МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
КАФЕДРА КОМП'ЮТЕРИЗОВАНИХ СИСТЕМ ЗАХИСТУ ІНФОРМАЦІЇ

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач кафедри Комп'ютеризованих систем захисту інформації

_____ Михайло СТЕПАНОВ
«____»_____ 2023 р.

На правах рукопису
УДК 004.056.5:510.22(004.05)

КВАЛІФІКАЦІЙНА РОБОТА

ЗДОБУВАЧА ВИЩОЇ ОСВІТИ
ОСВІТНЬОГО СТУПЕНЯ «МАГІСТР»

Тема: Модель захисту онлайн-повідомлень

Виконавець: Юлія КУЧЕРЕНКО
Керівник: д.т.н., доцент Людмила ТЕРЕЙКОВСЬКА
Консультант розділу «Охорона навколишнього середовища»: к.т.н., доцент Тетяна ДМИТРУХА
Нормоконтролер: д.т.н., доцент Людмила ТЕРЕЙКОВСЬКА

Київ 2023

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет: Кібербезпеки та програмної інженерії

Кафедра: Комп'ютеризованих систем захисту інформації

Освітній ступінь: Магістр

Спеціальність: 125 «Кібербезпека»

Освітньо-професійна програма: «Безпека інформаційних і комунікаційних систем»

ЗАТВЕРДЖУЮ
Завідувач кафедри Комп'ютеризованих систем захисту інформації

_____ Михайло СТЕПАНОВ
«__» _____ 2023 р.

ЗАВДАННЯ
на виконання кваліфікаційної роботи
здобувача вищої освіти Кучеренко Юлії Іванівни

1. Тема: *Модель захисту онлайн-повідомлень*
затверджена наказом ректора від «15» вересня 2023 р. № 1814/ст.
2. Термін виконання: з 16.10.2023 р. по 31.12.2023 р.
3. Вихідні дані: проаналізувати існуючі методи та засоби захисту онлайн-повідомлень; розробити модель захисту онлайн-повідомлень; реалізувати програмне забезпечення на основі розробленої моделі.
4. Зміст пояснювальної записки: аналіз сучасних досліджень у сфері захисту онлайн-повідомлень; розробка моделі захисту онлайн-повідомлень; розробка програмного забезпечення та дослідження розробленої системи.

5. КАЛЕНДАРНИЙ ПЛАН

виконання кваліфікаційної роботи

№ з/п	Етапи виконання кваліфікаційної роботи	Термін виконання етапів	Примітка
1.	Уточнення постановки задачі	16.10.2023	*Виконано*
2.	Аналіз літературних джерел	20.10.2023	*Виконано*
3.	Обгрунтування вибору рішення	30.10.2023	*Виконано*
4.	Збір інформації	01.11.2023	*Виконано*
5.	Аналіз сучасних досліджень у сфері захисту онлайн-повідомлень	07.11.2023	*Виконано*
6.	Розробка моделі захисту онлайн-повідомлень	14.11.2023	*Виконано*
7.	Розробка програмного забезпечення та дослідження розробленої системи	21.11.2023	*Виконано*
8.	Апробація роботи на міжнародній науково-практичній конференції «ЖИВУЧІСТЬ ТА РЕЗИЛЬЄНТНІСТЬ – 2023»	19.10.2023	*Виконано*
9.	Перевірка на антиплагіат	11.12.2023	*Виконано*
10.	Оформлення і друк пояснювальної записки	16.12.2023	*Виконано*
11.	Оформлення презентації	16.12.2023	*Виконано*
12.	Отримання рецензій від рецензента	22.12.2023	*Виконано*

6. Консультанти з окремих розділів

Розділ	Консультант (посада, П.І.Б.)	Дата, підпис	
		Завдання видав	Завдання прийняв
Охорона навколишнього середовища	Дмитруха Т.І.		

7. Дата видачі завдання: «16» жовтня 2023 р.

Здобувач вищої освіти Юлія КУЧЕРЕНКО

(підпис, дата)

Керівник кваліфікаційної роботи Людмила ТЕРЕЙКОВСЬКА

(підпис, дата)

РЕФЕРАТ

Кваліфікаційна робота на тему: «Модель захисту онлайн-повідомлень» складається зі вступу, основної частини, що містить 4 розділи, 3 висновки до кожного розділу, загального висновку, 1 додаток та списку використаної літератури. Загальний обсяг роботи – 96 сторінок. Робота містить 16 рисунків та 2 формули. Список використаних джерел включає 65 джерел.

Метою кваліфікаційної роботи є реалізація ефективної моделі захисту онлайн-повідомлень.

У кваліфікаційній роботі розглянуті питання щодо сучасних методів захисту онлайн-повідомлень.

Модель захисту онлайн-повідомлень базується на криптографічному пакеті з відкритим ключем, призначений для публічного використання – PGP та виконує автоматичне шифрування, дешифрування, підписання та перевірку цілісності. Вона також автоматично шукає відкриті ключі одержувачів на сервері ключів PGP і закриті ключі поточного користувача в Active Directory.

Розроблена модель відносяться до галузі інформаційної безпеки і може бути використана для невеликого підприємства.

Запропонована модель дозволяє забезпечити конфіденційність, автентичність та цілісність онлайн-повідомлень.

Ключові слова: PGP, ШИФРУВАННЯ, ОНЛАЙН-ПОВІДОМЛЕННЯ, КЛЮЧІ, ACTIVE DIRECTORY.

ЗМІСТ

ВСТУП

Актуальність. Онлайн-повідомлення стали невід'ємною частиною нашого життя. Ми використовуємо їх для спілкування з друзями та родиною, ведення бізнесу та участі в громадському житті. Однак, онлайн-повідомлення також вразливі до атак з боку кіберзлочинців. Модель захисту онлайн-повідомлень - це основа для захисту онлайн-повідомлень від кібератак. Вона може включати різні функції, такі як шифрування, аутентифікація та авторизація. Актуальність моделі захисту онлайн-повідомлень полягає в тому, що вона може допомогти захистити конфіденційність, безпеку і демократію. Онлайн-повідомлення можуть містити конфіденційну інформацію, таку як фінансові дані або особисті дані. Модель захисту онлайн-повідомлень може допомогти запобігти потраплянню цієї інформації в чужі руки.

Онлайн-повідомлення можуть бути використані для поширення шкідливого програмного забезпечення або фішингових атак. Модель захисту онлайн-повідомлень може допомогти захистити користувачів від цих атак. Онлайн-повідомлення можуть бути використані для поширення дезінформації або підриву демократичних процесів. Модель захисту онлайн-повідомлень може допомогти захистити демократію від цих загроз.

Бізнес покладається на онлайн-повідомлення для спілкування з клієнтами та партнерами. Модель захисту онлайн-повідомлень може допомогти бізнесу захистити свої конфіденційні дані та зберегти довіру клієнтів.

Уряди використовують онлайн-повідомлення для спілкування з громадянами та надання послуг. Модель захисту онлайн-повідомлень може допомогти урядам захистити конфіденційність громадян і забезпечити безпеку критично важливої інфраструктури.

Приватні особи використовують онлайн-повідомлення для спілкування з друзями та родиною, обміну інформацією та самовираження. Модель захисту онлайн-повідомлень може допомогти людям захистити їхнє приватне життя та реалізувати своє право на свободу вираження поглядів. Компанії покладаються

на онлайн-повідомлення для спілкування зі співробітниками та партнерами щодо конфіденційної інформації, наприклад, комерційної таємниці та планів розвитку продукту.

Модель захисту онлайн-повідомлень може допомогти бізнесу захистити свою інтелектуальну власність від несанкціонованого доступу та розголошення. Захист дітей: діти особливо вразливі до кібератак, оскільки вони можуть бути менш обізнані про ризики і менш здатні захистити себе. Модель захисту онлайн-повідомлень може допомогти захистити дітей від кібербулінгу, онлайн-хижаків та інших загроз. Сприяння довірі в онлайн-спілкуванні, коли люди знають, що їхні онлайн-повідомлення є безпечними та захищеними, вони з більшою ймовірністю будуть використовувати платформи обміну повідомленнями для спілкування. Це може призвести до створення більш пов'язаного та поінформованого суспільства.

Загалом, модель захисту онлайн-повідомлень є дуже актуальною темою для окремих осіб, організацій та суспільства в цілому.

Метою роботи є реалізація ефективної моделі захисту онлайн-повідомлень.

Для досягнення поставленої мети вирішуються такі **задачі:**

- аналіз існуючих методів та засобів захисту онлайн-повідомлень.
- розробка моделі захисту онлайн-повідомлень.
- реалізація програмного забезпечення на основі розробленої моделі.

Галузь застосування. Розроблена модель та програмне забезпечення відносяться до галузі кібербезпеки та інформаційної безпеки в онлайн-комунікаціях.

Об’єктом дослідження є процеси захисту онлайн-повідомлень.

Предметом дослідження є модель захисту онлайн-повідомлень.

Методи дослідження які базуються на нечіткій логіці для аналізу та оцінки ризиків у онлайн-комунікаціях, і на об'єктноорієнтованому програмуванні для розробки програмної реалізації розробленої моделі захисту повідомлень.

Новизна одержаних результатів полягає в наступному:

Розробка та впровадження моделі захисту онлайн-повідомлень, яка враховує сучасні виклики та загрози у сфері кібербезпеки. Ця модель базується на методі криптографічного захисту, для забезпечення конфіденційності, цілісності та доступності повідомлень. Крім того, новизна полягає в автоматичному пошуку відкритих ключів одержувачів на сервері PGP і закритих ключів поточного користувача в Active Directory.

Практична цінність отриманих результатів:

Розроблена модель захисту онлайн-повідомлень, базується на PGP, забезпечує ефективну конфіденційність для користувачів онлайн-комунікацій та організацій, що використовують ці технології. Вони можуть впроваджувати розроблену модель захисту для забезпечення високого рівня безпеки під час обміну повідомленнями, що дозволить запобігти несанкціонованому доступу, витоку інформації та атакам з боку зловмисників.

Застосування цієї технології гарантує безпеку важливої інформації, забезпечуючи надійний захист для користувачів і підприємств онлайн-повідомлень, такої як особисті дані, комерційні повідомлення, лікарські або фінансові дані, що є важливими для багатьох організацій та користувачів у цифровому світі.

Апробація. Основні положення роботи доповідалися та обговорювалися на конференції:

− Міжнародна науково-практична конференція «ЖИВУЧІСТЬ ТА РЕЗИЛЬЄНТНІСТЬ – 2023» (Київ: Інститут проблем моделювання в енергетиці ім. Г.Є. Пухова)

Розділ 1. АНАЛІЗ СУЧАСНИХ ДОСЛІДЖЕНЬ У СФЕРІ ЗАХИСТУ ОНЛАЙН-ПОВІДОМЛЕНЬ

1.1 Проблематика захисту онлайн-повідомлень

Цифрова трансформація, що триває, змінює спосіб роботи мільйонів людей та інструменти, які вони використовують для ведення бізнесу. Окрім електронної пошти, віддалені працівники все частіше використовують мобільні додатки для обміну миттєвими повідомленнями, такі як WhatsApp, Facebook Messenger і Telegram, а також додатки для спільної роботи, такі як Zoom, Microsoft Teams і Slack, у своїх повсякденних справах.

Хмарні комунікаційні інструменти допомагають вашим співробітникам спілкуватися з клієнтами, колегами, постачальниками та фінансовими установами. Але їхня популярність відкрила новий світ можливостей для кіберзлочинців. Зі збільшенням кількості атак, компрометація ділової електронної пошти (BEC) є одним з елементів більшої загрози - компрометації бізнес-комунікацій (BCC).

Фішинг електронної пошти, оригінальна форма кібератаки, вперше з'явилася в 1990-х роках. Ці атаки здійснювалися шахраями, які видавали себе за співробітників AOL, надсилаючи електронні листи передплатникам AOL з проханням "верифікувати" свій обліковий запис або "підтвердити" платіжну інформацію. Такий підхід до BEC все ще широко використовується і сьогодні.

Хоча компанії стали більш обізнаними про фішингові атаки через електронну пошту і впровадили рішення для виявлення таких атак, кіберзлочинці постійно розробляють нові, більш витончені методи, які використовують поширення хмарних комунікаційних додатків для проведення своїх атак. Зловмисники, особливо ті, що намагаються вчинити шахрайство, видачу себе за іншу особу та BCC, уникають виявлення, розпочинаючи

спілкування електронною поштою, а потім швидко переводячи розмови в інші канали, такі як Slack або Teams.

Маніпулюючи людьми, щоб змусити їх порушувати протоколи безпеки, зловмисники можуть отримати доступ до службової інформації, інтелектуальної власності та фінансових ресурсів організації.

Проникнення у вашу мережу є метою більшості зловмисників, а отримання доступу відкриває перед ними цілий ряд можливостей завдати шкоди вашому бізнесу. Хмарні засоби комунікації та спільної роботи є цікавими та інтерактивними, але вони, як правило, використовуються на нерегульованих персональних пристроях, таких як мобільні телефони та планшети, і не захищені з таким же ступенем безпеки, як корпоративні пристрої. Це робить персональні пристрої більш вразливими для зловмисників і ускладнює запобігання BCC-атакам.

Наприклад, якщо зловмисники успішно отримують доступ до системи керівника за допомогою фішингу або видачі себе за іншу особу, вони можуть змусити персонал обробляти несанкціоновані платежі на їхні власні рахунки. Інсайдери, обізнані з вашим бізнесом і мережею, також можуть завдати величезної шкоди. Насправді, більшість порушень щороку є справою рук інсайдерів, і фінансові втрати, які вони спричиняють, продовжують зростати.

Витончені злочинці навіть створюють фальшиві відео- та аудіо повідомлення, в яких видають себе за керівників організації. У цих повідомленнях, які надсилаються звичайними каналами зв'язку, зловмисники можуть змусити працівників відділу кредиторської заборгованості здійснити несанкціоновані платежі безпосередньо на їхній рахунок або створити фальшиві рахунки-фактури.

Зупинити шахрайство, що здійснюється через нерегульовані хмарні канали, надзвичайно складно, а виявлення та запобігання атакам до того, як вони можуть завдати шкоди, є критично важливим для успішної роботи бізнесу.

Проблематика захисту онлайн-повідомлень включає в себе різноманітні аспекти, пов'язані із забезпеченням конфіденційності, цілісності та доступності інформації в інтернеті.

Конфіденційність і особиста інформація.

Витоки даних. Це стосується ситуацій, коли зловмисники отримують несанкціонований доступ до баз даних чи інших зберігаючих особисту інформацію систем. Витік даних може виникнути через атаки на веб-сайти, служби електронної пошти або інші онлайн-платформи.

Шахрайства та обман. Здійснюється шляхом використання соціальних інженерних технік, фішингу, або інших методів обману для отримання конфіденційної інформації, такої як паролі, номери кредитних карт і т.д.

Безпека електронної пошти та миттєвих повідомлень.

Фішінг. Це атака, під час якої зловмисники вдаються в те, що вони є надійними джерелами (наприклад, банком чи інтернет-провайдером) з метою отримання конфіденційних даних.

Спам. Надмірні непопереджені розсилки повідомлень, які можуть містити рекламу, фішингові посилання, або шкідливий вміст.

Боротьба із кіберзлочинністю.

Шкідливе програмне забезпечення. Шкідливе програмне забезпечення може включати в себе віруси, троянці, різновиди шпигунського ПЗ, які можуть пошкодити або використовувати інформацію без згоди користувача.

Деніал-сервіс атаки (DDoS). Це атака, коли зловмисники переповнюють сервери або мережі трафіком, забираючи доступ до них для законних користувачів.

Інтернет-приватність.

Слідкування за користувачем. Деякі компанії та сервіси збирають велику кількість інформації про користувачів, таку як місцезнаходження, інтереси, покупки, що може порушувати приватність.

Контроль за конфіденційністю даних. Важливо забезпечити користувачам можливість контролювати, як їхня особиста інформація використовується та передається.

Захист від кібершахрайства та соціальних загроз.

Кібербулінг. Це включає в себе цифрове домагання, погрози та інші агресивні вчинки в мережі.

Загрози від невідомих осіб. Небезпека від спроб обману, атак або інших загроз від невідомих користувачів чи шахраїв.

Захист від технічних вразливостей.

Оновлення безпеки: Важливо регулярно оновлювати програмне забезпечення та операційні системи, щоб закрити вразливості, які можуть бути використані зловмисниками.

Ефективний захист від цих проблем включає в себе використання антивірусного програмного забезпечення, впровадження механізмів двофакторної аутентифікації, навчання користувачів основам кібербезпеки та дотримання найновіших стандартів безпеки в інтернеті.

1.2 Аналіз підходів та методів захисту онлайн-повідомлень

Захист онлайн-повідомлень є критично важливим питанням у цифрову епоху, оскільки конфіденційна інформація може легко потрапити до чужих рук, якщо її не захистити належним чином. Існує кілька підходів і технологій, що використовуються для вирішення цієї проблеми, кожен з яких має свої сильні і слабкі сторони.

- Наскрізне шифрування (E2E): Цей метод гарантує, що розшифрувати і прочитати повідомлення можуть тільки відправник і передбачуваний одержувач. Популярні програми для обміну повідомленнями E2E включають Signal, WhatsApp (для особистих чатів) і Telegram (у режимі секретного чату).

- Безпека на транспортному рівні (TLS): TLS використовується для шифрування даних, що передаються між пристроєм користувача та сервером (наприклад, HTTPS для веб-спілкування). Він запобігає підслуховуванню та несанкціонованому доступу під час передачі даних.

- Використання надійних, унікальних паролів для онлайн-акаунтів та двофакторної автентифікації (2FA) може захистити повідомлення від несанкціонованого доступу.

- Використання програм для обміну повідомленнями з вбудованими функціями безпеки, такими як самознищення повідомлень, безпечний спільний доступ до файлів і приватні групи, може посилити захист повідомлень.

- VPN створюють безпечний, зашифрований тунель для інтернет-трафіку, що ускладнює перехоплення та доступ до онлайн-повідомлень третіми особами.

- PGP - це програма для шифрування та дешифрування даних, яка забезпечує криптографічну конфіденційність та автентифікацію. Вона часто використовується для захисту електронної пошти.

- Деякі провайдери електронної пошти пропонують розширені функції безпеки, включаючи зашифровану електронну пошту, для захисту повідомлень під час передачі та зберігання.

- Цифрові підписи використовують асиметричну криптографію для перевірки автентичності та цілісності повідомлення. Це гарантує, що повідомлення не було підроблено під час передачі.

- Такі протоколи, як Signal Protocol (використовується в Signal, WhatsApp та інших), забезпечують наскрізне шифрування і секретність пересилання, що ускладнює доступ до вмісту повідомлень навіть для постачальника послуг.

- Блокчейн і DLT пропонують децентралізоване і стійке до несанкціонованого доступу зберігання повідомлень, що може підвищити безпеку онлайн-повідомлень.

- Брандмауери та системи виявлення вторгнень. Ці заходи мережевої безпеки можуть допомогти захистити повідомлення від зовнішніх загроз шляхом моніторингу та контролю мережевого трафіку.

- Постійне оновлення програмного забезпечення, операційних систем і програм обміну повідомленнями має вирішальне значення для запобігання вразливостям, якими можуть скористатися зловмисники.

- Інформування користувачів про найкращі практики безпеки в Інтернеті, фішингові загрози та тактики соціальної інженерії має важливе значення для захисту повідомлень від людських помилок.

- Ці системи дозволяють перевіряти дані, не розкриваючи основну інформацію, пропонуючи переваги конфіденційності та безпеки.

- MPC дозволяє декільком учасникам спільно обчислювати функцію над своїми вхідними даними, зберігаючи ці дані приватними. Це може бути використано для захисту конфіденційних даних у сценаріях спільної роботи.

- Гомоморфне шифрування дозволяє виконувати обчислення над зашифрованими даними без їх розшифрування, зберігаючи конфіденційність і дозволяючи обробляти дані.

Вибір відповідного підходу або комбінації підходів залежить від конкретного випадку використання, моделі загрози та рівня безпеки, необхідного для онлайн-повідомлень. На практиці багаторівневий підхід, який поєднує кілька з цих методів, часто є найефективнішим способом захисту онлайн-повідомлень від різних загроз.

Протоколи електронної пошти, що використовуються сьогодні, були розроблені дуже давно. Оскільки автори протоколів не дбали про конфіденційність електронної пошти, користувачі електронної пошти змушені використовувати додаткове програмне забезпечення для захисту свого електронного листування. Хоча стандарти для наскрізного шифрування електронної пошти існують вже більше двох десятиліть, їх впровадження не набуло широкого поширення, і більшість електронних листів все ще надсилаються у вигляді відкритого тексту. Існуючі стандарти шифрування електронної пошти базуються на криптографії з відкритим ключем. Отже, користувачі повинні обмінюватися своїми відкритими ключами і перевіряти їх автентичність, перш ніж вони зможуть почати безпечне спілкування. Це

призводить до низької зручності використання більшості існуючих програм для шифрування електронної пошти.

Наразі для наскрізного шифрування та підписання електронної пошти використовуються два основні стандарти: S/MIME [9] та OpenPGP [10]. Схеми шифрування дуже схожі, і обидва вони використовують комбінацію криптографії з симетричним ключем, криптографії з відкритим ключем і хешування. Повідомлення шифруються за допомогою симетричного алгоритму з ключем, який генерується для кожного повідомлення. Цей симетричний ключ потім шифрується відкритим ключем одержувача і додається до зашифрованого повідомлення. Щоб також забезпечити цілісність електронного листа, відправник використовує свій приватний ключ для підписання хешу повідомлення і додає його до листа перед шифруванням. Таким чином, щоб мати можливість здійснювати зашифроване спілкування, користувачі повинні спочатку обмінятися своїми відкритими
ключами.

Основна відмінність між цими двома стандартами полягає в тому, як розподіляються і перевіряються відкриті ключі. У той час як S/MIME використовує централізовану модель довіри, засновану на центрах сертифікації, стандарт OpenPGP був розроблений для використання децентралізованої моделі довіри, відомої як Web Of Trust.

OpenPGP: Стандарт OpenPGP [10] походить від програми шифрування PGP (Pretty Good Privacy), випущеної Філіпом Циммерманом у 1991 році [11]. Стандарт реалізовано у різних проектах включаючи GnuPG (GNU Privacy Guard, також відомий як GPG) [12], найпоширенішу реалізацію з відкритим вихідним кодом (Рис. 1.1).

Позначення PGP часто використовується у значенні стандарту OpenPGP, і тому я також використовую його в решті частини цієї роботи.

PGP був розроблений для використання децентралізованої моделі довіри для розподілу відкритих ключів, відомої як Web Of Trust, в якій кожен учасник може засвідчити відкритий ключ іншого учасника, додавши до ключа свій

підпис. Користувач вирішує, чи є ключ дійсним чи ні, на основі підписів на ключі.

Дійсність ключа також можна перевірити за відбитком пальця. Для розповсюдження ключів у PGP існує поняття серверів ключів. Користувач може завантажити свій відкритий ключ на сервер ключів, а інші користувачі можуть завантажити його, щоб мати змогу надсилати йому зашифровані повідомлення.

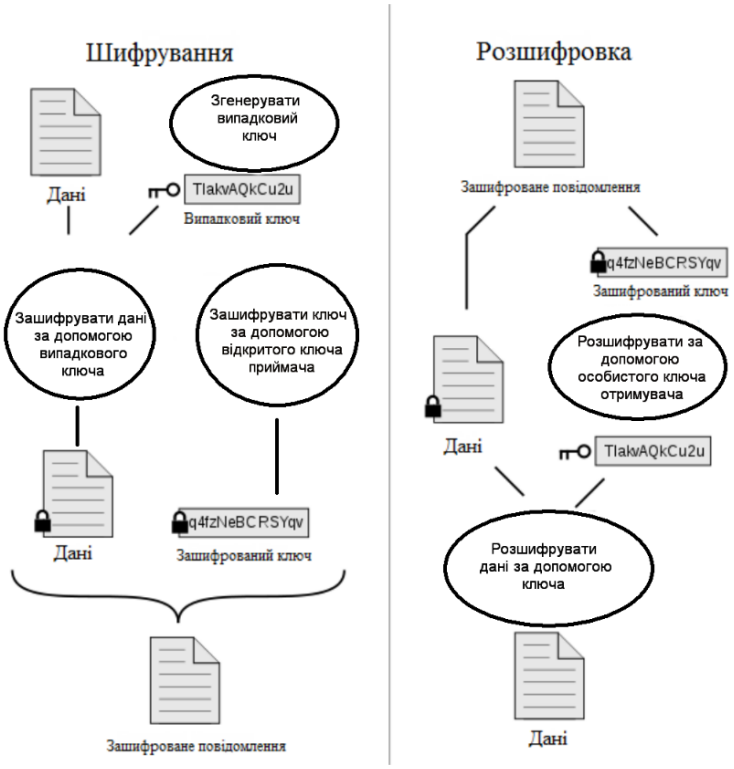

Рис. 1.1 Схема PGP шифрування

S/MIME: Стандарт S/MIME (Secure/Multipurpose Internet Mail Extensions) (Рис. 1.2) [9] по суті надає користувачам ті ж самі функції, що і PGP, тільки з іншими форматами шифрування. Єдина суттєва відмінність полягає в управлінні ключами. S/MIME використовує формат сертифікатів X.509 формат сертифікатів і використовує централізовану модель довіри, засновану на Центрах Сертифікації (ЦС), де користувач повинен спочатку отримати сертифікат від ЦС, який перевіряє і засвідчує його особу, а потім обмінюватися сертифікатами з іншими користувачами.

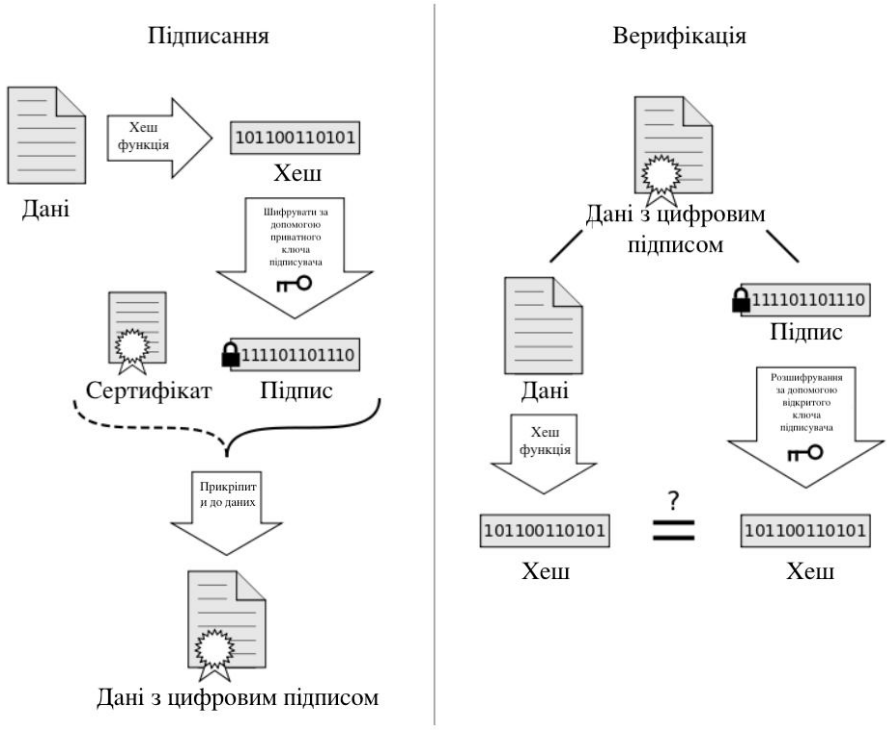

Рис. 1.2 Схема стандарту S/MIME

Обидві описані вище моделі довіри мають багато проблем. Павутина довіри погано масштабується, і недосвідчені користувачі не знають, якому відкритому ключу можна довіряти, а якому ні. Це вносить велику складність в систему і, таким чином, робить її незахищеною. У централізованій моделі перевірка відкритих ключів відбувається автоматично. Однак, це вимагає довіри до третьої сторони, і було багато випадків компрометації центрів сертифікації. Крім того, отримання сертифікатів є обтяжливим процесом, який приносить додаткові витрати.

Хоча обидві ці технології забезпечують автентифікацію, конфіденційність, неспростовність та цілісність електронних листів, вони все ще не забезпечують достатнього рівня конфіденційності для користувачів електронної пошти. Всі метадані, включаючи тему, одержувача і відправника, не шифруються, тому підслуховувач все одно зможе відстежити, хто з ким спілкується, коли, і навіть здогадатися, про що вони говорять, з незашифрованого заголовка підтеми. заголовку суб'єкта. Вирішити цю проблему можна було б, лише запровадивши абсолютно нову систему електронної пошти з вбудованим захистом.

Однак група дослідників, включаючи автора PGP, нещодавно запропонувала нові протоколи та архітектуру електронної пошти з наскрізним захистом, який мінімізує вплив метаданих на станції на шляху між відправником і одержувачем. Проект під назвою Dark Internet Mail Environment (DIME) [13] має на меті зробити електронну пошту захищеною за замовчуванням, а також зробити її більш безпечною, ніж існуючі підходи. Він також фокусується на зручності використання, що вимагає автоматизації управління ключами, включаючи генерацію, розповсюдження та перевірку ключів. Повідомлення шифруються на декількох рівнях, щоб конкретна станція обробки мала доступ лише до тієї інформації, яку їй потрібно бачити. Що стосується моделі довіри, то в цій системі постачальники послуг все ще займають довірчу позицію. Однак користувач може вибрати один з трьох рівнів довіри (Довірливий, Обережний і Параноїдальний), які визначають доступ сервера до закритих ключів користувача.

У *довірчому* режимі сервер вирішує всі питання конфіденційності від імені користувача. Ключі генеруються на сервері, тому провайдер має прямий доступ до приватних ключів, захищених лише парольною фразою. Передбачається, що акаунти, які працюють в цьому режимі, будуть використовувати традиційний доступ через протокол SMTP і отримувати повідомлення за допомогою протоколів POP або IMAP через SSL. Цей режим вимагає такого ж рівня довіри до постачальника послуг, як і в старій архітектурі електронної пошти або веб-служб електронної пошти, які шифрують повідомлення на сервері. Він забезпечує захист лише для повідомлень, що передаються через Інтернет.

Обережний режим є компромісом між конфіденційністю та зручністю використання системи. Він мінімізує довіру до поштового сервера, але користувачеві не потрібно нічого робити, щоб керувати ключами. Генерація ключів і шифрування відбувається на стороні клієнта. Сервер зберігає лише зашифровані копії приватних ключів користувача та зашифровані повідомлення,

що дозволяє користувачеві отримувати доступ до своєї поштової скриньки з різних пристроїв без передачі ключів вручну.

У *параноїдальному* режимі сервер ніколи не має доступу до закритих ключів користувача (навіть у зашифрованому вигляді). Цей режим вимагає найменшої довіри до сервера, але значно обмежує зручність використання системи, оскільки користувач не може отримати доступ до свого облікового запису з різних пристроїв без передачі закритих ключів вручну що також є причиною неможливості використання цього режиму для веб-пошти.

Автори також обговорюють, як можна було б додати ідеальну пряму секретність (Вважається, що протокол має ідеальну пряму секретність, якщо компрометація довгострокових ключів не призводить до компрометації ключів минулих сеансів. [14]) до асинхронної системи обміну повідомленнями, якою є електронна пошта, для параноїдального режиму. Протоколи узгодження ключів, що забезпечують ідеальну пряму секретність, вимагають обміну інформацією між обома сторонами для генерації сеансового ключа, що неможливо забезпечити в електронній пошті. Автори стверджують, що, покладаючись на схему автоматичного виявлення і перевірки ключів з використанням параноїдального режиму, користувачі можуть досягти ідеальної прямої секретності, регулярно оновлюючи свої ключі. Оскільки приватні ключі ніколи не зберігатимуться на сервері, користувач може бути впевнений, що ключі ніколи не зможуть бути відновлені. Хоча це і підвищує рівень безпеки, але не здається дуже ефективним обхідним шляхом.

Набір протоколів DIME має на меті, головним чином, надати інструменти, які люди можуть зрозуміти і використовувати, і які не потребують встановлення додаткового програмного забезпечення для шифрування. Він все ще вимагає довіри до постачальника послуг електронної пошти, принаймні для розподілу відкритих ключів, але, прозорість управління ключами, як і в PGP, не приносить кращої безпеки системі, оскільки звичайні користувачі не знають концепції криптографії з відкритим ключем. Це лише додає складності і, як наслідок, погіршує зручність використання.

1.3 Критичний огляд сучасних досліджень у сфері захисту онлайн-повідомлень

Розвиток криптографії з відкритим ключем є найбільшою революцією за всю історію криптографії. Криптографія з відкритим ключем забезпечує радикальний відхід від усього, що було до цього. Алгоритми з відкритим ключем базуються більше на математичних функціях, ніж на підстановці та перестановці. Крім того, криптографія з відкритим ключем є асиметричною, тобто передбачає використання двох окремих ключів, на відміну від симетричного шифрування, яке використовує лише один ключ [1]. Як ми побачимо, використання двох ключів має серйозні наслідки у сферах конфіденційності, цілісності та автентифікації. Одна з помилкових думок про асиметричне шифрування полягає в тому, що воно є більш захищеним від атак криптоаналізу, ніж симетричне шифрування.

Більше того, безпека схеми шифрування залежить від довжини ключа і обчислювальної роботи, пов'язаної з розкриттям зашифрованого тексту. Ні про симетричне шифрування, ні про шифрування з відкритим ключем не сказано нічого, що робить одне з них кращим за інше з точки зору з точки зору стійкості до криптоаналізу. Друга помилкова думка про шифрування з відкритим ключем полягає в тому, що воно зробило симетричне застаріло симетричне шифрування. Через обчислювальні накладні витрати сучасних методів шифрування з відкритим ключем, здається, немає ніякої що від симетричного шифрування відмовляться в найближчому майбутньому.

Застосування криптосистем з відкритим ключем. Системи з відкритим ключем використовують криптографічний алгоритм з двома ключами, один з яких називається приватним, а інший - публічним [1]. Відповідно до програми, відправник використовує або закритий ключ відправника, або відкритий ключ одержувача, або обидва ключі для виконання певних криптографічних функцій.

У широкому сенсі, ми можемо класифікувати використання криптосистем з відкритим ключем на три категорії:

- Шифрування (1.1) /розшифрування (1.2): Відправник шифрує повідомлення за допомогою відкритого ключа одержувача.

$$C = M^e \bmod n \tag{1.1}$$

$$M = C^d \bmod n \tag{1.2}$$

де C - зашифрований текст.

M - відкрите повідомлення.

e - відкритий експонента одержувача.

d - приватний експонента одержувача.

n - модуль.

- Цифровий підпис (1.3): При цифровому підписі відправник "підписує" повідомлення своїм власним закритим ключем. Цей процес підпису досягається за допомогою криптографічного алгоритму, який застосовується до невеликого блоку даних, що є функцією повідомлення.

$$S = H(M)^d \bmod n \tag{1.3}$$

де S - цифровий підпис.

H(M) - хеш повідомлення.

d - приватне число відправника.

e - публічний експонента відправника.

n - модуль.

- Обмін ключами: При обміні ключами обидві сторони співпрацюють для обміну ключами. Пропонується багато різних підходів, які включають приватний ключ (ключі) однієї або обох сторін.

Сертифікат X.509. X.509 визначає основу для надання послуг автентифікації користувачам. Кожен сертифікат містить відкритий ключ користувача і підписується закритим ключем довіреного центру сертифікації (ЦС) [1]. Крім того, X.509 визначає альтернативні протоколи автентифікації,

засновані на використанні сертифікатів з відкритим ключем. X.509 є важливим стандартом завдяки своїй структурі сертифікатів та різноманітним протоколам автентифікації, визначеним у ньому. X.509 є важливим стандартом, оскільки його структура сертифікатів і різні протоколи автентифікації, визначені в X.509, використовуються в різних контекстах. X.509 використовує концепцію криптографії з відкритим ключем і цифрових підписів.

Сертифікат X.509 видається Центром сертифікації (ЦС) і належним чином підписується закритим ключем ЦС.

Він включає в себе:

- Відкритий ключ власника
- Ім'я власника
- Дата закінчення терміну дії відкритого
- Назва видавця (ЦС, який видав цифровий сертифікат)
- Серійний номер цифрового сертифіката
- Цифровий підпис видавця

Найбільш поширений формат цифрових сертифікатів визначений міжнародним стандартом CCITT X.509 [2]. А найпоширенішим стандартом для цифрових сертифікатів є сертифікат X.509. Цифрові сертифікати є основою для ідентифікації інформації та пов'язують їх ідентичність з відкритими ключами. Електронний підпис, належним чином виданий центром сертифікації, який засвідчує повноваження особи, що підписує електронну форму. Кожен сертифікат містить відкритий ключ користувача і підписується закритим ключем центру сертифікації.

Вимоги до безпеки документа:

- Автентичність
- Конфіденційність
- Цілісність
- Невизнання

Генерація сертифіката відкритого ключа. Серцем схеми X.509 є сертифікат відкритого ключа (Рис. 1.3), пов'язаний з кожним користувачем. Ці

сертифікати генеруються довіреним центром сертифікації (ЦС) і записуються в каталог ЦС або користувачем, якому видано сертифікат.

Сервер каталогів не відповідає за створення відкритих ключів або за функцію сертифікації, а лише надає користувачам легкодоступне місце для отримання сертифікатів.

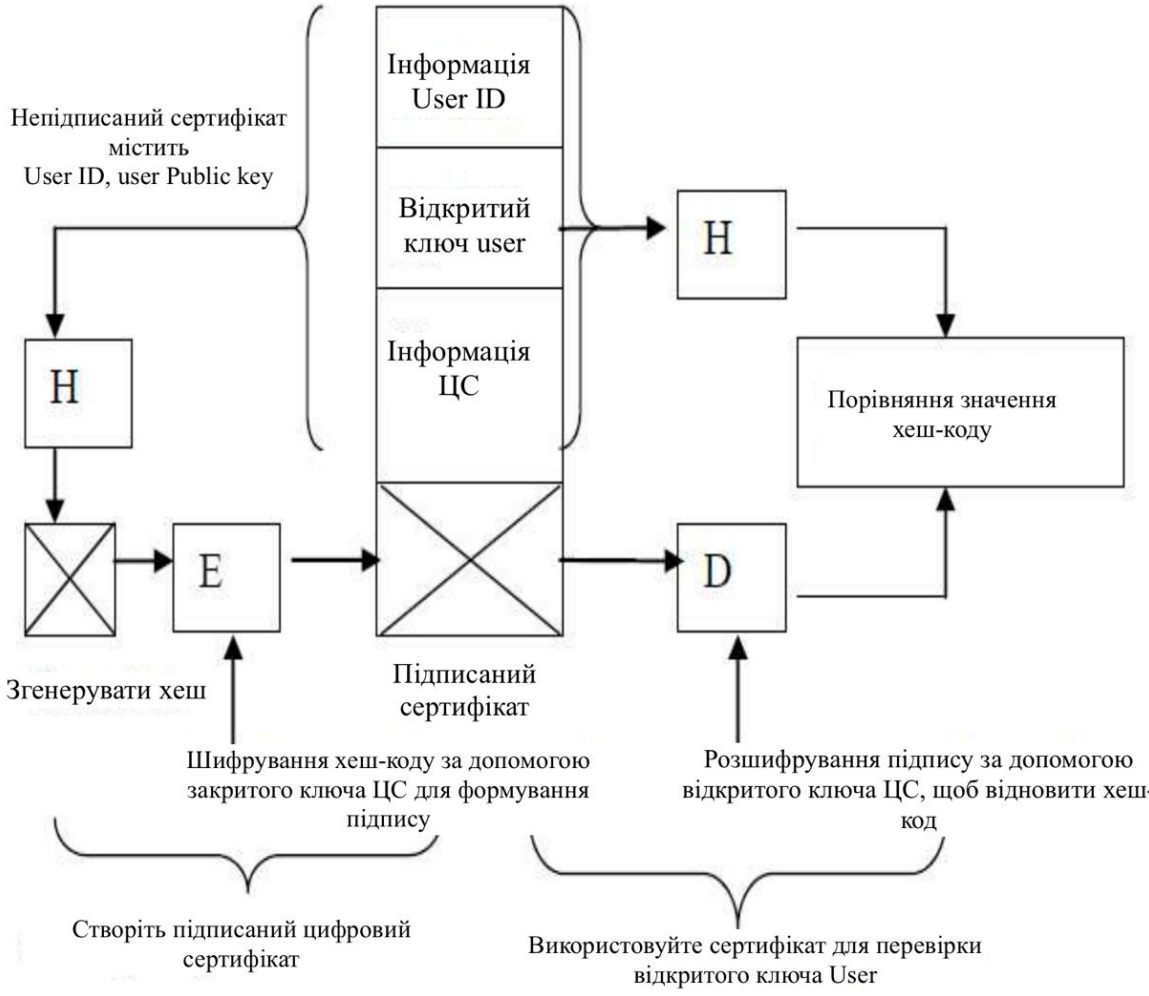

Рис. 1.3 Використання сертифіката відкритого ключа

На рисунку 1.3 показано звичайне використання сертифікату відкритого ключа як відправником, так і отримувачем, а також використання ключів для шифрування та розшифрування.

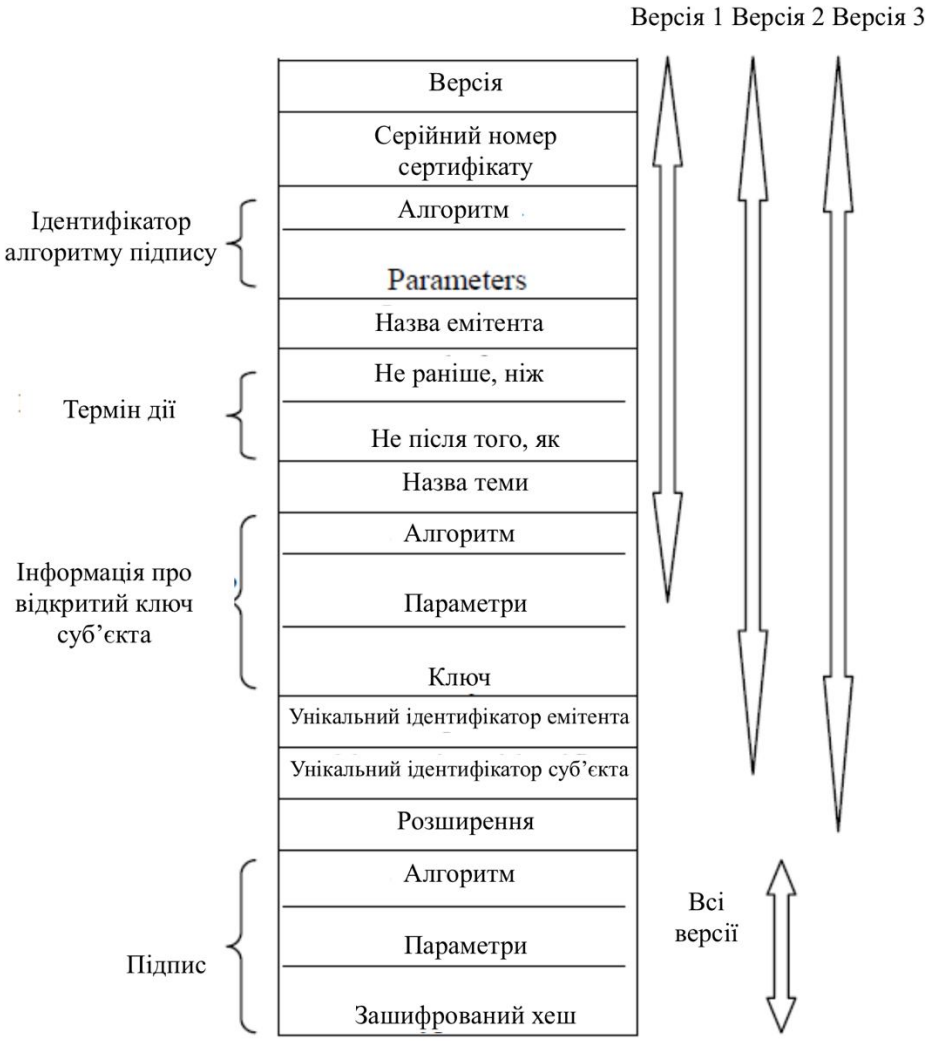

Рис. 1.4 Формат сертифіката X.509

На рисунку 1.4 зображено загальний формат сертифіката X 509, який включає наступні елементи.

- Версія: Розрізняє послідовні версії формату сертифіката; за замовчуванням версія дорівнює 1. Значення version має бути 2, якщо присутній унікальний ідентифікатор емітента або унікальний ідентифікатор суб'єкта, а якщо присутнє одне або декілька розширень, то значення версії має бути 3.
- Серійний номер: Серійний номер - це цілочисельне значення, яке є унікальним для ЦС, що видає сертифікат, і яке пов'язане з цим сертифікатом.

- Ідентифікатор алгоритму підпису: Містити алгоритм, використаний для підписання сертифіката, разом з будь-якими пов'язаними з ним параметрами.

Оскільки ця інформація повторюється в полі підпису в кінці сертифіката, тому це поле практично не має ніякої користі.

- Ім'я емітента: Ім'я емітента - це ім'я центру сертифікації, який створив і підписав цей сертифікат.
- Термін дії: Складається з двох дат: першої та останньої, протягом яких сертифікат дійсний.
- Ім'я суб'єкта: Це ім'я користувача, якому видається цей сертифікат. Це означає, що цей сертифікат засвідчує відкритий ключ суб'єкта, який володіє відповідним закритим ключем.
- Інформація про відкритий ключ суб'єкта: Містить відкритий ключ суб'єкта та ідентифікатор алгоритму, для якого цей ключ має бути використаний, а також будь-які пов'язані з ним параметри.
- Унікальний ідентифікатор емітента: Це необов'язкове бітове рядкове поле, що складається з ідентифікатора, який використовується для унікальної ідентифікації ЦС, що видає сертифікат.
- Унікальний ідентифікатор суб'єкта: Це необов'язкове бітове строкове поле, що складається з ідентифікатора, який використовується для однозначної ідентифікації суб'єкта [1].
- Розширення: Складається з набору одного або декількох полів розширення. Додаткові розширення додано у версії 3.
- Підпис: Підпис охоплює всі інші поля сертифіката; він містить хеш-код інших полів, який зашифрований за допомогою закритого ключа центру сертифікації. Це поле також включає ідентифікатор алгоритму підпису.

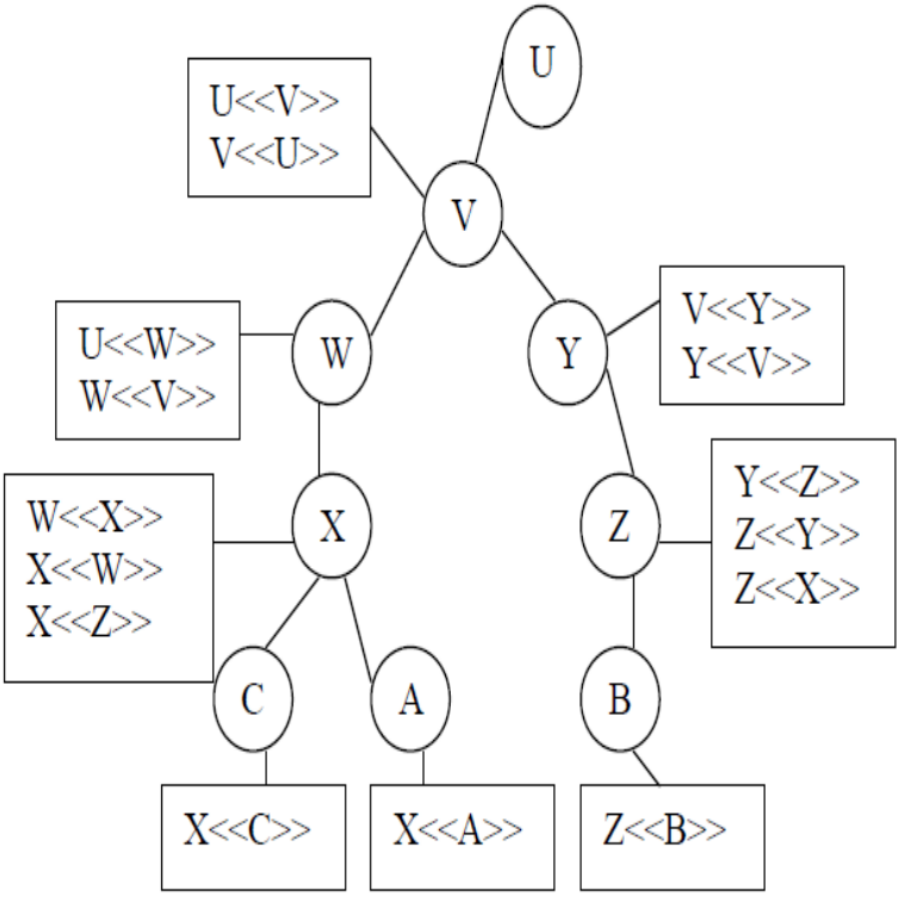

Рис. 1.5 Ієрархія X.509

Рисунок 1.5, взятий з X.509, є прикладом ієрархії. З'єднане коло показує ієрархічний зв'язок між центрами сертифікації, а клітинки показують сертифікати, що зберігаються в каталозі для кожного запису про центр сертифікації. Каталог для кожного ЦС включає два типи сертифікатів:

- Прямі сертифікати: Це сертифікати X, згенеровані іншими центрами сертифікації
- Зворотні сертифікати: Це сертифікати, згенеровані X, які є сертифікатами інших центрів сертифікації

У цьому прикладі користувач A може отримати наступні сертифікати з каталогу, щоб встановити шлях сертифікації до B:

Коли A отримає ці сертифікати, він може послідовно розгорнути шлях сертифікації, щоб відновити довірену копію відкритого ключа B відкритого ключа B. Використовуючи цей відкритий ключ, A може надсилати зашифровані повідомлення B. Тепер, якщо A хоче отримувати зашифровані повідомлення

назад від B, або знову надсилати підписані повідомлення B, то B потрібен відкритий ключ A. B може отримати цей набір сертифікатів з каталогу або A може надати їх як частину свого першого повідомлення B.

Відкликання сертифіката. Кожен сертифікат має термін дії, як, наприклад, картка банкомату або кредитна картка. Крім того, новий сертифікат видається безпосередньо перед закінченням терміну дії старого [2]. Крім того, іноді може виникнути потреба відкликати сертифікат до закінчення терміну його дії з однієї з наступних причин:

- Якщо існує ймовірність компрометації особистого ключа користувача.
- Коли користувач більше не сертифікований цим центром сертифікації.
- Коли вважається, що сертифікат ЦС скомпрометований.

Кожен ЦС повинен мати список, який містить всі відкликаних, але не прострочених сертифікатів, виданих цим ЦС, що включає сертифікати, видані як користувачам, так і іншим ЦС. Цей список буде розміщено в каталозі. Кожен список відкликання сертифікатів (CRL) (Рис. 1.6), розміщений у каталозі, підписується емітентом і містить назви емітента, дати створення списку, дати, на яку заплановано випуск наступного CRL, та запису для кожного відкликаного сертифіката. Кожен запис у каталозі складається з серійного номера сертифіката та дати відкликання для цього сертифіката. Оскільки серійні номери є унікальними для кожного центру сертифікації, цього номера достатньо для ідентифікації сертифіката.

Коли користувач отримує сертифікат разом з повідомленням, він повинен визначити, чи був цей сертифікат відкликаний, чи це оригінальний сертифікат. Користувач також може перевіряти каталог щоразу, коли отримує сертифікат. Щоб уникнути затримок, пов'язаних з пошуком у каталозі, користувач повинен підтримувати локальний кеш сертифікатів і список відкликаних сертифікатів.

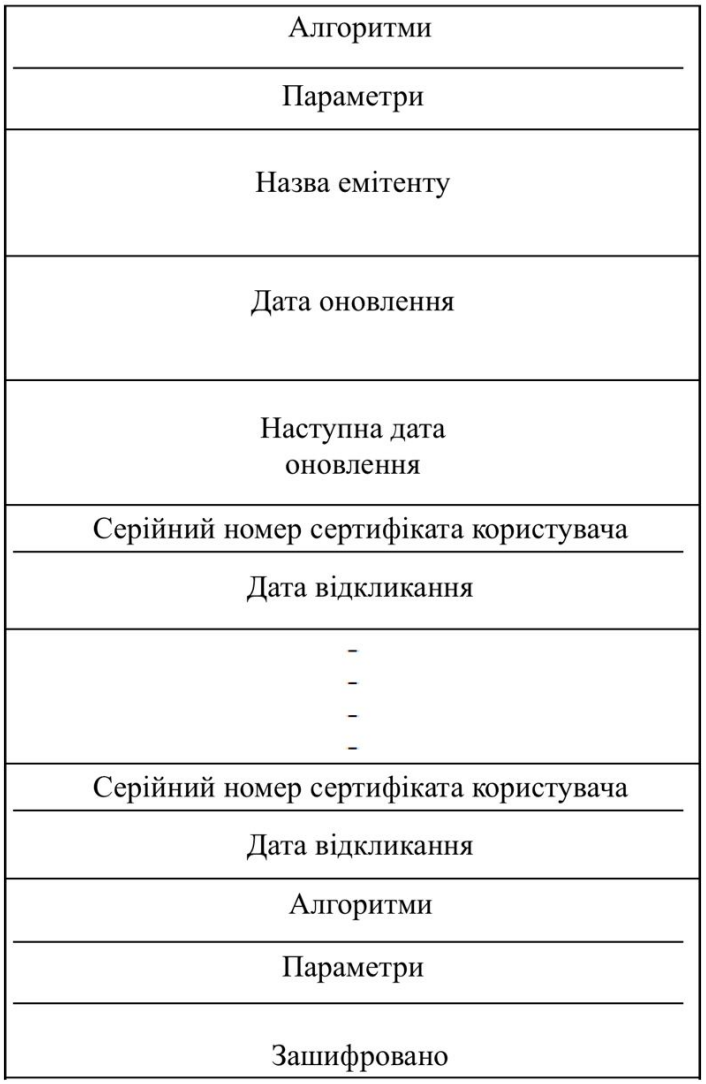

Рис. 1.6 Список відкликання сертифікатів

PGP. PGP - це криптографічний пакет з відкритим ключем, призначений для публічного використання. Він забезпечує конфіденційність, автентичність, цілісність і неможливість відмовитися від відправника. Хоча PGP може шифрувати будь-які дані або файли, він найчастіше використовується для електронної пошти, яка не має вбудованого захисту, як це було реалізовано спочатку. Спочатку він був розроблений Філом Циммерманом (Phil Zimmermann) у 1991[3]. На той час він був достатньо впливовим, щоб його алгоритми та формати даних були стандартизовані для сумісності між різними частинами програмного забезпечення. Врешті-решт, дизайн PGP був зроблений в Інтернет-стандарті специфікацію, відому як Open PGP. Вона описана в RFC2440 [4]. PGP поєднує в собі симетричну та асиметричну криптографію.

Користувач генерує пару: (відкритий ключ, закритий ключ), яка асоціюється з його унікальним ідентифікатором. Відкриті ключі зберігаються на кільцях відкритих ключів, а приватні ключі зберігаються на кільцях приватних ключів.

На стороні відправника PGP створює сеансовий ключ, який є випадковим числом, згенероване на основі характеристик натискання клавіш користувачем. Після того, як дані зашифровані цим ключем, сеансовий ключ шифрується за допомогою відкритого ключа одержувача і надсилається разом з текстом шифру одержувачу. Копія PGP одержувача використовує свій приватний ключ для відновлення сеансового ключа, що дозволяє одержувачу розшифрувати зашифрований текст. PGP використовує ключову фразу для шифрування закритого ключа на комп'ютері власника. Кодова фраза є довшою і складнішою версією пароля. Закритий ключ зашифрований на диску з використанням хешу ключової фрази як секретного ключа. Для того, щоб використовувати свій закритий ключ, користувач повинен розшифрувати його за допомогою ключової фрази. Розповсюдженням відкритих ключів зазвичай займаються сервери ключів. Вони дзеркально відображаються в різних місцях по всьому світу. Вони володіють відкритими ключами одержувачів і на вимогу відправника надають йому відкритий ключ одержувача.

PGP також можна використовувати для 4 речей:

- Зашифрувати повідомлення або файл так, щоб його міг розшифрувати і прочитати тільки призначений одержувач. Відправник, підписуючи повідомлення за допомогою PGP, також надає гарантію одержувачу, що повідомлення надійшло від уповноваженого відправника, а не від будь-якої неавторизованої особи.
- Чіткий підпис простого текстового повідомлення гарантує, що воно могло надійти тільки від відправника, а не від самозванця.
- Шифрування комп'ютерних файлів таким чином, щоб їх не міг розшифрувати ніхто, окрім особи, яка їх зашифрувала.

- Справжнє видалення файлів (тобто перезапис вмісту так, щоб його не міг відновити і прочитати ніхто інший), а не просто видалення імені файлу з каталогу/папки.

Сертифікат PGP включає (але не обмежується) наступну інформацію [3]:

- Номер версії PGP - номер версії PGP визначає, яка версія PGP була використана для створення ключа, пов'язаного з сертифікатом.
- Відкритий ключ власника сертифіката - містить відкритий ключ разом з алгоритмом ключа, таким як RSA, Elgamal або DSA.
- Інформація про власника сертифіката - це інформація про користувача, така як його ім'я, ідентифікатор користувача, адреса електронної пошти, номер ICQ, фотографія і так далі.
- Цифровий підпис власника сертифіката - його також називають самопідписом. Це підпис, який використовує відповідний закритий ключ відкритого ключа, пов'язаного з сертифікатом.
- Термін дії - це дата/час початку дії сертифіката та дата/час закінчення терміну дії, які вказують на те, коли термін дії сертифіката закінчується. Якщо пара ключів містить під-ключі, то це також включає термін дії кожного з під-ключів шифрування. Підключі дозволяють зручно використовувати окремі ключі для підпису та шифрування
- Бажаний симетричний алгоритм шифрування для ключа - вказує на алгоритм шифрування, за допомогою якого власник сертифіката надає перевагу шифруванню інформації. Підтримуваними алгоритмами можуть бути CAST, IDEA, Triple-DES, Blowfish тощо.

Порівняння. Основні відмінності між PGP та X.509 PKI можна розділити на три області: відмінності у сертифікаті, мережі довіри та процедурі відкликання.

1) Формат сертифікату: Формат сертифікату PGP містить самопідпис, а також може отримати декілька підписів. X.509 підтримує лише один цифровий сертифікат для засвідчення ключа.

2) Ключ: Сертифікат X 509 має лише одне ім'я власника ключа, тоді як сертифікат PGP має відкритий ключ з різними мітками які ідентифікують користувача різними способами.

3) Інтродуктор: У сертифікаті X 509 інтродуктором завжди є ЦС (Центр сертифікації), тоді як в PGP він може використовувати цифровий цифровий підпис в якості інтродуцента.

4) Ланцюг довіри: Коли будь-який користувач підписує ключ іншого користувача, він або вона стає інтродуктором цього ключа. В ході цього процесу цього процесу створюється ланцюжок довіри, тому будь-який користувач може виступати в ролі засвідчувального центру (ЦС). Таким чином, формується багато шляхів сертифікації для досягнення відмовостійкості в якості компенсації за те, що сертифікати підписують аматори. Користувач PGP сертифікат відкритого ключа може також підтвердити сертифікат відкритого ключа іншого користувача PGP. Більше того, такий сертифікат є дійсним для іншого користувача, якщо інша сторона визнає валідатора довіреною особою. Користувач PGP - це той, хто керує ключами, тоді як у випадку з X.509 управління ключами здійснює центр сертифікації.

5) Випуск сертифікатів: В організації, що використовує ІВК з сертифікатами X.509, завданням РЦ є схвалення запитів на сертифікати, а завданням ЦСК - видача сертифікатів. запити на сертифікати, а завданням ЦС є видача сертифікатів користувачам - процес, який, як правило, включає в себе відповідь на запит користувача на сертифікат. В організації, яка використовує сертифікати PGP, завданням ЦС є перевірка автентичності всіх сертифікатів PGP PGP-сертифікатів, а потім підписувати хороші з них.

6) Процедура відкликання: У сертифікатах X.509 відкликаний підпис майже такий самий, як і відкликаний сертифікат, з огляду на те, що єдиним підписом на сертифікаті є той, який зробив його дійсним, тобто підпис центру сертифікації. Сертифікати PGP надають додаткову можливість користувачеві відкликати весь свій сертифікат, якщо він вважає, що сертифікат було скомпрометовано. Відкликати PGP-сертифікат може власник сертифіката або

особа, яка його відкликала. Сертифікат X.509 може відкликати тільки емітент сертифіката. Передача відкликаних сертифікатів X.509 найчастіше здійснюється за допомогою CRL, який публікується центром сертифікації.

7) Формування спільноти: У випадку з сертифікатами PGP користувач зазвичай розміщує відкликаний сертифікат на сервері сертифікатів. Стандарт X.509 не містить жодних вказівок щодо використання перехресних сертифікатів, шляхів сертифікації та CRL, тому користувачі X.509-сертифікатів повинні виконувати ці процедури самостійно. Результатом цього є те, що створення спільнот користувачів X.509 займає більше часу, ніж спільнот користувачів PGP. спільноти користувачів X.509 займає більше часу, ніж спільноти користувачів PGP.

8) Синтаксичні: Це означає, що PGP дозволяє зберігати сертифікати у вигляді стеку, тоді як в X.509 сертифікати пов'язані один з одним так само, як в односторонньому зв'язаному списку (хоча X.509 може також включати синтаксис PGP).

9) Семантичний: Це означає, що PGP дозволяє пов'язувати ключі з реальними особами за правилами довіри в мережі, а не за правилами транзитивної довіри, тоді як в X.509 він прив'язує ключі до імен і приймає транзитивну довіру, хоча належна КЗЗ може також заборонити транзитивну довіру в X.509 як функцію політик ЦС.

1.4 Визначення прогалин та невирішених питань

В епоху, коли домінує цифрова комунікація, безпека онлайн-повідомлень має першорядне значення. З розвитком технологій зростають і виклики, пов'язані із захистом конфіденційної інформації від сторонніх очей.

Наскрізне шифрування і компроміс між зручністю та безпекою. Наскрізне шифрування (E2EE) є золотим стандартом для захисту онлайн-повідомлень, гарантуючи, що лише ті, кому вони призначені, зможуть

розшифрувати їхній вміст. Однак компроміс між надійним шифруванням і зручними для користувача функціями є постійним викликом. Потреба в таких функціональних можливостях, як пошук повідомлень, хмарне резервне копіювання та синхронізація між декількома пристроями, суперечить суворим вимогам безпеки E2EE. Такі інновації, як індексування на стороні клієнта та безпечне хмарне сховище, мають на меті досягти делікатного балансу, зберігаючи безпеку та надаючи користувачам зручність, якої вони очікують.

Дилема управління ключами. Наріжним каменем E2EE є ефективне управління ключами. Проте, створення системи, яка є одночасно безпечною та зручною для користувачів, залишається постійним викликом. Користувачі часто стикаються з проблемою ручної перевірки та управління ключами. Рішення, що включають ключі на основі пристроїв, біометричну автентифікацію та безпечні механізми відновлення ключів, покликані спростити цей процес. Мета полягає в тому, щоб зробити керування ключами інтуїтивно зрозумілим для користувачів без шкоди для загальних цілей безпеки наскрізного шифрування.

Вразливість метаданих. Навіть при застосуванні надійного наскрізного шифрування метадані залишаються вразливими для спостереження. Метадані, що включають інформацію про відправника, одержувача та мітки часу, можуть містити важливу інформацію. Досягнення повного захисту метаданих при збереженні основної комунікаційної функціональності становить значний виклик. Для зменшення цієї вразливості та посилення загального захисту повідомлень активно досліджуються такі протоколи, як "цибулева" маршрутизація та підходи до метаданих, що зберігають конфіденційність.

Загрози квантових обчислень. Привид квантових обчислень, що насувається, створює новий вимір загрози для сучасних алгоритмів шифрування. Потенціал квантових атак для компрометації існуючих криптографічних систем вимагає розробки квантово-стійких алгоритмів шифрування. Перехід існуючих систем на ці нові алгоритми і забезпечення сумісності з пост-квантовим криптографічним ландшафтом представляє величезний виклик як для дослідників, так і для практиків.

Стійкість до фішингу та соціальної інженерії. Незалежно від надійності шифрування, користувачі залишаються вразливими до фішингу та атак соціальної інженерії. Підвищення стійкості до цих загроз передбачає багатогранні стратегії, включаючи навчання користувачів, багатофакторну автентифікацію та вдосконалені алгоритми виявлення загроз. Виклик полягає в розробці технологій, які можуть не лише ідентифікувати та пом'якшувати спроби фішингу, але й розширювати можливості користувачів розпізнавати тактики соціальної інженерії та ефективно протистояти їм.

Баланс між доступом до бекдорів та конфіденційністю. Баланс між конфіденційністю користувачів і законними потребами правоохоронних органів та органів національної безпеки залишається складним завданням. Дебати про доступ до зашифрованих повідомлень через "чорні ходи" піднімають складні юридичні, етичні та технічні міркування. Дослідники прагнуть знайти рішення, які поважають конфіденційність користувачів, забезпечуючи при цьому прозорий і підзвітний процес законного доступу, забезпечуючи крихку рівновагу між безпекою і конфіденційністю.

Безпечний обмін ключами в групових повідомленнях. Поширення наскрізного шифрування на групові повідомлення створює унікальні проблеми з безпечним обміном ключами між кількома учасниками. Динамічні налаштування групи, де учасники можуть приєднуватися або залишати її в будь-який час, посилюють цю проблему. Поточні дослідження зосереджені на розробці протоколів для безпечного обміну ключами в групових повідомленнях, використовуючи такі методи, як ієрархічне управління ключами і децентралізовані системи розподілу ключів для забезпечення безпеки групового спілкування.

Таким чином, невирішені питання в галузі захисту онлайн-повідомлень підкреслюють складність забезпечення безпеки цифрової комунікації в технологічному ландшафті, що швидко розвивається. Дослідники, розробники і політики повинні спільно вирішувати ці проблеми, щоб зміцнити основи

захисту онлайн-повідомлень, забезпечуючи баланс між безпекою, зручністю використання і конфіденційністю в цифровій сфері.

1.5 Висновки до розділу 1

У цьому розділі розглянуто, що захист онлайн-комунікації є важливим питанням у сучасному світі. Оскільки спілкування все частіше відбувається онлайн, кіберзлочинці все частіше націлюються на онлайн-комунікацію. Онлайн-повідомлення вразливі до різноманітних кібератак, включаючи підслуховування, фальсифікацію та крадіжку. Ці атаки можуть мати серйозні наслідки для окремих осіб, організацій та суспільства в цілому.

Існують різні підходи та методи для захисту онлайн-комунікацій. Найпоширеніші з них включають шифрування, автентифікацію та авторизацію. Шифрування зашифровує повідомлення так, щоб їх міг прочитати лише той, кому вони призначені. Автентифікація перевіряє особу відправника та одержувача повідомлення. Автентифікація контролює, хто може надсилати та отримувати повідомлення.

На додаток до цих загальних підходів, існує ряд спеціалізованих методів, які можна використовувати для захисту онлайн-повідомлень. Наприклад, повідомлення електронної пошти можна захистити за допомогою шифрування PGP або S/MIME. Додатки для обміну миттєвими повідомленнями можна захистити за допомогою наскрізного шифрування, як, наприклад, у Signal і WhatsApp.

Хоча сфера безпеки онлайн-повідомлень все ще перебуває на ранніх стадіях розвитку, за останні роки було досягнуто значного прогресу. Дослідники розробляють нові, більш безпечні алгоритми шифрування, механізми автентифікації та протоколи обміну повідомленнями, що зберігають конфіденційність.

Однак все ще існує низка проблем, які потребують вирішення. Одна з них полягає в тому, що спектр загроз постійно розвивається. Постійно розробляються нові кібератаки, тому дослідники повинні постійно розробляти нові способи захисту онлайн-комунікацій. Інший виклик полягає в тому, що захист онлайн-повідомлень повинен бути одночасно безпечним і простим у використанні. Якщо захист онлайн-повідомлень занадто складний у використанні, користувачі не будуть ним користуватися. Дослідники повинні розробляти ефективні і прості у використанні рішення для захисту.

Незважаючи на те, що в галузі безпеки онлайн-повідомлень досягнуто значних успіхів, все ще існує багато проблем, які потребують вирішення. Дослідники працюють над новими та інноваційними способами захисту онлайн-комунікації, і ця сфера постійно розвивається.

З усіх методів, розглянутих у цьому розділі, шифрування виділяється як найефективніший через те, що шифрування забезпечує достатній рівень безпеки.

Розділ 2. РОЗРОБКА МОДЕЛІ ЗАХИСТУ ОНЛАЙН-ПОВІДОМЛЕНЬ

2.1 Новітні технології для підвищення безпеки повідомлень.

В епоху швидкісних технологічних змін та поширеної цифрової взаємодії, захист конфіденційності та цілісності онлайн-повідомлень стає завданням важливішим, ніж будь-коли раніше. У цьому контексті важливим стає використання передових технологічних рішень, спрямованих на підвищення рівня безпеки та ефективності засобів захисту.

Квантове шифрування. Оскільки традиційні методи шифрування стикаються зі зростаючими викликами з боку витончених кіберзагроз, квантове шифрування стає революційним рішенням, що використовує принципи квантової механіки для створення незламного щита для наших цифрових комунікацій.

В основі квантового шифрування лежать принципи квантової механіки - розділу фізики, який описує поведінку матерії та енергії в найдрібніших масштабах. На відміну від класичного шифрування, яке спирається на математичні алгоритми і складність вирішення складних математичних задач, квантове шифрування використовує унікальні властивості квантових частинок для захисту інформації.

Одним з фундаментальних принципів квантового шифрування є використання квантового розподілу ключів (QKD). У класичній системі шифрування ключ обмінюється між двома сторонами, і якщо його перехопити, це становить ризик для безпеки. Квантовий розподіл ключів, однак, використовує принципи квантової суперпозиції та заплутаності для створення ключа, який теоретично неможливо перехопити без виявлення.

Квантова суперпозиція: Квантовий стрибок у безпеці. Однією з ключових особливостей квантової механіки, яка використовується у квантовому

шифруванні, є суперпозиція. У класичних обчисленнях біт може перебувати в одному з двох станів: 0 або 1.

На відміну від цього, квантовий біт або кубіт може існувати в декількох станах одночасно, завдяки суперпозиції. Ця властивість дозволяє створювати квантові ключі, які за своєю суттю є більш безпечними, ніж їх класичні аналоги.

Коли кубіт перебуває в суперпозиції, підслуховувач, який намагається перехопити квантовий ключ, неминуче порушить його стан. Це порушення можна виявити, що забезпечує механізм виявлення та запобігання несанкціонованому доступу до ключа. Акт вимірювання у квантовій механіці за своєю суттю змінює стан системи, що робить будь-яку спробу підслуховування помітною в режимі реального часу.

Квантова заплутаність: Складний зв'язок. Заплутаність - ще одне захоплююче квантове явище, яке відіграє вирішальну роль у квантовому шифруванні. Коли дві частинки заплутуються, стан однієї частинки стає безпосередньо пов'язаним зі станом іншої, незалежно від відстані між ними. Цей зв'язок використовується у квантовому розподілі ключів для створення спільного секретного ключа між сторонами, що спілкуються.

Принадність квантової заплутаності полягає в її нелокальному характері. Навіть якщо заплутані частинки розділені величезними відстанями, зміна стану однієї частинки миттєво впливає на стан іншої. Ця особливість гарантує, що будь-яка спроба перехопити квантовий ключ буде знову виявлена, забезпечуючи рівень безпеки, який теоретично недосяжний при використанні класичних методів шифрування.

Виклики і реальна реалізація. Хоча потенціал квантового шифрування величезний, все ще є значні проблеми, які необхідно подолати, перш ніж воно стане широко застосовуватися. Створення та підтримка стабільних квантових каналів зв'язку, розробка масштабованих систем розподілу квантових ключів і вирішення проблем, пов'язаних з квантовим декодуванням, — ось деякі з перешкод, над подоланням яких активно працюють дослідники.

Реальне впровадження квантового шифрування поступово просувається вперед: кілька компаній і дослідницьких інститутів проводять успішні експерименти і випробування. Уряди, фінансові установи та організації, що працюють з конфіденційною інформацією, особливо зацікавлені в можливостях, які квантове шифрування пропонує для посилення їхнього захисту в сфері кібербезпеки.

Оскільки цифровий простір продовжує розвиватися, повинні змінюватися і наші підходи до захисту конфіденційної інформації. Квантове шифрування стоїть на передовій цієї еволюції, обіцяючи нову еру незламної безпеки завдяки використанню складних і таємничих принципів квантової механіки. Незважаючи на те, що виклики залишаються, потенційні переваги квантового шифрування роблять його променем надії в безперервній боротьбі з кіберзагрозами, забезпечуючи більш безпечне і захищене майбутнє для нашого взаємопов'язаного світу.

Використання технологій блокчейну. У відповідь на загрози технології блокчейн змінюють правила гри у сфері безпеки онлайн-повідомлень, пропонуючи децентралізоване і стійке до несанкціонованого втручання рішення для захисту конфіденційної інформації.

Блокчейн для шифрування повідомлень: Децентралізована фортеця.

Децентралізоване шифрування. Однією з головних переваг використання блокчейну для захисту онлайн-повідомлень є децентралізований характер технології. Традиційні платформи обміну повідомленнями часто покладаються на централізовані сервери, що робить їх вразливими до єдиної точки відмови. На відміну від них, системи обміну повідомленнями на основі блокчейну розподіляють дані через мережу вузлів, що знижує ризик несанкціонованого доступу або маніпуляцій. Кожне повідомлення шифрується і додається до блокчейну, створюючи незмінний запис, що підвищує безпеку і цілісність комунікації.

Записи, стійкі до несанкціонованого доступу. Властивий дизайн блокчейну забезпечує незмінність збережених даних. Як тільки повідомлення

додається до блокчейну, воно стає частиною ланцюжка блоків, пов'язаних між собою криптографічними хешами. Спроба втручання в один блок вимагатиме зміни всіх наступних блоків у ланцюжку, що робить такі зміни практично неможливими без виявлення. Ця функція захисту від підробки додає додатковий рівень безпеки до онлайн-повідомлень, зменшуючи ризики, пов'язані зі зловмисною діяльністю.

Смарт-контракти для безпечної комунікації. Смарт-контракти, угоди, що самостійно виконуються на основі заздалегідь визначених умов, можуть бути використані для автоматизації та захисту різних аспектів онлайн-комунікації. Наприклад, смарт-контракти можуть полегшити наскрізне шифрування, гарантуючи, що повідомлення будуть доступні лише тим, кому вони призначені. Такий децентралізований підхід зменшує залежність від сторонніх організацій і мінімізує ризик порушень безпеки.

Покращене управління ідентифікацією. Технологія блокчейн може революціонізувати управління ідентифікацією в онлайн-обміну повідомленнями. Завдяки безпечній прив'язці ідентифікаційних даних користувачів до криптографічних ключів у блокчейні, користувачі можуть мати більший контроль над своєю особистою інформацією. Це не тільки знижує ризик крадіжки особистих даних, але й гарантує, що обмін повідомленнями відбувається лише між перевіреними та автентифікованими сторонами.

Незмінні аудиторські сліди. У разі виникнення суперечок або юридичних проблем, блокчейн забезпечує незмінний аудиторський слід комунікації. Кожне повідомлення разом з його позначкою часу та криптографічним підписом зберігається в блокчейні, створюючи прозорий запис, який можна перевірити. Ця функція не лише посилює підзвітність, але й слугує цінним інструментом для дотримання нормативних вимог у галузях, де безпечний обмін повідомленнями має вирішальне значення.

Виклики та майбутні перспективи. Хоча потенційні переваги використання блокчейну для забезпечення безпеки онлайн-повідомлень є значними, необхідно вирішити такі проблеми, як масштабованість, адаптація

користувачів і регуляторні міркування. Крім того, розробка зручних для користувача інтерфейсів і забезпечення безперешкодної інтеграції з існуючими платформами обміну повідомленнями є критично важливими для широкого впровадження.

Оскільки технології блокчейн продовжують розвиватися, співпраця між технологічними новаторами, експертами з кібербезпеки та регуляторними органами матиме важливе значення для створення надійної основи для безпечного обміну повідомленнями в Інтернеті. Потенційний вплив на захист конфіденційної інформації, збереження приватності та зміцнення цифрових каналів зв'язку робить дослідження і впровадження блокчейну в безпеку онлайн-повідомлень захоплюючою межею в постійно мінливому ландшафті кібербезпеки.

Використання можливостей штучного інтелекту (ШІ). Кіберзагрози та витонченість зловмисників постійно розвиваються, що вимагає інноваційних рішень. Штучний інтелект (ШІ) стає потужним союзником у зміцненні безпеки онлайн-повідомлень, пропонуючи розширені можливості для виявлення, запобігання та реагування на потенційні ризики.

Середовище онлайн-комунікацій є динамічним і охоплює електронну пошту, миттєві повідомлення, соціальні мережі тощо. Традиційні заходи безпеки, попри їхню важливість, можуть не встигати за швидкозмінним характером кіберзагроз. Саме тут на допомогу приходить штучний інтелект, який використовує свої можливості машинного навчання, розпізнавання образів і аналізу в реальному часі для посилення захисту платформ онлайн-комунікацій.

Машинне навчання для виявлення загроз. Штучний інтелект, зокрема алгоритми машинного навчання, чудово розпізнають закономірності та аномалії. Аналізуючи величезні масиви даних, моделі машинного навчання можуть виявляти аномальні моделі комунікації або потенційні загрози. Такий проактивний підхід дозволяє швидко виявляти зловмисні дії, починаючи від спроб фішингу і закінчуючи більш складними тактиками соціальної інженерії.

Поведінковий аналіз і виявлення аномалій. Поведінковий аналіз на основі штучного інтелекту змінює правила гри в захисті онлайн-повідомлень. Шляхом безперервного моніторингу поведінки користувачів алгоритми ШІ встановлюють базову лінію нормальної активності. Відхилення від цієї базової лінії, що вказують на потенційні загрози безпеці, запускають сповіщення або автоматичні відповіді. Такий адаптивний підхід підвищує здатність запобігати новим і раніше небаченим атакам.

Обробка природної мови (NLP) для контент-аналізу. Обробка природної мови, підмножина ШІ, дозволяє машинам розуміти та інтерпретувати людську мову. У контексті захисту онлайн-повідомлень NLP відіграє вирішальну роль в аналізі контенту. Він може ідентифікувати шкідливий контент, спроби фішингу або інші форми кіберзагроз, оцінюючи мову, контекст і наміри повідомлень.

Автентифікація і контроль доступу на основі штучного інтелекту. ШІ вдосконалює механізми автентифікації, впроваджуючи інтелектуальні засоби контролю доступу. Адаптивна автентифікація на основі ШІ оцінює різні фактори, такі як поведінка користувача, інформація про пристрій і контекстні дані, щоб динамічно коригувати дозволи на доступ. Це гарантує, що лише авторизовані користувачі матимуть доступ до конфіденційних повідомлень та інформації.

Прогностичний аналіз і проактивні заходи безпеки. Прогностичні можливості штучного інтелекту дають змогу вживати випереджувальні заходи безпеки. Аналізуючи історичні дані та визначаючи тенденції, ШІ може передбачати потенційні загрози та вразливості. Таке передбачення дозволяє організаціям впроваджувати проактивні заходи безпеки, залишаючись на крок попереду кіберсупротивників.

Виклики. Хоча штучний інтелект приносить значні переваги для захисту онлайн-повідомлень, важливо визнати і вирішити потенційні проблеми. Етичні міркування, проблеми конфіденційності та ризик хибних спрацьовувань, спричинених ШІ, потребують ретельного калібрування та постійного контролю. Досягнення правильного балансу між безпекою та конфіденційністю

користувачів залишається ключовим аспектом впровадження ШІ в системах захисту повідомлень.

У постійно мінливому ландшафті кібербезпеки використання штучного інтелекту для захисту онлайн-повідомлень є не просто розкішшю, а необхідністю. Здатність ШІ адаптуватися, навчатися та аналізувати закономірності в режимі реального часу дає організаціям можливість посилити свій захист від різноманітних кіберзагроз. З розвитком технологій синергія між людським досвідом та заходами безпеки, керованими штучним інтелектом, відіграватиме ключову роль у забезпеченні конфіденційності та цілісності онлайн-спілкування.

Використання технологій визначення загроз. Технології виявлення загроз відіграють ключову роль у зміцненні безпеки онлайн-повідомлень, пропонуючи динамічний захист від безлічі потенційних ризиків.

Поведінковий аналіз. Технології виявлення загроз використовують передовий поведінковий аналіз для ретельного вивчення моделей поведінки користувачів. Встановлюючи базову лінію нормальної активності, ці системи можуть виявляти відхилення, які можуть свідчити про зловмисні наміри. Наприклад, раптові зміни в моделях спілкування або доступі до конфіденційної інформації можуть викликати тривогу, що спонукає до подальшого розслідування.

Машинне навчання та штучний інтелект. Алгоритми машинного навчання та штучний інтелект (ШІ) є невід'ємними компонентами сучасного виявлення загроз. Ці технології можуть швидко аналізувати великі масиви даних для виявлення закономірностей і аномалій, що вказують на потенційні загрози. Системи, керовані ШІ, можуть адаптуватися і навчатися на новій інформації, що підвищує їхню здатність виявляти нові і раніше невидимі загрози в режимі реального часу.

Виявлення на основі сигнатур. Виявлення на основі сигнатур спирається на базу даних відомих сигнатур загроз. Коли повідомлення збігається з розпізнаним сигнатурою, воно позначається як потенційно шкідливе. Хоча цей

підхід ефективний проти відомих загроз, він може бути недостатньо ефективним для виявлення нових загроз або загроз, що еволюціонують. Поєднання виявлення на основі сигнатур з поведінковим аналізом і машинним навчанням створює більш комплексну стратегію захисту.

Моніторинг в режимі реального часу. Технології виявлення загроз забезпечують моніторинг онлайн-повідомлень в режимі реального часу, що дозволяє негайно реагувати на потенційні загрози. Незалежно від того, чи це незвична спроба входу в систему, підозріле вкладення файлу або відхилення від шаблону комунікації, здатність виявляти та швидко реагувати на них має вирішальне значення для пом'якшення впливу кіберзагроз.

Інтеграція з протоколами шифрування. Інтеграція технологій виявлення загроз з протоколами шифрування підвищує загальний рівень безпеки онлайн-повідомлень. Поєднуючи виявлення загроз з надійними заходами шифрування, організації можуть створити багаторівневий захист, який не тільки захищає конфіденційність повідомлень, але й виявляє та нейтралізує потенційні загрози.

Хоча технології виявлення загроз пропонують потужний захист від кіберзагроз, проблеми залишаються. Хибні спрацьовування, коли легітимна діяльність помилково позначається як загроза, може викликати занепокоєння. Досягнення правильного балансу між чутливістю і специфічністю має вирішальне значення для уникнення непотрібних порушень нормальної комунікації.

Оскільки технології продовжують розвиватися, майбутнє технологій виявлення загроз у сфері безпеки онлайн-повідомлень відкриває захоплюючі можливості. На горизонті інтеграція з квантово-безпечним шифруванням, вдосконалене моделювання поведінки користувачів і розробка більш досконалих систем виявлення загроз на основі штучного інтелекту. Крім того, спільні зусилля фахівців з кібербезпеки, зацікавлених сторін галузі та дослідників мають вирішальне значення для того, щоб випереджати нові загрози.

Використання IoT. Інтернет речей (IoT) швидко трансформував спосіб нашої взаємодії з цифровим світом, об'єднуючи пристрої та системи в безпрецедентний спосіб. Хоча IoT часто асоціюється з розумними будинками та промисловою автоматизацією, його потенціал виходить далеко за межі цих сфер. У сфері безпеки онлайн-повідомлень використання Інтернету речей впроваджує інноваційні рішення, які посилюють захист і цілісність цифрової комунікації.

Захищені протоколи зв'язку. Пристрої Інтернету речей можуть відігравати вирішальну роль у забезпеченні безпеки каналів зв'язку. Інтегруючи надійні та зашифровані протоколи зв'язку, пристрої Інтернету речей сприяють створенню безпечного середовища для обміну повідомленнями. Це особливо важливо в сценаріях, коли кілька пристроїв взаємодіють в мережі, наприклад, в "розумних" будинках або промислових додатках Інтернету речей.

Датчики Інтернету речей для виявлення загроз. Вбудовування датчиків в екосистему IoT забезпечує динамічний засіб виявлення загроз. Ці датчики можуть відстежувати різні параметри, включаючи мережевий трафік, поведінку пристроїв і стан навколишнього середовища. Аномалії, виявлені цими датчиками, запускають сповіщення, допомагаючи попереджати потенційні загрози безпеці онлайн-повідомлень.

Керування ідентифікацією пристрою. Автентифікація та управління ідентифікацією є ключовими для безпеки онлайн-повідомлень. Пристрої Інтернету речей, оснащені унікальними ідентифікаторами та безпечними механізмами автентифікації, сприяють зміцненню інфраструктури управління ідентифікацією. Це гарантує, що лише авторизовані пристрої можуть брати участь у безпечному обміні повідомленнями, зменшуючи ризик несанкціонованого доступу.

Інтелектуальне управління шифруванням. Величезний обсяг даних, що генеруються пристроями Інтернету речей, вимагає інтелектуального управління шифруванням. Впровадження шифрування на різних рівнях екосистеми IoT гарантує, що дані, які генеруються і передаються пристроями, залишаються конфіденційними. Це стає особливо важливим, коли пристрої IoT беруть участь у передачі конфіденційної інформації або повідомлень.

Блокчейн для посилення безпеки. Інтеграція технології блокчейн з пристроями Інтернету речей пропонує децентралізоване і стійке до несанкціонованого втручання

рішення для захисту онлайн-повідомлень. Розподілений реєстр блокчейну підвищує прозорість і цілісність транзакцій повідомлень, що робить надзвичайно складним завданням для зловмисників поставити під загрозу безпеку комунікації.

Моніторинг та реагування в режимі реального часу. Системи моніторингу в режимі реального часу з підтримкою Інтернету речей забезпечують проактивний підхід до безпеки онлайн-повідомлень. Безперервний моніторинг мережевого трафіку, взаємодії пристроїв і факторів навколишнього середовища дозволяє негайно виявляти підозрілі дії. Після цього можуть бути запущені автоматичні реакції для зменшення потенційних загроз до їх ескалації.

Хоча інтеграція Інтернету речей у безпеку онлайн-повідомлень приносить значні переваги, для безперешкодного і безпечного впровадження необхідно вирішити певні проблеми. Різноманітна природа пристроїв Інтернету речей, різні протоколи зв'язку і величезний масштаб взаємопов'язаних пристроїв створюють труднощі в підтримці стандартизованого і безпечного середовища. Крім того, для усунення вразливостей і потенційних зловмисників критично важливим є забезпечення регулярного отримання пристроями Інтернету речей оновлень безпеки.

Майбутнє безпеки онлайн-повідомлень пов'язане з подальшим розвитком та інтеграцією технологій Інтернету речей. Оскільки пристрої Інтернету речей стають все більш поширеними і складними, їх роль у зміцненні цифрової комунікації буде зростати. Досягнення в області периферійних обчислень, виявлення загроз на основі штучного інтелекту і розробка галузевих стандартів для безпечної комунікації через Інтернет, ймовірно, визначатимуть ландшафт безпеки онлайн-повідомлень.

Поєднання Інтернету речей і безпеки онлайн-повідомлень представляє собою захоплюючу межу в сфері кібербезпеки. Використовуючи можливості пристроїв Інтернету речей, ми можемо створити стійкий і динамічний захист від нових загроз. Оскільки ми продовжуємо досліджувати потенціал цього перетину, співпраця між експертами з кібербезпеки, розробниками Інтернету речей та зацікавленими сторонами галузі матиме важливе значення для формування безпечного цифрового майбутнього, в якому онлайн-повідомлення будуть захищені з максимальною ретельністю.

2.2 Принципи та основи розробки моделі захисту

За останні кілька років було випущено багато нових реалізацій шифрування електронної пошти, які намагаються покращити зручність використання шифрування електронної пошти, щоб кожен міг його зрозуміти і легко ним користуватися.

Після проведеного дослідження я з'ясувала, що можу розділити доступне програмне забезпечення для шифрування електронної пошти на три основні категорії. До першого типу відносяться клієнтські програми, які постачають або розширюють існуюче настільне програмне забезпечення для роботи з електронною поштою, здебільшого з ручним керуванням ключами. Зазвичай це додаткові інструменти, які шифрують повідомлення, використовуючи існуючу інфраструктуру електронної пошти. До другого типу належать наскрізні сервіси шифрованої електронної пошти, які виконують шифрування автоматично, а керування ключами також відбувається автоматично на їхніх серверах. Здебільшого це веб-сервіси, і багато з них надають спеціальні облікові записи для підприємств, що дозволяють адміністраторам, наприклад, відновлювати втрачені ключі. Ще однією можливістю безпечного спілкування електронною поштою для організацій, які мають власний поштовий сервер, є шлюзи шифрування електронної пошти. Великою перевагою шлюзів шифрування є те, що з користувача знімається відповідальність за те, яке повідомлення потрібно зашифрувати. Шлюзи забезпечують контроль шифрування повідомлень на основі різних параметрів, включаючи тему, одержувача, відправника, вміст листа, вкладення та багато інших даних. З іншого боку, може бути складно правильно налаштувати політики, щоб уникнути помилкових спрацьовувань.

Однак, оскільки я зосереджуюся на наскрізному шифруванні, що забезпечує конфіденційність і у внутрішній інфраструктурі, не буду обговорювати цей варіант більш детально.

Далі представлено деяких представників першої категорії - плагіни на основі PGP для поштових клієнтів Microsoft Outlook і Mozilla Thunderbird.

Gpg4O: Gpg4O [15] - це надбудова для Outlook, що використовує програмне забезпечення Gpg4win [16] для шифрування електронної пошти. Інтерфейс користувача для ручного керування ключами вимагає багато кроків для генерації ключів та імпорту відкритих ключів, що дозволяє користувачеві використовувати неправильні ключі для шифрування. Шифрування повідомлень не є автоматизованим. Воно вимагає втручання користувача і дозволяє легко ненавмисно відправити повідомлення незашифрованим, коли користувач забуває натиснути на кнопку шифрування перед відправкою повідомлення.

Enigmail: Шифрування краще вирішено в Enigmail [17], доповненні для багатоплатформового поштового клієнта Thunderbird. Вона дозволяє користувачеві встановити автоматичне шифрування, коли доступні відкриті ключі одержувачів, і відображає попереджувальне повідомлення, що вимагає підтвердження користувача, коли повідомлення не зашифровано перед відправленням. Вона вимагає попередньої інсталяції програми GnuPG [12].

Що стосується управління ключами, обидва плагіни надають комплексні інструменти, які дозволяють розповсюджувати відкритий ключ будь-яким з трьох способів:

- Прикріпити відкритий ключ до електронного листа та імпортувати його прямо з вкладення, покладаючись на принцип довіри при першому використанні (Trust On First Use (TOFU) - це модель безпеки, яка передбачає відсутність супротивника при початковому з'єднанні та прийняття початкового ключа без перевірки. Якщо в наступних з'єднаннях використовується інший публічний ключ, він буде вважатися ненадійним. [18]).
- Завантаження та отримання відкритих ключів з сервера ключів. Однак розповсюдження ключів через сервери ключів ще не гарантує, що ключ належить саме тій особі, якій він призначений, оскільки будь-хто може створити і опублікувати ключ на цих серверах ключів, використовуючи будь-який ідентифікатор
- Імпортувати ключ з файлу.

Жоден з цих методів не є достатньо практичним і безпечним для розгортання в масштабах підприємства, коли користувачі повинні обмінюватися ключами один з одним і перевіряти автентичність ключів, перш ніж вони зможуть почати безпечну комунікацію.

Mailpile: Найбільш інтуїтивно зрозумілий користувацький інтерфейс як для шифрування, так і для управління ключами забезпечує поштовий клієнт Mailpile [19] з відкритим вихідним кодом на основі PGP [20], який, однак, наразі існує лише у бета-версії. Ключі генеруються під час створення облікового запису на стороні клієнта, а розповсюдження відкритих ключів є набагато більш автоматизованим. Вона не підтримує функції, що дозволяють використовувати принципи Web of Trust. Замість цього вона покладається на принцип довіри при першому використанні з серверами ключів і традиційною перевіркою за відбитками пальців в якості запасної стратегії. Коли створюється нове повідомлення, воно автоматично шукає ключі одержувачів. Спочатку в локальній зв'язці ключів відправника, а потім на серверах ключів, якщо потрібний ключ не зберігається локально.

Symantec Desktop Email Encryption: Ймовірно, найкращим рішенням для керування ключами є Symantec Desktop Email Encryption [20]. Цей поштовий клієнт автоматично шукає відкриті ключі одержувачів у глобальному каталозі PGP, який є сервером ключів, якому за замовчуванням довіряє поштовий клієнт.

Сервер ключів PGP Global Directory працює як центр сертифікації. Він перевіряє особу користувача за допомогою електронної пошти і підписує його ключ, щоб інші користувачі могли йому довіряти.

Однак користувацький інтерфейс також дозволяє легко відправити електронного листа ненавмисно у вигляді відкритого тексту. Якщо відкритий ключ одержувача не знайдено, лист надсилається незашифрованим без жодних попереджень. Однак це можна налаштувати так, щоб користувач отримував повідомлення про те, що ключ не знайдено, і мав можливість скасувати відправлення повідомлення. Як і в попередньому програмному забезпеченні, можна увімкнути автоматичне шифрування та підписання.

Mailvelope: Рішення для шифрування у веб-клієнтах електронної пошти пропонує Mailvelope, надбудова для браузерів Chrome і Firefox [21]. Оскільки неможливо дізнатися, що робить поштовий клієнт, наприклад, чи зберігає він незашифровану копію повідомлення при введенні, всі приватні дані, такі як вміст повідомлення та ключі, ізольовані від поштового клієнта. Mailvelope відкриває окреме вікно для введення повідомлення, а потім надсилає зашифрований текст назад у поштовий клієнт. Керування ключами здійснюється вручну так само, як і в описаних вище надбудовах для десктопних клієнтів.

S/MIME в Outlook: В той час як реалізації на основі PGP здебільшого доступні як доповнення до багатьох поштових програм, S/MIME інтегрований майже в кожну з найбільш поширених поштових програм. Однією з можливостей використання S/MIME є вбудована функціональність Microsoft Outlook S/MIME [22]. Користувацький інтерфейс тут не надто покращено, порівняно з описаними вище програмами. Також можна налаштувати клієнт на автоматичне шифрування повідомлень. Тоді клієнт вимагає підтвердження користувача для відправлення повідомлення, якщо воно не може бути зашифроване.

Управління ключами в цьому випадку не краще, ніж в інструментах на основі PGP, оскільки користувачі повинні мати справу з отриманням та обміном дійсних сертифікатів. Однак для підприємств, що мають інфраструктуру на базі Windows, можна використовувати служби сертифікації Active Directory для автоматичного розповсюдження сертифікатів серед усіх користувачів, щоб кожен міг надсилати зашифровані електронні листи один одному. Хоча користувачеві все одно потрібно виконати кілька кроків, щоб імпортувати свій сертифікат і опублікувати його для інших, це одне з найбільш практичних рішень, коли нам потрібно встановити безпечний зв'язок лише в межах внутрішньої інфраструктури.

Веб-сервіси: Сьогодні, з широким розповсюдженням мобільних пристроїв, для користувачів важливо мати доступ до своєї електронної пошти на кожному пристрої, яким вони користуються. Тому їм потрібно мати копію

свого приватного ключа на кожному з них. З описаним вище програмним забезпеченням для шифрування користувачі повинні передавати ключі вручну. Цю проблему вирішує останній тип продуктів для шифрування електронної пошти - хостингові поштові сервіси, які забезпечують наскрізне шифрування, а управління ключами здійснюється на їхніх серверах. Ключі синхронізуються на всіх пристроях користувача, тому користувачеві не потрібно переносити їх вручну. Надаю два цікаві приклади з цієї категорії: *ProtonMail* [23] та *Tutanota* [24]. В обох цих сервісах при створенні облікового запису на комп'ютері користувача генерується пара ключів, які зберігаються на серверах разом з профілем користувача, причому як приватний, так і відкритий ключі зберігаються на серверах.

Перед тим, як приватний ключ надсилається на сервер, він шифрується паролем користувача. Хоча ProtonMail базується на стандарті OpenPGP, Tutanota використовує власну схему шифрування, тому вона не сумісна з будь-яким іншим програмним забезпеченням для шифрування електронної пошти. Однак причина, чому автори Tutanota не використовують жодного зі стандартів, полягає в тому, що вони прагнуть підвищити рівень конфіденційності. На відміну від систем на основі PGP або S/MIME, Tutanota також шифрує рядок теми.

Ці сервіси також надають користувачам найкращу зручність у використанні. Єдине, що потрібно зробити - це створити новий обліковий запис електронної пошти, після чого всі повідомлення будуть автоматично зашифровані для користувачів одного сервісу. Обидва сервіси надають обхідний шлях для безпечного спілкування з користувачами, які не мають облікового запису на тому ж сервісі. Він надсилає стандартне повідомлення в поштову скриньку одержувача, надаючи йому посилання для відкриття зашифрованого повідомлення. Щоб розшифрувати лист, одержувач повинен ввести унікальний пароль до повідомлення, який відправник повинен надати йому заздалегідь.

Обидва ці сервіси мають веб-клієнти, а також мобільні додатки для iOS та Android. Крім того, Tutanota пропонує надбудову для Microsoft Outlook та

спеціальні акаунти для підприємств з правами адміністратора, що дозволяють скидати паролі користувачів. Недоліком цих сервісів є те, що вони вимагають довіри до третьої сторони для розповсюдження відкритих ключів без будь-якої можливості перевірити дійсність ключа.

2.3 Опис структури та функцій

Цільове середовище - це доменна мережа на базі Windows з Active Directory, доступною для централізованого управління доменом. Далі я більш детально розгляну середовище Active Directory та можливості використання цього середовища для розповсюдження ключів PGP.

Щодо інфраструктури електронної пошти, лише припускаю, що підприємства використовують клієнта Outlook. Оскільки шифрування/розшифрування виконується на стороні клієнта, немає особливих вимог до поштового сервера або використовуваних протоколів. Outlook підтримує наступні поштові протоколи.

- SMTP (Simple Mail Transfer Protocol) для надсилання повідомлень на поштовий сервер.
- POP3 (Post Office Protocol) та IMAP (Internet Message Access Protocol) для отримання повідомлень з поштового сервера.
- Outlook-Exchange Transport Protocol - власний протокол, який Microsoft Outlook використовує для зв'язку з поштовим сервером Microsoft Exchange. Сервер Exchange часто використовується на підприємствах з архітектурою на основі Windows з використанням поштового клієнта Outlook. Протокол використовує MAPI (Messaging Application Programming Interface), який забезпечує архітектуру обміну повідомленнями для Microsoft Outlook. MAPI надає набір інтерфейсів для маніпулювання даними електронної пошти, створення поштових

повідомлень і папок, а також для сповіщення про зміни в існуючих даних, пов'язаних з MAPI [28].

При роботі з зашифрованими повідомленнями важливо пам'ятати про структуру конкретного протоколу. Наприклад, транспортний протокол Outlook-Exchange синхронізує всі папки та підпапки з поштовим сервером, і якщо електронні листи зберігатимуться в розшифрованому вигляді на стороні клієнта, вони стануть доступними для постачальника послуг електронної пошти. Тому електронні листи повинні залишатися зашифрованими навіть на стороні клієнта.

Модель двіри. На відміну від існуючих рішень, які покладаються на центри сертифікації або провайдерів послуг зашифрованої електронної пошти для безпечного обміну ключами шифрування, я прагну не залучати жодну довірену третю сторону, щоб підвищити рівень конфіденційності корпоративних даних. При розробці мого рішення я розглядаю наступну модель довіри.

Тип супротивника: Як правило, ми розрізняємо пасивних та активних супротивників. Пасивний супротивник здатен лише підслуховувати комунікацію через мережу та аналізувати її. Активний зловмисник може змінювати передані повідомлення, видаляти їх або вставляти свої власні. Наше рішення спрямоване на захист комунікації від активного супротивника.

Ми також розрізняємо зовнішні та внутрішні атаки. Атака ззовні ініціюється нелегітимним користувачем системи. Інсайдерська атака ініціюється автентифікованим користувачем системи. У нашому випадку інсайдером буде зловмисний співробітник, який намагається отримати доступ до електронної пошти інших співробітників. Це рішення спрямоване на захист електронної пошти як від зовнішніх, так і від внутрішніх атак, за одним винятком - адміністратора системи. Адміністратор розглядається як довірена особа в нашій системі з можливістю доступу до приватних ключів користувачів.

На відміну від існуючих рішень, які мають автоматизоване управління ключами, ми також розглядаємо постачальника послуг електронної пошти як потенційного зловмисника.

Враховані загрози: При розробці системи зберігання та розповсюдження приватних ключів ми припускаємо, що зловмисник може отримати фізичний доступ до пристрою користувача. Однак ми припускаємо, що він не може автентифікуватися в обліковому записі домену користувача.

Ми також вважаємо, що зловмисник може отримати доступ до поштової скриньки користувача на поштовому сервері.

Криптографічні властивості: Я прагну забезпечити автентифікацію повідомлень, конфіденційність, неспростовність і цілісність. Я не ставлю за мету забезпечити пряму секретність, оскільки це нелегко зробити для асинхронного зв'язку.

Функціональні вимоги. *Мета* - автоматизувати все, окрім необхідних завдань, які потребують втручання користувача. Оскільки розглядаємо адміністраторів як довірених осіб, вони візьмуть на себе відповідальність за необхідні ручні завдання з управління ключами, щоб ми досягли майже нульової взаємодії з системою шифрування для кінцевого користувача.

Рішення повинно дозволити адміністраторам виконувати наступні завдання:

- Генерувати та зберігати нові ключі для кінцевих користувачів у захищеному центральному сховищі всередині підприємства.
- Відкликати ключі, коли вони стають скомпрометованими.
- Встановлювати зв'язок з іншими підприємствами.

Кінцевий користувач не повинен докладати додаткових зусиль для надсилання зашифрованої електронної пошти. Взаємодія з кінцевим користувачем буде потрібна лише в одному випадку. Коли ключі шифрування недоступні, але деякі з одержувачів повинні отримувати зашифровані електронні листи - вони з того ж підприємства або з якогось пов'язаного підприємства. У такому випадку користувач повинен мати можливість вирішити

надсилати незашифровані листи, щоб ми не відключали відправлення електронної пошти взагалі, коли ключі одержувачів недоступні.

Надбудова Outlook має виконувати шифрування та дешифрування наступним чином.

Надсилання підписаних і зашифрованих листів:

- Автоматично підписувати та шифрувати електронний лист разом з вкладеннями під час надсилання, якщо доступні відкриті ключі для всіх одержувачів.
- Відображення інформації про те, чи доступні відкриті ключі дійсних одержувачів, у графічному інтерфейсі Outlook GUI (Графічний інтерфейс користувача).
- Якщо одержувач є співробітником підприємства або пов'язаного з ним підприємства і його відкритий ключ недоступний, то запитує користувача, чи можна надіслати повідомлення у вигляді простого тексту.

Розшифрування та перевірка підпису:

- Автоматично розшифровувати електронні листи при відкритті.
- Відображати інформацію про підпис повідомлення та дійсність ключа у графічному інтерфейсі Outlook.

Зберігання електронної пошти: Як вже згадувала раніше, електронні листи повинні зберігатися в зашифрованому вигляді на стороні клієнта, оскільки будь-які зміни можуть бути синхронізовані з поштовим сервером. Тому зміни ніколи не повинні зберігатися в оригінальному повідомленні після того, як лист буде розшифровано.

Більшість існуючих клієнтських рішень зберігають повідомлення в оригінальному зашифрованому вигляді. Однак деякі з них забувають про чернетки. Чернетки можуть містити конфіденційну інформацію, і користувач не має можливості зашифрувати їх до моменту відправлення. Тим часом поштова скринька синхронізується з сервером і відкриває чернетки у вигляді звичайного текстом. Тому наше рішення має шифрувати чернетки автоматично при збереженні.

Звідси випливає ще одна проблема зі зберіганням листів у зашифрованому вигляді. Нам потрібно зберігати старі прострочені ключі, щоб мати можливість читати старі листи. При розшифровці листа з простроченим приватним ключем нам не потрібно інформувати користувачів про те, що ключ не дійсний, оскільки вони нічого не можуть з цим вдіяти. Однак, при перевірці підпису з простроченим або відкликаним ключем, ми повинні повідомити користувача про дійсність підпису. Однак ми не знаємо, чи був підпис створений вчасно, коли ключ був ще дійсним, чи він був підписаний вже недійсним ключем. Тому користувач повинен бути поінформований про можливе порушення цілісності.

Можна було б вирішити цю проблему, порівнюючи дату отримання листа з датою закінчення терміну дії/відкликання та інформувати користувача про можливе порушення цілісності лише в тому випадку, якщо лист було отримано після того, як ключ став недійсним. Однак, цілісність заголовків електронного листа, включаючи дату отримання, не захищена, і заголовки можуть бути легко підроблені. Отже, це буде проблемою безпеки. Цю проблему можна вирішити за допомогою спеціальних атрибутів об'єктів електронної пошти.

Керування приватними ключами: Однією з найважливіших функцій для користувачів електронної пошти сьогодні є можливість доступу до своєї електронної пошти з різних пристроїв. Таким чином, нам потрібно автоматично передавати не лише відкриті ключі між усіма користувачами, але й закриті ключі між усіма пристроями користувача.

Служби шифрування електронної пошти, описані в попередньому розділі, обробляють приватні ключі від імені користувачів, тому вони можуть отримати доступ до своєї поштової скриньки з будь-якого пристрою. Однак ключі зберігаються на серверах провайдера і захищені лише парольною фразою користувача. Оскільки багато користувачів обирають слабкі паролі, цей захист є недостатнім.

Оскільки ми розглядаємо адміністратора нашого цільового системного середовища як довірену особу, ми можемо використовувати інфраструктуру підприємства для централізованого управління закритими ключами та

безпечного розповсюдження їх серед кінцевих користувачів за допомогою існуючих механізмів автентифікації та авторизації.

Управління відкритими ключами. Мета - повністю автоматизувати розподіл, сертифікацію та перевірку відкритих ключів, щоб зняти цю відповідальність з користувача. Коли шифрування відбувається в межах одного підприємства, нам не потрібно мати жодних сертифікатів на відкриті ключі, оскільки вони можуть поширюватися лише через внутрішню інфраструктуру. Однак, коли залучені інші підприємства, нам доводиться поширювати ключі через Інтернет і додавати до них сертифікат, щоб користувачі інших підприємств могли перевірити автентичність відкритих ключів.

Ми не хочемо покладатися на стандартні центри сертифікації, оскільки на практиці може виявитися багато недоліків у безпеці, а отримання сертифікату від ЦС є обтяжливим і дорогим процесом. Крім того, принципи павутини довіри не підходять і не масштабуються для використання на підприємствах, як ми обговорювали в попередньому розділі.

Іншим варіантом перевірки автентичності відкритих ключів є перевірка їхніх відбитків пальців. Це також не є практичним для розгортання в масштабах підприємства. Однак, якщо ми використовуємо один головний ключ підприємства для сертифікації відкритих ключів працівників, то достатньо буде перевірити відбитки пальців лише для головного ключа, який потім буде використовуватися для перевірки сертифікатів працівників. У цьому випадку перевірка за відбитками пальців буде найпростішим варіантом перевірки.

Автоматичне оновлення ключів. Мета - розробити рішення, яке не потребує подальшого обслуговування після розгортання. Тому необхідно автоматично генерувати нові ключі після закінчення терміну дії старих.

Щоб гарантувати, що користувачі завжди мають під рукою дійсну пару ключів PGP, нам потрібно реалізувати автоматичну перевірку центрального сховища ключів користувачів, яке регулярно перевіряє терміни дії та генерує нові ключі, коли термін дії поточних ключів закінчується.

2.4 Розробка рішення захисту в рамках моделі

Під час аналізу проблем я виявила, що багато проблем, пов'язаних з управлінням закритими ключами, можна вирішити, використовуючи існуючу інфраструктуру підприємства в нашій системі шифрування електронної пошти.

Середовище Active Directory. Active Directory (AD) - це служба каталогів для доменних мереж Windows. Це набір процесів і служб, які автоматизують мережеве управління даними користувачів, безпекою і розподіленими ресурсами.

Однією з таких служб є, наприклад, служба сертифікатів [29], яку можна використовувати для розповсюдження сертифікатів для вбудованої підтримки шифрування S/MIME в Outlook, як ми згадували у другому розділі. Однак, ця служба не є достатньою для наших цілей, оскільки вона може забезпечити генерацію та розповсюдження сертифікатів лише в межах доменної мережі.
Найважливішою службою є служба домену Active Directory Domain Ser- vice (AD DS), яка зберігає інформацію про членів домену, включаючи пристрої та користувачів, перевіряє їхні облікові дані та визначає їхні права доступу [30]. Сервер, на якому працює ця служба, називається контролером домену.

Схема Active Directory Schema визначає атрибути та класи, що використовуються в AD DS. Базова схема містить багатий набір визначень класів, таких як User (Користувач), Computer (Комп'ютер) і Group (Група), а також визначень атрибутів, таких як userPrincipalName або objectSid. Крім того, базова схема може бути розширена користувацькими класами та атрибутами для зберігання користувацьких даних в Active Directory [31].

Для доступу до об'єктів у базі даних AD DS Active Directory підтримує протокол полегшеного доступу до каталогів (LDAP). LDAP це прикладний протокол для доступу та обслуговування розподілених служб каталогів через мережу Інтернет-протоколу [32].

Active Directory має автоматизоване управління для встановлення захищеного каналу між контролером домену та клієнтськими комп'ютерами, що

входять до домену [33], тому можна використовувати цю інфраструктуру для розповсюдження будь-яких конфіденційних даних кінцевим користувачам, таких як приватні ключі.

Архітектура. На основі вимог та аналізу проблеми я спроектувала систему шифрування з наступною архітектурою.

Я вирішила використовувати внутрішню інфраструктуру підприємства, оскільки це значно спрощує реалізацію централізованого управління ключами. Основна перевага полягає в тому, що не потрібно створювати нові облікові записи користувачів, тому користувачам не потрібно запам'ятовувати нові паролі.

Закриті ключі зберігатимуться в розширеному класі Active Directory User, який містить новий користувацький атрибут. Відкриті ключі зберігатимуться на публічному сервері ключів PGP, сертифікованому головним приватним ключем підприємства.

Пара майстер-ключ також зберігатиметься в AD. Головний відкритий ключ буде доступний для всіх користувачів з домену, щоб вони могли перевіряти сертифікати відкритих ключів, завантажених з сервера ключів. Крім того, головний відкритий ключ буде завантажено на сервер ключів, щоб інші компанії могли його завантажити. Для перевірки головного ключа даного підприємства, завантаженого з сервера ключів, адміністратор повинен отримати відбиток головного ключа від даного підприємства.

Розроблена система включає в себе адміністративний інструмент для управління ключами та їх конфігурації, а також надбудову Outlook для автоматизованого шифрування. Надбудова Outlook підключається до AD для отримання приватних ключів поточного користувача та головних відкритих ключів усіх залучених компаній. Надбудова також зв'язується з сервером відкритих ключів, щоб отримати сертифіковані відкриті ключі інших користувачів. Інструмент адміністратора також взаємодіє з обома - сервером ключів і AD - для зберігання згенерованих ключів.

Останній компонент системи шифрування - це проста програма для регулярної перевірки терміну дії ключів PGP. Програма перевіряє, чи всі користувачі в Active Directory мають дійсні пари ключів. Коли термін дії ключа закінчується, програма автоматично генерує новий ключ і завантажує його відкриту частину на сервер ключів. Ця програма запускатиметься регулярно, наприклад, раз на день, за допомогою Планувальника завдань Windows на контролері домену.

Контролери домену Active Directory зазвичай доступні тільки з локальної мережі, і не рекомендується виставляти їх на загальний огляд в Інтернеті. Щоб отримати доступ до ключів в AD з-за меж локальної мережі, співробітникам доведеться підключити локальну мережу через VPN (віртуальну приватну мережу). Іншим варіантом може бути зберігання копії ключів на комп'ютері користувача. Закриті ключі можна було б безпечно зберігати на комп'ютері користувача, зашифровані паролем користувача з використанням сховища сертифікатів Windows [34]. Не реалізуємо локальне зберігання закритих ключів у нашій перевірці концепції, оскільки це не є необхідним для тестування функціональності системи шифрування.

Загальний розподіл ключів показано на рисунку 2.1.

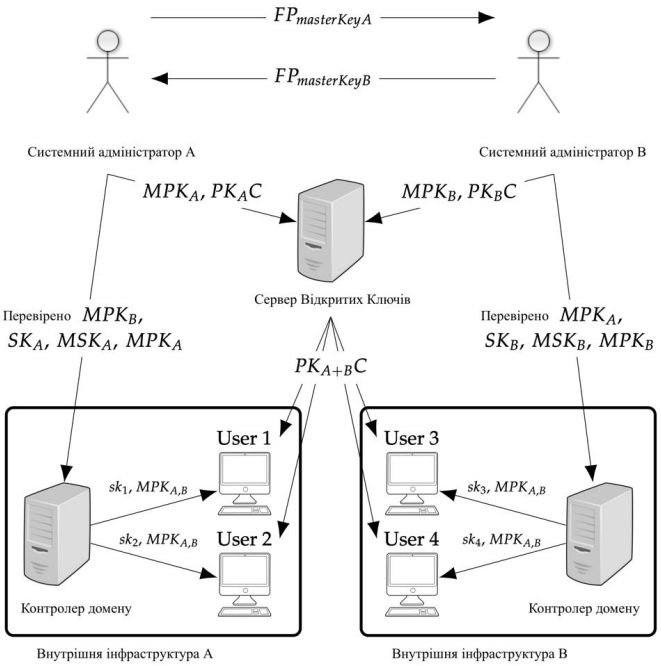

Рис. 2.1 Архітектура управління ключами.

- $FP_{masterKeyX}$позначає відбиток головного ключа адміністратора з підприємства X.
- $PK_X C$- набір відкритих ключів користувачів підприємства X, засвідчених головним ключем підприємства.
- SK_Xпозначає набір секретних зв'язок ключів користувачів підприємства X. sk_yпозначає набір секретних ключів користувача під номером y.
- MPK_X / MSK_X позначає головний відкритий/секретний набір ключів підприємства X відповідно.

Зберігання приватних ключів. Для зберігання закритих ключів користувачів в AD ми використовуємо спеціальний атрибут, доданий до класу *User* з назвою *pgpPrivateKey*. Атрибут *pgpPrivateKey* конкретного об'єкта користувача повинен бути доступний тільки цьому користувачеві як власнику. Цього можна досягти, встановивши права доступу за допомогою принципу безпеки SELF.

Однак, стандартного управління правами доступу в Active Directory недостатньо для приховування певних атрибутів, оскільки, наприклад, коли ми забороняємо право на читання певного атрибуту, це не переписує успадкований дозвіл на читання всіх атрибутів, встановлений для всього об'єкту. Однак, AD надає можливість додаткових перевірок безпеки, які контролюють доступ до вибраних атрибутів - встановлення так званого біту конфіденційності. Біт конфіденційності встановлюється як біт 7 у властивості searchFlags відповідного об'єкта attributeSchema у схемі AD. Додавання значення 128 до будь-якого існуючого значення у властивості searchFlags позначає атрибут як конфіденційний [27].

Тоді цей атрибут залишається прихованим для всіх, окрім користувачів зі спеціальним привілеєм *control_access*, встановленим безпосередньо для цього атрибуту. Адміністратори мають цей привілей за замовчуванням, і ми повинні встановити його також для кожного користувача за допомогою принципу безпеки SELF [35].

Щоб мати доступ до старих листів, нам потрібно зберегти старі ключі в AD. Це можливо, якщо встановити атрибут pgpPrivateKey як багатозначний. Однак існують обмеження на розмір атрибутів схеми AD, оскільки Active Directory не підтримує потокову передачу значень атрибутів. Тому слід уникати зберігання там великих багатозначних атрибутів.

Microsoft рекомендує, щоб жодне значення атрибута не перевищувало 500 кілобайт, включаючи суму багатозначних атрибутів. Також цілі об'єкти не повинні перевищувати 1 мегабайт [36]. Однак файл секретного кільця ключів з ключами довжиною 2048 біт має розмір близько 6 кілобайт. Таким чином, якщо ми встановимо термін дії ключа в один рік і будемо оновлювати ключі щороку, ми зможемо зберігати резервну копію ключів протягом декількох десятиліть без значного впливу на продуктивність AD DS.

Інструмент адміністрування. Метою адміністративного компонента є надання простого графічного інтерфейсу для полегшення завдань, пов'язаних з управлінням ключами, таких як генерація ключів, зберігання їх в Active Directory та завантаження на сервер ключів PGP. Основна мета - звести до мінімуму взаємодію адміністратора з системою.

Програма надає адміністратору наступні функціональні можливості:

- Відображення таблиці всіх користувачів домену з основною інформацією про їхні останні ключі.
- Генерація та автоматичне розповсюдження ключів користувачів.
- Відображення деталізації користувача зі списком всіх раніше згенерованих ключів з можливістю їх відкликання.
- Генерація, відкликання та автоматичне розповсюдження головного ключа.
- Відображення списку підключених підприємств.
- Встановлення з'єднання з іншим підприємством. Ця дія вимагає від адміністратора виконання наступних кроків:

  – Заповніть доменне ім'я електронної пошти підприємства. Після цього програма шукає на сервері ключів майстер-ключ з ідентифікатором користувача PgpMasterKey@<домен_пошти_підприємства>.

– Отримайте відбиток головного ключа підприємства і введіть його в підготовлену форму. Потім програма перевіряє головний ключ і, якщо він дійсний, зберігає його в AD. Цей ключ потім використовується для перевірки відкритих ключів користувачів з підключеного підприємства.

Надбудова для Outlook. Перш ніж розробити надбудову для Outlook, нам потрібно дослідити, як можна обробляти певні події, повідомлення електронної пошти і як можна розширити графічний інтерфейс Outlook. У цьому розділі буде представлені можливості надбудови Outlook і детально розглянуться, як їх можна використати для задоволення вимог до автоматизованого шифрування електронної пошти, які ми визначили в попередньому розділі.

Розробка надбудов для Microsoft Office Outlook можлива через взаємодію з об'єктами, які надаються об'єктною моделлю Outlook [37]. Об'єктна модель Outlook надає класи та інтерфейси, які представляють елементи користувацького інтерфейсу Outlook. Об'єктна модель складається з восьми об'єктів верхнього рівня. Найбільш важливими та релевантними для наших потреб є наступні об'єкти:

- Додаток: Об'єкт найвищого рівня, що представляє всю програму Outlook і надає доступ до інших об'єктів в ієрархії Outlook, таких як Провідники та Інспектори.
- Провідник: Являє собою вікно, в якому відображається вміст папки. Метод *ActiveExplorer* повертає об'єкт, що відповідає поточному активному провіднику.
- Інспектор: Відображає вікно, у якому відображаються елементи Outlook, такі як повідомлення, зустрічі або завдання. Метод *ActiveInspector* повертає об'єкт, що представляє поточний активний інспектор.
- MailItem: Представляє поштове повідомлення.

Потрібно автоматизувати шифрування та дешифрування на основі певних подій в Outlook. Outlook надає широкий спектр подій, які можуть сповіщати програми про зміну об'єкта або про виконання певної дії. Щоб отримати таке повідомлення, необхідно написати метод-обробник події.

Існує два типи подій Outlook: події на рівні об'єктів і події на рівні додатків. Події на рівні елемента сповіщають програму про те, що елемент було відкрито, надіслано, збережено або закрито, а також про те, що користувач відповів на повідомлення або переслав його. Події на рівні елемента можуть також повідомляти програму, коли користувач клацнув елемент керування на формі або коли змінилася властивість елемента.

Події на рівні програми пов'язані з самою програмою, дослідниками, інспекторами та теками. Ці події відбуваються, наприклад, щоразу, коли надсилається елемент Outlook або коли до поштової скриньки надходить новий лист.

Шифрування та підписання: Щоб автоматично підписувати і шифрувати електронний лист перед надсиланням, нам потрібно написати обробник події Application.ItemSend. Крім шифрування листа ключами всіх одержувачів, ми також шифруємо його ключем відправника, щоб він міг прочитати надіслані повідомлення.

Шифрування чернеток: Ми шифруємо чернетки за допомогою події Mailitem.Write, яка спрацьовує при збереженні листа. Однак, оскільки імейли автоматично зберігаються перед відправленням, обидві події Item.Sent і Mailitem.Write спрацьовують перед відправленням. Отже, нам потрібно переконатися, що вони не будуть зашифровані двічі. Ми обговоримо вирішення цієї проблеми у наступному розділі, де буде описано реалізацію.

Розшифрування та перевірка підпису: Ми прагнемо автоматично розшифровувати електронні листи, коли вони відображаються в інтерфейсі користувача Outlook.

Інтерфейс Outlook дозволяє користувачеві відображати вміст листа або у вікні з окремими розділами, або на панелі попереднього перегляду.

Ці вікна представлені об'єктами інспектора. Написавши метод-обробник для події Application.NewInspector, ми можемо розшифрувати повідомлення автоматично при відкритті листа.

Розшифрування листа під час відображення його в області попереднього перегляду можна виконати за допомогою події *Explorer.SelectionChanged*. Ця подія повертає поточне повідомлення, що відображається в області читання, як перший елемент у всьому масиві вибору.

При закритті вікна або зміні виділення в області читання незбережені зміни в електронному листі мають бути відкинуті, щоб лист зберігався у зашифрованому вигляді.

Є проблема, пов'язана з перевіркою підписів з простроченими сертифікатами. Коли користувачі відкривають лист, вони отримують інформацію про стан перевірки підпису. Наприклад, якщо для електронного повідомлення є дійсний підпис, а потім користувач відкриває той самий лист знову після того, як ключ було відкликано або термін його дії закінчився, надбудова Outlook намагається перевірити підпис ще раз. Однак, оскільки відповідний відкритий ключ більше не є дійсним, надбудова не повинна вважати цей підпис дійсним, оскільки ми не знаємо, чи використовувався ключ для підпису до або після відкликання або закінчення терміну дії.

Можливим рішенням може бути зберігання інформації про перевірку підпису разом з повідомленням у локальній поштовій скриньці. API Outlook дозволяє створювати власні властивості елементів або папок за допомогою об'єкта UserProperties [38]. Додавши нову властивість до об'єкта mailItem, ми могли б зберігати там інформацію про те, що підпис було перевірено. Однак це вирішує проблему частково, оскільки користувач може відкрити лист зі старим підписом на іншому пристрої, де ця інформація не зберігається. У нашій перевірці концепції ми просто інформуємо користувача про те, що ключ підпису більше не дійсний і що підпис не може бути перевірений.

Налаштування користувацького інтерфейсу Outlook. Нам потрібно інформувати користувача про стан шифрування та дійсність підпису, коли користувач створює або читає електронний лист. З цієї причини ми дослідили можливості кастомізації інтерфейсу користувача Outlook.

Outlook підтримує наступні два типи користувацьких панелей інтерфейсу, які дозволяють додавати елементи керування, що можуть відображати дані [39].

- Користувацькі панелі завдань - панелі, які можна прикріпити до будь-якої сторони вікна програми Outlook.
- Користувацькі області форм, які можуть розширювати будь-яку існуючу форму Outlook.

Ми вирішили використовувати області форм, оскільки їх можна автоматично прив'язати до Панелі читання або Інспекторів, тоді як користувацькі панелі завдань потрібно було б обробляти окремо у коді кожного разу, коли відкривається новий Інспектор.

Outlook використовує інформацію з реєстру для завантаження реквізитів форм. Тому інсталятор надбудови Outlook повинен додати запис для кожної області форми і пов'язане з ним значення, яке представляє назву надбудови Outlook до реєстру [32].

Розширення схеми Active Directory. Нам потрібно зберігати наступні дані в Active Directory:

- Закриті ключі всіх користувачів.
- Домени електронної пошти, для яких потрібне шифрування.
- Закритий майстер-ключ підприємства, який буде використовуватися для сертифікації відкритих ключів користувачів.
- Відкриті головні ключі всіх підприємств, залучених до системи шифрування. Ці ключі будуть використовуватися для перевірки сертифікатів користувачів.

Для зберігання приватних ключів користувачів в AD ми розширили стандартний об'єкт User зі схеми AD багатозначним конфіденційним атрибутом pgpPrivateKeys, доступ до якого надається лише користувачеві, якого представляє цей об'єкт.

Решта даних зберігатиметься у екземплярі новоствореного класу з назвою *PgpConfiguration*, що містить такі багатозначні атрибути:

- *encryptedDomains* - доступний для всіх користувачів.

- *trustedPublicMasterKeys* - доступний для всіх користувачів.
- *privateMasterKeys* - конфіденційний, доступний лише адміністратору.

Схему Active Directory можна розширити наступними способами [33].

- Вручну, за допомогою інструменту командного рядка LDIFDE, який є частиною операційної системи Wndows Server. Інструмент LDIFDE дозволяє імпортувати файли LDIF (LDAP Data Interchange Format). LDIF - це стандарт, який визначає формат даних, що використовується для зв'язку з LDAP-каталогами [34].
- Програмно, за допомогою коду на C++ з використанням інтерфейсів Active Directory Service Interfaces (ADSI) [35].

Хоча програмне розширення дозволило б нам скоротити процес розгортання, рекомендується встановлювати розширення схеми окремо від решти додатку. Це дозволяє клієнтам виконати оновлення схеми заздалегідь до решти інсталяції, а оскільки формат LDIF дозволяє представити дані LDAP у зручному для читання вигляді, адміністратор може перевірити, що саме буде змінено. Тому ми використали формат LDIF для розширення схеми AD.

Коли створюється новий клас або атрибут, йому має бути присвоєно унікальний ідентифікатор об'єкта (OID), згенерований з базового OID Active Directory. Крім того, для створення запису LDIF потрібно поточне доменне ім'я Windows. Щоб легко підготувати файл LDIF для розширення схеми, ми створили сценарій Powershell, який спочатку генерує OID, потім розпізнає ім'я поточного домену і генерує сценарій LDIF з правильно заповненими OID і ім'ям домену. Для генерації OID ми використали скрипт, наданий Miscrosoft [36].

Додавання атрибуту до класу зі схеми AD за замовчуванням: Не рекомендується додавати нові атрибути безпосередньо до класів зі схеми за замовчуванням. Замість цього слід створити допоміжний клас, що містить необхідний атрибут at-, і додати його як значення атрибуту *auxiliaryClasses* до оригінального класу [37].

Ми створили новий допоміжний клас pgpUser з атрибутом *pgpPrivateKeys* і додали його до класу *User* за замовчуванням.

У наступному прикладі показано частину LDIF-скрипту (Рис. 2.2), який створює атрибут конфіденційності *pgpPrivateKeys*. Конфіденційність встановлюється значенням 128 в атрибуті *searchFlags*, як ми згадували у попередній главі. Дивіться [38] для отримання додаткової інформації про створення та модифікацію атрибутів і класів у схемі AD.

```
dn: CN=pgpPrivateKeys,CN=Schema,CN=Configuration,
                                        DC=safetica,DC=com
changetype: add
objectClass: attributeSchema
attributeID: 1.2.840.113556.1.8000.999999.2
attributeSyntax: 2.5.5.12
oMSyntax: 64
isSingleValued: FALSE
lDAPDisplayName: pgpPrivateKeys
searchFlags: 128
```

Рис. 2.2 Приклад частини LDIF-скрипту.

Надання доступу до кастомного атрибуту: Коли створюється новий користувацький атрибут, доступ за замовчуванням надається лише адміністратору. Отже, ми повинні призначити всім користувачам права на читання для атрибутів *encryptedDomains* і *trustedPublicMasterKeys*. Дозволи також можна надати за допомогою LDIF-скрипта, як у наступному прикладі (Рис. 2.3).

```
dn: DC=safetica,DC=com
changetype: modify
add: aci
aci: (targetattr="encryptedDomains ||
        trustedPublicMasterKeys")
     (version 3.0; acl "read all";
        allow(read) userdn="ldap:///all";)
```

Рис. 2.3 Надання дозволів за допомогою LDIF-скрипта.

Надання доступу до користувацького конфіденційного атрибуту: вище я вже згадувала, що для доступу до конфіденційного атрибуту користувач

повинен мати дозвіл *control_access*, встановлений безпосередньо для цього атрибуту. Хоча базові права доступу, що підтримуються LDAP, можуть бути встановлені за допомогою інструменту LDIFDE, розширені права, такі як *control_access*, повинні бути встановлені іншим способом.

Можливість встановлення розширених прав доступу надає інструмент Dsacls - інструмент командного рядка, вбудований в операційну систему Windows Server [39]. Записи контролю доступу, що додаються за допомогою Dsacls, повинні бути специфічними для об'єкта, які перевизначають дозволи за замовчуванням, визначені в схемі Active Directory для цього типу об'єкта [39]. Тому, щоб надати доступ до закритих ключів усім користувачам, адміністратор повинен встановити дозволи для кожного користувача окремо. Коли ми тестували нашу перевірку концепції, ми встановили ці права вручну. Однак цю частину процесу розгортання можна усунути, написавши скрипти з використанням об'єктної моделі ADSI, яка дозволяє встановлювати розширені права на схему AD, а не тільки на конкретні об'єкти. [40].

Бібліотека Bouncy Castle OpenPGP. Для шифрування та генерації ключів ми використали бібліотеку Bouncy Castle OpenPGP з відкритим вихідним кодом, яка реалізує стандарт OpenPGP на мові C#. Бібліотека надає високорівневий API для шифрування PGP.

Шифрування, підписання та генерація ключів здійснюється в основному за допомогою наступних класів:

- *PGPEncryptedDataGenerator* - генератор зашифрованих об'єктів.
- *PGPSignatureGenerator* - генератор підписів PGP.
- *PGPKeyRingGenerator* - генератор пар ключів PGP.

Оскільки для підпису та шифрування рекомендується використовувати різні ключі (кожен ключ з окремої пари ключів), ми генеруємо одну пару ключів для шифрування та одну пару ключів для підпису. В іншому випадку якщо одна з операцій буде скомпрометована, це може порушити властивості безпеки інших операцій, де використовується той самий ключ.

У цій бібліотеці кільце ключів PGP складається з головного ключа і набору пов'язаних з ним під-ключів. Ключі підпису та шифрування розрізняються за допомогою ключових прапорців, представлених інтерфейсом *PgpKeyFlags*. Відкриті та секретні кільця ключів обробляються класами *PGPPublicKeyRing* та *PGPSecretKeyRing* відповідно. Секретний кільцевий ключ містить як відкритий, так і закритий ключі. Один об'єкт ключа представляється класами *PGPPublicKey* або *PGPPrivateKey*. Класи *PGPSecretKeyRingBundle* та *PGPPublicKeyRingBundle* представляють колекцію кілець ключів і слугують для зчитування ключів з потоку даних, що містить декілька кілець ключів [50].

Зв'язок з сервером ключів OpenPGP. Більшість серверів ключів OpenPGP підтримують зв'язок через протокол HTTP Keyserver Protocol (HKP) [42]. Протокол HKP визначає можливості для реалізації API сервера ключів OpenPGP з використанням протоколу передачі гіпертексту (HTTP).

Відкриті ключі PGP можуть бути програмно витягнуті з сервера ключів шляхом надсилання простого HTTP GET-запиту.

```
https://pgp.mit.edu/pks/lookup?op=get&search=0x99242560
                                                &options=mr
```

Рис. 2.4 Приклад URI запиту для отримання ключа з ідентифікатором 99242560 з сервера ключів pgp.mit.edu

Опція "mr" вказує серверу повернути дані у машинозчитуваному форматі. Надсилання даних на сервер ключів здійснюється за допомогою HTTP POST-запиту з тілом запиту, закодованим у вигляді URL-адреси. Тіло запиту (Рис. 2.6) містить змінну "keytext", яка встановлюється в значення ASCII-армований набір ключів.

```
https://pgp.mit.edu/pks/add
```

Рис. 2.5 Приклад URI запиту на надання ключа

```
keytext=-----BEGIN+PGP+PUBLIC+KEY+BLOCK--- ...
```

Рис. 2.6 Тіло запиту

Зв'язок з AD DS. Для легкого доступу до об'єктів, якими керує AD DS у коді, Microsoft надає наступні API для зв'язку з LDAP.

- *System.DirectoryServices*
- *System.DirectoryServices.AccountManagement* - високорівнева бібліотека, яка обгортає System.DirectoryServices для зручності використання. Однак, не всі операції доступні через цей API [57].

За допомогою простору імен *System.DirectoryServices.AccountManagement* ми можемо дуже легко виконувати базові операції, такі як отримання властивостей об'єктів. Властивості розкриваються безпосередньо в класах і типізуються відповідним чином. Однак цей простір імен не надає всього, що нам потрібно. Він розкриває лише невелику кількість полів LDAP і не надає доступу до користувацьких класів та атрибутів. Для доступу до наших користувацьких атрибутів ми повинні використовувати низькорівневий простір імен System.DirectoryServices [58].

Простір імен System.DirectoryServices містить два складові класи для роботи з об'єктами AD DS:

- *DirectoryEntry* - інкапсулює вузол або об'єкт в ієрархії AD DS.
- *DirectorySearcher* - виконує запити до AD DS.

Властивості об'єктів AD DS зберігаються у масивах як загальні об'єкти, що може спричинити багато проблем, пов'язаних з великою кількістю ітерацій та використанням приведення. Отже, використання простору імен *System.DirectoryServices* для отримання ключів PGP користувачів було б дуже складним.

Однак, можна використовувати високорівневий API для отримання об'єкта *UserPrincipal*, а потім отримати доступ до базового об'єкта LDAP за допомогою методу *GetUnderlyingObject*. У прикладі показано, як ми використовуємо ці API для отримання приватного ключа поточного користувача (Рис. 2.7).

```
var currentUser = UserPrincipal.Current;

var userEntry = (DirectoryEntry)currentUser
                                .GetUnderlyingObject();

var key = userEntry.Properties["pgpPrivateKey"][0];

using (var inputStream = new MemoryStream(
                    Encoding.UTF8.GetBytes(key)))

using (var pgpStream = PgpUtilities
                    .GetDecoderStream(inputStream))
{
  var bundle = new PgpSecretKeyRingBundle(pgpStream);

  foreach (PgpSecretKeyRing secretKey :
                                bundle.GetKeyRings())
  {
      ...
  }
}
```

Рис. 2.7 Приклад використання API для отримання приватного ключа поточного користувача

2.5 Висновки до розділу 2

Другий розділ присвячено розробці моделі захисту онлайн повідомлень. В результаті проведеного дослідження:

- Отримала подальший розвиток розв'язання питань пов'язаних з автоматизацією шифрування та управління ключами.
- Отримала подальший розвиток реалізації надбудови для поштового клієнта Microsoft Outlook, адміністративний інструмент, що полегшує управління ключами та їх конфігурацію, а також скрипти для розширення схеми Active Directory.
- Отримала подальший розвиток реалізації програми, яка перебирає всіх користувачів в AD, перевіряє, чи не закінчується термін дії ключів користувачів, і генерує нові ключі, якщо це необхідно.

Розділ 3. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ДОСЛІДЖЕННЯ РОЗРОБЛЕНОЇ СИСТЕМИ

3.1 Програмне забезпечення та його ключові функції

Надбудова виконує автоматичне шифрування, дешифрування, підписання та перевірку цілісності. Вона також автоматично шукає відкриті ключі одержувачів на сервері ключів PGP і закриті ключі поточного користувача в Active Directory.

Надбудови для Microsoft Office можна створювати за допомогою розширення для інструментів розробки Microsoft Office в Microsoft Visual Studio. Інструмент генерує проект з класом *ThisAddIn*. Цей клас є точкою входу для надбудови і забезпечує доступ до об'єктної моделі Outlook.

Клас *ThisAddIn* містить два обробники подій, *Startup* і *ShutDown*. Метод *Startup* викликається, коли Outlook завантажує надбудову [59]. У цьому методі ми ініціалізуємо надбудову та реєструємо обробники подій Outlook. Метод *ShutDown* слугує для очищення ресурсів, що використовуються надбудовою.

Шифрування та підписання.Нам потрібно шифрувати повідомлення в наступних двох випадках:

- При відправленні листа - на рівні додатку на події ItemSent.
- При збереженні чернетки - на рівні елемента MailItem.Write.

Надсилання зашифрованого листа: Метод обробника події *ItemSent* спочатку перевіряє, чи елемент, що надсилається, є екземпляром класу MailItem, і якщо так, то метод переходить до шифрування.

Потім метод перевіряє наявність дійсних відкритих ключів усіх одержувачів у властивості Mailitem.Recipients. Якщо всі ключі доступні, повідомлення разом з вкладеннями підписується закритим ключем відправника і шифрується всіма відкритими ключами одержувачів. Крім того, воно

шифрується відкритим ключем відправника, щоб він міг прочитати надіслані повідомлення.

Якщо відкриті ключі деяких одержувачів не знайдено, метод перевіряє поштові домени всіх одержувачів. Якщо жодна з адрес одержувачів не належить до домену зі списку, що зберігається в атрибуті *encryptedDomains* в Active Directory, це означає, що лист надсилається поза зоною шифрування. Тому він надсилається відкритим текстом без підтвердження користувача. В іншому випадку, якщо якийсь із доменів одержувача належить до списку *encryptedDomains*, з'являється діалогове вікно з проханням підтвердити, що повідомлення може бути надіслано у відкритому вигляді.

Пошук ключів: Надбудова шукає приватний ключ відправника для підпису в об'єкті Active Directory User, що представляє поточного користувача. Надбудова використовує найновіший дійсний (не прострочений і не відкликаний) приватний ключ, що зберігається в атрибуті *pgpPrivateKeys*.

Відкриті ключі одержувачів шукаються на сервері ключів OpenPGP за їхніми ідентифікаторами користувачів. Ідентифікатор користувача відкритого ключа містить адресу електронної пошти користувача. Сервер ключів повертає набір відкритих ключів з відповідним ідентифікатором користувача. Потім надбудова обирає відкритий ключ, який не відкликано, термін дії якого не закінчився і який має дійсний сертифікат.

Дійсний сертифікат - це сертифікат, який підписано деякими дійсними (не відкликаними і не простроченими) головними особистими ключами, які мають відповідний відкритий ключ, що зберігається в атрибуті *trustedPublicMasterKeys* в Active Directory.

Якщо сервер повернув більше дійсних відкритих ключів, надбудова використовує для шифрування найновіший з них.

Інформування користувача про стан шифрування: Як вже згадувалось раніше, я додала до графічного інтерфейсу Outlook спеціальну область форми Outlook з текстовою міткою для інформування користувача про стан шифрування та перевірки підпису (Рис. 3.1).

При створенні нового листа в області форми відображається інформація про наявність ключів у одержувачів. При додаванні або видаленні одержувачів текстова мітка динамічно оновлюється. Це відбувається за допомогою методу обробника події *MailItem.PropertyChange*.

Якщо для всіх одержувачів доступні дійсні ключі, на формі відображається текст про те, що автоматичне шифрування увімкнено, на зеленому фоні. Якщо деяких ключів не вистачає, в області форми відображається попередження про те, що повідомлення не може бути зашифроване, з інформацією про те, яких саме ключів не вистачає. Фон області форми змінює колір на помаранчевий, щоб користувач точно помітив, що шифрування вимкнено.

Збереження зашифрованих чернеток: при збереженні чернеток надбудова автоматично шифрує їх відкритим ключем поточного користувача. Оскільки подія *MailItem.Write* є подією на рівні елемента, нам потрібно спочатку якось ідентифікувати цю подію на рівні програми. Об'єкт *MailItem* відображається у вікні, яке відповідає об'єкту *Inspector*. Коли відкривається новий інспектор, викликається подія *NewInspector* на рівні програми. Об'єкти, що відтворюються в інспекторах, можна обробляти за допомогою обгорток інспекторів.

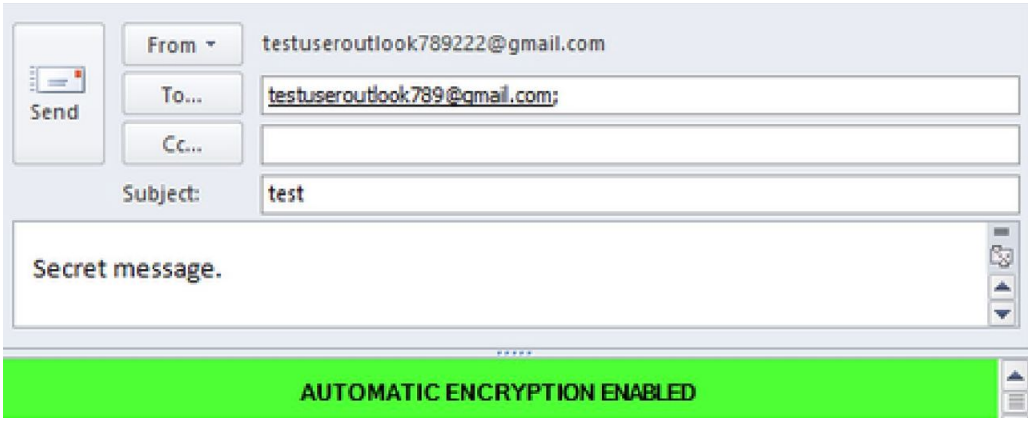

Рис. 3.1 Область форми Outlook, яка інформує користувача про стан шифрування.

Обгортки інспектора: У об'єктній моделі Outlook [60] вікно, у якому відображається MailItem та інші елементи Outlook, представляється об'єктом

Інспектора. Рекомендується писати обгортки для інспекторів Outlook за зразком, наданим Microsoft [47]. Завдяки обгорткам можна програмно ідентифікувати вікно, на якому клацнув користувач, коректно обробляти очищення пам'яті, незалежно обробляти об'єкти Інспектора для різних об'єктів Outlook і гарантувати, що жодні події не загубляться. Щоб написати обгортку для інспектора, потрібно створити базовий клас, який має властивість глобального унікального ідентифікатора (GUID), необхідного для ідентифікації кожного з екземплярів інспектора. Клас також повинен мати властивість *Inspector*, яка забезпечує доступ до екземпляру інспектора і необхідна для коректної обробки подій інспектора. Нарешті, необхідно зареєструвати кожну подію інспектора за допомогою відповідного методу об'єкта *Item*. У нашій реалізації ми написали обгортку інспектора для об'єкта *MailItem*.

Розпізнавання чернетки на події MailItem.Write: подія *MailItem.Write* спрацьовує у кількох випадках, і може статися так, що повідомлення вже зашифровано під час збереження листа. Щоб відрізнити чернетку від отриманих і відправлених повідомлень, надбудова перевіряє значення властивості *MailItem.Sent. Я*кщо ця властивість має значення *true*, доповнення не шифрує повідомлення оскільки воно або вже зашифроване, або було надіслане у відкритому вигляді і немає сенсу шифрувати його зараз. Однак, коли метод *Save* викликається в контексті надсилання нового листа, повідомлення вже зашифровано, але властивість *MailItem.Sent* все ще має значення *false*, тому ми не можемо відрізнити новостворений лист від чернетки за значенням цієї властивості.

Ми вирішили цю проблему, додавши до об'єкта *MailItem* кастомну властивість. Плагін зберігає в ній інформацію про те, що лист зашифровано під час шифрування. Після цього при збереженні повідомлення плагін завжди перевіряє, чи воно вже зашифроване, щоб не зашифрувати його двічі.

Розшифрування та перевірка підписів.Нам потрібно розшифровувати листи і перевіряти підписи в наступних ситуаціях.

- При відкритті листа в окремому вікні.

- При перегляді листа в області читання Outlook.

Розшифровка при відкритті листа в інспекторі: Для обробки подій об'єкта *MailItem* при відкритті його в інспекторі ми знову використовуємо обгортку інспектора, описану раніше. Лист відкривається в новому інспекторі за наступними подіями.

- *MailItem.Open*
- *MailItem.Reply*
- *MailItem.ReplyAll*
- *MailItem.Forward*

Отже, ми реєструємо ці чотири події через обгортку інспектора за допомогою методу обробника подій, який розшифровує лист і перевіряє його підпис. Метод обробника спочатку шукає відповідний приватний ключ для розшифрування в Active Directory, а потім розшифровує імейл. Ідентифікатор ключа використовується для пошуку приватного ключа, який відповідає публічному ключу, що використовується відправником для шифрування.

Перевірка підпису: Після розшифрування надбудова перевіряє, чи є підпис на повідомленні, і шукає відкритий ключ, що відповідає закритому ключу, який був використаний для підписання. Пошук відкритого ключа здійснюється за ідентифікатором закритого ключа на сервері ключів PGP. Потім надбудова перевіряє, чи не закінчився термін дії ключа, чи не відкликаний він, і чи не засвідчений він одним з дійсних приватних головних ключів, які мають відповідний відкритий ключ, що зберігається в атрибуті trustedPublicMasterKeys в Active Directory.

Якщо ключ підпису дійсний, підпис перевіряється, а інформація про дійсність підпису відображається в області користувацької форми Outlook, описаної раніше. В іншому випадку надбудова інформує користувача про те, що ключ підпису є недійсним і підпис не може бути перевірений.

Скасування змін в об'єктах MailItem: Якщо до повідомлення електронної пошти було внесено певні зміни, Outlook запитує користувача, чи хоче він їх зберегти, коли закриває інспектор з цим повідомленням. Щоб

уникнути збереження змін при закритті інспектора з розшифрованим листом, ми зареєстрували обробник події *MailItem.Close*. Обробник події відкидає зміни в об'єкті MailItem перед його збереженням і закриттям.

Розшифрування в області читання: До листа, відображеного в області читання, можна отримати доступ за допомогою об'єкта *Explorer*. *Explorer* являє собою вікно, в якому відображається вміст папки [61]. Вибір певного повідомлення в області читання відображається за допомогою події *Explorer.SelectionChange.* Поточний об'єкт *MailItem* можна отримати за допомогою властивості *Explorer.Selection*. Для коректної роботи з об'єктами *MailItem*, пов'язаними з *Explorer*, необхідно написати обгортку *Explorer*.

Однак подія *Explorer.SelectionChange* не надає доступу до попередньо вибраних об'єктів, тому ми не могли легко відкинути зміни до попередньо розшифрованого повідомлення під час переходу до іншого повідомлення у Провіднику. Можливим рішенням може бути обробка розшифрованих *MailItems* зі збереженням посилань на об'єкти та знищенням об'єктів перед їх збереженням при зміні виділення. Однак, якщо врахувати також розшифрування в інспекторах, то все стає набагато складніше. Наприклад, припустимо, що повідомлення вибрано в Панелі читання і одночасно відкрито в Інспекторі. Хоча повідомлення все ще відкрито в інспекторі, а виділення в області читання змінюється, зміни в об'єкті *MailItem* не слід відкидати, оскільки повідомлення буде переписано в зашифрованому вигляді в інспекторі.

Оскільки автоматичне розшифрування в області читання було б дуже складним для правильної обробки, ми не реалізували цю функцію в нашій перевірці концепції. Однак, наскільки нам відомо, існуючі надбудови Outlook, які зберігають електронні листи в зашифрованому вигляді, також не підтримують автоматичне розшифрування в області читання.

Інструмент адміністрування. Користувацький інтерфейс адміністративного інструменту та деталі реалізації управління ключами. Для реалізації простого графічного інтерфейсу ми використали платформу Windows Forms, яка є частиною Microsoft .NET Framework [62]. Графічний інтерфейс

програми складається з трьох вкладок, кожна з яких охоплює одну з таких функціональних можливостей: управління ключами користувачів, управління головними ключами підприємства та встановлення зв'язку з іншими підприємствами.

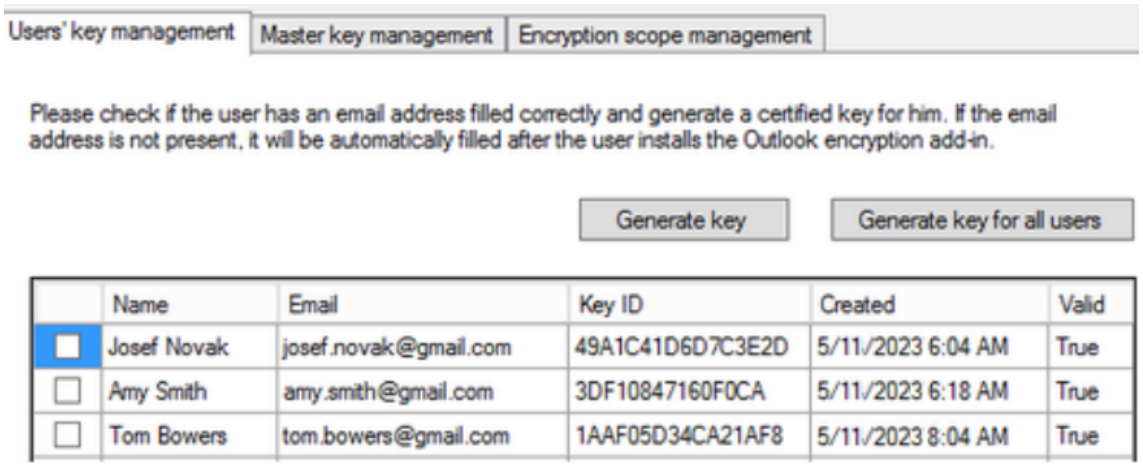

Рис. 3.2 Інтерфейс користувача інструменту адміністратора.

Керування ключами користувачів: Програма дозволяє адміністратору переглядати інформацію про користувачів та їхні ключі, генерувати нові ключі та відкликати вибрані ключі. Коли адміністратор натискає кнопку Згенерувати новий ключ, програма виконує наступні дії для обраного користувача:

- Генерує нову пару ключів, використовуючи адресу електронної пошти користувача як ідентифікатор пари ключів.
- Зберігає секретний набір ключів в Active Directory до атрибуту *pgpPrivateKeys* в об'єкті *User*, який представляє вибраного користувача.
- Сертифікує обидва відкриті ключі в наборі відкритих ключів (ключ підпису і ключ шифрування) дійсним головним приватним ключем з атрибуту *masterPrivateKeys* в об'єкті *PgpConfiguration*.
- Завантажує сертифікований набір відкритих ключів на сервер ключів PGP.

Під час генерації пари ключів PGP адреса електронної пошти користувача зчитується з атрибуту *User.EmailAddress* з Active Directory. Цей атрибут містить адресу електронної пошти користувача у разі використання Microsoft Exchange Server як поштового сервера. В іншому випадку атрибут може бути порожнім. При генерації пари ключів для одного користувача заповнення адреси вручну не

є проблемою. Однак при розгортанні рішення і генерації ключів для всіх співробітників це буде непрактично. Тому ми додали до надбудови Outlook функціонал, який перевіряє наявність адреси електронної пошти в AD і за потреби заповнює адресу, отриману з облікового запису Outlook. У цьому випадку адміністратор генерував би ключі після встановлення надбудови Outlook на комп'ютери кінцевих користувачів.

Оскільки зв'язка секретних ключів, що зберігається в AD, містить як відкриті, так і закриті ключі, відкликання ключів здійснюється наступним чином:

- Витягніть відкритий і закритий ключі з таємної зв'язки ключів, що зберігається в AD.
- Створіть сертифікат відкликання і додайте його до відкритого ключа.
- Виконайте попередні два кроки для обох відкритих ключів підпису і шифрування в зв'язці ключів.
- Створіть новий об'єкт зв'язки відкритих ключів, що містить відкликаний відкритий ключ підпису та відкликаний відкритий ключ шифрування.
- Завантажте відкликаний набір відкритих ключів на сервер ключів PGP.
- Замініть відкриті ключі в секретній зв'язці на відкликані відкриті ключі та оновіть значення секретної зв'язки ключів в AD.

Керування головним ключем: Під час генерації нового головного ключа програма виконує наступні кроки:

- Зберігає зв'язку секретних ключів в атрибуті *masterPrivateKeys* об'єкта *PgpConfiguration*.
- Зберігає зв'язку відкритих ключів у атрибуті *trustedPublicMasterKeys* об'єкта *PgpConfiguration*.
- Завантажує набір відкритих ключів на сервер ключів PGP.
- Засвідчіть ключі існуючих користувачів цим новим головним приватним ключем і завантажте оновлені ключі на сервер ключів PGP.

Коли головний ключ відкликається (так само, як описано для ключів користувачів), він оновлюється на сервері ключів, а також в AD в атрибутах *masterPrivateKeys* і *trustedPublicMasterKeys*.

З'єднання з іншим підприємством: Коли інструмент адміністрування запускається вперше, він генерує майстер-ключ з ідентифікатором користувача у вигляді *PGPmasterKey@<домен_пошти_підприємства>* і завантажує його на сервер ключів PGP. Таким чином, якщо підприємство-партнер вже розгорнуло рішення, ми можемо знайти майстер-ключ підприємства на сервері ключів. Додаток надає адміністратору можливість підключити автоматизовану систему шифрування, виконавши наступні кроки (Рис. 3.3).

Введіть цільовий домен електронної пошти підприємства та натисніть кнопку "Завантажити ключ". Після цього програма виконає пошук головного ключа підприємства на сервері ключів.

Якщо ключ знайдено, введіть відбиток завантаженого відкритого ключа. Якщо відбиток дійсний, програма зберігає відкритий ключ в атрибут *trustedPublicMasterKeys*, а також зберігає ім'я домену в атрибут *encryptedDomains*. Після того, як головний відкритий ключ і доменне ім'я зберігаються в AD, надбудова Outlook може перевірити сертифікати співробітників підключеного підприємства і починає автоматично шифрувати комунікацію з ними.

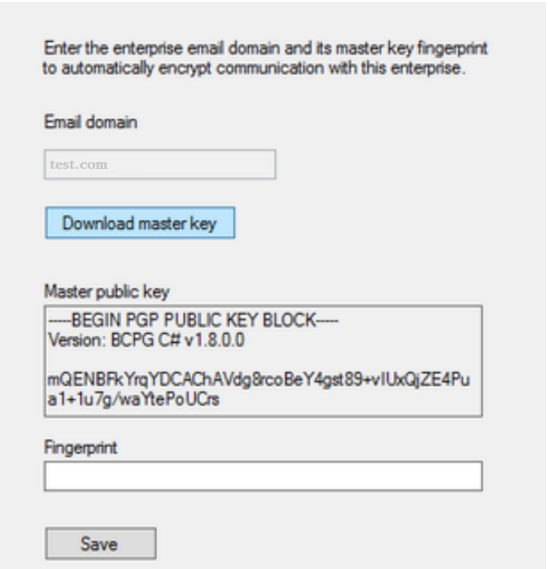

Рис. 3.3 Форма для встановлення зв'язку з іншим підприємством.

3.2 Експериментальне дослідження та аналіз результатів

Для розгортання нашого рішення в корпоративному середовищі на базі Windows з Active Directory необхідно виконати наступні кроки:

- Розширити схему AD та встановити належні права доступу для новостворених класів та атрибутів. Для розширення схеми користувач повинен бути членом групи адміністраторів схеми [63]. Розширення схеми включає наступні кроки:
    - Запуск підготовленого сценарію Powershell на контролері домену. Сценарій визначає поточне ім'я домену та генерує OID для нових атрибутів і класів. Потім він створює файл updateSchema.ldf, що містить LDIF-скрипт для розширення схеми.
    - Запустіть інструмент Ldifde з параметрами -i -f updateSchema.ldf. Ldifde - це інструмент командного рядка, який вбудовано в операційну систему Win- dows Server. [64].
    - Встановіть дозвіл control_access на конфіденційний атрибут pgpPrivateKeys для всіх користувачів за допомогою інструменту Dscals. Dscals - це інструмент командного рядка, який також доступний в операційній системі Windows Server [65].
- Встановіть інструмент адміністратора.
- Створіть заплановане завдання, яке запускатиме програму для перевірки терміну дії ключів у Планувальнику завдань Windows.
- Встановіть надбудову Outlook на комп'ютери користувачів.
- Запустіть інструмент адміністратора та згенерувати ключі для всіх користувачів.

Рішення було успішно розгорнуто в середовищі компанії та протестовано співробітниками. Тестування проводилося з Outlook версій 2013 та 2016.

Мені вдалося розгорнути рішення протягом 15 хвилин. Відгуки користувачів були в цілому позитивними. Вони оцінили користувацький інтерфейс як простий у використанні та зрозумілий. Внутрішня електронна

пошта була автоматично зашифрована одразу після розгортання без будь-яких додаткових зусиль з боку кінцевих користувачів. Користувачі не мали жодних проблем з розпізнаванням того, чи зашифрований електронний лист, який вони надсилають, чи ні.

Ми також протестували автоматичне оновлення ключів, встановивши короткий термін дії згенерованих ключів. Ключі були оновлені без жодних проблем з подальшою перевіркою сертифікатів, шифруванням та дешифруванням.

Оцінка та порівняння.

Розгортання: На відміну від веб-сервісів шифрування, розгортання нашого рішення не передбачає зміну провайдера електронної пошти або створення нових облікових записів.

Розгортання клієнтських рішень на підприємстві вимагало б ручної генерації, розповсюдження та перевірки ключів кожним користувачем, перш ніж вони зможуть почати надсилати зашифровані електронні листи один одному.

У нашому рішенні адміністратор генерує, сертифікує та розповсюджує ключі для всіх користувачів одним натисканням кнопки. Таким чином, все внутрішнє електронне листування автоматично шифрується.

Зручність використання: Веб-сервіси шифрування не викликають жодних проблем у користуванні, оскільки шифрування та управління ключами здійснюються самим сервісом і для шифрування електронної пошти не потрібна додаткова взаємодія з користувачем.

Більшість клієнтських рішень надають можливість шифрувати електронну пошту автоматично. Однак користувачі повинні запам'ятовувати паролі для закритих ключів і копіювати їх вручну на всіх пристроях, де вони хочуть використовувати електронну пошту.

Наше рішення використовує Active Directory для безпечного зберігання та автоматичної передачі приватних ключів на різні пристрої. Завдяки цьому ми позбулися паролів для закритих ключів. Крім того, надбудова для Outlook не

вимагає жодної взаємодії з користувачем для шифрування чи дешифрування електронних листів, а управління ключами повністю автоматизовано.

Обслуговування: У той час, як веб-сервіси не потребують подальшого обслуговування після розгортання, клієнтські рішення страждають від проблеми ручного управління ключами. Коли термін дії ключів закінчується, користувачам доводиться генерувати нові ключі та розповсюджувати їх знову.

Наше рішення забезпечує автоматичну перевірку терміну дії ключів, а також автоматично генерує та розповсюджує нові ключі. Таким чином, після розгортання нема потреби в подальшому обслуговуванні.

Безпека: Жодне з існуючих рішень не має достатньо безпечного управління ключами. Сервіси шифрування вимагають довіри для роботи з ключами користувача. Ручне керування ключами, що надається клієнтськими рішеннями, не є достатньо безпечним, оскільки користувачі можуть неправильно поводитися з ключами та сертифікатами, що підтверджується кількома дослідженнями щодо зручності використання PGP [17-19].

Це рішення забезпечує автоматизоване управління ключами без залучення будь-якої довіреної третьої сторони. Таким чином, наше рішення забезпечує кращу конфіденційність, ніж сервіси шифрування, а також є більш безпечним, ніж клієнтські рішення, оскільки запобігає неправомірному використанню ключів користувачами.

Що стосується шифрування електронної пошти, то веб-сервіси автоматично шифрують все, крім листів, що надсилаються зовнішнім одержувачам, які не користуються тим самим сервісом. Рішення на стороні клієнта здебільшого дозволяють користувачам ненавмисно надсилати електронні листи відкритим текстом.

Це рішення автоматично шифрує всі внутрішні електронні листи та листи, що надсилаються підключеним підприємствам. Воно також сповіщає користувачів, коли повідомлення неможливо зашифрувати, щоб вони не надсилали його ненавмисно відкритим текстом. Крім того, воно автоматично шифрує чернетки, щоб вони не потрапляли на поштовий сервер.

3.3 Висновки до розділу 3

Кожне з існуючих рішень для наскрізного шифрування має певні переваги, але також і багато недоліків. Жодне з них не забезпечує всю необхідну функціональність, ні з достатнім рівнем безпеки, ні з достатнім рівнем простоти використання. Основними перевагами мого рішення є повністю автоматизована перевірка та оновлення сертифікатів, просте розгортання в поштовій інфраструктурі підприємства та безперешкодне підключення системи шифрування між кількома підприємствами. Наскільки відомо, наразі не існує рішення, яке б забезпечувало таку ж функціональність, як моє рішення, і було б таким же простим у використанні, не залучаючи при цьому жодної довіреної третьої сторони.

Розділ 4. ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА

Атмосфера, тонкий шар газів, що оточує нашу планету, відіграє ключову роль у забезпеченні умов для існування життя. Проте, сучасний технологічний прогрес і розширення господарської діяльності людства призводять до надмірного викидання різноманітних речовин у атмосферу, створюючи загрозу для її екологічної стійкості. Екологічна безпека атмосфери, у вигляді збереження її чистоти та стійкості, стає актуальним завданням нашого часу.

В цьому розділі детально розглянемо виклики, які стоять перед атмосферою внаслідок антропогенного впливу, вивчимо наслідки цих викликів для клімату та здоров'я людини, і визначимо ефективні заходи та стратегії для забезпечення екологічної безпеки атмосферного середовища. Поглиблене розуміння цих питань є важливим етапом у впровадженні конкретних заходів для збереження атмосферної якості і забезпечення сталого розвитку для нашої планети.

Забруднення атмосфери. *Промислові та транспортні викиди.* Промисловість та транспортна система, які є основними двигунами сучасного господарства, відіграють ключову роль у формуванні атмосферного забруднення. Промислові викиди складаються з різноманітних газів, часток та інших забруднюючих речовин, що виходять з підприємств у процесі виробництва. Транспорт забезпечує величезні обсяги викидів через використання пального та інших енергетичних ресурсів. Це створює великі труднощі для атмосферної якості в населених пунктах та прилеглих територіях.

Парникові гази та їх вплив на клімат. Парникові гази, такі як вуглекислий газ (CO2), метан (CH4), оксид азоту (NOx) та інші, грають важливу роль у ретенції тепла в атмосфері. Хоча цей ефект необхідний для підтримання теплового балансу на Землі, надмірна концентрація парникових газів призводить до посилення теплового ефекту та змін клімату. Збільшення середньої температури, танення льодовиків та екстремальні погодні умови – це

лише кілька наслідків змін клімату, які можуть бути викликані викидами парникових газів.

Токсичні речовини та їх вплив на здоров'я людини. На додаток до загального атмосферного забруднення, промисловість та інші джерела викидають токсичні речовини, такі як бензопірен, свинець, сірководень та інші. Ці речовини можуть мати серйозний вплив на здоров'я людини, спричиняючи захворювання дихальних шляхів, серця, нирок та інших систем організму. Крім того, деякі токсичні речовини можуть бути канцерогенами, що ставить під загрозу загальну громадську безпеку.

Розглядаючи ці аспекти, стає очевидним, що раціональне управління та зменшення забруднення атмосфери стає важливим завданням для збереження здоров'я населення та екосистем, що є обов'язковим етапом у досягненні екологічної безпеки атмосфери.

Кліматичні зміни та їх вплив. *Зміни температури та екстремальні погодні умови.* Сучасні зміни клімату вже призвели до помітного підвищення середньорічної температури нашої планети. Збільшення температури впливає на різноманітні аспекти природного середовища, включаючи екосистеми, рослинність та фауну. Екстремальні погодні умови, такі як зливи, засухи, урагани та торнадо, стають більш поширеними та інтенсивнішими, що призводить до серйозних наслідків для сільського господарства, водних ресурсів та життєдіяльності людей.

Підйом рівня моря та його наслідки. Один із найбільш очевидних впливів кліматичних змін – це підйом рівня моря. Танення льодовиків та shelf-льоду призводить до виливання великих об'ємів води у море, що може викликати затоплення прибережних територій. Це має серйозні наслідки для міського планування, економічної інфраструктури та біорізноманіття.

Зв'язок між забрудненням атмосфери та кліматичними змінами. Забруднення атмосфери має безпосередній вплив на клімат через викиди парникових газів та інших забруднюючих речовин. Збільшення концентрації парникових газів, таких як CO2, призводить до збільшення теплового ефекту та

підвищення середньорічної температури. Це, у свою чергу, призводить до змін у кліматичних зонах, розподілу опадів та інших показників клімату. Такий взаємозв'язок стає причиною глобальних змін, які впливають на всі аспекти природного середовища та життя на Землі.

Розглядаючи ці впливи, важливо визнати необхідність розвитку сталого господарювання та екологічної безпеки для мінімізації негативних наслідків кліматичних змін. Ефективна стратегія має включати заходи з обмеження забруднення атмосфери, використання відновлюваних джерел енергії та глобальне співробітництво в розв'язанні цих проблем.

Заходи для зменшення забруднення. *Енергоефективність та використання відновлюваних джерел енергії.* Один із ключових напрямків для зменшення забруднення атмосфери – це перехід до енергоефективних технологій та використання відновлюваних джерел енергії. Впровадження ефективних технологій у будівництві, транспорті та виробництві дозволяє зменшити витрати енергії та викиди шкідливих речовин. Розробка та використання відновлюваних джерел енергії, таких як сонячна та вітрова енергія, сприяє зменшенню залежності від вугільних джерел, що є основним джерелом викидів парникових газів.

Технологічні інновації в промисловості. Впровадження технологічних інновацій у промисловості грає важливу роль у зменшенні викидів та покращенні екологічної безпеки. Розробка та застосування чистих виробничих технологій, які мінімізують викиди та використовують вторинні ресурси, стає ключовим завданням. Використання ефективних систем очищення газів та рідких відходів сприяє зменшенню токсичних викидів у повітря та воду.

Управління викидами та ефективна система контролю. Системи управління викидами грають важливу роль у забезпеченні контролю та моніторингу рівнів забруднення атмосфери. Ефективність контрольних механізмів та їх вчасна реакція на викиди дозволяє оперативно зменшити негативний вплив на довкілля. Розробка та впровадження новітніх систем

моніторингу, включаючи сучасні сенсори та великі дані (big data), сприяє створенню ефективної системи контролю за забрудненням.

Впровадження цих заходів відіграє критичну роль у досягненні екологічної безпеки атмосфери. Вони спрямовані на покращення якості повітря, зменшення викидів та прискорення переходу до сталого, екологічно чистого виробництва та споживання ресурсів.

Загальний погляд на екологічну безпеку атмосфери вказує на її критичність для збереження життя на Землі. Із зростанням господарської діяльності та розвитком технологій, атмосфера стає сценою значущих забруднень, що мають вплив на клімат та здоров'я людей. Освідомлення цих проблем та їх розв'язання визначають напрямок подальших досліджень та заходів для забезпечення сталої та екологічно безпечної майбутньої атмосфери.

Промислові та транспортні викиди, парникові гази та токсичні речовини впливають на всі аспекти природного середовища та здоров'я. Необхідно вжити ефективних заходів для обмеження цих викидів та переходу до більш чистих технологій, щоб забезпечити екологічну безпеку атмосфери.

Кліматичні зміни представляють велике викликання для нашого світу. Зміни температури, екстремальні погодні умови та підйом рівня моря – це лише частина проблеми, яка потребує комплексних рішень. Забруднення атмосфери та зміни клімату тісно пов'язані, і їх вирішення вимагає глобальної координації та впровадження екологічно ефективних заходів.

Важливим етапом у розв'язанні екологічних проблем є заходи для зменшення забруднення атмосфери. Використання енергоефективних технологій, технологічні інновації в промисловості та системи управління викидами є ключовими напрямками. Ці заходи допомагають знизити вплив антропогенного фактора на атмосферу та сприяють переходу до більш сталої та екологічно безпечної економіки.

ВИСНОВКИ

Результатом виконаної роботи є вирішення задачі розробки моделі захисту онлайн-повідомлень.

У процесі цієї роботи отримані наступні результати:

1. Проаналізовані існуючі методи та засоби захисту онлайн-повідомлень, та виявлено, що шифрування виділяється як найефективніший метод для забезпечення достатнього рівня безпеки.
2. Розроблена модель захисту онлайн-повідомлень, яка автоматизовує виконання більшості завдань, включаючи, але не обмежуючись, обов'язкові втручання користувача лише у випадках, коли це необхідно.
3. Реалізовано програмне забезпечення, яке виконує генерування та зберігання нових ключів для кінцевих користувачів у захищеному центральному сховищі всередині підприємства; відкликає ключі, коли вони стають скомпрометованими; встановлює зв'язок з іншими підприємствами.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ебінезер М і Суреш Б. 2015 Стратегії безпеки для соціальних мереж в Інтернеті Міжнародний журнал комп'ютерних тенденцій і технологій (IJCTT) 5(5)
2. Абдулла С. М. 2017 Підхід до спільного використання декількох секретів для виявлення вразливостей у соціальних мережах 1-а Міжнародна конференція з інформаційних технологій (ICoIT'17) с.45
3. Сушанка Т. 2017 Аналіз безпеки месенджера Telegram Магістерська дисертація, Чеський технічний університет.
4. Чжан З. і Гупт Б. Б. 2016 Безпека та надійність соціальних мереж: Огляд і новий напрямок Future Generation Computer Systems 86 с.914-925.
5. Кутілло Л. А. 2007 Збереження конфіденційності в соціальних мережах на основі довіри в реальному житті IEEE Communications Magazine 47(12) с.94-101
6. Бейе М., Джекманс А. Дж., Еркін З., Хартель П., Лагендейк Р. Л. і Танг К. 2012 Конфіденційність в онлайн-соціальних мережах. В Обчислювальні соціальні мережі, с. 87-113
7. Бошруй С. Т., Купц А. та Озкасап О. 2015 Безпека та конфіденційність розподілених онлайн-соціальних мереж у Міжнародній конференції з розподілених обчислювальних систем, с.112-115.
8. Абрахам А 2012 Безпека та конфіденційність обчислювальних соціальних мереж у Springer Science & Business Media ISBN 978-1-4471-4051-1
9. Вільям Сталлінгс, "Криптографія та мережева безпека, видання 5.
10. Пан Винод Сароха, Анну Малік, Мадху Пахал, "Величезний сертифікат: Сертифікат цифрового підпису"; Міжнародний Журнал передових досліджень в галузі комп'ютерних наук та програмної інженерії, том 3, випуск 6, червень 2013 р. ISSN: 2277 128X

11. J. Callas, L. Donnerhacke, H. Finney, R. Thayer, "OpenPGP Message Format", RFC2440, November1998,
12. H.L.Kesterson II, "Digital Signatures - Whom Do You Trust?", IEEE Electronic Database 0-7803-3741-7/97.
13. Р. Перлман, "Огляд моделей довіри PKI", IEEE Network, листопад/грудень 1999 р.
14. Cisco Systems, "Introduction to Secure Sockets Layer", Біла книга з Інтернету.
15. D.W. Chadwick, A. J. Young, N. Kapidzic Cicovic, "Merging and IEEE Extending the PGP and PEM Trust Models - The ICE-TEL Trust Model", network, May/June 1997.
16. H.L.Kesterson II, "Цифрові підписи - кому ви довіряєте?", Електронна база даних IEEE 0-7803-3741-7/97.
17. D.W. Chadwick, A. J. Young, N. Kapidzic Cicovic, "Merging and Extending the PGP and PEM Trust Models - The ICETEL Trust Model", IEEE network, May/June 1997.
18. Рамсделл Б. та Тернер С. Безпечний/багатоцільовий обмін повідомленнями електронної пошти в Інтернеті (S/MIME), версія 3.2 Специфікація повідомлень. 2010. url: https://tools.ietf.org/html/rfc5751
19. Каллас Дж. та ін. Формат повідомлень OpenPGP. 2007. url: https:// tools.ietf.org/html/rfc5751
20. Циммерманн П. Чому я написав PGP. 1999. url: http : / / www . philzimmermann.com/EN/essays/WhyIWrotePGP.html
21. GnuPG. 2017. url: https://www.gnupg.org.
22. Левісон Л. та ін. Альянс темної пошти. 2015. URL: https://darkmail. info/downloads/dark- internet- mail- environment- march- 2015.pdf
23. Менезес Альфред Ж., Ванстоун Скотт А., Ван Ооршот Пол К. Довідник з прикладної криптографії. 1-е видання. CRC Press, Inc., 1996. isbn: 0849385237.

24. Giegerich and Partner GmbH. Посібник з Gpg4O. 2017. URL: http:// download.giepa.de/gpg4o/latest/release/Manual_gpg4o_ EN
25. Компендіум Gpg4win. 2017. url: https :// www . gpg4win . org/doc/en/gpg4win-compendium.html
26. Раффо Даніеле та ін. Безпека електронної пошти OpenPGP для додатків mozilla. Посібник. 2017. url: https :// www . enigmail . net / documentation/Enigmail_Handbook_1.8_en.pdf
27. Вендландт Д., Андерсен Д.Г., Перріг А. Покращення аутентифікації хоста в стилі SSH за допомогою багатошляхового зондування.
28. Mailpile. 2017. url: https:/ /www.mailpile.is/
29. Symantec Encryption Desktop для Windows. Керівництво користувача. 2017.
30. Microsoft.S/MIME для підписання та шифрування повідомлень.2017.URL: https://technet.microsoft.com/en-us/library/dn626158(v= exchg.150).aspx .
31. Whitten A. and Tygar J. D. "Why Johnny Can't Encrypt: Оцінка зручності використання PGP 5.0". In: У матеріалах 8-го симпозіуму з безпеки USENIX. 1999.
32. Шенг С. та ін. "Чому Джонні досі не може шифруватися: Оцінка зручності використання програмного забезпечення для шифрування електронної пошти". In: У матеріалах другого симпозіуму по зручній конфіденційності та безпеці. 2006.
33. Руоті С. та ін. "Чому Джонні досі не може зашифрувати: Оцінка зручності використання сучасного клієнта PGP". B: (2016). arXiv: 1510. 08555v2.
34. Microsoft. Outlook MAPI Reference. 2017. URL: https :// msdn . microsoft.com/en- us/library/office/cc765775.aspx.
35. Корпорація Майкрософт. Огляд служб сертифікації Active Directory. 2017. url: https://technet.microsoft.com/en-us/library/hh831740(v= ws.11).aspx .
36. Microsoft. Огляд доменних служб Active Directory. 2017. URL: https://technet.microsoft.com/en-us/library/hh831484(v= ws.11).aspx.

37. Microsoft. Схема Active Directory. 2017. url: https : / / msdn . microsoft . com / en - us / library / ms675085(v = vs . 85) .aspx.
38. Microsoft. Полегшений протокол доступу до каталогів. 2017. url: https://msdn.microsoft.com/en- us/library/aa367008(v=vs.85).aspx
39. Microsoft. Детальні концепції: Пояснення безпечного каналу. 2015. url: https :// social . technet . microsoft . com / wiki /contents / articles/24644.detailed-concepts-secure-channel-explained. aspx
40. Майкрософт. Захист даних у Windows. 2017. url: https :// msdn . microsoft.com/en-us/library/ms995355.aspx#windataprotection- dpapi_topic03
41. Microsoft. Використання біту конфіденційності для приховування даних в активному каталозі. 2017. URL: http://windowsitpro.com/active-directory/ використання- біту- конфіденційності- приховування- даних- в- активному- каталозі.
42. Розширення схеми. 2017. url: https : / / msdn . microsoft . com / en - us / library / ms676900(v = vs . 85) .aspx.
43. Огляд об'єктної моделі Outlook. 2017. url: https://msdn.microsoft. com/en-us/library/ms268893.aspx.
44. Microsoft. Об'єкт UserProperties Object. 2017. URL: https://msdn.microsoft. com/en-us/library/office/ff862196.aspx.
45. Microsoft. Налаштування користувацького інтерфейсу Office. 2017. url: https : / / msdn . microsoft . com / en - us / library / bf08984t . aspx
46. Microsoft. Створення регіонів форм Outlook. 2017. URL: https:// msdn.microsoft.com/en-us/library/bb386301.aspx.
47. Microsoft. Як розширити схему. 2017. url: https://msdn. microsoft . com / en - us / library / ms676929(v = vs . 85) .aspx.
48. Microsoft. Сценарії LDIF. 2017. url: https://msdn.microsoft. com / en - us / library / ms677268(v = vs . 85) .aspx.
49. Microsoft. Програмне розширення. 2017. url: https : / / msdn . microsoft . com / en - us / library / ms677631(v = vs . 85) .aspx.

50. Microsoft. Генерування ідентифікатора об'єкта з Powershell. 2017. url: https :// gallery . technet . microsoft . com / scriptcenter / Generate-an-Object-4c9be66a .
51. Microsoft. Міркування щодо проектування схеми Active Directory та допоміжні класи. 2017. URL: https://blogs.msdn.microsoft.com/alextch/ 2007/05/16/active-directory-schema-design-considerations-and-auxiliary-class es///technet.microsoft.com/en- us/library/cc961575.aspx
52. Microsoft. Dsacls. 2017. url: https:// technet.microsoft.com/en-us/library/ cc771151(v = ws . 11) .aspx
53. Microsoft. Використання сценаріїв для керування безпекою Active Directory. 2017. url: https : / / technet . microsoft . com / en - us / library / ff406131.aspx
54. Пакет org.bouncycastle.openpgp. 2017. url: http://www.bouncycastle. org/docs/pgdocs1.5on/index.html.
55. Шоу Д. Протокол OpenPGP HTTP Keyserver Protocol (HKP). 2017. url: https://tools.ietf.org/html/draft-shaw-openpgp-hkp-00
56. Microsoft. System.DirectoryServices.AccountManagement Names- pace. 2017. url: https://msdn.microsoft.com/en-us/library/ system . directoryservices . accountmanagement(v = vs . 110).aspx
57. Microsoft. Простір імен System.DirectoryServices. 2017. url: https://msdn.microsoft.com/en-us/library/system.directoryservices(v= vs.110).aspx
58. Microsoft. Покрокова інструкція: Створення вашого першого VSTO-надбудови для зовнішнього вигляду. 2017. url: https://msdn.microsoft.com/en-us/library/ cc668191.aspx
59. Microsoft. Огляд об'єктної моделі Outlook. 2017. URL: https:// msdn.microsoft.com/en-us/library/ms268893.aspx
60. Microsoft. Розробка інспекторської оболонки для Outloo. 2017. url: https : / / msdn . microsoft . com / en - us / library / office / ff973716(v=office.14).aspx (дата звернення: 30.04.2017).

61. Microsoft. Об'єкт провідника. 2017. url: https://msdn.microsoft. com/en-us/library/office/ff860356.aspx
62. Microsoft. Windows Forms. 2017. url: https://msdn.microsoft. com /en - us / library/ dd30h2yb(v= vs .110) .aspx
63. Microsoft. Передумови для встановлення розширення Schema. 2017. url: https://msdn.microsoft.com/en-us/library/ms677628(v= vs.85).aspx
64. Microsoft. Методи розширення схеми. 2017. url: https ://technet.microsoft.com/en- us/library/cc961584.aspx
65. Microsoft. Мобільні додатки Office отримають додаткові можливості у 2015 році. 2017. url: https://dev.office.com/docs/add-ins/outlook/ outlook-mobile-addins

Додаток А

Слайди презентації

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ КІБЕРБЕЗПЕКИ ТА ПРОГРАМНОЇ ІНЖЕНЕРІЇ
КАФЕДРА КОМП'ЮТЕРИЗОВАНИХ СИСТЕМ ЗАХИСТУ ІНФОРМАЦІЇ

КВАЛІФІКАЦІЙНА РОБОТА
на тему
«Модель захисту онлайн-повідомлень»

Виконав: Кучеренко Ю. І.
Керівник: Терейковська Л.О.

КИЇВ - 2023

АКТУАЛЬНІСТЬ

- Онлайн-повідомлення стали невід'ємною частиною нашого життя. Однак, онлайн-повідомлення також вразливі до атак з боку кіберзлочинців. Модель захисту онлайн-повідомлень - це основа для захисту онлайн-повідомлень від кібератак. Вона може включати різні функції. Актуальність моделі захисту онлайн-повідомлень полягає в тому, що вона може допомогти захистити конфіденційність, цілісність і доступність.

2

Мета дипломної роботи є реалізація ефективної моделі захисту онлайн-повідомлень.

Об’єкт дослідження: процеси захисту онлайн-повідомлень.

Предмет дослідження: модель захисту онлайн-повідомлень

Методи дослідження: базуються на нечіткій логіці для аналізу та оцінки ризиків у онлайн-комунікаціях, і на об'єктноорієнтованому програмуванні для розробки програмної реалізації розробленої моделі захисту повідомлень.

3

Новизна одержаних результатів: Розробка та впровадження моделі захисту онлайн-повідомлень, яка враховує сучасні виклики та загрози у сфері кібербезпеки.

Практична цінність:

Розроблена модель захисту онлайн-повідомлень, базується на PGP, забезпечує ефективну конфіденційність для користувачів онлайн-комунікацій та організацій, що використовують ці технології. Застосування цієї технології гарантує безпеку важливої інформації, забезпечуючи надійний захист для користувачів і підприємств онлайн-повідомлень.

4

ЗАДАЧІ

- Аналіз існуючих методів та засобів захисту онлайн-повідомлень.
- Розробка моделі захисту онлайн-повідомлень.
- Реалізація програмного забезпечення на основі розробленої моделі.

5

PGP (Pretty Good Privacy)

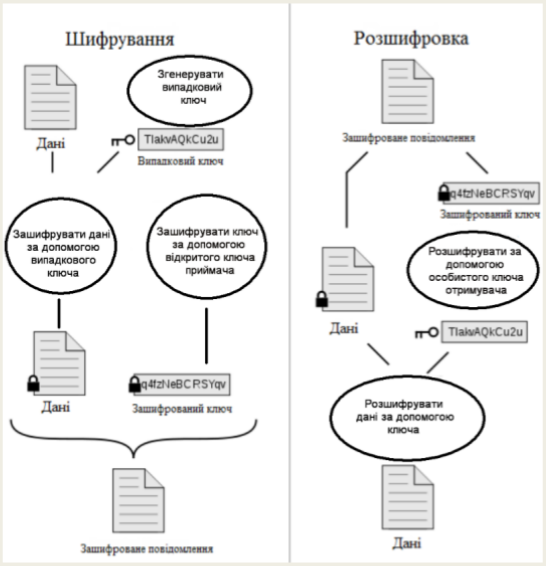

Рис. 1: Схема PGP шифрування

Генерація ключів

Вибираються два великі прості числа p, q

Обчислюється їх добуток $n=pq$

Обчислюється функція Ейлера $\phi(n) = (p-1)(q-1)$

Вибирається ціле число e $\gcd(\phi(n), e) = 1;\ 1 < e < \phi(n)$

Обчислюється d $d \equiv e^{-1} \bmod \phi(n)$

Відкритий ключ $KU = \{e, n\}$

Секретний ключ $KR = \{d, n\}$

Шифрування

Відкритий текст $M < n$

Зашифрований текст $C = M^e \pmod n$

Розшифрування

Зашифрований текст C

Відкритий текст $M = C^d \pmod n$

Рис. 2: Алгоритм RSA

6

Розробка рішення захисту

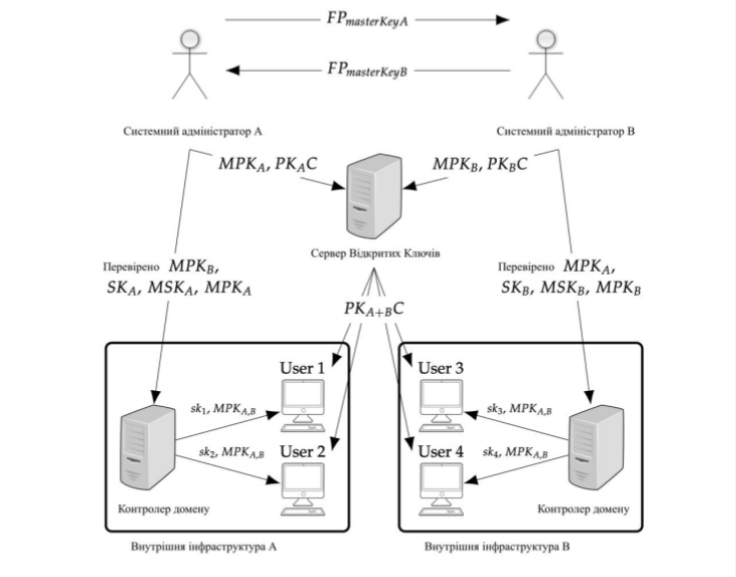

Рис. 3: Архітектура управління ключами

7

Інструмент адміністрування

- Керування ключами користувачів
- Керування головним ключем
- З'єднання з іншим підприємством

Надбудова для Outlook

- Шифрування та підписання
- Розшифрування та перевірка підпису

8

Експериментальні дані

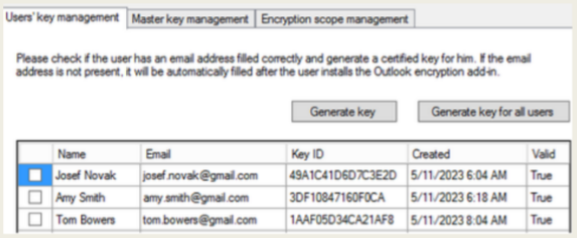

Рис. 4: Інтерфейс користувача інструменту адміністратора.

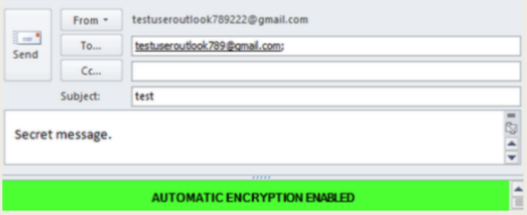

Рис. 6: Область форми Outlook, яка інформує користувача про стан шифрування.

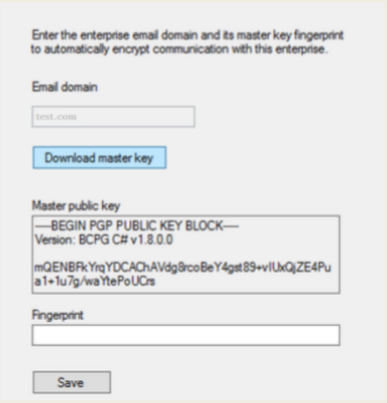

Рис. 5: Форма для встановлення зв'язку з іншим підприємством.

9

Апробація

Кучеренко Ю. І. Аналіз сучасних досліджень у сфері захисту онлайн-повідомлень// Живучість та резильєнтність – 2023: міжнародна науково-практична конференція 19 жовтня 2023 р.: тези доповіді. – К., 2023. – С.140-141.

10

Продовження додатку А

ВИСНОВКИ

- Проаналізовані існуючі методи та засоби захисту онлайн-повідомлень, та виявлено, що шифрування виділяється як найефективніший метод для забезпечення достатнього рівня безпеки.
- Розроблена модель захисту онлайн-повідомлень, яка автоматизовує виконання більшості завдань, включаючи, але не обмежуючись, обов'язкові втручання користувача лише у випадках, коли це необхідно.
- Реалізовано програмне забезпечення, яке виконує генерування та зберігання нових ключів для кінцевих користувачів у захищеному центральному сховищі всередині підприємства; відкликає ключі, коли вони стають скомпрометованими; встановлює зв'язок з іншими підприємствами.

11

ДЯКУЮ ЗА УВАГУ!

12