Звіт з виконання лабораторної роботи 4
Варіант 1В (Методи реалізації контролю доступу до ключової інформації на ЕОМ для User Endpoint Terminal. Порівняння запропонованих рішень в CNG та в PKCS)

ФІ-42мн Бондар Петро
ФІ-42мн Кістаєв Матвій

January 17, 2025

Огляд

Підгрупа: 1В.

Мета: Отримання практичних навичок побудови гібридних криптосистем.

Завдання (модифіковане):

1. Дослідити основні задачі, що виникають при програмній реалізації криптосистем.
2. Запропонувати методи вирішення задачі контролю доступу до ключової інформації, що зберігається в оперативній пам'яті ЕОМ для User Endpoint Terminal.
3. Запропонувати методи вирішення задачі контролю правильності функціонування програми криптографічної обробки інформації.
4. Порівняти з точки зору вирішення цих задач інтерфейси CNG, PKCS 11.

Основні проблеми та задачі програмної реалізації криптосистеми

При програмній реалізації будь-якої криптосистеми на пристрої, до якого у користувача буде прямий доступ, виникає набір цілком очевидних, але нетривіальних задач:

- Забезпечення коректної реалізації криптографічних алгоритмів – найбільш очевидне та вирішується на рівні проектування, розробки і тестування;
- Надійна та швидка генерація випадкових послідовностей (або випадкових чисел) – ця проблема вирішується шляхом використання криптографічно стійких генераторів ПВЧ та послідовностей (підхід до вирішення цієї задачі ми розглянули у другій роботі в цьому семестрі);
- Створення, зберігання та управління (в тому числі знищення) ключів криптографічної системи – це також включає унеможливлення атак із інформацією з побічного каналу, неможливість доступу до інформації про ключі, що знаходиться в системній та оперативній пам'яті ЕОМ.

Загалом, розробка безпечної криптосистеми для комунікації через мережу не обмежується тільки цими задачами, але конкретно в нашому контексті вони є основними, які треба розв'язати.

Задача контролю доступу до ключової інформації криптосистеми

Існує декілька способів забезпечити контроль доступу до ключів. Для цього можна, наприклад, використовувати апаратні засоби: будь-то **HSM** (*Hardware Security Module*), що являє собою окремий модуль на обчислювальних пристроях та є окремим від них пристроєм, чи **TPM** (*Trusted Platform Module*), що на сьогодні є невід'ємною частиною персональних комп'ютерів та їх безпеки. Обидва модулі дозволяють безпечно генерувати, зберігати криптографічні ключі, а також використовувати їх так само в межах цих модулів. Основна різниця полягає тільки в тому, як вони їх зберігають: TPM – в межах пристрою користувача, накладаючи обмеження на доступ до ключових даних відповідно до рівня доступу користувача, а HSM – у зашифрованому вигляді на окремому носії, що знаходиться на цьому модулі. Для найкращої безпеки ключових даних все ж найкраще використовувати HSM, так як це дозволяє краще відокремити основний пристрій від ключів криптосистеми, створюючи таким чином додатковий рівень безпеки.

Якщо ж розглядати програмний захист даних, тоді існують так звані програмні сховища ключів, що являють собою, наприклад, зашифровану базу даних. Або ж доступ до ключових даних можна обмежити за допомогою модулів безпеки операційної системи, яким є **DPAPI** (*Data Protection API*) у Windows. Саме про DPAPI ми також ще поговоримо, коли розглядатимемо **Microsoft CNG** (*CryptoAPI: Next Generation*).

Задача контролю правильності функціонування криптосистеми

Найважливішим для забезпечення правильного перебігу роботи криптосистеми є в першу чергу правильна її реалізація. Найкраще із цим допоможе, наприклад, дотримання криптографічних стандартів під час розробки, таких як, наприклад, FIPS-140 від NIST чи стандарти від ISO/IEC. Такі стандарти в точності визначають не тільки опис функціонування схеми, а також містять і еталонні тести для алгоритмів, які вони визначають.

Тож, другою важливою складовою у контролі правильності функціонування – є вичерпне тестування криптосистеми. Тести мають покривати всі крайові випадки, роботу криптосистеми за різних системних умов.

Також, у тестуванні важливо перевірити стійкість схеми до атак, які могли б значно (чи хоча б трошки) повпливати на коректність їх роботи. Але це включається у загальну криптографічну стійкість системи.

Рішення Crypto API: Next Generation vs PKCS #11

Наразі існує два поширених підходи до реалізації криптосистем на персональних комп'ютерах. CNG та PKCS мають свої особливості у реалізації криптосистем, підході до збереження ключів та обмеження до них доступу. Обидва з них надають певний інтерфейс взаємодії із криптосистемами, приховуючи процес виконання від користувача. Самі ж обчислення відбуваються в окремому модулі, будь-то окрема область в операційній системі, або вже вище згаданий HSM.

CNG (Crypto API: Next Generation)

У 2008 році разом із Windows Vista Microsoft випустили CNG, наступника і заміну Crypto API. Нова версія програмного інтерфейсу була створена із думкою про розширюваність та покращення поведінки криптографічних примітивів. Було додано декілька криптографічних примітивів, як, наприклад, ECDSA та ECDH (схеми на еліптичних кривих).

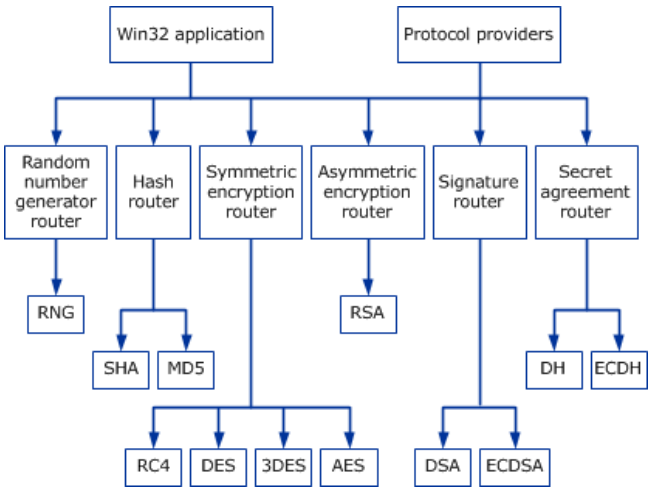

Figure 1: Структура запропонованих в CNG криптосистем.

Коректність та надійність алгоритмів.

CNG було сертифіковано із відповідністю з FIPS-140-2, що підтверджує коректність запропонованих систем, а також їх безпечність та надійність.

Збереження ключової інформації.

CNG відповідає вимогам Common Criteria, зберігаючи та використовуючи довгострокові приватні ключі

користувача в безпеці. Для цього має дотримуватись вимога невикористання цих ключів саме у процесі виконання додатку. Доступ до ключів надається тільки через KSR (*Key service router*), що керує використанням, оновленням та переміщенням цих ключів із сховища ключів або KSP (*Key service provider*).

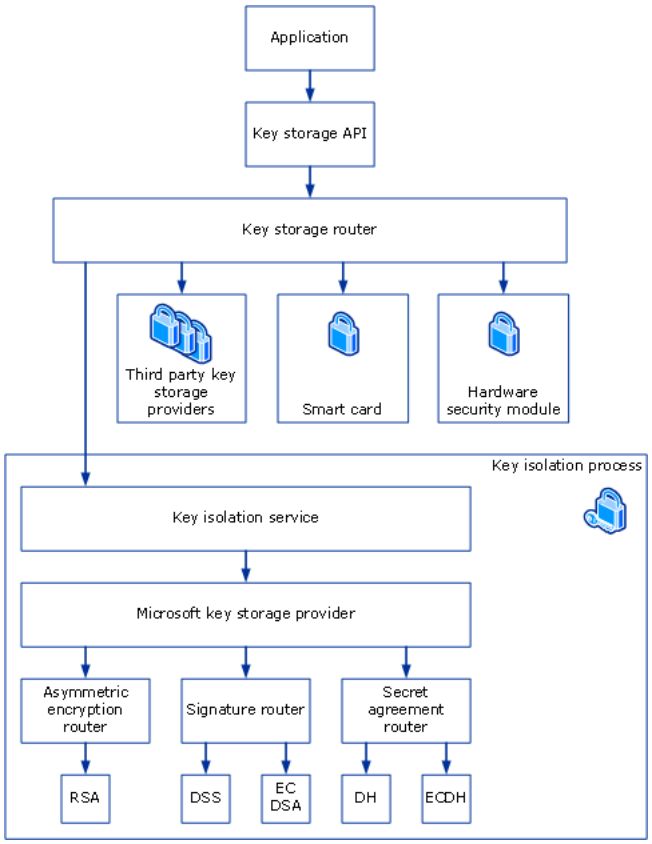

Figure 2: Структура KSP та доступу до них.

Ця функція називається ізоляцією ключа та ввімкнена автоматично, починаючи із Windows Vista та Windows Server 2008.

CNG також має функцію імпортування/експортування ключів, що зберігаються у сховищі, для передачі ключів між комп'ютерами. Тобто за потреби, отримати чи додати ключ користувач завжди в змозі.

Захист даних та DPAPI.

Для приховання даних в межах операційної системи із мінімальною потребою в знаннях особливостей криптографії Microsoft запропонували DPAPI (*Data Protection API*), що містить в собі всього 2 типи функцій `CryptProtectData` та `CryptUnprotectData`, які відповідно дозволяють зашифрувати та розшифрувати статичні текстові данні на ЕОМ.

PKCS #11

PKCS #11 — це стандарт криптографії з відкритим ключем, який визначає програмний інтерфейс C для створення та обробки криптографічних маркерів, які можуть містити секретні криптографічні ключі. Він часто використовується для зв'язку з апаратним модулем безпеки або смарт-картками. Із 2012 року стандарт PKCS #11 керується OASIS. PKCS #11 іноді називають «Cryptoki».

Загалом, концепція Cryptoki схожа CNG, тобто надавати простий інтерфейс користувачу, який створить безпечну абстракцію над певною криптосистемою та її ключами. Так само як і CNG, PKCS #11 дозволяє зберігати та використовувати ключі у відокремленому середовищі (наприклад HSM).

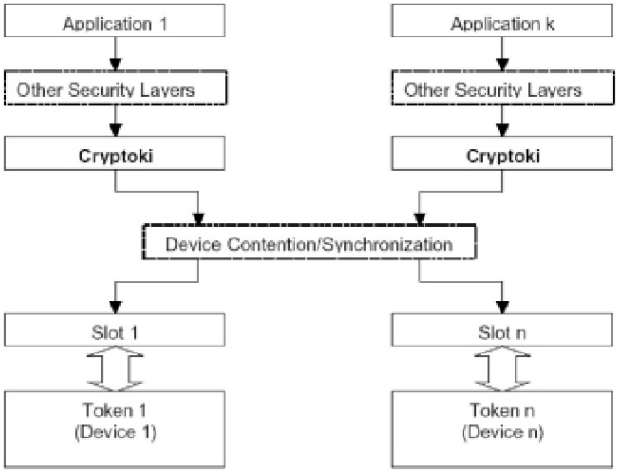

Figure 3: Абстракція Cryptoki та схема доступу до пристроїв.

Основним плюсом цього стандарту є багатоплатформність. Якщо MS CNG більше підходить до систем на Microsoft Windows, то Cryptoki дає дещо більш гнучкий вибір операційної системи.

Висновки

В ході цієї лабораторної роботи ми розглянули основні проблеми, що виникають при програмній реалізації криптосистем на персональних комп'ютерах. Проаналізувавши ці проблеми, було запропоновано декілька стандартних та найбільш поширених підходів до вирішення цих проблем. Також, ми розглянули два інтерфейси: Crypto API: Next Generation, запропонований Microsoft, що наразі використовується у ОС Windows, та Cryptoki, визначений стандартом PKCS #11; та подивилися на підходи вирішення зазначених вище проблем в межах цих інтерфейсів.