DOI 10.31891/2307-5732-2021-303-6-49-54
УДК 004.056

ЯРМІЛКО А. В.
Черкаський національний університет ім. Б. Хмельницького
ORCID ID: 0000-0003-2062-2694
e-mail: a-ja@ukr.net
РОЗЛОМІЙ І. О.
Черкаський національний університет ім. Б. Хмельницького
ORCID ID: 0000-0001-5065-9004
e-mail: inna-roz@ukr.net
МИСЮРА Ю. О.
Черкаський національний університет ім. Б. Хмельницького
ORCID ID: 0000-0002-3844-2563
e-mail: julmisura@ukr.net

ЗАСТОСУВАННЯ ХЕШ-МЕТОДІВ У КРИПТОГРАФІЧНОМУ АНАЛІЗІ ПОТОКІВ ІНФОРМАЦІЇ

Серед основних завдань системи інформаційної безпеки – ідентифікація ймовірних або діючих джерел загроз роботі інформаційних систем, а також мінімізація наслідків несанкціонованого впливу на них. Дана стаття пропонує методи для комплексного вирішення завдань інформаційної безпеки та аналізу інформаційних потоків засобами криптографії та представляє досвід розробки засобів їх надійної реалізації. Розроблені методи дозволяють виявити фальсифікації у інформаційній частині надісланого повідомлення та виконати відновлення дійсної інформації. У процесі криптографічного аналізу область локалізації зміни у блоці даних встановлюється із застосуванням перехресного хешування, яке виконується поблоковим обчисленням хеш-функції інформаційного повідомлення. За допомогою розробленого на базі пропонованого нами підходу інструментального засобу досліджено ефективність виявлення включень у потоці даних в залежності від чутливості алгоритму. Експериментально виявлено залежність частки детектованих системою включень у блоці інформації від заданих параметрів алгоритму. Очікується, метод буде ефективним на етапі первинного сегментування потоку даних перед адресним застосуванням до виділених фрагментів спеціалізованих алгоритмів.

Ключові слова: інформаційна безпека, ідентифікація джерела надходження інформації, хеш-метод, перехресне хешування.

ANDII YARMILKO, INNA ROZLOMII, YULIYA MYSIURA
Bohdan Khmelnytsky National University of Cherkasy

USAGE OF HASH METHODS IN THE CRYPTOGRAPHIC DATA ANALYSIS

The tasks of information security system include identifying potential or actual sources of threat to system's work and minimizing consequences of unauthorized influence on it. While solving them, arises the need of restoration of the initial state of the information system, especially the data integrity. While performing information message analysis the other task may be finding differences between two data fragments or their instances. This paper offers methods of the complex solution of the information security tasks and the analysis of data streams using the means of cryptography and presents the experience of developing the reliable implementation of these methods. The developed methods allow detecting falsifications in data part of the sent message and restoring the initial message. During the cryptographic analysis, the area of change in a data block is localized using cross hashing which is performed by computing the hash of information message block by block. The result is the program implementation of the offered method of information stream analysis that is based on comparing three frames of input data. The effectiveness of detecting falsifications in a data stream depending on algorithm's sensitivity was researched with the developed instrument. The dependence of the share of falsifications detected by the system in the information block on the established maximum allowable relative deviation from the median and the properties of the input stream, in particular, the division of the input data into frames, was experimentally revealed. It is expected that the advantages of the method will be higher in the preliminary stage of data flow analysis related to its segmentation before addressing the selected fragments to more accurate and specialized algorithms.

Keywords: information security, identification of the incoming data source, hash method, cross hashing.

Постановка проблеми

Стабільна та надійна робота інформаційних систем неможлива без підвищеної уваги до системи інформаційної безпеки, порушення в роботі якої може чинити прямий чи опосередкований негативний вплив. Захищеність від загроз інформаційної безпеки безпосередньо залежить від ефективності системи управління інцидентами інформаційної безпеки щодо їх обробки [1]. Одним із завдань такої системи є ідентифікація ймовірного або діючого джерела загроз. Таке завдання може бути подане у вигляді задачі ідентифікації об'єктів аналізу (моніторингу) за поточними даними. Початкова інформація про конкретний об'єкт у ході дослідження може надходити з різних джерел і бути неповною та неточною. Як правило, така проблема пов'язана з нестачею, недостовірністю та суперечливістю вхідних даних, на основі яких можлива реалізація процедури ідентифікації порушника (загрози). Іншим завданням є мінімізація наслідків несанкціонованого впливу на інформаційну систему. При вирішенні такої задачі постає необхідність відновлення первинного стану інформаційної системи, зокрема – цілісності даних. З огляду на це розробка методів і засобів вилучення, обробки та аналізу інформації з додаткових джерел даних з метою підвищення ймовірності ідентифікації та усунення загроз безпеці інформаційної системи є актуальним завданням.

У загальному випадку при вирішенні завдань захисту комунікаційних систем можуть братися до уваги різні інциденти інформаційної безпеки. Зокрема, при виконанні аналізу інформаційних повідомлень

між компонентами деякого комунікаційного кластеру нас може цікавити не тільки виявлення помилок, але й виявлення відмінностей двох інформаційних фрагментів або їх примірників. Природа відмінностей може бути різною і їхня оцінка залежатиме від мети аналізу інформації. Якщо метою буде контроль збереження інформації в комунікаційному процесі, то відмінність трактуватиметься як помилка з усіма можливими наслідками реагування на неї. Якщо ж нас цікавитиме структура чи динаміка змін інформаційних фрагментів, то відмінності свідчитимуть про факт наявності динамічних процесів, їх локалізацію та інші супутні параметри.

Отже, розглядаючи інформаційну безпеку ситуативного комунікаційного кластеру людино-машинної системи, маємо забезпечити, по-перше, контроль складових системи над джерелами надходження повідомлень та, по-друге, виявлення пошкодження або зміни інформації у них. Забезпечити такий кластер безпечною комунікацією у відкритому просторі покликані засоби аутентифікації та ідентифікації користувачів. Аутентифікація забезпечує підтвердження достовірності суб'єкта [3]. В теорії інформаційної безпеки етап аутентифікації визначається як найважливіший, тому даний процес вимагає надійної та якісної реалізації. Одним з найпоширеніших способів аутентифікації є парольна аутентифікація. Припущення, що використання пароля в якості аутентифікаційного фактору ще тривалий час буде найбільш розповсюдженим способом вирішення задач підтвердження достовірності, має стійке підґрунтя. Насамперед, це пов'язано з простотою реалізації і низькими витратами на підтримання такої технології [4]. Поряд з цим, комплексне вирішення завдань інформаційної безпеки потребує й пошуку ефективних методів відновлення пошкодженої або зміненої інформації та створення засобів їх надійної реалізації.

Аналіз останніх джерел

Питання порушення інформаційної безпеки комунікаційних мереж та методи протидії їм є актуальними та перебувають у фокусі уваги вітчизняних та зарубіжних дослідників [1; 2; 5; 6]. Враховуючи сучасні загальноприйняті підходи, зупинимо увагу на наступних моментах інформаційної безпеки: доступність – це можливість за певний час отримувати необхідний інформаційний сервіс; цілісність – полягає в актуальності та достовірності інформації; конфіденційність – це захищеність від несанкціонованого доступу.

Метою роботи є підвищення ймовірності ідентифікації джерел загроз інформаційної безпеки комунікаційного кластера шляхом використання методів хешування. Вирішення завдання їхньої ефективної ідентифікації обумовлює необхідність розробки нових технічних рішень, здатних підвищити якість ідентифікаційних даних про потенційних порушників. Таким технічним рішенням може бути комплексне застосування парольної аутентифікації та хеш-методів.

Виклад основного матеріалу

У цьому дослідженні ми виходили з припущення, що повідомлення, якими обмінюються компоненти ситуативного кластера, являють собою блоки інформації, складені з аутентифікаційної та інформаційної складової. У процесі ідентифікації користувача модуль-приймач порівнює аутентифікаційні дані повідомлення, отриманого від модуля-передавача, з даними, які зберігаються в доступній у межах комунікаційного кластера базі даних. В зв'язку з тим, що на етапі формування аутентифікатора у структурі кожного окремого повідомлення аутентифікаційні дані можуть бути введені некоректно (внаслідок навмисного втручання чи випадкових процесів), має вирішуватися задача виявлення таких фальсифікацій. Застосування з цією метою методів криптографії (encryption) є розповсюдженою практикою, прикладом якої є захищений протокол HTTPS, криптографічні протоколи SSL, TLS. Після встановлення достовірності аутентифікатора, необхідно перевірити відсутність фальсифікацій у інформаційній частині надісланого повідомлення, а в разі їхнього виявлення – виконати відновлення дійсної інформації.

Аналіз змісту інформаційної частини повідомлення також частково або повністю може бути виконаний криптографічними методами. Оскільки окремим випадком такого аналізу може бути встановлення факту та області локалізації зміни у блоці даних, необхідною є розробка методів, які дозволяють виявити декілька змін у окремому блоці інформації та встановити їх координати з метою ідентифікації та формування відповіді. Нами для аналізу спільно використовуваних інформаційних ресурсів були запропоновані методи, які поєднують принципи організації парольної автентифікації та ідентифікації з методами хешування.

У даному дослідженні не акцентувалася увага на форматі і розмірах автентифікаційного та інформаційного компонентів повідомлення. Тому у подальшому викладенні, зважаючи на однотипність задач виявлення фальсифікацій у кожній зі складових повідомлення, вони будуть розглядатися як блоки даних.

Для виявлення фальсифікованих блоків інформаційного потоку скористаємось методом перехресного хешування [7]. Суть запропонованого методу полягає в поблоковому обчисленні хеш-функції інформаційного повідомлення. Якщо обчислити значення хеш-функції і в горизонтальних, і в вертикальних блоках даних, то це дасть змогу, застосовуючи перевірку значень хеш-функції за формулами (1) та (2), виявити порушення цілісності в горизонтальному та вертикальному блоці [7].

Як зазначалося раніше, обчислення хеш-функції тільки в горизонтальних або тільки в вертикальних блоках даних не дає можливості виявлення помилок в конкретному записі. Однак, визначивши рядок (горизонтальний блок даних) та стовпець (вертикальний блок даних), де відбулися зміни, на їх перетині можна виявити фальсифікований фрагмент повідомлення.

Згідно методу, повідомлення розбивається на блоки A_{ij}, де $i \in [1,n], j \in [1,n]$. На наступному кроці необхідно обчислити значення хеш-функції кожного блоку інформації. Для розгляду цієї процедури введемо поняття міні хеш-функції $F^{\min}$. Обчислення міні хеш-функцій виконуються за аналогією з обчисленням хеш-функції всього інформаційного повідомлення. Для знаходження міні хеш-функцій може бути використаний будь-який алгоритм із запропонованих в статті [8].

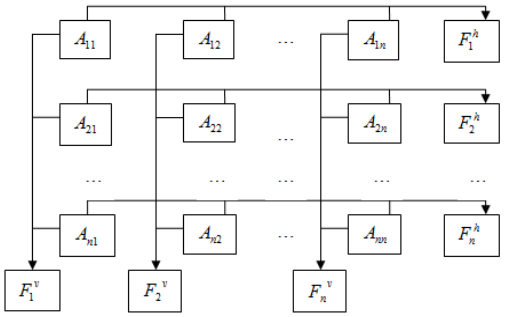

Рис. 1. Схема обчислення хеш-функції в горизонтальних/вертикальних блоках даних

Позначимо через послідовність $F_{11}, F_{12}, ..., F_{1n}$ обчислені значення хеш-функцій кожного горизонтального блоку інформації. Значення міні хеш-функцій горизонтальних блоків можна представити у вигляді системи:

$$\begin{cases} F_{21} = F(A_{11}) \cup F(A_{21}) \cup ... \cup F(A_{n1}); \\ F_{22} = F(A_{12}) \cup F(A_{22}) \cup ... \cup F(A_{n2}); \\ ..... \\ F_{2n} = F(A_{1n}) \cup F(A_{2n}) \cup ... \cup F(A_{nn}). \end{cases} \quad (1)$$

Аналогічно, $F_{21}, F_{22}, ..., F_{2n}$ – міні хеш-функції вертикальних блоків інформації. Значення міні хеш-функцій вертикальних блоків представлені системою:

$$\begin{cases} F_{21} = F(A_{11}) \cup F(A_{21}) \cup ... \cup F(A_{n1}); \\ F_{22} = F(A_{12}) \cup F(A_{22}) \cup ... \cup F(A_{n2}); \\ ..... \\ F_{2n} = F(A_{1n}) \cup F(A_{2n}) \cup ... \cup F(A_{nn}). \end{cases} \quad (2)$$

Процес отримання міні хеш-функцій повідомлення можна представити також таблично (рис. 4).

A_{11}	A_{12}	A_{13}	A_{14}	...	A_{1n}	Обчислення міні хеш-функцій $F^{\min}(A_{ie})$	$F_{11}^{\min}$
A_{21}	A_{22}	A_{23}	A_{24}	...	A_{2n}		$F_{12}^{\min}$
A_{31}	A_{32}	A_{33}	A_{34}	...	A_{3n}		$F_{13}^{\min}$
A_{41}	A_{42}	A_{43}	A_{44}	...	A_{4n}		$F_{14}^{\min}$
...	...	...	...	...	...		
A_{n1}	A_{n2}	A_{n3}	A_{n4}	...	A_{nn}		$F_{1e}^{\min}$
Обчислення міні хеш-функцій $F^{\min}(A_{ej})$							
$F_{21}^{\min}$	$F_{22}^{\min}$	$F_{23}^{\min}$	$F_{24}^{\min}$	...	$F_{2e}^{\min}$		

Рис. 2. Схема процесу отримання міні хеш-функцій

У разі порушення цілісності повідомлення зміняться і значення міні хеш-функцій. На основі того, які конкретно міні хеш-функції при перевірці змінили своє значення, можна судити, в якому саме інформаційному блоці повідомлення відбулося порушення цілісності. Тобто, зміна горизонтальної міні хеш-функції вкаже на рядок, вертикальної – на стовпець у двовимірній структурі записів повідомлення, на перетині яких була змінена інформація. З прикладу рис. 3 видно, що якщо при перевірці виявлено, що міні хеш-функції F_{24} і F_{13} змінили своє значення, це означає, що в інформаційному блоці A_{34} відбулися зміни, тобто даний фрагмент повідомлення був фальсифікований:

$$F_{13} \bigcap F_{24} \to A_{34}. \quad (3)$$

На базі описаного вище методу був розроблений метод криптографічного захисту повідомлень, який має потенціал не тільки виявлення, але й усунення будь-якої кількості пошкоджень блоків інформації, яка передається між вузлами комунікаційної мережі [9].

Одним з прикладних напрямів застосування пропонованих у [7-9] методів є аналіз послідовних потоків даних будь-якої природи, зокрема – аудіо- та відеоповідомлень. Метою такого аналізу може бути знаходження сторонніх включень – наприклад, нових об'єктів на зображенні чи у звуковому повідомленні – у потоці інформації. Зазначимо, що криптографічні методи – не єдиний спосіб вирішення задачі. Так, з цією метою можливе використання алгоритмів Deep Learning та засобів сторонніх бібліотек [10]. Але у

альтернативного підходу є один важливий недолік: він підходить лише для одного типу даних (відео, звук, т. і.). Тому аналіз на базі представлених у [7–9] методів вигідно відрізняється забезпеченням незалежності алгоритму розв'язання задачі від типу вхідного потоку даних.

A_{11}	A_{12}	A_{13}	A_{14}	...	A_{1n}	$F_{11}^{\min}$
A_{21}	A_{22}	A_{23}	A_{24}	...	A_{2n}	$F_{12}^{\min}$
A_{31}	A_{32}	A_{33}	A_{34}	...	A_{3n}	$F_{13}^{\min}$
A_{41}	A_{42}	A_{43}	A_{44}	...	A_{4n}	$F_{14}^{\min}$
...	...	...	...	...	...	...
A_{n1}	A_{n2}	A_{n3}	A_{n4}	...	A_{nn}	$F_{1n}^{\min}$
$F_{21}^{\min}$	$F_{22}^{\min}$	$F_{23}^{\min}$	$F_{24}^{\min}$	...	$F_{2n}^{\min}$	

Рис. 3. Локалізація пошкодженого інформаційного блоку A_{34} в процесі виявлення фальсифікацій у повідомленні за виявленими змінами міні хеш-функцій F_{24} і F_{13}.

Відомо, що дані будь-якого типу можна представити у двійковому вигляді. Звідси слідує, що потік даних може представляється у вигляді двійкового потоку. Але комп'ютерна система не може обробляти весь неперервний потік відразу. Це означає, що дані необхідно подавати порціями деякого заздалегідь визначеного розміру – назвемо ці порції фреймами.

Запропонований нами підхід до аналізу інформаційних потоків ґрунтується на порівнянні трьох сусідніх фреймів. Розглянемо його структуру та програмну реалізацію докладніше.

Задамо нашій програмній системі буфер розміром у три фрейми. Це потрібно для того, щоб програма приймала тільки три фрейми на поточному кроці аналізу потоку даних. Далі кожен із фреймів ділимо на менші фрагменти даних, розмір кожного з яких не перевищує *k* байт і визначається за формулою:

$$k = \log_2 Size_f,$$

де $Size_f$ – розмір фрейма.

Далі обчислюємо хеш-функції відповідних фрагментів поточної трійки фреймів (тобто на кожному кроці обчислюємо хеші *i*-го фрагмента першого фрейма, *i*-го фрагмента другого фрейма та *i*-того фрагмента третього фрейма).

Потім для кожного з цих трьох хешів обчислюємо відносне відхилення від середнього:

$$d_{ij} = \frac{\left|h_{ij} - \overline{h_{ij}}\right|}{\overline{h_{ij}}},$$

де h_{ij} – хеш *i*-го фрагмента *j*-го фрейма з трійки, $\overline{h_{ij}}$ – медіана трійки обчислених хешів. Залежно від використовуваної хеш-функції може додатково знадобитися виконання масштабування результату шляхом підвищення його значення на кілька порядків для того, щоб не оперувати занадто малими числами.

Для практичної реалізації та експериментальної перевірки запропонованого алгоритму було застосовано спеціальну програму. При її створенні використано мову C#, у якій є клас FileStream [11], що має усі необхідні для підтримання зазначеного аналізу методи. Разом з тим варто мати на увазі, що для реального впровадження подібного алгоритму у прикладні системи в якості інструменту розробки слід використати мову C або C++, оскільки реалізація програми цими мовами у загальному випадку забезпечуватиме більшу швидкодію.

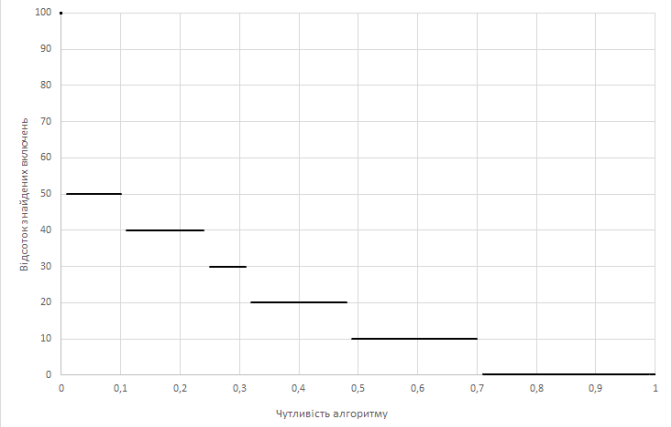

Рис. 4. Ефективність виявлення включень у потоці даних залежно від чутливості алгоритму при введенні фреймів по 32 байти

Зазначимо, що у прикладних задачах не завжди є сенс знаходити усі відхилення від медіани. Найчастіше має місце деяке допустиме відносне відхилення, пов'язане з особливостями джерела потоку даних. Назвемо цю величину чутливістю алгоритму. Якщо відхилення від медіани перевищує цю величину, то вважатимемо, що у цьому фрагменті даних є стороннє включення. Відповідно, найбільшу чутливість алгоритм матиме при допустимому відхиленні, рівному 0, а найменшу – при відносному відхиленні, рівному 1.

Дослідимо ефективність виявлення сторонніх включень зі зміною чутливості за допомогою отриманої програмної реалізації нашого алгоритму.

Візьмемо для прикладу деякий потік даних, що містить у собі три фрейми по 32 байти даних. Змінюючи чутливість алгоритму, отримуємо залежність, представлену на рис. 4.

Тепер візьмемо потік даних, що містить три фрейми по 256 байтів і подивимось на залежність відсотка знайдених включень від чутливості алгоритму (рис. 5).

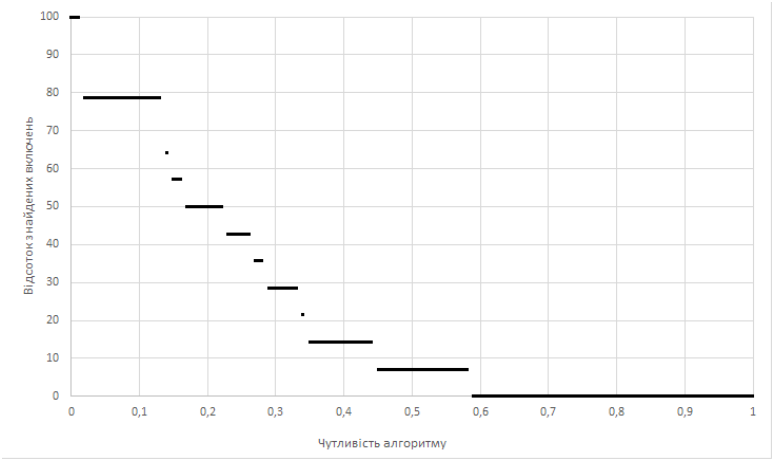

Рис. 5. Ефективність виявлення включень у потоці даних залежно від чутливості алгоритму при введенні фреймів по 256 байтів

Як бачимо, спостерігається залежність частки детектованих системою включень у блоці інформації від встановленого експериментатором максимального допустимого відносного відхилення від медіани та властивостей вхідного потоку, зокрема – розбиття вхідних даних на фрейми.

Висновки

Отже, нами було запропоновано та досліджено один із способів знаходження сторонніх включень у деякому послідовному потоці даних на основі криптографічних підходів. Пропоноване рішення є достатньо універсальним щодо вхідного потоку даних та спирається на оригінальний метод виявлення, локалізації та виправлення заданої кількості пошкоджень в інформаційному фрагменті. Отримано експериментальні залежності ефективності виявлення сторонніх включень у потоці даних в залежності від параметру чутливості алгоритму.

Отримані результати можуть бути корисні у розробці систем безпеки та аналізу даних різних типів. Універсальність запропонованого алгоритму є одночасно його перевагою і недоліком. Недолік полягає у тому, що алгоритм не враховує специфічні особливості формату вхідних даних, тому його точність може бути менша, ніж у алгоритмів, пристосованих до конкретного формату даних. Імовірно, новий метод переважно буде ефективним на попередньому етапі аналізу потоку даних, пов'язаному з його сегментуванням перед адресним застосуванням до виділених фрагментів більш точних та спеціалізованих алгоритмів.

Література

1. Рыженкова А. (2014) Управление инцидентами информационной безопасности: о чем говорят стандарты. Connect. Мир информационных технологий. 7–8. 62–65.

2. Соколов А.В. (2010) Методика оценки эффективности поиска по нечетким характеристикам в автоматизированных информационных системах. Автоматизация и современные технологии. М.: Лань. 3. 25–27.

3. Ignacio Velásquez, Angélica Caro, Alfonso Rodríguez (2018) Authentication schemes and methods: A systematic literature review. Information and Software Technology vol.94. 30–37.

4. R. Song (2010) Advanced smart card based password authentication protocol. Computer Standards and Interfaces. 5-6(32). 321–325.

5. Солодовников В.И., Евдокимов И.А. (2016) Анализ криптостойкости нейросетевого алгоритма симметричного шифрования. Новые информационные технологии в автоматизированных системах. 263–269.

6. Grimaila M.R. (2011) Design and Analysis of a Dynamically Configured Log-based Distributed Security Event Detection Methodology. Journal of Defense Modeling and Simulation: Applications, Methodology, Technology. 9 (3).

7. Розломій І.О., Косенюк Г.В. (2018) Виявлення порушень цілісності електронного документу шляхом перехресного хешування. Вісник ХНУ. Радіотехніка, електроніка та телекомунікації. 5 (265). 32–35.

8. Розломій І.О. (2016) Методи обчислення хеш-функції електронного документу на основі матричних криптографічних перетворень. Вісник ЧДТУ. Технічні науки. 4. 88–94.

9. Ярмілко А.В. (2020) Методи підвищення захищеності інформаційних потоків із застосуванням хешування / А.В. Ярмілко, І.О. Розломій, Г.В. Косенюк. Математичне та імітаційне моделювання систем. МОДС 2020: тези доповідей П'ятнадцятої міжнародної науково-практичної. Чернігів : ЧНТУ. 205–208.

10. Python: распознавание объектов в реальном времени : веб-сайт. URL: https://proglib.io/p/real-time-object-detection (дата звернення: 10.12.2021).

11. FileStream. Чтение и запись файла: веб-сайт. URL: https://metanit.com/sharp/tutorial/5.4.php (дата звернення: 10.12.2021).

References

1. Ryzhenkova A. (2014) Upravleniye intsidentami informatsionnoy bezopasnosti: o chem govoryat standarty. Information security incident management: what the standards say. Connect. Mir informatsionnykh tekhnologiy, 2014. 7–8. 62–65.

2. Sokolov A.V. (2010) Metodika otsenki effektivnosti poiska po nechetkim kharakteristikam v avtomatizirovannykh informatsionnykh sistemakh. Automation and modern technologies. Moscow: Deer. 3. 25-27.

3. Ignacio Velásquez, Angélica Caro, Alfonso Rodríguez (2018) Authentication schemes and methods: A systematic literature review Information and Software Technology vol.94. 30–37.

4. R. Song (2010) Advanced smart card based password authentication protocol. Computer Standards and Interfaces. 5–6(32). 321–325.

5. Solodovnikov V.I. Evdokimov I.A. (2016) Analysis of the cryptographic strength of the neural network symmetric encryption algorithm. Novyye informatsionnyye tekhnologii v avtomatizirovannykh sistemakh. Moscow. 263–269.

6. Grimaila M.R. (2011) Design and Analysis of a Dynamically Configured Log-based Distributed Security Event Detection Methodology. Journal of Defense Modeling and Simulation: Applications, Methodology, Technology. 9 (3).

7. Rozlomii I.O., Koseniuk H.V. (2018) Detection of violations of the integrity of the electronic document by cross-hashing. // Visnyk KhNU. Radiotekhnika, elektronika ta telekomunikatsii. 5 (265). P. 32–35.

8. Rozlomii I.O. (2016) Methods for calculating the hash function of an electronic document based on matrix cryptographic transformations. Visnyk ChDTU. Tekhnichni nauky. 4. 88–94.

9. Yarmilko A.V., Rozlomii I.O., Koseniuk H.V. (2020) Methods of increasing the security of information flows with the use of hashing. Matematychne ta imitatsiine modeliuvannia system. MODS 2020. Chernihiv: ChNTU. 205–208.

10. Python: raspoznavaniye obyektov v realnom vremeni / Python: real time object detection: URL: https://proglib.io/p/real-time-object-detection (retrieved: 10.12.2021).

11. FileStream. Chteniye i zapis fayla / FileStream. Reading and writing to file: URL: https://metanit.com/sharp/tutorial/5.4.php (retrieved: 10.12.2021).

Рецензія/Peer review : 13.10.2021 Надрукована/Printed :30.12.2021