НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. Ігоря СІКОРСЬКОГО»
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ

Звіт з виконання комп'ютерного практикума

ВИБІР ТА РЕАЛІЗАЦІЯ БАЗОВИХ ФРЕЙМВОРКІВ ТА БІБЛІОТЕК

Виконали студентки
групи ФІ-32мн
Зацаренко А. Ю.
Футурська О. В.

Перевірила:
Селюх П. В.

Київ — 2023

ЗВІТ

1.1 Мета роботи

Вибір базових бібліотек/сервісів для подальшої реалізації криптосистеми.

1.2 Завдання на лабораторну роботу

Вибір бібліотеки розробки гібридної криптосистеми з точки зору їх ефективності за часом та пам'яттю для різних програмних платформ. Оформлення результатів роботи.

Опис функції бібліотеки реалізації основних криптографічних примітивів обраної бібліотеки, з описом алгоритму, вхідних та вихідних даних, кодів повернення. Контрольний приклад роботи з функціями. Обґрунтування вибору бібліотеки.

1.3 Необхідні теоретичні відомості

Бібліотека реалізації основних криптографічних примітивів – це набір програмних інструментів або функцій, завдяки яким стають можливі реалізації основних криптографічних алгоритмів. Ці бібліотеки надають можливість використовувати криптографічні функції: шифрування, розшифрування, геш-функції, підписи, генерація ключів, тощо, без необхідності реалізації алгоритмів з самого початку.

Варто зазначити, що також бібліотеки мають підтримувати різні режими роботи, так як деякі криптографічні операції, такі як шифрування, можуть бути виконані в різних режимах, таких як ECB (Electronic Codebook) або ж CBC (Cipher Block Chaining).

Зазвичай бібліотеки реалізовані з урахуванням найсучасніших стандартів безпеки та випробувані в різних умовах використання, що робить їх надзвичайно корисними та відносно безпечними.

Добре відомі приклади бібліотек реалізації основних криптографічних

примітивів включають OpenSSL, Bouncy Castle (для Java та .NET), PyCryptodome (Python), і багато інших, які надають надійні та ефективні інструменти для захисту інформації.

У даному комп'ютерному практикумі було обрано дві наступні бібліотеки: PyCryptodome та M2Crypto.

1.3.1 PyCryptodome

PyCryptodome - це бібліотека Python, яка надає криптографічні функції та алгоритми, включаючи шифрування, дешифрування, цифрові підписи, хешування тощо. Це автономний пакет низькорівневих криптографічних примітивів для Python, який замінює стару бібліотеку PyCrypto і пропонує більш сучасну та активно підтримувану версію криптографічних інструментів.

Ключові особливості PyCryptodome включають:

1) Пропонується простий у використанні інтерфейс, що дозволяє впроваджувати криптографічні функції;

2) Є програмним забезпеченням з відкритим вихідним кодом, що дозволяє розробникам використовувати та налаштовувати його згідно умов ліцензії;

3) Активно розробляється і оновлюється, що забезпечує сумісність з останніми версіями Python і стандартами безпеки. Наразі вона сумісна з Python 2 та Python 3;

4) Є універсальною і може використовуватися для різних задач, таких як шифрування даних, цифровий підпис, хешування паролів і безпечні протоколи зв'язку.

Враховуючи всі переваги, PyCryptodome - це потужна і гнучка бібліотека Python для реалізації криптографічних функцій в додатках, забезпечення безпеки і цілісності даних.

1.3.2 M2Crypto

M2Crypto - це бібліотека Python, яка дозвOliає програмам на Python викликати функції OpenSSL для виконання різних криптографічних операцій, таких як шифрування, дешифрування, цифровий підпис і хешування, а також для SSL/TLS-зв'язку.

Ключові особливості PyCryptodome включають:

1) Забезпечує безпечний зв'язок через протоколи SSL/TLS. Це дозвOliає програмам на Python створювати SSL/TLS-з'єднання, керувати сертифікатами та виконувати безпечну передачу даних;

2) Надає широкий спектр криптографічних операцій, включаючи симетричне та асиметричне шифрування, дешифрування, цифрові підписи, дайджести повідомлень та генерацію ключів. Він підтримує такі популярні алгоритми, як AES, RSA, DSA та SHA;

3) Програмне забезпечення з відкритим вихідним кодом, що дозвOliає розробникам отримувати доступ до його вихідного коду, змінювати і поширювати його відповідно до умов ліцензії;

4) Інтегрується з OpenSSL, широко використовуваною криптографічною бібліотекою з відкритим вихідним кодом.

Щодо останнього, то M2Crypto забезпечує прив'язку Python до функцій OpenSSL, що дозвOliає безперешкодно інтегрувати можливості OpenSSL в додатки на Python. Сама бібліотека надає високорівневий інтерфейс для загальних криптографічних завдань, однак M2Crypto також дозвOliає розробникам отримати доступ до низькорівневих функцій OpenSSL, забезпечуючи гнучкість і тонкий контроль над криптографічними операціями.

Утотожнюючи, можна сказати, що дана бібліотека корисна тим, хто бажає використовувати мову програмування Python для своїх цілей. Окрім цього, доступ до OpenSSL, бібліотеки, яка широко відома своїми стандартами безпеки та криптографії, забезпечує надійність та багатофункціональність.

1.4 Побудована гібридна криптосистема

Нами було реалізовано гібридну криптосистему на основі двох схем:

– AES (Advanced Encryption Standard) – алгоритм блочного шифрування з розміром блоку 128 бітів та ключем довжиною теж 128 бітів;

– RSA – криптографічний алгоритм з відкритим ключем, що базується на обчислювальній складності задачі факторизації великих цілих чисел.

Кроки алгоритму:

1) Генерація ключів RSA користувачем Olia;

2) Публікація публічного ключа по відкритому каналу;

3) Створення сеансового ключа користувачем Anastasia та шифрування самого повідомлення цим ключем за допомогою AES;

4) Шифрування сеансового ключа публічним ключем RSA користувача Olia;

5) Надсилання зашифрованого повідомлення і відповідного зашифрованого ключа відкритим каналом користувачу Olia;

6) Розшифрування користувачем Olia сеансового ключа через свій приватний ключ;

7) Отримання вихідного повідомлення.

1.5 Отримані результати

Порівняльний аналіз двох бібліотек за обсягом часу та пам'яті наведено в таблиці 1.1.

Алгоритм / Показники	PyCryptodome	M2Crypto
Ефективність за часом, с	3.78	9.76
Ефективність за пам'яттю, ГБ	0.9	1

Таблиця 1.1 – Порівняльна таблиця

M2Crypto займає більше оперативної пам'яті, оскільки по відкритому каналу ми також передаємо початковий стан та реалізовуємо фінальне перетворення.

Також недоліком M2Crypto є те, що ініціалізація початковго стану іде власноруч, а не автоматично, як у випадку PyCryptodome.

В цілому, інтерфейс бібліотеки PyCryptodome є більш зручним у використані: функції є інтуїтивно зрозумілими і користувачу простіше орієнтуватися, оскільки більшість потрібних функцій реалізовано автоматично всередині інших методів. PyCryptodome має докладну документацію та активну спільноту користувачів і розробників, що полегшує навчання і вирішення проблем. M2Crypto може мати менш розвинену документацію та меншу спільноту.

1.6 Програмна реалізація

Було реалізовано описану вище гібридну криптосистему за допомогою бібліотек PyCryptodome та M2Crypto. У ролі відкритого каналу виступила папка на Google Drive. Приклад використання моделі цієї криптосистеми користувачами Anastasia та Olia наведено нище.

Для PyCryptodome:

1) **Користувач Olia:** https://colab.research.google.com/drive/1YdT9sst0qa9RQDc75gbjo_QjyEmfR9_a?usp=share_link

2) **Користувач Anastasia:** https://colab.research.google.com/drive/1wc69q20sG2myo1a1Xt6lDC6X_2kAKAm1?usp=share_link

Для M2Crypto:

1) **Користувач Olia:** https://colab.research.google.com/drive/1ZSrdTWWLP_odFJr1Na7xbTPgCUTV-JJI?usp=share_link

2) **Користувач Anastasia:** https://colab.research.google.com/drive/1shv6nohGZX6gAspT612iKl2I0Zrlce3M?usp=share_link

ВИСНОВКИ

У даній лабораторній роботі було розглянуто дві бібліотеки реалізації основних криптографічних примітивів: PyCryptodome та M2Crypto. З допомогою них було реалізовано гібридну криптосистему на основі симетричного алгоритму AES та схеми з відкритим ключем RSA.

Базуючись на отриманих результатах, можна стверджувати, що бібліотека PyCryptodome є більш зручним та ефективним засобом для реалізацій криптографічних примітивів на мові програмування Python в середовищі Google Collaboratory.