УДК 004

І.Науменко, магістр гр. КІ-21М-1,4,
Центральноукраїнський національний технічний університет

ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ПОВІДОМЛЕНЬ ЕЛЕКТРОННОЇ ПОШТИ

У статті розроблено програмне забезпечення, яке призначено для системи повідомлень електронної пошти. Метою розробки є дослідження та програмна реалізація системи повідомлень електронної пошти. Об'єктом дослідження є процес повідомлень електронної пошти. Предметом дослідження є методи повідомлень електронної пошти. Методи дослідження базуються на методах теорії комп'ютерних мереж, методах математичної статистики, методах розробки програмного забезпечення. Результат роботи – програмна реалізація системи повідомлень електронної пошти. В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

комп'ютерна інженерія, електронна пошта

Постановка проблеми. Адміністратори багатьох організацій намагаються захистити повідомлення електронної пошти співробітників. Secure MIME (S/MIME) – рішення безпеки, реалізоване в більшості сучасних поштових програм, що допоможе зберегти конфіденційність, цілісність поштових повідомлень і перевірити дійсність даних. S/MIME забезпечує наскрізний захист – не тільки в процесі пересилання повідомлень, але й при зберіганні в базі даних поштового сервера. S/MIME забезпечує перевірку дійсності даних, конфіденційність і цілісність повідомлень у форматі MIME. S/MIME – відмінний приклад гібридного рішення шифрування, у якому об'єднані достоїнства симетричного й асиметричного шифрів і функцій гешування. Якщо TLS забезпечує безпеку даних у момент пересилання по незахищеній мережі, такій як Інтернет, то S/MIME забезпечує безпеку даних між кінцевими користувачами: S/MIME повідомлення шифрується відправником (з використанням багатобічного шифрування) і передається на сервер відправника в зашифрованій формі. Та ж зашифрована форма використовується, коли повідомлення передається через мережу, коли воно зберігатися на проміжних серверах, і коли воно міститься в папках одержувачів. Тільки одержувачі, використовуючи свої закриті ключі, можуть розшифрувати повідомлення й в той момент, коли вони фактично читають повідомлення: саме повідомлення залишається зашифрованим у папках одержувачів. Для того, що б кінцеві користувачі могли використовувати S/MIME безпеку, всі вони повинні мати своїми власними PKI-ключі. Кожний користувач повинен мати закритий ключ, що безпечно зберігається в місці, доступному тільки для цього користувача, і відповідний йому відкритий ключ, убудований у сертифікат. Цей Сертифікат повинен бути виданий центром сертифікації (удостовірюючим центром), якому довіряють інші користувачі. S/MIME забезпечує потужні розширення функцій поштової безпеки, які можуть бути дуже корисними користувачам.

Аналіз останніх досліджень і публікацій. При аналізі останніх досліджень і публікацій [1-10] було виявлено певні прогалини у забезпеченні системи повідомлень електронної пошти.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи повідомлень електронної пошти.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

– Огляд існуючих систем повідомлень електронної пошти.

– Дослідження системи повідомлень електронної пошти.
– Програмна реалізація системи повідомлень електронної пошти.

Об'єктом дослідження є процес повідомлень електронної пошти.

Предметом дослідження є методи повідомлень електронної пошти.

Методи дослідження базуються на методах теорії комп'ютерних мереж, методах математичної статистики, методах розробки програмного забезпечення.

Виклад основного матеріалу.

Опис S/MIME на основі PKI

Специфікація S/MIME визначає два типи MIME-конверта: один для цифрових підписів, іншої – для шифрованих повідомлень. Обидва типи базуються на синтаксисі криптографічних повідомлень стандарту PKCS#7 [6]. Якщо повідомлення повинне бути зашифроване, а шифртексту повинні бути привласнені деякі атрибути, використовуються вкладені конверти. Зовнішній і внутрішній конверт призначаються для захисту цифрових підписів, а проміжний конверт – для захисту шифртексту.

Крім забезпечення цілісності повідомлення під час передачі, S/MIME ідентифікує власника конкретного відкритого ключа за допомогою сертифіката X.509. Цифровий сертифікат засвідчує, що відкритий ключ дійсно належить тому, хто є суб'єктом сертифіката.

S/MIME v3 підтримує наступні важливі властивості, які відсутні в S/MIME v2:

–завірені цифровим підписом квитанції;

–мітки безпеки;

–списки розсилання;

–гнучке керування ключами.

Завірені цифровим підписом **квитанції** дозволяють відправникові повідомлення впевнитися в тому, що воно було отримано адресатами без змін. Одержувач повідомлення не може згенерувати валідну квитанцію доти, поки не перевірить підпис відправника на отриманому повідомленні.

Мітки безпеки дозволяють відправникові задавати керуючі вимоги до змісту повідомлення. Найчастіше мітка безпеки свідчить про включення в зміст повідомлення приватної або конфіденційної інформації.

Підтримка захищеної електронної пошти на основі PKI

Всі сервіси, пропонованими S/MIME (обох версій), покладаються на сертифікати й надійність зв'язування електронної адреси суб'єкта з його відкритим ключем. Адреса електронної пошти (часто називаєма адресою RFC 822 [8]) повинна бути представлена у доповненні сертифіката Subject Alternative Name (альтернативне ім'я суб'єкта). Якщо використовується друга версія S/MIME, то адреса електронної пошти вказується як відмітне ім'я суб'єкта (emailAddress).

Відправник зашифрованого повідомлення повинен бути впевнений, що відкритий ключ належить саме тому одержувачеві, якому він адресує своє повідомлення, у противному випадку доступ до змісту повідомлення може одержати сторонній. Аналогічно, поширюючи ключі шифрування симетричного ключа тільки членам списку розсилання відправника, агент MLA покладається на правильність зв'язування ідентичності одержувача і його відкритого ключа. Відправник і агент MLA порівнюють ідентифікаційну ознаку одержувача, зазначену в сертифікаті, з адресою електронної пошти одержувача, якому посилає своє повідомлення відправник.

Одержувач повідомлення, завіреного цифровим підписом, повинен бути впевнений, що відкритий ключ підпису, що необхідний для верифікації підпису на повідомленні, належить відправникові. Для цього він порівнює адресу електронної пошти, зазначену в полі SENDER (або FROM) отриманого повідомлення, з адресою, представленою в сертифікаті.

Аналогічні дії виконуються при перевірці квитанцій, завірених цифровим підписом, – для цього що перевіряє порівнює адресу електронної пошти із запиту на квитанцію з адресою електронної пошти в сертифікаті особи, що подписали квитанцію.

Засоби безпеки транспортного рівня

Протокол безпеки транспортного рівня Transport Layer Protocol (TLS) [9] забезпечує захист комунікацій між додатками, розробленими в архітектурі "клієнт-сервер", в основному між web-браузером і web-сервером. World Wide Web є найбільш популярний Інтернет-сервісом після електронної пошти. TLS найбільше часто застосовується для захисту web-контента, але може використовуватися з будь-яким протоколом прикладного рівня. Специфікація TLS базується на популярному протоколі Secure Socket Layer (SSL) [10], розробленому корпорацією Netscape. Ці протоколи створювалися для забезпечення автентифікації, цілісності й конфіденційності даних, якими обмінюються взаємодіючі один з одним додатки. Обидва протоколи мають дворівневу організацію: протокол установлення з'єднання (Handshake Protocol) і протокол передачі записів (Record Protocol).

Протокол установлення з'єднання дозволяє серверу й клієнтові виконати взаємну автентифікацію, погодити застосовуваний алгоритм шифрування й криптографічні параметри перед тим, як протокол прикладного рівня почне передачу даних. **Протокол передачі записів** забезпечує захист протоколів більше високого рівня, включаючи протокол установлення з'єднання. Протокол передачі записів залежить від надійності транспортного протоколу, такого як TCP.

Протоколи SSL і TLS незалежні від протоколів прикладного рівня, тому будь-який протокол прикладного рівня може прозоро оперувати поверх SSL і TLS. Протоколи SSL і TLS забезпечують три сервіси безпеки [11]:

−автентифікацію (підтвердження ідентичності з'єднання: протокол установлення з'єднання використовує сертифікати й верифікацію цифрових підписів для підтвердження ідентифікаційних ознак і повноважень вилученого додатка);

−цілісність (захист даних протоколу від несанкціонованої модифікації: протокол передачі записів використовує значення біта контролю цілісності для підтвердження того, що передані дані не змінювалися);

−конфіденційність (забезпечення таємності з'єднання: після узгодження симетричного ключа шифрування на основі протоколу встановлення з'єднання виконується шифрування даних, якими обмінюються сторони під час сеансу зв'язку).

Протоколи SSL і TLS здатні підтримувати взаємну автентифікацію сторін, але зазвичай на базі сертифіката виконується автентифікація сервера клієнтом, а потім клієнт автентифікується іншим способом, наприклад, уводячи по запиту сервера своє ім'я й пароль або номер своєї кредитної карти й дату закінчення її терміну дії.

Протокол установлення з'єднань

Протокол установлення з'єднань Handshake Protocol складається як би із трьох підпротоколів, які дозволяють виконати автентифікацію сторін, погодити алгоритми й параметри безпеки для протоколу передачі записів [11].

Handshake Protocol відповідає за організацію сеансу взаємодії між клієнтом і сервером для протоколу передачі записів, зокрема за узгодження характеристик сеансу:

−ідентифікатора сеансу (Session identifier), тобто довільної послідовності біт, обраної сервером для ідентифікації сеансу;

−сертифіката з'єднання (Peer certificate), що представляє собою сертифікат X.509; цей елемент може бути відсутнім, якщо автентифікація не потрібна;

−методу стиску (Compression method), тобто алгоритму стиску даних перед їхнім шифруванням;

−специфікатора шифрування (Cipher spec), що задає ідентифікатори алгоритму шифрування даних і алгоритму гешування, а також деякі криптографічні атрибути (наприклад, розмір геш-коду);

−головного секрету (Master secret), що представляє собою секретне значення, розділене між клієнтом і сервером;

−ознаки встановлення нового з'єднання (Is resumable) на основі поточного сеансу.

Ці характеристики потім використовуються для установки параметрів безпеки в протоколі передачі записів. Можливість установити кілька захищених з'єднань під час одного сеансу особливо важлива, коли клієнтові й серверу необхідно встановити кілька короткочасних з'єднань.

У результаті роботи протоколу Handshake Protocol формуються криптографічні параметри. Коли клієнт і сервер починають взаємодію, то погоджують версію протоколу, криптографічні алгоритми, автентифікують один одного (за бажанням) і використовують криптографію з відкритими ключами для поділу загального секрету. Робота протоколу Handshake Protocol виконується за шість кроків [12].

1-й крок. Обмін привітальними повідомленнями для узгодження алгоритмів і обміну випадковими числами.

2-й крок. Обмін криптографічними параметрами для узгодження початкового секрету.

3-й крок. Опціональний обмін сертифікатами й криптографічною інформацією для взаємної автентифікації клієнта й сервера.

4-й крок. Генерація головного секрету на основі початкового секрету й обмін випадковими числами.

5-й крок. Формування параметрів безпеки для протоколу передачі записів.

6-й крок. Оповіщення про розрахунок тих же самих параметрів безпеки й коректному завершенні з'єднання.

Коли клієнт бажає встановити нове з'єднання на основі даного сеансу або повторити цей сеанс, то відправляє своє привітальне повідомлення з ідентифікатором сеансу, що повинен бути відновлений. Якщо сервер усе ще зберігає в кеш-пам'яті параметри сеансу й бажає відновити з'єднання, то у відповідь відправляє своє привітальне повідомлення з тим же самим ідентифікатором сеансу. У цей момент клієнт і сервер обмінюються повідомленнями про зміну параметрів шифрування й завершенні формування з'єднання.

При обміні сертифікатами й ключами передаються дані, необхідні для генерації початкового секрету. Якщо використовується алгоритм RSA, то клієнт генерує випадкове початкове число й шифрує його за допомогою відкритого RSA-ключа сервера. Якщо застосовується алгоритм Діффі-Хеллмана, клієнт і сервер обмінюються відкритими ключами, а потім як початковий секрет використовується результат обчислення ключа узгодження ключів по алгоритму Діффі-Хеллмана.

Головний секрет і симетричні ключі генеруються за допомогою псевдовипадкової функції PRF (PseudoRandom Function), отриманої на основі алгоритмів SHA-1 і MD5. Щоб гарантувати безпеку, застосовуються дві різні односпрямовані геш-функции. При першому застосуванні функції PRF генерується головний секрет з початкового секрету й випадкових чисел, отриманих на основі привітальних повідомлень клієнта й сервера. У результаті повторного застосування функції PRF до тих же самим вихідним даним одержують два симетричних ключі, два значення початкового вектора й два значення MAC (коду автентифікації повідомлення) секрету. При поновленні сеансу використовується вже відоме для даного сеансу значення головного секрету, але генеруються нові випадкові числа на основі нових привітальних повідомлень клієнта й сервера, тому в результаті формується новий ключовий матеріал.

Протокол передачі записів використовує один симетричний ключ, одне значення початкового вектора й один MAC секрет для захисту трафіку "клієнт-сервер", а також інші значення перерахованих параметрів для захисту трафіку "сервер-клієнт", – у цілому, для коректної роботи протоколу передачі записів потрібно шість секретних чисел.

Протокол передачі записів

Протокол передачі записів Record Protocol складається з декількох підрівнів. Обробка даних протоколу прикладного рівня полягає в їхній розбивці на керовані блоки, стиску, постачанні імітовставки, шифруванні й наступній передачі результату. Отримані дані обробляються у зворотному порядку: розшифровуються, перевіряються на цілісність, піддаються декомпресії, складанню, а потім доставляються додатку [6].

На підрівні фрагментації інформація розбивається на записи, довжина кожного запису не перевищує 16384 байтів. Допускається агрегування декількох однотипних повідомлень в один запис, а також розбивка одного повідомлення протоколу прикладного рівня на кілька записів. На підрівні стиску виконується стиск або декомпресія всіх фрагментів. Очевидно, що при компресії не повинне бути втрати даних. Потім обчислюється імітовставка, тобто перевіряється цілісність стислого запису, а отримане значення й стислий запис шифруються. Перевірка цілісності здійснюється за допомогою коду автентифікації повідомлення (MAC) або коду автентифікації, отриманого на основі геш-коду повідомлення (HMAC).

Після прийняття даних текст розшифровується, для перевірки його цілісності повторно обчислюється MAC, причому обчислення виконуються з використанням порядкового номера запису з метою виявлення загублених, зайвих або повторно стислих записів.

Підтримка безпеки транспортного рівня на основі PKI

Сертифікати є центральним компонентом всіх сервісів автентифікації й керування ключами, пропонованих як TLS, так і SSL. Ці сервіси покладаються на зв'язування ідентичності суб'єкта з його відкритим ключем. Для ідентифікації web-серверів рекомендується використовувати DNS-імена (типу www.alpha.com) і вказувати їх у доповненні сертифікатів Subject Alternative Name (параметр dNSName). Якщо DNS-ім'я не представлене в сертифікаті, то для ідентифікації використовується відмітне ім'я суб'єкта.

Зазвичай при взаємодії клієнта й сервера сервер пред'являє сертифікат, а клієнт – ні, у результаті чого відбувається однобічна автентифікація сервера клієнтом, що зберігає свою анонімність. Сервер може зажадати від клієнта автентифікації, запитуючи сертифікат по протоколі Handshake Protocol.

У цьому випадку клієнт повинен мати сертифікат і надати його серверу. Зазвичай клієнт пред'являє сертифікат ключа підпису.

У процесі верифікації завіреного цифровим підписом повідомлення, відправленого по протоколу Handshake Protocol, з'ясовується, чи здатний клієнт згенерувати підпис за допомогою свого секретного ключа, що відповідає відкритому ключу підпису, що втримується в сертифікаті.

Сертифікати, використовувані для підтримки безпеки транспортного рівня, також повинні містити доповнення Key Usage, що відбиває відповідне призначення відкритого ключа, що втримується в сертифікаті:

– цифровий підпис, якщо необхідно виконувати верифікацію підписів;

– шифрування ключів, щоб забезпечити RSA-шифрування;

– узгодження ключів, якщо необхідно підтримку узгодження ключів методом Діффі-Хеллмана.

Клієнт повинен бути впевнений, що відкритий ключ належить серверу. Якщо використовується некоректний відкритий ключ, то стороння особа може одержати початковий секрет і можливість згенерувати головний секрет і всі інші секретні параметри, що обчислюються з його допомогою. Сертифікат підтверджує приналежність відкритого ключа серверу.

При автентифікації клієнта сервер покладається на приналежність відкритого ключа клієнтові й ухвалює рішення щодо можливості доступу. Якщо використовується некоректний відкритий ключ, то доступ до сервера може одержати сторонню особу, а не та сторона, який доступ необхідний. Сертифікат забезпечує необхідне зв'язування відкритого ключа з ідентичністю клієнта.

Засоби безпеки IP-рівня

Сукупність механізмів IPsec забезпечує основу для захисту мережного трафіку на IP-рівні, безпеку IP-пакетів, захищеного взаємодії мобільних систем з корпоративною мережею, реалізації віртуальних приватних мереж (Virtual Private Networks – VPN) і т.п. Сімейство специфікацій IPsec представлено серією з 10 документів, розроблених робочою групою IP Security Protocol організації IETF і утримуючі відомості про архітектуру IPsec [14],

формуванні контекстів безпеки, керуванні ключами й базовими протоколами. Ядро IPsec становлять три протоколи: протокол автентифікуючого заголовку (Authentication Header, AH) [14], протокол інкапсулюючий захист вмісту (Encapsulating Security Payload, ESP) [14] і протокол обміну ключами в Інтернеті (Internet Key Exchange, IKE) [14]. Функції по підтримці захищеного каналу передачі даних по мережах IP розподіляються між цими протоколами в такий спосіб:

–протокол AH забезпечує цілісність IP-пакетів, автентифікацію джерела даних, а також захист від відтворення раніше переданих IP-пакетів;

–протокол ESP підтримує конфіденційність, автентифікацію й цілісність IP-пакетів, а також частковий захист від аналізу трафіку;

–протокол IKE дозволяє взаємодіючим сторонам автоматично генерувати й безпечно розподіляти симетричні секретні ключі.

Контексти безпеки

Контексти безпеки (Security Associations) утворюють основу криптографічних сервісів безпеки на базі протоколів IPsec. Для захисту двостороннього зв'язку між вузлами мережі необхідні два контексти безпеки: один – для вхідних потоків, інший – для вихідних. Контексти безпеки містять інформацію про IP-адреси, тип захисного протоколу (AH або ESP), криптографічних алгоритмах, ключах для автентифікації й шифрування й періоді їхньої дії.

Контекст безпеки унікально ідентифікується трьома елементами:

–індексом параметрів безпеки (Security Parameters Index – SPI);

–цільовою IP-адресою;

–ідентифікатором захисного протоколу.

Таблиця 1 – Режими, використовувані для різних типів з'єднань

	Хост	Маршрутизатор або міжмережний екран
Хост	Транспортний режим або тунельний режим	Тунельний режим
Маршрутизатор або міжмережний екран	Тунельний режим	Тунельний режим

Тунельний режим зазвичай реалізують на спеціально виділених захисних шлюзах, у ролі яких можуть виступати маршрутизатори або міжмережні екрани (рисунок 1).

Рисунок 1 – Стеки протоколів у різних режимах

У транспортному режимі заголовок протоколу (AH або ESP) розташовується в стеці протоколів після заголовка вихідного IP-пакету й перед заголовками протоколів більш високого рівня [7]. У тунельному режимі заголовок протоколу (AH або ESP) розташовується в стеці протоколів між двома заголовками: після заголовка зовнішнього IP-пакета й перед заголовком внутрішнього вихідного IP-пакета (рисунок 1).

Протокол автентифікуючого заголовка AH

Протокол автентифікуючого заголовка AH забезпечує:

–цілісність IP-пакетів, даних протоколів більше високого рівня й певних полів IP-заголовків;

–автентифікацію джерела даних (на основі IP-адреси вузла мережі або ім'я кінцевого користувача);

–захист від помилкового відтворення раніше переданих IP-пакетів.

Контроль цілісності базується на перевірці коду автентифікації гешированного повідомлення Hashed Message Authentication Code (HMAC), що обчислюється за допомогою геш-функції MD5 або SHA-1 із секретним симетричним ключем, що відомий обом взаємодіючим сторонам.

Рисунок 2 ілюструє типові поля даних протоколу AH. Протокол містить п'ять полів: Next Header, Length, SPI, Sequence Number і Authentication Data.

Рисунок 2 – Поля даних протоколу AH

Поле Next Header (наступний заголовок) указує, який протокол більш високого рівня інкапсулюється за допомогою AH. У тунельному режимі це поле зазвичай містить IP v4 або IP v6, а в транспортному режимі – TCP, UDP або ICMP.

Поле Length (довжина) задає розмір заголовка протоколу AH. Розмір залежить від типу використовуваної геш-функції, значення HMAC утримується в єдиному полі змінної довжини.

Поле SPI (індекс параметрів безпеки) містить 32-розрядне довільне значення, що ідентифікує контекст безпеки.

Поле Sequence Number (порядковий номер) використовується для завдання значення лічильника IP-пакетів (32-розрядного монотонно зростаючого) і захисту від відтворення пакетів. Відправник пакета повинен задавати це значення, а одержувач пакета може або обробляти його, або ігнорувати.

Поле Authentication Data (автентифікаціонные дані) містить значення HMAC для даного IP-пакета. Це поле має змінну довжину, що повинна бути кратна 32 розрядам.

При передачі пакета його порядковий номер, що вказується в поле Sequence Number, збільшується, а потім поля IP-заголовка й протоколу більш високого рівня гешируются для створення HMAC на основі загального секретного симетричного ключа. Після одержання IP-пакета одержувачем виконується та ж сама послідовність операцій. Якщо обчислене їм значення HMAC не відповідає значенню, отриманому по протоколу AH, то пакет не приймається. Крім того, якщо контекст безпеки містить інформацію про застосування засобу захисту від відтворення пакетів, то значення поля Sequence Number зменшується на одиницю, тобто відновлюється колишнє значення лічильника IP-пакетів.

Протокол інкапсулюючий захист вмісту ESP

Протокол інкапсулюючий захист вмісту ESP підтримує конфіденційність, автентифікацію й цілісність IP-пакетів. Конфіденційність забезпечується шляхом шифрування вмісту IP-пакетів, а також частини заголовка й трейлера (хвостової частини) протоколу ESP; надійність шифрування залежить, насамперед, від використовуваного

алгоритму шифрування. Автентифікація джерела даних і захист цілісності здійснюється на основі HMAC (як і в протоколі AH). Хоча сервіси конфіденційності й автентифікації (який включає цілісність) є опціональними, у кожному контексті безпеки повинен бути заданий, принаймні, один сервіс безпеки.

Як алгоритми шифрування в протоколі ESP використовуються алгоритми DES і Triple-DES, для обчислення HMAC застосовується геш-функція типу MD5 або SHA-1. рисунок 3 ілюструє типові поля даних протоколу ESP. Заголовок ESP містить два поля: SPI і Sequence Number, їхній синтаксис і семантика збігається з однойменними полями протоколу AH. Трейлер ESP складається із чотирьох полів: Padding, Pad Length, Next Header і Authentication Data.

Поле Padding (заповнювач) використовується для того, щоб розмір шифруемых даних був кратний розміру криптографічного блоку.

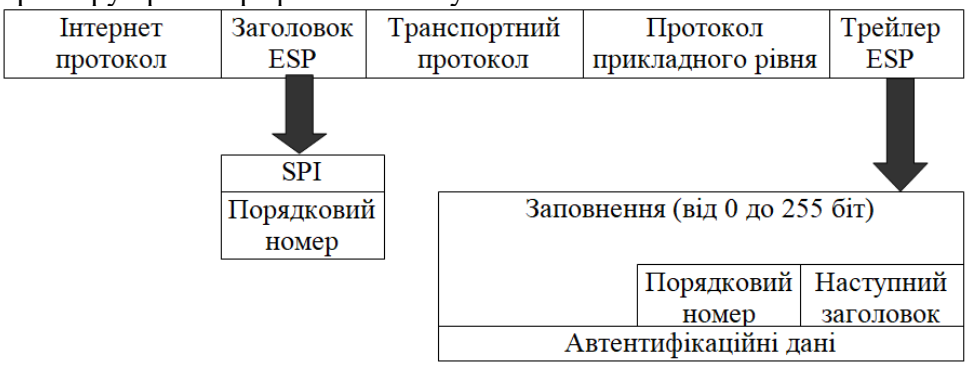

Рисунок 3 – Поля даних протоколу ESP

Поле Pad Length (довжина заповнювача) характеризує розмір заповнювача й залежить від використовуваного алгоритму шифрування й заданого рівня конфіденційності IP-трафіку.

Поле Next Header (наступний заголовок) містить інформацію про те, який протокол більше високого рівня інкапсулюється за допомогою ESP. У тунельному режимі це поле зазвичай містить IP v4 або IP v6, а в транспортному режимі – TCP, UDP або ICMP.

Поле Authentication Data (автентифікаційні дані) містить значення HMAC для даного IP-пакета. Це поле має змінну довжину, що повинна бути кратна 32 розрядам. Якщо автентифікація джерела даних або захист цілісності не потрібна, то це поле відсутнє або має нульову довжину.

При передачі пакета його порядковий номер, що вказується в поле Sequence Number, збільшується, а потім поля заголовка ESP, протоколу більш високого рівня й трейлера ESP гешуються для створення HMAC на основі загального секретного симетричного ключа. Потім поля протоколу більш високого рівня й трейлер ESP (за винятком автентифікаціоних даних) шифруються; якщо необхідно початковий вектор, то він випереджає шифртекст. Після одержання IP-пакета одержувачем виконується розшифрування й розрахунок того ж самого значення HMAC. Якщо обчислене їм значення HMAC не відповідає значенню, отриманому в трейлері ESP, то пакет не приймається. Крім того, якщо контекст безпеки містить інформацію про використання засобу захисту від відтворення пакетів, то значення поля Sequence Number зменшується на одиницю, тобто відновлюється колишнє значення лічильника IP-пакетів.

Протокол обміну ключами IKE

Широке використання IPsec вимагає масштабованого, автоматизованого керування контекстами безпеки. Формування контекстів безпеки по запиту й використання засобів захисту від відтворення пакетів у протоколах AH і ESP неможливо без роботи протоколу обміну ключами в Інтернеті IKE [14]. Протокол IKE розроблений на основі протоколу керування ключами й контекстами безпеки Інтернету – Internet Security Associations and Key Management Protocol (ISAKMP) [14] і протоколу обчислення ключів – OAKLEY Key Determination Protocol (OAKLEY) [14]. Протокол ISAKMP забезпечує незалежну від

криптографічного механізму автентифікацію й задає структуру обміну ключами. Протокол IKE базується на функціональності протоколу ISAKMP, а для формування симетричного ключа використовує можливості протоколу OAKLEY. Як алгоритм узгодження ключів застосовується алгоритм Діффі-Хеллмана.

Робота протоколу IKE виконується за два етапи. На першому етапі встановлюється автенифікований і шифруємий канал зв'язку. Це вимагає формування двох контекстів безпеки – по одному для зв'язку в одному напрямку. Для автентифікації сторін використовуються сертифікати відкритих ключів підпису (як алгоритм цифрового підпису застосовується DSA). На другому етапі формуються контексти безпеки для AH і ESP.

Підтримка безпеки IP-рівня на основі PKI

Сертифікати служать головними компонентами автентифікації на основі IKE. Коли ідентифікується хост або захисний шлюз, найбільш кращою формою ідентифікаційних даних є DNS-імена або IP-адреси, які вказуються в доповненні сертифіката Subject Alternative Name (параметри dNSName і iPAddress відповідно). При ідентифікації користувача рекомендується використовувати адресу електронної пошти або звичайне ім'я. Адреса електронної пошти втримується в доповненні сертифіката Subject Alternative Name, а звичайне ім'я входить до складу відмітного ім'я суб'єкта. Неможливість зв'язати ідентичність суб'єкта з його відкритим ключем робить автентифікацію неможливою. Неправильна автентифікація на основі протоколу IKE може привести до помилкового зв'язування ключа й реалізації IPsec, у результаті неавторизований користувач (або навіть група комп'ютерів) може одержати доступ, наприклад, до віртуальної приватної мережі.

Сертифікати, призначені для захисту трафіку Інтернету, повинні включати доповнення Key Usage, що відбиває відповідне призначення відкритого ключа, що втримується в сертифікаті (наприклад, цифровий підпис, якщо необхідно виконувати верифікацію підписів). Крім того, у доповненні сертифіката Extended Key Usage повинні явно вказуватися ті додатки, для підтримки яких призначений даний відкритий ключ, зокрема додаток IPsec.

Отже, сертифікати є базовим компонентом сервісів безпеки, надаваних S/MIME, TLS і IPsec.

S/MIME за допомогою сертифікатів ідентифікує користувачів. Сервіси автентифікації, цілісності, невідказуємості й конфіденційності захищеної електронної пошти залежать від сертифікатів. Сертифікати підтримують шифрування й завірення цифровим підписом поштових повідомлень.

TLS за допомогою сертифікатів ідентифікує користувачів і сервери. Від сертифікатів залежать сервіси автентифікації, цілісності й конфіденційності. Сертифікати підтримують шифрування потоку, автентифікацію потоку й цілісність потоку.

IPsec за допомогою сертифікатів ідентифікує користувачів, хости й шлюзи безпеки. Від сертифікатів залежать сервіси автентифікації. На базі сертифікатів виконується взаємна автентифікація взаємодіючих сторін при обміні відкритими ключами Діффі-Хеллмана, які, у свою чергу, використовуються для безпечного розподілу симетричних секретних ключів. Секретні ключі забезпечують підтримку автентифікації, цілісності й конфіденційності IP-пакетів.

Типові сценарії використання PKI

Повнофункціональна PKI – це ідеальне подання можливої інфраструктури, поки невідомі PKI-продукти, що реалізують всі перераховані в таблиці 2 функції. Сучасні PKI зазвичай призначені для рішення певної задачі або ряду задач. Конкретні реалізації PKI являють собою деякі підмножини повнофункціональної архітектури.

Таблиця 2 – Повнофункціональна PKI

УЦ	**Репозиторій**	**Анулювання сертифікатів**
Резервне зберігання ключів	Відновлення ключів	Автоматичне відновлення ключів
Керування історіями ключів	Крос-сертифікація	Клієнтське ПЗ
Автентифікація	Цілісність	Конфіденційність
Захищене датування	Нотаризація	Невідказуємість
Захищений архів даних	Розробка повноважень/політики	Перевірка повноважень/політики

Розглянемо чотири розповсюджених на сьогоднішній день сценарія використання PKI [4]. У таблиці 3 представлений Інтернет-PKI, що підтримує звичайну електронну пошту (між знайомими) і навігацію в World Wide Web за допомогою SSL-сервера автентифікації. Такий сценарій вимагає наявності УЦ для випуску сертифікатів відкритих ключів і підтримки основних сервісів автентифікації, цілісності й конфіденційності.

У цьому сценарії не передбачене використання репозитория (сертифікати пересилаються по протоколі зв'язку), не виконується перевірка статусу одержувача електронної пошти (або навіть сертифіката сервера) і керування життєвим циклом ключів і сертифікатів, відсутнє клієнтське програмне забезпечення (як окремий модуль, викликуваний за допомогою браузера), не потрібно ні крос-сертифікація, ні додаткові сервіси, що базуються на PKI.

Таблиця 3 – Інтернет-PKI

УЦ	Репозиторій	Анулювання сертифікатів
Резервне зберігання ключів	Відновлення ключів	Автоматичне відновлення ключів
Керування історіями ключів	Крос-сертифікація	Клієнтське ПЗ
Автентифікація	**Цілісність**	**Конфіденційність**
Захищене датування	Нотаризація	Невідказуємість
Захищений архів даних	Розробка повноважень/політики	Перевірка повноважень/політики

Таблиця 4 ілюструє функції PKI у сценарії, коли для доступу до корпоративної мережі ззовні використовується браузер і виконується SSL-автентифікація клієнтів.

Таблиця 4 – Екстранет-безпека (через SSL-автентифікацію клієнтів)

УЦ	Репозиторій	**Анулювання сертифікатів**
Резервне зберігання ключів	Відновлення ключів	Автоматичне відновлення ключів
Керування історіями ключів	Крос-сертифікація	Клієнтське ПЗ
Автентифікація	**Цілісність**	**Конфіденційність**
Захищене датування	Нотаризація	Невідказуємість
Захищений архів даних	**Розробка повноважень/політики**	**Перевірка повноважень/політики**

У таблиці 5 представлений набір функцій PKI для сценарію захищеної корпоративної електронної пошти. У цьому сценарії може знадобитися керування життєвим циклом ключів і сертифікатів і вбудоване клієнтське програмне забезпечення, тому що стандартні пакети електронної пошти не завжди підтримують безпеку, засновану на PKI. У цьому випадку не потрібні додаткові сервіси, що базуються на PKI, і крос-сертифікація.

Нарешті, у сценарії підтримки міжкорпоративних транзакцій з використанням цифрових підписів можуть знадобитися багато можливостей повнофункціональної PKI, зокрема сильна автентифікація й авторизація, перевірка статусу сертифікатів, розробка й перевірка повноважень і політики, сервіс невідказуємості (підтримка множинних пар ключів, зберігання прийнятих електронних документів із цифровим підписом і т.д.). Якщо корпорації мають свої власні PKI, то необхідно крос-сертифікацію. У даному сценарії можна обійтися без архівування даних, датування й нотаризації.

Таблиця 5 – Захищена корпоративна електронна пошта

УЦ	**Репозиторій**	**Анулювання сертифікатів**
Резервне зберігання ключів	Відновлення ключів	Автоматичне відновлення ключів
Керування історіями ключів	Крос-сертифікація	Клієнтське ПЗ
Автентифікація	Цілісність	Конфіденційність
Захищене датування	Нотаризація	Невідказуємість
Захищений архів даних	Розробка повноважень/політики	Перевірка повноважень/політики

Таблиця 6 – Міжкорпоративні транзакції із цифровим підписом

УЦ	**Репозиторій**	**Анулювання сертифікатів**
Резервне зберігання ключів	Відновлення ключів	Автоматичне відновлення ключів
Керування історіями ключів	Крос-сертифікація	Клієнтське ПЗ
Автентифікація	Цілісність	Конфіденційність
Захищене датування	Нотаризація	Невідказуємість
Захищений архів даних	Розробка повноважень/політики	Перевірка повноважень/політики

Таблиці 3, 4, 5 і 6 підтверджують, що PKI, що реалізують приватні сценарії, є підмножинами повнофункціональної PKI. Технологія PKI продовжує розвиватися, але вже зараз ясно, що багато постачальників програмного й апаратного забезпечення PKI будуть орієнтуватися на реалізацію повнофункціональних систем, а не на PKI-продукти вузького призначення. Очевидно, що в багатьох випадках простіше й економічно більш ефективно адаптувати повнофункціональний продукт для рішення специфічної проблеми, чим розробляти й підтримувати кілька окремих продуктів, кожний з яких призначений для рішення однієї або двох специфічних проблем. У багатьох середовищах PKI відбудеться неминучий перехід від приватних рішень приватних проблем до повнофункціонального PKI, що пропонує універсальне рішення проблем безпеки для широкого кола додатків.

Розробка структурної схеми

На рисунку 4 зображено структурну схему інфраструктури відкритих ключів, та місце S/MIME у ній. Розглянемо компоненти структурної схеми. Почнемо розгляд з точену. Замість використання доступу по паролю до корпоративних ресурсів, для входу в домен і при використанні корпоративної пошти можна використовувати апаратну автентифікацію й захист електронної переписки в мережах, побудованих на базі Windows 10/11. У пропонованому рішенні використовуються вбудовані інструменти безпеки Windows і електронні ідентифікатори **токен** як носії ключової інформації.

Що забезпечується при використанні цього продукту? Авторизація користувачів (вхід у домен при підключенні токена й блокування сесії після його від'єднання). При роботі з поштою – електронний цифровий підпис поштових повідомлень і їхнє шифрування. Доступ

до корпоративних ресурсів по пред'явленні токена й, звичайно, надійне зберігання й використання сертифікатів. Частина з перерахованих задач може бути використана й на домашньому комп'ютері, наприклад, при роботі з поштою, а також для віддаленого підключення до корпоративних ресурсів. Давайте розберемо, що і як необхідно виконати для реалізації перерахованих задач.

Токен як додатковий пристрій не може бути пізнаний операційною системою, оскільки він не є стандартним устаткуванням. Неможлива й установка токена за допомогою inf-файлу, оскільки для коректної установки потрібний вимір деяких параметрів для автоматичної конфігурації драйвера. Тому доводиться для установки використовувати спеціальне програмне забезпечення. І от що ще варто пам'ятати. Не підключайте токен до комп'ютера до того, як ви встановите драйвер. Якщо все-таки підключення буде виконано раніше, припиніть роботу стандартного майстра установки USB-пристроїв, витягніть ключ із порту й установіть драйвер. Крім драйверів у процесі інсталяції на диск будуть скопійовані й утиліти для роботи з токеном (утиліта адміністрування й браузер сертифікатів). Кількість одночасно використовуваних токенів залежить від операційної системи й кількості USB-портів.

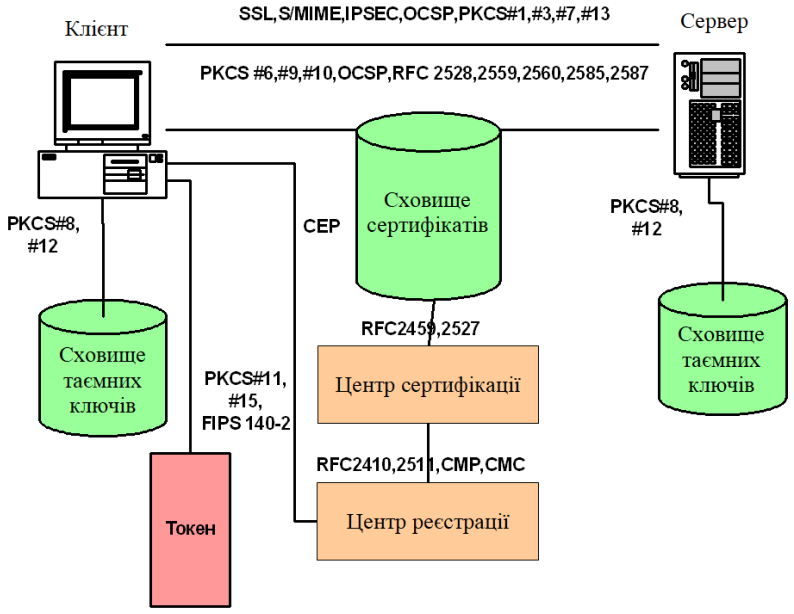

Рисунок 4 – Структурна схема PKI

Центр сертифікації (Удостовірюючий центр) (Certification authority, CA) – це організація, що випускає сертифікати ключів електронного цифрового підпису.

Центр сертифікації – це компонента глобальної служби каталогів, відповідальназа керування криптографічними ключами користувачів. Відкриті ключі й інша інформація про користувачів зберігається центрами сертифікації у вигляді цифрових сертифікатів, що мають наступну структуру:

– серійний номер сертифіката;
– об'єктний ідентифікатор алгоритму електронного підпису;
– ім'я центра, що засвідчує;
– строк придатності;
– ім'я власника сертифіката (ім'я користувача, якому належить сертифікат);
– відкриті ключі власника сертифіката (ключів може бути трохи);
– об'єктні ідентифікатори алгоритмів, асоційованих з відкритими ключами власника сертифіката; електронний підпис, згенерований з використанням секретного ключа центра, що засвідчує (підписується результат гешування всієї інформації, що зберігається в сертифікаті).

Центр реєстрації – опціональна компонента інфраструктури, призначена для реєстрації кінцевих користувачів і забезпечення їхньої взаємодії із центром сертифікації.

CRL – список відозваних сертифікатів. Оснащення «Центру сертифікації» можна використовувати для відкликання сертифіката, для адміністрування публікації списків відкликання сертифікатів (CRL) і для завдання точок поширення списків CRL, які опубліковані в кожному сертифікаті, виданому центром сертифікації (ЦС).

Відкликання сертифікатів

Щоб допомогти в досягненні цілісності інфраструктури відкритого ключа (PKI) організації, адміністратор ЦС відзиває сертифікат, якщо суб'єкт сертифіката залишає організацію, порушений закритий ключ сертифіката або існують інші причини, по яких сертифікат більше не може вважатися дійсним. При відкликанні сертифіката ЦС він додається в список відкликання сертифікатів (CRL) цього ЦС. Це може відбуватися зі створенням нового списку CRL або з використанням різницевого списку CRL, що є невеликим списком сертифікатів, відкликаних з моменту останнього заповнення списку CRL.

Розклад публікації списку відкликання сертифікатів (CRL)

Одна з можливостей служб сертифікації складається в автоматичній публікації оновленого списку CRL по закінченні певного періоду часу, заданого адміністратором ЦС. Цей інтервал часу називається періодом публікації CRL. Після початкової установки ЦС період публікації встановлюється рівним одному тижню (відповідно до часу на локальному комп'ютері, починаючи з дати початкової установки ЦС). Періоди публікації списків CRL і різницевого CRL можуть задаватися незалежно.

LDAP (Lightweight Directory Access Protocol – «полегшений протокол доступу до каталогів») – це мережний протокол для доступу до служби каталогів X.500, розроблений IETF як полегшений варіант розробленого ITU-T протоколу DAP. LDAP – відносно простий протокол, що використовує TCP/IP і дозволяє робити операції автентифікації (bind), пошуку (search) і порівняння (compare), а також операції додавання, зміни або видалення записів. Звичайно LDAP-Сервер приймає вхідні з'єднання на порт 389 по протоколах TCP або UDP. Для LDAP-сеансів, інкапсульованих в SSL, звичайно використовується порт 636.

Висновки. У статті наведені теоретичне узагальнення й рішення наукового завдання дослідження методів повідомлень електронної пошти. Рішення даного завдання полягало у вирішенні наступних задач: Був проведений огляд існуючих систем повідомлень електронної пошти. Досліджена система повідомлень електронної пошти. На основі отриманих результатів досліджень створена програмна реалізація системи повідомлень електронної пошти. Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання повідомлень електронної пошти. Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Список літератури

1. Smirnov O., Kovalenko O., Kovalenko A., Kavun S. «Quantitative Risk Assessment Method Development in the Context of the SDLC-model». 2021 IEEE 8th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T), 2021, pp. 203-208, doi: 10.1109/PICST54195.2021.9772143 (Scopus).

2. Smirnov O., Neskorodieva T., Fedorov E., Rymar P. «Neural Network Modeling Method of Transformations Data of Audit Production with Returnable Waste». CEUR Workshop Proceedings Volume 3101, 2021, Pages 192-207. (Scopus).

3. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova K. «Data hiding scheme based on spread sequence addressing». CEUR Workshop Proceedings Volume 2805, 2020, Pages 44-58. (Scopus).

4. Смірнова Т.В., Поліщук Л.І., Смірнов О.А., Буравченко К.О., Макевнін А.О., «Дослідження хмарних технологій як сервісів», Кібербезпека: освіта, наука, техніка. № 3(7). С. 43-62. 2020.

5. Смірнов О.А. Дисперсійний аналіз мережного трафіку для забезпечення інформаційної безпеки телекомунікаційних систем / О.О. Кузнецов, О.А. Смірнов, Д.О. Даниленко // Інформаційна та економічна безпека: сучасний стан та тенденції розвитку : монографія за заг. ред. – Х.: ХІБС УБС НБУ – 2014 – С. 82-100.
6. Смірнов О.А. Дослідження методів виявлення вторгнень в телекомунікаційні системи та мережі / Д.О. Даниленко, О.А. Смірнов, Є.В. Мелешко // Системи озброєння і військова техніка. – Випуск 1(29) – Х.: ХУПС – 2012. – С. 92-100
7. Смирнов А.А. Метод обнаружения вредоносного программного обеспечения. Часть 1. Корреляционный анализ сетевого трафика // А.А.Смирнов, Д.А. Даниленко, Е.В.Мелешко // Науково-технічний журнал «Інформаційно-керуючі системи на залізничному транспорті» – Випуск 4(95). – Х.: УкрДАЗТ – 2012. – С. 8-14.
8. Смирнов А.А. Методы обнаружения вредоносного программного обеспечения в телекоммуникационных системах и сетях / Д.А. Даниленко // Збірник наукових праць "Системи обробки інформації". – Випуск 3(101) том 2. – Х.: ХУПС – 2012. – С. 152-155.
9. Смирнов А.А. Системы обнаружения и предотвращения вторжений для защиты телекоммуникационных сетей от вредоносного программного обеспечения / Д.А. Даниленко, А.А. Смирнов, А.В. Коваленко // Системи управління, навігації та зв'язку. – Випуск 1 (21) том 2. – Київ: ДП «ЦНДІНУ». – 2012. – С. 183-186.
10. Смирнов А.А. Системы обнаружения и предотвращения вторжений для защиты компьютерных сетей от вредоносного программного обеспечения / Д.А. Даниленко, А.А. Смирнов, И.Г. Кирилов // Збірник тез доповідей науково-практичної конференції «Застосування інформаційних технологій у підготовці та діяльності сил охорони правопорядку». м. Харків. 21-22 березня 2012 р. – Харків. АВВ МВС. – 2012. – С. 70-71.
11. Смірнов О.А. Дослідження методів виявлення вторгнень в телекомунікаційні мережі для підвищення інформаційної безпеки // Д.О. Даниленко // Збірник тез науково-практичної конференції «Захист інформації в інформаційно-комунікаційних системах». м. Київ. 24-27 квітня 2012 р. – Київ: НАУ. – 2012. – С. 22-25.
12. Смирнов А.А. Исследование систем обнаружения и предотвращения вторжений для защиты телекоммуникационных сетей от вредоносного программного обеспечения / Д.А. Даниленко // Збірник тез доповідей VIII наукової конференції «Новітні технології – для захисту повітряного простору». Харків. 18-19 квітня 2012 р. – м. Харків. ХУПС. – 2012. – С. 45.
13. Смирнов А.А. Исследование методов сигнатурного обнаружения вредоносного программного обеспечения в телекоммуникационных системах и сетях // Д.А. Даниленко // Збірник тез XIII міжнародного науково-практичного семінару «Комбінаторні конфігурації та їх застосування». м. Кіровоград. 13-14 квітня 2012 р. – Кіровоград: КНТУ. – 2012. – С. 43-45.
14. Смирнов А.А. Исследование методов проактивной защиты от вредоносного программного обеспечения в телекоммуникационных системах и сетях / Д.А. Даниленко // Збірник тез V міжнародної науково-практичної конференції «інтегровані інтелектуальні робототехнічні комплекси» (ІПРТК-2012). м. Київ. 15-16 травня 2012 р. – Київ: НАУ. – 2012. – С. 314-315.
15. Смирнов А.А. Метод обнаружения вредоносного программного обеспечения на основе корреляционного анализа сетевого трафика / Д.А. Даниленко // Матеріали XII всеукраїнської наукової інтернет-конференції «Наукові дослідження: зв'язок теорії і практики». м. Тернопіль. 29-30 квітня 2012 р. – Тернопіль: ТНЕУ. – 2012. – С. 9-10.
16. Смирнов А.А. Метод детектирования вредоносного трафика в телекоммуникационных сетях на основе использования bds-тестирования / Д.А. Даниленко // Збірник тез V міжнародної науково-практичної конференції «Комп'ютерні системи та мережні технології» (CSNT-2012). м. Київ. 13-15 червня 2012 р. – Київ: НАУ. – 2012. – С. 121.
17. Смирнов А.А. Обнаружение и предотвращение вторжений в компьютерных сетях на основе статистического анализа сетевого трафика / А.А. Смирнов, Д.А. Даниленко // Збірник тез доповідей науково-практичної конференції «Застосування інформаційних технологій у підготовці та діяльності сил охорони правопорядку». м. Харків. 12-13 березня 2014 р. – Харків. АВВ МВС. – 2014. – С. 13-14.
18. Смірнов О.А. дисперсійний аналіз мережного трафіку для забезпечення інформаційної безпеки телекомунікаційних систем та мереж / О.А. Смірнов, Д.О. Даниленко // Збірник тез V Всеукраїнської науково-практичної конференції "Інформатика та системні науки". м. Полтава. 13-15 березня 2014 р. – Полтава: ПУЕТ. – 2014. – С. 289-291.
19. Смирнов А.А. Метод дисперсионного анализа сетевого трафика для обнаружения и предотвращения вторжений в телекоммуникационных системах и сетях/ А.А. Смирнов, Д.А. Даниленко // Збірник тез VI міжнародної науково-практичної конференції "Проблеми і перспективи розвитку ІТ-індустрії". м. Харків. 17-18 квітня 2014 р. – Харків: ХНЕУ. – 2014. – С. 258.
20. Смірнов О.А. метод забезпечення інформаційної безпеки телекомунікаційних систем з використанням дисперсійного аналізу мережного трафіку / О.А. Смірнов, Д.О. Даниленко // Збірник тез міжнародної науково-практичної конференції «Інформаційна та економічна безпека» (INFECO-2014)». м. Харків. 15-16 травня 2014 р. – Харків: ХІБС УБС НБУ. – 2014. – С. 135-139.