Лабораторна робота № 4

**Тема: Дослідження особливостей реалізації існуючих програмних систем, які використовують криптографічні механізми захисту інформації.**

**Мета роботи: Отримання практичних навичок побудови гібридних криптосистем.**

**Завдання на лабораторну роботу:**

Для першого типу лабораторних робіт – дослідити основні задачі, що виникають при програмній реалізації криптосистем. Запропонувати методи вирішення задачі контролю доступу до ключової інформації, що зберігається в оперативній пам’яті ЕОМ для різних (обраних) операційних систем. Запропонувати методи вирішення задачі контролю правильності функціонування програми криптографічної обробки інформації. Порівняти з точки зору вирішення цих задач інтерфейси Crypto API, PKCS 11.

Оформлення результатів роботи: звіт з методами вирішення поставлених задач

***Вимоги до криптосистемами***

Процес криптографічного закриття даних може здійснюватися як програмно, так і апаратно. Апаратна реалізація відрізняється істотно більшою вартістю, проте їй притаманні такі переваги: висока продуктивність, простота, захищеність і тощо. Програмна реалізація більш практична, допускає відому гнучкість у використанні.

Для сучасних криптографічних систем захисту інформації сформульовані наступні загальноприйняті вимоги:

* Зашифроване повідомлення повинно піддаватися читанню тільки при наявності ключа;
* Число операцій, необхідних для визначення використаного ключа шифрування за фрагментом шифрованого повідомлення і відповідного йому відкритого тексту, має бути не менше загального числа можливих ключів;
* Число операцій, необхідних для розшифрування інформації шляхом перебору різноманітних ключів повинно мати сувору нижню оцінку і не виходити за межі можливостей сучасних комп'ютерів (з урахуванням можливості використання мережевих обчислень);
* Знання алгоритму шифрування не повинно впливати на надійність захисту;
* Незначна зміна ключа повинно приводити до істотної зміни виду зашифрованого повідомлення навіть при використанні одного і того ж ключа;
* Структурні елементи алгоритму шифрування повинні бути незмінними;
* Додаткові біти, що вводяться в повідомлення в процесі шифрування, повинні бути повністю та надійно сховані в зашифрованому тексті;
* Довжина шифрованого тексту повинна бути рівною довжині вихідного тексту;
* Не повинно бути простих і легко встановлюваних залежностей між ключами, що послідовно використовуються в процесі шифрування.

***Завдання які потрібно вирішити при створені криптосистеми***

Впровадження криптосистем включає в себе кілька важливих завдань для забезпечення належного функціонування та безпеки програмного забезпечення:

* Вибір алгоритму: Вибір відповідних криптографічних алгоритмів на основі вимог безпеки, обмежень продуктивності та сумісності з системою. Це включає вибір алгоритмів шифрування, дешифрування, хешування та обміну ключами.
* Управління ключами: Розробка надійної системи управління ключами для безпечної генерації, зберігання, розподілу та ротації ключів. Сюди входить генерація ключів, зберігання ключів, протоколи обміну ключами та обробка відкликання ключів.
* Шифрування та дешифрування: Реалізація процесів шифрування та дешифрування з використанням обраних алгоритмів. Забезпечення правильної інтеграції цих процесів у програмне забезпечення, зберігаючи при цьому ефективність і безпеку.
* Генерація випадкових чисел: Генерація високоякісних випадкових чисел, необхідних для криптографічних операцій, таких як створення криптографічних ключів і векторів ініціалізації. Забезпечення випадковості та непередбачуваності цих чисел має вирішальне значення.
* Безпечна передача даних: Впровадження безпечних протоколів зв'язку (наприклад, SSL/TLS) для захисту даних під час передачі. Це передбачає інтеграцію криптографічних протоколів у мережевий зв'язок для створення захищених каналів.
* Безпечне зберігання: Розробка механізмів для безпечного зберігання конфіденційної інформації, наприклад, ключів шифрування та даних користувачів. Сюди входить впровадження протоколів безпечного зберігання та механізмів контролю доступу.

Кожне з цих завдань вимагає ретельного планування, реалізації та постійного моніторингу для забезпечення безпеки та ефективності криптографічних функцій програмного забезпечення.

***Методи вирішення проблеми контролю доступу до оперативної пам’яті***

Контроль доступу до ключової інформації, що зберігається в оперативній пам'яті комп'ютера, має вирішальне значення для підтримки безпеки криптографічних систем. Ось кілька методів вирішення цієї проблеми:

* Шифрування оперативної пам'яті: Впровадження повного шифрування пам'яті (FME) або методів скремблювання пам'яті для шифрування всього вмісту оперативної пам'яті. Це запобігає несанкціонованому доступу до конфіденційних даних, включаючи криптографічні ключі, шляхом їх шифрування при зберіганні та розшифрування лише при доступі до них авторизованих процесів.
* Апаратні модулі безпеки (HSM): Використання спеціальних апаратних пристроїв, таких як HSM, які безпечно зберігають криптографічні ключі та керують ними. HSM забезпечують безпечне середовище для зберігання ключів і криптографічних операцій, часто з фізичним захистом від підробки та несанкціонованого доступу.
* Ізоляція пам'яті: Використання методів ізоляції пам'яті, таких як Trusted Execution Environments (TEE) або безпечні анклави (наприклад, Intel SGX або AMD SEV) для створення ізольованих областей в пам'яті. Ці ізольовані середовища запобігають доступу несанкціонованих процесів або програмного забезпечення до конфіденційних даних, включаючи криптографічні ключі.
* Поділ ключів/поділ секретів: Поділ криптографічних ключів на кілька частин за допомогою схем розділення ключів або спільного використання секретів. Кожна частина зберігається окремо в різних місцях пам'яті, і для відновлення ключа потрібна комбінація цих частин. Це додає додатковий рівень безпеки, оскільки зловмиснику складніше отримати повний ключ з однієї комірки пам'яті.
* Безпечне видалення та очищення: Впровадження алгоритмів або процедур безпечного видалення для стирання криптографічних ключів з пам'яті, коли вони більше не потрібні. Це гарантує, що залишкові сліди ключів не залишаться в пам'яті, зменшуючи ризик несанкціонованого доступу через скрейпінг пам'яті або інші методи.

Впровадження комбінації цих методів, адаптованих до конкретних вимог і обмежень системи, може значно підвищити безпеку криптографічної ключової інформації, що зберігається в оперативній пам'яті комп'ютера.

***Методи вирішення проблеми коректності криптографічної обробки інформації***

Забезпечення коректності та точності програм криптографічної обробки інформації має вирішальне значення для їх надійності та безпеки. Ось кілька методів для вирішення цього завдання:

* Перегляд коду та статичний аналіз: Проведення ретельних перевірок коду та використання інструментів статичного аналізу для виявлення потенційних вразливостей, помилок або логічних недоліків у криптографічному коді. Це передбачає ретельну перевірку кодової бази на відповідність найкращим практикам та рекомендаціям з безпеки.
* Тестування та покриття тестами: Впровадження комплексних стратегій тестування, включаючи модульні тести, інтеграційні тести та наскрізні тести, спеціально сфокусовані на криптографічних функціях. Забезпечення високого рівня тестового покриття допомагає виявляти помилки і гарантує очікувану поведінку криптографічних операцій.
* Підписання коду та перевірка цілісності: Впровадження механізмів підписання коду для забезпечення цілісності та автентичності криптографічного програмного забезпечення. Перевірка підписів перед виконанням допомагає запобігти фальсифікації та гарантує, що код не був змінений зловмисниками.
* Моніторинг і логування: Впровадження надійних механізмів реєстрації та моніторингу для відстеження поведінки криптографічних функцій в режимі реального часу. Моніторинг аномалій може допомогти виявити потенційні проблеми або несанкціонований доступ до конфіденційних даних.
* Регулярні оновлення та управління виправленнями: Постійне оновлення криптографічних бібліотек і залежностей для включення останніх виправлень і поліпшень безпеки, що знижує ризик використання відомих вразливостей.

Поєднуючи ці методи та інтегруючи мислення, орієнтоване на безпеку, протягом усього життєвого циклу розробки програмного забезпечення, розробники можуть значно підвищити коректність та надійність програм для криптографічної обробки інформації.

***Crypto API***

Призначення:

Інтерфейс операційної системи: Crypto API - це, як правило, інтерфейс, що надається операційною системою (наприклад, Windows CryptoAPI або Linux libcrypto) для полегшення криптографічних операцій в середовищі ОС.

Доступ до криптографічних сервісів: Пропонує розробникам набір функцій та інструментів для доступу до криптографічних сервісів, що надаються операційною системою, включаючи шифрування, дешифрування, хешування, генерацію ключів та цифрових підписів.

Основні можливості:

Управління ключами: Crypto API включає функції для керування криптографічними ключами, як симетричними, так і асиметричними, у сховищі ключів операційної системи.

Криптографічні операції: Надає стандартизовані інтерфейси для різних криптографічних операцій, забезпечуючи узгодженість та інтероперабельність між додатками, які використовують API.

Варіанти використання:

Криптографія системного рівня: Використовується переважно для криптографічних операцій системного рівня, часто інтегрується в додатки для забезпечення безпеки даних у середовищі ОС.

Вбудована інтеграція: Пропонує вбудовану інтеграцію з операційною системою, що робить її зручною для розробників, які працюють у цих специфічних середовищах.

***PKCS***

Призначення:

Стандартизація: PKCS відноситься до набору стандартів, розроблених RSA Laboratories (зараз є частиною RSA Security), які визначають формати, протоколи та операції, пов'язані з криптографією з відкритим ключем.

Інтероперабельність: Стандарти PKCS спрямовані на забезпечення сумісності між різними системами та додатками, які дотримуються цих стандартів.

Ключові особливості:

Стандартизовані формати: Визначає формати для структур даних, таких як цифрові сертифікати, пари ключів і криптографічні повідомлення, забезпечуючи сумісність між різними системами.

Протоколи та API: Деякі стандарти PKCS, такі як PKCS#11, визначають API для криптографічних пристроїв (наприклад, смарт-карт і апаратних модулів безпеки) для управління ключами і виконання криптографічних операцій.

Варіанти використання:

Міжплатформенна сумісність: Широко використовуються для крос-платформної сумісності, визначаючи стандартизовані формати і протоколи для обміну криптографічними даними.

Стандарти управління ключами: Надає специфікації для управління ключами, забезпечуючи сумісність і безпеку при роботі з криптографічними ключами в різних системах і пристроях.

***Crypto API vs PKCS***

***Контроль доступу до ключової інформації, що зберігається в оперативній пам'яті комп'ютера:***

Crypto API:

* Управління ключами: Crypto API надає набір інтерфейсів і функцій для управління криптографічними ключами в операційній системі. Він пропонує механізми зберігання та доступу до ключів, що дозволяє програмам безпечно отримувати доступ до ключів, що зберігаються в системі.
* Контроль доступу: Залежно від базової операційної системи та реалізації, Crypto API може пропонувати контроль доступу та дозволи для управління доступом до ключів. Однак рівень деталізації та безпеки, що надається, може відрізнятися залежно від конкретної реалізації та операційної системи.

PKCS:

* Формати та зберігання ключів: PKCS визначає стандарти для форматів ключів, криптографічних токенів та зберігання ключів. PKCS#11 визначає API для криптографічних токенів, за допомогою якого можна керувати ключами та виконувати криптографічні операції. Він часто взаємодіє з апаратними модулями безпеки (HSM) для безпечного зберігання ключів.

***Функція контролю коректності програми обробки криптографічної інформації:***

Crypto API:

* Реалізація алгоритмів: Crypto API включає в себе реалізації різних криптографічних алгоритмів і примітивів. Він пропонує стандартизований інтерфейс для криптографічних операцій, що зменшує ймовірність помилок при реалізації.
* Якість коду: Якість і коректність роботи Crypto API залежить від конкретної реалізації в операційній системі. Регулярні оновлення та дотримання стандартів безпеки мають вирішальне значення для забезпечення коректності.

PKCS:

* Відповідність стандартам: Стандарти PKCS визначають різні криптографічні операції, формати і протоколи, забезпечуючи сумісність і коректність у різних системах. Дотримання цих стандартів може допомогти підтримувати коректність криптографічних операцій.
* Специфікації та настанови: PKCS також надає специфікації та настанови щодо реалізації криптографічних алгоритмів та управління ключами, які, за умови точного дотримання, можуть сприяти коректності криптографічного програмного забезпечення.

***Підсумки порівняння:***

* Сфера застосування: Crypto API зазвичай працює в середовищі певної операційної системи, забезпечуючи доступ до криптографічних сервісів системного рівня. PKCS, з іншого боку, визначає стандарти, які можуть бути реалізовані на різних платформах.
* Функціональність: Crypto API в першу чергу займається наданням доступу до криптографічних сервісів в рамках операційної системи, в той час як PKCS фокусується на стандартизації форматів, протоколів і методів управління ключами для забезпечення інтероперабельності.
* Інтеграція: Crypto API тісно інтегрований у конкретну операційну систему, для якої він розроблений, тоді як стандарти PKCS спрямовані на ширшу крос-платформну сумісність та інтероперабельність.
* Контроль доступу: PKCS#11, частина стандартів PKCS, більше фокусується на апаратних механізмах контролю доступу, пропонуючи більш високий рівень безпеки для доступу до ключів.
* Коректність і стандарти: Як Crypto API, так і PKCS, при правильному впровадженні та старанному дотриманні, сприяють коректності та надійності криптографічних операцій, надаючи стандартизовані інтерфейси, алгоритми та практики управління ключами.

Таким чином, Crypto API і PKCS служать різним цілям у криптографічному ландшафті. Crypto API діє як інтерфейс в середовищі операційної системи, в той час як стандарти PKCS забезпечують стандартизовану основу для забезпечення інтероперабельності та стандартизованих криптографічних операцій на різних платформах і системах. Інтеграція обох підходів в рамках комплексної стратегії безпеки може підвищити загальний рівень безпеки криптографічних систем.