МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет кібербезпеки комп’ютерної та програмної інженерії

Кафедра безпеки інформаційних технологій

**Blockchain-технології**

**Лабораторна робота №3**

Тема: Знайомство з Хеш-функціями

Тривалість заняття: 90 хв.

Київ 2023

**Тема:** Робота з Хеш-функціями.

**Мета:** Навчитись використовувати програму хешування для перевірки цілісності даних.

**Теоретична частина**

Хеш-функція, або геш-функція — [функція](https://uk.wikipedia.org/wiki/%D0%9F%D1%96%D0%B4%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%B0), що перетворює вхідні дані будь-якого (як правило великого) розміру в дані фіксованого розміру.

Хешування (гешування, [англ.](https://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) *hashing*) — перетворення вхідного масиву даних довільної довжини у вихідний бітовий рядок фіксованої довжини. Такі перетворення також називаються хеш-функціями, або функціями згортання, а їхні результати називають хешем, хеш-кодом, хеш-сумою, або дайджестом повідомлення (англ. *message digest*).

Хеш-функція використовується зокрема у структурах даних — хеш-таблицях, широко вживаних у програмному забезпеченні для швидкого пошуку даних. Хеш-функції використовуються для оптимізації таблиць та баз даних користуючись з того, що в однакових записів однакові значення хеш-функції. Такий підхід пошуку дублікатів ефективний у файлах великого розміру. Прикладом цього буде знаходження подібних ділянок у послідовностях [ДНК](https://uk.wikipedia.org/wiki/%D0%94%D0%9D%D0%9A). Криптографічна геш-функція дозволяє легко перевірити, що деякі вхідні дані зіставляються із заданим значенням хешу, але, якщо вхідні дані невідомі, то навмисно важко відновити вхідне значення (або еквівалентну альтернативу), знаючи збережене значення хеш-функції. Це використовується для забезпечення цілісності переданих даних, і є будівельним блоком для HMACs, які забезпечують аутентифікацію повідомлень.

Хеш-функції пов'язані (і їх часто плутають) з [контрольною сумою](https://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BD%D1%82%D1%80%D0%BE%D0%BB%D1%8C%D0%BD%D0%B0_%D1%81%D1%83%D0%BC%D0%B0), контрольними цифрами, відбитками пальців, рандомізацією функцій, кодами, що виправляють помилки, і з шифрами. Хоча ці поняття певною мірою збігаються, кожне з них має свою власну сферу застосування і вимоги та є розробленим і оптимізованим по-різному.

Хешування застосовується для побудови асоціативних масивів, пошуку дублікатів в серіях наборів даних, побудови унікальних ідентифікаторів для наборів даних, контрольного підсумовування з метою виявлення випадкових або навмисних помилок при зберіганні або передачі, для зберігання паролів в системах захисту (у цьому випадку доступ до області пам'яті, де знаходяться паролі, не дозволяє відновити сам пароль), при виробленні електронного підпису (на практиці часто підписується не саме повідомлення, а його геш-образ).

У загальному випадку однозначної відповідності між вихідними даними і геш-кодом немає в силу того, що кількість значень геш-функцій менша, ніж число варіантів значень вхідного масиву. Існує безліч масивів з різним вмістом, що дають однакові геш-коди — так звані колізії. Імовірність виникнення колізій відіграє важливу роль в оцінці якості геш-функцій.

Розроблено багато алгоритмів гешування з різними властивостями (розрядність, обчислювальна складність, криптостійкість тощо). Вибір тієї чи іншої геш-функції визначається специфікою розв'язуваної задачі. Найпростішими прикладами геш-функцій можуть служити контрольна сума або CRC.

Прості хеш-функції (ненадійні, але які просто і головне швидко обчислюються) застосовують для перевірки цілістності передачі пакетів по мережевому протоколу TCP/IP та інших для виявлення апаратних помилок — так зване "надлишкове кодування". Якщо хеш отриманого пакета даних співпадає з відправленим разом з пакетом даних (так званою "контрольною сумою), то це може означати, що втрати чи помилок при передачі пакета даних по мережі не сталось. Якщо ж хеші не співпадають, то можливо при передачі пакета відбулась втрата даних, у такому разі пакет пересилається знову. У даному випадку використовується проста хеш-функція, тому що при передачі даних важлива швидкість.

Великі по складності хеш-функції використовують у криптографії. Головна умова для них — неможливість по кінцевому результату (хешу) обчислити початковий масив даних. Ще одна головна умова — стійкість до колізій, тобто низька ймовірність отримати два однакових хеша з двох різних масивів даних при обробці їх такою функцією. Розрахунки по таким алгоритмам складні і ресурсовитратні, але тут вже головне не швидкість, а надійність.

Для доступа до сайтів та серверів по логіну і паролю часто використовують хешування. Зберігати паролі у відкритому вигляді для подальшої звірки тими, що ввів користувач, досить ненадійно з точки зору можливості їх викрадення. Тому зберігають хеші усіх паролів. Користувач вводить пароль, миттєво розраховується його хеш і звіряється з тим, що є у базі даних. Надійно і дуже просто. Як правило для такого типу хешування використовують складні функції з дуже високою криптостійкістю, щоб по хешу було неможливим відновити пароль.

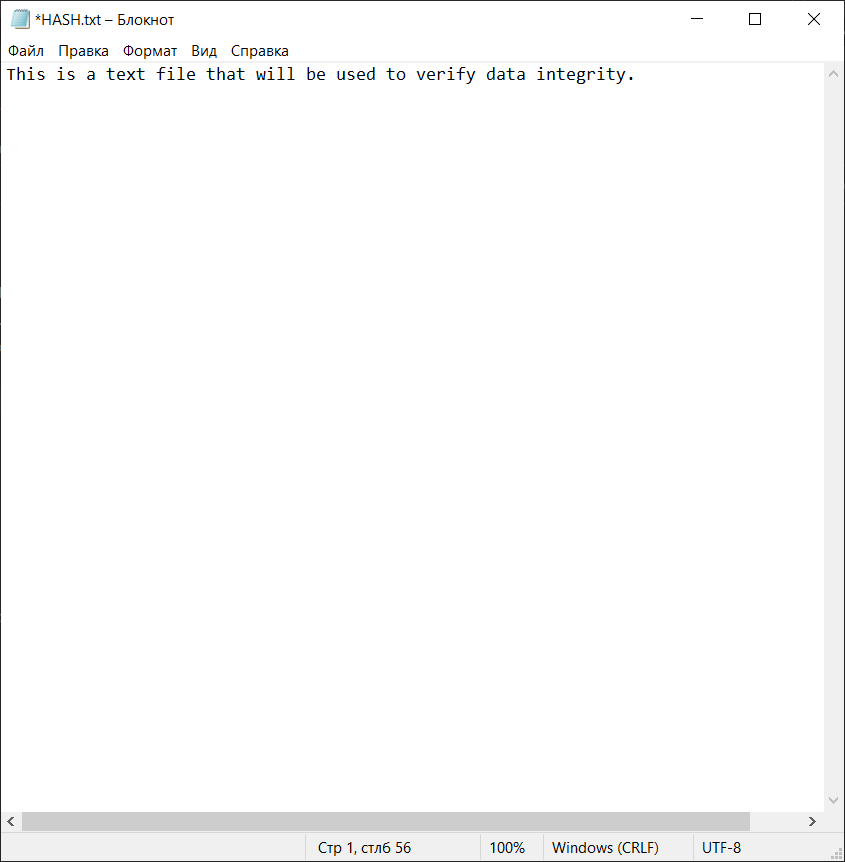
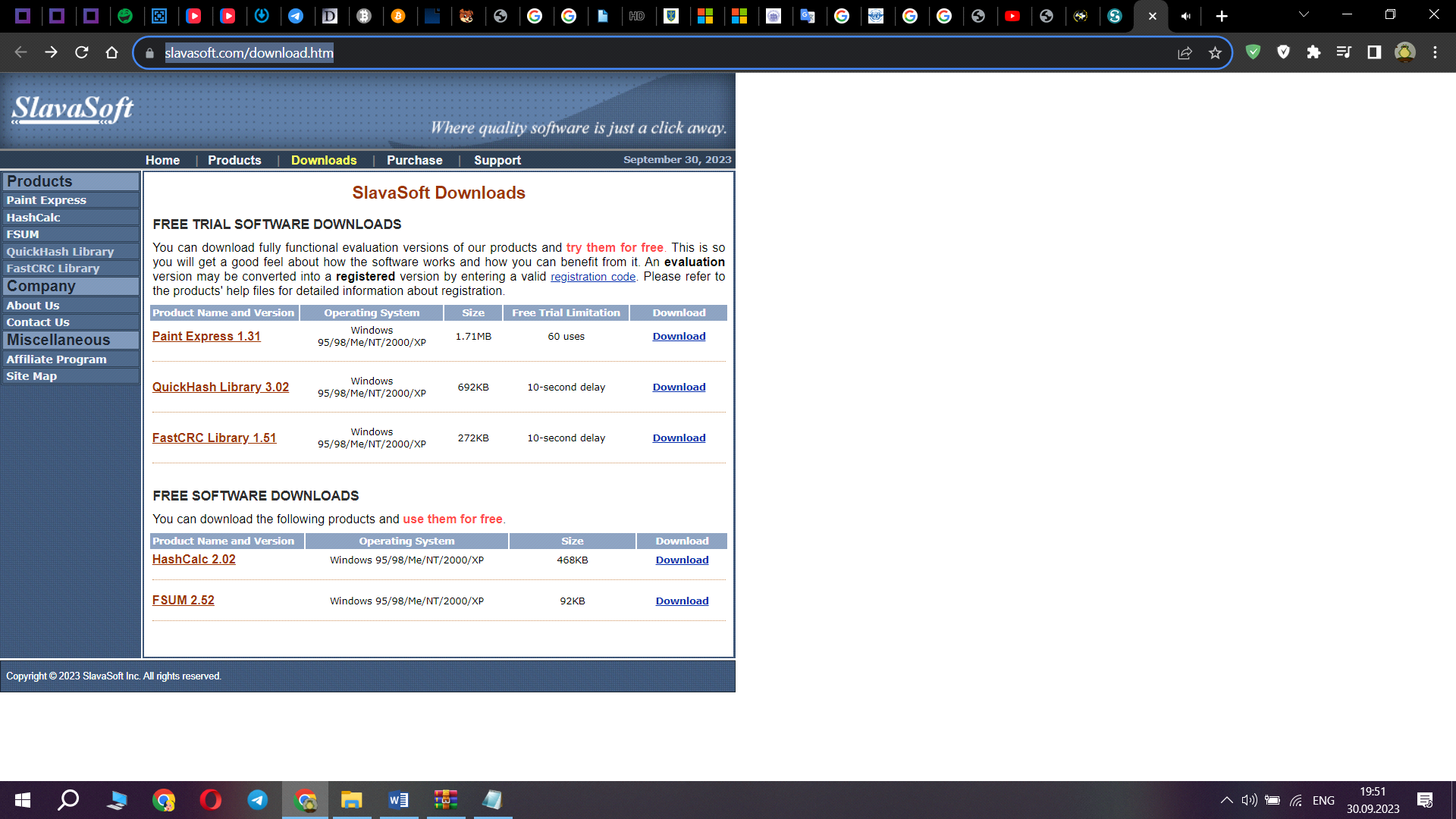
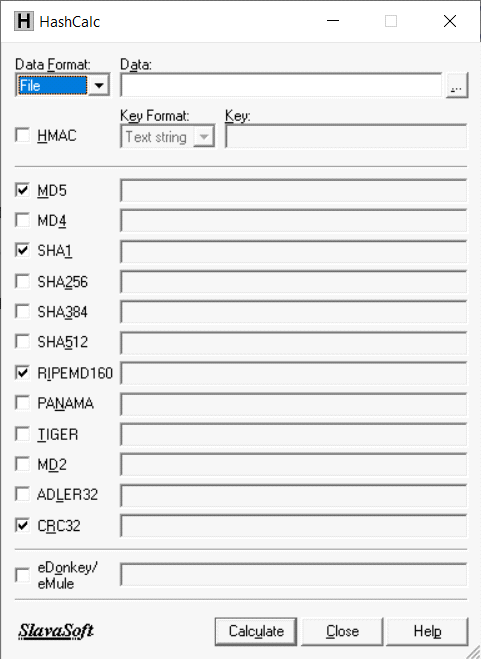
Хешування використовують в технології електроного цифрового підпису. За допомогою хеша переконуються, що, наприклад, підписують саме той документ, що потрібно. Також хеш гарантує, що документ підписано саме тою людиною, яка заявлена, тобто за допомогою можна посвідчувати особу.

Хеш-функції використовуються в технології блокчейн, де хеш є гарантією цілістності ланцюжка транзакції (платежів) і захищає її від несанкціонованих змін. Завдяки хешу і розподіленим обчислюванням зламати блокчен дуже складно, на основі блокчейна існує багато криптовалют. Сама популярна — біткоін — існує з 2009 року, і до цього часу її не було зломано.

Популярні хэш-алгоритми

* CRC32 — використовується для створення контрольних сум. Ця функція не є криптографічною. Існує багато варіацій цього алгоритма, число після "CRC" означає довжину отриманого хеша в бітах. Функція досить проста і нересурсоємна. Використовується для перевірки цілістності пакетів в різних протоколах передачі даних.
* MD5 — стара, але до цього часу дуже популярна версія криптографічного алгоритма, яка створює хеш довжиною 128 біт. Хоча стійкість цієї функції на сьогодні і не дуже висока, вона все одно часто використовується для шифрування паролів.
* SHA-1 — криптографічна функція. Дає хеш довжиною 160 байт. Зараз відбувається активна міграція в бік SHA-2 — більш стійкої хеш-функції.

**Завдання**

1. Відкрийте блокнот та введіть в блокнот таку фразу, в кінці фрази додайте своє ім’я та прізвище на англійській.
   1. 
2. Збережіть файл на робочому столі з назвою HASH.
3. Перейдіть за посиланням <https://www.slavasoft.com/download.htm> , та завантажте програму з назвою HashCalc 2.02.
   1. 
4. Відкрийте завантажений zip файл, та запустіть файл setup.exe, для встановлення програми слідуйте інструкції установника.
5. Після встановлення програми, відкрийте її.
   1. 
6. Вкажіть наступні елементи у HashCalc:

* Формат даних( Data format ): Файл ( File )
* Дані (Data) : виберіть попередньо збережений файл (Hash.txt) на робочому столі.

1. Зніміть галочки з усіх типів хешів, окрім MD5
2. Натисніть кнопку обчислити, занесіть отримані результати в звіт.
3. Прорахуйте всі хеш функції для цього тексту, отримані результати занесіть до звіту.
4. Після чого зайдіть в раніше створений файл Hash.txt внесіть зміни в текст файлу, та повторіть обчислення хешу MD5, та для всіх типів хешів, отримані зміни занесіть до звіту.
5. Дайте відповіді на контрольні питання, зробіть висновки та внесіть в звіт.

**Контрольні питання**

1. Що таке хеш-функція?
2. Для чого використовують хеш-функції?
3. Які є типи хеш-функцій?
4. Яким чином хеш-функції використовуються в блокчейні?
5. Які є популярні хеш-алгоритми?
6. Яким чином обчислюється хеш-функція?
7. Чому різні хеш-функції створюють хеші різної довжини?