DOI: https://doi.org/10.33216/2222-3428-2023-24-2
УДК 004.056

Рязанцев О.І., Кардашук В.С., Кравцов С.В.

ЗАСТОСУВАННЯ ФУНКЦІЙ ХЕШУВАННЯ ДЛЯ СТВОРЕННЯ КВАЛІФІКОВАНОГО ЕЛЕКТРОННОГО ПІДПИСУ

У статті розглянуті сучасні алгоритми створення кваліфікованого електронного підпису (КЕП) з застосуванням функцій хешування. Проведено аналіз алгоритмів з метою програмної реалізації систем електронного підпису на мові Java с використанням бібліотеки Open SSL.
Розглянута нормативно-правова база створення електронного підпису та відмічена його уразливість. Серед недоліків існуючих схем формування електронного підпису відмічається повільна робота алгоритмів формування, перевірки підпису та обмеження на довжину повідомлення. Програмні рішення, що використовуються, мають обмеження на довжину повідомлення, його розбиття на фрагменти і підпис кожного фрагмента. Таке рішення часто неприйнятно для використання на практиці, так як збільшує обсяг повідомлення, час виконання процедури створення і перевірки електронного підпису.
Серед характеристик криптосистеми з відкритими ключами визначено, що слабкою ланкою в таких системах є ключів. Відзначено, що криптографічні системи, розроблені на основі алгоритму RSA, породжують аналоги в алгоритмах на еліптичних кривих. В результаті дослідження вироблені практичні рекомендації по довжині блоків хеш-функції і довжині ключа алгоритмів електронного підпису.
При розробленні інфраструктури відкритих ключів слід орієнтуватись на тести Національного інституту по стандартам і технологіям США (NIST). Дослідження на основі тестів NIST показало недосконаліть засобів електронного підпису та шифрування документів. В результаті дослідження намічені подальші шляхи удосконалення алгоритмів електронного підпису, що направлені на зменшення кількості операцій кодування та збільшення криптографічної стійкості. В результаті дослідження вироблені практичні рекомендації по довжині блоків хеш-функції і довжині ключа алгоритмів КЕП.
Ключові слова: *кваліфікований електронний підпис, хеш-функція, програмна реалізація, бібліотеки Open SSL, алгоритм, розподіл ключів, тести NIST.*

Актуальність дослідження новітніх методів забезпечення безпеки інформації показує, що зростає роль достовірності інформації, що передається по каналам зв'язку. Одним з найважливіших положень Закону № 2155 [1] є взаємне визнання українських та іноземних сертифікатів відкритих ключів та електронних підписів. Поняття «електронний цифровий підпис (ЕЦП)», згідно закону, замінено на «кваліфікований електронний підпис» (КЕП). Важливу роль в цій передачі відіграє ідентифікація користувачів на основі електронного підпису. ЕЦП (КЕП) за майже сорокарічну історію свого існування пройшов стрімку еволюцію від математичної ідеї Уільяма Діффі і Мартіна Геллмана, висловленої у 1976 р. [2], до невід'ємного елементу сучасного захищеного мережевого електронного документообігу.

Постановка проблеми. Незважаючи на повсякденне використання КЕП, на сьогоднішній день склалася ситуація коли паралельно існують і застосовуються різні державні та комерційні стандарти КЕП, при цьому користувач системи КЕП зазвичай погано уявляє собі ефективність і якість застосовуваної їм системи, ступінь її криптостійкості і може реально оцінити тільки зручність інтерфейсу програмної реалізації КЕП і її швидкодію. Практично всі протоколи КЕП використовують хеш-функції, що приймають довільний блок даних та шляхом математичного перетворення повертають блок фіксованого розміру. Порівняльний аналіз систем КЕП повинен супроводжуватися дослідженням показників ефективності застосовуваних хеш-функцій.

Критеріями для порівняння систем криптографічного перетворення є зручність програмної реалізації системи КЕП та час її роботи, яке як зазначено вище, безпосередньо пов'язано із зручністю користування.

В силу істотної нелінійності, як алгоритмів криптографічного хешування, так і алгоритмів КЕП, не представляється можливим отримати хоча б грубі оцінки обчислювальної складності систем КЕП. Тому основним методом їх порівняльного аналізу є програмна реалізація різних систем КЕП з подальшим їх тестуванням і профілюванням.

Аналіз останніх досліджень і публікацій. Незважаючи на велику кількість публікацій з криптографічних протоколів, куди входить і КЕП [3], залишається невідомий ґрунтовний порівняльний аналіз існуючих і застосовуваних систем КЕП. В цьому плані завдання порівняльного дослідження різних алгоритмів і систем КЕП по набору критеріїв ефективності є досить актуальним.

Метою статті є дослідження сучасних систем КЕП шляхом їх програмного моделювання. За критеріями зручності програмної реалізації і швидкодії це дозволить визначити відповідність їх функціональним можливостям для практичного застосування щодо довжини блоків хеш-функції і довжини ключа алгоритмів КЕП.

Вирішення проблеми. Фахівцями достатньо вивчені питання ненадійності практичних реалізацій алгоритмів формування і перевірки КЕП. Аналіз вразливостей існуючих систем КЕП дозволяє фахівцям стверджувати, що «число вразливих точок КЕП, що базується на шифруванні з відкритим ключем, настільки велике, що доцільність використання подібного методу викликає великі сумніви». Першою причиною є

вразливість алгоритму шифрування з відкритим ключем для КЕП. Друга причина – передача відкритого ключа в одному «конверті» з електронним підписом (у структуру КЕП, згідно з міжнародним стандартом CCITT X.509 [4], входить не тільки відкритий ключ відправника, а й його ім'я, серійний номер КЕП, назва і власний КЕП уповноваженої організації, що видала набір секретного і відкритого ключів). В даний час реалізовані і опубліковані схеми механізмів злому КЕП, засновані на генерації нової пари (відкритий, секретний) ключів і включення нового відкритого ключа в «конверт» КЕП. Надійність системи КЕП складається з надійності окремих елементів, до яких крім алгоритмів вироблення і перевірки підпису відносяться механізм генерації, розподілу ключів і ряд інших елементів. На надійність системи КЕП важливий вплив надає розподіл ключів між абонентами обміну.

На практиці такий розподіл здійснюється двома способами: створенням центру генерації і розподілу ключів та прямим обміном ключами між абонентами. У першому випадку компрометація центру призводить до компрометації всієї інформації, що передається. У другому випадку – необхідно забезпечити ідентифікацію кожного абонента. Помилки реалізації систем КЕП істотно впливають на зниження рівня надійності схем.

Поширеними помилками є: періодичне повторення одних і тих же значень в алгоритмами генерації випадкових чисел. Генератори випадкових чисел видають псевдовипадкові числа (при кожному виконанні алгоритму генерації випадкових чисел отримаємо той же самий список випадкових чисел). Тому часто на практиці використовують спеціальну апаратуру для генерації справжніх випадкових чисел.

В основі тестів лежать так звані критерії простоти. Існує два типи критеріїв простоти: детерміновані і ймовірнісні [5]. Детерміновані тести дозволяють довести, що число, яке тестується, – просте. Практично застосовувані детерміновані тести здатні дати позитивну відповідь не для кожного простого числа, оскільки використовують лише достатні умови простоти.

Детермінований тест використовується, наприклад, в процедурах обчислення несекретних параметрів цифрового підпису типу Ель-Гамаля [6].

Результати аналізу асиметричних алгоритмів зведені в таблицю 1.

Таблиця 1

Основні характеристики алгоритмів КЕП

Алгоритм	Хеш-функція	Рекомендований розмір відкритого ключа, біт	Рекомендований розмір закритого ключа	Рік створення, країна
DSA	SHA-1 або SHA-2	1024-3072	160-256	1994, США
ECDSA	SHA-1 або SHA-2	112-320	80-512	1999, США
ГОСТ Р34.10-2012	ГОСТ Р34.112012	80-320	256-512	2012, РФ
ДСТУ 4145-2002	ГОСТ 34311.95	162-768	256-1024	2002, Україна

Аналізуючи застосування в КЕП криптографічних хеш-функцій, відмітимо що всі вони засновані на алгоритмах блокового шифрування. В даному дослідженні КЕП застосовані хеш-функції MD-2 MD-5 SHA-1 SHA-2 (з розмірами блоку 256, 384, 512 біт), також ГОСТ 34311.95 і ГОСТ Р34.112012, специфіковані відповідними стандартами КЕП.

Система спроектована відповідно до основних вимог компонентного програмування. Кожний модуль системи відповідає за реалізацію чіткого та мінімального переліку функцій та взаємодії з іншими компонентами за допомогою гнучкого інтерфейсу. Для роботи Web-інтерфейсу необхідно мати сучасний Web-браузер та стабільний доступ до мережі Інтернет. Для запуску системи на сервері необхідно принаймні 2 гігабайти оперативної пам’яті та Java Development Kit версії 8 або вище. Шар представлення, реалізований у вигляді web-інтерфейсу, сполучається контролером із шаром сервісу, в якому можна відокремити самостійні елементи, відповідальні за збереження інформації про користувачів та безпосередньо створення і перевірку цифрового підпису. Для порівняльного аналізу систем КЕП шляхом моделювання розроблено універсальний програмний засіб з наступними функціями блоків: хешування; генерації відкритого ключа; генерації секретного ключа; верифікації КЕП.

Програма розроблена на мові Java з застосуванням бібліотеки криптографічних примітивів Open SSL [7] (secure socket layer – система безпечних сокетів) – криптографічний пакет з відкритим вихідним кодом, що надає засоби для профілювання програмних фрагментів за витратами процесорного часу.

Структурна схема створеної системи зображена на рисунку 1.

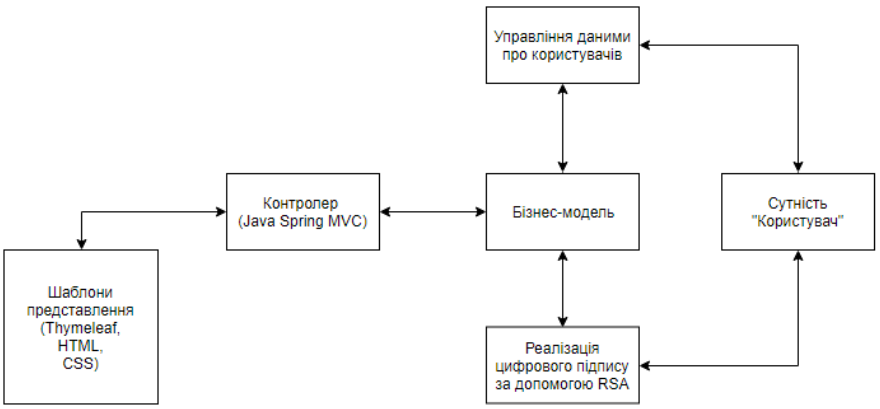

Рисунок 1 — Структурна схема створеної системи

,

Діаграма прецедентів, що зображена на рисунку 2, описує набір дій, які створюють цілісний та повний функціональний сценарій роботи користувача з системою.

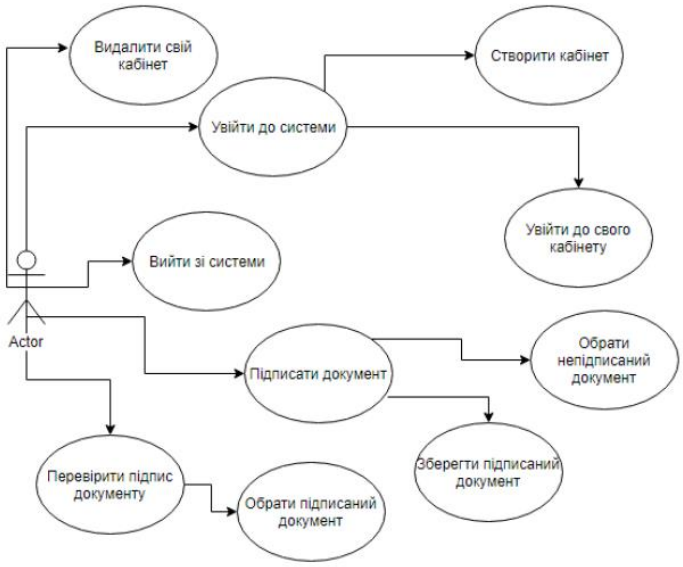

Рисунок 2 — Діаграма прецедентів системи

Задачею модулю взаємодії зі сховищем даних є взаємна конвертація між вище описаною структурою даних та фізичним форматом, у якому вони зберігаються у певних файлах.

Аналізуючи поставлену задачу, програмне забезпечення реалізоване у вигляді веб-додатку. Це оптимальне рішення, тому що робота передбачає застосування програми користувачами, які знаходяться віддалено один від одного, що дозволяє зробити систему універсальною та сумісною з більшою кількістю пристроїв.

Засоби розробки. Логіка додатку та внутрішні обчислювальні методи реалізовані за допомогою мови програмування JAVA 8 версії. Для створення користувацького інтерфейсу використано мову розмітки HTML та мову стилів CSS. Для збірки проекту використано фреймворк Maven. Для зручності написання та побудови структури проекту використано фреймворк Spring. Використані зовнішня бібліотека Integer для множення великих чисел без колізії. В якості сховища даних обрано файл json формату, дані в якому видозмінені за допомогою методів серіалізації/десеріалізації.

Середовище розробки. Ссередовищем для написання коду на мові програмування Java є IntelliJ IDEA від Jet Brains. IntelliJ IDEA – це, перш за все, середовище розробки для Java, включаючи Java 8. Дослідження моделювання КЕП в цілому і хеш-функцій проводилося на апаратно-програмних конфігураціях двох типів комп’ютерів:

Конфігурація А: CPU: Pentium 987 (ядро Sandy Bridge) 1.5 ГГц (2 МБ L1 cache); RAM: 4 ГБ DDR3 1300 МГц; ОС: Windows 10.

Конфігурація Б: CPU: Core i5 3240T (ядро Ivy Bridge) 2.9 ГГц (3 МБ L1 cache); RAM: 8 ГБ DDR3 1600 МГц; ОС: Windows 10.

Результати досліджень. В процесі моделювання досліджена продуктивність (швидкість роботи) хеш-функцій. Усереднені по базі файлів документів різних форматів і розмірів обсягом $3 \cdot 10^4$ файлів. Показники швидкості оцінені програмними засобами Java і Open SSL (таблиця 2).

Таблиця 2

Результати дослідження швидкодії роботи хеш-функцій

Функція хешування	Кількість раундів	Мова реалізації	Швидкість роботи на конфігурації А, Мбіт/с	Швидкість роботи на конфігурації Б, Мбіт/с
SHA-1	80	Java	206	344
SHA-2 (256)	64		81	135
SHA-2 (512)	64		41	68
ГОСТ 34311.95	256		4928	83
ГОСТ 34.112012	256		2458	46

За даними таблиці 2 слідує, що хешування за допомогою SHA-2 з довжиною блоку 512 біт є найбільш продуктивною обчислювальною процедурою. З іншого боку, алгоритм SHA-2 є досить сучасною і перспективною функцією хешування, що гарантує її криптографічну стійкість.

Проведений аналіз при дослідженні алгоритмів КЕП показав, що програмна реалізація алгоритмів ГОСТ Р34.10-2012 і ДСТУ 4145-2002 істотно складніше алгоритмів DSA і ECDSA, тому при необхідних довжинах ключів ГОСТ Р 34.10-2012 і ДСТУ 4145-2002 практично не витримують конкуренції з алгоритмами DSA і ECDSA по швидкодії.

Результати дослідження щодо оцінки швидкодії систем КЕП на базі алгоритмів DSA і ECDSA для обох апаратних конфігурацій в залежності від довжини ключа з розбивкою по етапах представлені в таблицях 3,4.

Таблиця 3

Часовий аналіз етапів DSA

Розмір ключа, біт	512		1024		2048	
Конфігурація	А	Б	А	Б	А	Б
Генерація ключа, мс	100	80	180	120	620	490
Генерація КЕП, мс	15	12	25	20	130	105
Верифікація, мс	80	60	170	120	220	205

Таблиця 4

Часовий аналіз етапів ECDSA

Розмір ключа, біт	512		1024		2048	
Конфігурація	А	Б	А	Б	А	Б
Генерація ключа, мс	50	25	65	48	160	140
Генерація КЕП, мс	75	60	100	90	210	200
Верифікація, мс	125	85	250	255	310	380

Аналіз результатів дослідження дозволяє зробити наступні висновки. Підпис на базі алгоритму ECDSA вимагає істотно більшого часу на верифікацію, ніж на формування ключів. Для конфігурації А користувача цілком прийнятною з точки зору швидкодії для системи DSA рекомендувати довжину ключа 1024 біта, а для системи ECDSA - довжину ключа 163 біта. Для конфігурації Б ці рекомендації зберігаються.

Загальні рекомендації, отримані в результаті дослідження, сформулюємо таким чином. Найбільш ефективними по сформульованим критеріям зручності програмної реалізації і швидкодії є системи КЕП DSA і ECDSA з застосуванням хеш-функції SHA-2 довжиною блоку 256 або 512 біт. При порівнянні систем DSA і ECDSA останню, безумовно, слід вважати більш перспективною, оскільки більшість діючих стандартів КЕП орієнтовані на еліптичну криптографію.

Щоб оцінити роботу представлених програмних засобів, що реалізують технології КЕП та шифрування документів на основі тестів NIST, виконано порівняльне тестування коректності їх роботи. Для порівняння були відібрані такі програмні продукти: Litoria Desktop (збірка 1.0.44), КриптоАРМ (збірка 5.4.1.37), Admin-PKI (збірка 5.1.1.1), КАРМА (збірка 56.0.80), КриптоНУЦ (збірка 1.12.2), КріптЕК-Д (демоверсія 1.1.3.42), File- PRO (збірка 2.4.0.15), 8. VipNet CryptoFile (збірка 4.0.1.43722)

Слід відзначити, що програмні комплекси КріптЕК-Д, File-PRO і VipNet CryptoFile не дають можливості переглянути статус сертифіката через власний інтерфейс. Відповідно, зробити висновки щодо коректності їх функціонування не представляється можливим. Тому з тестування вони виключені.

У підсумковій таблиці 6 приведені результати проходження тестів по NIST зазначених програм.

Таблиця 6

Підсумкові результати проходження тестів по NIST

Litoria Desktop	КриптоАРМ	Admin-PKI	КАРМА	КриптоНУЦ
Пройдено тестів				
224	160	173	166	168
Не пройдено тестів (причина – статус сертифікату КЕП визначений невірно)				
0	10	18	13	13
Не пройдено тестів (причина – результат невірний)				
0	34	13	25	23
Число нереалізований тестів				
0	20	20	20	20

Виходячи з результатів тестування беззаперечним лідером є програма Litoria Desktop, яка пройшла всі тести по NIST та може бути рекомендована до використання при створенні КЕП.

ВИСНОВКИ. Схеми формування КЕП, що базуються на асиметричних алгоритмах, є принципово уразливі. Швидкодія асиметричних алгоритмів для формування КЕП є досить низькою. З огляду на бурхливий розвиток обчислювальних потужностей сучасних комп'ютерних систем і математичних методів криптоаналізу, практична схема КЕП повинна гарантувати достатній рівень захисту на роки вперед. При використанні КЕП об'єктом захисту поряд з самим об'єктом є і його КЕП.

Серед недоліків існуючих схем формування КЕП відмічається: повільна робота алгоритмів формування і перевірки підпису; обмеження на довжину повідомлення, яке підписується. Відомі програмні рішення мають недоліки щодо довжини, розбиття повідомлення на фрагменти і підпис кожного фрагмента.

Проведено дослідження сучасних систем КЕП шляхом їх програмного моделювання. За критеріями зручності програмної реалізації і швидкодії це дозволило визначити , що вони в найбільшій мірі задовольняють сформульованим критеріям і можуть бути рекомендовані для практичного застосування.

В результаті дослідження вироблені практичні рекомендації по довжині блоків хеш-функції і довжині ключа алгоритмів КЕП. Дослідження на основі тестів, запропонованих NIST, показало недосконаліть засобів КЕП та шифрування документів.

Література

1. Закон України «Про електронні довірчі послуги». [Електронний ресурс]. – Режим доступу: https://zakon.rada.gov.ua/laws/show/2155-19#Text (дата звернення 14.05.2023).
2. Протокол Діффі-Геллмана. [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Протокол_Діффі_—_Геллмана (дата звернення 12.05.2023).
3. Інформація для кваліфікованих надавачів електронних довірчих послуг. [Електронний ресурс]. – Режим доступу: http://www.ukrstat.gov.ua/elektr_zvit/inf_akcen/centr.htm (дата звернення 24.05.2023).
4. Стандарт CCITT X.509. [Електронний ресурс]. – Режим доступу: https://proverkassl.com/docs_x.509.html (дата звернення 04.05.2023).
5. Ю. Подолюк, А. Переймибіда. Захист інформації в інтернет-застосуваннях на основі криптосистем з еліптичними кривими. [Електронний ресурс]. – Режим доступу: visnyk-ami.lnu.edu.ua (дата звернення 24.10.2023)
6. Шифр Эль-Гамаля. [Електронний ресурс]. – Режим доступу: https://it.rfei.ru/course/~k017/~V8u3Fj4l/~hIGNMjZS (дата звернення 18.05.2023).
7. Использование библиотеки OpenSSL в проектах на C++. [Електронний ресурс]. – Режим доступу: https://habr.com/ru/post/277935/ (дата звернення 22.05.2023).

Reference

1. Zakon Ukrainy «Pro elektronni dovirchi posluhy». [Elektronnyi resurs]. – Rezhym dostupu: https://zakon.rada.gov.ua/laws/show/2155-19#Text (data zvernennia 14.05.2023).
2. Protokol Diffi-Hellmana. [Elektronnyi resurs]. – Rezhym dostupu: https://uk.wikipedia.org/wiki/Protokol_Diffi_—_Hellmana (data zvernennia 12.05.2023).
3. Informatsiia dlia kvalifikovanykh nadavachiv elektronnykh dovirchykh posluh. [Elektronnyi resurs]. – Rezhym dostupu: http://www.ukrstat.gov.ua/elektr_zvit/inf_akcen/centr.htm (data zvernennia 24.05.2023).
4. Standart CCITT X.509. [Elektronnyi resurs]. – Rezhym dostupu: https://proverkassl.com/docs_x.509.html (data zvernennia 04.05.2023).
5. Iu. Podoliuk, A. Pereimybida. Zakhyst informatsii v internet-zastosuvanniakh na osnovi kryptosystem z eliptychnymy kryvymy. [Elektronnyi resurs]. – Rezhym dostupu: visnyk-ami.lnu.edu.ua (data zvernennia 24.10.2023)
6. Shyfr Эl-Hamalia. [Elektronnyi resurs]. – Rezhym dostupu: https://it.rfei.ru/course/~k017/~V8u3Fj4l/~hIGNMjZS (data zvernennia 18.05.2023).
7. Yspolzovanye byblyoteky OpenSSL v proektakh na C++. [Elektronnyi resurs]. – Rezhym dostupu: https://habr.com/ru/post/277935/ (data zvernennia 22.05.2023).

The article discusses modern algorithms for creating a qualified electronic signature (QES) using hashing functions. Algorithms were analyzed for the purpose of software implementation of electronic signature systems in the Java language using the Open SSL library. The regulatory framework for creating an electronic signature is considered and its vulnerability is noted. Among the shortcomings of the existing electronic signature formation schemes, the slow operation of the formation algorithms, signature verification and message length limitation is noted. The software solutions used have limitations on the length of the message, its division into fragments and the signature of each fragment. Such a decision is often unacceptable for use in practice, as it increases the volume of the message, the time required for the creation and verification of the electronic signature.

Among the characteristics of a cryptosystem with public keys, it is determined that the weak link in such systems are keys. It is noted that cryptographic systems developed on the basis of the RSA algorithm generate analogues in algorithms on elliptic curves. As a result of the study, practical recommendations on the length of the hash function blocks and the length of the key of the electronic signature algorithms were developed.

When developing a public key infrastructure, you should be guided by the tests of the US National Institute of Standards and Technology (NIST). A study based on NIST tests showed the imperfection of electronic signature and document encryption tools. As a result of the study, further ways of improving electronic signature algorithms aimed at reducing the number of coding operations and increasing cryptographic stability are planned. As a result of the study, practical recommendations on the length of hash function blocks and the length of the key of KEP algorithms were developed.

Keywords: *qualified electronic signature, hash function, software implementation, Open SSL libraries, algorithm, key distribution, NIST tests.*

Рязанцев О.І. – професор, завідувач кафедри комп'ютерних наук та інженерії Східноукраїнського національного університету ім. В. Даля, ryazancev@snu.edu.ua

Кардашук В.С. – доцент кафедри комп'ютерних наук та інженерії Східноукраїнського національного університету ім. В. Даля, kardashuk@snu.edu.ua , kardashuk1@gmail.com .

Кравцов С.В. – аспірант кафедри комп'ютерних наук та інженерії Східноукраїнського національного університету ім. В. Даля.