**КРИПТОГРАФІЯ**

**КОМП’ЮТЕРНИЙ ПРАКТИКУМ №4**

**Вивчення криптосистеми RSA та алгоритму електронного підпису; ознайомлення з методами генерації параметрів для асиметричних криптосистем**

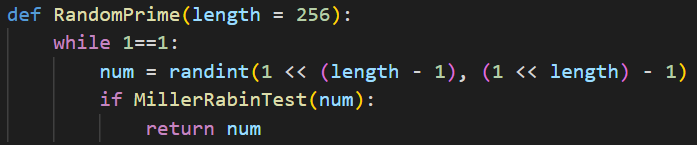
**Мета та основні завдання роботи:** Ознайомлення з тестами перевірки чисел на простоту і методами генерації ключів для асиметричної криптосистеми типу RSA; практичне ознайомлення з системою захисту інформації на основі криптосхеми RSA, організація з використанням цієї системи засекреченого зв'язку й електронного підпису, вивчення протоколу розсилання ключів.

**Порядок виконання роботи:**

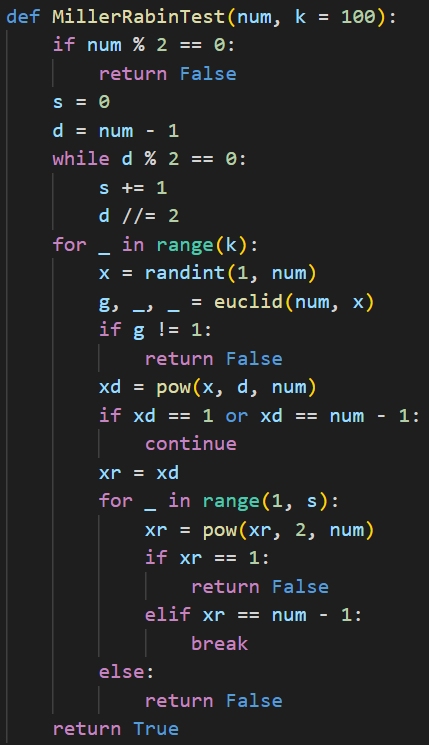
1. Написати функцію пошуку випадкового простого числа з заданого інтервалу або заданої довжини, використовуючи датчик випадкових чисел та тести перевірки на простоту. В якості датчика випадкових чисел використовуйте вбудований генератор псевдовипадкових чисел вашої мови програмування. В якості тесту перевірки на простоту рекомендовано використовувати тест Міллера-Рабіна із попередніми пробними діленнями. Тести необхідно реалізовувати власноруч, використання готових реалізацій тестів не дозволяється.
2. За допомогою цієї функції згенерувати дві пари простих чисел p, q і p1, q1 довжини щонайменше 256 біт. При цьому пари чисел беруться так, щоб pq < p1q1; p і q – прості числа для побудови ключів абонента А, p1 і q1 – абонента B.
3. Написати функцію генерації ключових пар для RSA. Після генерування функція повинна повертати та/або зберігати секретний ключ (d, p, q) та відкритий ключ (n, e) . За допомогою цієї функції побудувати схеми RSA для абонентів А і B – тобто, створити та зберегти для подальшого використання відкриті ключі (e, n), (e1, n1) та секретні d і d1.
4. Написати програму шифрування, розшифрування і створення повідомлення з цифровим підписом для абонентів А і B. Кожна з операцій (шифрування, розшифрування, створення цифрового підпису, перевірка цифрового підпису) повинна бути реалізована окремою процедурою, на вхід до якої повинні подаватись лише ті ключові дані, які необхідні для її виконання. За допомогою датчика випадкових чисел вибрати відкрите повідомлення M і знайти криптограму для абонентів А и B, перевірити правильність розшифрування. Скласти для А і B повідомлення з цифровим підписом і перевірити його.
5. За допомогою раніше написаних на попередніх етапах програм організувати роботу протоколу конфіденційного розсилання ключів з підтвердженням справжності по відкритому каналу за допомогою алгоритму RSA. Протоколи роботи кожного учасника (відправника та приймаючого) повинні бути реалізовані у вигляді окремих процедур, на вхід до яких повинні подаватись лише ті ключові дані, які необхідні для виконання. Перевірити роботу програм для випадково обраного ключа 0 < k < n.

**Хід роботи:**

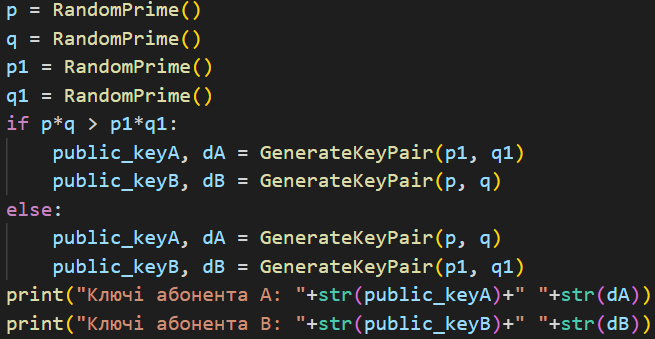
1. Напишемо функцію пошуку випадкового простого числа заданої довжини:

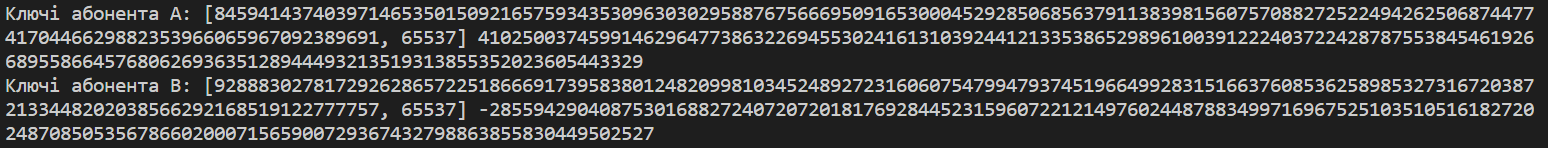


Для перевірки на простоту будемо використовувати тест Міллера-Рабіна:

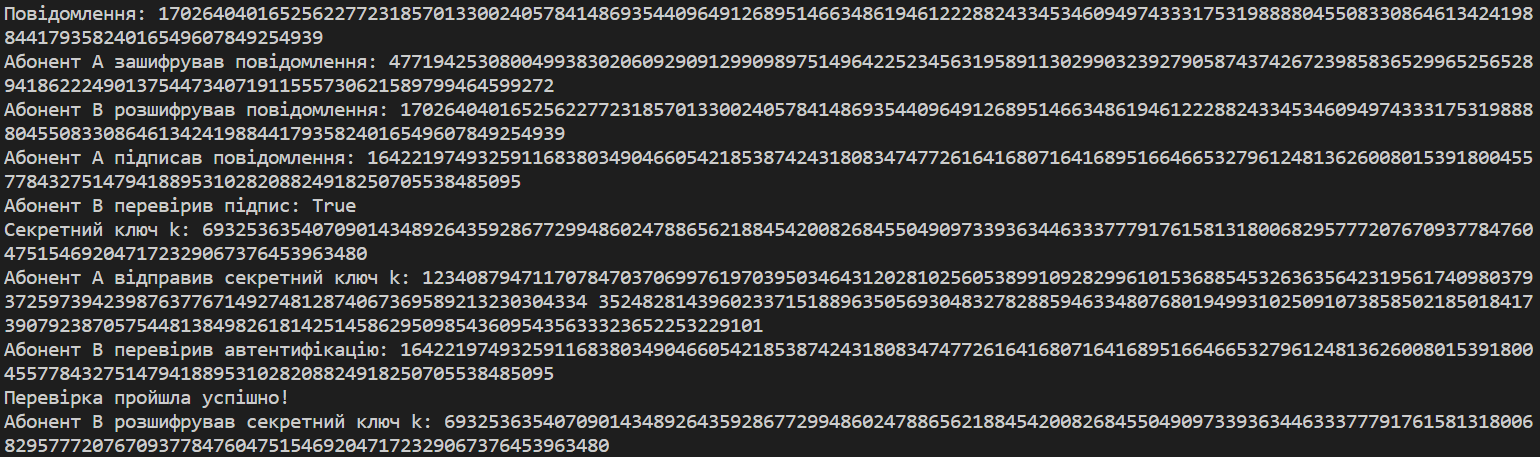


3-5. Створимо пару ключів для абонентів А та B, так щоб pq < p1q1:





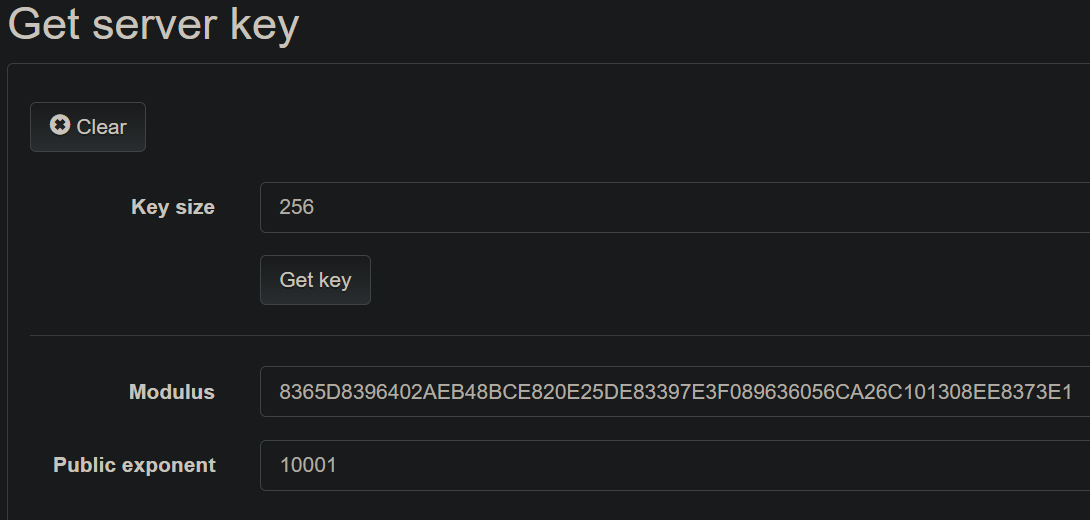
Зашифруємо, розшифруємо, підпишемо та перевіримо підпис. Також перевіримо протокол конфіденційного розсилання ключів.

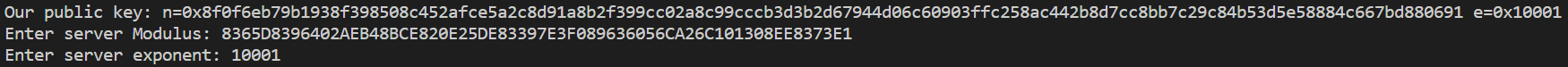


Все працює як потрібно.

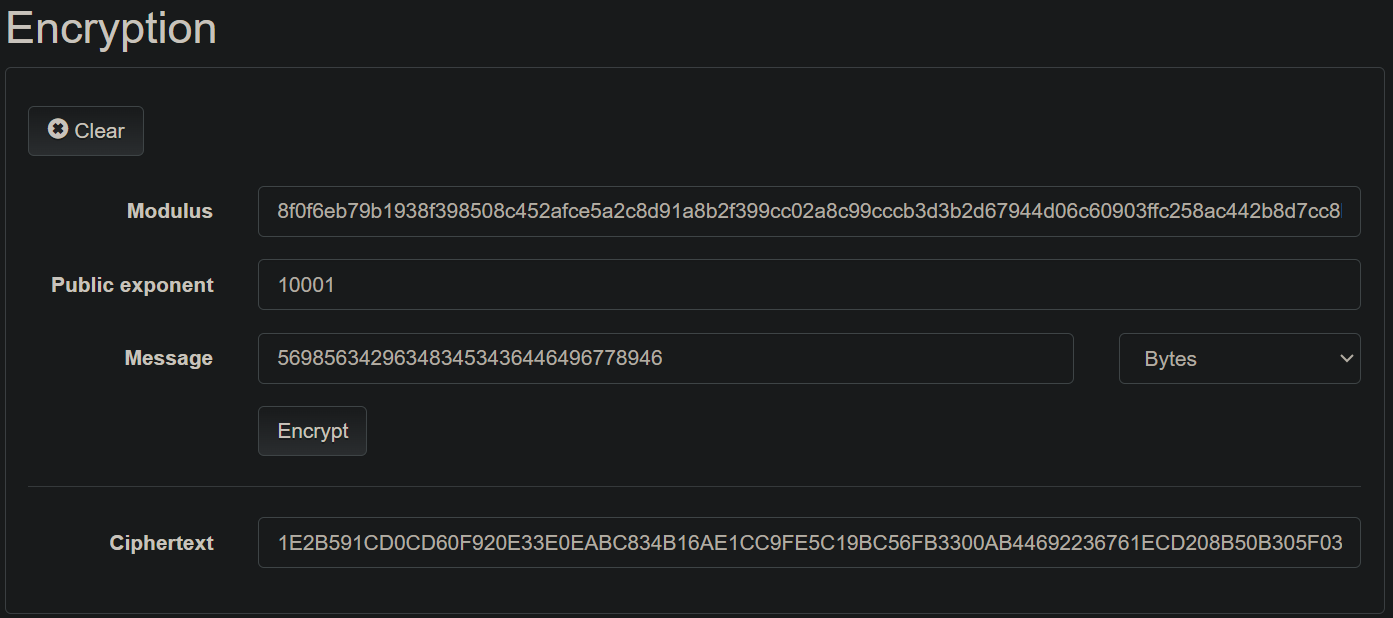
**Тестування з сервером**

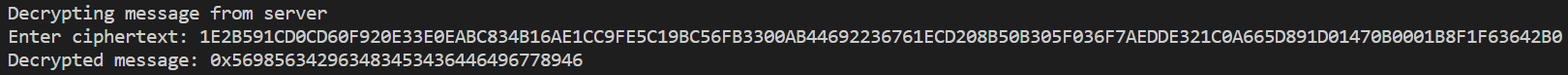
Сгенеруємо ключі для сервера та для себе:



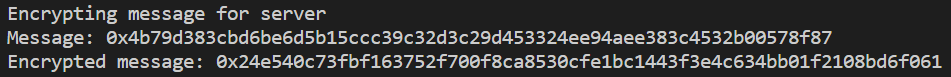


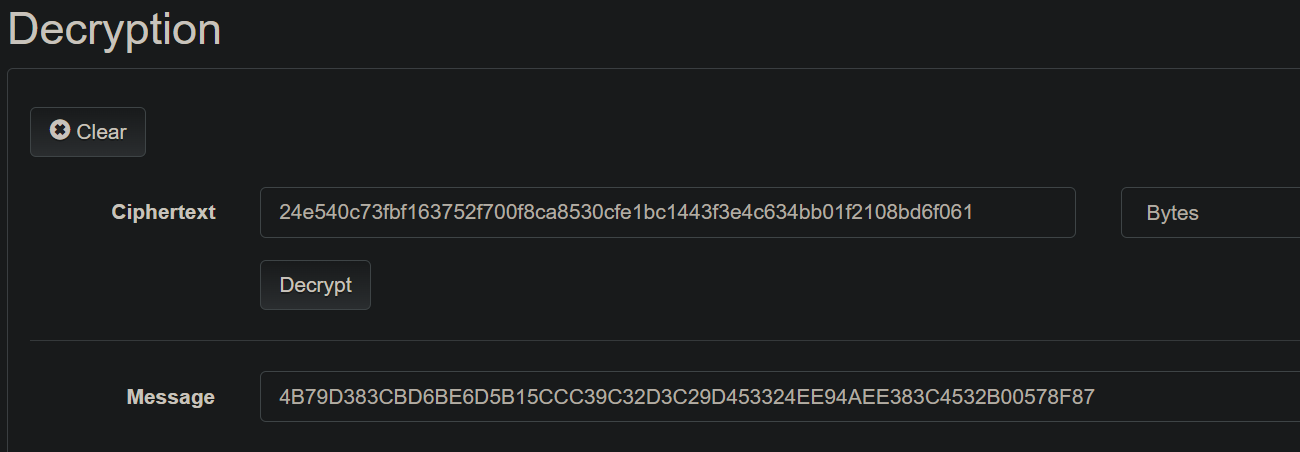
Розшифруємо повідомлення з сервера:



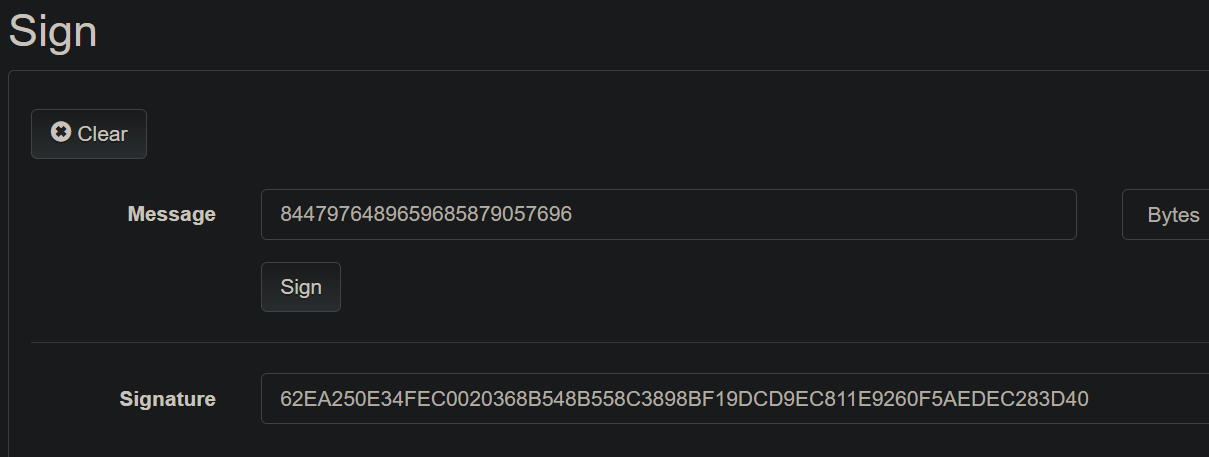


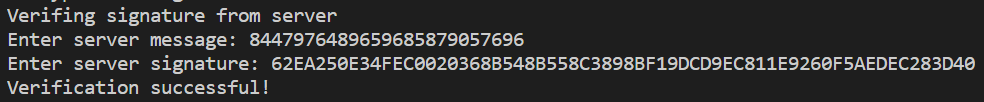
Зашифруємо повідомлення для сервера:



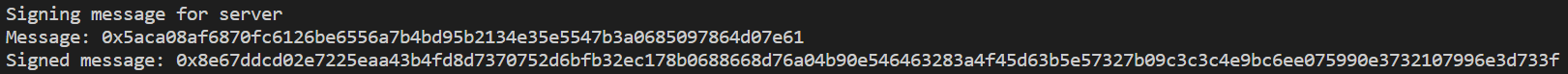


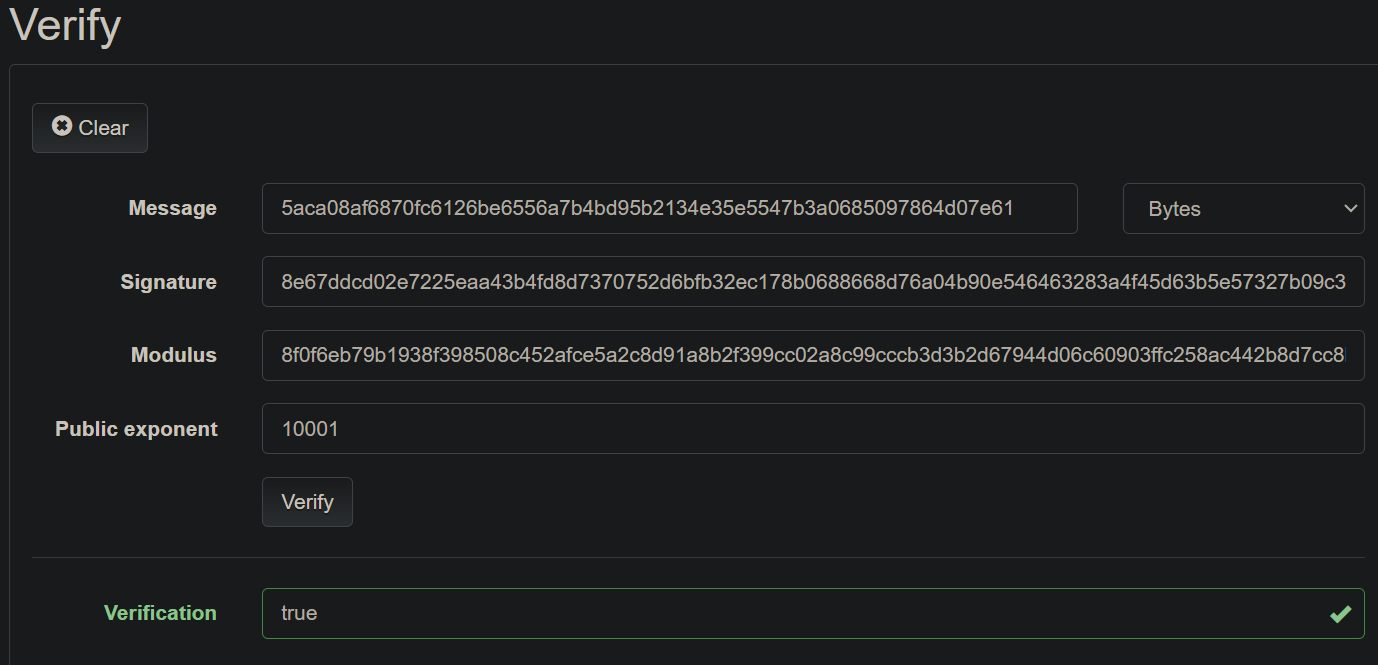
Перевіримо підпис повідомлення, отриманого з сервера:



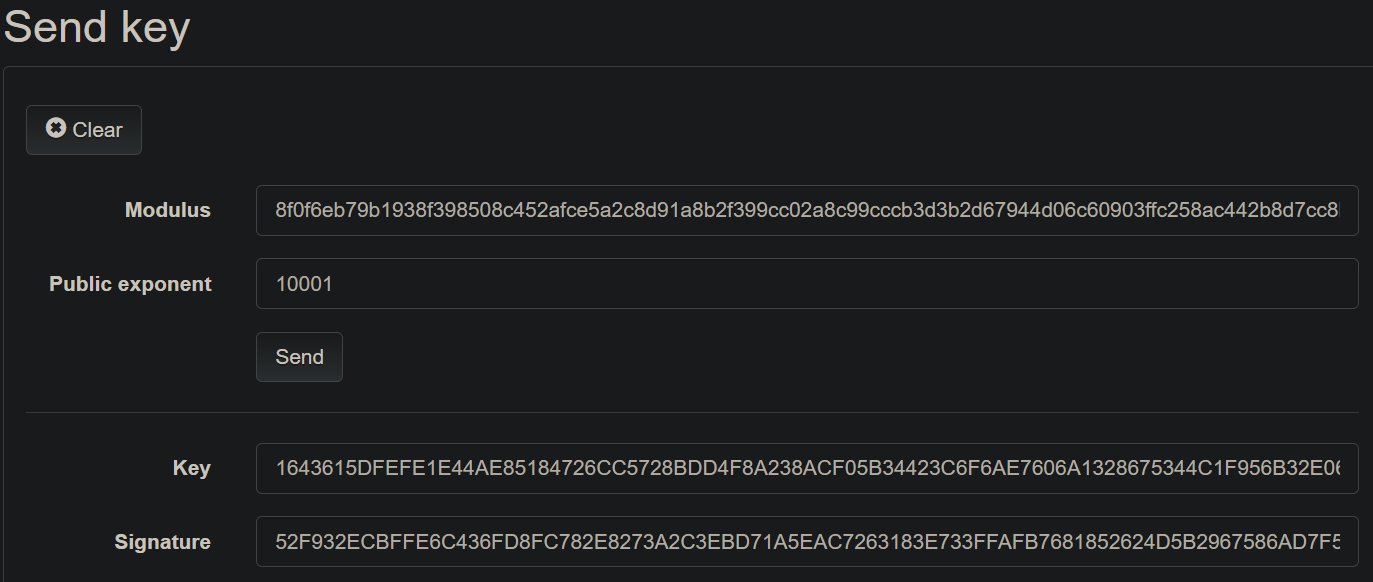


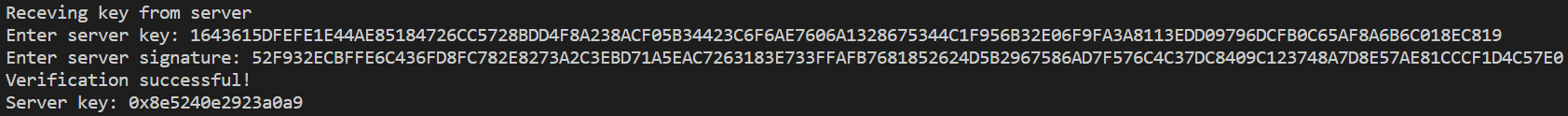
Підпишемо повідомлення для сервера:



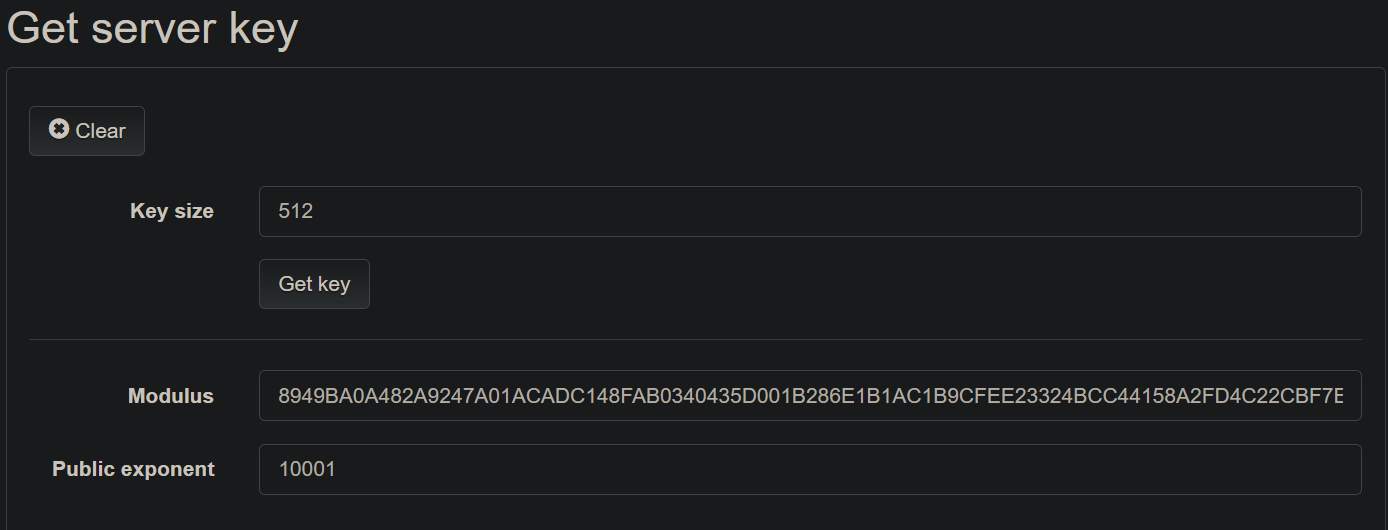


Отримаємо секретне значення з сервера:

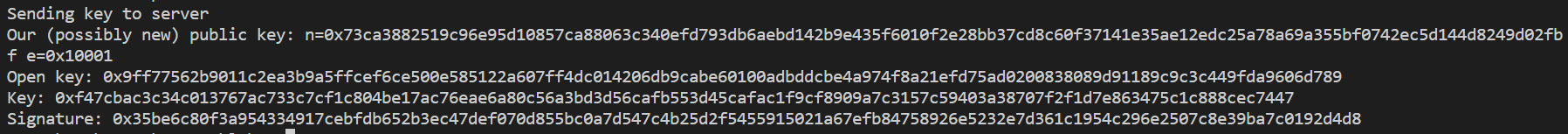


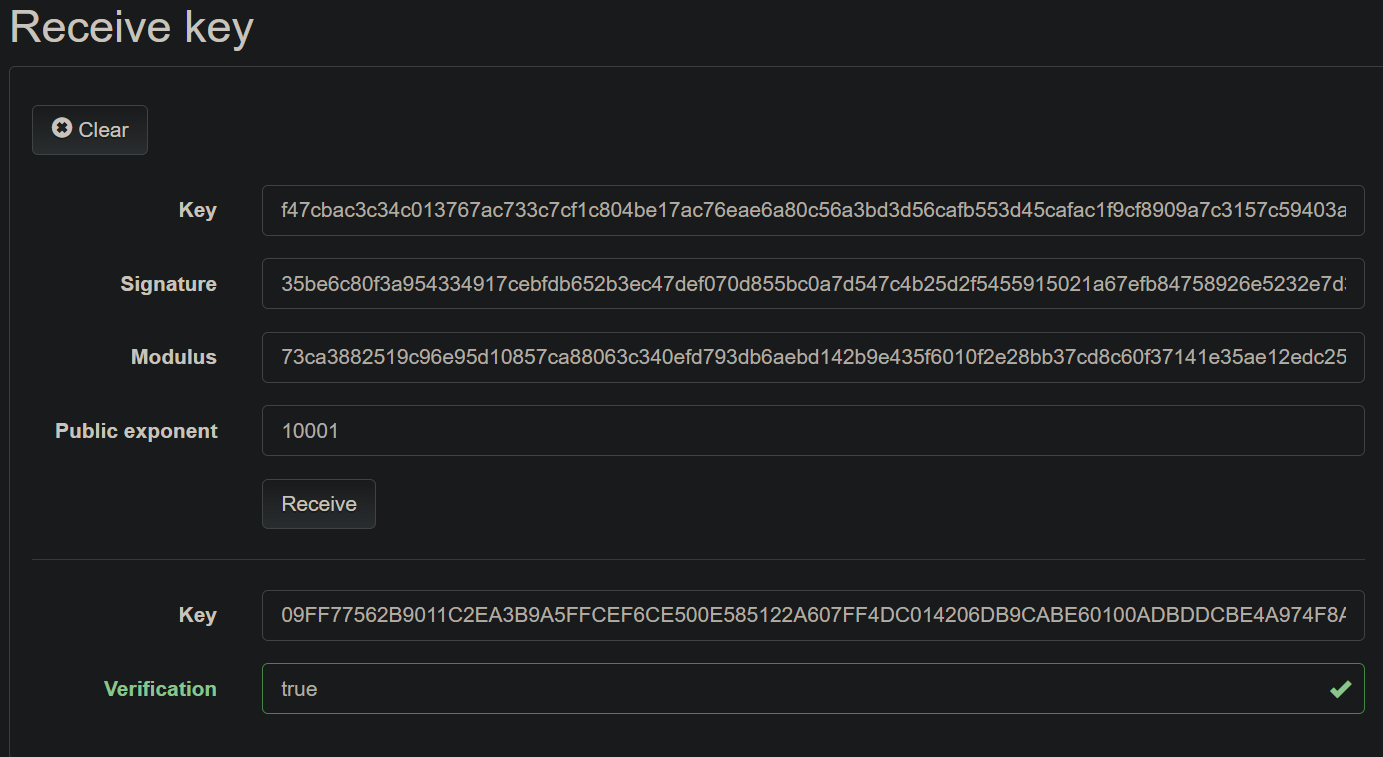


Після багатьох неуспішних спроб відправлення секретного значення, я зрозумів потрібно було брати більший ключ для сервера:



Відправимо секретне значення на сервер:





**Висновок:**

В ході цієї лабораторної я навчився використовувати криптосистему RSA та алгоритм електронного підпису.