Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Інститут електроенергетики
Факультет інформаційних технологій
Кафедра безпеки інформації та телекомунікацій

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеня магістра

студента *Павленка Євгенія Сергійовича*

академічної групи *125м-22-2*

спеціальності *125 Кібербезпека*

спеціалізації[1]

за освітньо-професійною програмою *Кібербезпека*

на тему *Розробка системи програмно-апаратного шифрування комп'ютерної інформації*

Керівники	**Прізвище, ініціали**	**Оцінка за шкалою**		**Підпис**
		рейтинговою	**інституційною**	
кваліфікаційної роботи	к.т.н., доц. Мацюк С.М.			
розділів:				
спеціальний	к.т.н., доц. Мацюк С.М.			
економічний	к.е.н., доц. Пілова Д.П.			
Рецензент				
Нормоконтролер	ст. викл. Мєшков В.І.			

Дніпро
2023

ЗАТВЕРДЖЕНО:
завідувач кафедри
безпеки інформації та телекомунікацій
_____ д.т.н., проф. Корнієнко В.І.

«_____»_____20___року

ЗАВДАННЯ
на кваліфікаційну роботу
ступеня магістра

студенту *Павленку Євгенію Сергійовичу* **академічної групи** *125м-22-2*
(прізвище ім'я по-батькові) (шифр)

спеціальності *125 Кібербезпека*

за освітньо-професійною програмою *Кібербезпека*

на тему *Розробка системи програмно-апаратного шифрування комп'ютерної інформації*

затверджену наказом ректора НТУ «Дніпровська політехніка» від _____ № ____

Розділ	Зміст	Термін виконання
Розділ 1	Стан питання. Постановка задачі	02.11.2023
Розділ 2	Спеціальний розділ	16.11.2023
Розділ 3	Економічний розділ	30.11.2023

Завдання видано _____ Мацюк С.М.
(підпис керівника) (прізвище, ініціали)

Дата видачі: _____

Дата подання до екзаменаційної комісії: _____

Прийнято до виконання _____ Павленко Є.С.
(підпис студента) (прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка містить: 77 с., 5 рис., 2 табл., 5 додатків, 15 джерел.

Мета кваліфікаційної роботи: зменшення загрози порушення конфіденційності інформації в комп'ютерній системі і програмно-апаратній системи криптографічного перетворення формованих даних.

У загальній частині проаналізовані можливі види атак, що призводять до несанкціонованого доступу до інформації; існуючі принципи і методи, що застосовуються в сучасних налаштуваннях апаратного шифрування файлів.

У спеціальній частині приведена схема криптографічного перетворення даних і ключів шифрування, показана функціональна і структурні схеми апаратної системи вибіркового шифрування інформації.

В економічній частині визначена трудомісткість і витрати на розробку і програмну реалізацію облаштувань апаратного шифрування.

ЗАХИСТ ДАНИХ, АПАРАТНІ МЕТОДИ, ВИБІРКОВЕ ШИФРУВАННЯ ФАЙЛІВ, КЛЮЧІ ШИФРУВАННЯ

THE ABSTRACT

Explanatory note consists of: 77 p., 5 fig., 2 tab., 5 appendices, 15 sources.

The objective of the qualification work: reduction of threat of violation of information confidentiality in the computer system and software/hardware systems of encryption transformation of formed data.

The general part analyzes the types of possible attacks that result in unauthorized access to information, and the existent principles and methods used in the modern hardware settings for files encryption.

The special part represents the chart of encrypted transformation of data and encryption keys; the functional and structural schemes of hardware system of selective information encryption was showed.

The economic part calculates the labor intensive and costs for development and software realization of hardware encryption.

PROTECTION OF DATA, HARDWARE METHODS, SELECTIVE ENCRUPTION OF FILES, ENCRYPTION KEYS

СПИСОК УМОВНИХ СКОРОЧЕНЬ

СКЗІ - засоби криптографічного захисту інформації;

ОС - операційна система;

ЕЦП – електронно-цифровий підпис;

НСД - несанкціонований доступ;

ПЗП - постійний пристрій, що запам'ятовує;

ОЗУ - оперативний пристрій (оперативна пам'ять), що запам'ятовує;

ПЗ - програмне забезпечення;

СЗІ - система захисту інформації;

ДВЧ - датчик випадкових чисел.

ЗМІСТ

ВСТУП

Поява глобальних комп'ютерних мереж зробила простим доступ до інформації. Легкість і швидкість доступу до даних за допомогою комп'ютерних мереж, таких як Інтернет, зробили значними наступні загрози безпеки даних:

– неавторизований доступ до інформації;

– неавторизована зміна інформації;

– неавторизований доступ до мереж і інших сервісів;

Незалежно від того, наскільки цінна інформація, що зберігається на комп'ютері, законодавством вона визнається об'єктом приватної власності. Власник цієї інформації, має право визначати правила її обробки і захисту, а також робити необхідні заходи для запобігання витоку, розкраданню, втрати і підробці інформації.

Разом з можливістю видаленого НСД, необхідно розглядати ще і фізичний доступ до певних комп'ютерів мережі. У багатьох випадках це завдання вирішується системами відеоспостереження, сигналізаціями в приміщеннях, а також правилами допуску сторонніх осіб, за дотриманням яких строго стежить служба безпеки і самі співробітники. Окрім цього, використовуються засоби розмежування доступу користувачів до ресурсів комп'ютера, засобу шифрування файлів, каталогів, логічних дисків, засоби захисту від завантаження комп'ютера з USB, парольний захист BIOS і тощо. Проте є способи обійти будь-який з перерахованих засобів залежно від ситуації і встановленого захисту. Здійснюватися це може різними шляхами: перехоплення управління комп'ютера на стадії завантаження BIOS або операційної системи, а також введення на апаратному рівні додаткового BIOS. Отже, програмно реалізувати хороший захист досить-таки проблематично.

Саме тому для запобігання подібним атакам призначений апаратний або програмно-апаратний захист комп'ютера.

Програмно-апаратні системи для захисту інформації надають користувачеві гнучкість налаштування і високу захищеність даних. Програмно-апаратна криптосистема складається з електронного пристрою, який підключається до персонального комп'ютера, і програмного забезпечення для роботи з пристроєм. У таких системах виконання функцій, некритичних до швидкості роботи і безпеки, перекладається на програмне забезпечення, що сприяє зниженню їх вартості.

Метою цієї кваліфікаційної роботи є зменшення загрози безпеки даних, порушення її конфіденційності. Це досягається за рахунок застосування програмно-апаратної системи криптографічного перетворення даних.

1. СТАН ПИТАННЯ. ПОСТАНОВКА ЗАДАЧІ

1.1 Засоби криптографічного захисту інформації.

До засобів криптографічного захисту інформації (СКЗІ) відносяться апаратні, програмно-апаратні і програмні засоби, що реалізовують криптографічні алгоритми перетворення інформації.

СКЗІ використовуються в деякій комп'ютерній системі (у ряді джерел - інформаційно-комунікаційній системі або мережі зв'язку) спільно з механізмами реалізації і гарантування деякої політики безпеки.

Засоби криптографічного захисту інформації, що забезпечують підвищений рівень захисту, можна розбити на п'ять основних груп:

Першу групу утворюють системи ідентифікації і автентифікації користувачів, такі системи застосовуються для обмеження доступу випадкових і незаконних користувачів до ресурсів комп'ютерної системи. Загальний алгоритм роботи цих систем полягає в тому, щоб отримати від користувача інформацію про особу, що засвідчує його, перевірити її достовірність і потім надати (чи не надати) цьому користувачеві можливість роботи з системою.

Другу групу засобів, що забезпечують підвищений рівень захисту, складають системи шифрування дискових даних.

Основне завдання, що вирішується такими системами, полягає в захисті від несанкціонованого використання даних, розташованих на носіях.

Забезпечення конфіденційності даних, що розташовуються на носіях, зазвичай здійснюється шляхом їх шифрування з використанням симетричних алгоритмів шифрування. Основною класифікаційною ознакою для комплексів шифрування служить рівень їх вбудовування в комп'ютерну систему.

Системи шифрування даних можуть здійснювати криптографічні перетворення даних:

- на рівні файлів (захищаються окремі файли);

- на рівні носіїв (захищаються носії цілком).

До програм першого типу можна віднести архіватори типу WinRAR, які дозволяють використати криптографічні методи для захисту архівних файлів.

Прикладом систем другого типу може служити програма шифрування Diskreet, що входить до складу популярного програмного пакету Norton Utilities.

Іншою класифікаційною ознакою систем шифрування даних є спосіб їх функціонування.

За способом функціонування системи шифрування даних ділять на два класи:

- системи "прозорого" шифрування;
- системи, що спеціально викликаються для здійснення шифрування.

У системах "прозорого" шифрування (шифрування "на льоту") криптографічні перетворення здійснюються в режимі реального часу, непомітно для користувача. Яскравим прикладом є шифрування теки Temp, Мої документи при використанні EFS Win2000 - при роботі шифруються не лише самі документи, але і створювані тимчасові файли, притому користувач не помічає цього процесу.

Системи другого класу зазвичай є утилітами, які необхідно спеціально викликати для виконання шифрування. До них відносяться, наприклад, архіватори з вбудованими засобами парольного захисту.

До третьої групи засобів, що забезпечують підвищений рівень захисту, відносяться системи шифрування даних, що передаються по комп'ютерних мережах.

Розрізняють два основні способи шифрування :

- канальне шифрування;
- крайове (абонентське) шифрування.

У разі канального шифрування захищається уся передавана по каналу зв'язку інформація, включаючи службову. Відповідні процедури шифрування реалізуються за допомогою протоколу канального рівня семирівневої еталонної моделі взаємодії відкритих систем OSI (Open System Interconnection).

Крайове (абонентське) шифрування дозволяє забезпечити конфіденційність даних, що передаються між двома прикладними об'єктами (абонентами). Крайове шифрування реалізується за допомогою протоколу прикладного або представницького рівня еталонної моделі OSI. В цьому випадку захищеним виявляється тільки зміст повідомлення, уся службова інформація залишається відкритою. Цей спосіб дозволяє уникнути проблем, пов'язаних з шифруванням службової інформації, але при цьому виникають інші проблеми. Зокрема, зловмисник, що має доступ до каналів зв'язку комп'ютерної мережі, дістає можливість аналізувати інформацію про структуру обміну повідомленнями, наприклад, про відправника і одержувача, про час і умови передачі даних, а також про об'єм даних, що передаються.

Четверту групу засобів захисту складають системи автентифікації електронних даних.

При обміні електронними даними по мережах зв'язку виникає проблема автентифікації автора документу і самого документу, тобто встановлення достовірності автора і перевірка відсутності змін в отриманому документі.

Для автентифікації електронних даних застосовують код автентифікації повідомлення або електронний цифровий підпис. При формуванні коду автентифікації повідомлення і електронного цифрового підпису використовуються різні типи систем шифрування.

П'яту групу засобів, що забезпечують підвищений рівень захисту, утворюють засоби управління ключовою інформацією.

Під ключовою інформацією розуміється сукупність усіх використовуваних в комп'ютерній системі або мережі криптографічних ключів.

Як відомо, безпека будь-якого криптографічного алгоритму визначається використовуваними криптографічними ключами. У разі ненадійного управління ключами зловмисник може оволодіти ключовою інформацією і отримати повний доступ до усієї інформації в комп'ютерній системі або мережі.

Основною класифікаційною ознакою засобів управління ключовою інформацією є вид функції управління ключами. Розрізняють наступні основні види функцій управління ключами: генерація ключів, зберігання ключів і розподіл ключів.

Способи генерації ключів розрізняються для симетричних і асиметричних криптосистем. Для генерації ключів симетричних криптосистем використовуються апаратні і програмні засоби генерації випадкових чисел. Генерація ключів для асиметричних криптосистем представляє істотно складніше завдання у зв'язку з необхідністю отримання ключів з певними математичними властивостями.

Функція зберігання ключів припускає організацію безпечного зберігання, обліку і видалення ключів. Для забезпечення безпечного зберігання і передачі ключів застосовують їх шифрування за допомогою інших ключів. Такий підхід призводить до концепції ієрархії ключів. У ієрархію ключів зазвичай входять головний ключ (майстер-ключ), ключ шифрування ключів і ключ шифрування даних. Генерація і зберігання майстер-ключів є критичними питаннями криптографічного захисту.

Розподіл ключів є найвідповідальнішим процесом в управлінні ключами. Цей процес повинен гарантувати скритність розподілюваних ключів, а також оперативність і точність їх розподілу. Розрізняють два основні способи розподілу ключів між користувачами комп'ютерної мережі:

- застосування одного або декількох центрів розподілу ключів;
- прямий обмін сеансовими ключами між користувачами.

1.2 Вимоги до засобів криптографічного захисту інформації

Для сучасних криптографічних систем захисту інформації сформульовані наступні загальноприйняті вимоги:

1. зашифроване повідомлення повинне піддаватися читанню тільки за наявності ключа, спроба ж читання без попереднього знання ключа має бути необхідно зв'язана з обчислювально складним завданням, час вирішення яких на сучасній комп'ютерній техніці перевищує час життя інформації, що захищається;

2. число операцій, необхідних для визначення використаного ключа;

3. шифрування по фрагменту шифрованого повідомлення і відкритого тексту, що відповідає йому, має бути не менше загального числа можливих ключів. Взагалі кажучи, в середньому при лобовій атаці криптоаналітику доводиться перебрати половину усіх можливих ключів, але в найгіршому випадку йому доведеться перебрати усі ключі;

4. число операцій, необхідних для розшифрування інформації шляхом перебору всіляких ключів повинно мати строгу нижню оцінку і виходити за межі можливостей сучасних комп'ютерів (з урахуванням можливості використання розподілених обчислень);

5. знання алгоритму шифрування не повинне впливати на надійність захисту (принцип Кірхгофа);

6. незначна зміна ключа повинна призводити до істотної зміни виду зашифрованого повідомлення - так званий принцип поширення помилки;

7. структурні елементи алгоритму шифрування мають бути незмінними, тобто має бути реалізований їх контроль цілісності;

8. додаткові біти, що вводяться в повідомлення в процесі шифрування, (наприклад, при доповненні відкритого тексту до довжини, кратній довжині блоку алгоритму шифрування) мають бути повністю і надійно приховані в шифрованому тексті;

9. довжина шифрованого тексту має бути рівна довжині відкритого тексту;

10. не повинно бути простих і легко встановлюваних залежностей між ключами, послідовно використовуваними в процесі шифрування;

11. будь-який ключ з множини можливих повинен забезпечувати надійний захист інформації, тобто з ключової множини мають бути виключені свідомо слабкі ключі. До таких можуть відноситися не лише ключі, що не задовольняють вимогам статистичної незалежності і рівно ймовірності знаків, але і деякі специфічні для цього алгоритму шифрування;

12. алгоритм повинен допускати як програмну, так і апаратну реалізацію, при цьому зміна довжини ключа не повинна вести до якісного погіршення алгоритму шифрування.

Процес криптографічного закриття даних може здійснюватися як програмно, так і апаратно.

1.3 Програмні СКЗІ

Як вже вказувалося, основною перевагою програмних СКЗІ є їх дешевизна і гнучкість. Разом з цими істотними перевагами програмних СКЗІ є і істотні недоліки, що зв’язані, власне, з їх найбільшою перевагою, - можливість легкої модифікації. Програма, що реалізовує деяку функцію захисту інформації, може бути досить просто модифікована зловмисником. Для усунення загрози модифікації слід якимсь чином здійснити контроль цілісності цієї програми, проте це можливо тільки за допомогою іншої програми. Перевірка цілісності одних програм за допомогою інших не є надійною. Необхідно чітко уявляти, яким чином забезпечується цілісність власне програми перевірки цілісності. Якщо обидві програми знаходяться на одних і тих же носіях, довіряти результатам такої перевірки не можна.

Ще однією серйозною проблемою програмних СКЗІ є використання оперативної пам'яті системи для операцій з криптографічним ключем - кінцевий проміжок часу криптографічний ключ є присутнім в пам'яті у відкритому виді і може бути з неї витягнутий.

Крім того, є ще одна проблема, пов'язана швидше з недоліками програмування, а не із специфікою програмних СКЗІ, наприклад, неакуратне використання тимчасових файлів, при якому в них може залишатися цінна для криптоаналізу інформація.

Дуже серйозною проблемою програмних СКЗІ є датчик випадкових чисел, використовуваних для формування ключа. Часто для генерації ключового матеріалу використовуються показання системних годинників, дані з оперативної пам'яті, і інша псевдовипадкова інформація. Слід зазначити, що жоден метод отримання випадкового числа, окрім фізично випадкових (наприклад, тепловий шум) не може бути визнаний істинно випадковим і, цілком імовірно, підкоряється деякій закономірності, а, отже, при його використанні може бути отриманий слабкий ключ. У деяких сучасних комп'ютерах є вбудований апаратний датчик випадкових чисел, проте в контексті використання програмних СКЗІ слід пам'ятати, що цей датчик доступний ОС, тому для його гарантованої стійкості потрібне використання довіреної ОС.

У зв'язку з перерахованим до програмних систем захисту інформації слід відноситися з особливою обережністю, хоча вони можуть бути дуже ефективні для захисту інформації.

У більшості поширених операційних систем передбачені вбудовані засоби шифрування дискових даних. Наприклад, в MS Windows передбачена система шифрування файлів на NTFS-дисках. Це прозоре шифрування, технологія якого заснована на сертифікатах відкритих ключів.

Такі криптографічні сервіси, як контроль цілісності, автентифікацію користувачів і даних, шифрування дискових даних і канальне шифрування можна реалізувати, використовуючи базові криптографічні примітиви: хеш-функції, схеми шифрування і електронно-цифрового підпису. Це робить обґрунтованим ієрархічний підхід до створення СКЗІ: на базі основного пристрою (апаратного або програмного), що реалізовує основні криптографічні примітиви. За допомогою використання бібліотек функцій

створюються прикладні програмні продукти, що здійснюють дискове (в т.ч. прозоре), абонентське, канальне шифрування, ЕЦП, автентифікацію користувачів і даних. Такий підхід реалізований, наприклад, в розробках фірми "Анкад", де основним пристроєм є плати КРИПТОН або програмний емулятор плати Crypton Emulator.

Багато в чому аналогічний підхід використаний Microsoft при створенні свого CryptoAPI - криптографічного інтерфейсу застосовних програм. CryptoAPI є набором функцій, призначених для роботи різних криптографічних сервісів (шифрування, хеш-функцій, ЕЦП, перевірки сертифікатів). За допомогою функцій CryptoAPI можна створювати прикладне ПЗ, покликане вирішувати криптографічні завдання різної міри складності. Особливо варто відмітити, що CryptoAPI містить тільки опис функцій криптографічних примітивів; безпосередня їх реалізація міститься в окремій бібліотеці, що називається криптопровайдером.

Microsoft надає двох своїх криптопровайдерів - простий і розширений. Вони містять реалізацію криптографічних функцій, використовуваних ОС Windows для власних потреб.

Таким чином, CryptoAPI дозволяє, з одного боку, використати криптографічні засоби Windows, а з іншого, не обмежує програміста своєю власною реалізацією криптоалгоритмів, дозволяючи використати інші (довірені) криптопровайдери.

При цьому слід враховувати, що використовуючи криптопровайдер від Microsoft, ми видаємо певний кредит довіри самій ОС Windows, незважаючи на закритість її коду і фактичну неможливість встановити факт відсутності помилок, програмних закладок недобросовісних розробників або відповідних служб. Крім того, криптопровайдер від Microsoft містить опис обмеженої кількості алгоритмів шифрування і має ряд дуже неприємних для розробника СКЗІ обмежень.

1.4 Апаратні і апаратно-програмні СКЗІ

Сучасні СКЗІ не можна строго віднести до апаратних, їх було б правильніше називати апаратно-програмними, проте, оскільки їх програмна частина непідконтрольна ОС, в літературі їх часто називають апаратними. Основною особливістю апаратних СКЗІ є апаратна реалізація (за рахунок створення і застосування спеціалізованих процесорів) основних криптографічних функцій - криптографічних перетворень, управління ключами, криптографічних протоколів.

Концепція побудови апаратних СКЗІ ставить дві основні мети: максимальне підвищення рівня захисту інформації і захисту від НСД та максимальне збільшення швидкодії криптоперетворень.

Апаратно-програмні засоби криптографічного захисту інформації поєднують гнучкість програмного рішення з надійністю апаратного. При цьому за рахунок гнучкої програмної оболонки можна швидко міняти призначений для користувача інтерфейс, кінцеві функції продукту, робити його кінцеве налаштування; а апаратна компонента дозволяє захистити від модифікації алгоритм криптографічного примітиву, забезпечити високу захищеність ключового матеріалу і часто більш високу швидкість роботи.

Використання апаратних засобів знімає проблему забезпечення цілісності системи. У більшості сучасних систем захисту від НСД застосовується зашивання програмного забезпечення в ПЗП або в аналогічну мікросхему. Таким чином, для внесення змін до ПЗ необхідно отримати доступ до відповідної плати і замінити мікросхему. У разі використання універсального процесора реалізація подібних дій потребує застосування спеціального устаткування, що ще більше утруднить проведення атаки.

Використання спеціалізованого процесора з реалізацією алгоритму роботи у вигляді інтегральної мікросхеми повністю знімає проблему порушення цілісності цього алгоритму.

На практиці часто функції автентифікації користувача, перевірки цілісності, криптографічні функції, що утворюють ядро системи безпеки, реалізуються апаратно, усі інші функції - програмно.

Перелік переваг апаратних шифраторів полягає в наступному:

- апаратний датчик випадкових чисел створює дійсно випадкові числа для формування надійних ключів шифрування і електронного цифрового підпису;
- апаратна реалізація криптоалгоритму гарантує його цілісність;
- шифрування і зберігання ключів здійснюються в самій платі шифратора, а не в оперативній пам'яті комп'ютера;
- завантаження ключів в шифруючий пристрій з електронних ключів Touch Memory (і - Button) і смарт-карт робиться безпосередньо, а не через системну шину комп'ютера і ОЗУ, що унеможливлює перехоплення ключів;
- за допомогою апаратних шифраторів можна реалізувати системи розмежування доступу до комп'ютера і захисту інформації від несанкціонованого доступу;
- застосування спеціалізованого процесора для виконання усіх обчислень розвантажує центральний процесор комп'ютера; також можна встановити декілька апаратних шифраторів на одному комп'ютері, що ще більше підвищує швидкість обробки інформації (ця перевага властива шифраторам для шини PCI);
- застосування парафазних шин при створенні шифрпроцессору виключає загрозу читання ключовій інформації по коливаннях електромагнітного випромінювання, що виникають при шифруванні даних, в ланцюгах «земля — живлення» пристрою.

При встановлені на комп'ютер спеціалізованого шифрувального устаткування виникатиме менше проблем, ніж при додаванні в системне програмне забезпечення функцій шифрування даних. В найкращому випадку шифрування повинне робитися так, щоб користувач не помічав його. Щоб

зробити це за допомогою програмних засобів, вони мають бути заховані досить глибоко в операційній системі. Виконати цю операцію безболісно з відлагодженою операційною системою дуже непросто. Але під'єднати шифрувальний пристрій до персонального комп'ютера або до модему зможе будь-який непрофесіонал.

1.5 Види налаштувань апаратного шифрування

Сучасний ринок пропонує 3 різновиди апаратних засобів шифрування інформації потенційним покупцям:

- блоки шифрування в каналах зв'язку;
- самодостатні шифрувальні модулі (вони самостійно виконують усю роботу з ключами);
- шифрувальні плати розширення для установки в персональні комп'ютери.

Майже усі налаштування перших двох типів вузько спеціалізовані. І тому треба досконально досліджувати обмеження, які при установці ці пристрої накладають на відповідні пристрої, прикладне програмне забезпечення і операційні системи до того, як приймати кінцеве рішення про їх купівлю. Інакше можна даремно витратити гроші, ніскільки не наблизившись до бажаної мети. Правда, існують компанії, які продають комунікаційне устаткування разом із заздалегідь встановленими облаштуваннями апаратного шифрування, що іноді полегшує вибір. Плати розширення для персональних комп'ютерів є більше універсальним засобом апаратного шифрування і, як правило, їх дуже легко настроїти так, щоб вони шифрували усю інформацію, що записується на жорсткий диск або пересилається в порти і дисководи. Зазвичай захист від електромагнітного випромінювання в платах розширення для апаратного шифрування відсутній, оскільки безглуздо захищати ці плати, якщо увесь комп'ютер не захищається аналогічним чином.

1.6 Узагальнена структурна схема програмно-апаратних шифраторів

До складу апаратного шифратора як правило входять:

- блок управління;
- шифропроцесор;
- апаратний датчик випадкових чисел;
- контролер;
- мікросхеми пам'яті;
- перемикачі режимів роботи;
- інтерфейси для підключення ключових носіїв.

Блок управління служить для управління роботою усього шифратора. Зазвичай він реалізований на базі мікроконтролера. Шифропроцесор є спеціалізованою мікросхемою або мікросхемою програмованої логіки (PLD - Programmable Logic Device), яка виконує шифрування даних. Для генерації ключів шифрування в пристрої передбачений апаратний датчик випадкових чисел (ДВЧ), випадковий і непередбачуваний сигнал, що виробляє статистично, перетворюється потім в цифрову форму. Обмін командами і даними між шифратором і комп'ютером забезпечується контролером. Для зберігання програмного забезпечення мікроконтролера потрібна енергонезалежна пам'ять, реалізована на одній або декількох мікросхемах. Цей же внутрішній ПЗП використовується для запису журналу операцій і інших цілей.

При зберіганні ключової інформації на носії її прочитування робиться через системну шину комп'ютера і існує можливість перехоплення. Тому апаратні шифратори зазвичай забезпечують інтерфейсом для безпосереднього підключення облаштувань зберігання ключів. Найбільш поширені серед них - роз'єми для підключення зчитувачів смарт-карт і роз'ємів для роботи з електронними пігулками Touch Memory.

Окрім функцій шифрування інформації, кожен шифратор повинен забезпечувати:

– виконання різних операцій з ключами шифрування: їх завантаження в шифропроцесор і вивантаження з нього, а також взаємне шифрування ключів;

– розрахунок імітоприставки для даних і ключів (імітоприставка є криптографічною контрольною сумою, вичисленою на певному ключі);

– генерацію випадкових чисел за запитом.

Апаратні шифратори повинні підтримувати декілька рівнів ключів шифрування. Зазвичай реалізується трирівнева ієрархія ключів. Трирівнева ієрархія передбачає використання сеансових або пакетних ключів (1-й рівень), довготривалих призначених для користувача або мережевих ключів (2-й рівень) і головних ключів (3-й рівень).

Кожному рівню ключів відповідає ключовий елемент пам'яті шифропроцесору. Мається на увазі, що шифрування даних виконується тільки на ключах першого рівня, інші призначені для шифрування самих ключів при побудові різних ключових схем.

При розшифруванні файлу спочатку за допомогою довготривалого ключа користувача розшифровується сеансовий ключ, а потім з його допомогою відновлюється інформація.

Переваги багаторівневої ключової схеми:

· Знижується навантаження на довготривалий ключ - він використовується тільки для шифрування коротких сеансових ключів; це ускладнює потенційному зловмисникові криптоаналіз зашифрованої інформації з метою отримання довготривалого ключа.

· При зміні довготривалого ключа можна швидко перешифрувати файл: досить перешифрувати сеансовий ключ із старого довготривалого на новий.

· Розвантажується ключовий носій - на ньому зберігається тільки головний ключ, а усі довготривалі ключі можуть зберігатися в зашифрованому за допомогою головного ключа виді навіть на жорсткому диску ПК.

Додаткові можливості апаратних шифраторів

Використання цілої плати розширення тільки для апаратного шифрування занадто марнотратно. Окрім функцій шифрування, виробники намагаються додати у свої пристрої різноманітні додаткові можливості, наприклад:

- Генератор випадкових чисел. Він потрібний в основному для генерації криптографічних ключів. На додаток, велика кількість алгоритмів шифрування застосовує їх і для інших цілей. Приміром, алгоритм електронного підпису ГОСТ Р 34.10 — 2001: При обчисленні підпису використовується кожного разу нове випадкове число.
- Довірене завантаження. Контроль входу на комп'ютер. Кожного разу, коли користувач включає персональний комп'ютер, пристрій вимагатиме від нього введення персональної інформації (наприклад, вставити носій з ключами). Тільки якщо пристрій розпізнає надані ключі і визнає їх «своїми», завантаження буде продовжено . Інакше користувач буде вимушений розбирати комп'ютер і виймати звідти плату шифратора, щоб включити комп'ютер (проте, як відомо, інформація на жорсткому диску також може бути зашифрована).
- Контроль цілісності файлів операційної системи. Зловмисник не зможе у вашу відсутність що-небудь поміняти в операційній системі. Шифратор зберігає у своїй пам'яті перелік усіх важливих файлів із заздалегідь визнаними для кожного контрольними сумами (чи хеш-значеннями), і комп'ютер буде блокований, якщо при черговому завантаженні контрольна сума хоч би одного з файлів не співпадатиме.

Налаштуванням криптографічного захисту даних (НКЗД) називається плата розширення з усіма вищеперерахованими можливостями. Облаштування апаратного шифрування, контролююче вхід на персональний комп'ютер і перевіряюче цілісність усіх файлів операційної системи, називається також «електронним замком». Зрозуміло, що аналогія не зовсім повна — звичайні замки сильно поступаються цим інтелектуальним

пристроям. Хоча зрозуміло, що останнім потрібне програмне забезпечення — потрібно утиліту, що генерує ключі для користувачів і зберігає їх список для упізнання «свій/чужий».

Окрім цього, потрібна програма для вибору важливих файлів і підрахунку їх контрольних сум. Доступ до цих додатків зазвичай є тільки у адміністратора з безпеки. Він повинен заздалегідь конфігурувати усі пристрої для користувачів, а якщо з'являться проблеми, повинен розібратися в їх причинах.

1.7 Атаки на елементи комп'ютера

Існує дуже цікавий принцип, який зародився у форумах і блогах при обговоренні різних програм і пристроїв з захисту даних. Цей принцип свідчить "Шифрування не досить" (у оригіналі 'Encryption is not enough') - це означає, що крім того, що дані зашифровані або ще яким-небудь чином закриті, є багато місць, де ці ж дані можна перехопити - незахищені бекапи, тимчасові файли, заражені ПК з шпигунським ПЗ і т.д.

Взагалі, основну небезпеку для захисту даних несуть локальні мережі, в які підключені клієнтські машини (особливо Інтернет) і самі користувачі. Саме дії останніх найчастіше ставлять під загрозу безпеку ПК, які залишають відкритим доступ до комп'ютера, корпусів, портів, приводів, відкривають паролі третім особам, забувають їх робити досить складними і регулярно змінювати їх, забувають встановлювати паролі на скрін-сейвери тощо.

Програмні атаки, як правило, використовують можливості інших програм в процесі їх роботи для доступу до конфіденційних даних. Будь-яка програма, що запускається, поміщає свою частину в оперативну пам'ять, а при зверненні до неї з периферійних пристроїв результати обробки запитів передає в пам'ять або буфери цих пристроїв. Це можуть бути оперативна пам'ять, канали DMA, буфери облаштувань введення-виводу, буфери відеоадаптерів. Через контролер DMA можна отримати доступ безпосередньо до захищеної пам'яті, через оперативну пам'ять доступне нульове кільце захисту системи, у тому числі усі паролі, ключі, реєстр. Через

буфер клавіатури і кадровий буфер відеокарти можна відстежувати введення паролів користувачем. Причому, резидентні програми можуть не лише прочитувати інформацію з пам'яті, буферів, але і відстежувати уведення-виведення і передачу даних (keyloggers, trackers, trojans, malware) і до того ж підміняти і емулювати запити на отримання інформації, міняти зміст пам'яті.

Загрози безпеки файлових даних розділимо на дві групи.

До першої групи віднесемо спеціальний шкідливий софтвер (програмне забезпечення), кейлогери, бекдори, а також різні апаратні способи отримання пароля.

До другої групи віднесемо витоки в незашифровані системні файли ОС. Розглянемо загрози кожної групи детальніше.

У загальному випадку троянські й інші віруси, що відносяться до першої групи, діють таким чином.

Для отримання пароля від додатка першим кроком підміняється екран для введення пароля. Далі введений пароль прочитується з буфера клавіатури. Після цього пароль прочитується з пам'яті, а додаток його оброблює і підміняє для ігнорування введення пароля.

Наступною поширеною технологією отримання паролів є копіювання буфера клавіатури у момент набору пароля на терміналі. Цей метод використовується рідко, так як для нього потрібний доступ до термінальної машини з можливістю запуску програм. Але якщо зловмисник все-таки дістає подібний доступ, ефективність цього методу дуже висока.

Кейлогери - це реєстратори натиснень клавіш. У більшості джерел можна знайти наступне визначення кейлогера: кейлогер (клавіатурний шпигун) - програмне забезпечення, основним призначенням якого є прихований моніторинг натиснень клавіш і ведення журналу цих натиснень. Кейлогер може використовуватися як програмне забезпечення, так і апаратні засоби. Апаратні кейлогери зустрічаються значно рідше, ніж програмні.

Перехоплення натиснень клавіш може використовуватися звичайними програмами і часто застосовується для виклику функцій

програми з іншого застосування за допомогою "гарячих клавіш" (hotkeys) або, наприклад, для перемикання неправильної розкладки клавіатури (як Keyboard Ninja). Існує маса легального ПЗ, яке використовується адміністраторами для спостереження за тим, що робить працівник впродовж дня, або для спостереження користувачем за активністю сторонніх людей на своєму комп'ютері. Те ж "легальне" ПЗ може використовуватися в цілях умисного викрадення секретних даних користувача - наприклад, паролів.

Бекдор - це шкідлива програма, яка використовується зловмисником для несанкціонованого доступу до певного комп'ютера. Як правило, подібного роду програмне забезпечення розраховане на забезпечення неодноразового доступу до зламаної системи і у зв'язку з цим проєктується так, щоб не бути поміченим як самим користувачем, так і засобами захисту від шкідливого ПЗ - антивірусами і файєрволами.

Нині найбільш поширені два типи бекдорів. Перший - це бекдори, побудовані за технологією "клієнт-сервер". У такому бекдорі можна, взагалі кажучи, виділити цілих дві програми - перша з них потайно встановлюється на комп'ютер, що вражається, а друга використовується для видаленого управління першою з них і встановлюється, відповідно, на комп'ютер зловмисника. Бекдори другого типу використовують для видаленого управління вбудований клієнт, працюючий через Telnet, HTTP або IRC. Для управління таким бекдором, само собою, вже не потрібно спеціальне клієнтське програмне забезпечення. Нині зловмисниками досить активно застосовуються обидва види бекдорів.

Бекдори, використовувані для несанкціонованого доступу до видаленого комп'ютера, дозволяють зловмисникам отримувати самого різного роду інформацію. До такої інформації відносяться не лише різні документи, з якими на зараженому комп'ютері працює користувач, але і його розмови в соцмережах, повідомлення, отримані по електронній пошті, і так далі, і тому подібне. Крім того, бекдор дозволяє секретно управляти комп'ютером, модифікувати паролі, здійснювати видалений доступ до

системного реєстру або конфігураційних файлів додатків, перезавантажувати систему і так далі. При цьому найбільш неприємна небезпека бекдору полягає в тому, що дуже часто бекдори використовуються для зараження комп'ютерів вірусами і троянами. Багато сучасних мережевих черв'яків або містять в собі бекдор-компоненту, або встановлюють її після зараження комп'ютера. Потім цей бекдор зазвичай використовується зловмисниками для сканування вразливостей і злому мережі через заражений комп'ютер користувача.

Друга група загроз використовує наступні уразливості комп'ютера

Файл підкочування ОС

Це файл на жорсткому диску, в який при нестачі оперативної пам'яті переміщаються окремі запущені процеси (зазвичай неактивні), звільняючи ОЗУ для завантаження інших процесів. Це означає, що конфіденційні дані, які зберігаються тільки в ОЗУ персонального комп'ютера, можуть бути записані на жорсткий диск в незашифрованому виді операційною системою Windows.

Спроби заблокувати області пам'яті, в яких знаходяться конфіденційна інформація, ОС Windows може залишити без уваги.

Буферний файл системи виводу на принтер

Є чотири основні компоненти, залучені в друк на принтері завдання, згенерованого на комп'ютері:

- Графічний інтерфейс пристрою (GDI)
- Драйвер Принтера
- Спулер друку
- Монітор Порту

Коли посилається завдання по виводу на друк на принтер, спочатку починає роботу GDI. Він відповідає за створення візуального виводу, незалежно від того, екран це або принтер. GDI викликає відповідний драйвер принтера, повідомляючи інформацію про використовуване облаштування друку, і тип даних, використовуваний для генерації завдання . Це означає, що завдання по виводу на друк заздалегідь відформатує до того, як воно буде

послано на принтер. Роль посередника між ОС і принтером виконують драйвери принтера. Драйвери принтера специфічні для конкретної ОС.

Після того, як драйвер принтера підготував створення завдання на друк для відповідного принтера, драйвер принтера передає завдання в *спулер* друку, набір DLL і драйверів пристроїв, які отримують, обробляють, планують і розподіляють завдання по виводу на друк. Подібно до драйвера принтера, спулер друку фактично складається з декількох частин, працюючих разом: маршрутизатор друку, буферний файл і процесор друку.

Монітор друку, останній в цьому ланцюжку, отримує завдання по виводу на друк з додатка-клієнта на облаштування друку. Фактично існує два монітори. Монітор мови, створений при установці двонаправленого драйвера принтера, який може послати значимі повідомлення про стан завдання на комп'ютер, встановлює зв'язок з принтером, а потім передає управління на монітор порту. Монітор порту передає завдання або на облаштування друку, або на інший сервер. Він управляє потоком інформації в порту введення-виводу, з яким пов'язано облаштування друку.

Завдання по виводу на друк було вже конфігуроване процесором друку, тому монітор порту повинен тільки турбуватися про напрям завдання в правильний порт.

Тимчасові файли

В деяких випадках ОС Windows створює тимчасові файли на диску.

Вони завжди створюються в наступних випадках:

– При спільній роботі декількох застосунків, таких як Microsoft Excel або Word;

– При виконанні стандартного застосування на основі MS - DOS;

– При друку з Windows або будь-якого додатку Windows за допомогою диспетчера черги друку .

Windows створює тимчасові файли на жорсткому диску. Завдання друку для тимчасових файлів Windows і потім відправляє для відповідного принтера в якості фонової операції.

Може з'явитися декілька файлів на жорсткому диску в різних каталогах починаючи з символу тильди (~) і закінчуючи розширенням TMP. Це можуть бути тимчасові файли, створені в Windows, які залишаються на жорсткому диску через нерегулярний вихід з сеансу Windows. У звичайних умовах ці файли закриваються і видаляються операційною системою Windows при виході з сеансу Windows. Проте при виході з Windows незвичайним способом (наприклад, перезавантаження комп'ютера або відключення під час активного сеанс Windows) файли не закриваються і не видаляються.

Сплячий режим

Коли комп'ютер переходить в сплячий режим (чи в енергозберігаючий режим), вміст системної пам'яті записується у файл на жорсткому диску. В цьому випадку ОС не може запобігти запису конфіденційної інформації (кешованих паролів, ключів шифрування і вмісту зашифрованих файлів), що знаходиться в розшифрованому вигляді в оперативній пам'яті, від запису на жорсткий диск при переході в сплячий режим. Окрім цього, у разі використання систем шифрування, змонтованих в системі, його майстер-ключ також зберігається в оперативній пам'яті комп'ютера в розшифрованому вигляді. Тому рекомендується відключати сплячий режим при роботі із зашифрованими файлами і дисками, щоб уникнути можливого просочування конфіденційної інформації.

Файли дампів пам'яті

Більшість операційних систем, включаючи Windows, можуть бути конфігуровані так, щоб записувати налагоджувальну інформацію і вміст системної пам'яті в так званий дамп пам'яті при виникненні різних помилок (системних збоїв, «блакитних екранів смерті» і так далі). Таким чином, дамп пам'яті може містити конфіденційну інформацію.

Розшифрована інформація в оперативній пам'яті комп'ютера

Більшість програм після закриття не очищають області пам'яті, з якими вони працювали. Тобто, навіть після того, як закрито додаток, який працював із зашифрованим файлом, вміст цього файлу (чи його частина) в

розшифрованому вигляді може залишитися в оперативній пам'яті до перезавантаження персонального комп'ютера.

1.8 Постановка задачі

Виходячи з описаних вище загроз, метою цієї кваліфікаційної роботи зменшення загрози порушення конфіденційності інформації в комп'ютерній системі і програмно-апаратної системи криптографічного перетворення формованих даних.

Для досягнення поставленої мети вирішувалися наступні завдання:

1. Аналіз загроз при зберіганні файлової інформації і засобів її інформаційного захисту;
2. Аналіз існуючих методів налаштувань апаратного шифрування файлів;
3. Розробка загальної схеми і функціональної структури апаратної системи криптоперетворення файлів;
4. Обґрунтування вибраних схем криптоперетворень даних, створення і зберігання ключів шифрування;
5. Розробка структурної схеми пристроїв для апаратного шифрування інформації.

Структура запропонованої системи заснована на реалізації трьох основних умов формування, зберігання і використання такої конфіденційної інформації:

1. Уся конфіденційна інформація зашифровується в процесі її набору на клавіатурі комп'ютера, тому в оперативну пам'ять поступають вже зашифровані дані.
2. Сформовані інформаційні файли також містять тільки шифровані дані.
3. Отримати цю інформацію в розшифрованому вигляді можна тільки на паперовому носії за допомогою спеціального дешифрувального пристрою.

2 СПЕЦІАЛЬНИЙ РОЗДІЛ

ОПИС ПРОГРАМНО-АПАРАТНОЇ СИСТЕМИ КРИПТОГРАФІЧНИХ ПЕРЕТВОРЕНЬ

2.1 Основи апаратного захисту інформації

Апаратний захист можна класифікувати наступним чином:

- апаратний захист програмного забезпечення;

- локальний апаратний захист (апаратний захист комп'ютера і інформації);

- апаратний захист мережі (апаратний захист внутрішньої локальної мережі з одним або декількома виходами в Інтернет).

Апаратний захист програмного забезпечення

На даний момент існує окремий напрям в комп'ютерній індустрії, що займається забезпеченням захисту ПЗ від несанкціонованого використання. У зв'язку з цим останнім часом все більшої популярності серед виробників програмного забезпечення набувають нові вдосконалені програмно-апаратні засоби захисту, відомі як "апаратні (електронні) ключі", що є одним з досить надійних способів боротьби з нелегальним копіюванням.

Електронні ключі

Під терміном "електронний ключ" передбачається апаратна реалізація системи захисту і відповідного програмного забезпечення. Сам ключ є платою, захищеною корпусом, в архітектуру якої обов'язково входять мікросхеми пам'яті і, іноді, мікропроцесор. Ключ може підключатися в слот розширення материнської плати ISA, або ж до LPT, COM, PCMCIA, USB-порту комп'ютера. У його програмне забезпечення входить модуль, який вбудовується в те, що захищається ПЗ (таким чином це програмне забезпечення "прив'язується" до ключа, а не до конкретного комп'ютера), і драйвера під різні операційні системи. Ключі у більшості своїй засновані на одній з трьох моделей існуючих апаратних реалізацій: на основі FLASH-пам’яті, PIC або ASIC-чипів. Окрім цього, в деякі ключі вбудовуються

додаткові можливості у вигляді енергонезалежної пам'яті, таймерів, вибору алгоритму кодування даних.

Апаратна реалізація електронного ключа на основі FLASH-пам’яті досить проста і є найменш стійкою до злому (стійкість визначається типом програмної частини).

В архітектуру такого ключа не входить мікропроцесор, а в подібних системах критична інформація (таблиця переходів і ключ дешифрування) зберігається в пам'яті. Крім того, такі ключі мають найменшу міру прозорості для стандартних протоколів обміну. Захист полягає в прочитуванні з ключа певних даних і ділянок коду на етапі перевірки легальності використання. Щоб дезактивувати такий захист у більшості випадків зломщикові знадобиться апаратна частина системи захисту.

Методика злому заснована на перехопленні діалогу між програмною і апаратною частинами для доступу до критичної інформації. Тобто визначається алгоритм обміну інформацією між ключем і комп'ютером, прочитується інформація з FLASH-пам’яті і пишеться відповідний емулятор.

Для PIC- і ASIC-ключів захист будується за принципово іншим методом. У їх архітектуру вже входить мікропроцесор. Окрім цього ці мікросхеми включають невелику кількість оперативної пам'яті, пам'ять команд і пам'ять для зберігання мікропрограми. У апаратній частині міститься ключ дешифрування і блоки шифрування/дешифрування даних. Ключі на цій основі набагато стійкіші до злому і є прозорішими для стандартних протоколів обміну. PIC-чіпи програмуються розробниками ключів, тому PIC-ключі є дорожчою перспективою для замовника.

Апаратну копію такого ключа зробити досить проблематично за рахунок того, що мікропрограма і внутрішня пам'ять захищені від зовнішнього прочитування, Але до таких ключів застосовані методи криптоаналізу. Досить складним є також завдання перехоплення ключа (основна обробка здійснюється апаратною частиною). Проте залишається

можливість збереження захищеної програми у відкритому вигляді після того, як система захисту відпрацювала.

Смарт-карти

Останнім часом в якості електронного ключа широко поширилися Смарт-карти. Носієм інформації в них є мікросхема. Умовно їх можна розділити на мікропроцесорні, карти з пам'яттю і криптографічні (підтримка алгоритмів DES, RSA і інших) карти.

Карти з пам'яттю або memory cards є найпростішими з класу Смарт-карт. Об'єм їх пам'яті складає величину від 32 байт до 16 кілобайт. Такі карти діляться на два типи: із захищеною і незахищеною (повний доступ) пам'яттю. Рівень захисту карт пам'яті вищий, тому вони можуть бути використані в прикладних системах невеликих фінансових оборотів.

Картами з мікропроцесором або CPU cards є мікрокомп'ютери і містять усі відповідні основні компоненти. Частина даних операційної системи мікропроцесорної карти доступна тільки її внутрішнім програмам. Також вона містить вбудовані криптографічні засоби. За рахунок усього перерахованого подібна карта досить захищена і може бути використана у фінансових застосуваннях.

Можна сказати, що, незважаючи на уявну універсальність апаратних ключів, вони все-таки схильні до злому. Розробники ключів застосовують різні способи для зведення вірогідності злому до мінімуму: захист від реасемблювання, трасування, відладчиків і багато що інше. Проте захист від трасування і відладчиків практично даремний, якщо використовується SoftIce.

Апаратний захист комп'ютера і інформації

Апаратний захист комп'ютера використовується для розмежування доступу користувачів до ресурсів комп'ютера, засобів шифрування файлів, каталогів, логічних дисків, засобів захисту від завантаження комп'ютера з носія, парольного захисту BIOS тощо.

Такий захист базується на контролі усього циклу завантаження комп'ютера для запобігання використанню різних завантажувальних носіїв і реалізується у вигляді плати, що підключається у вільний слот материнської плати комп'ютера. Її програмна частина проводить аудит і виконує функції розмежування доступу до певних ресурсів.

Шифруючі плати

Застосування засобів криптозахисту є ще одним способом забезпечення збереження інформації, що міститься на локальному комп'ютері.

Неможливо використати і модифікувати інформацію в зашифрованих файлах і каталогах. У такому разі конфіденційність інформації, що міститься на носії, прямо пропорційна стійкості алгоритму шифрування.

Шифруюча плата вставляється у вільний слот розширення PCI або ISA на материнській платі комп'ютера і виконує функцію шифрування даних. Плата дозволяє шифрувати каталоги і диски. Оптимальним є спосіб шифрування усього утримуваного жорсткого диска, включаючи завантажувальні сектори, таблиці розбиття і таблиці розміщення файлової системи. Ключі шифрування зберігаються на окремому носії.

Шифруючі плати гарантують високу міру захисту інформації, але їх застосування значно знижує швидкість обробки даних.

Апаратний захист мережі

На сьогодні досить багато розвинених компаній і організації мають внутрішню локальну мережу. Що підключаються до WWW ЛВС у більшості випадків дуже уразливі до неавторизованого доступу і зовнішніх атак без належного захисту. Такий захист забезпечує міжмережевий екран (брандмауер або firewall).

Брандмауери існують двох видів: програмні і апаратні.

Для програмних брандмауерів потрібний окремий комп'ютер на базі традиційних операційних систем Unix або Windows NT. Такий брандмауер може служити єдиною точкою входу у внутрішню мережу. Слабкість і

ненадійність подібного захисту полягає не стільки в можливих порушеннях коректної роботи самого програмного брандмауера, скільки в уразливості використовуваних операційних систем, на базі яких функціонує міжмережевий екран.

Апаратні брандмауери побудовані на базі спеціально розроблених для цієї мети власних операційних систем. У простому випадку, брандмауер - це пристрій, що запобігає доступу у внутрішню мережу користувачів ззовні. Він не є окремою компонентою, а є цілою стратегією захисту ресурсів організації. Основна функція брандмауера - централізація управління доступом. Він вирішує багато видів завдань, але основними є аналіз пакетів, фільтрація і перенаправлення трафіку, автентифікація підключень, блокування протоколів або вмісту, шифрування даних.

Апаратні реалізації ключів захисту

Апаратні ключі захисту складаються з ключа, що підключається до LPT або COM-порту комп'ютера, USB-шині, і програмного забезпечення (драйверів для різних операційних систем і модуля, що вбудовується в програму, що захищається).

Апаратна частина таких ключів виконана на мікросхемах FLASH-пам’яті, на PIC-котролерах або на замовлених ASIC-чипах. Ця елементна база відрізняється дуже низьким енергоспоживанням, тому для живлення ключів використовуються виводи, спочатку для цього не призначені (- AUTO FEED, - INIT, - SLCT IN, - STROBE або одна з інформаційних шин для LPT-порту; DTR, RTS для COM-порту). Інформаційний обмін між ключем і комп'ютером відбувається, зазвичай, в послідовному вигляді, з використанням сигналу, що формується драйвером. Як вихідні лінії використовуються вищеперераховані виводи, а як вхідна лінія використовуються сигнали - STROBE, - ACK, BUSY, PE, SLCT або ERROR для LPT-порту і DSR, CTS для COM-порту.

Апаратні ключі, що підключаються до LPT- і COM-портів, повинні забезпечувати "прозорий" режим обміну по стандартних для цих портів

протоколах. Наприклад обмін з ключами, що підключаються до LPT-порту, вестиметься тільки при пасивному рівні сигналу - SLCT IN (тобто "принтер не вибраний"), а обмін з ключами для COM-портів відбуватиметься тільки при пасивному рівні DTR ("Data terminal ready"). Втім, ці хитрощі все одно не допомагають уникнути конфліктів із стандартними пристроями, призначеними для підключення до цих портів (останніми моделями принтерів і сканерів, що використовують двонаправлений обмін по паралельного порту або з маніпуляторами типу "миша" і модемами, що підключаються до послідовного порту). Від подібних недоліків мають бути вільні ключі, що підключаються до USB-шини.

Ключі, зроблені на основі PIC або ASIC-чипів, мають на порядок велику стійкість до злому і "прозорість" для штатних протоколів обміну. Обидві ці мікросхеми є контролерами, що містять в собі процесор, деяку кількість оперативної пам'яті, FLASH-пам’ять команд і пам'ять для зберігання мікропрограми. Мікропрограма і внутрішня пам'ять зазвичай захищається від зовнішнього прочитування, так що зробити апаратну копію ключа досить проблематично. Основна відмінність PIC-ключів від ASIC-ключів в тому, що PIC-чіпи програмуються розробником ключів (тобто він може відносно легко змінити алгоритми роботи), а ASIC-чипи є замовленими мікросхемами (тобто алгоритми жорстко задаються на етапі виробництва мікросхем). Тому ASIC-ключі виходять дешевшими, ніж зібрані на основі PIC-чипів, але з цієї ж причини захист на їх основі менш надійний (визначивши алгоритм обробки даних в одному з ASIC-чипів, можна написати емулятор ключа для усієї партії, яка, - в силу особливостей виробництва - зазвичай буває досить великий). Широко відомий випадок, коли був визначений алгоритм роботи електронного ключа виробництва компанії "Aladdin Software Security R.D". (як виявилося, ця функція може бути реалізована одним рядком на мові С) після чого з'явилася велика кількість емуляторів ключів цієї фірми. Окрім цього слід зазначити наявність в деяких ключах додаткових можливостей: енергонезалежних таймерів (для

обмеження роботи програми, що захищається, за часом), енергонезалежної пам'яті, можливість використання одного і того ж ключа для захисту декількох пакетів прикладного програмного забезпечення.

Програмна частина захисного комплексу складається з драйвера апаратного ключа і модуля, що вбудовується в застосовну програму.

1. Драйвер ключа.

Сучасні операційні системи не дозволяють застосовній програмі безпосередньо спілкуватися з портами введення/виводу, потрібно наявність драйвера, що виконується в режимі ядра ОС.

Завдання драйвера, - забезпечити найнижчий рівень обміну даними між ключем і застосовною програмою. Для ключів, що підключаються до COM, - і LPT-портам, драйвер відповідає за формування синхронізуючих і інформаційних сигналів на виходах відповідних роз'ємів і за розшифрування послідовного коду, що отримується від апаратного ключа. Крім того, для PIC - і ASIC-ключів драйвер формує послідовність (дані, прийнявши які, ключ починає обробляти усі подальші дані по заданому алгоритму), що ініціює.

Вбудовуваний модуль

Для ключів на основі PIC - і ASIC-чипів захист будується по наступному методу.

На етапі програмування ключа (чи виробництва ASIC-чипа) в нього записується мікропрограма, що реалізовує деяку функцію y = F(x1, x2,...xn), де x1...xn - вхідні параметри, а y - вихідний параметр. Звичайні один або декілька параметрів x1...xn є випадковими числами (для утруднення визначення виду функції F), один з параметрів - унікальний номер ключа ("серійний номер"), ще один - ідентифікатор програмного забезпечення, що захищається, і тому подібне. Після обробки ключем вхідних параметрів він формує і видає в комп'ютер вихідне значення y. Це значення передається в модуль, інтегрований в застосовну програму, де над цим значенням і параметрами x1...xn робиться перетворення виду

y’ = f(y,x1,x2…xn).

Результуючим значенням y' можуть бути константи, необхідні для роботи програми, ділянки програмного коду, адреси підпрограм і тому подібне. При цьому необхідна умова - неможливість відновлення функції F(x1, x2...xn) по функції f(y, x1, x2...xn), яку досить легко отримати, реасембльована ділянка застосовної програми, що відповідає за перевірку даних, отриманих від ключа. За наявності в ключі енергонезалежної пам'яті і/або таймерів можна додати їх поточні значення в якості аргументів функції F(x1, x2…xn), що розширює можливості побудови систем захисту.

Для злому апаратних ключів захисту застосовують:

- знімання інформації безпосередньо з роз'єму і створення апаратних емуляторів ключів;

- визначення алгоритму обміну шляхом перехоплення звернень до регістрів управління COM - і LPT -портами.

При цьому досить мати один екземпляр легального ключа, щоб визначити якщо не вид функції F(x1, x2.xn), то хоч би набір значень у залежно від набору вхідних аргументів x1.xn. Природно, якщо один з аргументів є випадковим числом або значенням вбудованого таймера або енергонезалежної пам'яті, то написання емулятора сильно утруднюється; але в цьому випадку залишається можливість просто "відкусити" захисний модуль, підставивши у відповідних місцях програми значення, вичислені при використанні одного "легального" ключа.

Для ускладнення злому захисту застосовуються наступні методи:

– захист від реасемблювання (умовні і безумовні переходи по вмісту регістрів або елементів пам'яті, після яких ставляться дещо байт, реасембльованих в реальну команду) процесора;

– «розмивання» програмного коду шляхом розміщення його в різних місцях програмного модуля з виконанням безумовних або неочевидних умовних переходів після кожної асемблерної команди);

– захист від трасування (перехоплення INT1 і INT3, модифікація регістрів SS/SP/ESP, робота в режимі заборонених переривань і тому подібне);

– захист від відладчиків (перевірка їх наявності через API - функції);

– перевірка зміни коду драйвера ключа або вбудовуваного модуля (перевірка контрольної суми, перевірка по контрольних точках і т.п).

Якщо система захисту виявляє спробу злому, працездатність застосовної програми навмисно порушується. Це може проявлятися і як неможливість запуску застосовної програми, і як її неправильне функціонування. У останньому випадку злом програми, що захищається, стає ще важче, оскільки незрозуміло, на якому етапі спрацював захист.

2.2 Опис структури системи

Пропонована система криптографічних перетворень призначена для апаратного шифрування комп'ютерних файлів і конструктивно є ключем шифрування і ключем дешифрування. Кожен з цих ключів поєднує в собі апаратний ключ захисту з програмною реалізацією криптографічного перетворення інформації.

Схема застосування цих пристроїв полягає в наступному;

– Шифрування робиться посимвольно у момент набору тексту файлу на клавіатурі. При цьому облаштування шифрування встановлюється між клавіатурою і самим комп'ютером за допомогою стандартних інтерфейсів USB.

– Дешифрування раніше зашифрованих таким чином файлів робиться при передачі потоку байт на принтер. При дешифруванні пристрій встановлюється між комп'ютером за допомогою стандартного інтерфейсу USB і паралельного LPT-порта принтера.

Схема підключення облаштувань шифрування і розшифрування утримуваного файлу пристроїв показана на рис 2.1.

Схема криптографічних перетворень повністю аналогічна схемі криптографічних систем з відкритим ключем – відкритим ключем шифрується інформація, закритим – розшифровується.

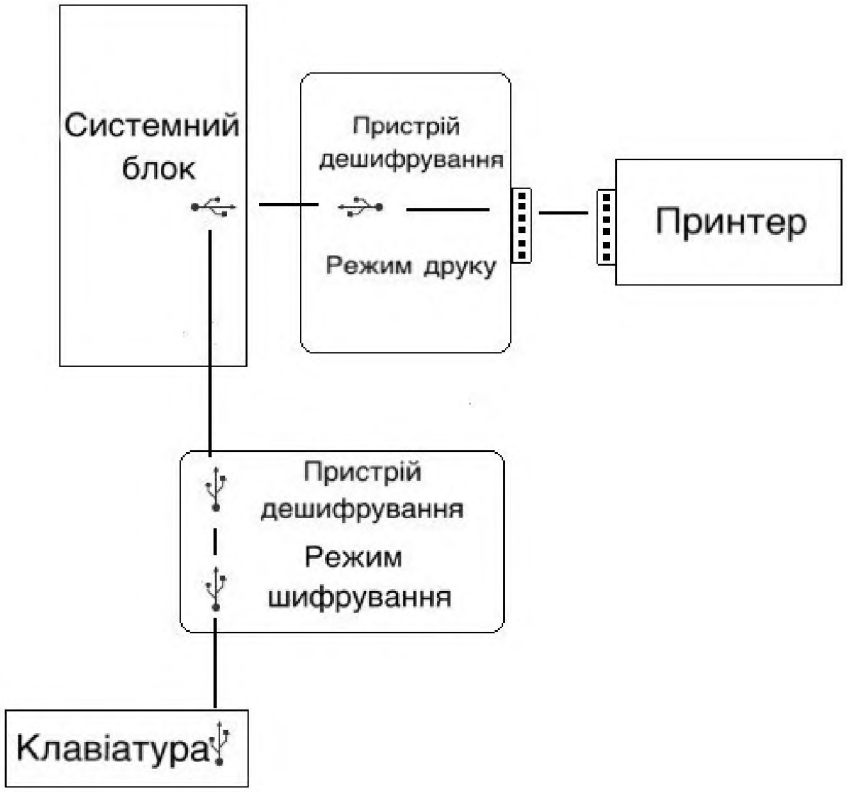

Рисунок 2.1 - Схема підключення облаштувань системи

Шифрування даних робиться у момент їх набору на клавіатурі і тому в оперативну пам'ять потрапляє вже шифрована інформація. Розшифрування даних також проводиться поза оперативною пам'яттю і тому така схема і підключення облаштування шифрування виключає загрози витоку інформації вказані в п.1.2.

2.3 Схема шифрування і розшифрування

Апаратні шифратори повинні підтримувати декілька рівнів ключів шифрування. Зазвичай реалізується дворівнева або трирівнева ієрархія ключів. Більша кількість рівнів не дає помітного поліпшення якості захисту.

Дворівнева ієрархія передбачає використання сеансових або пакетних ключів (1-й рівень) і головних ключів (2-й рівень).

Кожному рівню ключів відповідає ключовий елемент пам'яті мікропроцесора. Шифрування даних виконується тільки на ключах першого рівня, інші призначені для шифрування самих ключів при побудові різних ключових схем. У системі, що розробляється, передбачається наступна дворівнева ключова схема криптографічних перетворень (рис.2.2).

Попередній етап:

- Закритий (головний) ключ зберігається зовнішньому USB-накопичувачі.

- За допомогою цього головного ключа за спеціальною процедурою робиться генерація відкритого ключа користувача.

- Цей відкритий видається інтерфейсу файлу, що здійснює шифрування.

- Закритий (головний) ключ зберігається у довіреної особи.

Етапи шифрування:

1. Відкритий ключ вводиться в облаштування шифрування;

2. За допомогою вбудованого в облаштування шифрування генератора випадкових чисел генерується сеансовий ключ;.

3. Сеансовий ключ шифрується за допомогою відкритого ключа користувача;

4. За допомогою цього ж сеансового ключа шифрується вміст файлу і створюється новий файл, що зберігає зашифровану інформацію;

5. У зашифрований файл додається зашифрований сеансовий ключ.

Етапи розшифрування:

1. Головний ключ (закритий), що знаходиться на зовнішньому USB-накопичувачі підключається до облаштування дешифрування;

2. Із зашифрованого файлу прочитується зашифрований сеансовий ключ, отриманий в процесі шифрування файлу;

3. Цей зашифрований сеансовий ключ розшифровується за допомогою закритого ключа, що знаходиться в пам'яті зовнішнього USB-накопичувача;

4. За допомогою цього розшифрованого ключа відновлюється зашифрована інформація файлу.

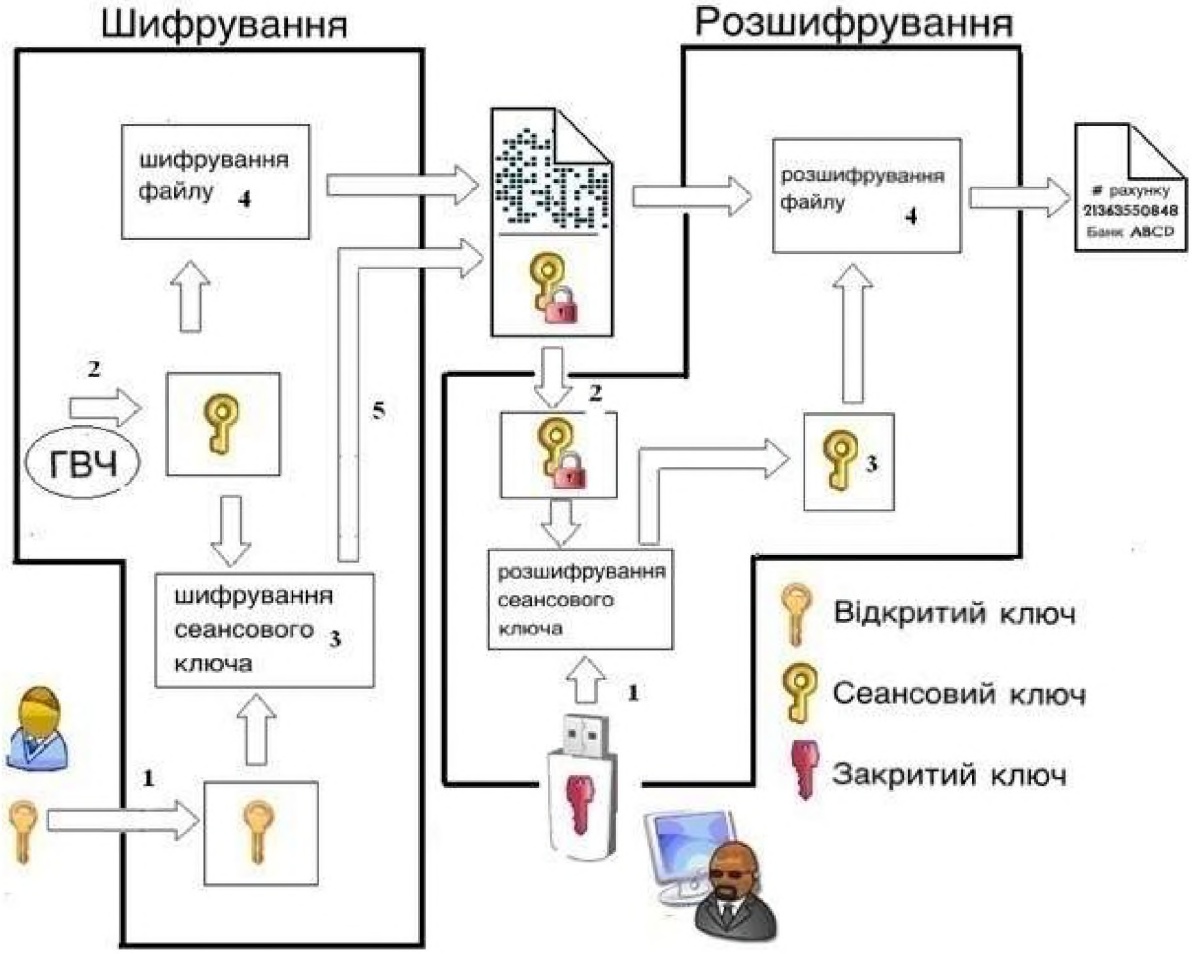

Рисунок 2.2 - Схема криптографічних перетворень

2.4 Використовувані алгоритми шифрування

Вибираючи алгоритм шифрування, який використовуватиметься в створюваній криптосистемі, враховувалися характеристики алгоритмів шифрування:

– *Криптостійкість.* Алгоритм має бути ретельно проаналізований світовою криптографічною спільнотою впродовж тривалого часу і визнаний криптостійким до різних видів атак;

– *Довжина ключа.* Ключ, використовуваний в алгоритмі шифрування, повинен визначатися вимогами безпеки і оцінками можливих атак;

– *Швидкість шифрування.* Передбачається взаємодія пристрою з комп'ютером через повношвидкісний інтерфейс USB2.0.(12 Мбіт/сік). Тому швидкість шифрування даних по вибраному алгоритму має бути настільки високою, щоб не виникало простоїв при передачі даних на максимальній швидкості.

– *Ресурсоємність.* Алгоритм має бути оптимізований для апаратної реалізації. Кількість оперативної пам'яті і необхідна продуктивність мікропроцесора повинні знаходитися у рамках, які обмежують мікроконтролери загального застосування.

2.4.1 Алгоритм шифрування файлів

Виходячи із структурної схеми застосування пристрою, що розробляється, при якій шифрується кожен символ, що набирається на клавіатурі, в цьому пристрої застосовується потоковий метод шифрування.

Потоковий шифр — це симетричний шифр, в якому кожен символ відкритого тексту перетвориться в символ шифрованого тексту в залежності не лише від використовуваного ключа, але і від його розташування в потоці відкритого тексту. Потоковий шифр реалізує інший підхід до симетричного шифрування, ніж блокові шифри.

Проста реалізація потокового шифру реалізується за допомогою генератора гамми, який видає ключовий потік (гамму):

$k_1, k_2, k_3, \ldots, k_L$ (див. рис. 1)

Позначимо потік бітів відкритого тексту як _. Тоді потік бітів шифротексту виходить за допомогою застосування операції складання по модулю 2:

$c_1, c_2, c_3, \ldots, c_L,$

де $c_i = m_i \oplus k_i. m_i = c_i \oplus k_i.$

Дешифрування робиться цією ж операцією між тією ж самою гаммою і зашифрованим текстом:

$m_i = c_i \oplus k_i.$

Потокові шифри підрозділяються на синхронні і самосинхронізуючі.

Синхронним називається потоковий шифр, в якому ключовий потік генерується незалежно від початкового тексту і шифротексту.

Процес шифрування за допомогою синхронного потокового шифру може бути описаний наступними рівняннями:

$\sigma_{i+1} = f(\sigma_i, k)$,

$z_i = g(\sigma_i, k)$,

$c_i = h(z_i, m_i)$,

де σ_0 — початковий стан, може бути визначено ключем k,

f — функція стану,

g — функція, що дає ключовий потік z_i,

h — вихідна функція, що комбінує ключовий потік і початковий текст m_i для отримання шифротексту c_i.

Властивості синхронних потокових шифрів:

Вимоги синхронізації. При використанні синхронних потокових шифрів відправник і одержувач мають бути синхронізовані — використовувати однакові ключі і обробляти однакові позиції (стани) в ключі — для забезпечення правильного розшифрування. При втраті синхронізації внаслідок вставки або видалення символу з шифротексту в процесі передачі розшифрування призводить до помилки, і відновлення можливо лише при застосуванні додаткових прийомів для ресинхронізації. Техніка ресинхронізації включає повторну ініціалізацію, переміщення спеціальних маркерів через певні інтервали шифротексту, або, у разі достатньої надмірності початкового тексту, випробування усіх можливих зміщень ключового потоку.

Відсутність поширення помилок. Символ шифротексту, який був змінений (але не видалений) в процесі передачі, не впливає на розшифрування інших символів шифротексту.

Активні атаки. Внаслідок попередньої властивості), вставка, видалення або відтворення символів шифротексту активним супротивником призводять до негайної втрати синхронізації і, отже, можуть бути виявлені

дешифрувальником. Внаслідок попередньої властивості, активний супротивник здатний змінити певні символи шифротексту і точно знать, що ці зміни вплинуть на початковий текст. Це показує, що для забезпечення автентифікації джерела даних і гарантії цілісності даних необхідно використати додаткові механізми.

Що самосинхронізованим, або асинхронним, називають потоковий шифр, в якому ключовий потік генерується як функція ключа і деякого числа попередніх символів:

$$\sigma_i = (c_{i-t}, c_{i-t+1}, \ldots, c_{i-1}),$$

$$z_i = g(\sigma_i, k),$$

$$c_i = h(z_i, m_i),$$

де $\sigma_0 = (c_{-t}, c_{-t+1}, \ldots, c_{-1})$ — (не секретний*) початковий стан,*

k — секретний ключ,

g — функція, що дає ключовий потік z_i,

h — вихідна функція, що комбінує ключовий потік

m_i — початковий текст

c_i — шифротекст.

Властивості потокових шифрів, що самосинхронізуються:

Самосинхронізація. Якщо символи шифротексту видалені або вставлені, можлива самосинхронізація, оскільки перетворення, що розшифровує, залежить лише від фіксованого числа попередніх символів шифротексту. Такі шифри здатні відновлювати правильність розшифрування після втрати синхронізації, так що лише фіксоване число символів початкового тексту не підлягає відновленню.

Обмежене поширення помилок. При зміні (або навіть видаленні або вставці) поодинокого символу в процесі передачі розшифрування не більше t подальших символів може бути невірною, після чого відновлюється правильний процес розшифрування.

Активні атаки. Попередня властивість має на увазі, що будь-яка зміна символу шифротексту активним супротивником приведе до того, що деякі інші символи шифротексту будуть розшифровані невірно, що підвищує (в порівнянні з синхронними потоковими шифрами) вірогідність виявлення атаки дешифрувальником. Внаслідок властивості (і), складніше (чим для синхронних потокових шифрів) виявити вставку, видалення або відтворення символів шифротексту активним супротивником. Це показує, що для забезпечення автентифікації джерела даних і гарантії цілісності даних необхідно використати додаткові механізми.

Дифузія статистики початкового тексту. Оскільки кожен символ початкового тексту робить вплив на увесь подальший шифротекст, статистичні властивості початкового тексту розсіюються по шифротексту. Отже, потокові шифри, що самосинхронізуються, можуть бути стійкішими, ніж синхронні потокові шифри, до атак, заснованих на надмірності початкового тексту.

Основна частина потокових шифрів, запропонованих до теперішнього часу в літературі, є адитивними потоковими шифрами.

Двійковий адитивний потоковий шифр — це синхронний потоковий шифр, в якому символи ключового потоку, початкового тексту і шифротексту є двійковими цифрами, і вихідна функція h — функція XOR.

Двійковий адитивний потоковий шифр показаний на рис. 2.3

Генератор потоку ключів створює бітовий потік, який схожий на випадковий, але насправді детермінований і може бути безпомилково відтворений при дешифруванні. Чим ближче вихід генератора потоку ключів до випадкового, тим більше часу буде потрібно криптоаналітику, щоб зламати шифр.

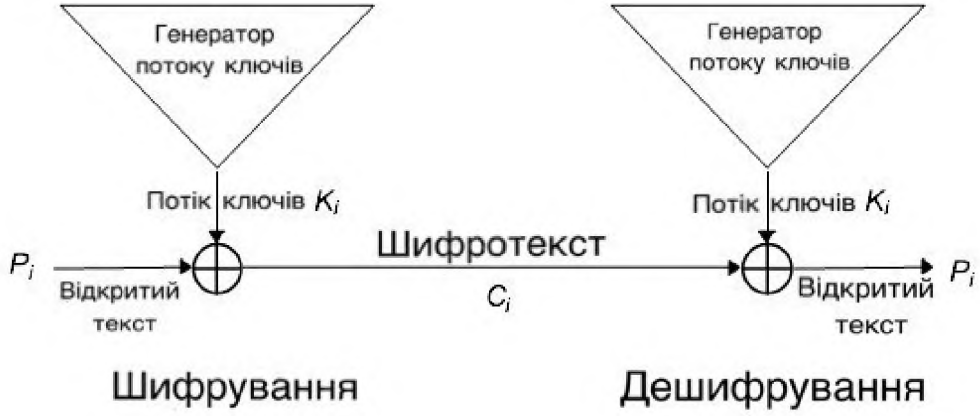

Рисунок 2.3 - Двійковий адитивний потоковий шифр

Проте, якщо генератор потоку ключів при кожному включенні створює один і той же бітовий потік, то його криптосистему зламати неважко. Покажемо на прикладі, чому це так.

Якщо до супротивника потрапив шифротекст і відповідний відкритий текст, то він, виконуючи операцію XOR над відкритим текстом і шифротекстом, розкриває потік ключів. Чи, якщо у нього є два різні шифротексти, зашифрованих однаковим ключем, він може виконати над ними операцію XOR, отримуючи два відкриті тексти повідомлень, над якими виконана операція XOR. Це неважко зламати, і потім він може отримати потік ключів, виконуючи операцію XOR над одним з відкритих текстів і шифротекстом. Тепер, перехопивши будь-яке інше шифроване повідомлення, він зможе розшифрувати його, використовуючи отриманий потік ключів. Крім того, він може розшифрувати і прочитати будь-яке з раніше перехоплених повідомлень. Коли супротивник отримає пару відкритий текст/шифротекст, він зможе читати усе.

Тому для усіх потокових шифрів використовуються ключі. Вихід генератора потоку ключів є функцією ключа. Тепер, якщо супротивник отримає пару відкритий текст/шифротекст, він зможе читати тільки ті повідомлення, які зашифровані тим же ключем. Генератор потоку ключів складається з трьох основних частин (рис. 2.4).

Внутрішній стан описує поточний стан генератора потоку ключів. Два генератори потоку ключів, з однаковим ключем і однаковим внутрішнім станом, видають однакові потоки ключів. Функція виходу по внутрішньому стану генерує біт потоку ключів. Функція наступного стану по внутрішньому стану генерує новий внутрішній стан.

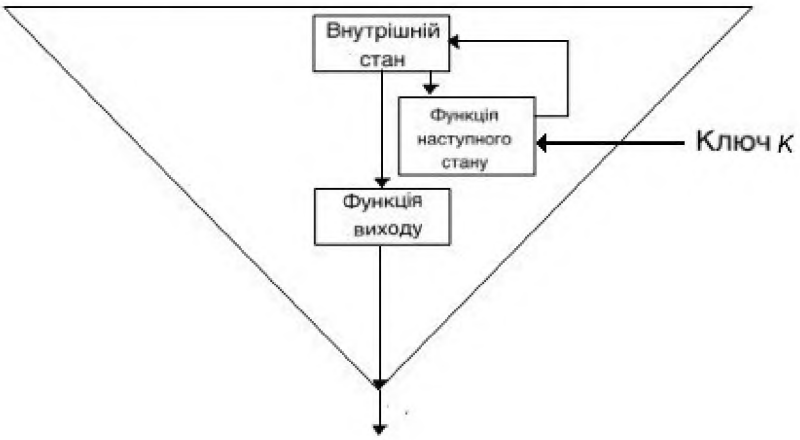

Рисунок 2.4 - Облаштування генератора потоку ключів

Для проєктованого пристрою вибраний синхронний потоковий шифр, а в якості його генератора ключового потоку лінійний зсувний регістр, початковий вектор ініціалізації якого визначається сеансовим ключем шифрування.

2.4.2 Алгоритм шифрування ключа

В якості алгоритму шифрування сеансового ключа вибраний алгоритм шифрування RSA.

RSA - криптографічна система відкритого ключа, що забезпечує такі механізми захисту як шифрування і цифровий підпис (автентифікація - встановлення достовірності). Криптосистема RSA розроблена в 1977 році і названа на честь її розробників Ronald Rivest, Adi Shamir і Leonard Adleman.

Алгоритм RSA працює таким чином: беруться два досить великих простих числа p і q і обчислюється їх добуток n = p*q; n називається модулем.

Потім вибирається число e, що задовольняє умові

1< e < (p - 1)*(q - 1) і що не має загальних дільників окрім 1 (взаємно просте) з числом (p - 1)*(q - 1).

Потім обчислюється число d таким чином, що (e*d - 1) ділиться на (p - 1)*(q – 1).

- e - відкритий (public) показник
- d - приватний (private) показник.
- (n; e) - відкритий (public) ключ
- (n; d). - приватний (private) ключ.

Дільники (чинники) p і q можна або знищити або зберегти разом з приватним (private) ключем.

Якби існували ефективні методи розкладання на співмножники, то, розклавши n на співмножники (чинники) p і q, можна було б отримати приватний (private) ключ d. Таким чином надійність криптосистеми RSA заснована на важковирішуваному - практично нерозв'язному - завданню розкладання n на співмножники (тобто на неможливості факторингу n) оскільки нині ефективного способу пошуку співмножників не існує.

Рекомендована довжина ключа

Розмір ключа в алгоритмі RSA пов'язаний з розміром модуля n. Два числа p і q, результатом яких є модуль n, повинні мати приблизно однакову довжину оскільки в цьому випадку знайти співмножники (чинники) складніше, ніж у разі коли довжина чисел значно розрізняється. Наприклад, якщо передбачається використати 768-бітовий модуль, то кожне число повинне мати довжину приблизно 384 біта. Але якщо два числа надзвичайно близькі один до одного або їх різниця близька до деякого зумовленого значення, то виникає потенційна загроза безпеки, проте така вірогідність - близькість двох випадково вибраних чисел - незначна.

1. Візьмемо M = (p+q)/2

2. При p < q, маємо 0 =< м - sqrt (n) =< (q - p)**2/8p.

Оскільки p = M*(+- sqrt (m2 - n)), то значення p і q можна легко знайти, якщо різниця p - q досить мала.

Оптимальний розмір модуля n визначається вимогами безпеки: модуль більшого розміру забезпечує велику безпеку, але і уповільнює роботу

алгоритму RSA. Довжина модуля вибирається в першу чергу на основі значущості даних, що захищаються, і необхідної стійкості захищених даних і в другу чергу - на основі оцінки можливих атак Нині Лабораторія RSA рекомендує для звичайних завдань ключі розміром 1024 біта, а для особливо важливих завдань - 2048 бітів (наприклад, для головного ключа Сертифікатів).

Вибрана довжина ключа 1024 біта.

У практичних застосуваннях для відкритого ключа зазвичай вибирається відносно невеликий показник, а часто групи користувачів використовують один і той же відкритий показник, але кожен з різним модулем. Якщо відкритий показник незмінний, то вводяться деякі обмеження на головні співмножники (чинники) модуля. При цьому шифрування даних йде швидше за розшифрування, а перевірка підпису швидша, ніж підписання.

Якщо k — кількість бітів в модулі, то в зазвичай використовуваних для RSA алгоритмах кількість кроків, необхідних для виконання операції з відкритим ключем, пропорційно другій мірі k, кількість кроків для операцій приватного ключа - третьої міри k, кількість кроків для операції створення ключів — четвертої міри k.

Виходячи із структурної схеми застосування пристрою, що розробляється, при якому цим методом шифрується тільки сеансовий ключ швидкість криптоперетворення не критична.

2.4.3 Алгоритм генерації псевдовипадкових чисел

При використанні як симетричних, так і асиметричних криптосистем необхідно мати хороше джерело випадкових чисел для створення ключів. Кращим варіантом є створення випадкових чисел на основі деякого фізичного процесу, оскільки багато фізичних процесів дійсно випадкові. Наприклад, для цього можна використати апаратні засоби типу "шумлячого" діода. Можна використати які-небудь фізичні рухи користувача, наприклад, швидкість друку в мікросекундах, переміщення миші і т.д.

Практично усі методи генерації випадкових чисел мають деяку кореляцію і не дозволяють забезпечити достатню статистичну хаотичність. Тому перед використанням, отримані ключі слід обробляти надійною хеш-функцією.

Інший підхід полягає в тому, щоб використати генератор псевдовипадкових чисел, що запускається випадковим числом. Оскільки генератори псевдовипадкових чисел є детермінованими алгоритмами, то важливо знайти серед них криптографічний захищений, а окрім цього використати хороше випадкове число для запуску генерації. У пристрої використовуватимуться випадкові числа, отримані від апаратури мікроконтролера і перетворюватися в 384, - бітове число за допомогою хеш-функції SHA2 - 384. Це необхідно робити для поліпшення рівномірності розподілу випадкових чисел.

Для отримання випадкових чисел, в пристрої використовуються 32-х розрядний лічильник, на вхід якого подається максимально можлива частота. При включенні пристрою, лічильник ініціалізувався значенням збереженим раніше в EEPROM. Оскільки операції шифрування файлів ініціюються користувачем у випадкові проміжки часу, на початку кожної такої операції вміст лічильника подається на вхід функції хешування. Отримане значення використовується як сеансовий ключ.

2.5 Вибіркове шифрування інформації файлу

Використання запропонованого способу криптографічного захисту дозволяє застосувати вибіркове шифрування інформації файлу, що складається в тому, що шифрується не уся інформація файлу, а тільки деякі, спеціально відмічені його фрагменти. В цьому випадку застосовується так звані дескриптори. Дескриптори - це спеціально підготовлені текстові фрагменти, які не шифруються і не розшифровуються, але вставляються в шифрований файл. Дескриптори описують ділянку шифрованого тексту файлу і підрозділяються на початкові і кінцеві.

Шифрування даних робиться тільки між цими двома дескрипторами, а їх установка в шифрований файл здійснюється програмою автоматично. Приклад вибіркового шифрування і можливих дескрипторів показаний на наступних фрагмента відкритого і шифрованого текстів.

Відкритий текст з дескрипторами ZZZ і QQQ

Орендна плата – **ZZZ 500 QQQ**

Реклама – **ZZZ 250 QQQ**

Заробітна плата штатних працівників – **ZZZ 234 QQQ**

Інші витрати – **ZZZ 400 QQQ**

Шифрувальний текст з дескрипторами ZZZ и QQQ

Орендна плата – **ZZZ ↥‡‡ QQQ**

Реклама – **ZZZ ↕↥‡ QQQ**

Заробітна плата штатних працівників – **ZZZ ◢△↕ QQQ**

Інші витрати – **ZZZ ↑↑↖ QQQ**

При розшифруванні такої інформації робиться тільки розшифрування символів між початковим і кінцевим дескрипторами, а самі вони не включаються в розшифрований текст.

2.6 Структурна схема пристрою

1. Пристрій робитиме шифрування даних з великою швидкістю (до 12 Мбіт/сік). Тому основою пристрою має бути високопродуктивний 32-х розрядний мікроконтролер.

2. Пристрій пов'язаний з комп'ютером через інтерфейс USB на швидкості 12 Мбіт/сек. Тому мікроконтролер, використовуваний в пристрої, має бути оснащений full - speed USB контролером.

Для вибору контролера складена таблиця основних характеристик найбільш відповідних сучасних типів контролерів - Atmel AT91SAM7S64, Atmel AT89C5131, Philips LPC2141, Philips LPC2142 (таблиця 2.2)

Таблиця 2.1 Характеристики сучасних типів контролерів

Мікроконтролер	Швидкодія, MIPS	Об'єм flash, Кб	Об'єм ОЗУ, Кб	Ціна, USD
AT91SAM7S64	50	64	16	5
AT89C5131	4	32	1	8
LPC2141	55	32	8	5
LPC2142	55	64	16	7

Виходячи з даних цієї таблиці для цього пристрою вибраний контролер Atmel AT91SAM7S64. Основні характеристики цього контролера приведені в додатку Б.

3. Для запобігання впливу на облаштування високочастотних перешкод з лінії зв'язку USB-інтерфейсу, до складу пристрою необхідно включити фільтр USB-сигналу.

4. Живлення пристрою забезпечується інтерфейсом USB. Для забезпечення надійної роботи апаратного шифратора, необхідно передбачити стабілізацію і, якщо необхідно, перетворення отриманого від USB напруги.

5. Необхідно передбачити індикацію подання живлення на пристрій і індикацію нормальної роботи пристрою.

6. Для генерації сеансових ключів шифрування в пристрої має бути реалізований апаратно-програмний генератор випадкових чисел.

7. В пристрої має знаходитись енергонезалежна EEPROM пам'ять даних для зберігання майстер-ключів.

Структурна схема запропонованого пристрою для апаратного шифрування інформації приведена на Рисунку 2.5.

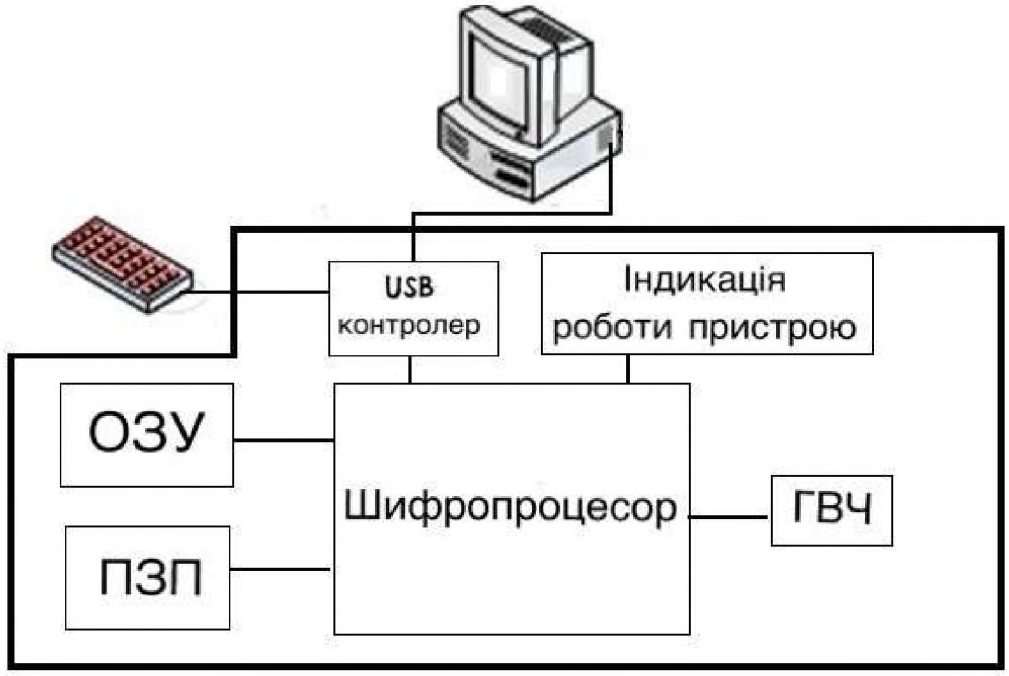

Рисунок 2.5 - Структурна схема пристрою для апаратного шифрування.

2.7 Підключення пристрою в лінію клавіатури

Клавіатура є одним з найважливіших пристроїв, що визначає умови роботи на комп'ютері. У неї вбудований контролер. Незалежно від того, як механічно реалізований процес натиснення клавіш, сигнал при натисненні клавіші реєструється контролером клавіатури (наприклад, 8049) і передається у вигляді так званого скен-коду на материнську плату.

Скен-код — це однобайтне число, перші 7 біт якого представляють ідентифікаційний номер, присвоєний кожній клавіші.

На материнській платі персонального комп'ютера для підключення клавіатури також використовується спеціальний контролер. Для персональних комп'ютерів типу AT зазвичай застосовується мікросхема типу UPI 8042.

Коли скен-код поступає в контролер клавіатури, то ініціалізується апаратне переривання (IRQ1), процесор припиняє свою роботу і виконує процедуру, що аналізує скен-код. Це переривання обслуговується спеціальною програмою, що входить до складу ROM BIOS. При зміні скен-коду від клавіш зрушення (<Alt>,<Ctrl>) або перемикача (<Shift>,

<CapsLock>) зміна статусу записується в оперативну пам'ять. У усіх інших випадках скен-код трансформується в код символу (так звані коди ASCII або розширені коди).

При цьому оброблювальна процедура спочатку визначає установку клавіш і перемикачів, щоб правильно отримати код, що вводиться. Потім введений код поміщається у буфер клавіатури, що є областю пам'яті, здатною запам'ятати до 15 символів, що вводяться, поки застосовна програма не може їх обробити. Буфер організований за принципом FIFO (перший увійшов — перший вийшов).

Кожна клавіша генерує два типи скен-коду "код натиснення", коли клавіша натискається, і "код звільнення", коли клавіша опускається. Для "кодів натиснення" і "кодів звільнення" використовується один і той же ланцюжок бітів, коди звільнення складаються з двох байтів, перший з яких завжди рівний 0F0H. Таким чином для того, щоб зімітувати натиснення клавіш потрібно усього лише ввести шифрований символ у буфер клавіатури. У AT - BIOS для цього існує функція 5 переривань 16H, опис якої знайти не складно.

Не набагато складніше безпосередньо працювати з буфером клавіатури.

Наведемо текст функції на мові Pascal для цієї мети

```
procedure StuffKey(W : Word);
  {-Stuff one key into the keyboard buffer}
 const
  KbdStart = $1E;
  KbdEnd = $3C;
 var
  KbdHead : Word absolute $40 : $1A;
  KbdTail : Word absolute $40 : $1C;
  SaveKbdTail : Word;
 begin
```

```
  SaveKbdTail := KbdTail;
  if KbdTail = KbdEnd then
    KbdTail := KbdStart
  else
    Inc(KbdTail, 2);
  if KbdTail = KbdHead then
    KbdTail := SaveKbdTail
  else
    MemW[$40:SaveKbdTail] := W;
end;
```

2.8 Висновок

У спеціальній частині кваліфікаційної роботи розроблено загальну схему і функціональну структуру апаратної системи криптоперетворення файлів; обґрунтована і вибрана схема криптоперетворення даних, створення і зберігання ключів шифрування; розроблені структурні схеми пристроїв для апаратного шифрування і дешифрування інформації; запропоновано схему вибіркового шифрування інформації.

3 ЕКОНОМІЧНИЙ РОЗДІЛ

Завданням даної роботи є підвищення безпеки підприємства за рахунок впровадження розробленого програмно-апаратного шифрування комп'ютерної інформації. У даному розділі були виконані наступні розрахунки:

1) розрахунок капітальних витрат;
2) розрахунок поточних витрат;
3) визначена величина можливого збитку;
4) визначені та проаналізовані показники економічної ефективності системи інформаційної безпеки.

На підставі отриманих результатів було зроблено висновок щодо економічної ефективності створення цього алгоритму.

3.1 Визначення трудомісткості розробки алгоритму

Трудомісткість створення алгоритму визначається тривалістю кожної робочої операції, починаючи зі складання технічного завдання і закінчуючи оформленням документації.

$$t = t_{тз} + t_в + t_а + t_{пр} + t_{опр} + t_д, \text{ год.},$$

де $t_{тз}$ = 20 год. – тривалість складання технічного завдання на розробку алгоритму;

$t_в$ = 6 год. – тривалість вивчення технічного завдання, літературних джерел за темою тощо;

$t_а$ = 6 год. – тривалість розробки блок-схеми алгоритму;

$t_{пр}$ = 4 год. – тривалість програмування за готовою блок-схемою;

$t_{опр}$ = 10 год. – тривалість опрацювання алгоритму;

$t_д$ = 4 год. – тривалість підготовки технічної документації.

t = 20 год. + 6 год. + 6 год. + 4 год. + 10 год. + 4 год. = 50 год.

3.2 Розрахунок витрат на створення алгоритму

Витрати на створення алгоритму Крп складаються з витрат на заробітну плату виконавця розробки Ззп і вартості витрат машинного часу, що необхідний для опрацювання алгоритму на ПК Змч:

$$Крп = Ззп + Змч,$$

де Крп – витрати на створення алгоритму;

Ззп – заробітна плата спеціаліста з інформаційної безпеки;

Змч – вартість витрат машинного часу, що необхідні для створення алгоритму.

Заробітна плата виконавця враховує основну і додаткову заробітну плату, а також відрахування на соціальне потреби (пенсійне страхування, страхування на випадок безробіття, соціальне страхування тощо) і визначається за формулою:

$$Ззп = t * Зіб = 50 * 125 = 6250 \text{ грн.}$$

де t – загальна тривалість розробки алгоритму, год.;

Зіб – середньогодинна заробітна плата спеціаліста з інформаційної безпеки становить 125 грн/год.

Вартість машинного часу для розробки алгоритму на ПК визначається за формулою:

$$Змч = t * Смч, \text{ грн.}$$

де t – трудомісткість розробки алгоритму на ПК, год.;

Смч – вартість 1 години машинного часу ПК, грн/год.

Вартість 1 години машинного часу ПК визначається за формулою:

$$Смч = Р * tнал * Се + \frac{Фзал * На}{Fр} + \frac{Клпз * Напз}{Fр} =$$

$$= 0{,}3 * 2 * 1{,}68 + \frac{(8000 * 0{,}5)}{1920} + \frac{6500 * 0{,}5}{1920} =$$

$$= 1{,}01 + 2{,}08 + 1{,}69 = 4{,}78 \text{ грн/год},$$

Де Р- встановлена потужність апаратури інформаційної безпеки, 0,3 кВт - середня потужність одного комп'ютера;

tнал – кількість машин на яких розроблюється політика безпеки;

Се – тариф на електричну енергію, 1,68 грн/кВт·год;

Фзал – залишкова вартість ПК на поточний рік, 8000 грн.;

На – річна норма амортизації на ПК, 0.5 частки одиниці;

Напз – річна норма амортизації на ліцензійне програмне забезпечення, 0,5 частки одиниці;

Клпз – вартість ліцензійного програмного забезпечення, 6500 грн.;

Fр – річний фонд робочого часу (за 40-годинного робочого тижня Fр = 1920 год.)

$$Змч = t * Смч = 50 * 4{,}78 = 239 \text{ грн.}$$

Визначена таким чином вартість створення алгоритму Крп є частиною одноразових капітальних витрат разом з витратами на придбання і налагодження апаратури системи інформаційної безпеки.

$$Крп = Ззп + Змч = 6250 + 239 = 6489 \text{ грн.}$$

3.3 Розрахунок (фіксованих) капітальних витрат

Капітальні (фіксовані) витрати на проєктування та впровадження проєктного варіанта системи інформаційної безпеки складають:

$$К = Кпр + Кзпз + Крп + Каз + Кнавч + Кн,$$

де Кпр – вартість розробки проєкту інформаційної безпеки та залучення для цього зовнішніх консультантів, 12000 грн.;

Кзпз – вартість закупівель ліцензійного основного й додаткового програмного забезпечення (ПЗ), 12000 грн.;

Крп – вартість розробки політики безпеки інформації, 6489 грн.;

Каз – вартість закупівель апаратного забезпечення та допоміжних матеріалів відсутня;

Кнавч - витрати на навчання технічних фахівців і обслуговуючого персоналу, 2000 грн.;

Кн – витрати на встановлення обладнання та налагодження системи інформаційної безпеки відсутні.

Відповідно до заданих даних розраховуємо капітальні витрати

$$К = Кпр + Кзпз + Крп + Каз + Кнавч + Кн =$$
$$= 12000 + 12000 + 6489 + 0 + 2000 + 0 = 32489 \text{ грн.}$$

3.4 Розрахунок поточних (експлуатаційних) витрат

Поточні витрати включають:

- навчання персоналу в питаннях інформаційної безпеки;
- витрати на керування системою інформаційної безпеки.

1. Витрати на навчання персоналу в питаннях інформаційної безпеки включають в себе послуги сторонніх організацій, що створюють політику безпеки інформації та відповідно до неї розробляють інструкції для персоналу, що є користувачами системи. Вартість навчання адміністративного персоналу й кінцевих користувачів розглянутої системи:

Со = 5000 грн. – витрати на навчання персоналу.

2. Обов'язки з керування системою інформаційної безпеки виконує керівник та адміністратор безпеки (за відсутності керівника), тому річний фонд заробітної плати складає додаткову заробітну плату директора та системного адміністратора за рік:

$$Сз = Зк + Заб = 1500 + 1000 = 2500 \text{ грн. (за 1 місяць)}$$

$$Сз = 2500 * 12 = 30000 \text{ грн. (за 1 рік)}$$

де Зк – додаткова заробітна плата керівника, 18000 грн. на рік.

Заб – додаткова заробітна плата адміністратора безпеки, 12000 грн. на рік.

Вартість електроенергії, що споживається апаратурою системою інформаційної безпеки протягом року (Се), визначається за формулою:

$$Се = Р * Fp * Це$$

де Р – встановлена потужність апаратури інформаційної безпеки (0,3 кВт*10 комп'ютерів = 3 кВт);

Fр = 12 міс * 20 робочих діб/міс * 8 робочих годин * 10 комп'ютерів = 19200 год – річний фонд робочого часу системи інформаційної безпеки;

Це = 1,68 грн за 1 кВт/год. – тариф на електроенергію на 01.01.2023 року.

$$Се = 3 * 19200 * 1,68 = 96768 \text{ грн.}$$

Витрати на технічне й організаційне адміністрування та сервіс системи інформаційної безпеки (Стос) визначаються у відсотках від вартості капітальних витрат (2%).

$$Стос = К * 0,02 = 32489 * 0,02 = 649,78 \text{ грн.}$$

Отже, річні поточні (експлуатаційні) витрати на функціонування системи інформаційної безпеки складають:

$$С = Со + Сз + Се + Стос =$$
$$= 5000 + 30000 + 96768 + 649{,}78 = 132417{,}78 \text{ грн.}$$

3.5 Розрахунок оцінки величини збитку

Втрати від зниження продуктивності співробітників атакованої системи мережі являють собою втрати їхньої заробітної плати за час простою внаслідок атаки (Пп).

Таблиця 3.1 Заробітні плати працівників за місяць

Посада	Розмір зар. плати	Кількість співробітників	Витрати на зар. плату на міс., грн
Директор	30000	1	30000
Секретар	20000	1	20000
Менеджер по роботі з клієнтам	20000	4	80000
Менеджер по роботі з персоналом	20000	1	20000
Головний бухгалтер	25000	1	25000
Юрист	25000	1	25000
Системний адміністратор	20000	1	20000
		Сума	220000

Місячний фонд робочого часу складає 160 годин. Річний – 1920 годин. Час простою внаслідок атаки tп = 4 год.

$$Пп = \left(\frac{Зс}{Fр}\right) * тп = \left(\frac{220000}{160}\right) * 4 = 5500 \text{ грн.}$$

Витрати на відновлення працездатності системи включають кілька складових:

Пви – витрати на повторне введення інформації, грн.;

Ппв – витрати на відновлення системи, грн.;

Пзч – вартість заміни частин системи, грн.

Витрати на повторне введення інформації розраховуються виходячи з розміру заробітної плати співробітників системи Зс, які зайняті повторним введенням втраченої інформації, з урахуванням необхідного для цього часу тви = 4 год.:

$$Пви = (220000/160) * 4 = 5500 \text{ грн.}$$

Витрати на відновлення системи визначаються часом відновлення після атаки тв = 4 год. і розміром середньогодинної заробітної плати адміністратора безпеки:

$$Ппв = (20000/125) * 4 = 640 \text{ грн.}$$

Витрати на відновлення працездатності системи:

$$Пв = Пви + Ппв + Пзч = 5500 + 640 + 2000 = 8140 \text{ грн.}$$

Пзч = 2000 грн. - вартість для витрат на заміну частин;

О = 1500000 грн. - обсяг чистого прибутку за рік.

Втрати від зниження працездатності атакованої системи:

$$V = \frac{О}{Fр} * (тп + тв + тви) = \frac{1500000}{1920} * (4 + 4 + 4) = 9375 \text{ грн.}$$

Fp – це річний фонд часу роботи, 1920 годин;

tп – 4 години простою після атаки;

tв – 4 години відновлення після атаки;

tви – 4 години повторного введення загубленої інформації під час атаки.

Таким чином, загальний збиток від атаки на інформаційну систему при реалізації загрози складе:

$$U = Пп + Пв + V = 5500 + 8140 + 9375 \ = 23015 \text{ грн.}$$

Таким чином, загальний збиток від атак на вузол або сегмент корпоративної мережі організації складе:

$$B = \sum_{i} \sum_{n} * U = 10 * 4 * 23015 \ = 920600 \text{ грн.}$$

де: i- число атакованих вузлів, 10 комп'ютери;

n – середнє число атак на рік, 4 рази.

Загальний ефект від впровадження системи інформаційної безпеки визначається з урахуванням B – загального збитку від атаки; R – очікуваної ймовірності атаки на систему; C – щорічних витрат на експлуатацію системи інформаційної безпеки.

Ймовірність R (0…1). Якщо реалізація загроз найймовірніша 1 раз на 3 місяці, тобто 4 рази на рік, то R=0,25.

Загальний ефект від впровадження політики безпеки:

$$E = B * R - C = 920600 * 0{,}25 - 132417{,}78 = 97732{,}22 \text{ грн.}$$

3.6 Визначення та аналіз показників економічної ефективності системи інформаційної безпеки

Оцінка економічної ефективності системи захисту інформації, розглянутої у спеціальній частині кваліфікаційної роботи, здійснюється на основі визначення та аналізу наступних показників:

- коефіцієнт повернення інвестицій (ROI). У сфері інформаційної безпеки йому відповідає показник ROSI (Return on Investment for Security);
- термін окупності капітальних інвестицій То.

Коефіцієнт повернення інвестицій ROSI показує, скільки гривень додаткового прибутку приносить одна гривня капітальних інвестицій на впровадження системи інформаційної безпеки.

Щодо інформаційної безпеки говорять не про прибуток, а про запобігання можливих втрат від атаки на сегмент або вузол корпоративної мережі.

Коефіцієнт ROSI розраховують за допомогою показників:

Е – загальний ефект від впровадження системи інформаційної безпеки тис. грн;

К – капітальні інвестиції за варіантами, що забезпечили цей ефект, тис. грн.

$$ROSI = E/K = 97732{,}22 / 32489 = 3$$

Термін окупності капітальних інвестицій показує, за скільки років інвестиції окупляться за рахунок загального ефекту від впровадження політики безпеки.

$$To = K/E = 1/ROSI = 1 / 3 = 0{,}33 \text{ року} = 4 \text{ місяці.}$$

3.7 Висновок

У даному розділі були проведені розрахунки витрат на проєкт системи захисту інформації з врахуванням впровадження розробленого програмно-апаратного шифрування комп'ютерної інформації.

В результаті отримано наступні дані:

- капітальні витрати на впровадження інформаційної політики безпеки становлять 32489 грн.;
- експлуатаційні витрати на впровадження інформаційної політики безпеки становлять 132417,78 грн.;
- загальний збиток від атаки на вузол складає 920600 грн.;
- ефект від впровадження системи інформаційної безпеки становить 97732,22 грн.

Отже, проєкт системи інформаційної безпеки є доцільним і економічно вигідним. Термін окупності капітальних інвестицій складає 4 місяці.

ВИСНОВКИ

У кваліфікаційній роботі було проаналізовані можливі види атак, що призводять до несанкціонованого доступу інформації, яка зберігається на комп'ютері. Проведено аналіз існуючих принципів і методів, що застосовуються в сучасних налаштуваннях апаратного шифрування файлів.

В спеціальній частині розроблено загальну схема і функціональну структуру апаратної системи криптоперетворювання файлів. Також обгрунтовано вибір схеми криптоперетворювання даних, створення і зберігання ключів шифрування; розроблено структурні схеми пристроїв для апаратного шифрування і дешифрування інформації; запропоновано схему вибіркового шифрування інформації.

Переваги роботи полягають у тому, що розшифровані дані не знаходяться в оперативній пам'яті комп'ютер, і тому не можуть бути перехоплені спеціальними програмними закладками. І за рахунок виділення операцій шифрування і дешифрування в окремі процедури, що вимагають застосування спеціального апаратного засобу, підвищується рівень контролю і обліку за використанням конфіденційної інформації.

Список використаної літератури

1 НД ТЗІ 3.7-003-05 "Порядок проведення робіт із створення комплексної системи захисту інформації в інформаційно-телекомунікаційній системі" [Електронний ресурс]. – 2005. – Режим доступу до ресурсу: http://www.dsszzi.gov.ua/control/uk/publish/article?art_id=46074.

2 Закон України "Про захист персональних даних" [Електронний ресурс]. – 2018. – Режим доступу до ресурсу: https://zakon.rada.gov.ua/laws/show/2297-17.

3 НД ТЗІ 2.5-004-99 "Критерії оцінки захищеності інформації в комп'ютерних системах від несанкціонованого доступу" [Електронний ресурс]. – 1999. – Режим доступу до ресурсу: www.dsszzi.gov.ua/dsszzi/doccatalog/.

4 НД ТЗІ 2.5-005-99 "Класифікація автоматизованих систем і стандартні функціональні профілі захищеності оброблюваної інформації від несанкціонованого доступу" [Електронний ресурс]. – 1999. – Режим доступу до ресурсу: http://www.dsszzi.gov.ua/dsszzi/control/uk/publish/.

5 НД ТЗІ 1.1-002-99 "Загальні положення щодо захисту інформації в комп'ютерних системах від несанкціонованого доступу" [Електронний ресурс]. – Режим доступу до ресурсу: www.dsszzi.gov.ua/dsszzi/.

6 Закон України "Про інформацію" [Електронний ресурс]. – 2017. – Режим доступу до ресурсу: https://zakon.rada.gov.ua/laws/show/2657-12.

7 Закон України "Про захист інформації в інформаційно-телекомунікаційних системах" [Електронний ресурс]. – 2014. – Режим доступу до ресурсу: https://zakon.rada.gov.ua/laws/show/80/94-%D0%B2%D1%80.

8 НД ТЗІ 1.1-003-99 "Термінологія в галузі захисту інформації в комп'ютерних системах від несанкціонованого доступу" [Електронний ресурс]. – Режим доступу до ресурсу: https://tzi.com.ua/downloads/1.1-003-99.pdf.

9 ДЕРЖАВНИЙ СТАНДАРТ УКРАЇНИ Захист інформації. Технічний захист інформації. Терміни та визначення. ДСТУ 3396.2-97 [Електронний ресурс]. – 1998. – Режим доступу до ресурсу: http://www.dsszzi.gov.ua/dsszzi/control/.

10 НД ТЗІ 1.4-001-2000 "Типове положення про службу захисту інформації в автоматизованій системі" [Електронний ресурс]. – 2012. – Режим доступу до ресурсу: http://www.dsszzi.gov.ua/dsszzi/control/uk/publish/.

11 ЕКСПЛУАТАЦІЯ СИСТЕМ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ [Електронний ресурс]. – 2013. – Режим доступу до ресурсу: https://lektsii.org/15-1903.html.

15 Крижанівський В. Б. КОНСПЕКТ ЛЕКЦІЙ з курсу «Безпека інформаційних систем» [Електронний ресурс] / В'ячеслав Борисович Крижанівський. – 2012. – Режим доступу до ресурсу: https://learn.ztu.edu.ua/mod/resource/view.php?id=201.

ДОДАТОК А. Відомість матеріалів кваліфікаційної роботи

№	Формат	Найменування	Кількість листів	Примітки
Документація				
1	А4	Реферат	2	
2	А4	Список умовних скорочень	1	
3	А4	Зміст	1	
4	А4	Вступ	2	
5	А4	Розділ 1	21	
6	А4	Розділ 2	26	
7	А4	Розділ 3	10	
8	А4	Висновки	1	
9	А4	Перелік посилань	2	
10	А4	Додаток А	1	
11	А4	Додаток Б	4	
12	А4	Додаток В	1	
13	А4	Додаток Г	1	
14	А4	Додаток Ґ	2	

ДОДАТОК Б. Технічні характеристики мікроконтролера AT91SAM7S64

Характеристики мікроконтролера:

– Містить ядро процесора ARM7TDMI® ARM® Thumb®;

– Високопродуктивна 32-разр. RISC -архітектура;

– Великий набір 16-разр. інструкцій;

– Вбудоване ядро внутрішньосхемної емуляції з налагоджувальним комунікаційним каналом;

– Внутрішня високошвидкісна флеш-пам’ять розміром 64 кбайт і організацією 512 сторінок по 128 байт в кожній;

– Однотактний доступ при частотах до 30 Мгц. Попереджуючий буфер оптимізує виконання Thumb-інструкцій при максимальній швидкодії;

– Час програмування сторінок : 4 мс, в т.ч. автоматичне стирання сторінки; час повного стирання: 10 мс;

– 10000 циклів запису, 10-річний термін зберігання даних, функції захисту секторів, біт захисту флеш-пам’яті;

– Інтерфейс швидкого програмування флеш-пам’яті для серійного виробництва;

– 16 кбайт внутрішнього високошвидкісного СОЗУ, однотактний доступ при максимальній швидкодії;

– Контролер пам'яті (MC);

– Вбудований контролер флеш-пам’яті, визначення некоректного доступу і формування статусу помилки;

Контролер скидання (RSTC)

– Складається зі схеми скидання при поданні живлення і схеми детектора зниження напруги живлення з відкаліброваним в заводських умовах порогом;

– Виконує обробку зовнішнього сигналу скидання і формує інформацію про джерело скидання;

Тактовий генератор (CKGR)

– Малопотужний RC-генератор, вбудований генератор частот від 3 до 20 Мгц;

– Одна схема ФАПЧ;

– Контролер управління енергоспоживанням (PMC);

– Можливість програмної оптимізації енергоспоживання, в т.ч. з використанням режимів зниженої швидкодії (Slow Clock), можливе зниження частоти до 500 Гц) і режим холостого ходу (Idle);

– Три програмованих зовнішніх тактових сигналу;

– Вдосконалений контролер переривань (AIC);

– Індивідуальне маскування, вісім рівнів пріоритетів, векторизовані джерела переривань;

– Два зовнішні джерела переривання + одне зовнішнє джерело переривання зі швидким реагуванням, захист від неправдивих переривань.

Блок відладки (DBGU);

– 2-пров. УАПП + підтримка переривання по налагоджувальному комунікаційному каналу, програмоване запобігання доступу з боку внутрішньосхемного емулятора;

– Інтервальний таймер (PIT);

– 20-разр. програмований лічильник + 12 разр. лічильник інтервалів;

– Сторожовий таймер (WDT);

– 12-разр. програмований лічильник із захистом ключа;

– Виконує скидання або генерує запит на переривання системи;

– Лічильник може бути зупинений, коли процесор знаходиться в стані відладки або в режимі холостого ходу.

Таймер реального часу (RTT)

- 32-разр. циклічний лічильник з сигналізатором;

- Працює від внутрішнього RC-генератора;

- Один контролер паралельного введення/виводу (PIOA)
- 42 програмовані лінії введення-виводу, мультиплексові з двома вбудованими периферійними модулями;
- Можливість генерації переривання по зміні на вході будь-якої лінії введення виводу;
- Індивідуально програмовані відкритий стік, що підтягує резистор і синхронізований вихід;
- 11 канальний контролер периферійних даних (PDC);
- Один повношвидкісний контролер USB 2.0 (12 Мбіт/сік), режим пристрою;
- Вбудований трансивер, вбудовані буфери FIFO, що конфігуруються, місткістю 328 байт кожен;
- Один синхронний послідовний контролер (SSC);
- Окремі синхронізація і сигнали синхронізації кадру у кожного приймача і передавача;
- Підтримка аналогового інтерфейсу I2S, підтримка тимчасового ущільнення;
- Можливість високошвидкісної безперервної передачі потоку даних в 32-разр. форматі;
- Два універсальних синхронних/асинхронних приймача (УСАПП);
- Роздільні генератори швидкості зв'язку, інфрачервона модуляція/демодуляція (IrDA);
- Підтримка смарт-карт ISO7816 T0/T1, апаратне підтвердження зв'язку, підтримка RS485;
- Повний інтерфейс модему на УСАПП1;
- Послідовний периферійний інтерфейс SPI з режимами ведучий/підлеглий;
- Програмована довжина даних від 8 до 16 біт, чотири зовнішні виходи вибору мікросхем;
- Один трьохканальний 16-разр. таймер-лічильник (TC);

- Три зовнішні тактові входи, дві лінії універсального введення-виводу на кожен канал;

- Два ШИМ-генератори, режим захоплення і генерації імпульсів, можливість реверсування рахунку;

- Один чотирьохканальний 16-разр. ШИМ-контролер (PWMC);

- Один двопровідний інтерфейс (TWI);

- Працює тільки в режимі ведучого, підтримуються усі двопровідні ЕСППЗУ фірми Atmel;

- Один 8-канальний 10-разр. аналогово-цифровий перетворювач, чотири канали мультиплексовані з лініями цифрового введення-виводу;

- Граничне сканування усіх цифрових ліній відповідно до стандарту IEEE 1149.1 через інтерфейс JTAG;

- Лінії введення-виводу сумісні 5В рівнями і мають підвищену здатність навантаження, до 16 мА кожна.

Джерела живлення

- Вбудований стабілізатор напруги 1,8 В із здатністю навантаження до 100 мА для живлення ядра і зовнішніх компонентів;

- Напруга живлення введення-виведення VDDIO = 1,8В або 3,3В, окреме живлення флеш-пам'яті VDDFLASH = 3,3В;

- Напруга живлення ядра VDDCORE = 1,8В (з детектором пониження напруги);

- Напруга живленні аналогової схеми VDDANA = 3,3В;

- Статична робота на частотах до 55 Мгц за найгірших умов роботи : напруга живлення 1,65 В, температура 85°С.

ДОДАТОК В. Перелік документів на оптичному носії

1 Презентація_Павленко.ppt

2 Кваліфікаційна робота_Павленко.doc

ДОДАТОК Г. Відгук керівника економічного розділу

Керівник розділу _____ _____

(підпис) (прізвище, ініціали)

ДОДАТОК Ґ. Відгук керівника кваліфікаційної роботи

В І Д Г У К

на кваліфікаційну роботу студента групи 125м-22-2 Павленка Є.С.

на тему: «Розробка системи програмно-апаратного шифрування комп'ютерної інформації»

Пояснювальна записка містить: 77 с., 5 рис., 2 табл., 5 додатків, 15 джерел.

Метою кваліфікаційної роботи є зменшення загрози порушення конфіденційності інформації в комп'ютерній системі і програмно-апаратній системи криптографічного перетворення формованих даних.

В першій частині проаналізовано можливі види атак, що призводять до несанкціонованого доступу до інформації; проаналізовано існуючі принципи і методи, що застосовуються в сучасних налаштуваннях апаратного шифрування файлів.

У спеціальній частині приведена схема криптографічного перетворення даних і ключів шифрування, показана функціональна і структурні схеми апаратної системи вибіркового шифрування інформації.

В економічній частині виконано розрахунок витрат на розробку і впровадження алгоритму на підприємстві. Також зроблений висновок щодо економічної ефективності впровадження створеного алгоритму.

Практичне значення полягає у підвищенні рівня безпеки за рахунок зменшення загрози порушення конфіденційності інформації в комп'ютерній системі.

Наукова новизна полягає в удосконаленні системи інформаційної безпеки за допомогою створення схеми криптографічного перетворення даних.

Серед недоліків роботи слід відзначити: недостатньо глибоке опрацювання теми, незначні відхилення від стандартів при оформленні.

Рівень запозичень у кваліфікаційній роботі відповідає вимогам «Положення про систему виявлення та запобігання плагіату».

В цілому робота задовольняє усім вимогам, а її автор Павленко Є.С. заслуговує на оцінку « » та присвоєння кваліфікації «Магістр з кібербезпеки» за спеціальністю 125 Кібербезпека.

Керівник роботи,
к.т.н., доц. Мацюк С.М.