ЗВІТ

«Використання бібліотеки rsa у Python для асиметричного шифрування повідомлень в чаті»

1. Які конкретні задачі планували вирішувати за допомогою цієї бібліотеки?

Задача була забезпечення безпеки комунікації в чаті шляхом шифрування та розшифрування повідомлень.

2. Чому було обрано саме цю бібліотеку, а не аналоги?

Обираючи бібліотеку RSA, я враховувала високий рівень безпеки, який надає цей алгоритм та простоту використання бібліотеки. Алгоритм RSA є одним із найбезпечніших методів асиметричного шифрування в сучасному світі криптографії. Його безпека базується на складності факторизації великих простих чисел, що ускладнює взлом алгоритму.

3. Наскільки просто та зрозуміло було отримати, встановити, налаштувати та почати використовувати цю бібліотеку?

Встановлення бібліотеки було простим, вона доступна через pip. Налаштування не вимагало особливих дій, оскільки бібліотека має зручний інтерфейс.

4. Наскільки зрозумілою та корисною була документація бібліотеки?

Документація бібліотеки була зрозумілою та докладною. Надається опис основних функцій та приклади використання. Деякі приклади навіть були використані в коді.

5. Наскільки було зрозуміло, як саме використовувати бібліотеку, які класи/методи/функції використовувати для розв'язання поставлених задач?

Використання бібліотеки було досить простим, тому як все докладно описано в документації. Інтерфейс класів та методів був логічним і добре задокументованим.

6. Наскільки зручно було використовувати бібліотеку, чи не треба було писати багато надлишкового коду?

Бібліотека дозволила зручно шифрувати та розшифровувати повідомлення з мінімальною кількістю коду.

7. Наскільки зрозумілою була поведінка класів/методів/функцій з бібліотеки?

Поведінка бібліотеки була передбачуваною і відповідала очікуванням. Методи та класи працювали так, як описано в документації.

8. Наскільки зрозумілою була взаємодія між різними класами/методами/функціями цієї бібліотеки, а також взаємодія між бібліотекою та власним кодом?

Бібліотека надала можливість зручної інтеграції шифрування/розшифрування в мій вже написаний код без особливих поправок.

9. Чи виникали якісь проблеми з використанням бібліотеки? Чи вдалось їх вирішити, як саме?

Власне з самою бібліотекою проблем не було, єдине що проблемно було зрозуміти, чи потрібно спочатку кодувати повідомлення, а потім вже шифрувати за допомогою наданих бібліотекою функцій, та навпаки, коли отримуєш повідомлення і розшифровуєш його, чи потрібно потім знову розкодувати. Також проблемно було зрозуміти, як обмінятися ключами між сервером та клієнтами, як їх приймати, і в якому форматі.

10. Що хорошого можна сказати про цю бібліотеку, які були позитивні аспекти використання бібліотеки?

Простота використання, зрозуміла документація, ефективність у розв'язанні поставленої задачі.

11. Що поганого можна сказати про цю бібліотеку, які були негативні аспекти використання бібліотеки?

Найбільшою проблемою було і є обмеження на розмір повідомлення, яке можна зашифрувати, тому як мій ключ розміром 1024 біт, то максимальним розміром повідомлення є 117 байтів, що не дуже доречно, якщо в чаті відправляються довгі повідомлення.

12. Якби довелось розв'язати аналогічну задачу, але вже враховуючи досвід використання в цій лабораторній роботі, що варто було б робити так само, а що змінити? Можливо, використати інші бібліотеки, чи використати інші можливості цієї бібліотеки, чи інакше організувати код, чи ще щось?

Загалом ця бібліотека ефективно розв'язує задачу забезпечення безпеки комунікації в чаті. Але, скоріш за все, я б використала інший алгоритм асиметричного шифрування, який не накладає жорстких обмежень на розмір повідомлення, яке можна зашифрувати за один раз.

**Корисні посилання:**

[**https://www.section.io/engineering-education/rsa-encryption-and-decryption-in-python/**](https://www.section.io/engineering-education/rsa-encryption-and-decryption-in-python/)

[**https://stuvel.eu/python-rsa-doc/**](https://stuvel.eu/python-rsa-doc/)