ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

DOI: https://doi.org/10.15407/elmodel.48.04.017
УДК 004.056

І.В. Мартинюк [2,3], аспірант,
e-mail: imartyniukiv@gmail.com; orcid.org/0009-0003-5565-0828
Н.В. Заїка [1,2], аспірант,
e-mail: nazar2611@gmail.com; orcid.org/0000-0002-5791-8926
[1] Інститут проблем моделювання в енергетиці ім. Г.Є. Пухова НАН України
Україна, 03164, Київ, вул. Олега Мудрака, 15
[2] Державний науково-дослідний інститут технологій кібербезпеки
Україна, 03142, Київ, вул. М. Залізняка, 3, корпус 6
[3] Національний університет «Київський авіаційний інститут»
Україна, 03058, Київ, пр-т Гузара Любомира 1

Інтелектуальний моніторинг та криптографічний захист машинного коду в системах критичної інфраструктури

Розглянуто інтелектуальний моніторинг кіберфізичних систем, що базується на поєднанні безперервного спостереження та моделей машинного навчання для виявлення аномалій і підвищення проактивності кіберзахисту. Ефективність таких підходів визначено якістю даних і коректністю моделей, які побудовані на системах з моніторингу, де криптографічні механізми інтегруються безпосередньо у процес обміну даними між агентами та сервером. Криптографічний захист розробляється для забезпечення цілісності системи безпеки на рівні машинного коду. Верифікація алгоритмів здійснюється через тестові вектори відповідно до практик National Institute of Standards and Technology (NIST). Сукупно це формує комплексний підхід до кіберстійкості ОКІ.

К л ю ч о в і с л о в а: інтелектуальний моніторинг, кіберфізичні системи, критична інфраструктура, машинне навчання, виявлення аномалій, криптографічний захист, кіберрезильєнтність.

Постановка проблеми. Забезпечення безпеки кіберфізичних систем і об’єктів критичної інфраструктури ускладнюється зростанням кількості кіберзагроз і складністю їх своєчасного виявлення. Традиційні підходи, засновані на сигнатурному аналізі, не забезпечують ефективного виявлен-

І.В. Мартинюк, Н.В. Заїка, 2026
Стаття поширюється на умовах ліцензії відкритого доступу CC BY 4.0

ня нових або модифікованих атак, що обумовлює необхідність переходу до методів інтелектуального моніторингу та поведінкового аналізу. Водночас ефективність таких підходів суттєво залежить від якості даних і коректності моделей, а також від їх інтеграції з існуючими системами моніторингу. У той же час зростання частки зашифрованого мережевого трафіку ускладнює застосування традиційних методів аналізу, що базуються на доступі до вмісту пакетів, і потребує переходу до нових методів моніторингу стосовно аналізу метаданих або контролю на кінцевих вузлах.

Окрему проблему становить забезпечення довіри до програмного середовища на базовому рівні, у процесах завантаження системи. Це визначає необхідність застосування криптографічних механізмів контролю цілісності на рівні уніфікованого розширюваного інтерфейсу прошивки Basic Input/Output System (BIOS) / Unified Extensible Firmware Interface (UEFI), а також використання перевірки цифрових підписів виконуваного коду. Окрім цього, актуалізується проблема захисту ланцюга постачання програмного забезпечення, де атаки можуть реалізовуватися ще на етапі оновлення або виробництва прошивки обладнання на об'єктах критичної інфраструктури, забезпечення цілісності, достовірності та захищеності даних є ключовою передумовою їх надійного функціонування. Отже, задача полягає у систематизації підходів до застосування криптографічних механізмів у системах моніторингу та визначенні їх ролі у забезпеченні достовірності, цілісності та безпеки даних в умовах сучасних кіберзагроз.

Метою дослідження є узагальнення підходів до криптографічного захисту машинного коду, управління ключами та верифікації алгоритмів хешування, протоколів захищеного обміну даними та механізмів забезпечення цілісності інформації у системах моніторингу, а також визначення їх впливу на підвищення надійності та забезпеченні кіберстійкості кіберфізичних систем і об'єктів критичної інфраструктури.

Основні завдання дослідження:

• проаналізувати загрози, пов'язані з компрометацією прошивки та механізмів завантаження, а також їх вплив на безпеку систем;

• дослідити роль безперервного моніторингу та поведінкового аналізу у виявленні кіберінцидентів;

• дослідити застосування криптографії для забезпечення цілісності та автентичності цифрових доказів у процесах реагування на інциденти;

• проаналізувати рекомендації щодо впровадження криптографічних механізмів, включаючи використання сучасних протоколів та управління ключами;

• узагальнити взаємозв'язок між інтелектуальним моніторингом і криптографічним захистом як складовими комплексної системи кібербезпеки.

Інтелектуальний моніторинг та виявлення аномалій. *Огляд підходів і джерел.* Сучасні підходи до інтелектуального моніторингу кіберфізичних систем базуються на поєднанні класичних механізмів спостереження та методів машинного навчання, наприклад, для задач виявлення аномалій. У керівництві National Institute of Standards and Technology (NIST) підкреслюється, що безперервний моніторинг (continuous monitoring) є необхідною умовою для підтримання актуального рівня безпеки системи, а аналіз поведінкових відхилень виступає одним із ключових інструментів виявлення інцидентів [1].

Застосування алгоритмів виявлення аномалій розглядається як ефективний механізм для ідентифікації невідомих або нових типів атак, які не можуть бути виявлені сигнатурними методами. У звітах European Union Agency for Cybersecurity (ENISA) зазначається, що поведінковий аналіз і моделі машинного навчання дозволяють переходити від реактивних до проактивних підходів у кіберзахисті середовищ критичної інфраструктури [2].

Практичне впровадження таких моделей передбачає інтеграцію з системами моніторингу (наприклад, Zabbix), а також із системами управління інформацією та подіями безпеки, Security Information and Event Management (SIEM-платформи), де джерелом даних виступають журнали подій, мережеві метрики та системна телеметрія. У рекомендаціях NIST SP 800-137 підкреслюється, що ефективність моніторингу визначається не лише збором даних, а й здатністю їх інтерпретувати у контексті нормальної поведінки системи [1].

У галузевих звітах VerizonData Breach Investigations Report акцентується роль людського фактора та помилок конфігурації як ключових причин інцидентів, що підтверджує необхідність автоматизованого аналізу поведінкових відхилень [3]. Це підсилює значення моделей машинного навчання, здатних працювати з великими обсягами даних і виявляти нетипові патерни, недоступні для ручного аналізу. Наукові публікації у сфері anomaly detection та оглядові роботи в Institute of Electrical and Electronics Engineers (IEEE) / Association for Computing Machinery (ACM) наголошують на критичній залежності якості моделей від навчальних даних. Використання неперевірених або відкритих джерел без належної валідації може призвести до викривлення результатів (data bias, data poisoning), що особливо небезпечно в системах моніторингу критичної інфраструктури [4].

Криптографічний захист машинного коду. Криптографічний захист машинного коду на рівні базових систем ініціалізації (BIOS/UEFI) та керування ключами розглядається як один із фундаментальних елементів довіри до обчислювального середовища, особливо в контексті об’єктів критичної інфраструктури. У сучасних архітектурах цей рівень безпеки

формує так званий root of trust — початкову точку, з якої починається перевірка цілісності всієї програмної системи [5].

Одним із ключових механізмів є використання процесу Secure Boot, стандартизованого в специфікаціях UEFI, сутність якого полягає у криптографічній перевірці цифрових підписів виконуваного коду на етапі завантаження системи. Кожен компонент — від прошивки до завантажувача операційної системи — перевіряється на відповідність довіреним сертифікатам, що зберігаються у захищеній пам'яті платформи. Такий підхід дозволяє запобігти виконанню модифікованого або шкідливого коду ще до запуску операційної системи [6].

У рекомендаціях NIST підкреслюється, що безпечне завантаження є критично важливим для систем, які виконують функції керування технологічними процесами, оскільки компрометація на цьому рівні робить неефективними всі подальші механізми захисту [7]. УIndustrial Control Systems (ICS) / Supervisory Control and Data Acquisition (SCADA) контроль цілісності прошивки розглядається як необхідна умова забезпечення кіберстійкості.

Криптографічна модель Secure Boot базується на ієрархії ключів. Серед яких основними елементами є:

• ***Platform Key (PK)*** — визначає власника платформи та дозволяє керувати політикою безпеки;

• ***Key Exchange Keys (KEK)*** — використовуються для оновлення списків довірених ключів;

• ***db (database)*** — містить дозволені сертифікати та хеші;

• ***dbx (revocation list)*** — список відкликаних або заборонених ключів.

Ця модель забезпечує контроль довіри на кожному етапі завантаження системи та дозволяє централізовано керувати політикою безпеки [6].

Додатковим рівнем захисту є використання апаратних модулів довіри Trusted Platform Module (TPM). TPM забезпечує захищене зберігання криптографічних ключів і підтримує механізми вимірюваного завантаження (measured boot), у межах якого обчислюються хеші компонентів системи та зберігаються у спеціальних регістрах платформи Polymerase Chain Reaction (PCR). Це дозволяє виявляти несанкціоновані зміни навіть у разі компрометації програмного забезпечення [8].

Реалізація криптографічних алгоритмів у програмному та апаратному середовищі об'єктів критичної інфраструктури (ОКІ) розглядається як окремий рівень забезпечення безпеки, що суттєво відрізняється від теоретичної коректності самих алгоритмів. Наявні дослідження демонструють, що навіть за умови використання стійких криптографічних примітивів помилки реалізації можуть призводити до компрометації системи. Значна частка вразливостей у криптографічному програмному забезпеченні пов'я-

зана з помилками програмування, а не з недоліками алгоритмів [2, 3]. Це узгоджується з рекомендаціями NIST, де підкреслюється, що вразливості у промислових системах часто виникають на рівні реалізації та конфігурації, а не лише через використання застарілих механізмів захисту [7].

Одним із найбільш характерних класів помилок є порушення безпечної роботи з пам'яттю, що може призводити до витоку конфіденційних даних. Відомим прикладом є вразливість Heartbleed у бібліотеці OpenSSL, яка дозволяла отримувати дані з оперативної пам'яті сервера внаслідок некоректної перевірки меж буфера [9]. Подібні помилки не пов'язані з криптографічною схемою Transport Layer Security (TLS) як такою, але впливають на її безпечність у практичному застосуванні. Для систем ОКІ це має критичне значення, оскільки компрометація ключового матеріалу або службових даних може впливати на керування технологічними процесами. Практика показує, що компрометація прошивки або механізмів завантаження може мати довготривалі наслідки. У дослідженнях безпеки прошивок підкреслюється, що такі атаки є складними для виявлення та можуть зберігатися навіть після перевстановлення операційної системи [10]. Це пояснює необхідність застосування криптографічних механізмів саме на рівні BIOS/UEFI.

У контексті машинного коду важливим є також захист виконуваних файлів через цифрові підписи та перевірку їх цілісності. Використання надійних криптографічних підписів дозволяє гарантувати автентичність програмного забезпечення та запобігти підміну виконуваних компонентів. Водночас ефективність цього підходу залежить від безпечного управління ключами, що включає їх генерацію, зберігання, ротацію та відкликання. Згідно з рекомендаціями NIST щодо управління ключами, компрометація ключового матеріалу є однією з найбільш критичних подій у криптографічній системі, оскільки вона зводить нанівець усі механізми системи захисту [5]. Тому особлива увага приділяється використанню апаратних засобів захисту ключів, ізоляції середовищ виконання та контролю доступу до ключової інформації.

У сучасних умовах додатковим викликом є необхідність захисту ланцюга постачання (supply chain). Дослідження показують, що компрометація може відбуватися ще на етапі виробництва або оновлення прошивки, що потребує використання механізмів перевірки підписів і довіри до джерела оновлень [10].

Методика перевірки реалізації алгоритму гешування за ДСТУ 7564:2014 на основі тестових прикладів. Реалізація криптографічних алгоритмів хешування потребує обов'язкової перевірки на відповідність стандарту, оскільки навіть незначні відхилення в реалізації можуть призвести до втрати криптографічної стійкості або некоректної обробки даних.

Для алгоритму, визначеного у ДСТУ 7564:2014 (алгоритм «Купина»), така перевірка здійснюється шляхом порівняння результатів обчислення геш-значень із еталонними тестовими векторами, наведеними у стандарті. ДСТУ 7564:2014 встановлює специфікацію криптографічної хеш-функції, яка підтримує різні довжини вихідного значення (256, 384 та 512 біт) і побудована на основі перетворень, подібних до блочних шифрів, із використанням підстановок та лінійних операцій [11]. У стандарті наведено тестові приклади (test vectors), що включають фіксовані вхідні повідомлення та відповідні їм геш-значення. Саме ці приклади використовуються як базовий інструмент валідації реалізації.

Методика перевірки передбачає кілька послідовних етапів. На першому етапі формується набір тестових даних, що включає повідомлення різної довжини, у тому числі порожній рядок, короткі повідомлення та довгі послідовності байтів. Такі підходи відповідають загальній практиці тестування криптографічних алгоритмів, у рекомендаціях NIST щодо перевірки реалізацій криптографічних функцій [12].

Другий етап полягає у виконанні обчислення геш-значення відповідно до реалізованого алгоритму. При цьому критично важливим є дотримання всіх етапів обробки даних, визначених стандартом: попереднє доповнення повідомлення (padding), розбиття на блоки, ітераційні перетворення та формування фінального геш-значення. Помилки на будь-якому з цих етапів призводять до розбіжності з еталонними значеннями.

Третій етап — порівняння отриманих результатів із тестовими векторами. Якщо хоча б один результат не збігається з еталонним значенням, реалізація вважається некоректною. Такий підхід є стандартним для криптографічної валідації та використовується, у програмі Cryptographic Algorithm Validation Program (CAVP), що застосовується NIST [12].

Важливим аспектом є тестування граничних випадків: обробка повідомлень нульової довжини; обробка повідомлень, довжина яких кратна або некратна розміру блоку; перевірка коректності роботи при великих обсягах даних.

Згідно з дослідженнями у сфері криптографічної інженерії, використання тестових векторів є необхідною, але недостатньою умовою повної верифікації алгоритму. Додатково рекомендується застосування методів диференційного тестування та аналізу стійкості до помилок реалізації [13]. Однак у відкритих джерелах відсутні підтверджені методики, що прямо стандартизують такі підходи саме для ДСТУ 7564:2014, тому їх застосування слід розглядати як загальну практику, а не вимогу стандарту.

Слід зазначити, що у відкритому доступі відсутній повний опис документа **ЄАБА.460001.028,** тому неможливо підтвердити його зміст або методичні положення на основі офіційних джерел. Відповідно, у даному ма-

теріалі не використовуються твердження, що прямо посилаються на цей документ.

Узагальнюючи, перевірка реалізації алгоритму гешування за ДСТУ 7564:2014 базується на порівнянні результатів із еталонними тестовими прикладами, що визначені стандартом, і відповідає загальноприйнятим міжнародним практикам валідації криптографічних алгоритмів. Водночас подальші дослідження мають бути спрямовані на розробку формалізованих методик тестування, що враховують специфіку національних стандартів і сучасні вимоги до безпеки.

Сучасні підходи до захисту даних у системах моніторингу базуються на використанні криптографічних протоколів, які забезпечують конфіденційність, цілісність та автентичність інформації під час передавання та обробки. Одним із базових стандартів є протокол Transport Layer Security (TLS), що застосовується для захисту мережевих з'єднань і реалізує механізми узгодження ключів, симетричного шифрування та перевірки цілісності повідомлень. Важливою властивістю сучасних реалізацій TLS є підтримка режимів із прямою секретністю (forward secrecy), що унеможливлює дешифрування раніше перехопленого трафіку навіть у разі компрометації ключів у майбутньому [9].

Криптографічні хеш-функції з алгоритмами родини SHA-2 відіграють ключову роль у забезпеченні цілісності даних. Вони формують фіксований за довжиною хеш-образ, що дозволяє виявляти будь-які зміни у вхідних даних. Властивості стійкості до колізій та до знаходження прообразу визначають їх придатність для застосування у цифрових підписах, протоколах автентифікації та системах контролю цілісності [14]. У практичних системах моніторингу ці механізми застосовуються для перевірки журналів подій і телеметрії перед їх подальшим аналізом.

У системах моніторингу інфраструктури, які побудовані на базі Zabbix, криптографічні механізми інтегруються безпосередньо у процес обміну даними між агентами та сервером. Документація платформи передбачає використання TLS для захисту каналів зв'язку, а також підтримку сертифікатної та ключової автентифікації, що дозволяє забезпечити контроль джерел даних та їхню достовірність [15]. Це має принципове значення для подальшого використання даних у системах аналітики та машинного навчання, де навіть незначні спотворення можуть призвести до некоректних результатів.

З практичної точки зору, рекомендації щодо впровадження криптографічних механізмів у системах критичної інфраструктури передбачають використання сучасних версій протоколів (TLS 1.2 і вище), відмову від застарілих алгоритмів і забезпечення належного управління ключами [16]. Додатково підкреслюється необхідність ізоляції мережевих сегментів і ви-

користання захищених каналів передання даних, що дозволяє зменшити ризики несанкціонованого доступу до даних моніторингу.

Окрему увагу в сучасних дослідженнях приділено проблемі взаємодії криптографії та систем моніторингу. Збільшення частки зашифрованого мережевого трафіку ускладнює застосування традиційних методів аналізу, оскільки доступ до вмісту пакетів обмежується криптографічними механізмами захисту. За таких умов виникає потреба у використанні альтернативних підходів, таких як аналіз метаданих або інтеграція механізмів контролю на кінцевих вузлах [17]. Паралельно з цим використання сильного шифрування розглядається як необхідна умова забезпечення безпеки критичних систем.

Практика застосування криптографії у середовищах критичної інфраструктури також включає забезпечення цілісності та автентичності цифрових доказів. У дослідженнях, присвячених автоматизованому збору доказової інформації, показано, що використання криптографічних методів дозволяє гарантувати незмінність даних та їх придатність для подальшого аналізу [18]. Це є важливим не лише для реагування на інциденти, а й для проведення розслідувань.

Висновки

Інтелектуальний моніторинг, що спирається на моделі машинного навчання, доцільно розглядати як інтегрований елемент багаторівневої системи кіберзахисту, а не як автономне рішення. Його результативність формується на перетині якості вхідних даних, адекватності побудови моделей і здатності системи трансформувати аналітичні результати у практичні управлінські дії. Відсутність хоча б одного з цих компонентів суттєво знижує ефективність усього підходу.

Криптографічний захист машинного коду на рівні BIOS/UEFI та механізмів керування ключами виконує роль фундаменту довіри для всієї інформаційної системи. Стійкість цього рівня визначається не лише алгоритмічними властивостями, а передусім коректністю реалізації процедур перевірки підписів, надійністю інфраструктури керування ключами та використанням апаратних засобів довіри. Також у дослідженнях, що знаходяться у відкритому доступі, бракує узагальнених кількісних оцінок впливу таких механізмів на рівень кіберрезильєнтності, що ускладнює їх формалізоване порівняння та обґрунтування ефективності.

Криптографічні протоколи інтегруються у сучасні системи моніторингу та кіберзахисту як їх структурний компонент, забезпечуючи захист даних і довіру до процесів обміну інформацією. Їх практична ефективність визначається не стільки вибором окремих алгоритмів, скільки якіс-

тю вбудовування у загальну архітектуру системи. Водночас відсутність узагальнених емпіричних оцінок їхнього впливу на такі показники обмежує можливості статистично підтвердженого аналізу та свідчить про необхідність подальших досліджень у цьому напрямі.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. National Institute of Standards and Technology. Information Security Continuous Monitoring (ISCM) for Federal Information Systems and Organizations (NIST SP 800-137). URL: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-137.pdf. (дата звернення: 11.01.2026)
2. European Union Agency for Cybersecurity (ENISA). Threat Landscape 2023. URL: https://www.enisa.europa.eu/publications/enisa-threat-landscape-2023. (дата звернення: 13.01.2026)
3. Verizon. 2024 Data Breach Investigations Report. URL: https://www.verizon.com/business/resources/reports/dbir/. (дата звернення: 15.01.2026)
4. Chandola V., Banerjee A., Kumar V. Anomaly Detection: A Survey. ACM Computing Surveys, 2009. URL: https://doi.org/10.1145/1541880.1541882. (дата звернення: 16.01.2026)
5. NIST SP 800-57. Recommendation for Key Management. URL: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57pt1r5.pdf. (дата звернення: 01.02.2026).
6. UEFI Specification. Unified Extensible Firmware Interface Forum. URL: https://uefi.org/specifications (дата звернення: 24.01.2026).
7. NIST SP 800-82 Rev.2. Guide to Industrial Control Systems (ICS) Security. URL: https://nvlpubs.nist.gov/nistpubs/specialpublications/nist.sp.800-82r2.pdf. (дата звернення: 11.02.2026).
8. Trusted Platform Module Library Specification. URL: https://trustedcomputinggroup.org/resource/tpm-library-specification. (дата звернення: 09.03.2026).
9. Transport Layer Security (TLS). URL: https://en.wikipedia.org/wiki/Transport_Layer_Security (дата звернення: 16.02.2026).
10. Eclypsium Research. Firmware Security Analysis. URL: https://eclypsium.com/research/ (дата звернення: 27.02.2026).
11. ДСТУ 7564:2014. Криптографічний захист інформації. Функція гешування «Купина». — Київ: Мінекономрозвитку України, 2014. URL: https://usts.kiev.ua/wp-content/uploads/2020/07/dstu-7564-2014.pdf. (дата звернення: 13.03.2026).
12. NIST FIPS PUB 180-4. Secure Hash Standard (SHS). URL: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf (дата звернення: 19.02.2026).
13. Kelsey J., Schneier B. Second Preimages on n-bit Hash Functions for Much Less than 2^n Work. URL: https://www.schneier.com/wp-content/uploads/2016/02/paper-preimages.pdf (дата звернення: 14.01.2026).
14. SHA-2 Cryptographic Hash Functions. URL: https://en.wikipedia.org/wiki/SHA-2 (дата звернення: 12.03.2026).
15. Zabbix Documentation – Encryption. URL: https://www.zabbix.com/documentation/current/en/manual/encryption (дата звернення: 06.02.2026).
16. Cyber Security Standards and Best Practices. URL: https://www.scribd.com/document/936566211. (дата звернення: 15.03.2026).
17. Challenges of Monitoring Encrypted Traffic. URL: https://arxiv.org/abs/2104.09828 (дата звернення: 07.03.2026).
18. Elia P. Real-Time Automated Forensic Evidence Collection in Critical Infrastructure. URL: https://webthesis.biblio.polito.it/37933/1/tesi.pdf. (дата звернення: 02.03.2026).

REFERENCES

1. National Institute of Standards and Technology. Information Security Continuous Monitoring (ISCM) for Federal Information Systems and Organizations (NIST SP 800-137). URL: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-137.pdf (accessed: 11.01.2026).
2. European Union Agency for Cybersecurity (ENISA). Threat Landscape 2023. URL: https://www.enisa.europa.eu/publications/enisa-threat-landscape-2023. (accessed: 13.01.2026).
3. Verizon. (2024). Data Breach Investigations Report. URL: https://www.verizon.com/business/resources/reports/dbir/. (accessed: 15.01.2026).
4. Chandola, V., Banerjee, A., Kumar, V. (2009). Anomaly Detection: A Survey. ACM Computing Surveys. URL: https://doi.org/10.1145/1541880.1541882. (accessed: 16.01.2026).
5. National Institute of Standards and Technology. Recommendation for Key Management (NIST SP 800-57). URL: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57pt1r5.pdf. (accessed: 01.02.2026).
6. Unified Extensible Firmware Interface Forum. UEFI Specification. URL: https://uefi.org/specifications. (accessed: 24.01.2026).
7. National Institute of Standards and Technology. Guide to Industrial Control Systems (ICS) Security (NIST SP 800-82 Rev.2). URL: https://nvlpubs.nist.gov/nistpubs/specialpublications/nist.sp.800-82r2.pdf. (accessed: 11.02.2026).
8. Trusted Computing Group. Trusted Platform Module Library Specification. URL: https://trustedcomputinggroup.org/resource/tpm-library-specification. (accessed: 09.03.2026).
9. Transport Layer Security (TLS). URL: https://en.wikipedia.org/wiki/Transport_Layer_Security. (accessed: 16.02.2026).
10. Ministry of Economic Development and Trade of Ukraine. (2014). DSTU 7564:2014. Information security. Cryptographic hash function “Kupyna”. Kyiv. URL: https://usts.kiev.ua/wp-content/uploads/2020/07/dstu-7564-2014.pdf. (accessed: 13.03.2026).
11. National Institute of Standards and Technology. Secure Hash Standard (SHS) (FIPS PUB 180-4). URL: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf. (accessed: 19.02.2026).
12. Kelsey, J., Schneier, B. Second Preimages on n-bit Hash Functions for Much Less than 2^n Work. URL: https://www.schneier.com/wp-content/uploads/2016/02/paper-preimages.pdf. (accessed: 14.01.2026).
13. Eclypsium. Firmware Security Analysis. URL: https://eclypsium.com/research/. (accessed: 27.02.2026).
14. SHA-2 Cryptographic Hash Functions. URL: https://en.wikipedia.org/wiki/SHA-2. (accessed: 12.03.2026).
15. Zabbix SIA. Zabbix Documentation — Encryption. URL: https://www.zabbix.com/documentation/current/en/manual/encryption. (accessed: 06.02.2026).
16. Cyber Security Standards and Best Practices. URL: https://www.scribd.com/document/936566211. (accessed: 15.03.2026).
17. Challenges of Monitoring Encrypted Traffic. URL: https://arxiv.org/abs/2104.09828 (accessed: 07.03.2026).
18. Elia, P. Real-Time Automated Forensic Evidence Collection in Critical Infrastructure. URL: https://webthesis.biblio.polito.it/37933/1/tesi.pdf. (accessed: 02.03.2026).

I. Martyniuk, N. Zaika

INTELLIGENT MONITORING AND CRYPTOGRAPHIC PROTECTION
OF MACHINE CODE IN CRITICAL INFRASTRUCTURE SYSTEMS

Intelligent monitoring of cyber-physical systems based on a combination of continuous observation and machine learning models to detect anomalies and enhance the proactiveness of cyber defense is examined. The effectiveness of such approaches is determined by the quality of the data and the accuracy of the models, which are built on monitoring systems where cryptographic mechanisms are integrated directly into the process of data exchange between agents and the server. Cryptographic protection is designed to ensure the integrity of the security system at the machine code level. Algorithm verification is performed using test vectors in accordance with National Institute of Standards and Technology (NIST) practices. Collectively, this forms a comprehensive approach to the cyber resilience of critical infrastructure.

K e y w o r d s: intelligent monitoring, cyber-physical systems, critical infrastructure, machine learning, anomaly detection, cryptographic protection, cyber resilience.

МАРТИНЮК Ігор Вадимович, аспірант Національного університету «Київський авіаційний інститут»; інженер Державного науково-дослідного інституту технологій кібербезпеки. У 2011 р. закінчив Національний авіаційний університет. Область наукових досліджень — кібербезпека та захист інформації.

ЗАЇКА Назар Валентинович, аспірант Інституту проблем моделювання в енергетиці ім. Г.Є. Пухова НАН України; мол. наук. співробітник Державного науково-дослідного інституту технологій кібербезпеки. У 2014 р. закінчив Національний технічний університет України «Київський політехнічний інститут». Область наукових досліджень — комп'ютерні науки.

Дата надходження статті: 06.04.2026;
дата прийняття статті до друку після рецензування: 14.04.2026
дата публікації (оприлюднення) статті: .08.2026