УДК 004.4

А.О. Сидорук
О.В. Романюк

ВИКОРИСТАННЯ ЕЛЕКТРОННОГО ЦИФРОВОГО ПІДПИСУ ДЛЯ ПІДВИЩЕННЯ ЗАХИСТУ QR-КОДІВ

Вінницький національний технічний університет

Анотація

Розглянуто принципи створення та перевірки цифрового підпису на основі криптографічних алгоритмів SHA256 і RSA, а також етапи генерації ключів, шифрування та декодування. Проаналізовано процес інтеграції унікального ідентифікатора (installationID) та форматування даних у захищеному вигляді (data:SIGN:…). Запропоновано метод перевірки цілісності та автентичності даних із використанням роздільника :SIGN: та поверненням відповідних результатів (Valid, Invalid, NotSigned), що забезпечує надійний захист від маніпуляцій.

Ключові слова: *електронний цифровий підпис, QR-коди, криптографія, SHA256, RSA, автентифікація, безпека даних.*

Abstract

This paper explores the implementation of an electronic digital signature to enhance data security in mobile applications handling QR codes. The principles of creating and verifying a digital signature using cryptographic algorithms SHA256 and RSA, along with the stages of key generation, encryption, and decoding, are examined. The process of integrating a unique identifier (installationID) and formatting data in a secure format (data:SIGN:…) is analyzed. A method for verifying data integrity and authenticity using the :SIGN: delimiter and returning corresponding results (Valid, Invalid, NotSigned) is proposed, ensuring robust protection against manipulations.

Keywords: *electronic digital signature, QR codes, cryptography, SHA256, RSA, authentication, data security.*

Вступ

Електронний цифровий підпис - це вид електронного підпису отриманого за результатом криптографічного перетворення набору електронних даних, який додається до цього набору або логічно з ним поєднується і дає змогу підтвердити його цілісність та ідентифікувати підписувача. Електронний цифровий підпис накладається за допомогою особистого ключа та перевіряється за допомогою відкритого ключа [1].

Підхід з використанням цифрового підпису, забезпечує додаткову безпеку користувачам, що є допоможе втримати цільову аудиторію створюваного мобільного застосунку. Під час написання коду для даного фрагменту програми було використано поєднання криптографічних алгоритмів SHA256, який використовується для хешування даних та RSA для шифрування цифрового підпису. Спочатку генерується фіксована довжина коду в 256 біт з вхідних даних з використанням Secure Hash Algorithm, згодом підключається алгоритм RSA і поступово відбуваються чотири наступні кроки. Перш за все генерується ключ, під час якого обираються два простих числа, визначається модуль, застосовується функція Ейлера для, вибирається відкрита експонента і знаходиться приватна експонента оберненого за модулем числа. Другим кроком відбувається розподіл ключа на приватну і публічну частини, публічний ключ далі буде пересилатися за допомогою надійного каналу зв’язку Подальшими кроками є шифрування та дешифруванння [2]. На рисунку 1 зображено алгоритм роботи цифрового підпису.

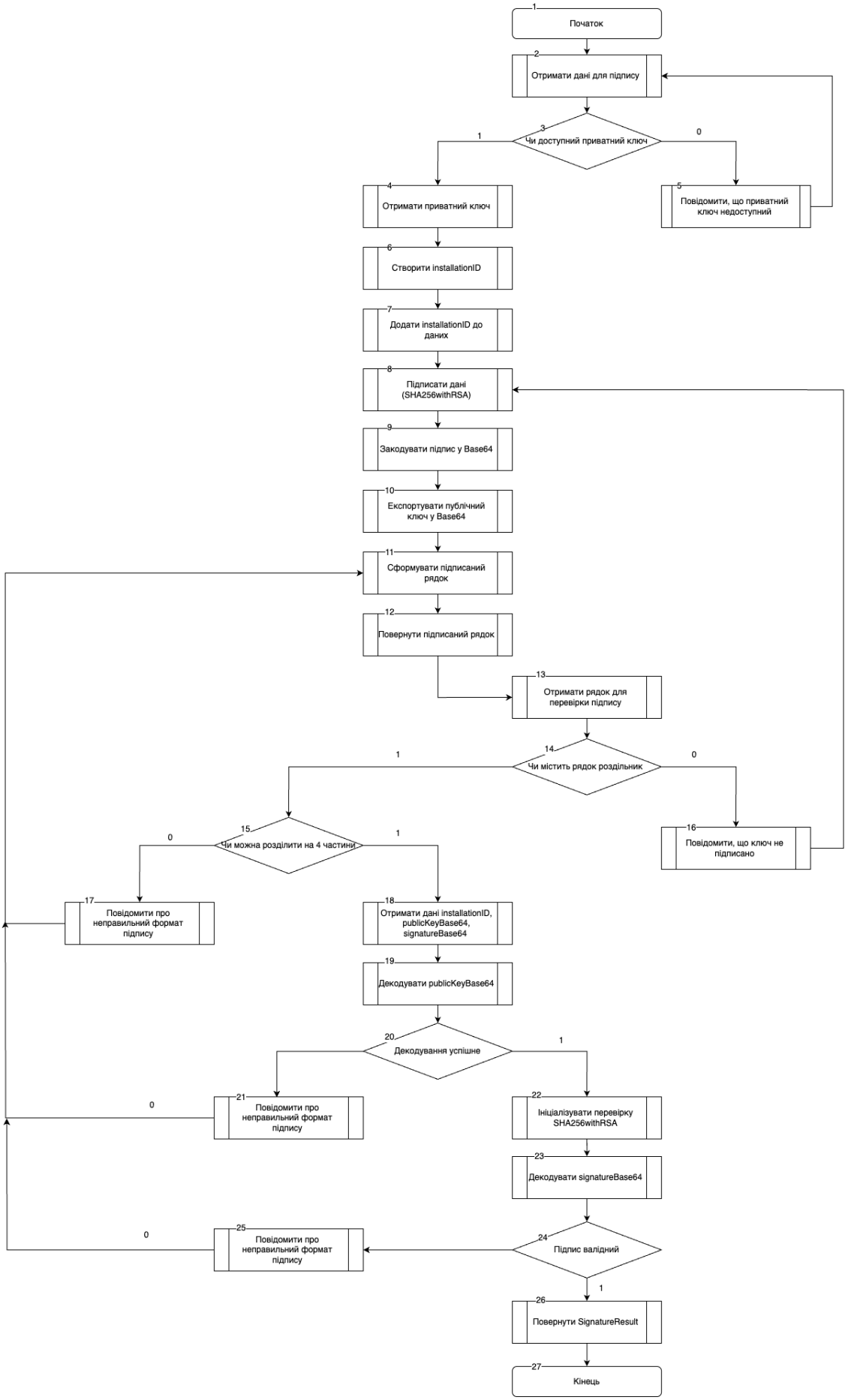

Рисунок 1 – Алгоритм роботи цифрового підпису

Починається все з простої перевірки: чи доступний приватний ключ у надрах «SecureKeyManager». Якщо ключ наявний, то наступним кроком створюється новий унікальний «installationID», далі його необхідно додати до даних, та підписати за алгоритмом SHA256withRSA. Таким чином відбувається кодування підпису у формат Base64, наступним кроком йде експортування відкритого ключа у Base64 і з метою формування рядку у форматі data:SIGN:installationID:SIGN:publicKeyBase64:SIGN:signatureBase64. У разі відсутності приватного ключа користувачу повідомляється про помилку, і повертається значення null.

Після цього нагально необхідно перевірити підпис. Усе починається з отримання рядка аби перевірити та проаналізувати наявність роздільника :SIGN:. Якщо перевірка пройдена успішно і роздільник присутній, рядок треба розділяти на частини, і перевірити, чи є всі потрібні чотири компоненти: дані, «installationID», закодований відкритий ключ і підпис. Якщо всі етапи пройшли без помилок, то відкритий ключ декодується, а опісля ініціалізується перевірка за алгоритмом SHA256withRSA, додаються дані та «installationID», а сам підпис надалі декодується з Base64 для подальшої валідації.

Останнім етаом є аналіз даних, що було отримано в результаті перевірки. Якщо все гаразд і підпис валідний, то буде повернено «SignatureResult.Valid» і користувач зможе підписати QR-код, який буде генеруватися в майбутньому. У випадку ж виникнення помилки і невідповідності підпису буде встановлено результат «SignatureResult.Invalid» і виведено повідомленням для користувача з метою повідомити про недійсність підпису. У випадку виникнення іншої помилки, наприклад, декодування ключа чи формат підпису неправильні, то буде повернено «SignatureResult.Invalid» із відповідним повідомленням про помилку. В кінці-кінців, якщо рядок не містить роздільника :SIGN:, то результатом буде повернення «SignatureResult.NotSigned».

Висновки

В ході дослідження було розглянуто основні принципи роботи електронного цифрового підпису. Було проаналізовано алгоритми, які використовуються для впровадження подібного роду захисту для застосунків.

Запропонований підхід є інноваційним, оскільки поєднує традиційні методи захисту даних з QR-кодами, які наразі використовуються більше для передачі більше публічної інформації аніж конфіденційної. Використання даного методу дозволить розширити межі використання QR-кодів для більш особистої інформації, такої як адреси проживання, банківських рахунків чи медичних карток.

Тому впровадження, електронного цифрового підпису є важливим кроком для підвищення надійності QR-кодів.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Електронний цифровий підпис | Вікіпедія [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/%D0%95%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D0%BD%D0%B8%D0%B9_%D1%86%D0%B8%D1%84%D1%80%D0%BE%D0%B2%D0%B8%D0%B9_%D0%BF%D1%96%D0%B4%D0%BF%D0%B8%D1%81 (дата звернення: 06.06.2025).

2. Як правильно впроваджувати електронний цифровий підпис у додатки | DOU [Електронний ресурс] – Режим доступу до ресурсу: https://dou.ua/forums/topic/43026/ (дата звернення: 07.06.2025).

Сидорук Анна Олександрівна – студентка групи 4ПІ-21б, факультет інформаційних технологій та комп’ютерної інжене

рії, Вінницький національний технічний університет, м. Вінниця, e-mail: alanamiael.sid@gmail.com.

Романюк Оксана Володимирівна – кандидат технічних наук, доцент кафедри програмного забезпечення, Вінницький національний технічний університет, м. Вінниця, e-mail: romaniukoksanav@gmail.com.

Anna Sydoruk – student of group 4PI-21b, Faculty for Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: alanamiael.sid@gmail.com.

Oksana Romaniuk – Ph.D., Associate Professor of Software Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: romaniukoksanav@gmail.com.