у термінах стану моделі і немає потреби використовувати низькорівневі конструкції. Angular входить у пакет серверного програмного забезпечення (ПЗ) MEAN, який також включає MongoDB, Express.js та Node.js [2]. Тому він дозволяє керувати front-end та back-end частинами проєкту за допомогою JavaScript. Як альтернативи для back-end можна використовувати фреймворк Ruby on Rails (RoR), написаний на мові програмування Ruby.

Основні переваги фреймворка Angular, що підвищують швидкість і продуктивність – це синтаксис шаблону та інтерфейс командного рядка (Command Line Interface, CLI) для швидкого створення прототипів. Angular краще всього підходить для написання односторінкових додатків, але його також використовують і для розробки великих корпоративних веб-застосунків [2].

2. ReactJS – це бібліотека JavaScript, створена Facebook у 2013 році, вона чудово підходить для створення масштабних веб-додатків, де дані можуть змінюватися на регулярній основі.

React представив концепцію віртуального DOM, що представляє веб-сторінки в браузері. React володіє власни мвіртуальним DOM, який керує фактичним DOM браузера і, так як він набагато швидше, ніж DOM браузер, значно підвищує продуктивність. DOM React може створювати більше 200 000 вузлів за секунду, що перевищує середній показник вузлів для більшості сайтів.

Однією з особливостей React є введення JSX (розширення програмної мови JavaScript). Необхідно розуміти: JSX – це не HTML і не JavaScript. Перевага JSX полягає в тому, що він допомагає розробнику візуалізувати вміст сторінок, на ньому набагато простіше писати, ніж на традиційному JavaScript [3].

З погляду рівня складності, React є одним з найпростіших в освоєнні. Він ґрунтується на легких мовах програмування – достатньо згадати, як працює HTML. Не потрібно глибоко вивчати TypeScript, як у Angular, вистачає поверхневі знання.

React набув популярності завдяки архітектурі на основі компонентів, яку інші платформи почали використовувати набагато пізніше. Такий структурний підхід дозволяє порівняно швидко та просто створювати інтерфейс. Варто відзначити, що бібліотека спрямована не лише на створення користувацького інтерфейсу односторінкового веб-додатку (Single Page Application, SPA), але і на мобільну розробку.

Платформа React Native – це фреймворк призначений для розробки кросплатформних високоякісних програм для iOS та Android. Перевагами даного інструменту є різноманітність сумісних модулів (Browserify, RequireJS, ECMAScript та інші), встановлені компоненти, односпрямований потік коду, бібліотека Redux [3].

3. Vue – прогресивний JavaScript-фреймворк для розробки користувацького інтерфейсу. Відмінною рисою Vue від монструозних фреймворків є висока ступінь адаптивності. Вона полягає в орієнтованості на рівень уявлення (View) та простоті інтеграції інших бібліотек або в існуючі проєкти [4].

Vue дозволяє розробляти складні односторінкові програми (SPA) за рахунок розширення HTML-атрибутів про директивами. Існують як вбудовані директиви, що так і визначаються програмістом. Vue та розглянутий раніше React дуже схожі. Швидкість роботи обох фреймворків дуже висока, проте існують деякі нюанси функціонування Vue, на які стоїть звернути увагу.

На відміну від React, в якому зміна стану компонента веде до перемальовування всього піддерева цього компонента, Vue автоматично відстежує залежності компонентів. Такий підхід дозволяє системі точно знати, які компоненти необхідно перемальовувати.

Це усуває необхідність цілого класу оптимізацій. У React все побудовано на JavaScript, Vue ж охоплює класичні веб-технології і ґрунтуються на них. React відрізняється складністю вивчення, оскільки для застосування цього фреймворка необхідно володіти знаннями про JSX та ES2015+ та системи складання. Всі ці знання не є обов'язковими для початку розробки на Vue [4-5].

Порівняно з Angular, Vue має злегка велику продуктивність. Як уже було сказано раніше, для роботи з Angular потрібно знання TypeScript. Застосування TypeScript має свої переваги, такі як перевірка статичних типів це веде до збільшення накладних витрат. Vue надає офіційні декларації типів та офіційний декоратор для тих, хто хоче використовувати TypeScript разом з Vue.

Фреймворки для веб-розробки багато в чому схожі, навіть якщо реалізовані на різних мовами програмування. Проте кожен з перерахованих фреймворків індивідуальний. У них різні підходи , методи та поведінка у розробці. Не можна вибрати кращий фреймворк серед наявних, все залежить від того, з чим планується працювати. Фреймворки мають масу різних переваг та недоліків. Можна порівняти всі функції та технології, перелічені в цій статті, щоб було легше зрозуміти, який з фреймворків краще підходить для бізнесу або потреб конкретного проєкту, та зробити правильний вибір, спостерігаючи в результаті, як вибрані інструменти та бібліотеки здійснюють величезну допомога у роботі.

Список літератури

1. Рейсиг, Д. JavaScript. (2008) Профессиональные приемы программирования. Pro JavaScrip Techniques. Санкт-Петербург: Питер. 352 с.

8. Angular. JavaScript-фреймворк Habr. – [Електронний ресурс] – Режим доступу до ресурсу: http://habr.com/ru/hub/angular (Дата звернення: 05.05.2022).

2. React. JavaScript-библиотека для создания пользовательских интерфейсов. – [Електронний ресурс] – Режим доступу до ресурсу: http://ru.reactjs.org (Дата звернення: 05.05.2022).

3. Стефанов, С. React.js. Быстрый старт React: Up & Running. СанктПетербург: Питер, 2017. 304 с.

4. Vue.js – Introduction Vue.js. – [Електронний ресурс] – Режим доступу до ресурсу: http://vuejs.org/v2/guide/index.html (Дата звернення: 05.05.2022).

5. Vue.js –Comparison with Other Frameworks Vue.js. – [Електронний ресурс] – Режим доступу до ресурсу: http://vuejs.org/v2/guide/comparison.html (Дата звернення: 05.05.2022).

УДК 004.453, 004.33

В.С. Катаєв, І.С. Каплун, І.О. Бондаренко
kataev@vntu.net, kaplun.irka@gmail.com, fm.ub15b.bondarenko@gmail.com
Вінницький національний технічний університет, м. Вінниця

ПІДВИЩЕННЯ ЗАХИЩЕНОСТІ ПРОГРАМ ВІД НЕСАНКЦІОНОВАНОГО ДОСТУПУ З ВИКОРИСТАННЯМ АПАРАТНОГО ЗАСОБУ

Вступ. Будь – який продукт, котрий використовує кожна людина, є товаром, який має автора, ціну, гарантії якості, а отже може бути об'єктом крадіжки та незаконного використання. Необхідність внесення до програмного забезпечення захисних функцій протягом його життєвого циклу від етапу з'ясування задуму на створення програм і їх розробки до етапів випробувань, експлуатації, модернізації і супроводу програм є досить поширеною [1].На сьогодні актуальним завданням є підвищення захищеності програм від НСД, зокрема шляхом використання протоколу активного захисту.

Розроблення алгоритмів роботи. Пропонується застосовувати для електронного ключа, що дає доступ до програмного забезпечення, покращений протокол активного захисту (при кожному запуску програми перевіряється наявність ключа у портах ПК). Специфіка апаратної розробки полягає у самознищенні ключа у разі його відсутності, або невірному використанні. Така функція є новою серед його аналогів.

Переваги запропонованої модифікації електронного ключа: наявність алгоритму активного захисту (якщо ключ отримує невірний запит, відбувається очищення пам'яті мікроконтролера)

Це може бути реалізовано з допомогою власного бутлоадера і зроблено для унеможливлення використання ключа після спроби використання його на іншому комп'ютері, або спроби його вивчення методом чорної скриньки (коли зловмисники вивчають реакцію приладу на різні запити).

Для перевірки валідності запиту, ключ перевіряє:

- базову швидкість порту, з яким він працює;
- коректність протоколу роботи з комп'ютером та коректність запиту, що надходить до пристрою.

Також система захищена від підключення ключа в інші ПК, відправки йому запитів і дослідження, що ключ відповість на той чи інший запит. У разі некоректного запиту, flash - пам'ять очищується.

Принцип роботи розробки полягає у тому, що після первинної активації (яка проводиться довіреною особою), ключ виявляється в ПК і програма його розпізнає, відбувається прив'язка. Програма при запуску перевіряє наявність ключа в портах. Ключ, в свою чергу, під час надходження до нього запиту, перевіряє: швидкість передачі даних, на якій йому прийшло повідомлення; коректність протоколу (реалізовано т.зв. "рукостискання", воно ж handshake); коректність запиту, який він отримав від комп'ютера. Запит є динамічним, тобто кожен раз відрізняється.

Структура запиту може бути описана наступним чином:

- комп'ютер послідовно відкриває всі доступні послідовні порти;
- при відкритті порту, він перевіряє, чи приходять в порт певні дані (ключ при цьому в свій порт відправляє хеш - код свого серійного номера, щоб його могли впізнати);
- якщо виявлено певний (не обов'язково правильний) ключ, відбувається його верифікація;
- ПК здійснює відправку ключеві дати активації в хешованому вигляді;
- ключ здійснює хешування своєї дати активації та зіставляє хеш-коди;
- у випадку, якщо хеш-коди збігаються, то відбувається робота з програмою далі;
- у випадку, якщо хеш – коди не збігаються, відбувається очищення flash - пам'яті.

Після запуску програми, активації, перевірки ключа, і запуску захищуваного програмного забезпечення, відбувається періодична перевірка присутності ключа.

Для виконання вказаної функції відбуваються такі дії:

- послідовний порт залишається відкритим при активації;
- у порт, в якому вставлений ключ, відправляється хешована кількість секунд з моменту активації;
- ключ здійснює перевірку правильності отриманої інформації;
- у випадку, якщо фіксуються некоректні дії – відбувається очищення флеш-пам'яті ключа;
- у випадку, якщо вся робота коректна, то відбувається трансформація рядка, отриманого від ПК та його відправка (таким чином, якщо витягти ключ, захищуваний модуль перестане працювати).

Отже, запропонована модифікація є новою серед аналогів електронних ключів для захисту програмного забезпечення, проте може бути достатньо дієвою, що є підставою для проведення подальшої розробки. Алгоритм роботи системи контролю доступу до програмного забезпечення базується на двоетапній авторизації будь – якого користувача, що дає можливість забезпечити її коректну роботу та надійний захист.

Засоби для реалізації розробки. Система контролю доступу до програмного забезпечення складається з програмної частини (що знаходиться на ПК разом з захищуваним ПЗ та прив'язується до апаратної частини ПК) та електронного ключа, виконаного на базі мікроконтролера Atmega 328.

При розробці апаратної частини пристрою, увага приділяється практичній реалізації електронного ключа, що в результаті надає користувачу можливість отримати доступ до захищуваного програмного

забезпечення. Апаратна реалізація ключа виконана на мікроконтролері AtMega328 сімейства AVR виробництва компанії Atmel. Плата розробки, на базі якого зроблений ключ – Arduino Nano [2].

Користувацька частина системи контролю доступу до програмного забезпечення з використанням USB-ключів розроблена в середовищі програмування Visual Studio на мові об'єктно - орієнтованого програмування C# [3].

Для виконання поставленої задачі, розроблені модулі додатку: модуль користувацької частини; модуль апаратної частини реалізації. Для програмної реалізації модуля користувацької частини проекту, слід віднести такі блоки: блок 1 - робота з послідовними портами, блок 2 - індикація даних, блок 3 - перевірка повідомлень, блок 4 - авторизація, блок 5 - контроль портів, блок 6 - розробка захищуваного додатку та його інтерфейсу. Модульне проектування апаратної частини практичної реалізації містить такі блоки: блок 1 - передача даних, блок 2 – завантажувач, блок 3 - процес здійснення перевірки коректності здійснюваних запитів.

Тестування розробки. Наступним етапом роботи є практична перевірка надійності електронного ключа для захисту програмного забезпечення.

Перша перевірка буде здійснюватися за допомогою сніфера. Отже, було встановлено команду, якою обмінюються ключ і комп'ютер при спільний автентифікації, дана команда записується. Злам ключа реалізовано за допомогою термінальної програми, адже розроблюваний ключ працює з перетворювачем usb – com. Якщо електронний ключ від'єднати від ПК, він відповідно зникне з вікна диспетчера пристроїв. Такі дані свідчать про те, що ключ «спілкується» з ПК за допомогою USB – UART перетворювача.

Наступний метод – спробувати проаналізувати ключ та дізнатись, що він складається з Atmega328, який являється мікроконтролером сімейства AVR, а отже в більшості випадків прошивається через ISP. Виходи SCK, MISO, MOSI, Reset використовуються для підключення програматора. Для того, щоб здійснити подальше підключення необхідно спеціальне програмне забезпечення для роботи з програматором. Спробуємо зчитати прошивку ключа з флеш – пам'яті мікроконтролера. Для цього скористаємось додатком extreme burner. Спробуємо здійснити підключення. З отриманих даних можна побачити, що підключення не відбулось. Такий результат пояснюється тим, що підключення не відбулось з тієї причини, що в мікроконтролері відключений ISP для неможливості виведення, а також перезавантаження як логічного виходу. Такі налаштування суттєво ускладнюють процес зчитування програматора для зловмисника.

Отже, можна припустити, що теоретично ключ може піддатись зламу у випадку майже неможливої випадковості, методом підбору і т.д. Проте слід зауважити, що часу на здійснення спроб для несанкціонованого доступу до ключа, потрібно чи мало, проте, ключ – лише один. І запрограмований він таким чином, що в жодному разі не допускає невірної активації чи найменшої спроби зламу. Такі ключі орієнтовно можна використовувати для особливо важливого програмного забезпечення, використання якого несанкціонованими користувачами взагалі не допускається.

Висновки. В даній роботі було описано основні етапи розробки системи контролю доступу з використанням форми авторизації та електронного USB – ключа. В ході розробки, головним структурним елементом підсистеми було обрано пристрій Arduino Nano на основі мікроконтролера ATmega328, оскільки в ньому міститься усе необхідне для зручної роботи з контролером.

Тестування програми проводилось на ПК з операційною системою Windows XP, 7, 8, 8.1 та 10. На вказаних вище операційних системах розроблювальна система захисту ПЗ працює коректно за умови встановленого драйверу для мікросхеми CH-340g (USB – UART конвертер).

На практиці протестовано результативність та стійкість розроблюваного додатку. Тестування показало, що розроблювальна система захисту ПЗ працює коректно, а дана розробка потребує незначну кількість ресурсів для своєї коректної роботи.

Список літератури

1. Огляд сучасних методів захисту програмного забезпечення. StudFiles. URL: https://studfile.net/preview/3905114/ (дата звернення: 01.05.2022).

2. Arduino Nano: все, що потрібно знати про плату розробки. Hardware libre. URL: https://www.hwlibre.com/uk/arduino-nano/ (дата звернення: 01.05.2022).

3. Вступ в C# URL: https://programm.top/uk/c-sharp/tutorial/introduction/ (дата звернення:

УДК 004.056

Ю.Є. Яремчук, А.В. Грицак
yurevyar@vntu.edu.ua, grytsak.a.v@gmail.com
Вінницький національний технічний університет,м. Вінниця

УДОСКОНАЛЕННЯ МЕТОДУ ПОБУДОВИ КРИПТОСТІЙКИХ ФУНКЦІЙ ГЕШУВАННЯ

Веб-браузери постійно розширюють свої функціональні можливості та надають користувачам можливість збереження своїх конфіденційних даних, документів, пошти та ін. У зв'язку з цим, забезпечення захищеного доступу до Веб-ресурсів та обмін інформацією між ними займає одне з пріоритетних напрямів в процесі забезпечення захисту інформації та потребує постійного вдосконалення. Одним з найпоширеніших методів захисту є використання криптографічних сертифікатів – цифрових сертифікатів, які забезпечують конфіденційний обмін даними між клієнтом та сервером шляхом шифрування та аутентифікації цифрового сертифікату. Цифровий сертифікат являє собою відкритий ключ користувача, завірений ЕЦП сертифікаційного центру. Однак цифровий сертифікат це не лише відкритий ключ з інформацією, а так званий підпис сервера чи веб-ресурсу, який реалізується використовуючи геш-функції. За останні роки тенденція зростання кількості кібератак збільшується в геометричній прогресії. Так, при збільшення кількості атак, а отже і виявленні нових уразливостей, спостерігається ряд проблем з реалізацією і застосуванням цифрових сертифікатів. Тому підвищення надійності цифрових сертифікатів, як найпоширеніших методів захисту обміну даними через канали зв'язку, є актуальним та потребує вдосконалення.

Сьогодні є багато систем, для яких критичним є параметр стійкості – це цифрові сертифікати, захищені протоколи і системи, що містять інформацію з обмеженим доступом тощо.

Прототипом для методу побудови функцій гешування (стійкого) було обрано функцію гешування SHA-2 [1]. Дана функція гешування базується на структурі Меркла-Демґарда. Порівняно з прототипом було змінено:

1. Початкове повідомлення доповнюється розміром цього повідомлення та псевдовипадковою послідовністю salt (розраховується на основі самого повідомлення за допомогою функції F_{Gen}). Слід зауважити, що для кожної нової функції гешування розробленої за допомогою даного методу можна задавати свою унікальну функцію F_{Gen} .

2. Введено параметри l, L, при фіксації яких формується нова структура нової функції гешування (змінюється розрядність операцій).

3. Кількість раундів у функції стиснення F_g визначається за допомогою фіксації параметра R.

4. У функцію стиснення F_g для підвищення не лінійності введено операцію підстановки $S(x)$. Слід зауважити, що для кожної нової функції гешування розробленої за допомогою даного методу можна задати свою унікальну операцію підстановки $S(x)$.

5. У функції стиснення F_g запропоновано свої етапи розбиття блоків на слова та ініціалізації змінних (на основі використання операцій підстановок $S(x)$).

6. У функції стиснення F_g в етапі безпосереднього стиснення запропоновано свій порядок операцій (на основі 6-ти не лінійних функцій, операцій підстановки, додавання за модулем 2 і 2^n, циклічних і лінійних зсуві), введено дві нових нелінійні функції $JQ(x,y)$, $SH(x,y)$, введено використання операцій підстановок $S(x)$, змінено дві нелінійні функції $Sigma_0(x,y)$, $Sigma_1(x)$.

При зміні/фіксації параметрів l, L, R визначенні операцій F_{Gen} та $S(x)$ можна буде будувати різноманітні функції гешування.

Отже, запропоновано новий метод побудови функцій гешування, який базується на структурі Меркла-Демґарда та за рахунок доповнення вхідного повідомлення розміром цього повідомлення та псевдовипадковою послідовністю salt, використання у функції стиснення нової послідовності операцій (на основі 6-ти не лінійних функцій, операцій підстановки, додавання за модулем 2 і 2^n, циклічних і лінійних зсувів), дозволить будувати криптостійкі функції гешування.

Список літератури

1. Бабенко Л.К. Современные алгоритмы блочного шифрования и методы их анализа / Бабенко Л.К., Ищукова Е.А. – М.: Гелиос АРВ. – 2006. – 376 с.

ЗМІСТ

СЕКЦІЯ 1. ІНФОРМАЦІЙНА БЕЗПЕКА ДЕРЖАВИ, СУСПІЛЬСТВА ТА ОСОБИСТОСТІ

СЕКЦІЯ 2. ПРОГРАМУВАННЯ ТА ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНІ ТЕХНОЛОГІЇ

СЕКЦІЯ 3. ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ В ЕКОНОМІЦІ, МЕДИЦИНІ ТА ОСВІТІ

НАУКОВЕ ВИДАННЯ

ТЕЗИ ДОПОВІДЕЙ

V Міжнародної науково-практичної конференції "Інформаційна безпека та комп'ютерні технології"

19–20 травня 2022 р.

Матеріали публікуються в авторській редакції.
За достовірність викладених фактів, цитат та інших відомостей
відповідальність несуть автори.

Відповідальний за випуск: *О.А. Смірнов*

Комп`ютерна верстка: *Р.М. Минайленко*

Електронне видання

Центральноукраїнський національний технічний університет
пр-кт Університетський, 8, м. Кропивницький, 25006.
тел. (0522) 559-245, www.kntu.kr.ua