17. Коваленко А.С. Створення систем технічної діагностики для автоматизації процесів керування в інтегрованих інформаційних системах / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Проблеми і перспективи розвитку ІТ-індустрії: VI між нар. наук.-практ. конф., 17-18 квіт. 2014 р., м. Харків: зб. тез. – Харків: ХНЕУ, 2014. – С. 241.
18. Коваленко А.С. Визначення понятійного апарату та напрямів досліджень для синтезу систем технічної діагностики інтегрованих інформаційних систем / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Комп'ютерне моделювання у наукоємних технологіях (КМНТ-2014): наук.-техн. конф. з міжнар. участю, 28-31 трав. 2014 р., м. Харків: зб. наук. праць. – Харків: ХНУ, 2014. – С. 190-193.
19. Коваленко А.С. Розробка структури бази даних інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Проблеми і перспективи розвитку ІТ-індустрії: VII міжнар. наук.-практ. конф., 17-18 квіт. 2015 р., м. Харків: зб. тез. – Харків: ХНЕУ, 2015. – С. 15.
20. Коваленко А.С. Дослідження елементів інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Комбінаторні конфігурації та їх застосування: XVII між нар. наук.-практ. сем., 17-18 квіт. 2015 р., м. Кіровоград: зб. тез – Кіровоград: КНТУ, 2015. – С. 5.

УДК 004

В.Ковальчук, магістр гр. КН-21М-1,4,
Центральноукраїнський національний технічний університет

ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ХМАРНИХ СЕРВІСІВ З ВИКОРИСТАННЯМ ЦСК

У статті розроблено програмне забезпечення, яке призначено для системи хмарних сервісів з використанням ЦСК. Метою розробки є дослідження та програмна реалізація системи хмарних сервісів з використанням ЦСК. Об'єктом дослідження є процес хмарних сервісів з використанням ЦСК. Предметом дослідження є методи хмарних сервісів з використанням ЦСК. Методи дослідження базуються на методах хмарних технологій та хмарних технологій та захисту інформації, методах математичної статистики, методах розробки програмного забезпечення. Результат роботи – програмна реалізація системи хмарних сервісів з використанням ЦСК. В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.
комп'ютерні науки, хмарні сервіси, центр сертифікації ключів

Постановка проблеми. Робота присвячена питанням забезпечення безпеки інформації в сервіс-орієнтованих хмарних архітектурах, за рахунок розробки центру сертифікації та розподілу ключів (ЦСК). При згадуванні абревіатури SOA (Service-Oriented Architecture) більшість ІТ-фахівців першою справою згадують Web-сервіси й протокол HTTP, які є складовими хмарних сервісів, хоча цей термін позначає набагато більше широке поняття.

Хмарна архітектура SOA зовсім незалежить від мов програмування, платформ або протокольних специфікацій, за допомогою яких сервіси розробляються, а також від того, де й за допомогою чого вони розгорнуті.

Практично хмарна архітектура SOA вимагає наявності не тільки сервісів, але й засобів, за допомогою яких ці сервіси можуть бути виявлені й підключені незалежно від нижчележачої інфраструктури. SOA – це не продукт або специфікація. Хмарна архітектура ретельно вибудовується – складається з множини компонентів, таких, як сервери додатків, що зв'язують ПЗ, репозиторій і навіть спеціалізовані пакети централізованого управління SOA.

Строго говорячи, SOA не можна відносити ні до нової реалізації CORBA, ні до оновленої хмарної архітектури RMI (Remote Method Invocation). Ключовий компонент SOA – сервіс. Сервіси тут є бізнес-функціями, призначеними для забезпечення погодженої роботи великих, що складаються з множини частин додатків.

По суті, це будівельні блоки для відбиття бізнес-логіки в розроблювальних додатках. А кінцевим місцем, де сервіси “живуть”, є сервер додатків, будь то WebLogic від BEA Systems, WebSphere від IBM, Application Server від Oracle або Java AS від Sun Microsystems.

Функції, або операції, в SOAP (Simple Object Access Protocol) повинні бути інтуїтивно зрозумілими й відповідати своїм назвам – наприклад, submitPurchaseOrder (“підтвердити замовлення на покупку”) або validateCustomerAccounts (“перевірити особовий рахунок замовника”).

На відміну від звичайних додатків сервіс в хмарній архітектурі SOA призначається для використання всім реалізованим бізнес-функціям. У той час як звичайні корпоративні додатки містять у собі схожі фрагменти бізнес-логіки або навіть дублюють окремі об'єкти – наприклад, об'єкт клієнтського замовлення, – в хмарній архітектурі SOA вам потрібно запустити лише єдиний екземпляр такої бізнес-функції.

Таким чином, можливо повторно використовувати функціональність у середовищі із множинними додатками й швидко коректувати бізнес-логіку, для того щоб мати можливість пристосовуватися до мінливих умов ринку. У цьому й складається головна перевага SOA.

Зі зміною єдиного екземпляра бізнес-функції в SOA автоматично вносяться корективи й в усі додатки, що опираються на цю функцію. Так що, наприклад, будь-які зміни в правилах ціноутворення або політики знижок застосовуються у всіх додатках.

Аналогічно будь-які зміни в підтримуючій інфраструктурі залишаються прозорими для всіх додатків, що використовують сервіси. Наприклад, якщо ви переходите з однієї версії бази даних на іншу, то будуть модифіковані лише пов'язані з нею сервіси, оскільки додатки в хмарній архітектурі SOA працюють із усіма інфраструктурними додатками тільки за допомогою сервісів. У недавнім минулому зміни в сполучному ПЗ або в базі даних змушували переробляти всю систему, включаючи клієнтські настільні додатки.

Аналіз останніх досліджень і публікацій. При аналізі останніх досліджень і публікацій [1-10] було виявлено певні прогалини у забезпеченні системи хмарних сервісів з використанням ЦСК.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи хмарних сервісів з використанням ЦСК.

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

– Огляд існуючих систем хмарних сервісів з використанням ЦСК.
– Дослідження системи хмарних сервісів з використанням ЦСК.
– Програмна реалізація системи хмарних сервісів з використанням ЦСК.

Об’єктом дослідження є процес хмарних сервісів з використанням ЦСК.

Предметом дослідження є методи хмарних сервісів з використанням ЦСК.

Методи дослідження базуються на методах хмарних технологій та захисту інформації, методах математичної статистики, методах розробки програмного забезпечення.

Виклад основного матеріалу. Опис технології WS Security

Надаючи вільно зв'язані сервіси, сервіс-орієнтована хмарна архітектура дозволяє гнучко реагувати на постійно мінливі ділові процеси. При цьому необхідно приділити увагу не тільки функціональним аспектам, але й створенню гнучкої інфраструктури безпеки, оскільки зміни ділових процесів роблять на неї серйозний вплив. Приміром, залучення нових ділових партнерів або включення конфіденційних відомостей у важливі корпоративні процеси вимагає адекватного стандартизованого рішення для забезпечення безпеки.

Як основна технологія забезпечення безпеки повідомлень на базі SOAP (Simple Object Access Protocol) міцно закріпився стандарт безпеки служб Web (Web Services Security, WS Security), ратифікований OASIS, організацією по розвитку стандартів структурованої інформації. WS Security складається із цілого пакета специфікацій і множини механізмів, які комбінуються відповідно до необхідного сценарію застосування.

До честі творців стандартів у рамках SOA вони приділили підвищену увагу безпеки при розробці цих стандартів. Механізми безпеки органічно вбудовуються в концепцію Web-сервісів і дозволяють не тільки уникнути основних проблем, але й істотно підвищити ефективність як механізмів захисту, так і засобів керування політикою безпеки.

Стандарти

Основний пул стандартів безпеки Web-сервісів розробляється в рамках консорціуму OASIS. Структуру специфікацій безпеки SOA можна зобразити у вигляді наступної ієрархічної конструкції (рисунок 1).

Розглянемо ці стандарти:

– Базові стандарти (SOAP Foundation) містять у собі специфікації XML Signature і XML Encryption, які визначають відповідно формати ЕЦП і шифрування SOAP-транзакцій. Дані специфікації ніяк не обмежують список алгоритмів шифрування й ЕЦП, що робить вбудовування українського ДСТУ в SOA-архітектуру неважким завданням. Також до базових понять можна віднести інформацію в складі SOAP-заголовка (security-token, маркер безпеки), використовувану для автентифікації й авторизації запиту. Наприклад, security-token може містити в собі сертифікат X.509 і/або ім'я/пароль. Одним з видів security-token є SAML (Security Assertion Markup Language), що включає в себе інформацію про статус автентифікації, авторизації й атрибутах учасників транзакції. Це дозволяє забезпечити побудову відносин довіри (trust) в SOA-хмарній архітектурі й виключити необхідність автентифікації/авторизації для кожного запиту.

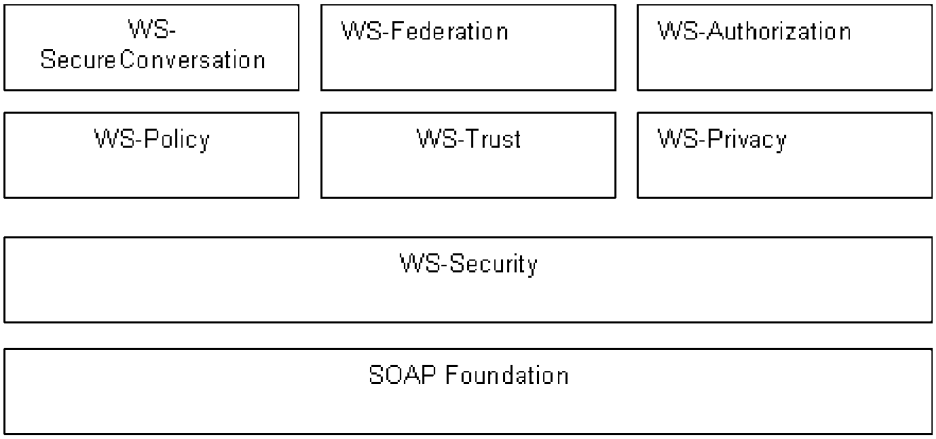

Рисунок 1 – Структура пула стандартів безпеки Web-сервісів

– WS-Security визначає базові механізми й формати використання security-token у складі SOAP-запитів. Основною метою WS-Security є абстрагування реалізації політик безпеки Web-сервісів від конкретних методів (наприклад, протоколів автентифікації й авторизації). За допомогою уточнюючих специфікацій, описаних нижче, WS-Security дозволяє досягти сумісності методів реалізації політик безпеки, описаних з використанням даних стандартів.

– WS-Policy визначає шаблони й правила опису політики безпеки для Web-сервісів.

– WS-Trust описує правила організації довірених відносин між учасниками Web-взаємодії.

– WS-Privacy визначає формати політики конфіденційності при обміні SOAP-Повідомленнями.

– WS-SecureConversation регламентує правила безпечного обміну повідомленнями в SOA-хмарній архітектурі.

– WS-Federation є специфікацією, що визначає встановлення довірених відносин між різними доменами безпеки.

– WS-Authorization описує формати опису правил розмежування доступу до Web-сервісів.

Отже, стає ясно, що безпека в SOA-хмарній архітектурі описується досить великим набором специфікацій. Втішно, однак, що даний набір є невід'ємною частиною пула стандартів SOA і розробляється одночасно з ним. Це дає підстави думати, що додатки в складі Web-сервісної хмарної архітектури можуть створюватися безпечними вже на стадії проектування.

Хмарна архітектура

Розглянемо тепер типову архітектуру безпеки Web-сервісів, що застосовується в більшості рішень корпоративного рівня.

Ключові завдання, покладені на таку архітектуру:

– Керування доступом до Web-сервісів і однократна автентифікація (Single Sign-on, SSO). Призначено для забезпечення однократної автентифікації, авторизації й аудита Web-сервісів.

– Централізоване керування політикою безпеки. Дозволяє мінімізувати необхідність дублювання зусиль для застосування політики безпеки для кожного Web-сервісу за допомогою використання централізованої інфраструктури безпеки, не вимагаючи при цьому переробки самих Web-сервісів.

– Уніфікація процесу моніторингу. Дозволяє проводити аудит роботи Web-сервісів, що показує, які користувачі (додатка) здійснювали доступ до Web-сервісів, які дії вони виконували і які дані при цьому передавали.

– Маршрутизація запитів до Web-служб. Дозволяє, аналізуючи вміст запиту, проводити його перетворення й перенапрямок до того або інший Web-сервісу.

Схема керування захистом в SOA-хмарній архітектурі

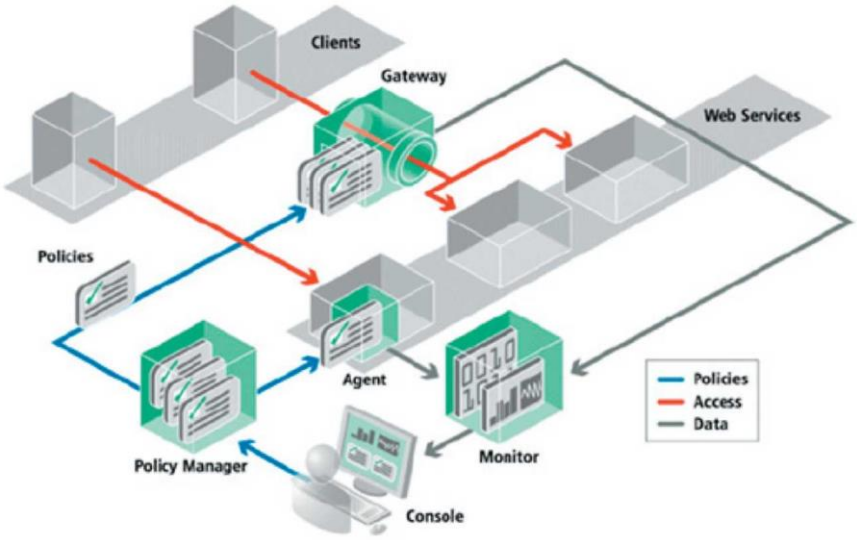

Рисунок 2 – Схема керування захистом в SOA-хмарній архітектурі

До складу такої схеми входять наступні компоненти:

– менеджер політик (Policy Manager);

– компоненти застосування політики: агенти (Agents) і шлюзи (Gateways);

– панель моніторингу (Monitor).

Менеджер політик – це графічний інструмент для визначення нових політик безпеки й експлуатації, зберігання політик, а також для керування поширенням і відновленням політик на агентах і шлюзах.

Компоненти застосування політик діляться на шлюзи (Policy Gateways) і агенти (Policy Agents). Шлюзи політик установлюються перед групою додатків або сервісів, перехоплюючи запити до цих додатків з метою застосування політик, підвищуючи безпеку вже встановлених додатків і додаючи в них нові правила. Агенти політик забезпечують

додатковий диференційований рівень безпеки й розміщаються на серверах додатків, що забезпечують виконання додатка або сервісу. Таким чином, забезпечується можливість автентифікації й авторизації запитів до Web-сервісів по наявним на підприємстві репозитаріям користувачів (наприклад, LDAP-каталог).

На панелі моніторингу адміністратор може задати рівні якості обслуговування для кожного додатка, визначити правила видачі попереджень і повідомлень, якщо додаток перевищить заданий рівень якості обслуговування.

Таким чином, архітектуру безпеки SOA можна побудувати без переробки безпосередньо Web-сервісів. Це одне з основних достоїнств наявності стандартів безпеки, що є частиною загального пула стандартів SOA.

Базові концепції

OASIS прийняла стандарт WS Security у березні 2004 р. як доповнення до протоколу SOAP. До теперішнього часу він визнаний цілком зрілим і придатним до застосування. WS Security не визначає ніяких нових технологій, а опирається на вже існуючі стандарти, приміром, XML Encryption, XML Signature, сертифікати X.509 або різні криптографічні алгоритми. Базова концепція ґрунтується на механізмах повідомлень, тому замість захисту, орієнтованої на транспорт, можливе забезпечення безпеки від краю до краю (End-to-End Security), приміром, за допомогою протоколу SSL. Такий підхід необхідний, щоб уникнути виникнення наскрізних комунікаційних структур у межах SOA, а також забезпечити передачу асинхронних повідомлень або використання проміжних станцій (приміром, сервісної шини підприємства – Enterprise Service Bus, ESB).

Основне завдання WS Security – забезпечення цілісності, конфіденційності й автентичності повідомлення і його відправника при одночасному збереженні відкритості для розширень. Основними елементами стандарту є наступні базові механізми (рисунок 3): токени безпеки, шифрування, підписи й оцінки про час.

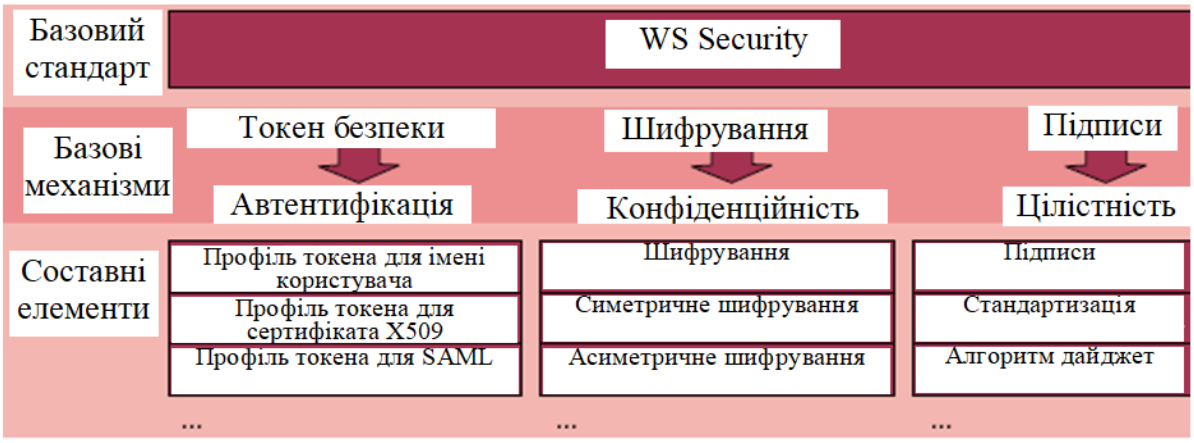

Рисунок 3 – Базові механізми WS Security

Токени безпеки (Security Token). Автентифікація відправника – базова передумова для забезпечення контролю доступу (Access Control) з боку сервісу, а крім того, вона необхідна для організації обліку й контролю. Підтвердження ідентифікації (Credentials), без яких неможлива автентифікація, передаються усередині повідомлення у вигляді токенів. Сама автентифікація не входить до складу WS Security – це самостійний процес провайдеру послуг. Для різних форматів токенів OASIS пропонує окремі специфікації у вигляді профілів WS Security. Так, «Профіль токена з ім'ям користувача» (Username Token Profile) регулює алгоритм широко розповсюдженого методу автентифікації користувача за допомогою ідентифікаційного номера (User ID) і відповідного пароля.

Ідентифікація додатків або ділових процесів звичайно здійснюється за допомогою сертифікатів, і в цьому випадку управляти паролями на стороні клієнта не потрібно. Обіг із сертифікатами для зазначеного методу автентифікації описується в профілі X.509 Certificate

Token Profile. Існують і інші профілі, приміром, для використання токенів мови розмітки тверджень безпеки (Security Assertion Markup Language, SAML) або Kerberos.

Двійкові або базовані на XML токени безпеки потрібні не тільки для автентифікації. Вони виконують ще одну функцію, являючи собою основу для транспорту або прив'язки ключів (Keys), застосовуваних у криптографії.

Шифрування. Щоб забезпечити захист конфіденційних даних, використовується криптографічне шифрування. Оскільки протокол SOAP базується на XML, то WS Security не визначає новий стандарт, а використовує специфікацію XML Encryption з W3C. Зашифровані дані і їхня метаінформація, у свою чергу, включаються в повідомлення у вигляді структур XML. Однак, відповідно до специфікації SOAP, не можна шифрувати елементи «конверт» (Envelope), «заголовок» (Header) і «тіло» (Body), оскільки вони задають структуру повідомлення й повинні бути читаємі завжди.

Принципово розрізняють два механізми шифрування: симетричне й асиметричне. При симетричному шифруванні (метод «секретного ключа» – Secret Key) для шифрування й дешифрування використовується загальний ключ, завжди доступним обом сторонам. При асиметричному шифруванні (алгоритм із відкритими ключами – Public Key) для шифрування й дешифрування застосовуються різні ключі, що істотно скорочує витрати зусиль на їхній розподіл: особистий ключ (Private Key) залишається у власника, а загальний ключ (Public Key) поширюється вільно. Однак у порівнянні із секретними ключами механізм відкритих ключів працює значно повільніше, тому обидва підходи часто поєднують, у результаті чого з'являються нові гібридні варіанти. Клієнт генерує симетричний ключ сеансу (Session Key) і використовує його для симетричного шифрування більших обсягів даних. На закінчення симетричний ключ шифрується за допомогою асиметричного алгоритму, вкладається в повідомлення й надається в розпорядження сервісу.

Підпис (Signature). Для підтвердження цілісності повідомлень застосовуються підписи. Вони дозволяють розпізнати неправомірні модифікації: зміна, видалення або додавання даних. Реалізація цього підходу в рамках WS Security опирається на стандарт XML Digital Signature від W3C. Принцип підписів заснований на створенні контрольних сум за допомогою спеціальних алгоритмів (дайджет). Результати приєднуються до повідомлення й передаються в частково зашифрованому виді. Сервісна сторона формує контрольну суму й порівнює неї зі значенням, присланим клієнтом. Оскільки в XML різні способи написання логічно ідентичні, перед формуванням контрольної суми необхідно зробити нормалізацію даних. Для цього використовуються стандартизовані алгоритми XML Canonicalization, також запозичені з W3C.

Крім того, підпису надають можливість установлення автентичності відправника. Цю інформацію можна використовувати в юридичних цілях для встановлення авторства.

Оцінка про час (Timestamp). Ідея послуг у рамках SOA має на увазі, що сервіси повинні робити визначену дію й у такий спосіб підтримувати взаємодію без обліку стану (Stateless). Однак даний принцип комунікації без установлення сеансу відкриває простір для атак скидання (Replay), що коли атакує повторно відправляє або повідомлення цілком, або окремі їхні частини. Щоб перешкодити таким атакам, необхідно гарантувати унікальність повідомлень, для чого кожне з них одержує свій ідентифікаційний номер (Message ID), що сервіс перевіряє на предмет його унікальності. Тому ідентифікаційні номери вже отриманих повідомлень необхідно зберігати. Термін дії, а виходить, і час зберігання окремих ідентифікаційних номерів повідомлень на стороні сервісу обмежується оцінкою, що втримується в повідомленні, про час.

Крім використання традиційної структури оцінок про час у заголовку безпеки (Security Header), токен з ім'ям користувача пропонує власне керування оцінками про час щоб уникнути несанкціонованого повторного використання даних для автентифікації. Ідентифікаційний номер повідомлення повинен відповідати специфікації WS Addressing. А токени з ім'ям користувача одержують випадкове криптографічне значення (Nonce).

У рамках SOA кожний зі згаданих чотирьох базових механізмів охоплює лише один аспект забезпечення безпеки. Сформувати цілісне рішення можна лише за умови взаємодії всіх компонентів. Цілком традиційна комбінація механізмів безпеки на основі повідомлень (WS Security), орієнтованих на транспорт (SSL). Сценарій, представлений на рисунку 4, докладно роз'ясняє необхідність використання комбінації різних механізмів.

Автентифікація. Будь-який контроль доступу на стороні сервісу припускає автентифікацію клієнта. Сервісній стороні необхідно мати відомості про підтвердження ідентифікації. У випадку методу з користувальницьким ідентифікаційним номером і паролем WS Security надає механізм токенів з ім'ям користувача, де пароль є конфіденційною інформацією, тому необхідно запобігти його зчитуванню в процесі транспорту. Шифрування необхідно, навіть якщо механізм, визначений у специфікації токенів з ім'ям користувача, припускає передачу пароля тільки у вигляді контрольної суми. При використанні контрольної суми, що читається, виникає погроза атаки методом підбора пароля (Brute Force) шляхом перевірки всіх можливих комбінацій, оскільки паролі обмежені по довжині й набору символів.

Крім того, у випадку застосування контрольної суми пароля сервісній стороні знадобиться пароль відкритим текстом. Тому даний підхід у багатьох випадках неприйнятне або адміністрування паролів зажадає додаткових заходів безпеки.

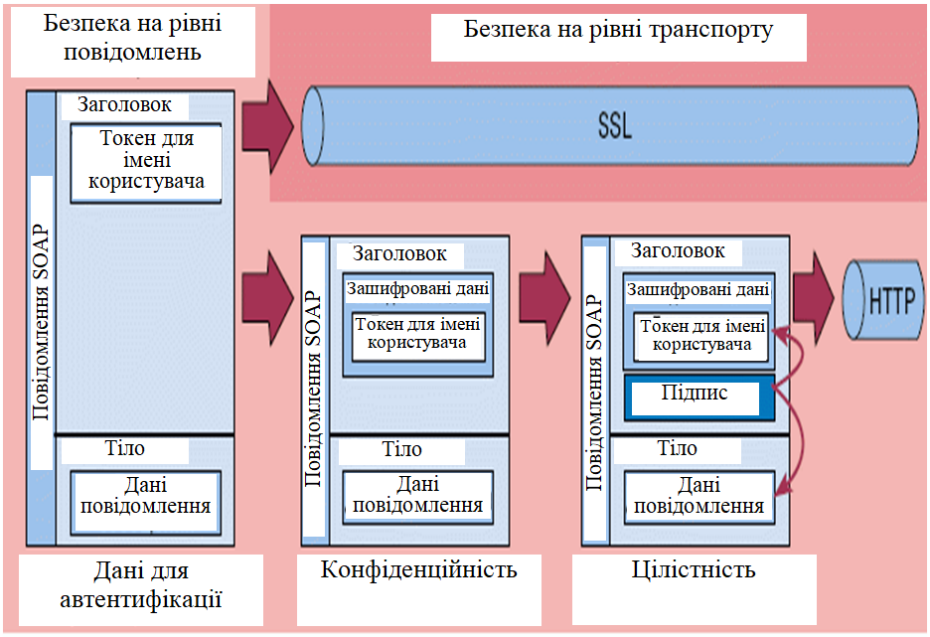

Рисунок 4 – Взаємодія токенів, шифрування й підписів WS Security

Конфіденційність. Запобігти розкраданню пароля під час пересилання повідомлення покликане шифрування токену з ім'ям користувача. У деяких випадках досить застосувати широко розповсюджений протокол SSL. Однак необхідно врахувати, що внаслідок принципу з'єднання двох точок, властивого SSL, використання проміжних вузлів, приміром, сервісної шини підприємства (Enterprise Service Bus, ESB), неможливо, і захист даних після їхньої передачі не забезпечується.

У той же час механізм шифрування WS Security надає метод на основі повідомлень: вихідні дані шифруються й заміняються за допомогою алгоритму шифрування XML. Додатково в повідомлення вкладається метаінформація, приміром, про використані алгоритми або ключі, і тепер воно може передаватися навіть за допомогою незахищених протоколів (приміром, HTTP), а конфіденційність даних не піддається погрозі.

Цілісність. У використаному як приклад сценарії відсутня зв'язок між токеном з ім'ям користувача, що перебуває в заголовку SOAP, і даними в тілі SOAP. У результаті виникає погроза підміни ключових елементів повідомлення. Зокрема, зашифрована інформація про

користувача, зазначена в заголовку, може бути постачена підробленим запитом сервісу. Однак ця проблема легко вирішується за допомогою підписів. Механізми підписів, використовувані в WS Security, «скріплюють» трохи пространственно розділених блоків даних, з яких складається повідомлення, що дозволяє перевірити цілісність усього повідомлення або окремих його частин. У контексті безпеки на базі повідомлень підпису виконують роль елементарних конструктивних компонентів і зачіпають не тільки тему цифрових підписів.

Унікальність повідомлення. Для того щоб запобігти повторному відправленню повідомлення (атака Replay), на сервісній стороні необхідно перевірити унікальність повідомлення. Для цього до повідомлення, представленому в стандартизованому виді, додається ідентифікаційний номер. Стандарт WS Addressing, визначений в W3C, передбачає, серед іншого, завдання ідентифікаційного номера повідомлення, що допомагає встановити його унікальність. Визначена в рамках WS Security структура вказує час створення повідомлення й закінчення строку його дії.

На закінчення потрібно відзначити, що ідентифікаційні номери повідомлень, як і оцінки про час, повинні бути прив'язані до існуючих блоків даних (інформація про користувачів і дані повідомлень). Для цього потрібно розширити діапазон охвату підпису, що дозволяє включити нові елементи при контролі цілісності.

Підписи і їхні завдання

Завдяки своїй інфраструктурі на основі повідомлень, сервіси Web підтримують можливість включення будь-яких проміжних інстанцій (Intermediaries) між кінцевими точками. С допомогою такої хмарної архітектури можна розширити функціональність сервісів Web. Крім того, ця хмарна архітектура стає основою для організації поділу відповідальності за реалізацію властивостей сервісу, особливо вимог, не пов'язаних з функціональністю (якість сервісу – Quality of Services, QoS). При виклику сервісу повідомлення із запитом і відповіддю повинні пройти через проміжні інстанції, причому кожна витягає з повідомлення дані, необхідні для виконання її завдань, і, якщо знадобиться, постачає його додатковою інформацією. Відповідно, необхідно, щоб визначені частини повідомлень були придатні для читання й зміни проміжними інстанціями. Так, за допомогою даних WS Addressing з повідомлення можна управляти функціями маршрутизації в межах ESB.

Для забезпечення конфіденційності й цілісності повідомлення, з одного боку, і читаності й розширюваності, з іншої, до механізмів безпеки пред'являються підвищені вимоги. Приміром, шифрування всього повідомлення за допомогою протоколу SSL перешкоджало б гнучкому використанню проміжних інстанцій. Крім того, стандарт SOAP вимагає, щоб конверт, заголовок і тіло повідомлення представлялися в незашифрованому виді.

Підписи виконують кілька важливих завдань для забезпечення всебічної безпеки в рамках такої вільно зв'язаної хмарної архітектури. Крім загальновідомої ролі цифрового підпису, вони надають механізми для перевірки цілісності й автентичності частин повідомлення. Через основну роль підписів необхідно бути в курсі їхніх базових принципів.

Цілісність даних

Підпису, крім іншого, дозволяють перевірити цілісність окремих блоків даних і розпізнати маніпуляції з повідомленнями (зміна, видалення й додавання даних). Для цього за допомогою спеціального криптографічного алгоритму створення гешу (приміром, SHA1, MD5) розраховуються контрольні суми для важливих блоків даних (Message Digest). Геш-алгоритми – необоротні функції, і відновлення вихідних даних за відомим значенням гешу неможливо. Крім того, його величина для різних даних не повинна збігатися (відсутність колізій). Для перевірки геша приймаюча сторона ще раз розраховує контрольну суму й порівнює отриманий результат із присланим. Якщо обоє значення ідентичні, то цілісність даних дотримана, тоді як в інших випадках велика ймовірність змін повідомлення. Однак цей

механізм не дозволяє встановити, які саме зміни внесені, оскільки підпису повідомляють тільки про вірний або невірний результат.

Принцип підписів базується на формуванні контрольних сум обома сторонами (відправником і одержувачем), тому однакова форма подання даних є обов'язковою умовою. Приміром, для того щоб різне написання XML не привело до різниці контрольних сум, варто ввести проміжний етап – нормалізацію XML (Canonicalization).

Однак одного доказу цілісності окремих блоків даних недостатньо для підтвердження автентичності всього повідомлення – потрібно забезпечити єдність окремих блоків. Для цієї мети створюється загальна контрольна сума шляхом об'єднання значення гешу для всіх блоків даних. У результаті здійснюється криптографічний зв'язок блоків, що не залежить від їхнього положення усередині повідомлення: таким чином, загальна контрольна сума дає можливість перевірити автентичність усього повідомлення. Крім того, так можна розпізнати маніпуляцію зі значеннями гешу окремих блоків даних. У процесі транспорту повідомлення загальна контрольна сума захищається від зміни за допомогою криптографічного механізму шифрування. Для реалізації автентичності повідомлення можна застосовувати симетричне шифрування (приміром, HMAC). При цьому ключ, спільно використовуваний відправником і одержувачем, або передається заздалегідь, або створюється відправником у момент передачі, а потім відправляється в зашифрованому виді разом з повідомленням.

Крім перевірки цілісності даних і автентичності повідомлень, підпису надають можливість автентифікації відправника всього повідомлення або його частин (рисунок 5).

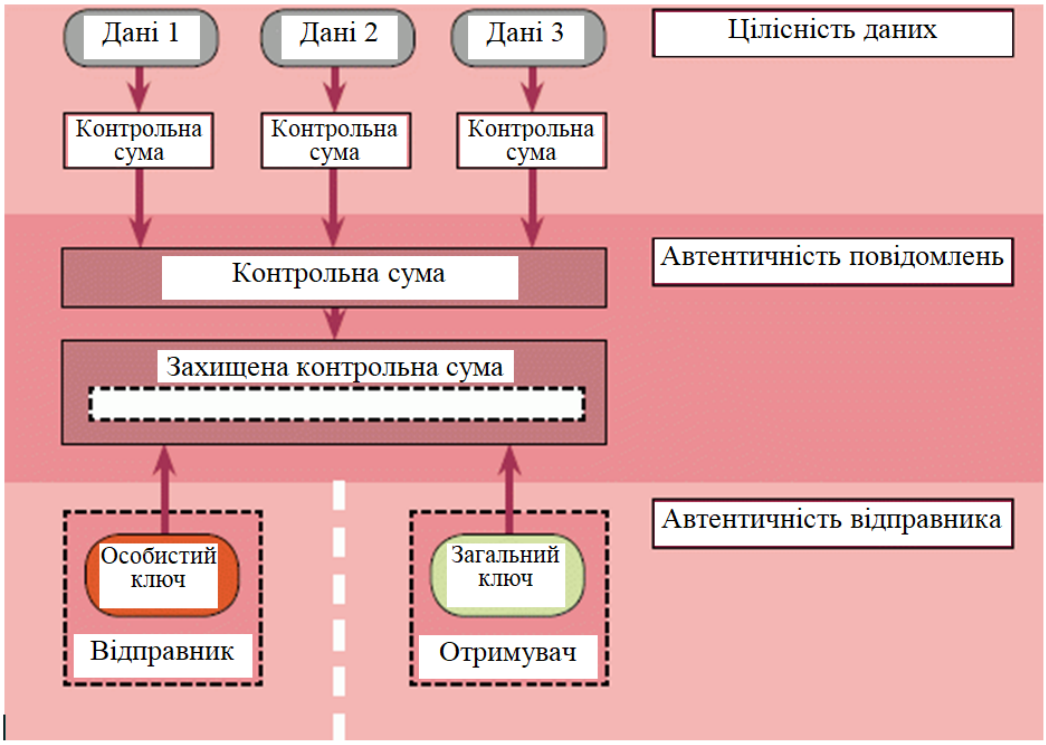

Рисунок 5 – Перевірка цілісності даних і автентичності повідомлень, а також автентифікації відправника всього повідомлення або його частин

Це властивість відомо як цифровий підпис. У цьому випадку застосовується не симетричний алгоритм шифрування загальної контрольної суми, а асиметричний алгоритм із застосуванням відкритих ключів: відправник використовує власний ключ для шифрування значення гешу, а прочитати його можна лише за допомогою відкритого ключа, сертифікат якого надає інформацію про власника особистого ключа. Якщо сертифікат, а виходить, і відкритий ключ, викликає довіру, то з його допомогою можна визначити відправника повідомлення.

На перший погляд, набір стандартів і реалізація хмарної архітектури безпеки в SOA здаються нетривіальними. Але його переваги переважують всі складності опису й

реалізації. До них відносяться:

– Відділення політики безпеки сервісів від самих сервісів дозволяє побудувати універсальні сервіси захисту для всіх бізнес-додатків без необхідності втручання в бізнес-логікові й "прошивання" функцій безпеки в код бізнесів-додатків.

– Чітке розмежування експертизи. Розроблювачі сервісів формують бізнес-логіку, архітектори й адміністратори визначають політику безпеки й керування.

– Єдина точка керування політикою ІБ.

– Зниження витрат на адміністрування, оскільки зміни в політику безпеки вносяться централізовано, а не в кожному Web-сервісі. Крім того, аудит безпеки для всіх сервісів ведеться з єдиної точки адміністрування.

– Спрощення підтримки й внесення змін у середовище керування й забезпечення безпеки Web-сервісів за рахунок використання єдиних сервісів безпеки для всіх Web-сервісних додатків.

Визначенено, що такий значний набір переваг SOA з погляду безпеки послужить достатнім стимулом для співробітників підрозділів ІБ підтримати зусилля своїх колег з IT-підрозділів по побудові Web-сервісної хмарної архітектури.

Розробка структурної схеми

Одним з найважливіших завдань забезпечення інформаційної безпеки в сервіс-орієнтованих хмарних архітектурах (SOA) є захист потоків корпоративних даних, переданих по каналах загального користування, у тому числі й через Internet. Перспективним методом надійного захисту інформації є метод кодування даних.

Для рішення цього завдання необхідно здійснити кодування інформації на виході з локальної мережі й декодування вхідних у неї даних. Ці функції реалізуються спеціальними програмними або програмно-апаратними засобами. Якщо захист сегмента корпоративної мережі вже забезпечений міжмережевим екраном, природно покласти на нього також виконання функцій кодування й декодування.

Для реалізації можливостей кодування/декодування повинне бути виконане попередній (початковий) розподіл ключів. Сучасні технології пропонують для цього цілий ряд методів. Після сертифікації та розподілу ключів з'являється можливість здійснення процесу виробітку спільних секретних ключів, що обслуговують сеанс спілкування абонентів.

У результаті кодування весь обмін даними між територіально-віддаленими локальними мережами є захищеним і для користувачів виглядає як обмін усередині однієї локальної мережі, при цьому від користувачів не потрібно застосування яких-небудь додаткових захисних засобів.

Комплекс кодування міжмережевих потоків

Програмний комплекс кодування міжмережевих потоків (ККМП) реалізує функції кодування міжмережевих інформаційних потоків у мережах передачі даних протоколу TCP/IP для забезпечення обміну інформацією між територіально-віддаленими локальними мережами. Це забезпечується за допомогою організації віртуальних захищених мереж (Virtual Private Networks – VPN).

Комплекс виконує наступні функції:

– **Кодування міжмережевих потоків.** Функції кодування міжмережевих інформаційних потоків у відкритих мережах передачі даних виконуються шляхом організації VPN. Кожна мережа в складі VPN захищена своїм модулем, що кодує, установлюваним у точці її з'єднання із зовнішніми мережами. Інформація, що захищається, кодується на передавальному модулі й декодується на приймаючому, тобто передається у відкритому виді в межах локальних мереж і в кодованому – за їхніми межами. Кодований трафік передається по протоколу IPsec.

– Створення контуру безпеки. Розроблена система дозволяє сформувати контур безпеки, що поєднує IP-адреси всіх абонентів, що мають доступ у віртуальну захищену

мережу. Абонентами VPN можуть бути цілі мережі, підмережі й окремі робітники станції. Крім того, що кодує модуль може бути встановлений на окрему робочу станцію.

– Вибіркове кодування трафіку. Формування контуру безпеки служить для поділу трафіку на кодуємий і некодуємий потоки.

– Модуль, що кодує. Розроблена система робить виділення пакетів, які необхідно кодувати, на підставі IP-адрес відправника пакета й одержувача пакета й, крім того, перевірки інтерфейсу, через який проходить пакет.

– Управління ключовою системою. У розробленій системі реалізована несиметрична ключова система, коли потенційні учасники обміну даними використовують пари довгострокових секретних й відкритих ключів кодування. Кодування здійснюється на основі сеансових ключів, автоматично сформованих за допомогою довгострокових ключів і що мають обмежений час життя. Комплекс здійснює всі необхідні дії по управлінню ключами: генерацію й розподіл довгострокових ключів, виробіток сеансових ключів абонентів, сертифікацію відкритих ключів у довіреному центрі, планову й позаштатну зміну ключів кодування.

– Реєстрація подій, моніторинг і управління міжмережевими потоками. Розроблена система здійснює збір і зберігання статистичної й службової інформації про всі штатні й позаштатні події, що виникають при автентифікації вузлів, передачі кодованої інформації, обмеженні доступу абонентів ЛОМ. Засоби моніторингу проводять збір і аналіз протоколів реєстрації від всіх модулів комплексу по кодованому каналі.

– Захист з'єднань із мобільними клієнтами. До складу віртуальної захищеної мережі можуть входити мобільні користувачі – віддалені комп'ютери, що підключаються по виділеним або каналам зв'язку, що комутуються. Основною відмінністю Мобільного клієнта є динамічно-призначувана IP-адреса. Носієм ключової інформації для них є електронний ключ eToken.

Состав Комплексу

Комплекс складається з наступних компонентів:

1. Набір шлюзів кодування.
2. Центр генерації ключів.
3. Центр сертифікації та розподілу ключів.
4. Центр реєстрації мобільних клієнтів.
5. Центр підготовки електронних ключів мобільних клієнтів.
6. Мобільний клієнт.
7. Центр моніторингу.
8. Програма контролю цілісності.

Шлюз із модулем, що кодує/декодувальним

Шлюз є основним модулем комплексу, що виконує функції маршрутизації, фільтрації й кодування пакетів. Кожний Шлюз призначений для закриття визначеної групи локальних мереж. На комп'ютері-шлюзі встановлюється ядерний модуль с функціями кодування й декодування й запускається програма автентифікації. Функціями шлюзу є:

- Фільтрація трафіку (розподіл на кодуємий/некодуємий потоки).
- Кодування трафіку (кодуємий потік).
- Автентифікація з іншими Шлюзами.
- Реєстрація подій у Центрі моніторингу.
- Забезпечення власного захисту.

Центр сертифікації та розподілу ключів

Центр сертифікації та розподілу ключів здійснює управління контуром безпеки, а також виконує наступні функції:

- Одержання зі змінного носія відкритих ключів Шлюзів.

– Видачу будь-якому Шлюзу відкритих ключів будь-яких інших Шлюзів і інформації про відповідні сегменти структури мережі.

– Розсилання Шлюзам повідомлень про зміни структури закритої мережі.

– Вироблення і виконання процедури зміни сеансових ключів.

– Зберігання інформації про структуру мережі.

Центр реалізований у вигляді програмного комплексу, що виконує функції зберігання й видачі відкритих ключів кодування по мережевому запиті від модулів кодування. Центр сертифікації та розподілу ключів може бути встановлений або на окремому (виділеному) комп'ютері, або разом з одним зі Шлюзів кодування.

Центр генерації ключів

Даний модуль служить для генерації пар комплементарних ключів, а також є репозитарієм всіх відомих системі ключів. У функції Центра генерації ключів входить:

– генерація пар відкритого й секретного ключів модулів, що кодують;

– генерація пари ключів для сертифікації (еталонного завірення) відкритих ключів модулів, що кодують;

– генерація сертифікатів відкритих ключів, підписаних секретним ключем сертифікації;

– приміщення підписаних сертифікатів відкритих ключів на змінні носії;

– зберігання еталонних копій сертифікованих відкритих ключів в архіві.

Центр генерації ключів – програма, що виконується на ізольованому автоматизованому робочому місці.

Рисунок 6 – Структурна схема системи

Центр реєстрації ключів

Центр реєстрації ключів служить репозитарієм всіх відомих системі ключів. У його функції входить:

– Введення зі змінного носія відкритого ключа.

– Введення зі змінного носія закритого ключа Адміністратора безпеки.

– Підпис нового ключа ключем Адміністратора безпеки.

– Приміщення підписаного відкритого ключа в архів довгострокового зберігання й на змінний носій.

– Зберігання еталонних копій сертифікованих (зареєстрованих) відкритих ключів.

Центр реєстрації ключів виконаний у вигляді програми, що виконується на ізольованому автоматизованому робочому місці й призначеної для сертифікації (еталонного завірення) відкритих ключів.

Центр реєстрації мобільних клієнтів і Мобільний клієнт

Для забезпечення доступу до корпоративних даних, які захищаються, мобільних абонентів, не підключених до локальних мереж, які захищаються, використовується Центр реєстрації мобільних клієнтів і програмне забезпечення мобільного клієнта комплексу.

Центр реєстрації мобільних клієнтів являє собою спеціальний модуль, що кодує, для підключення довільної кількості мобільних клієнтів.

Мобільний клієнт являє собою програмний модуль, що працює під управлінням ОС Windows і використовуючий апаратні ключі для автентифікації абонента в VPN.

Центр моніторингу

Центр моніторингу являє собою мережеве автоматизоване робоче місце із установленим на ньому набором програм, що здійснюють збір і аналіз протоколів, що надходять від всіх модулів комплексу.

Програма контролю цілісності

Комплекс містить у собі засобу формування й перевірки контрольних сум файлів. Ці засоби реалізовані у вигляді Програми контролю цілісності, що призначена для визначення й повідомлення системного Адміністратора про зміну, додавання й видалення файлів.

Адміністрування комплексу

Настроювання й адміністрування компонентів комплексу здійснюється централізовано з робочого місця Адміністратора безпеки за допомогою графічного інтерфейсу або командного рядка. Віддалене управління здійснюється по захищеному каналі. Комплекс забезпечує автентифікацію Адміністраторів і розмежування доступу до функцій адміністрування.

Основні особливості комплексу

– Основними особливостями розробленої системи є: Повнофункціональна схема управління ключами, що дозволяє здійснювати динамічний розподіл ключів з використанням довіреного центра сертифікації, перевірку дійсності ключової інформації й оповіщення систем кодування про компрометацію ключів; Висока надійність функціонування, забезпечувана засобами контролю цілісності, протоколювання й аудита, стійкості до збоїв і відновлення у випадку збоїв і відмов; Прозорість кодування переданих даних для абонентів і використовуваного ними програмного забезпечення; Висока продуктивність (робота в мережі 100 Мбіт/с без істотного впливу на пропускну здатність); Забезпечення необхідної якості сервісу (QoS) і підтримка роботи із сервісами, що пред'являють високі вимоги до величин тимчасових затримок (IP-телефонія, відеоконференцзв’язок); Можливість використання в комплексі з міжмережевими екранами, антивірусними рішеннями й засобами контекстного аналізу; Використання відкритих стандартів – протокол тунелювання мережевих пакетів відповідає стандартам IETF IPsec.

Висновки. У статті наведені теоретичне узагальнення й рішення наукового завдання дослідження методів хмарних сервісів з використанням ЦСК.

Рішення даного завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем хмарних сервісів з використанням ЦСК.

– Досліджена система хмарних сервісів з використанням ЦСК.

– На основі отриманих результатів досліджень створена програмна реалізація системи хмарних сервісів з використанням ЦСК.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання хмарних сервісів з використанням ЦСК.

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.

Розроблене програмне забезпечення має простий, дружній та зручний інтерфейс користувача, що забезпечує легкість у освоєнні роботи програмного продукту, зручність у використанні, і не потребує особливих спеціальних знань.

Список літератури

1. Smirnov, O., Kuznetsov, A., Potii, O., Poluyanenko, N., Stelnyk, I., Mialkovsky, D. «Combining and filtering functions in the framework of nonlinear-feedback shift register». International Journal of Computing; 2020, Volume 19, Issue 2 – Research Institute for Intelligent Computer Systems – 2020. – P. 247-256. (Scopus).
2. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». CEUR Workshop Proceedings. Volume 2740, 2020, Pages 102-114. (Scopus).
3. Smirnov O.A., Alimseitova Zh., Adranova A., Akhmetov B., Lakhno V., Zhilkishbayeva G. «Models and algorithms for ensuring functional stability and cybersecurity of virtual cloud resources». Journal of theoretical and applied information technology Vol.98. No 21, 2020, P. 3334-3346. (Scopus).
4. Smirnov O., Kuznetsov A., Arischenko A., Chepurko I., Onikiychuk A., Kuznetsova T. «Pseudorandom sequences for spread spectrum image steganography». CEUR Workshop Proceedings Volume 2654, 2020, Pages 122-131. (Scopus).
5. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New technique for data hiding in cover images using adaptively generated pseudorandom sequences». CEUR Workshop Proceedings Volume 2654, 2020, Pages 1-14. (Scopus).
6. Smirnov O., Lutsenko M., Kuznetsov A., Kiian A., Kuznetsova T., «Biometric cryptosystems: overview, state-of-the-art and perspective directions». Lecture Notes in Networks and Systems, vol 152. Springer, Cham. 2021, pp 66-84. (Scopus).
7. Smirnov O., Kuznetsov A., Pushkar'ov A., Serhiienko R., Babenko V., Kuznetsova T., «Representation of Cascade Codes in the Frequency Domain». In: Radivilova T., Ageyev D., Kryvinska N. (eds) Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies, vol 48. Springer, Cham. 2021. pp 557-587. (Scopus).
8. Smirnov, O., Markovets, O. Vovk, N., Turchyn, Y., «Model of informational support for social network administrators' content creation». CEUR Workshop Proceedings Volume 2616, 2020, Pages 125-136. (Scopus).
9. Smirnov, O., Drieieva, H., Drieiev, O., Polishchuk, Y., Brzhanov, R., Aleksander, M. «Method of fractal traffic generation by a model of generator on the graph». CEUR Workshop Proceedings Volume 2616, 2020, Pages 366-379. (Scopus).
10. Smirnov, O., Shekhanin, K., Kuznetsov, A., Krasnobayev, V. «Detecting Hidden Information in FAT». International Journal of Computer Network and Information Security (IJCNIS). Vol. 12, No. 3, 2020. PP.33-43. (Scopus).
11. Smirnov, O., Drieieva, H., Drieiev, O., Simakhin, V., Bondar, S., Odarchenko, R. «Managing multifractal properties of the binary sequence generated with the Markov chains», CEUR Workshop Proceedings Volume 2608, 2020, Pages 633-645. (Scopus).
12. Smirnov, O., Kuznetsov, A., Gorbacheva, L., Babenko, V., «Hiding data in images using a pseudo-random sequence», CEUR Workshop Proceedings Volume 2608, 2020, Pages 646-660., (Scopus).
13. Smirnov, O., Kuznetsov, A., Kolovanova, I., Kuznetsova, T., «Noise immunity of the algebraic geometric codes». International Journal of Computing; 2019, Volume 18, Issue 4 – Research Institute for Intelligent Computer Systems – 2019. – P. 393-407. (Scopus).
14. Smirnov, O., Ulichev, O., Meleshko, Y., Khokh, V., Goncharenko, I. «Method of Choosing Objects for Informational Influence in Social Networks during Information Campaign Based on the Analytic Hierarchy Process». CEUR Workshop Proceedings, Vol 2588, P. 215-227, 2019. (Scopus).
15. Smirnov, O., Krasnobayev, V., Yanko, A., Kuznetsova, T. «Methods of nulling numbers in the system of residual classes». CEUR Workshop Proceedings, Vol 2588, P. 90-106, 2019. (Scopus).
16. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Pastukhov, M., Kuznetsova, K., Prokopovych-Tkachenko, D., «Discrete Signals with Special Correlation Properties», CEUR Workshop Proceedings Volume 2353, CEUR Workshop Proceedings 2019, Pages 618-629. (Scopus).
17. Smirnov, O., Kuznetsov, A., Kiian, A., Kuznetsova, K., Ivko, T., Prokopovych-Tkachenko, D., «Soft Decoding Based on Ordered Subsets of Verification Equations of Turbo-Productive Codes», CEUR Workshop Proceedings Volume 2353, CEUR Workshop Proceedings 2019, Pages 873-884. (Scopus).
18. Smirnov, O., Kuznetsov, A., Prokopovych-Tkachenko, D. «Hiding Data in Images Using a Pseudo-Random Sequence». ISCI'2020: Information Security in Critical Infrastructures. Collective monograph. Edited by Ivan D. Gorbenko, Victor A. Krasnobayev and Alexandr A. Kuznetsov. ASC Academic Publishing, USA, 2020. pp. 46-59. – ISBN: 978-1-7362833-0-1 (Hardback), ISBN: 978-1-7362833-1-8 (Ebook).

19. Smirnov, O., Kuznetsov, A., Shekhanin, K., Chepurko, I. Detecting Hidden Information in FAT. Монографія: In.: ISCI'2019: Information Security in Critical Infrastructures. Collective monograph. Edited by Ivan D. Gorbenko and Alexandr A. Kuznetsov, ASC Academic Publishing, USA, 2019, pp. 412-429. – ISBN: 978-0-9989826-8-7 (Hardback), ISBN: 978-0-9989826-9-4 (Ebook).
20. Smirnov, O., Kuznetsov, A., Kuznetsova., K. Synthesis of Discrete Signals with Improved Correlation Properties. Монографія: In.: ISCI'2019: Information Security in Critical Infrastructures. Collective monograph. Edited by Ivan D. Gorbenko and Alexandr A. Kuznetsov, ASC Academic Publishing, USA, 2019, pp. 281-299. – ISBN: 978-0-9989826-8-7 (Hardback), ISBN: 978-0-9989826-9-4 (Ebook).

УДК 004

Р.Ковтуненко, магістр гр. КН-21М-1,4,
Центральноукраїнський національний технічний університет

ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ХМАРНОГО СЕРВІСУ ЕЛЕКТРОННОЇ БІБЛІОТЕКИ У НАВЧАЛЬНОМУ ЗАКЛАДІ

У статті програмне забезпечення, яке призначено для системи хмарного сервісу електронної бібліотеки у навчальному закладі. Метою розробки є дослідження та програмна реалізація системи хмарного сервісу електронної бібліотеки у навчальному закладі. Об'єктом дослідження є процес хмарного сервісу електронної бібліотеки у навчальному закладі. Предметом дослідження є методи хмарного сервісу електронної бібліотеки у навчальному закладі. Методи дослідження базуються на методах хмарних технологій, методах математичної статистики, методах розробки програмного забезпечення. Результат роботи – програмна реалізація системи хмарного сервісу електронної бібліотеки у навчальному закладі. В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.
комп'ютерні науки, хмарний сервіс, електрона бібліотека

Постановка проблеми. Темпи створення й нагромадження інформації, ускладнення й глобалізація знання зробили необхідним пошук інструментів, що дозволяють забезпечити швидкий і ефективний доступ до цього знання, незалежно від різних країн і різних сховищ інформації. Одним з таких інструментів по праву вважаються технології електронних бібліотек (ЕБ), розвиток яких почалося на початку 90-х років XX століття. Електронні бібліотеки як напрямок розвитку електронних ресурсів багато в чому визначає політику бібліотек при плануванні своєї діяльності у формуванні сучасного автоматизованого бібліотечно-інформаційного середовища.

У цей час активно розвиваються електронні бібліотеки ЗВО різних профілів, що у свою чергу створює об'єктивні передумови для підвищення рівня утворення. Ресурси, розміщені в електронних бібліотеках ЗВО, здатні істотно вплинути на інтенсивність процесів навчання й наукових досліджень, а «забезпечення публічного (у тому числі віддаленого) доступу до них стало однієї з першочергових задач обслуговування утворення, науки й культури. Сьогодні загальновизнано, що рішення цієї задачі найбільше ефективно досягається шляхом створення електронних бібліотек» [1]. У цьому змісті університетське середовище є найбільш оптимальним для використання існуючих, створення нових інформаційних ресурсів, розвитку нових інформаційних і комунікаційних технологій, тому що саме у ЗВО одночасно й у різних формах, у навчанні й наукових дослідженнях створюються й використовуються такі інформаційні ресурси й технології. Створювані в університетах інформаційні ресурси мають різну природу – це наукові видання й навчально-методичні посібники, дисертації й автореферати, бібліографічні покажчики й огляди, довідкова література, матеріали теле– і відеоконференцій, електронні журнали й електронні