НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

імені Ігоря Сікорського»

Факультет прикладної математики

Кафедра прикладної математики

Звіт

з лабораторної роботи № 6

із дисципліни «Криптографічні методи захисту інформації»

на тему

*Хеш-функції та їх застосування в електронних цифрових підписах*

|  |  |
| --- | --- |
| Виконав: | Керівник: |
| студент групи КМ-01 | *ст. викладач Бай Ю. П.* |
| *Дюбакін Р. С.* |  |

Київ — 2022

ЗМІСТ

[Постановка завдань і хід виконання лабораторної роботи 3](#_Toc58564390)

[Відповіді 4](#_Toc58564391)

[Список літератури 5](#_Toc58564392)

[Додаток 1 6](#_Toc58564393)

***Мета роботи:*** засвоїти *поняття хеш-функції, властивості хеш-функцій, електронний цифровий підпис, алгоритми створення та перевірки ЕЦП*. Отримати практичні навички роботи з найбільш використовуваними на практиці хеш-функціями, отримати практичний досвід створення електронного цифрового підпису з використанням алгоритму RSA, опанувати бібліотеку *rsa* для Python.

# Постановка завдань і хід виконання лабораторної роботи

1. Дати визначення хеш-функцій і навести властивості ідеальної хеш-функції. Дати визначення колізії хеш-функції.

2. Обчислити значення хеш-функцій від заданих вхідних даних, заповнити *Таблицю 1*.

3. Створити файл *zayava.docx* від власного імені, записати Прізвище Ім’я По-батькові. Обчислити *sha1* від вмісту даного файлу, записати в *Таблицю 1.*

4. Навести теоретичні відомості з електронного цифрового підпису. Навести схему створення та верифікації ЕЦП за допомогою асиметричних алгоритмів. Створити електронний цифровий підпис до власного файлу *zayava.docx,* використовуючи алгоритм RSA та бібліотеку *rsa* для Python:

4.1. Встановити бібліотеку *rsa*:

python -m pip install rsa

pip install --upgrade rsa

4.2. Згенерувати секретний (private) і відкритий (public) ключі для алгоритму RSA – для цього необхідно один раз виконати файл ***genkeys.py.***

4.3. Записати секретний і відкритий ключі(згенеровані ключі знаходяться у файлах *private\_key\_1024.pem, public\_key\_1024.pem*) ***display\_key.py***

4.4. Приєднати **метадані** до хеш-коду від вхідного файлу. Зашифрувати одержаний результат, користуючись функцією *rsa.encrypt().* Налаштувати і запустити на виконання файл ***hash\_sign.py****.* Цифровий підпис буде записано в файл ***zayava.docx.sgn****.* Заповнити *Таблицю 2.*

4.5. Передати одержувачу документ *zayava.docx,*  цифровий підпис до нього − *zayava.docx.sgn* тапублічний ключ.

5. Імітувати дії одержувача, верифікувати цифровий підпис (див. файл ***verify\_signature.py***).

6. Створити і верифікувати цифровий підпис до документу *zayava.docx* **без метаданих**, користуючись вбудованими функціями бібліотеки rsa: *rsa.sign(), rsa.verify().* Якщо ЕЦП до файлу вірний, то функція *rsa.verify()* повертає тип хеш-функції, яка використовувалась при хешуванні вхідного файлу.

# Відповіді

1.

Криптографічна хеш-функція (“hash” – мішанина) – це функція (в загальному випадку – алгоритм), що перетворює вхідні дані будь-якого розміру в дані фіксованого розміру.

Найбільш поширені хеш-функції:

MD4, MD5

SHA1, SHA256, SHA512, SHA3-256, SHA3-512

RIPEMD – 160

Функція хешування Н(x) повинна мати такі властивості:

1. Бути застосовною до блоку даних будь-якої довжини.
2. Давати на виході значення фіксованої довжини.
3. Значення Н(x) має обчислюватися відносно легко для будь-якого заданого х, а алгоритм обчислення має бути практичним з погляду як апаратної, так і програмної реалізації.
4. Односторонність (незворотність). Для будь-якого даного коду h повинно бути практично неможливо вирахувати х, для якого Н(x) = h.
5. Стійкість до колізій першого роду. Для будь-якого заданого блоку х має бути практично неможливо обчислити у ≠ х, для якого Н(x) = Н(у).
6. Стійкість до колізій другого роду. Має бути обчислювально неможливо підібрати пару повідомлень х , у, для яких Н(x) = Н(у).
7. Лавиновий ефект.

Колізією хеш-функції H називаються два різних вхідних блоки даних x і y таких, що H(x)=H(y).

Колізії існують для більшості хеш-функцій, але для «хороших» хеш-функцій частота їх виникнення близька до теоретичного мінімуму. В деяких окремих випадках, коли множина різних вхідних даних є скінченною, можна задати ін'єктивну хеш-функцію, за визначенням без колізій. Однак, для хеш-функцій, які приймають вхідні дані змінної довжини і повертають хеш постійної довжини (таких як MD5), колізії обов'язково будуть існувати, оскільки хоча б для одного значення хеш-функції відповідна йому вхідна множина значень буде безкінечною — і будь-які два значення з цієї множини утворюють колізію.

2. Визначення дайджестів (хеш-кодів) від рядків та вмісту файлу.

*Таблиця 1*

|  |  |  |  |
| --- | --- | --- | --- |
| № | Вхідні дані, *M* | *hash* | Результат,  *h(M)* |
| 1 | *‘Hello, world!’* | sha1 | 943a702d06f34599aee1f8da8ef9f7296031d699 |
| 2 | *‘12345’* | sha1 | 8cb2237d0679ca88db6464eac60da96345513964 |
| 3 | *‘12346’* | sha1 | 94ae0a96d83a445d72a93417b63ac90d79db5eca |
| 4 | *‘’* | sha1 | da39a3ee5e6b4b0d3255bfef95601890afd80709 |
| 5 | *‘Дюбакін Р.С.’ (власні ПІБ)* | md5 | 1bdf518c2e1bf3962f18e6323eed5beb |
| 6 | *‘Дюбакін Р.С.’* | sha1 | 2e3f4e380f97f7e67267127e9024c4e786e3655d |
| 7 | *‘Дюбакін Р.С.’* | sha224 | 01bb124577e103cf42aeab2e20978d30542742a477fcfd27baa66436 |
| 8 | *‘Дюбакін Р.С.’* | sha256 | 0891744866e9aac8e9683e23452ad6f6903f596427ae7ebfdac8af3213ab94a4 |
| 9 | *‘Дюбакін Р.С.’* | sha512 | 275a81bc8f42e2651b0c8eeb893848f3120d724cf4eb51d8b791837da847983e84d6dcfd7661115fdf5ecb8d2b6e113ea7a43e617aa8629bbba7be553485396e |
| 10 | *zayava.docx* | sha1 | 872c7c6d8254a5382506d80828190334d31e489e |

3. Див. рядок 10 в *Таблиці 1*.

4. Основні відомості з електронного цифрового підпису.

ЕЦП – це дані в електронній формі, отримані за результатами криптографічного перетворення, які додаються до інших даних або документів і забезпечують їх цілісність та ідентифікацію автора.

Електронно-цифровий підпис (ЕЦП) ставиться під електронним документом з тією ж метою, як і звичайний ручний підпис під паперовим

документом:

* Для перевірки, що електронний документ створено саме особою, яка поставила під ним свій ЕЦП
* Для гарантії того, що електронний документ не змінювався після його підписання

При цьому цифровий підпис має таку ж юридичну силу, як і звичайний.

**Закритий ключ** – рядок символів, згенерований для конкретного користувача (унікальний рядок). Зберігається у користувач.

**Відкритий ключ** – рядок символів, згенерований для конкретної людини (унікальний рядок), і пов'язаний із закритим ключем математичною залежністю. Зберігається в базі даних. Відкритий ключ працює тільки в парі із закритим ключем. На відкритий ключ видається сертифікат, який автоматично передається разом із листом, підписаним ЕЦП.

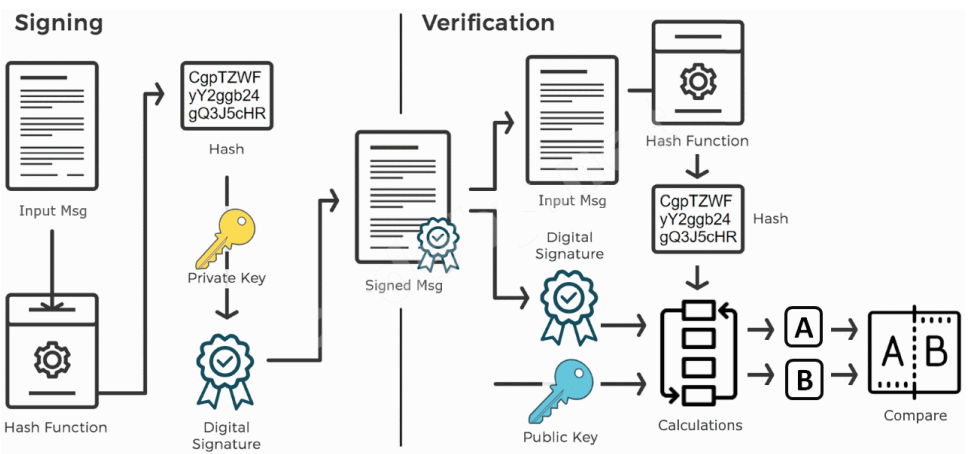
Результат перевірки підпису – твердження ТАК / НІ, отримане в результаті криптоперетворення з файлу з ЕЦП та відкритого ключа.

Схема створення ЕЦП з використанням асиметричних алгоритмів.

1. Генерація ключів
2. Створення підпису
3. Перевірка (верифікація) підпису

**Генерація ключів**

Ключі надаються ПІДПИСУВАЧУ (автору документа) кваліфікованим надавачом електронних довірчих послуг – акредитованим Центром сертифікації ключів (див. https://czo.gov.ua/ca-registry)

**Створення ЕЦП**

ВІДПРАВНИК (автор документа):

1. Обчислює хеш-код документа.
2. З хеш-коду та інших метаданих (дата підписання, ім'я підписанта та ін.) формує відкритий текст М.
3. Шифрує текст М закритим ключем (private key) і одержує підпис.
4. Передає одержувачу відкритий ключ, документ та підпис:

RSA: (e, n) (M, S(h(M))

**Перевірка ЕЦП**

ОДЕРЖУВАЧ:

1. Розшифровує підпис публічним ключем (public key) відправника та отримує відкритий текст M — хеш-код та метадані.
2. Обчислює хеш-код документа.
3. Порівнює хеш-коди з пунктів 5 і 6. Якщо вони рівні, то підпис вірний.
4. Перевіряє метадані

4.3. Генерація ключів.

а) ключі довжиною 16 біт

*Private Key: n =38911 ; e =65537 ; d =32209 ; p =233 ; q =167 .*

*Public Key: n =38911 ; e =65537 .*

б) ключі довжиною 1024 біт

*Private Key: n =169520929194790473459417687057537759676312674683232915262613609724585581969198603417971208980218215345326372755215682352334954969058221466693065561444012135740805509293909433738884517952867839752669955980007768300630295550537357910719384188754759171829896423174967885610689735070209687194558693261120315655583 ; e =65537 ; d =22695219994126853748766815481984776590322526323613921884037594209736696769729291032382919382828549085858272343168933533109341210286049638353372243899282250906079169180587952345049641142025239830652891517581241613728566003860843520736033555054951637913898885300912886687506101777381963917593512879163733476513 ; p =55582687668265114771050473923396097088344430959826396122851815969836791284730326763123656875336765250375081486350886781757558987134693360008167569891647076517272553 ; q =3049887227594039044433498637074496017834400844322178533250683799503443015907193934885879571316194508460529581858163004024904886447663536207677511 .*

*Public Key: n =169520929194790473459417687057537759676312674683232915262613609724585581969198603417971208980218215345326372755215682352334954969058221466693065561444012135740805509293909433738884517952867839752669955980007768300630295550537357910719384188754759171829896423174967885610689735070209687194558693261120315655583 ; e =65537 .*

4.4. Метадані до файлу *zayava.docx* та цифровий підпис:

*Таблиця 2*

|  |  |  |
| --- | --- | --- |
| 1 | Ім’я файлу | zayava.docx |
| 2 | Розмір файлу | 12980 |
| 3 | Дата підписання | 2022-06-22 |
| 4 | Хеш-код вмісту файлу | 872c7c6d8254a5382506d80828190334d31e489e |
| 5 | Цифровий підпис | b"\xaf\xd1psh\xe7\xdb\xf2\xb8\xd5\xa3\xe7P\xc1\x089\xd4R\x83\xbc\xac\x80h8eg\xa35@\xa9\xa2\xa8\xee\xca>\xfe\x99uao\xc1\x8ea\xe8\t\x1a\xce\xc0s'\xc1\xe8$a\x1b\xdd$\xdb\x8cY\xb5M@\*\xd4\_\xb1\xb6;\xe5\x1b)\_\xb8\xfb\xf7\x0e]=\xb3\xb4\xfc\x97k\xe4c\x99\xfbm\x83\xa2\x15^\xdey\x9f\xcd\x9c\xbb\x0e\xd5F\xbeUz\xc2\xabj?i\xb7\xbf\xdb^\xe7F?\xb9\xcaI\xdd@\xa0m\xc4\xe2\xdf\xea" |

Зображення, що містить текст

Автоматично згенерований опис5. Скріншот перевірки цифрового підпису (дії одержувача)

# Список літератури

1. Тарнавський Ю.А. Технології захисту інформації [Електронний ресурс] / Ю. А. Тарнавський. – Київ: КПІ ім. Ігоря Сікорського, 2018. – 162 с.
2. Шнайер Б. Прикладная криптография: Протоколы, алгоритмы, исходные тексты на языке Си / Б. Шнайер. – М.: Диалектика, 2003. – 610 с.
3. Алферов А.П., Зубов А.Ю., Кузьмин А.С., Черемушкин А.В. Основы криптографии. – М.: Гелиос АРВ, 2001. – 480 с.
4. Столлингс В. Криптография и защита сетей: принципы и практика, 2-е изд.: Пер. с англ. – М.: «Вильямс», 2001. – 672 с.
5. Menezes A.J., Van Oorschot P.C., Vanstone S.A. Handbook of Applied Cryptography. – CRC Press, Inc., 1997. – 795 p.

# Додаток 1

**Скріншоти виконання файлів**

Результат виконання файлу hash.py:

Зображення, що містить текст

Автоматично згенерований опис

Результат виконання файлу hash\_sign.py:

Зображення, що містить текст

Автоматично згенерований опис

Результат перевірки підпису:

Зображення, що містить текст

Автоматично згенерований опис