Національний технічний університет України

«Київський політехнічний інститут»

Фізико технічний інститут
Кафедра математичних методів захисту інформації

МЕТОДИ РЕАЛІЗАЦІЇ КРИПТОГРАФІЧНИХ МЕХАНІЗМІВ

Лабораторна робота №4

Тема: “Дослідження особливостей реалізації існуючих програмних систем, які використовують криптографічні механізми захисту інформації”

Виконали:
Корж Нікіта ФІ-12мн
Тафтай Анастасія ФІ-12мп
Мазур Анастасія ФІ-12мн

Перевірила
Селюх П.В.

Київ - 2021

Мета роботи: Отримання практичних навичок побудови гібридних криптосистем.

Завдання: Розробити реалізацію асиметричної криптосистеми у відповідності до стандартних вимог Crypto API або стандартів PKCS та дослідити стійкість стандартних криптопровайдерів до атак, що використовують недосконаліть механізмів захисту операційної системи.

Хід роботи

Стандарти криптографії з відкритим ключем (Public-Key Cryptography Standards, PKCS) розроблени компанією RSA Data Security Inc.

Компанія Google анонсувала проект **`Wycheproof`** в 2016 році, в рамках якого був підготовлений інструментарій для виявлення в різних реалізаціях алгоритмів шифрування недоробок та невідповідностей з очікуваною поведінкою.

На момент 2016 року у **`Wycheproof`** реалізовано понад 80 тестів, що перевіряють на наявність більше 40 видів помилок у реалізаціях алгоритмів RSA, DSA, ECDH та Diffie-Hellman, таких як некоректний вибір констант для побудови еліптичної кривої, повторне використання параметрів у схемах формування цифрових підписів, різні види атак на AES-EAX, AES-GCM, DH, DHIES, DSA, ECDH, ECDSA, ECIES та RSA.

Зокрема, інструментарій може використовуватись для перевірки криптопровайдерів на базі Java Cryptography Architecture, таких як **Bouncy Castle**.

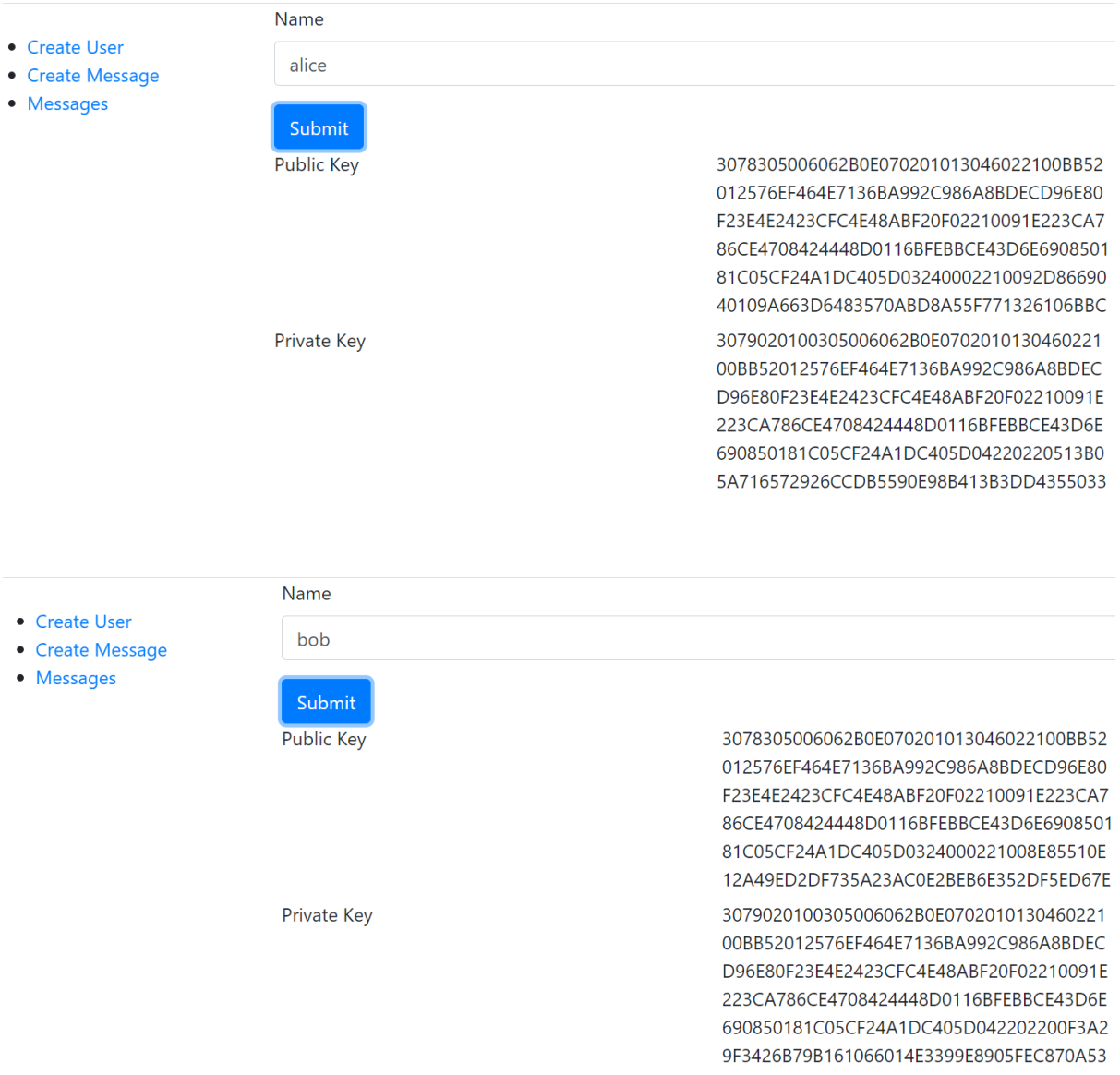
Генерація ключів в **Bouncy Castle** відбувається відповідно до стандарту **PKCS # 14** - стандарт генерації псевдовипадкових чисел.

Архітектура

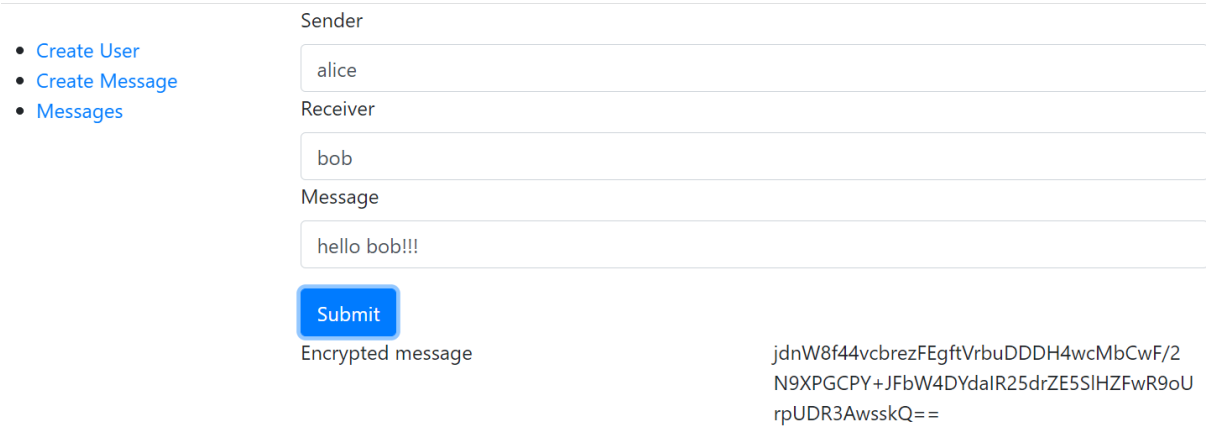
Було реалізовано клієнт серверний веб-додаток. Він підтримує функції створення користувачів разом з їх приватними та публічними ключами. Також створення повідомлення між існуючими користувачами. Під час створення отримується повідомлення, дістається публічний ключ отримувача по імені отримувача вказаному відправником, та за його допомогою воно шифрується і зберігається сервісом. Також користувачі можуть переглядати повідомлення в зашифрованому виді та дешифрувати за допомогою приватного ключа. Шифрування відбувається алгоритмом Ель Гамаля.

Приклад роботи:

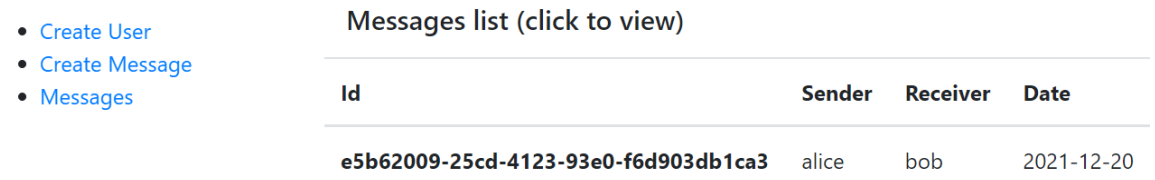
Створюємо користувачів Алісу та Боба і генеруємо їм ключі



Створюємо повідомлення від Аліси до Боба, яке шифрується і зберігається в зашифрованому виді:



Бачимо, що воно з’явилось в списку повідомлень:



Вводимо приватний ключ отримувача - Боба, щоб дешифрувати повідомлення :