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

Інститут електроенергетики
Факультет інформаційних технологій
Кафедра безпеки інформації та телекомунікацій

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеню магістра

студента **Сидоренко Дар'ї Миколаївна**

академічної групи **125м-17-1**

спеціальності **125 Кібербезпека**

спеціалізації

за освітньо-професійною програмою **Кібербезпека**

на тему ***Створення сертифікованої аутентифікації відкритих ключів у спеціальних мережах***

Керівники	**Прізвище, ініціали**	**Оцінка за шкалою**		**Підпис**
		Рейтин-говою	**інститу-ційною**	
кваліфікаційної роботи	д.т.н., проф. Корнієнко В.І.			
розділів:				
спеціальний	ст..викл. Святошенко В.О.			
економічний	к.е.н., доц. Пілова Д.П.			

Рецензент				

Нормоконтролер	ст. викл. Мєшков В.І.			

Дніпро
2018

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

«_____»_____20___року

ЗАВДАННЯ
на кваліфікаційну роботу ступеня магістра

студенту **Сидоренко Д.М.** **академічної групи** **125м-17-2**
(прізвище та ініціали) (шифр)

спеціальності ***125 Кібербезпека***

спеціалізації[1] ***Інформаційні технології***

за освітньо-професійною програмою ***Кібербезпека***

на тему ***Створення сертифікованої аутентифікації відкритих ключів у спеціальних мережах***

1 ПІДСТАВИ ДЛЯ ПРОВЕДЕННЯ РОБОТИ

Наказ ректора НТУ «Дніпровська політехніка» від ___29.11.18_____ № _2025-л__

2 МЕТА ТА ВИХІДНІ ДАНІ ДЛЯ ПРОВЕДЕННЯ РОБІТ

Об'єкт досліджень ***Безпека данних у спеціальні мережі Ad-hoc***

Предмет досліджень ***Необхідність захисту данних при роботі у спеціальних мережах***

Мета ***Зменшити вразливість мережі до атак на мережу за допомогою сертифікату аутентифікації***

Вихідні дані для проведення роботи ***Інформація, що передається по мережі та їх хеш-функції***

3 ОЧІКУВАНІ РЕЗУЛЬТАТИ

Наукова новизна ***Вирішенні задачі забезпечення захищеності данних під час їх передачі та обрабки у спеціальних мережах за допомогою створення сертифікації аутентифікації відкритих ключів***

Практична цінність ***розкриття практичного застосування створення сертифікації для публічних та приватних ключів у спеціальних мережах***

4 ВИМОГИ ДО РЕЗУЛЬТАТІВ ВИКОНАННЯ РОБОТИ

Зменшення вразливості спеціальних мереж до зовнішніх атак

5 ЕТАПИ ВИКОНАННЯ РОБІТ

Найменування етапів робіт	**Строки виконання робіт** (початок-кінець)
Огляд джерел за темою та напрям досліджень	03.09.18-06.10.18
Методи досліджень	07.10.18-31.10.18
Результати досліджень	01.11.18-24.11.18
Виконання економічного розділу	25.11.18-04.12.18
Оформлення пояснювальної записки	05.12.18-10.12.18

6 РЕАЛІЗАЦІЯ РЕЗУЛЬТАТІВ ТА ЕФЕКТИВНІСТЬ

Економічний ефект ***Зменшення економічних втрат після зовнішніх атак на мережу.***

Соціальний ефект ***Розповсюдження застосування спеціальних мереж. Захист персональних данних користувачів мережі***

7 ДОДАТКОВІ ВИМОГИ

Завдання видано _____ Корнієнко В.І.
(підпис керівника) (прізвище, ініціали)

Дата видачі: 03.09.18р.

Дата подання до екзаменаційної комісії: 14.12.18р.

Прийнято до виконання _____ Сидоренко Д.М
(підпис студента) (прізвище, ініціали)

РЕФРАТ

Пояснювальна записка: 67 с., 12 рис., 4 таб., 1 додатків, 22 джерел

Об’єкт дослідження: Безпека дпнних у спеціальні мережі Ad-hoc.

Предмет дослідження: Необхідність захисту данних при роботі у спеціальних мережах.

Мета дипломної роботи: Зменшити вразливість мережі до атак на мережу за допомогою сертифікату аутентифікації.

В першому розділі розглядається необхідність створення аутентифікації для роботи у спеціальних мережах.

В спеціальній частині описується створення сертифікату аутентифікації

В економічній частині розраховується економічні показники продуктивності запровадження аунтифікації

Наукова новизна полягає в вирішенні задачі забезпечення захищеності данних під час їх передачі та обрабки у спеціальних мережах за допомогою створення сертифікації аутентифікації відкритих ключів

КРИПТОГРАФІЧНІ СИСТЕМИ ВІДКРИТИХ КЛЮЧІВ, ХЕШ ПАРИ, АУТЕНТИФІКАЦІЯ, ПРИВАТНІ ТА ПУБЛІЧНІ КЛЮЧИ, БЕЗПРОВОДНІ МЕРЕЖІ

РЕФЕРАТ

Пояснительная записка: 67 с., 12 рис., 4 табл., 1 приложение, 22 источника

Объект исследования: Безопасность дпнних в специальные сети Ad-hoc.

Предмет исследования: Необходимость защиты данных при работе в специальных сетях.

Цель дипломной работы: Уменьшить уязвимость сети к атакам на сеть с помощью сертификата аутентификации.

В первой главе рассматривается необходимость создания аутентификации для работы в специальных сетях.

В специальной части описывается создание сертификата аутентификации

В экономической части рассчитывается экономические показатели производительности введения аунтификации

Научная новизна заключается в решении задачи обеспечения защищенности данных при их передаче и обрабкы в специальных сетях посредством создания сертификации аутентификации открытых ключей

КРИПТОГРАФИЧЕСКАЯ СИСТЕМА ОТКРЫТЫХ КЛЮЧЕЙ, ХЭШ ПАРЫ, АУТЕНТИФИКАЦИИ, И ОТКРЫТЫЕ КЛЮЧИ, БЕСПРОВОДНЫЕ СЕТЕВЫЕ

ABSTRACT

Explanatory note: 67 p., 12 figures, 4 tables, 1 annex, 22 sources

Object of research: Safety of secondary schools in special ad-hoc networks.

Subject of research: The need to protect data when working in special networks.

The purpose of the thesis: Reduce the vulnerability of the network to attacks on the network using an authentication certificate.

The first section discusses the need to create authentication for work in special networks.

The special section describes the creation of an authentication certificate

In the economic part, the economic performance indicators of the introduction of authentication are calculated

The scientific novelty consists in solving the problem of ensuring the security of data during their transmission and processing in special networks by creating a certificate of authentication of public keys

CRYPTOGRAPHIC SYSTEMS OF OPEN KEYS, HESH PATHS, AUTHENTICATION, PRIVATE AND PUBLIC KEYS, SECURITY NETWORKS

СПИСОК УМОВНИХ СКОРОЧЕНЬ

CA –орган сертифікацій
CRL – список відкликання сертифікатів
OCSP – протокол статусу онлайн-сертифікату
МШП – шифрування на основі ідентичності
ІБ – інформаційна безпека
ПК – особистий комп'ютер
PKG – енератор приватних ключів
ЦС – центр сертификации
PKI – інфраструктура відкритого ключа
CRL – список відкликання сертифікатів
ПП – програмний продукт
МШП – шифрування на основі ідентичності

ЗМІСТ

ВСТУП

Існує два основних типи з'єднання, Ad-Hoc і Інфраструктурі. Ad-Hoc використовується для простого з'єднання комп'ютерів між собою за методом "точка-точка". Для організації подібної мережі потрібно мінімум обладнання - досить, щоб кожен комп'ютер був обладнаний контролером Wi-Fi.Такой тип з'єднання може використовуватися для підключення до восьми комп'ютерів у однорангові з'єднання, де кожен комп'ютер буде пов'язаний з іншим. Але насправді, його варто використовувати для з'єднання в мережу двох, максимум - трьох комп'ютерів. Більша кількість комп'ютерів об'єднувати за цією схемою непрактично і незручно. Наприклад, щоб гість отримав доступ до глобальної мережі, вам буде потрібно постійно тримати включеним комп'ютер, підключений до інтернету.

Тип "інфраструктура" вимагає наявності точки доступу, яку ви можете використовувати і як маршрутизатор, якщо вам буде потрібно з'єднувати між собою мережі і ділити на всіх бездротових користувачів з'єднання з інтернетом. У найпростішому ж випадку при використанні точки доступу, ви отримуєте компактний пристрій, який можна підключити безпосередньо до Internet-каналу (Ethernet або ADSL кабелю) і встановити в зручному для вас місці. Наприклад, в центрі будинку, квартири або офісу, де немає можливості встановити комп'ютер, щоб рівномірно покрити сигналом всі приміщення. Тепер ви не прив'язані до вашого компьютеру.Но ваш комп'ютер підключений до інтернету так само через точку доступу або через маршрутизатор. Плюси цього підключення ми вже позначили: спрощене підключення більшого числа клієнтів, зручність фізичного розташування точки доступу, немає потреби використовувати один з комп'ютерів в якості шлюзу в інтернет.

Створення безпечних каналів зв'язку між двома вузлами вимагає, щоб вони були взаємно автентичними. Коли вузли, які потребують безпечної комунікації, не розпізнають один одного і не ділять секретних ключів, можуть бути використані пари публічних/приватних ключів за умови

існування інфраструктури, що дозволяє вузлам видавати сертифікати один одному і бути отримувати сповіщення, коли сертифікат був скасований. Однак це припущення не виконується в спеціальних мережах.

Спеціальні мережі - це бездротові мережі, які не покладаються на будь-яку інфраструктуру. Вони складаються з бездротових вузлів, вільних для переміщення, що спричиняє часті зміни топології мережі; мережеві розділи можуть навіть виникнути. Це означає, що неможливо забезпечити підключення до центрального підрозділу, такого як орган сертифікації (CA), репозиторію списків скасування сертифікатів (CRL) або рецензент протоколу статусу онлайн-сертифіката (OCSP).

Суть самоорганізованих мереж - надання абоненту можливості доступу до різних мережевих послуг за допомогою передачі та прийому «свого» трафіку через сусідніх абонентів.

Структура найпростішої самоорганізується мережі вдає із себе велику кількість абонентів на деякій площі, яку спрощено можна назвати площею покриття мережі, і одну або кілька точок доступу до зовнішніх мереж. Кожне з абонентських пристроїв, в залежності від його потужності, має свій радіусом дії. Якщо абонент, перебуваючи «на периферії» посилає пакет абоненту, що знаходиться в центрі мережі або на точку доступу, відбувається так званий многоскачковий процес передачі пакета через вузли, що знаходяться на шляху заздалегідь прокладеного маршруту. Таким чином можна сказати, що кожен новий абонент за рахунок своїх ресурсів збільшує радіус дії мережі. Отже, потужність кожного окремого пристрою може бути мінімальною. А це передбачає як менші вартості абонентських пристроїв, так і кращі показники безпеки і електромагнітної сумісності.

Спеціальні мережі також самоорганізуються, що означає, що вони працюють без центральних адміністративних повноважень. Як наслідок, жодна частина мережі не призначена для надання конкретних послуг. Тому служби безпеки, що надаються виділеними організаціями, такими як CA, репозиторії CRL та ОССП, повинні надаватися самими вузлами.

• Перевірка ідентифікації: вузли повинні мати можливість самостійно перевіряти, що суб'єкт, який запитує сертифікат для певної ідентичності, є законним власником цієї ідентичності. Цю задачу традиційно виконує CA. Проте в спеціальній мережі не гарантується, що CA завжди є доступним. Крім того, немає гарантії, що в мережі присутні лише вузли, ідентичність яких відомо заздалегідь, оскільки жодна інстанція, відповідальна за членство в мережі, завжди недосяжна. Отже, в мережах adhoc майже неможливо гарантувати, що вузли завжди можуть генерувати сертифікати, що дозволяють встановлювати безпечні канали зв'язку з потрібним об'єктом.

• Скасування сертифікату: вузли повинні мати змогу визначити, чи сертифікат був скасований, чи ні, тобто сертифікат все ще являє собою дійсне обов'язкове з'єднання між ідентифікатором та відкритим ключем. Ця дія повинна бути можлива навіть тоді, коли в мережі відсутній вузол, який містить поточну інформацію стосовно статусу анулювання сертифікату. Без будь-яких способів визначити там статус сертифіката, вузли в спеціальних мережах ніколи не можуть бути впевнені, чи взаємодіють вони з передбачуваним об'єктом, або з зловмисником, який використовує сертифікат, який був раніше скасований у недоступну частину ad hoc мережі або в іншій мережі.

• Компроміс вузла: нарешті, оскільки вузли в спеціальних мережах є мобільними пристроями, вони менш фізично захищені від "не мобільних пристроїв", які можуть бути заблоковані. Це може дозволити зловмисникові скомпрометувати пару публічних/приватних ключів, що зберігаються на вузлі, просто отримавши контроль над цим вузлом.

У роботі приймається додатковий підхід до визначення рішення, яке дозволяє вузлам в adhoc-мережах створювати по запиту декілька паролів державних/приватних ключових слів і сертифікатів, не спираючись на своїх сусідів, без необхідності створювати та зберігати заздалегідь усі ключі та сертифікати, які вони можуть використовувати в майбутньому. Натомість наше рішення покладається на унікальний сертифікат, виданий AC під час

ініціалізації, що підтверджує автентичність всіх відкритих ключів, які створюються пізніше. Початковий сертифікат не містить відкритого ключа. Він зв'язує ідентифікацію вузла, підтверджену АС у фіксованій мережі під час ініціалізації, до значення коду хешу. Тоді, коли вузол активний в спеціальній мережі, він генерує відкритий ключ, автентичність якого можна перевірити, перевіряючи, чи він правильно пов'язаний з хеш-кодом, що міститься в сертифікаті. Тільки той вузол, для якого сертифікат був виданий сертифікатом під час ініціалізації, може генерувати дійсну пару публічних/приватних ключів, які правильно пов'язані з сертифікованим хеш-кодом. Рішення покладається на нову схему, яка дозволяє перевіряється зв'язування декількох пар державних/приватних ключів з одним хеш-кодом, які існуючі рішення не. Визначене рішення також виріує проблему існування адвока в мережах використовуючи пари публічних/приватних ключів з коротким терміном служби. Нарешті, наше рішення дозволяє уникнути зловмисників доступу до загальнодоступних/приватних пар ключів, що зберігаються на вузлах без фізичного захисту та без будь-яких захищених пристроїв, таких як захист від несанкціонованого доступу. Це досягається за рахунок взаємодії з користувачами для створення ключових пар.

РОЗДІЛ 1. СТАН ПИТАННЯ ТА ПОСТАНОВКА ЗАДАЧ

1.1. Стан питання

Децентралізована природа безпровідних ad-hoc мереж робить їх корисними для широкої сфери застосувань де центральні вузли можуть бути не надійними і така архітектура може покращити масштабованість мережі в порівнянні з керованими безпровідними мережами, оскільки теоретичного і практичного обмеження загальної ємності таких мереж виявлено не було.

Мінімальні потреби в конфігурації і швидке розгортування робить ad hoc мережі придатними при використанні в екстрених умовах, таких як стихійні лиха і військові конфлікти. Наявність динамічних і адаптивних протоколів маршрутизації дозволяє швидко організовувати однорангові мережі.

Джерела вразливостей в бездротових самоорганізованих мережах

- Уразливість каналів до прослуховування і підміни повідомлень, у зв'язку з загальною доступністю середовища передачі, як і в будь-яких бездротових мережах.
- Незахищеність вузлів від зловмисника, який легко може отримати один вузол в розпорядження, оскільки зазвичай вони не перебувають у безпечних місцях, таких як сейфи.
- Відсутність інфраструктури робить класичні системи безпеки, такі як центри сертифікації та центральні сервери, непридатними.
- Топологія, яка динамічно змінюється вимагає використання складних алгоритмів маршрутизації, що враховують ймовірність появи некоректної інформації від скомпрометованих сайтів або в результаті зміни топології.

Відомі методи підвищення безпеки

- Доступність серверів через деяку кількість вузлів

Доступність серверів, що надають сервіси, які забезпечують безпеку в мережі, такі як центри сертифікації, через деяку кількість вузлів, забезпечить

доступність сервісу навіть у разі, коли невелика частина цих вузлів буде скомпрометована.

- Схема поділу секрету

Для забезпечення стійкості до збоїв серверів, зазвичай застосовують такі механізми, як репліковані сервера і кворумні системи, але вони підвищують ймовірність розкриття секрету, внаслідок компрометації одного з серверів. Схема розподілу секрету між серверами бореться з цією проблемою таким чином, що секрет може бути відновлений тільки у випадку, якщо достатня кількість часток секрету буде отримана з серверів. Для цих цілей можна використовувати розподілену криптосистему[1].

- Оновлення часткою секрету

Розділення секрету не захищає від зловмисника, який пересувається від сервера до сервера, атакуючи, компрометуючи і контролюючи їх, так як через деякий час він зможе зібрати достатню кількість інформації, щоб відновити секрет. Створення серверами нового, незалежного набору часток і заміна ними старого, рятує ситуацію, оскільки нові частки не можуть бути зіставлені зі старими для розкриття секрету. Для розкриття секрету зловмисникові потрібно буде скомпрометувати достатню кількість серверів між двома послідовними оновленнями часток.

Оновлення часток може бути узагальнене і на той випадок, коли нові частки будуть розподілені між іншим набором серверів і, можливо, навіть з іншою конфігурацією. Це узагальнення дозволяє сервісу адаптуватися до випадків, коли певні сервера скомпрометовані перманентно, або коли сервіс виявляється в більш недружній обстановці.

- Підтримка різних шляхів

Для підтримки досяжності слід знаходити і підтримувати різні шляху між двома вузлами, щоб невелика кількість скомпрометованих сайтів не змогла підірвати всі шляхи.

- Виявлення маршрутів шляхом передачі повідомлень

Навіть захищений протокол виявлення маршрутів, який запобігає спробі обману скомпрометованими вузлами, нічого не зможе вдіяти, якщо скомпрометовані вузли скооперуються під час виявлення маршрутів, однак при передачі повідомлень обчислити їх через їх некоректну роботу буде легко.

- Схема ймовірнісної безпечної маршрутизації

Ця схема полягає в тому, що для кожного призначення вузол підтримує імовірнісний розподіл по всіх сусідах. Цей розподіл засновано на відносній ймовірності того, що даний сусід передасть і в кінцевому рахунку доставить повідомлення до адресата. На кожному хопі повідомлення передається конкретному сусідові з якоюсь ймовірністю, грунтуючись на ймовірнісному розподілі досяжності сайту. Таким чином підтримуються різні шляхи. Також передача повідомлень сама по собі забезпечує відгук для коригування імовірнісного розподілу. Наприклад, підписане підтвердження про доставку повідомлення буде позитивним відгуком про досяжності по шляху, по якому його відправляли. Таким чином схема є самокорегованою.

Автентифіка́ція — процедура встановлення належності користувачеві інформації в системі пред'явленого ним ідентифікатора.

З позицій інформаційної безпеки Автентифікація є частиною процедури надання доступу для роботи в інформаційній системі, наступною після ідентифікації і передує авторизації.

1.2.Мета роботи

Мета роботи полягає у визначенні схеми, яка дозволяє вузлам створювати по запиту серії пар державних/приватних ключів, автентичність яких можна перевірити за допомогою єдиного сертифіката, виданого заздалегідь CA.

Для досягнення вищезазначеного, підтвердження тотожності вузлів здійснюється один раз CA з фіксованою мережею під час фази ініціалізації. Після успішної перевірки особистих даних CA видає сертифікат на вузол. З часом вузлам не потрібно взаємодіяти з CA чи будь-яким третім суб'єктом

для створення або перевірки відкритих ключів. Після, розрізняються рішення з раніше запропонованих, які керують процесом підтвердження ідентичності в фіксованій мережі. Але в цих попередніх рішеннях вузли повинні мати доступ до третьої сторони, яка може надіслати оновлену інформацію про статус анулювання, щоб мати змогу перевіряти відкриті ключі.

Проте, така ціль не завжди досяжна. Для обробки скасування використовуються пари ключових слів, які дійсні лише протягом короткого періоду часу. Але тільки законний власник сертифіката може генерувати дійсну пару ключів, які правильно пов'язані з цим сертифікатом протягом його періоду дії. Оскільки пари ключових слів стають автоматично недійсними через короткий час (чим коротше, тим краще), немає потреби розподіляти інформацію про відкликання. Нарешті, вузли не зберігають жодного таємного значення, наприклад, парольної фрази чи приватного ключа. Замість цього, приватні ключі створюються після того, як користувачі вводять свою парольну фразу, і парольна фраза негайно стирається. Приватний ключ також негайно стирається після його використання. Цей підхід захищає всі пристрої від компромісу вузлів, навіть без будь-яких пристроїв, що захищають від несанкціонованого доступу.

1.3. Постановка задачі

Звичайно, якщо всі партії ключових слів є попередньо створеними, і всі відкриті ключі містяться в єдиному сертифікаті, то проблема буде вирішена. Однак це означатиме, що сертифікат може стати дуже великим, а також накладатиме серйозні накладні витрати на зберігання в вузлах. Крім того, в такій схемі, коли вузол скомпрометований, всі попередньо сформовані загальнодоступні/приватні ключі легко працюють. Таким чином, нашою метою є створення єдиного сертифіката звичайного розміру, який підтверджує автентичність послідовності відкритих ключів.

Одне часткове вирішення цієї проблеми експлуатує властивості односторонніх хеш-ланцюгів. Хеш-коди в односторонньому хеш-ланцюзі однозначно пов'язані з одним "кінцевим" хеш-кодом криптографічною

(односторонньою) хеш-функцією. Коли цей "остаточний" хеш-код підписується АС, коди хешу в ланцюжку однозначно пов'язані з цим підписом. У схемі Веймерскірх - Вестхофф хеш-коди в хеш-ланцюзі розглядаються як секретні ключі. Лише той вузол, на якому був виданий сертифікат, може створити ключ, який правильно пов'язаний з хеш-кодом та підписом. Це рішення дозволяє вузлам в спеціальних мережах довести достовірність відкритих ключів, розкриваючи ключі з їх односторонніх хеш-ланцюгів. Проте такий підхід підлягає повторній атаці.

Якщо односторонні хеш-ланцюжки складалися з пар публічний/приватний ключів, і якщо використання публічного ключа засвідчує знання відповідного приватного ключа, то зловмисник не зможе побудувати успішну атаку, відтворюючи відкритий ключ. Вузли зможуть довести достовірність їх генерованого відкритого ключа, довівши, що він правильно пов'язаний з хеш-кодом, що містяться в їх сертифікаті. Однак, оскільки вони в даний час визначені, однонаправлені хеш-ланцюжки не дозволяють перевіряється прив'язування пар з публічними/приватними ключовими словами до одного хеш-коду.

Запропонована у роботі схема подолає цю проблему і це дозволе генерувати односторонніх хеш-ланцюжків публічних/приватних ключів, які мають подібні властивості до "традиційних" односторонніх хеш-ланцюгів. Ці пари публічних/приватних ключів підходять для надання послуг безпеки, таких як конфіденційність, автентифікація, цілісність тощо.

1.4. Висновки

На даний момент більшість фірм і підприємств все більше уваги приділяють використанню безпосередньо Wi-Fi-мереж. Обумовлено це зручністю, мобільністю і відносною дешевизною при зв'язку окремих офісів і можливістю їх переміщення в межах дії обладнання. У Wi-Fi-мережах застосовуються складні алгоритмічні математичні моделі аутентифікації, шифрування даних, контролю цілісності їх передачі - що дозволять бути відносно спокійним за збереження даних при використанні даної технології.

Однак дана безпеку відносна, якщо не приділяти належної уваги налаштування бездротової мережі. До даного моменту вже існує список «стандартних» можливостей які може отримати хакер при недбалості в налаштуванні бездротової мережі:

- доступ до ресурсів локальної мережі;

- прослуховування, злодійство (мається на увазі безпосередньо інтернет-трафік) трафіку;

- спотворення проходить в мережі інформації;

- впровадження підробленої точки доступу;

- розсилка спаму від імені вашої мережі.

Безпека є важливою для спеціальних мереж, особливо для програм, чутливих до уражень. При захисті мережи, є наступні атрибути: доступність, конфіденційність, цілісність, автентифікація та відсутність відмови.

Доступність забезпечує живучість мережевих сервісів, незважаючи на атаки відмови в обслуговуванні. Атака відмови в обслуговуванні може бути запущена на будь-якому рівні спеціальної мережі. На фізичних і мультимедійних рівнях контролю доступу противник може застосувати перешкоду для перешкоджання спілкуванню на фізичних каналах. На рівні мережі супротивник може порушити протокол маршрутизації та відключати мережу. На вищих рівнях зловмисник може знищити послуги високого рівня. Однією з таких цілей є ключова служба управління – найважливіший сервіс для будь-якої системи безпеки.

Конфіденційність гарантує, що певна інформація ніколи не розкривається неавторизованим особам. Передача конфіденційної інформації через мережу, таку як стратегічна або тактична військова інформація, вимагає конфіденційності. Витік такої інформації ворогам може мати руйнівні наслідки. У деяких випадках інформація про маршрути також повинна залишатись конфіденційною, оскільки інформація може бути корисною для ворогів для визначення та розміщення своїх цілей на полі бою.

Цілосність гарантує, що передане повідомлення ніколи не пошкодить. Повідомлення може бути пошкодженим через добрі невдачі, такі як погіршення поширення радіосигналу, або через шкідливі нападки на мережу.

Аутентифікація дозволяє вузлу забезпечити ідентичність однорангового вузла, з яким він спілкується. Без перевірки автентичності противник може маскувати вузол, таким чином, отримуючи несанкціонований доступ до ресурсу та конфіденційної інформації та втручаючись у роботу інших вузлів.

І, нарешті, нерозголошення гарантує, що походження повідомлення не може заперечувати відправлення повідомлення. Беззаперечно, корисно для виявлення та ізоляції зламаних вузлів. Коли вузол А отримує помилкове повідомлення з вузла В, невідхилення дозволяє А звинувачувати В за допомогою цього повідомлення та переконати інші вузли в тому, що В скомпрометовано.

РОЗДІЛ 2. СПЕЦІАЛЬНА ЧАСТИНА

2.1. Генерування хеш-функції публічної та приватної пар

2.1.1. Проста схема генерування хеш-функції

2.1.1.1. Специфікація.

Почнемо з того, що запропонував дуже просту схему, інспіровану оригінальною ієрархією. Перед використанням схеми, певні параметри системи повинні бути узгоджені всіма сторонами, а саме:

• $G = (G,*)$ – це скінченна циклічна група порядку q (для деякого великого q), $g \in G$ є генератором G, і припускається, що обчислення дискретних логарифмів в G стосовно g є обчислювальним нездійсненним. Наприклад, G може бути великою мультиплікативною підгрупою Z^*_p для деякого великого простого p, де q - велике просте ділення p-1; Альтернативно G може бути групою точок на еліптичній кривій (звичайно написана додатково).

• h – це криптографічна хеш-функція, яка зображає бінарні рядки довільної довжини до струн фіксованої довжини ℓ

• f – це криптографічна (одностороння) хеш-функція, що має множину $\{0,1, ..., q\text{-}1\}$ на себе. На практиці f може бути отримано з h.

• $m \geq 1$ - це позитивне ціле число, яке визначає кількість ключових пар, доступних для вузла.

Коли схема ініціалізується, вузол A повинен спочатку вибрати секретний $s \in \{0,1, ..., q\text{-}1\}$. A генерує загальну кількість приватних ключів, K_i, $0 \leq i$

$$K_i = f(s + i) \tag{2.1}$$

де $s + i$ обчислено за модулем q. Відповідний відкритий ключ для K_i просто g^{Ki} (де тут, як і в усьому світі, ми використовуємо мультиплікативне позначення як скорочення для операції групи). Перевірчий вузол A, тобто вузол, який повинен бути автентифікований, потім генерує значення перевірки як:

$$\text{Ʊ} = h\,(h\,(g^{K_0}) \,\|\, h\,(g^{K_1}) \,\|\, \cdots \,\|\, h\,(g^{K_{m-1}})) \qquad (2.2)$$

де g^{Ki} перетворюється на бітову рядок (деякими засобами) перед застосуванням h, а тут, як і всьому, || представляє операцію конкатенації.

Нарешті, υ включено в сертифікат, підписаний деякими ЦА. Перш ніж продовжувати, зверніть увагу, що ініціалізація схеми потенційно передбачає значну кількість обчислень, особливо якщо величина m велика. Це може бути головною перешкодою для обмежених пристроїв. Проте, ми бачимо, що ці обчислення могли бути здійснені для A під час розподілу пристрою, наприклад, тією самою КА, яка генерує сертифікат, що містить υ.

Тепер зберігаються сертифікати, а також значення $h\,(g^{K_i})$, $0 \le i < m$, A також надійно зберігає таємниці s. Це може ґрунтуватися на паролі або пропущенні фрази, які не зберігаються у форматі A, але замість цього вони вводяться до A кожного разу, коли треба створити приватний ключ K_i для деяких i.

У часовому інтервалі T_i, $0 \le i < m$, A може використовувати приватний ключ $K_i = f(s+i)$ та відповідний відкритий ключ g^{Ki}. Щоб увімкнути перевіряючий вузол B, тобто вузол, який автентифікує A, для підтвердження цього відкритого ключа, доказовий вузол A надсилає йому значення

- g^{K_i}
- $h(g^{K_j})$, $0 \le j < m$, $j \ne i$
- сертифікат, що містить υ

В обчислює $h\,(g^{Ki})$ і поєднує його з іншими хеш-значеннями, наданими компанією A, для обчислення:

$$\upsilon^* = h\,(h\,(g^{K_0}) \,\|\, h\,(g^{K_1}) \,\|\, \cdots \,\|\, h\,(g^{K_{m-1}})) \qquad (2.3)$$

і, нарешті, перевіряє, що $\upsilon = \upsilon^*$

2.1.1.2. Властивості

Ця схема має такі властивості:

- Зберігає єдине таємне значення, s.
- Сертифікат повинен бути єдиним ℓ кодом, υ.

• Знання про загальний стан (і інформацію) необхідний для перевірки не розкриває ніякої інформації про інші відкриті ключі.

• Знання одного приватного ключа *K* я не розкриваю нічого інформація про інші приватні ключі.

• нарешті, *A* потребує доступ до значень:

$$h\,(g^{K_0}),\ h\,(g^{K_1}),\ \ldots,\ h\,(g^{K_{m-1}}), \tag{2.4}$$

їх не потрібно зберігати, оскільки вони є спільними. Навіть якщо вони пошкоджені, то немає небезпеки для загальної системи (тільки для сервісу).

2.1.1.3. Обмеження

Є деякі недоліки простої схеми. Особливо, коли вузол завжди дозволяє перевіряючому вузлу мати доступ до перевірення публічного ключа, та обов'язково надіслає *m*-1 хеш-кодів з *A* до *B*. Для великих *m* це може бути значним обсягом зв'язку. Тому в наступному розділі – схема, яка діє дуже схожим чином з поточною схемою, при тому, що комунікації істотно зменшена.

2.1.2. Покращена проста схема

2.1.2.1. Специфікація.

У цій схемі параметри системи точно такі як раніше, як і метод генерації приватних і громадських ключі. Єдина відмінність у методі, який використовується для перевірки значення υ.

Возмем $m=2^r$ для деякого цілого *r*. Далі обчислимо υ, використовуючи ієрархію. Тобто ми обчислюємо бінарну ієрархію хеш-кодів:

• При $H_{0,\,i} = h\,(g^{K_i})$, $0 \le i \le 2^r - 1$ (= *m*-1).

• Для кожного значення *k*, $1 \le k \le r$, виконується:

$$H_{k,\,i} = h\,(H_{k-1,2i} \,||\, H_{k-1,2i+1}) \tag{2.5}$$

для $0 \le i < 2^{r-k}$.

• Тоді, отримуємо

$$\upsilon = H_{r,0}. \tag{2.6}$$

Далі додаемо: (а) сертифікат, що містить υ (як і раніше); і (б) всю ієрархію хеш-кодів H i, j (з яких вони існують)

$$2^r + 2^{r-1} + \cdots + 2^0 = 2^r + 1^{-1} = 2^{m-1}. \quad (2.7)$$

Отже, B має змогу отримати будь-який відкритий ключ, а A надсилає відповідний набір *r* хеш-кодів (разом із сертифікатом, що містить υ).

2.1.2.2. Властивості

Змінена схема тепер скорочує зв'язоку від m хеш-кодів до $log_2(m)$ хеш-кодів.

Недолік схеми, що подвоює вимогу щодо зберігання *A*.

2.1.2.3. Обмеження

Схеми, які ми вже описали, є загальними Проблема, а саме вимога зберігати *m* (або майже 2*m*) ℓ -бікові хеш-коди. Якщо величина m велика, скажімо $m = 2^{20}$, а $\ell = 256$, то це вимагатиме, щоб провідний вузол зберігався навколо 30 мільйонів повідомлень. Ти можеш зареєструватися як користувач пробний рівень не обмежений. Необхідно визначити, в наступний розділ, схема, яка не вимагає докази вузол для зберігання *m* (або майже 2*m*) ℓ -бікових хеш-кодів.

2.1.3. Схема, заснована на експонування

2.1.3.1. Специфікація.

Параметри системи такі ж, як і в попередньому дві схеми Тим не менш, ключ пар і чек-вартість відрізняються, додатковими загальними параметрами, який може бути системним параметром, якщо це не так системний параметр, який він може включити до сертифіката *A*).

Це значення $t \in \{2,3, ..., q-1\}$ з властивістю *t* має великий мультиплікативний порядок по модулю *q*. Це може, для наприклад, нанесені на придбання, що перевищує вартість, а також знижується

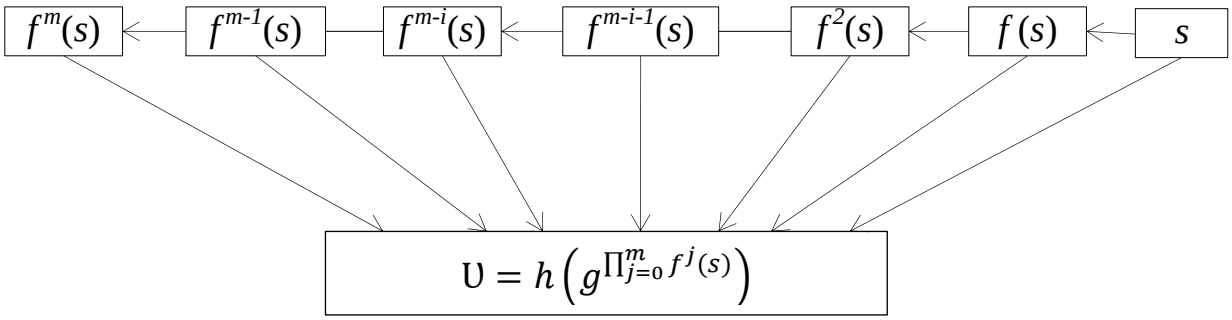

Рисунок. 2.1: Генерація контрольного значення υ

• $g^{\,st^{m-1-i}}$, and

• сертифікат, що містить υ.

B упізнається *i* (тому що це залежить від часу), і тоді можна обчислити: Примітивні елементи модуляції. Також здійснюється переваги при виборі малих t, наприклад, t = 2. Коли схема ініціалізується, вибирає таємницю s∈ {0,1, ..., q-1}. Тоді генерує значення перевірки υ як:

$$\upsilon^* = \left(g^{st^{m-1-i}}\right)^{i+1} = g^{st^m} \tag{2.8}$$

2.1.3.2. Обмеження

Важливо відзначити, що деякі бажані властивостібули загублені По-перше, пари публічних / приватних ключів не є пов'язані криптографічною хеш-функцією. По-друге, знання краєнфонив приватним користувачам не достатньо зазначитиінший приватні ключі. Схема, яку ми розглянемо в наступному розділіуникає цієї проблеми.

2.1.4. Схема хеш-ланцюга

2.1.4.1. Специфікація.

Параметри системи такі ж, як і в попередньому схема Тим не менш, ключ пар і контрольна вартість є виведені по-різному.

Коли схема ініціалізується, *A* вибирає таємницю *s*∈ {0,1, ..., *q*-1}. Як показано на малюнку 1, потім генерує A перевірка значення υ як:

$$\upsilon = h\left(g \prod_{j=0}^{m} f^j(s)\right) \tag{2.9}$$

Значення υ входить у сертифікат, підписаний CA.Згодом, як показано на малюнку 2, в інтервалі часу T_i 0≤i <m, A використовується як приватний ключ $K_{m\text{-}i\text{-}1}$, сформований как открытый ключ

$$K_{\,m\text{-}i\text{-}1} = \prod_{j=0}^{m-i-1} f^j(s)$$

і як відповідний відкритий ключ $g^{K\,m\text{-}i\text{-}1}$.

Увімкнути B, щоб отримати перевірену копію відкритого ключа g^{K_w}, A відправляє до B:

• $g^{K\,w}$,

• $f^{\,w+1}(s)$, i

• сертифікат, що містить υ.

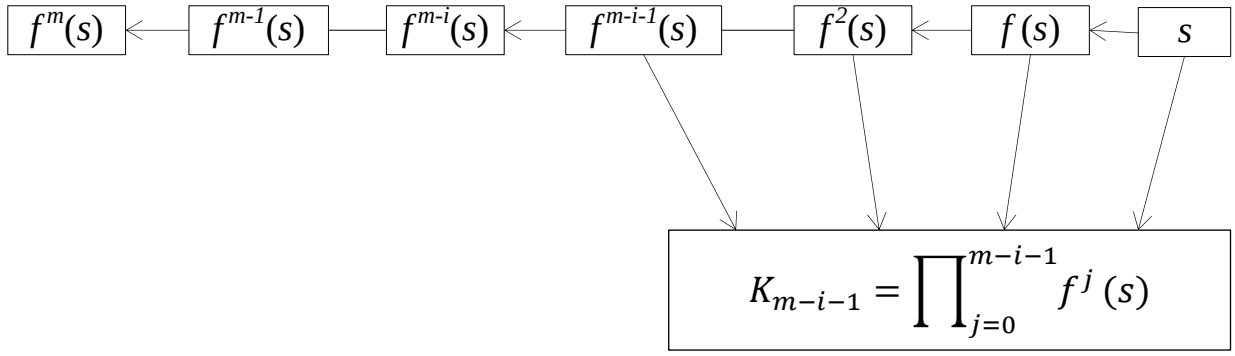

Рисунок 2.2.: Генерація приватного ключа K_{m-i-1}

B виводить *w* (тому що це залежить від часу), і тоді він може обчислити:

$$f^j(s),\ w+1 \le j \le m. \tag{2.10}$$

Використовуючи ці значення *i*, тоді B обчислює υ *

$$\begin{aligned} \upsilon^* &= (g^{K_w}) \prod_{j=w+1}^{m} f^j(s) = \\ &= g \prod_{j=0}^{w} f^j(s) \prod_{j=w+1}^{m} f^j(s) = g \prod_{j=0}^{w} f^j(s) \end{aligned} \tag{2.11}$$

Нарешті, *B* перевіряє, що υ = *h* (υ*).

2.1.3.2 Властивості

Ця схема має такі бажані властивості:

• Тільки *A* потрібно зберегти єдине таємне значення, *s*.

• Сертифікат повинен містити тільки один ℓ –Біт хеш-код, υ.

• Знання одного приватного ключа K_i не є самостійно розкривайте будь-яку інформацію про інші приватні ключі.

• Знання одного відкритого ключа g^{Kw} не виявляє інформацію про майбутні відкриті ключі.

• Необхідно дотримуватися загальноприйнятого та підтвердження. Нарешті, зверніть увагу, що певні оптимізації в відкритому ключі перевіркапроцесуможливопотрібнопорівняти звитратами довіреною копією минулого відкритого ключа доказуючого вузла.

Ця схема є кращим рішенням і використовується в решті частини папір

2.2. Приклад використаня

Ви хочете, щоб виникла проблема Попередній виклик. Як ми можемо створювати публічні ключі, які можуть бути автентифіковані унікальними сертифікат, виданий CA. Це рішення дозволяє собі по колінку, що свідчить про те, яка автентичність може перевірте, перевіривши підпис, який був згенерований CA при ініціалізації мережі.

2.2.1. Реєстрація у фіксованій мережі

Як пояснюється Елісоном та Шнайєром, перед видачею сертифікатів для використання в наданні аутентифікації суб'єктів або створення ключових документів, CA має спочатку запустити процес підтвердження ідентичності, під час якого перевіряється ідентифікація запитувача сертифіката. В спеціальних мережах ми не можемо припустити, що адміністративний орган буде присутнім, який зможе керувати процесом підтвердження особистості. Отже, у нашому рішенні сертифікати видаються CA, розташованим у фіксованій мережі. Це CA може, наприклад, бути мережевим провайдером, який головний реєстр має для того, щоб його зв'язок передавалася через мережу, що не працює. Процес реєстрації в мережевому провайдері часто вимагає від керівників подання дійсних паперових повноважень. Ці посвідчення можуть бути використані постачальником послуг для

автентифікації принципалу перед створенням сертифіката. Детальний процес автентифікації полягає в наступному.

Таблиця 2.1– Виконання вимог

Властивість	Дуже проста схема	Покращена проста схема	Експоненційна схема	Схема Хеш-ланцюга
Єдине таємне значення	Виконує	Виконує	Виконує	Виконує
Зв'язування пар ключів за допомогою функції хешування	Виконує	Виконує	Не виконує	Виконує
Криптосистема відкритого ключа на основі дискретного логарифму	Виконує	Виконує	Виконує	Виконує
Від'єднані приватні ключі	Виконує	Виконує	Не виконує	Виконує
Верифікація відкритого ключа від хеш-коду	Виконує	Виконує	Виконує	Виконує

Таблиця 2.2. – Переваги схем.

Схеми	Переваги
Дуже проста схема	Необхідно зберегти лише одну секретну цінність Сертифікат повинен містити лише один ℓ-бітовий хеш-код, υ Знання одного відкритого ключа не розкриває жодної інформації про інші відкриті ключі Знання одного приватного ключа не розкриває жодної інформації про інші приватні ключі
Покращена проста схема	Такі ж переваги, як дуже проста схема. У порівнянні з дуже простою схемою, вартість зв'язку зменшується від коду х хеш-коду до хеш-кодів журналу 2 (м)
Експоненційна схема	Необхідно зберегти лише одну секретну цінність Сертифікат повинен містити лише один ℓ-бітовий хеш-код хеш-код, υ Знання одного відкритого ключа не розкриває жодної інформації про майбутні відкриті ключі. Лише відкритий ключ та сертифікат необхідно надіслати комунікаційній стороні

Продовження Таблиці 2.2.

Схеми	Переваги
Схема Хеш-ланцюга	Такі ж переваги, як схема, що спирається на експонування. Але в порівнянні з показом на основі схема, схема хеш-ланцюга додатково відповідає вимозі, що знання одного приватного ключа само по собі не дозволяє розрахувати будь-які інші приватні ключі

Таблиця 2.3. Недоліки запропонованих схем.

Схеми	Недоліки
Дуже проста схема	Код хешу *m*-1 повинен бути опублікований, щоб перевірити дійсність одного відкритого ключа
Покращена проста схема	Порівняно з дуже простою схемою, він подвоює вимоги до зберігання для аутентифікованого об'єкта
Експоненційна схема	Пари публічних / приватних ключів не пов'язані криптографічною хеш-функцією Знання одного приватного ключа достатньо для визначення всіх інших приватних ключів

Коли вузол *A* знаходиться у фіксованій мережі, він зв'язується з ЦС, щоб синхронізувати його годинник та отримати надійну копію системних параметрів, як це визначено в розділі 3.1, і *K* CA, відкритого ключа CA. CA надсилає ті самі параметри всім об'єктам, які запитують їх. Коли *A* отримує ці параметр. Зауважте, що не зберігається A; Замість цього, він генерується з сильної парольної фрази. Тоді *A* генерує контрольний значення υ.

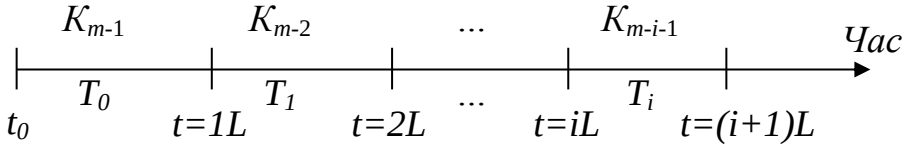

Рисунок 2.3. Відображення між приватними ключами та інтервалами часу

Потім *A* відправляє своїй ідентифікаційній ідентифікаційній одиниці *A*, контрольній величині υ та хеш-коду $f^{m+1}(s)$ до CA, щоб отримати сертифікат, який зв'язує ці значення. Наступне CA перевіряє, що дійсно належить

ідентифікатор A, і те, що він був виданий, містить в собі і $f^{m+1}(s)$. Якщо вся перевірка буде успішною, тоді CA надсилає до A сертифікат $Cert_A$

$$[\mathrm{ID}_A, \upsilon, f^{m+1}(s), t_0, L, nb\_keys]_{K^{-1}} \quad (2.12)$$

де t_0 - час випуску сертифіката, L - ціле число, яке може бути вказано в записі сертифіката A або CA, nb ключі, як визначено у розділі 4.4 нижче, і $[...]_{K^{-1}}$ CA позначає підпис, згенерований CA за допомогою його приватного ключа K^{-1} CA. Як тільки A отримав свій сертифікат, не потрібно знову звертатися до CA. A розподіляє час на інтервали, що дорівнюють однаковій довжині L. Для кожного інтервалу призначається ключ, який генерується. Відображення між клавішами та інтервалами часу показано схематично на рисунку 4.

2.2.2. Встановлення захищених каналів зв'язку в спеціальних мережах

У часовому інтервалі T_i, $0 \le i < m$, тобто між інтервалами $t = iL$ та $t = (i+1)L$, A може бути автентифікований B, і обидва можуть встановити аутентифікований сеанс ключ за допомогою протоколу, який ми зараз опишемо. Спочатку надсилає B:

- g^{Kw}
- $f^{w+1}(s)$
- $Cert_A$

Коли він отримує ці значення, B спочатку перевіряє підпис CA на Cert_A. Якщо це дійсно, то B визначає і за місцевим часом, значення L в сертифікаті та час випуску $Cert_A$. B потім перевіряє, що $f\,w + 1$ (s) належить до одного і того ж хеш-ланцюжка в односторонньому порядку, як $f^{m+1}(s)$. Якщо так, то B породить υ^*. Нарешті, B перевіряє, чи є $\upsilon = h(\upsilon^*)$. Якщо верифікації успішні, B знає, що g^{Kw} було створено суб'єктом, якому сертифікат $Cert_A$ був виданий CA. B тепер надсилає поточно діючий відкритий ключ g^{Ky} до A. A може перевірити дійсність ключа B за допомогою того самого методу, що і описано. Після того, як A має затверджувати g^{Ky}, A та B поділиться наступною аутентифікованою ключа сеансу Діффі-Хеллмана SK:

$$\mathrm{SK} = (g^{Kw})^{Ky} = (g^{Ky})^{Kw=}\, g^{KwKy}. \quad (2.13)$$

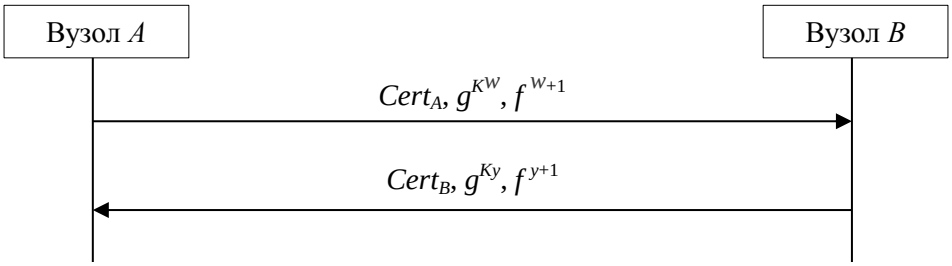

Рисунок 2.4. Повідомлення обмінюються A та B, щоб встановити аутентифікований сеанс ключ SK.

SK може використовуватися командами *A* та *B* для отримання загальних ключових секретних сеансів для взаємної автентифікації, конфіденційності, цілісності тощо. Обмін повідомленнями, який використовується для створення SK, підсумовується на рисунку 2.4. *A* не зберігає SK та K^w, але генерує їх, коли вони повинні бути використані і стирає їх після їх використання. Тільки g^{Ky} зберігається на *A* протягом часового інтервалу T_i. Коли T_i минув, g^{Ky} стає недійсним і може бути стертий на *A*.

Ця схема також може використовуватися з асиметричною криптосистемою для підтримки шифрування відкритих ключів та цифрового підпису. *A* та *B* можуть обговорити криптосистему, яка буде використовуватися, коли вони обмінюються своїми публічними цінностями.

Інформація g^{Kw} та $f^{w+1}(s)$, випущені компанією *A* протягом часового інтервалу T_i, не дозволяють зловмисникові встановити автентифіковану сеансову ключу з B або видати себе за атаку. Для цього злодію потрібно буде виявити K_w від g^{Kw} в інтервалі T_i (довжини *L*). Однак це обчислення неможливо. Крім того, оскільки тільки *A*, який знає правильну парольну фразу, може генерувати s, лише *A* може встановити дійсний ключ сеансу SK з B протягом часового інтервалу T_i.

2.2.3. Створення сертифікатів у спеціальних мережах

Схема, визначена в розділі 2.1.3, вимагає вузлів для створення нових пар ключів протягом кожного інтервалу часу. Це може бути непросто, якщо,

наприклад, очікується, що безпечний обмін буде тривати довше, ніж один інтервал часу. Щоб вирішити цю проблему, ми тепер визначаємо рішення, яке дозволяє вузлам генерувати сертифікати відкритих ключів з дуже коротким терміном служби, що, тим не менше, триває довше одного інтервалу часу. Це рішення призначене для ситуацій, коли вузол *A* знає або може оцінити, як довго триватиме безпечна обмін з вузлом *B*. Тому вузол *A* може вказати в своєму сертифікаті термін придатності, що обмежує дійсність цих сертифікатів до відповідного періоду часу. *A* також може зменшити використання цього сертифіката до певного контексту, наприклад, додати в сертифікат поле, що містить ідентифікатор *B*. Таким чином, B автоматично вважає, що сертифікат *A* недійсний, як тільки захищена взаємодія припиняється. Це також робить цей сертифікат недійсним, якщо він не належить до іншого пристрою, крім *B*. Справжність цих сертифікатів може бути підтверджена перевіркою, що вони правильно пов'язані з підписом, згенерованим ЦС при ініціалізації. Наше рішення працює таким чином.

У часовому інтервалі T_i, *A* генерує пару публічний/приватний ключ *A*/*Priv A*, а також генерує сертифікат $Cert2_A$ для Pub_A з наступним вмістом:

• [ID_A, υ, $f^{m+1}(s)$, t_0, *L*, *nb keys*] $_K$-1 CA, тобто сертифікат $Cert_A$, який видав КА при ініціалізації;

• Pub_A;

• ідентичність *B*;

• час випуску;

• час закінчення терміну дії;

• Підпис, згенерований на попередніх полях за допомогою приватного ключа K_{m-i-1} для цього інтервалу часу.

Приватний ключ K_{m-i-1} використовується для підпису $Cert2_A$. Після підписання сертифіката *A* видаляє K_{m-i-1} і не генерує його ще раз. Це не дозволяє будь-якому злодію генерувати дійсний сертифікат.

Тоді, для автентифікації *B*, *A* надсилає $Cert2_A$ разом з g^{Kw} і f^{w+1} до *B*. Якщо сертифікат містить його ідентифікацію, B потім перевіряє, що час

закінчення не пройшов, і що [ID *A*, υ, $f^{m+1}(s)$, t_0, *L*, *nb keys*] було підписано CA. Якщо так, то B використовує час t_0, *L* та час випуску для визначення поточного інтервалу часу в *A*. Тоді *B* перевіряє автентичність g^{Kw}, як описано у розділі 3.4, використовуючи знання часового інтервалу та f^{w+1}. Якщо перевірка буде успішною, то *B* перевіряє підпис, згенерований *A* на $Cert2_A$. Нарешті, якщо все верифікація виконується, B знає, що $Cert2_A$ був сформований вузлом, який видав сертифікат [ID *A*, υ, $f^{m+1}(s)$, t 0, *L*, *nb keys*] *K* -1 CA за CA. Тоді *B* вважає Pub_A дійсним. Сертифікат $Cert2_A$ повинен бути підтверджений у T_i і може бути використаний до закінчення терміну дії сертифіката.

У випадку, коли *A* зберігає $Pub_A/Priv_A$, зловмисник може отримати інформацію про цю пару публічного/приватного ключа. Однак, оскільки *s* не зберігається на пристрої, зловмисник не може створити новий дійсний сертифікат для пари ключів. Тоді, якщо *A* вже припинив взаємодію з *B*, *B* вважає $Cert2_A$ недійсним. Тому зловмисник не може видати себе за *A*. Однак якщо *A* ще не припинив взаємодії з *B*, то *B* нарешті вважатиме $Cert2_A$ дійсним, і атакуючий може виставити *A* на *B* до закінчення терміну дії сертифіката $Cert2_A$. В останньому випадку *A* може знадобитися інформувати *B*, що $Cert2_A$ слід вважати недійсним. Для досягнення цього *A* може відправити заяву про відкликання, підписану з $Priv_A$. Однак, як раніше обговорювалося, часто мінлива топологія мережі adhoc робить, що B не може отримати цей випис. Альтернативний підхід вимагає, щоб не зберігати $Priv_A$, але щоб створити його, коли це необхідно. Що стосується ключа $K_{m\text{-}i\text{-}1}$, $Priv_A$ слід стерти відразу після його використання. Внаслідок цього зловмисник не в змозі вивчати $Priv_A$. Отже, зловмисник не може видати себе за апробацію, підтверджуючи право власності на пару публічний/приватний *A* до *B*.

Важливо зазначити, що обговорення відкликання не стосується приватного ключа $K_{m\text{-}i\text{-}1}$, яке ніколи не зберігається і яке використовується лише один раз для підписання сертифікату $Cert2_A$. Важливо також помітити, що якщо ключ пари $Pub_A/Priv_A$ зберігається, наше рішення значно зменшує

кількість вузлів, які потребують отримання заяви про відкликання - у порівнянні з існуючими рішеннями. Це також обмежує необхідність скасування до короткого періоду часу, коли сертифікат $Cert2_A$ дійсний.

2.2.4. Робота з проблемами синхронізації часу

У вищенаведених схемах *A*, *B* та CA мають синхронізовані годинники. Це досягається як частина етапу ініціалізації, коли *A* та *B* з'єднуються з фіксованою мережею. Однак згодом їх годинник буде працювати незалежно від CA та фіксованої мережі. Дрифт годинника між годинниками *A* та *B* означає, що коли *A* та *B* знаходяться в спеціальній мережі, їх годинники більше не будуть точно синхронізовані. Це може зробити, що поточний інтервал часу A в точці B відрізняється від поточного інтервалу часу *A*. Попереднє випливає з того, що значення i, обчислене B, для перевірки того, що розкритий відкритий ключ правильно пов'язаний з υ, відрізняється від правильного значення i, яке дійсно дозволяє перевірити цю зв'язок. Тому в цьому випадку аутентифікація *A* на *B* завжди не вдається.

Для того, щоб автентифікуватись B в цьому контексті, запропоновано рішення, яке вимагає, щоб *A* вибрав параметр *d*, що представляє максимальну диспропорцію, яка допускає між її годинником і годинником *B*. Потім, протягом кожного інтервалу часу, *A* може розкривати два відкритих ключі, час початку яких діє, як показано на малюнку 6. Кожна з розкритих відкритих ключів має термін дії *L*. Отже, запропоноване рішення дозволяє використовувати *B* дійсний відкритий ключ протягом періоду часу (*L*+*d*), тобто протягом періоду часу, збільшеного на *d*, у порівнянні з випадком, коли тільки один відкритий ключ розкривається у *A*. На рисунку 7 показано, як запропоноване рішення дає можливість для B для автентифікації *A*, якщо годинник дрейфує між їх годинником більше, ніж 0 і менше або дорівнює *d*. У цій цифрі, коли *A* знаходиться в першому проміжку часу, він посилає до B відкриту клавішу $g^{K\ m\text{-}1}$, термін дії якого йде від часу $t = 0$ до часу $t = 1L$, а відкритий ключ $g^{K'm\text{-}1}$, термін дії якого йде від часу $t = d$ до часу $t = L+d$. Тому запропоноване рішення гарантує, що принаймні одне з розкритих відкритих

ключів дозволяє B автентифікувати *A* і встановити аутентифікований сеансовий ключ з *A* в кожному інтервалі часу. Якщо протягом певного проміжку часу слід розкривати декілька ключових слів, CA може вказати його в сертифікаті за допомогою параметрів *nb keys*.

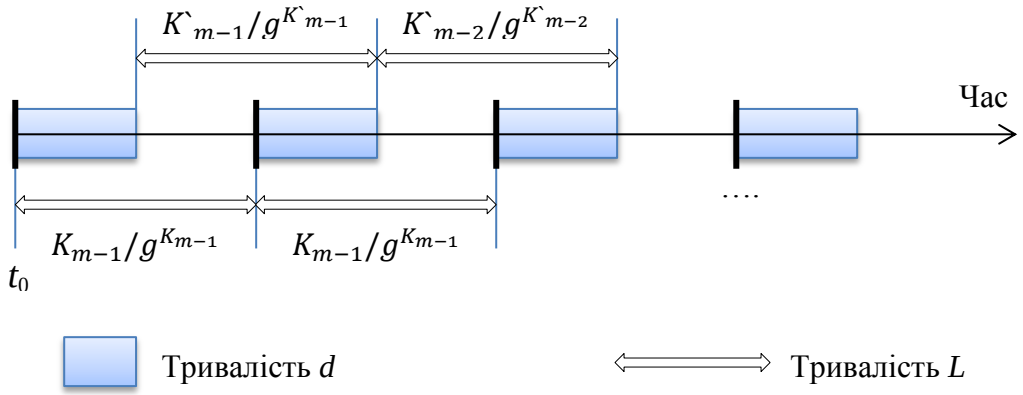

Рисунок 2.5. Залежність ключів за інтервалами часу, враховуючи максимальний відрізок часу d

У порівнянні з випадком, коли *A* розкриває один відкритий ключ у кожному інтервалі часу, запропоноване рішення збільшує кількість ключових значень фактору *nb* кількості обчислень, які *A* повинна виконувати для того, щоб генерувати відкриті ключі, які він розкриває до *B* у кожному інтервалі часу *L*. Тому вибір для значення *nb_keys* повинен бути компромісом між обчислювальними витратами та практичністю врахування великих дрейфів годинника. Якщо використання декількох ключів у кожному інтервалі часу впливає на обчислювальні витрати на рішення в A, це не збільшує обчислювальні витрати в B, оскільки B потрібно лише перевірити один відкритий ключ серед двох розкритих, період дії якого охоплює поточний час A у B. Можливо, буде прийнятним додавання у повідомлення, яке *A* надсилає до *B*, для того, щоб бути автентифікованим, деякі поля, що дозволяють B легко ідентифікувати період дії, який охоплює кожна з розкритих відкритих ключів .

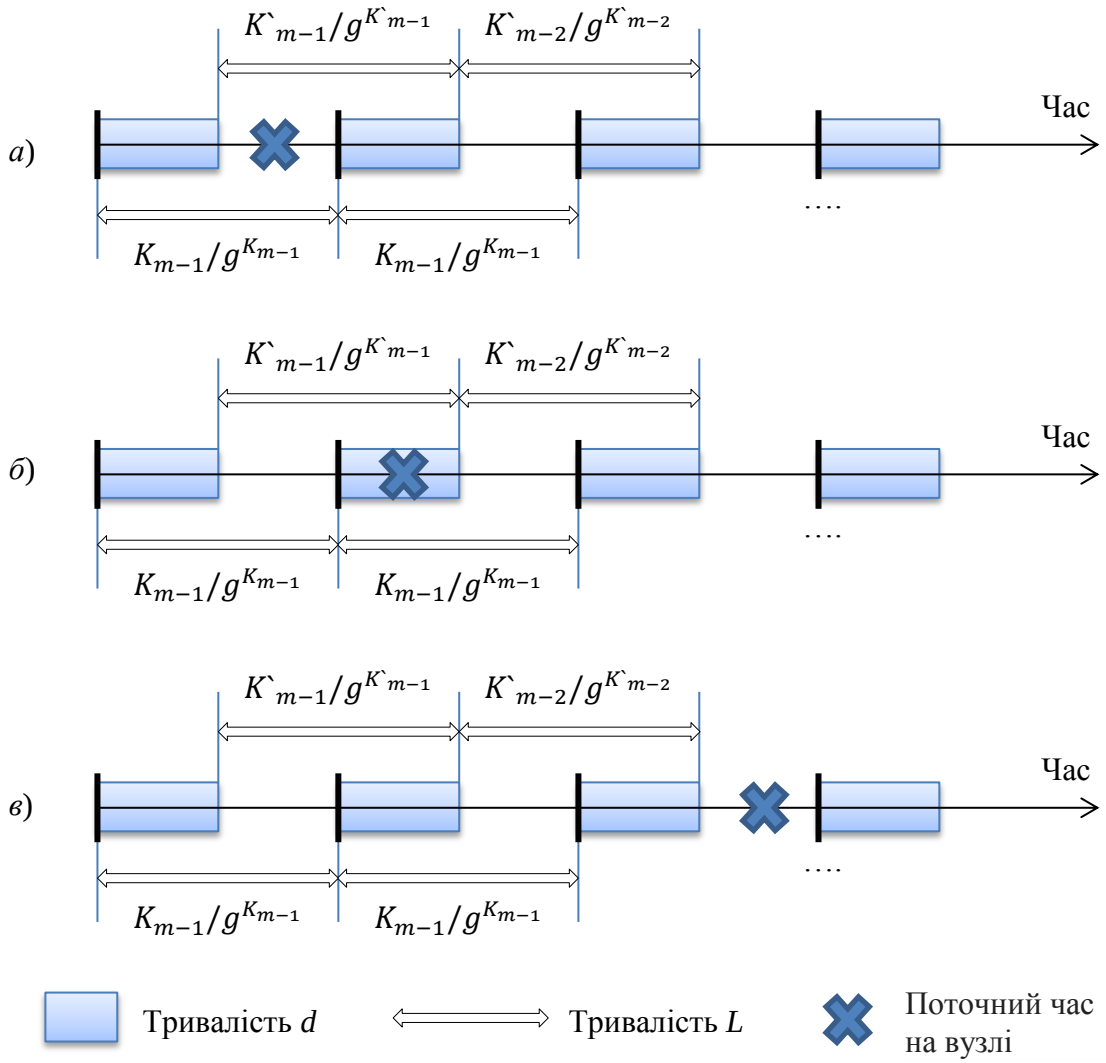

Рисунок 2.6. Відхидення значень A та B.

а) Момент *A; б*) Момент *A у вузлі B, при часовому інтервалі менше значення d;*

в) Момент *A у вузлі B, при часовому інтервалі більше значення d*

Захист схем, наведених у розділі 3, збільшується, коли L стає коротшим. Проте, чим коротше *L*, тим точніше вузли потрібно синхронізувати. Таким чином, вибір для значення *L* повинен бути компромісом між цими двома впливами. Деякі інші обмеження також застосовуються до значення *d*:

• Якщо *d* більше, ніж *L*, то рішення, визначене раніше, не гарантує, що автентифіковану сеансову клавішу завжди можна встановити між *A* і *B*, якщо існує велика різниця між їх годинником. Це тому, що термін дії двох

відкритих ключів, розкритих *A*, може мати період дії, який не обов'язково покриває поточний час *A* в *B*, як показано на рисунку 2.6.

• Якщо *d* менше, ніж *L*, то рішення, визначене раніше, дозволяє встановлювати аутентифіковану сеансову клавішу між *A* і *B*, навіть якщо існує невідповідність між їх годинником. Однак, якщо *d* занадто мале, то *A* може не змогти встановити аутентифіковану сеансову клавішу з усіма вузлами, з якими він хоче обміняти конфіденційні повідомлення.

Тому *d* має бути меншим, ніж *L*, і повинен бути вибраний *A* таким чином, щоб максимально розбити існуючу різницю між його часами та часами інших вузлів мережі.

2.3. Мережева робота

Далі йде перевірка особи, анулювання сертифіката та компромісу вузла. Перш ніж вдаватися до деталей, ми детально розповімо про відповідні роботи в області хеш-ланцюгів та аутентифікації на основі хеш-ланцюга.

2.3.1. Хеш ланцюги

Односторонній хеш-ланцюг був введений Лампортом. Початкова програма полягала в тому, щоб перешкоджати перехоплювати клієнта через відтворення перехоплених паролів на сервер. Кінцевий хеш-ланцюг одностороннього зв'язку генерується клієнтом, застосовуючи криптографічну хеш-функцію f багаторазово до його секретних с. Кожен елемент у ланцюжку має інший пароль, який клієнт може використовувати при кожній новій автентифікації. Паролі пов'язані між собою та між собою:

$$K_i = f^i(s),$$
$$K_{i+1} = f(K_i), \qquad (2.14)$$
$$K_j = f^{j-i}(K_i).$$

де $0 \le i \le m$, $1 \le j \le m$, $i < j$, і $f\,j\,(K\,i)$ означає, що криптографічна хеш-функція f застосовується j разів до K і. Для і-ї аутентифікації на сервер клієнт використовує пароль $K\ m\text{-}i$. В принципі, наша схема хеш-ланцюгів побудована на цій схемі, хоча елементи хеш-ланцюга різні і не поділяють такого недоліку. Сервер автентифікує клієнта, перевіряючи, що K m-і

пов'язано з K_{m-i} + 1, тобто паролем, використовуваним при попередній автентифікації, за співвідношенням. Оскільки криптографічні хеш-функції обчислюються важко інвертувати, зловмисник не може генерувати дійсний пароль з компромісного. Однак, як добре відомо, це означає, що підрозділи, пов'язані з персональними нападами.

З іншого боку, рішення, запропоноване в цьому документі, не є піддані цим нападам, оскільки клієнт ніколи не розкриває на сервері ключ, який тільки дозволяє серверу видати себе за клієнта на сторонню сторону.

Протокол автентифікації трансляції TESLA спирається на попередню схему аутентифікації джерел широкомовних пакетів. Проте рішення не забезпечує засоби для перевірки автентичності під час завантаження. Це повинно бути забезпечено за допомогою звичайної системи автентифікації даних під час налаштування сеансу. Але, як обговорюється далі в цьому розділі, звичайні системи аутентифікації даних вимагають, щоб довірений учасник - наприклад, TTP або заздалегідь відомий об'єкт - завжди був доступним, щоб дозволити встановлення загальних таємних ключів або перевірки дійсність сертифікатів. Проте вищесказане не може бути гарантовано в adhoc-мережах. Наше рішення не має цього обмеження, оскільки він не вимагає вузлів для взаємодії з сторонніми особами, щоб мати змогу автентифікувати та встановлювати загальні секретні ключі.

Рішення для цієї проблеми завантаження пропонується, де вузол має ряд сертифікатів, виданих CA. Кожен сертифікат пов'язує ідентифікацію вузла з зобов'язанням (останньою згенерованої ключа) його одностороннього хеш-ланцюга. Більш конкретно, A вибирає секретний ключ x 0, який є прив'язкою ланцюга. Враховуючи безпечну криптографічну хеш-функцію *h*, системний параметр $m \geq 1$ і $x_i = h^i(x_0)$ ($i \geq 0$), останній згенерований хеш-ключ x_{2m} з ланцюжка *A* міститься в сертифікаті *A*. Цей сертифікат також містить час випуску та випадкове насіння. *A* розподіляє час на інтервали з однаковою довжиною *L*, де *L* є загальнодоступною, і призначає дві клавіші його одностороннього хеш-ланцюга для кожного інтервалу. Під час

інтервалу $T_i(0 \leq i < m)$ між часами $t = iL$ та $t = (i+1)\ L$, A використовує ключі $x_{2m-2i-1}$ і $x_{2m-2i-2}$ з його ланцюга, щоб дозволити B для автентифікації A. B. може перевірити правильність цих двох значень, спочатку перевіривши сертифікат, щоб отримати надійну копію x_{2m}, а потім вилучити криптографічну хеш-функцію відповідного числа разів на розкритих клавішах і порівняти результат з значеннях $2m$. Проте, як тільки розкриваються $x_{2m-2i-1}$ і $x_{2m-2i-2}$, зловмисник може відтворити їх суб'єкту C, щоб прикинутися, що він є A. Щоб зменшити цей ризик, B повинен попросити A використовувати сертифікат із специфічним значення насіння Як наслідок, зловмисник може повторно відтворювати раніше розкриті ключі, якщо це запитує перевіряючий об'єкт, щоб він представив сертифікат, що містить насіння, вже запитуване протягом поточного періоду часу або протягом попереднього. Схема має такі цікаві властивості:

• У вузлах потрібно лише тримати одну секретну (x_0).

• У сертифікат потрібно включити лише одну публічну цінність.

• Знання ключів, випущених в попередні часові інтервали, не розкриває жодної інформації про ключі, використовувані в більш пізні інтервали часу.

Однак, оскільки ключі повинні бути розкриті для виконання автентифікації організації, вони не можуть бути використані для встановлення безпечного каналу зв'язку. Це короткий прихід, який подолано нашою схемою, оскільки наші ключі рішення (з хеш-ланцюга) не розкриваються безпосередньо, а замість цього використовуються для генерації публічних ключів, які розкриваються для виконання аутентифікації організації.

Коли вони використовуються в спеціальних мережах, існуючі рішення для аутентифікації на основі хеш-ланцюгів мають всі обмеження, обговорені в попередньому. Наскільки нам відомо наше рішення - це єдине, що було запропоновано, і це не має цих обмежень.

2.3.2. Перевірка посвідчення особи

Мета сертифіката - довести принципові перевірки, що зв'язок між ідентифікацією та ключем є автентичною. Це досягається за допомогою підпису, яке TTP або вступник генерує на сертифікаті, якщо він містить дійсне обов'язкове квотування перевіренного ключа. Після успішної перевірки дійсності цього підпису, перевіряючий принцип отримує впевненість, що сертифікат є автентичним. Щоб дозволити перевірку справжності сертифікатів в мережах adhoc, було запропоновано три підходи. Тоунси, Сангсгири та ін., а Веймерскірх - Вестхофф пропонують рішення, які покладаються на сертифікати, випущені в фіксованих мережах. Це дозволяє сертифікати бути підписані деякими ЦС, недоступними в мережах adhoc. Однак це не дозволяє вузлам бути видані нові сертифікати, коли вони знаходяться в спеціальних мережах, хоча це може бути необхідним. Це, наприклад, випадок, коли анонімність повинна бути надана. Це питання, яке вирішується нашим рішенням, як ми обговорили в розділі 4.3.

Для того, щоб дозволити видавати сертифікати в спеціальних мережах, запропонований підхід полягав у адаптації PKI. Наприклад, Чжоу та Хаас визначають рішення, яке використовує порогове криптографія, і пропонує розподілити питання про сертифікати для n спеціальних вузлів, які називаються серверами. Проте, навіть якщо в мережі є більше серверів, ніж номер для з'єднання, щоб отримати повний підпис, динаміка вузлів не гарантує, що завжди достатньо серверів. Luoetal.aswell як Kong і співавт. [20] вирішити цю проблему, поширюючи служби CA всім вузлам мережі, не покладаючись на онлайн-TTP. Проте, як буде показано далі, ці рішення мають інші обмеження.

Альтернативним підходом є адаптація PGP, оскільки PGP не покладається на TTP для видачі сертифікатів. Це підхід що використовується Капкуном та ін. і Лі і ін. Наприклад, вважається, що вузли знають своїх сусідів і завжди видає їм сертифікати з дійсним обов'язковим ключем користувача. Всі вузли, що підписали сертифікат, вважаються чесними і їм

довіряють однаково. Проте немає гарантії, що вузли, які заздалегідь відомі, завжди присутні в спеціальних мережах, а також, що ці вузли завжди поводяться правильно. PGP не спирається на таке припущення. Відомі принципи не всі довіряють, щоб належним чином перевірити прив'язки "ідентичності-ключ". Використовуються рівні довіри, які відрізняють цілком довірливих представників від менш довірливих вступників. Крім того, вказується глибина довіри, яка визначає, чи може довіра, надана принципалу, поширюватися на принципів, яких довіряє цей принцип. Всі ці питання уникаються нашим пропонованим рішенням.

Деякі рішення, як ідентифікація перевіряється при реєстрації. Наше рішення не вимагає довіри до будь-якого вузла в adhoc-мережі. За відсутності будь-якої валідації, не можна гарантувати, що безпечні канали зв'язку встановлюються з передбачуваним вузлом при використанні не перевіреного сертифіката або невиправданого довготривалого секретного ключа. Ло і співавт. вирішити цю проблему, спираючись на людське сприйняття. Однак, якщо людське сприйняття використовується в спеціальних мережах, то лише вузли, якими керують особи, які знають один одного, можуть бути автентифіковані. Аналогічно, Stajano пропонує рішення, яке спирається на фізичний та електронний контакт, щоб перенести біти секретного ключа з одного вузла в інший. Імпринт має виконуватися перед входом в спеціальну мережу. Безпечне передавання бітів секретного ключа не дозволяє зв'язувати таємницю з ідентифікацією дійсного директора.

Ло і співавт. альтернативно пропонують покладатися на біометрику. Проте використання біометрії для перевірки ідентичності також викликає стурбованість, оскільки під час процесу реєстрації біометрична характеристика повинна бути зібрана у вигляді шаблону. Біометричні шаблони всіх керівників, які повинні бути перевірені, повинні зберігатися на вузлі. Це накладає вимоги на зберігання на вузлі, які уникнути пропонованим рішенням.

Денг, Бойо та Мірі пропонують використовувати шифрування на основі ідентичності (МШП). МШП вперше був представлений Шаміром, але перша ефективна схема МШП була визначена Бонехом і Франкліном. Це криптосистема відкритого ключа, де відкриті ключі є довільними рядками, які представляють їхні власники. Початкова мотивація для МШП полягала в усуненні потреби в каталогах та сертифікатах. Однак генератор приватних ключів (PKG), як-от ЦС в PKI, є центральним об'єктом, доступ якого не може бути гарантований в спеціальних мережах. Тому випуск ідентифікаційних приватних ключів в мережах adhoc містить ті самі обмеження, що й випуск сертифікатів у спеціальних мережах.

Вищевказаний аналіз показує, що рішення, запропоновані раніше для перевірки тотожності та видачі сертифікатів adhoc мережі представляють проблеми, які залишаються невирішеними, але які вирішуються нашим пропонованим рішенням.

2.3.3. Анулювання сертифіката

Деякі рішення, такі як, не визначають жодного способу керування скасуванням. У літературі було запропоновано три підходи, спрямовані на управління відкликанням та розповсюдженням інформації про відкликання в спеціальних мережах. У цьому розділі ми обговорюємо ці підходи. Відкликання за часом, зменшує необхідність скасування. Однак це не обов'язково змусить його зникнути.

Наприклад, якщо термін дії досить довгий, наприклад, рік, то емітент, можливо, доведеться відкликати його відкритий ключ до кінця року. У цьому випадку необхідність відкликання така ж, як і при звичайних сертифікатах. Але коли термін дії дуже короткий, то потреба в ануляції може бути суттєво зменшена, як у вирішеному в даному документі рішенні.

Інший підхід, що використовується, покладається на виписування актів скасування, зберігання цих тверджень шляхом отримання вузлів у CRL та обміну цими CRL між вузлами для розповсюдження списку відкликаних сертифікатів у мережі. Однак можливі розділи в мережі можуть заважати

вузлам оновлювати свої CRL. Деякі рішення покладаються на поради інших вузлів для оцінки дійсності сертифіката. У запропонованому рішенні, наприклад, використовується моніторинг сусідів для виявлення неправильних вузлів, чиї сертифікати необхідно скасувати. Цей підхід спирається на інтерпретацію, що вузли мають поведінку інших вузлів. Тому такий підхід може призвести до ситуацій, коли помилково виявлені вузли ізольовані від мережі.

Використовується підхід, який базується на механізмі оцінки довіри, який дозволяє вузлу з даного кластера - групи вузлів - для аутентифікації вузла з іншого кластера. У такому підході сертифікат вважається дійсним, як тільки є більшість вузлів, які надіслали йому позитивні рекомендації. Але оскільки ніякого рішення не передбачено для того, щоб вузли могли перевірити статус відкликання сертифіката, ці вузли не можуть дізнатись, чи було скасовано сертифікат. Тому факт того, що більшість вузлів стягує сертифікат, не може довести свою дійсність.

Таким чином, жоден із підходів, визначених у цьому розділі та запропонованих у літературі, не гарантує, що вузли в спеціальних мережах ніколи не використовуватимуть сертифікати, які були скасовані. Підхід розглядає скасування неявно, оскільки сертифікати та пари ключових елементів є недовговічними. Вузол припиняє використовувати відкритий ключ, доки нова пара клавіш не стане доступною.

2.3.4. Компонування вузла

Зверніть увагу, що з перерахованих вище схем, тільки Raya та Hubaux враховують відсутність фізичного захисту мобільних пристроїв, явним чином захищаючи пари публічних/приватних ключів, використовуючи захищену від несанкціонованого обладнання. Однак, як зазначалося раніше, таке обладнання не доступне на кожному мобільному пристрої. Секрет, який використовується для створення хеш-ланцюга, ніколи не зберігається на мобільному пристрої, але стирається після його використання.

2.4. Оцінка ефективності

2.4.1. Параметри оцінки

Для того, щоб оцінити характеристики рішення, що спирається на схему хеш-ланцюга, реалізовується схема хеш-ланцюгів в Java і перевіряють редактор ПК, а також мобільний телефон. ПК має такі параметри системи:

• Процесор Pentium 4 3,40 ГГц.

• 1 Гб оперативної пам'яті.

• Операційна система Windows XP.

Мобільний телефон мав наступну конфігурацію:

• 130 процесорів DMIPS ARM7.

• 80 Мб пам'яті.

• Операційна система Symbian OS 7.0S з персональним профілем J2ME.

Експерименти, спрямовані на визначення того, чи можна використовувати наше рішення для автентифікації та створення підписів та перевірки.

Максимальний час, необхідний пристроям, щоб:

• Створити приватний ключ $K_{m\text{-}1}$, коли m збільшується.

• Генеруйте відкритий ключ $g^{K\,m\text{-}1}$ коли $K_{m\text{-}1}$, коли m збільшується.

• Створіть цифровий підпис з $K_{m\text{-}1}$, коли $K_{m\text{-}1}$ збільшується

• Перевірте цифровий підпис з $g^{K\,m\text{-}1}$, коли m збільшується.

Таблиця 2.4. – Параметри, що використовуються для експерименту.

Тип параметра	Значення
Довжина відкритого ключа	1024 біт
Алгоритм підпису	DSA
Функція хешу в односторонньому режимі f	SHA-1
Односторонній хеш-функція h	SHA-1
Мінімальне значення m	10
Максимальне значення m	1000
Довжина s	160 біт

Ми не вимірювали час, необхідний для перевірки відкритого ключа, оскільки це може бути виведено з часу, необхідного для створення відкритого ключа.

Параметри, використані для експериментів, наведені в таблиці 4.

2.4.2. Результати та коментарі

2.4.2.1. Тривалість генерації публічних та приватних ключів.

Генерація приватного ключа швидше, ніж генерація відкритого ключа. Це пояснюється тим, що операція множення менш дорога, ніж операція експонування.

Що значно більше ресурсів центрального процесора та пам'яті, ніж мобільний телефон, то час, який потрібний ПК для генерування ключів:

• Від приватного ключа від 0,5 до 327мс.

• Від 55 мс до 5 с для відкритого ключа.

Хоча час, необхідний мобільному телефону для генерування ключів:

• Від 78 мс до 25 сек. Для приватного ключа.

• Від 4,9 до 472,6 секунди для відкритого ключа.

Ці результати показують, що наше рішення можна використовувати на будь-якому ПК. Вони також показують, що наше рішення можна використовувати на мобільному телефоні за умови вибору відповідного значення для m. m повинен визначатися на підставі строку служби сертифіката, виданого CA і *L*. Якщо, наприклад, CA видає початковий сертифікат, який має термін служби одного року, і якщо політика безпеки вузла вимагає використання нових пар ключів щотижня , то достатньо визначити $m = 52$.

Результати показують, що для $m = 60$ годин, необхідний мобільному телефону для генерації ключів, є:

• 547мс для приватного ключа.

• 28,7 для відкритого ключа.

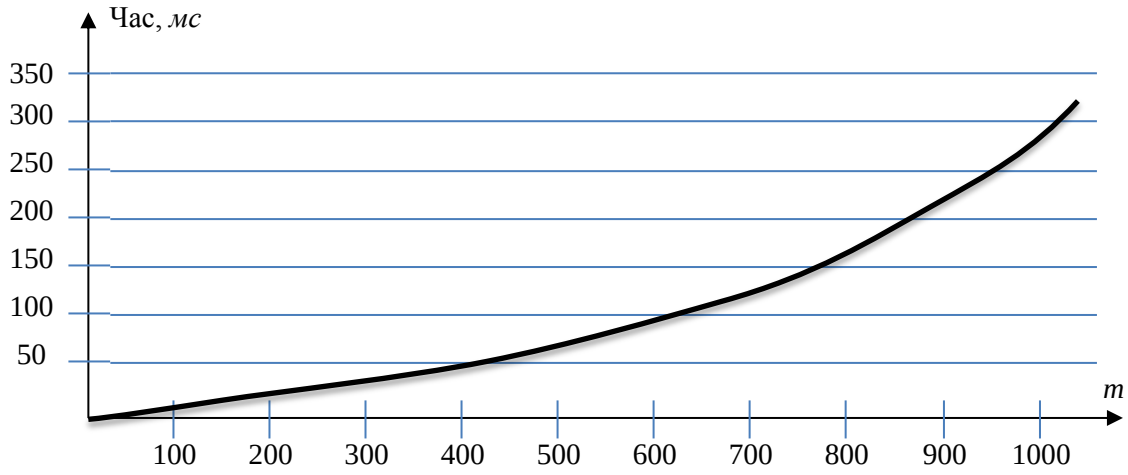

Рисунок 2.7. Тривалість генерації приватних ключів на ПК.

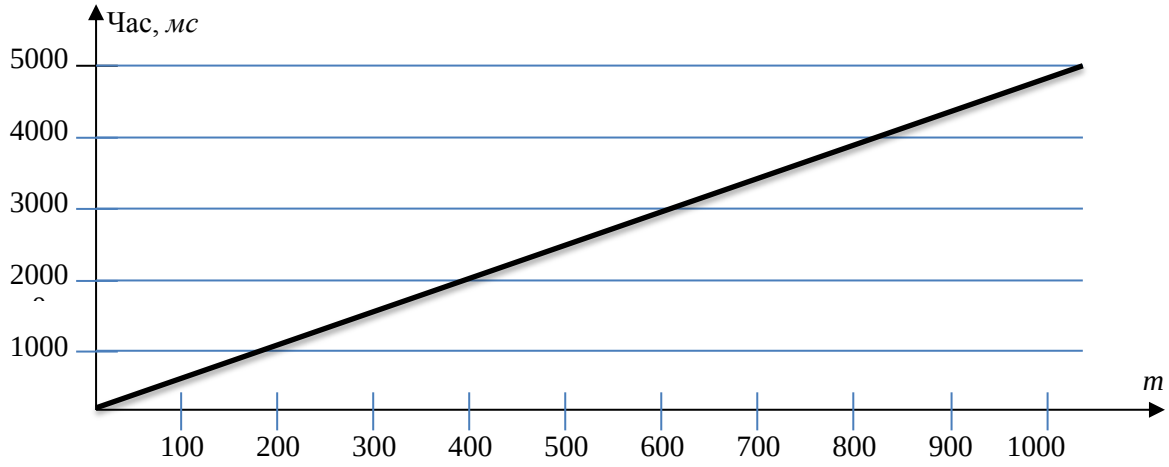

Рисунок 2.8. Тривалість генерації відкритого ключа на ПК.

Генерація публічного ключа на мобільному телефоні досить довга (тобто 28,7 секунди). Однак новітнє покоління мобільних телефонів обладнано процесорами, що мають набагато вищу продуктивність; наприклад, процесор ARM11 MPCore забезпечує до 2600 DMIPS, тобто восьми разів більш потужних, ніж мобільний телефон, який ми використовували для нашої оцінки. Це дозволяє зробити висновок, що генерація ключів та перевірка буде набагато швидше з новими мобільними телефонами.

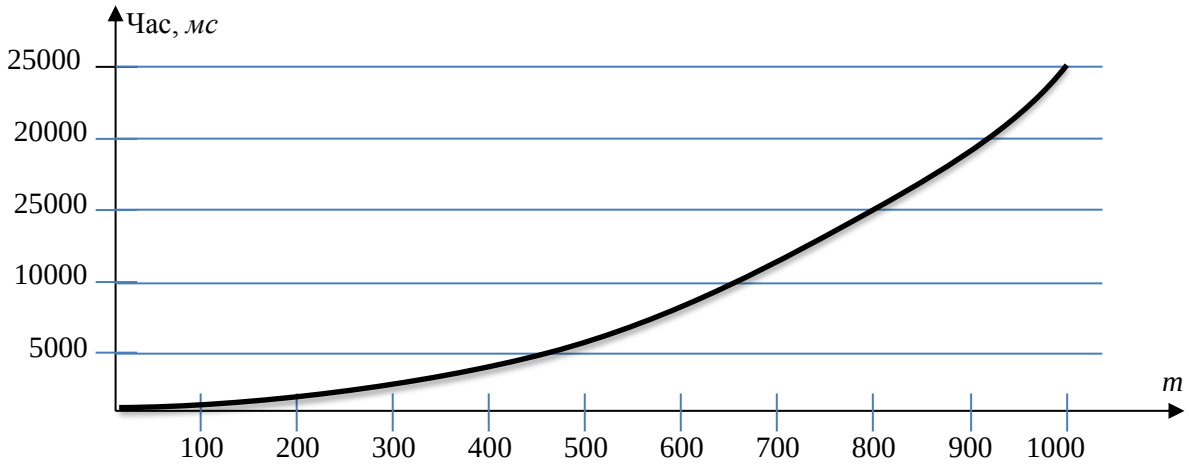

Рисунок 2.9. Тривалість генерації приватних ключів на мобільному телефоні.

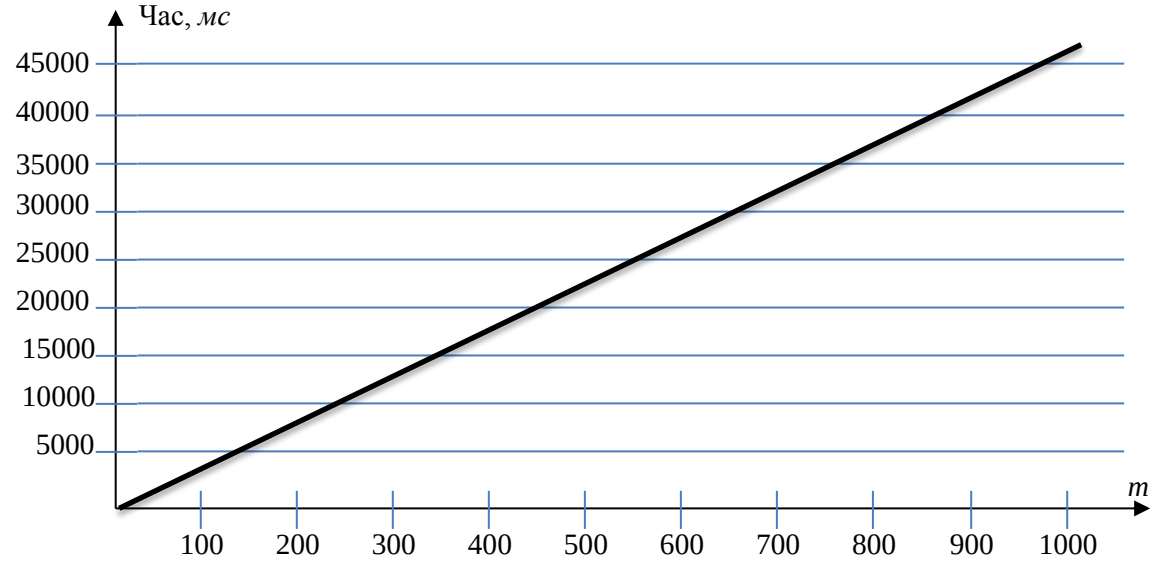

Рисунок 2.10. Тривалість генерації відкритих ключів на мобільному телефоні.

2.4.3. Створення підпису та перевірка

Час, необхідний для створення та перевірки підпису, є важливим, оскільки для вузлів може знадобитися генерувати цифровий підпис для автентифікації або, можливо, буде потрібно перевірити цифровий підпис для автентифікації інших вузлів. Як показано на рисунках 14 і 15, час, необхідний для генерації підпису, зростає повільно, коли m збільшується. На

ПК потрібно менше 9 мс, а на мобільному телефоні - менше 750 мс. Час, необхідний для перевірки підпису, майже стабільний, коли m збільшується. ПК займає менше 12 мс, а на мобільному телефоні - менше 2 секунд. Ці результати показують, що основні пари, створені нашим рішенням, можуть бути використані для створення підпису та перевірки на ПК, а також на мобільному телефоні.

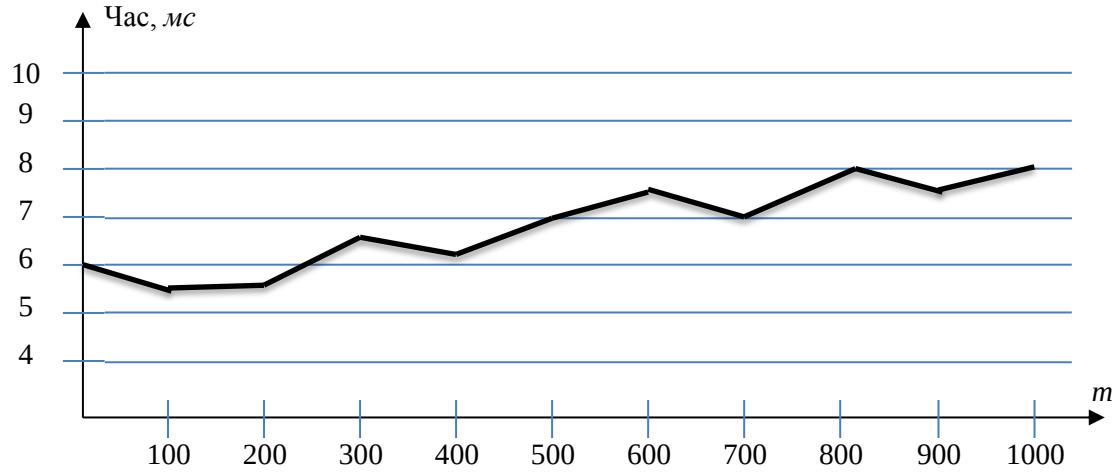

Рисунок 2.11. Тривалість генерації сигнатур на ПК.

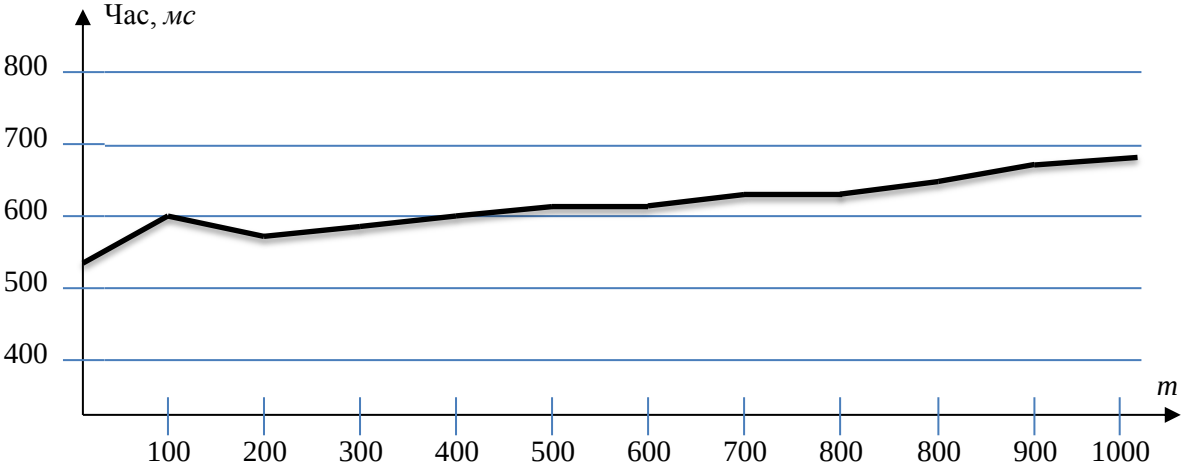

Рисунок 2.12. Тривалість створення підпису на мобільному телефоні.

2.5. Висновки

У спеціальній мережі, вузлам може знадобитися генерувати нові відкриті ключі. У той же час, через відсутність постійної інфраструктури зв'язку, CA, який може видавати сертифікати для цих відкритих ключів, може

бути недоступним. Це може запобігти встановленню безпечних каналів зв'язку. Раніше запропоновані рішення не гарантують, що ідентифікація, що міститься в сертифікатах, є дійсною або, якщо вони це зробити, вони або вимагають, щоб вузли зберігали безліч сертифікатів заздалегідь або покладалися на сусідів, щоб перевіряти прив'язки користувальницьких ключів. Проте для першого підходу може знадобитися великий об'єм пам'яті, тоді як під час другого підходу сусіди не завжди можуть перевірити ідентифікацію вузла перед випуском сертифіката.

У даній роботі було визначенно рішення, яке дозволяє вузлам в спеціальних мережах створювати по запиту декілька пари державних/приватних ключів та сертифікати, не покладаючись на своїх сусідів, без необхідності заздалегідь створювати та зберігати всі ключі та сертифікати, які вони можуть використовувати в майбутньому. Рішення покладається на нову схему, яка дозволяє перевіряється прив'язування пар державних/приватних ключів до єдиного хеш-коду, які існуючі рішення не дозволяють. Наше рішення має приємну властивість, що з часом змінюються загальнодоступні / приватні пари ключів. Хеш-ланцюг використовується для генерації приватних ключів, а від них - для отримання відкритих ключів. Верифікація відкритих ключів є простим, і лише для підтвердження вузла потрібно розкрити поточний діючий відкритий ключ, а також хеш-код. Використовуючи ці розкриваються значення, перевіряючий вузол може оцінити дійсність відкритого ключа, пов'язавши його з перевіреним значенням, включеним в сертифікат перевіряючого вузла. Крім того, як показує наша оцінка продуктивності, відповідність є можливим для обчислювальних пристроїв, таких як мобільні телефони.

Захист схеми хеш-ланцюгів залежить від захисту секретних значень s. З огляду на той факт, що s ніколи не зберігається в системі, але створюється з пароля, біометричної ознаки або інших користувальницьких засобів, коли це необхідно, тоді компрометований вузол може бути використаний тільки до закінчення терміну дії поточної пари публічного / приватного ключів, у

випадку де ці приватні ключі зберігаються в системі. Якщо їх немає, зловмисний вузол може бути використаний неправильно, лише якщо зловмисник вдається розбити відкритий ключ. Проте через криптографічні властивості використовуваних основних елементів, включаючи хеш-функції, хеш-ланцюжки та способи отримання публічних ключів, неможливо розбити ключі або хеш-функції (тобто неможливо з обчисленнями).

РОЗДІЛ 3 ЕКОНОМІЧНИЙ РОЗДІЛ

3.1 Вступ

ІТ-послуга (сервіс) - спосіб надання цінності замовникам через сприяння їм в отриманні результатів на виході, яких замовники хочуть досягти без володіння специфічними витратами і ризиками.

Надання послуг інформаційної безпеки користувачам за розумною ціною визначається трьома факторами: якістю (якість в плані операційної діяльності визначається потужністю, доступністю, продуктивністю, відновленням після надзвичайних обставин, підтримкою), вартістю (витрати і інвестиції), вимогами замовника. вартість і якість повинні відповідати потребам бізнес-користувача.

Складання бюджету починається з планування потреб замовника в послугах і визначення пов'язаних з цим витрат. План або прогноз може складатися на основі аналізу накопичених статистичних даних з урахуванням поточних тенденцій або з використанням даних про витрати на аналогічні сервіси. Моніторинг витрачання фінансових коштів на ІТ-послуги здійснюється з допомогою бухгалтерського та управлінського обліку, для реалізації якого повинна бути розроблена система другого кола рахунків. Витрати на ІТ послуги повинні оцінюватися по тому, наскільки вони сприяють досягненню бізнес-переваг організації в цілому і можуть бути класифіковані різними способами

3.2. Визначення витрат на проектування та експлуатацію систем інформаційної безпеки

Розрахунок (фіксованих) капітальних витрат

Капітальні інвестиції – це кошти, призначені для створення і придбання основних фондів і нематеріальних активів, що підлягають амортизації.

До фіксованих (капітальних) відносяться наступні витрати:

- вартість розробки проекту інформаційної безпеки (розробка схем пристоїв, політики функціонування системи тощо);
- витрати на залучення зовнішніх консультантів;

- вартість первісних закупівель ліцензійного основного й додаткового програмного забезпечення (ПЗ);
- вартість створення основного й додаткового програмного забезпечення (ПЗ);
- витрати на первісні закупівлі апаратного забезпечення;
- витрати на інтеграцію системи інформативної безпеки у вже існуючу корпоративну систему (встановлення обладнання, програмного забезпечення та налагодження системи інформаційної безпеки);
- витрати на навчання технічних фахівців і обслуговуючого персоналу.

3.2.1. Визначення витрат на створення програмного засобів захисту інформації

При виконанні дипломних проектів, спрямованих на розробку і використання програмного забезпечення (ПЗ) в системах інформаційної безпеки, техніко-економічні розрахунки мають містити:

- визначення трудомісткості розробки та опрацювання ПЗ;
- розрахунок витрат на створення програмного продукту;
- оцінку швидкодії та надійності роботи програмного продукту.

3.2.2. Визначення трудомісткості розробки та опрацювання програмного продукту

Трудомісткість створення ПЗ визначається тривалістю кожної робочої операції, починаючи з складання технічного завдання і закінчуючи оформленням документації (за умови роботи одного програміста):

$$t = tmз + tв + ta + tпр + tопр + tд, \text{ годин,} \quad (3.1)$$

де $tmз$ – тривалість складання технічного завдання на розробку ПЗ;

$tв$ – тривалість вивчення ТЗ, літературних джерел за темою тощо;

ta – тривалість розробки блок-схеми алгоритму;

$tпр$ – тривалість програмування за готовою блок-схемою;

$tопр$ – тривалість опрацювання програми на ПК;

$t_д$ – тривалість підготовки технічної документації на ПЗ.

Звідси, $t = 40 + 16 + 16 + 32 + 16 + 24 = 144$ години

Складові трудомісткості визначаються на підставі умовної кількості операторів у програмному продукті ***Q*** (з урахуванням можливих уточнень у процесі роботи над алгоритмом і програмою).

Умовна кількість операторів у програмі:

$$Q = q \cdot c\ (1 + p), \quad \text{штук}, \tag{3.2}$$

де q – очікувана кількість операторів;

c – коефіцієнт складності програми;

p – коефіцієнт корекції програми в процесі її опрацювання.

Звідси, $Q = 3 \cdot 1{,}5\ (1 + 0{,}07) \approx 5$, штук

Тривалість вивчення технічного завдання, опрацювання довідкової літератури з урахуванням уточнення ТЗ і кваліфікації програміста можливо оцінити за формулою:

$$t_в = \frac{Q \cdot B}{(75...85) \cdot k}\ , \text{годин}, \tag{3.3}$$

де B – коефіцієнт збільшення тривалості етапу внаслідок недостатнього опису завдання, В = 1,3;

k – коефіцієнт, що враховує кваліфікацію програміста і визначається стажем роботи за фахом:

- до 2 років – 0,8;
- від 2 до 3 років – 1,0;
- від 3 до 5 років – 1,1...1,2;
- від 5 до 7 років – 1,3...1,4;
- понад 7 років – 1,5...1,6.

Отже, $t_в = \frac{5 \cdot 1{,}3}{80 \cdot 1{,}0} = 0{,}081$, годин,

Тривалість розробки блок-схеми алгоритму:

$$ta = \frac{Q}{(20...25) \cdot k}\ , \text{годин}. \tag{3.4}$$

$$ta = \frac{Q}{(20...25)\cdot k} = \frac{5}{20\cdot 1{,}0} = 0{,}25 \text{ , годин.}$$

Тривалість складання програми за готовою блок-схемою:

$$tпр = \frac{Q}{(20...25)\cdot k}, \text{ годин.} \quad (3.5)$$

$$tпр = \frac{Q}{(20...25)\cdot k} = \frac{5}{20\cdot 1{,}0} = 0{,}25, \text{ годин.}$$

Тривалість опрацювання програми на ПК:

$$tопр = \frac{1{,}5Q}{(4...5)\cdot k} \text{ годин.} \quad (3.6)$$

$$tопр = \frac{1{,}5Q}{(4...5)\cdot k} = \frac{1{,}5\cdot 5}{5\cdot 1{,}0} = 1{,}5 \text{ годин.}$$

Тривалість підготовки технічної документації на ПЗ:

$$t_{д} = \frac{Q}{(15...20)\cdot k} + \frac{Q}{(15...20)}\cdot 0{,}75 \quad (3.7)$$

$$t_{д} = \frac{5}{15\cdot 1{,}0} + \frac{5}{15}\cdot 0{,}75 = 0{,}58 \text{ годин}$$

3.1.3 Розрахунок витрат на створення програмного продукту

Витрати на створення програмного продукту *Кпз* складаються з витрат на заробітну плату виконавця програмного забезпечення *Ззп* і вартості витрат машинного часу, що необхідний для опрацювання програми на ПК *Змч*:

$$К_{пз} = З_{зп} + З_{мч}. \quad (3.8)$$

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

$$З_{зп} = t \cdot З_{пр} \text{ грн,} \quad (3.9)$$

$$З_{зп} = t \cdot З_{пр} = 144\cdot 100 = 14400 \text{ грн,}$$

де t – загальна тривалість створення ПЗ, годин;

Зпр – средньогодинна заробітна плата програміста з нарахуваннями, грн/годину.

Вартість машинного часу для налагодження програми на ПК визначається за формулою:

$$З_{мч} = t_{опр} \cdot C_{мч} + t_{д}\ ,\ \text{грн,} \qquad (3.10)$$

де $t_{опр}$ – трудомісткість налагодження програми на ПК, годин;

$t_д$ – трудомісткість підготовки документації на ПК, годин;

$C_{мч}$ – вартість 1 години машинного часу ПК, грн./година.

Вартість 1 години машинного часу ПК визначається за формулою:

$$C_{мч} = P\ \cdot t_{нал} \cdot C_e + \frac{Ф_{зал} \cdot Н_а}{F_р} + \frac{К_{лпз} \cdot Н_{апз}}{F_р}\ ,\ \text{грн,} \qquad (3.11)$$

де P – встановлена потужність ПК, кВт;

$C_е$ – тариф на електричну енергію, грн/кВт·година;

$Ф_{зал}$ – залишкова вартість ПК на поточний рік, грн.;

$Ф_{зал}$=10000

$Н_а$ – річна норма амортизації на ПК, частки одиниці;

$Н_{апз}$ – річна норма амортизації на ліцензійне програмне забезпечення, частки одиниці;

$К_{лпз}$ – вартість ліцензійного програмного забезпечення, грн.;

$F_р$ – річний фонд робочого часу (за 40-годинного робочого тижня $F_р$ = 1920.

$$C_{мч} = P\ \cdot t_{нал} \cdot C_e + \frac{Ф_{зал} \cdot Н_а}{F_р} + \frac{К_{лпз} \cdot Н_{апз}}{F_р} =$$

$$= 0{,}5 \cdot 144 \cdot 0{,}168 + \frac{10000 \cdot 0{,}2}{1920} + \frac{1120 \cdot 1}{1920} = 12{,}1 + 5{,}02 + 0{,}58 = 13{,}72\ \text{грн}$$

Залишкова вартість ПК визначається виходячи з фактичного терміну його експлуатації як різниця між первісною вартістю та зносом за час використання.

Визначена таким чином вартість створення програмного забезпечення *Кпз* є частиною одноразових капітальних витрат разом з витратами на придбання і налагодження апаратури системи інформаційної безпеки.

Таким чином, капітальні (фіксовані) витрати на проектування та впровадження проектного варіанта системи інформаційної безпеки складають:

$$К = К_{пр} + К_{зпз} + К_{пз} + К_{аз} + К_{навч} + К_{н}\ , \qquad (3.12)$$

де $К_{пр}$ – вартість розробки проекту інформаційної безпеки та залучення для цього зовнішніх консультантів, тис. грн;

$К_{пр}$=15000

$К_{зпз}$ – вартість закупівель ліцензійного основного й додаткового програмного забезпечення (ПЗ), тис. грн;

$К_{зпз}$ додатково не закупались

$К_{пз}$ – вартість створення основного й додаткового програмного забезпечення, тис. грн;

$К_{аз}$ – вартість закупівлі апаратного забезпечення та допоміжних матеріалів, тис. грн;

$К_{аз}$=40000, а саме Ноутбук MSI GP63

$К_{навч}$ – витрати на навчання технічних фахівців і обслуговуючого персоналу, тис. грн;

$К_{навч}$ відсутні

$К_{н}$ – витрати на встановлення обладнання та налагодження системи інформаційної безпеки, тис. грн.

$К_{н}$ відсутні

Аналогічно визначаються капітальні витрати альтернативного варіанта $К_а$, якщо вирішення поставленої задачі може бути досягнуто іншим способом.

$$К = К_{пр} + К_{зпз} + З_{зп} + З_{мч} + К_{аз} + К_{навч} + К_{н} =$$

$$= К_{пр} + З_{зп} + t_{опр} \cdot С_{мч} + t_{д} + К_{аз} =$$

$$= 15000 + 14400 + 16 \cdot 13{,}72 + 24 + 40000 + = 69643{,}52$$

3.3. Розрахунок поточних (експлуатаційних) витрат

Експлуатаційні витрати – це поточні витрати на експлуатацію та обслуговування об'єкта проектування за визначений період (наприклад, рік), що виражені у грошовій формі.

Річні поточні (експлуатаційні) витрати на функціонування системи інформаційної безпеки складають:

$$C = C_в + C_к + C_{ак}\ ,\quad \text{тис. грн.} \qquad (3.13)$$

Витрати на Upgrade-відновлення й модернізацію системи інформаційної безпеки $C_в = 20000$ грн.

Витрати на керування системою інформаційної безпеки ($C_к$) складають:

$$C_к = C_н + C_а + C_з + C_{єв} + C_{ел} + C_о + C_{тос}, \text{ грн.} \qquad (3.14)$$

Витрати на навчання адміністративного персоналу й кінцевих користувачів визначаються за даними організації з проведення тренінгів персоналу, курсів підвищення кваліфікації тощо ($C_н$) відсутні.

Річний фонд амортизаційних відрахувань ($C_а$) визначається у відсотках від суми капітальних інвестицій за видами основних фондів і нематеріальних активів (ПЗ)

$$C_а = K_{аз} \cdot 0{,}1, \text{ грн.} \qquad (3.15)$$

$C_а = 40000 \cdot 0{,}1 = 4000$ грн.

Річний фонд заробітної плати інженерно-технічного персоналу, що обслуговує систему інформаційної безпеки ($C_з$), складає:

$$C_з = З_{осн} + З_{дод}\ , \text{ грн.} \qquad (3.16)$$

$З_{осн} = 150000$

$З_{дод} = 12500$

$C_з = З_{осн} + З_{дод} = 150000 + 12500 = 175000$ грн.

де $З_{осн}$, $З_{дод}$ – основна і додаткова заробітна плата відповідно, грн на рік.

Розмір єдиного внеску на загальнообов'язкове державне соціальне страхування визначається $C_{єв}$.

$$C_{єв} = C_з \cdot 22\% \text{ грн.} \qquad (3.17)$$

$C_{єв} = C_з \cdot 8,4$ $C_{єв}$= 175000 · 22% = 38500 грн

Вартість електроенергії, що споживається апаратурою системою інформаційної безпеки протягом року ($C_{ел}$), визначається за формулою:

$$C_{ел} = P \cdot F_p \cdot Ц_е \text{ , грн,} \qquad (3.18)$$

де P – встановлена потужність апаратури інформаційної безпеки, кВт;

F_p – річний фонд робочого часу системи інформаційної безпеки (2080);

$Ц_е$ – тариф на електроенергію, грн/кВт·годин.

$C_{ел} = P \cdot F_p \cdot Ц_е = 0,6 \cdot 2080 \cdot 1,64 =$ 2046,72 грн

Витрати на залучення сторонніх організацій для виконання деяких видів обслуговування, навчання та сертифікацію обслуговуючого персоналу (C_o) відсутні.

Витрати на технічне й організаційне адміністрування та сервіс системи інформаційної безпеки $C_{тос}$ = 20000

Витрати, викликані активністю користувачів системи інформаційної безпеки ($C_{ак}$) відсутні.

Отже $C_к = C_а + C_з + C_{єв} + C_{ел} + C_{тос} =$

4000 + 175000 + *38500* + 2046,72 + *20000* = 239546,72 грн.

Тоді $C = C_в + C_к$ = *20000* + 239546,72 = 256221,72 грн.

У кожному конкретному випадку можуть бути враховані й інші види поточних витрат, що визначаються специфікою експлуатації проектованої системи інформаційної безпеки.

3.4. Оцінка можливостей збитку від атаки (злому)

Кінцевим результатом впровадження й проведення заходів щодо забезпечення інформаційної безпеки є величина *відвернених втрат*, що розраховується, виходячи з імовірності виникнення інциденту інформаційної безпеки й можливих економічних втрат від нього. По суті, ця величина відображає ту частину прибутку, що могла бути втрачена.

3.4.1. Оцінка величини збитку

Загалом можливо виділити такі види збитку, що можуть вплинути на ефективність комп'ютерної системи інформаційної безпеки (КСІБ):

- порушення конфіденційності ресурсів КСІБ (тобто неможливість доступу до них неавторизованих суб'єктів або несанкціонованого використання каналів зв'язку);
- порушення доступності ресурсів КСІБ (тобто можливість доступу до них авторизованих суб'єктів (завжди, коли їм це потрібно);
- порушення цілісності ресурсів КСІБ (тобто їхня неушкодженість);
- порушення автентичності ресурсів КСІБ (тобто їхньої дійсності, непідробленості).

Для розрахунку вартості збитку можна застосувати наступну спрощену модель оцінки.

Необхідні вихідні дані для розрахунку:

$t_п$ – час простою вузла або сегмента корпоративної мережі внаслідок атаки, годин;

$t_в$ – час відновлення після атаки персоналом, що обслуговує корпоративну мережу, годин;

$t_{ви}$ – час повторного введення загубленої інформації співробітниками атакованого вузла або сегмента корпоративної мережі, годин;

$З_о$ – заробітна плата обслуговуючого персоналу (адміністраторів та ін.), грн на місяць;

$З_о$ = 10000 тис.грн.

$З_с$ – заробітна плата співробітників атакованого вузла або сегмента корпоративної мережі, грн на місяць;

$З_с$ = 9·10000 = 90000 тис.грн.

$Ч_о$ – чисельність обслуговуючого персоналу (адміністраторів та ін.), осіб.;

$Ч_с$ – чисельність співробітників атакованого вузла або сегмента корпоративної мережі, осіб.;

$О$ – обсяг продажів атакованого вузла або сегмента корпоративної мережі, грн у рік;

$П_{зч}$ – вартість заміни встаткування або запасних частин, грн;

I – число атакованих вузлів або сегментів корпоративної мережі;

N – середнє число атак на рік.

Упущена вигода від простою атакованого вузла або сегмента корпоративної мережі становить:

$$U = П_п + П_в + V, \quad (3.19)$$

де $П_п$ – оплачувані втрати робочого часу та простої співробітників атакованого вузла або сегмента корпоративної мережі, грн;

$П_в$ – вартість відновлення працездатності вузла або сегмента корпоративної мережі (переустановлення системи, зміна конфігурації та ін.), грн;

V – втрати від зниження обсягу продажів за час простою атакованого вузла або сегмента корпоративної мережі, грн.

Втрати від зниження продуктивності співробітників атакованого вузла або сегмента корпоративної мережі являють собою втрати їхньої заробітної плати (оплата непродуктивної праці) за час простою внаслідок атаки:

$$П_п = \frac{\sum Зс}{F} \cdot t_п \ , \quad (3.20)$$

$$П_п = \frac{\sum Зс}{F} \cdot t_п = \frac{90000}{176} \cdot 12 = 7140$$

де *F* – місячний фонд робочого часу (при 40-а годинному робочому тижні становить 176 ч).

Витрати на відновлення працездатності вузла або сегмента корпоративної мережі включають кілька складових:

$$П_в = П_{ви} + П_{пв} + П_{зч} \quad (3.21)$$

де $П_{ви}$ – витрати на повторне уведення інформації, грн;

$П_{пв}$ – витрати на відновлення вузла або сегмента корпоративної мережі, грн;

$П_{зч}$ – вартість заміни устаткування або запасних частин, грн.

Витрати на повторне введення інформації $П_{ви}$ розраховуються виходячи з розміру заробітної плати співробітників атакованого вузла або

сегмента корпоративної мережі $З_с$, які зайняті повторним введенням втраченої інформації, з урахуванням необхідного для цього часу $t_{ви}$:

$$П_{ви} = \frac{\sum Зс}{F} \cdot t_{ви} \ . \quad (3.22)$$

$$П_{ви} = \frac{90000}{176} \cdot 12 = 7140 \text{ грн.}$$

Витрати на відновлення вузла або сегмента корпоративної мережі $П_{пв}$ визначаються часом відновлення після атаки $t_в$ і розміром середньогодинної заробітної плати обслуговуючого персоналу (адміністраторів):

$$П_{пв} = \frac{\sum Зо}{F} \cdot t_в \ . \quad (3.23)$$

$$П_{пв} = \frac{10000}{176} \cdot 12 = 682 \text{ грн.}$$

Тому $П_в = П_{ви} + П_{пв} + П_{зч} = 7140 + 682 + 10000 = 17822$ грн

Втрати від зниження очікуваного обсягу продажів за час простою атакованого вузла або сегмента корпоративної мережі визначаються виходячи із середньогодинного обсягу продажів і сумарного часу простою атакованого вузла або сегмента корпоративної мережі:

$$V = \frac{O}{F_г} \cdot (t_п + t_в + t_{ви}), \quad (3.24)$$

$$V = \frac{3 \cdot 10^6}{2080} \cdot (12 + 12 + 12) = \frac{3 \cdot 10^6}{2080} \cdot 36 = 51923$$

де $F_г$ – річний фонд часу роботи організації (52 робочих тижні, 5-ти денний робочий тиждень, 8-ми годинний робочий день) становить близько 2080 ч.

Отже, $U = П_п + П_в + V = 7140 + 17822 + 51923 = 76885$

Загальний збиток від атаки на вузол або сегмент корпоративної мережі організації розраховується

$$B = \sum_i \sum_n U \quad (3.25)$$

Таким чином, $B = \sum_i \sum_n U = 2 \cdot 3 \cdot 76885 = 461310$

3.5. Загальний ефект від впровадження системи інформаційної безпеки

Загальний ефект від впровадження системи інформаційної безпеки визначається з урахуванням ризиків порушення інформаційної безпеки і становить:

$$E = B \cdot R - C, \qquad (3.26)$$

де B – загальний збиток від атаки на вузол або сегмент корпоративної мережі, тис. грн;

R – очікувана імовірність атаки на вузол або сегмент корпоративної мережі, частки одиниці;

C – щорічні витрати на експлуатацію системи інформаційної безпеки, тис. грн.

$$E = B \cdot R - C = 461310 \cdot 80\% - 256221{,}72 = 245088{,}28 \text{ грн.}$$

3.6. Визначення та аналіз показників економічної ефективності системи інформаційної безпеки

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

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

$$ROSI = \frac{E}{K}, \quad \text{частки одиниці,} \qquad (3.27)$$

$$ROSI = \frac{E}{K} = \frac{112836{,}28}{69643{,}52} = 1{,}62$$

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

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

1. Якщо організація здійснює фінансування капітальних інвестицій у систему інформаційної безпеки за рахунок позикових коштів, тобто за рахунок банківського кредиту, то в якості бажаного значення $E_н$ варто приймати величину плати за кредит (кредитної ставки) $N_{кр}$.

Проект визнається економічно доцільним, якщо розрахункове значення коефіцієнта повернення інвестицій перевищує величину банківської кредитної ставки з урахуванням інфляції:

$$ROSI > (N_{кр} + N_{інф})/100, \quad (3.28)$$

де $N_{кр}$ – банківська кредитна ставка, %;

$N_{інф}$ – річний рівень інфляції, %.

$(8,9 + 108,9)/100 = 1,178$

$1,62 > 1,178$

2. Термін окупності капітальних інвестицій То показує, за скільки років капітальні інвестиції окупляться за рахунок загального ефекту від впровадження системи інформаційної безпеки:

$$T_O = \frac{K}{E} = \frac{1}{ROSI}, \quad \text{років.} \quad (3.29)$$

$T_O = \frac{1}{ROSI} = 0,62$ років

3.7 Висновки

В економічній частині дипломного проекту обґрунтовано доцільність розробки нового програмного продукту, розраховано його собівартість (256221,72 грн, розрахована загальна сума збитки при відсутності запровадження (461310 грн).

Проаналізувавши основні техніко-економічні характеристики нового ПП та аналогу, виявлено, що за кількістю наявних функції розроблений ПП рівний аналогу, це: вартість ПП, простота та зручність інтерфейсу, вага ПП, час виконання запиту, можливість нарощування функціональних характеристик та час відновлення системи після збою. Проте, кількість людей, необхідних для обслуговування усього процесу за допомогою розробленого ПП, є більшою за аналог.

При виконанні економічної частини, зробивши всі необхідні розрахунки, встановлено, що технологія розробки програмного продукту відповідає оптимальному рівню витрат, і, в кінцевому підсумку, розроблений

продукт є економічно доцільним та конкурентоспроможним для впровадження в технічному відділі підприємства та окупається замовником за менше ніж рік.

ВИСНОВКИ

У дипломній роботі було проведенно створення сертифікованої аутентифікації відкритих ключів у спеціальній мережі Ad-hoc.

Децентрализованная беспроводная сеть, не имеющая постоянной структурі сьогодні є одним з популярніших видів зв’язку, а тому має велику необхідність створення аутентифікації для постійної роботи у мережі без втрати важливих данних.

При кожній атаці на мережу трачається та пошкоджується богато інформації, яка має свою ціну, тому кожнному користувачу мережі важливо її захистити. Безпека є важливою для спеціальних мереж, особливо для програм, чутливих до уражень. При захисті мережи, є наступні атрибути: доступність, конфіденційність, цілісність, автентифікація та відсутність відмови.

У економічному розділі розраховано вартість розробки аутентифікації та фінансові втрати кожної атаки.

ПЕРЕЛІК ПОСИЛАНЬ

1 Методичні рекомендації до виконання дипломних робіт (проектів) бакалаврів та магістрів спеціальностей 125 Кібербезпека, 172 Телекомунікації та радіотехніка / Упоряд.: О.Ю. Гусєв, О.В. Герасіна, О.М. Алексєєв, О.В. Кручінін. – Дніпро: НГУ, 2018. – 50 с.

2 Методичні вказівки до виконання економічної частини дипломного проекту зі спеціальності 125 Кібербезпека / Упорядн.: І.В. Шереметьєва, Д.П. Пілова, Н.М. Романюк. – Дніпро: Національний технічний університет "Дніпровська політехніка", 2017. – 17 с.

3 М. Кастро та Б. Лісков. Практична візантійська відмова від толерантності. У працях третього USENIX Симпозіум з розробки та впровадження операційних систем (OSDI'99), стор 173-186, Новий Орлеан, LA США, 22-25 лютого 1999 р. Асоціація USENIX, IEEE TCOS та ACM SIGOPS.

4 Р. Деннаро, С. Ярецкі, Х. Кравчик, Т. Рабін. Надійний порог DSS підписів. В UM Maurer, редактор, Advances in Cryptology-Eurocrypt'96, Міжнародна конференція з теорії та застосування криптографічних методів, Сарагоса, Іспанія, 12-16 травня 1996 р., Праці, том 1233 з лекційних заміток в галузі інформатики, стор 354- 371. Springer, 1996.

5 В. Н. Денисов. Ад гок // Юридична енциклопедія : [в 6-ти т.] / ред. кол. Ю. С. Шемшученко (відп. ред.) [та ін.]. — К. : Українська енциклопедія, 1998. — ISBN 966-749-200-1.

6 Сангили К, Д Лафламм, Дахилл В, Левін В, Щити С, Белдінг-Ройер Е. ідентифіковані маршрутизації для мереж. IEEE Journal on Selected Areas in Communications 2005; 23 (3): 598--610.

7 Рая М, Хубау JP. Захист автотранспортних спеціальних мереж. SASN'05: Матеріали третього семінару ACM з безпеки спеціальних та сенсорних мереж, ACM Press: Нью-Йорк, штат Нью-Йорк, США, 2005; 11-21. DOI: http://doi.acm.org/10.1145/1102219.1102223.

8 Ло Хо, Зерфос П, Конг Д, Лу С, Чжан Л. Самозахищені спеціальні бездротові мережі. ISCC '02: Матеріали Сьомого Міжнародного симпозіуму "Комп'ютери та зв'язок" (ISCC'02). Комп'ютерне товариство IEEE: Вашингтон, округ Колумбія, США, 2002; 567-557.

9 Конг Дж, Зерфос П, Ло Хо, Лу С, Чжан Л. Надання надійної та повсюдної підтримки безпеки мобільних спеціальних мереж. ICNP'01: Матеріали дев'ятої міжнародної конференції про мережеві протоколи, IEEE Computer Society: Вашингтон, округ Колумбія, США, 2001; 251.

10 Капкун С, Буттян Л, Хубаус Дж. П. Самоорганізоване управління загальним ключем для мобільних спеціальних мереж. Транзакції IEEE на мобільних комп'ютерах 2003; 2 (1): 52--64. DOI: http://dx.doi.org/10.1109/TMC.2003.1195151.

11 Лі Р, Лі Дж, Камеда Х, Лю П. Локалізоване управління публічними ключами для мобільних спеціальних мереж. У роботі Глобальної телекомунікаційної конференції 2004 р. Т. 2, 2004; 1284-1289.

12 Gagné M. Шифрування на базі ідентичності: ассурвання. RSA Laboratories CryptoBytes 2003; 6: 10--19.

13 Weimerskirch A, Thonet G. Розповсюджена легка модель автентифікації для спеціальних мереж. ICISC '01: Матеріали 4-ї міжнародної конференції "Сеул з інформаційної безпеки та криптології", Springer-Verlag, Лондон, Великобританія, 2002; 341--354.

14 Ngai E, Lyu M. Служби автентифікації на основі довіри та кластеризації в мобільних спеціальних мережах. ICDCSW'04: Матеріали 24-ї Міжнародної конференції з семінарів з розподілених обчислювальних систем - W7: EC (ICDCSW'04). Комп'ютерне суспільство IEEE: Вашингтон, округ Колумбія, США, 2004; 582-587.

15 Нгай Е, Лю М, Чин Р.Т. Служба автентифікації проти нечесних користувачів у мобільних спеціальних мережах. Праці 2004 р. Авіаційно-космічної конференції IEEE, т. 2, 2004; 1275--1285.

16 Weimerskirch A, Westhoff D. Identity certified authentication for ad hoc networks. SASN ’03: Proceedings of the 1st ACM Workshop on Security of Ad hoc and Sensor Networks, ACM Press, New York, NY: New York, NY, USA, 2003; 33--40. DOI: http://doi.acm.org/10.1145/986858.986864.

17 Lamport L. Password authentication with insecure communication. Communications of the ACM 1981; 24(11): 770--772. DOI: http://doi.acm.org/10.1145/358790.358797.

18 PerrigA,CanettiR,TygarJ,SongD.TheTESLAbroadcast authentication protocol. CryptoBytes 2002; 5(2): 2--13.

19 Merkle RC. A digital signature based on a conventional encryption function. Advances in Cryptology -- CRYPTO ’87, Lecture Notes in Computer Science, Vol. 293, Pomerance C (ed). Springer--Verlag: Berlin, 1988; 369--378.

20 Deng H, Mukherjee A, Agrawal DP. Threshold and identity-based key management and authentication for wireless ad hoc networks. ITCC ’04: Proceedings of the International Conference on Information Technology: Codingand Computing (ITCC’04), Volume2, IEEE Computer Society: Washington, DC, USA, 2004; 107--111.

21 Bohio MJ, Miri A. An authenticated broadcasting scheme for wireless ad hoc network. CNSR ’04: Proceedings of the Second Annual Conference on Communication Networksand Services Research(CNSR’04), IEEE Computer Society: Washington, DC, USA, 2004; 69--74.

22 Лампорт Л. Пароль аутентифікації з небезпечним спілкуванням. Повідомлення від ACM 1981; 24 (11): 770--772. DOI: http://doi.acm.org/10.1145/358790.358797

ДОДАТОК А.Відомість матеріалів дипломної роботи

№	Формат	Найменування	Кількість листів
Документація			
1	А4	Реферат	3
2	А4	Список умовних скорочень	1
3	А4	Вступ	7
4	А4	Стан питання. Постановка задачі	7
5	А4	Спеціальна частина	30
6	А4	Економічний розділ	13
7	А4	Висновки	1
8	А4	Перелік посилань	3
9	А4	Додаток А	1