ЗМІСТ

ВСТУП

Актуальність даної роботи визначається необхідністю у збереженні персональних даних користувача та відсутністю на ринку програмних продуктів які призначених для захисту інформації від її доступу третій стороні при використанні хмарних рішень збереження даних. Для їх захисту великі провайдери, такі як Amazon та Azure, дають змогу користувачам шифрувати інформацію яку вони зберігають на сервісах, проте таке шифрування відбувається вже на стороні провайдера і не може забезпечити збереження інформації при деяких з вище перерахованих сценаріях.

Ринок програмного забезпечення мав до недавнього часу досить простий вектор розвитку. Програмісти розробляли додаток, який потім розповсюджувався традиційним чином – на носіях – і встановлювався на комп’ютер. Щоб програма працювала до ПК встановлювались певні системні вимоги: мінімальний об’єм пам’яті, продуктивності процесора, кількість вільного простору на жорсткому диску, та інші. Паралельно з цим розвивався Інтернет – сервісне обладнання, що обслуговувало роботу сайтів також вдосконалювалось. Але в певний момент виявилось що можна об’єднати обчислювальні можливості для підтримки програмних сервісів, аналогічно тим, які задіюються звичайним користувачем, наприклад – текстові або табличні процесори, а розвиток технологій віртуалізації призвів до появи доступних через доступних з будь-якого місця обчислювальних ресурсів, що отримали назву "хмари".

Розповсюдження мереж з високою потужністю, низька вартість комп'ютерів і пристроїв зберігання даних, а також широке впровадження віртуалізації, сервіс-орієнтованої архітектури привели до величезного зростання хмарних обчислень. Кінцеві користувачі можуть не перейматися роботою обладнання технологічної інфраструктури "в хмарі", яка їх підтримує. Аналогією обчислювальних "хмар" зі звичного життя можуть служити електростанції. Хоча домовласник може купити електрогенератор і піклуватися про його справність самостійно, більшість людей вважає за краще отримувати енергію від централізованих постачальників.

Хмарні обчислення — це модель забезпечення зручного доступу на вимогу через мережу до обчислювальних ресурсів, які можуть бути оперативно надані та звільнені з мінімальними управлінськими затратами та зверненнями до провайдера.

Кінцеві користувачі мають доступ до власних даних і можуть не перейматися роботою обладнання технологічної інфраструктури (а в залежності від моделі обслуговування і програмним забезпеченням) "хмари", яка їх підтримує.

Проте в цьому проявляється головний недолік хмари – приватна інформація користувача фактично стає доступна третій стороні – провайдеру, крім цього данні можуть стати вразливими під час їх передачі по каналам зв'язку.

Метою даної роботи є покращення ефективності захисту приватної інформації користувача при використанні хмарних технологіях комп’ютерних обчислень на основі аналізу існуючих хмарних рішень та моделі їх представлення користувачам.

Об'єкт досліджень – процес обробки даних засобами хмарних технологій.

Предмет досліджень – методи та засоби захисту даних при застосуванні хмарних рішень.

Тому задачами даної роботи є розробка програмного забезпечення що допоможе захистити персональні дані користувача при використанні хмарних технологій комп’ютерних обчислень. Новизною даної роботи є запрпонований метод захисту інформації шляхом її шифрування саме перед відправленням її на сервер на стороні клієнта. Практичною цінністю роботи є створена програма-клієнт для сервісу DropBox, що шифрує файли перед їх відправленням на сервіс. Дана робота була апробована у доповіді на XLII науково-технічній конференції ВНТУ у 2014 році.

1 ЗАГАЛЬНІ ВІДОМОСТІ

1.1 Основні характеристики хмарних обчислень

Хмарні обчислення (англ. Cloud Computing) – це модель забезпечення повсюдного та зручного доступу на вимогу через мережу до спільного пулу обчислювальних ресурсів, що підлягають налаштуванню (наприклад, до комунікаційних мереж, серверів, засобів збереження даних, прикладних програм та сервісів), і які можуть бути оперативно надані та звільнені з мінімальними управлінськими затратами та зверненнями до провайдера.[1]

При використанні хмарних обчислень програмне забезпечення надається користувачеві як Інтернет-сервіс. Комп’ютери, що надають необхідні сервіси, називаються "обчислювальною хмарою". При цьому навантаження між комп’ютерами, що входять до "обчислювальної хмари", розподіляється автоматично (фактично, cloud computing – це повернення але вже на новому рівні стратегії розвитку, до епохи мейнфреймів – гігантських суперкомп’ютерів).

Користувач має доступ до власних даних, але не може управляти і не повинен піклуватися про інфраструктуру, операційну систему і програмне забезпечення, з яким він працює. «Хмарою» метафорично називають Інтернет, який приховує всі технічні деталі. Згідно з документом IEEE, опублікованим у 2008 році, «Хмарні обчислення – це парадигма, в рамках якої інформація постійно зберігається на серверах у мережі інтернет і тимчасово кешується на клієнтській стороні, наприклад на персональних комп'ютерах, ігрових приставках, ноутбуках, смартфонах тощо»

Провайдери хмарних рішень дозволяють орендувати через інтернет обчислювальні потужності та дисковий простір. Переваги такого підходу – доступність (користувач платить лише за ті ресурси, які йому потрібні) і можливість гнучкого масштабування. Клієнти позбавляються від необхідності створювати і підтримувати власну обчислювальну інфраструктуру.

За оцінками експертів, використання хмарних технологій в багатьох випадках дозволяє скоротити витрати в два-три рази в порівнянні з утриманням власної розвиненої IT-структури.

"Хмара" відкриває новий підхід до обчислень, при якому ані обладнання, ані програмне забезпечення не належать підприємству. Замість цього провайдер надає замовнику вже готовий сервіс.

До допомоги "хмар" часто вдаються молоді компанії-стартапи, які потребують великих обчислювальних ресурсів для обслуговування користувачів, але не можуть дозволити собі створення і експлуатацію власного дата-центру.

Одним з перших широкодоступних хмарних інтернет-сервісів стала електронна пошта з веб-інтерфейсом. У цьому випадку всі дані зберігаються на віддалених серверах, а користувач отримує доступ до своїх листів через браузер з будь-якого комп'ютера або достатньо потужного мобільного пристрою.

Національним інститутом стандартів і технологій США встановлені такі обов'язкові характеристики хмарних обчислень:

- Самообслуговування на вимогу (англ. self service on demand), споживач самостійно визначає і змінює обчислювальні потреби, такі як серверний час, швидкості доступу та обробки даних, обсяг збережених даних без взаємодії з представником постачальника послуг;
- Універсальний доступ по мережі, послуги доступні споживачам через мережу передачі даних незалежно від термінального пристрою;
- Об'єднання ресурсів (англ. resource pooling), постачальник послуг об'єднує ресурси для обслуговування великої кількості споживачів в єдиний пул для динамічного перерозподілу потужностей між споживачами в умовах постійної зміни попиту на потужності; при цьому споживачі контролюють тільки основні параметри послуги (наприклад, обсяг даних, швидкість доступу), але фактичний розподіл ресурсів, що надаються споживачеві, здійснює постачальник (в деяких випадках споживачі все ж можуть керувати деякими фізичними параметрами перерозподілу, наприклад, вказувати бажаний центр обробки даних з міркувань географічної близькості);

- Еластичність, послуги можуть бути надані, розширені, звужені в будь-який момент часу, без додаткових витрат на взаємодію з постачальником, як правило, в автоматичному режимі;
- Облік споживання, постачальник послуг автоматично обчислює спожиті ресурси на певному рівні абстракції (наприклад, обсяг збережених даних, пропускна здатність, кількість користувачів, кількість транзакцій), і на основі цих даних оцінює обсяг наданих споживачам послуг.

З точки зору постачальника, завдяки об'єднанню ресурсів та непостійному характеру споживання з боку споживачів, хмарні обчислення дозволяють економити на масштабах, використовуючи менші апаратні ресурси, ніж при виділенні апаратних потужностей для кожного споживача, а за рахунок автоматизації процедур модифікації виділення ресурсів істотно знижуються витрати на абонентське обслуговування.

З точки зору споживача, ці характеристики дозволяють отримати послуги з високим рівнем доступності (англ. high availability) і низькими ризиками непрацездатності, забезпечити швидке масштабування обчислювальної системи завдяки еластичності без необхідності створення, обслуговування і модернізації власної апаратної інфраструктури.

Зручність і універсальність доступу забезпечується широкою доступністю послуг і підтримкою різного класу термінальних пристроїв (персональних комп'ютерів, мобільних телефонів, інтернет-планшетів).

1.2 Моделі обслуговування

Послуги які надає провайдер хмарного сервісу розподіляються в залежності повноважень користувача, програмного забезпечення та методів використання хмари.

Програмне забезпечення як послуга (англ. Software as a service, SaaS) — це модель пропозиції програмного забезпечення споживачеві, при якій постачальник

розробляє веб-застосунок, розміщує його і управляє ним (самостійно або через третіх осіб) з метою і можливістю використання замовниками через інтернет. Замовники платять не за володіння програмним забезпеченням як таким, а за його використання (через API, що доступний через веб і часто використовує веб-служби). Близьким до терміну SaaS є термін «застосунок за запитом» (On-Demand).

Принциповою відмінністю моделі SaaS від раніших (Hosted Applications і Application Service Provider (ASP)) є те, що отримується саме послуга і інтерфейс (призначений для користувача або програмний), тобто деяка функціональність без жорсткої прив'язки до способу її реалізації.

Тому у такій моделі обслуговування можна виділити наступні особливості:

- Застосунок пристосований для віддаленого використання;
- Одним застосунком користується декілька клієнтів;
- Оплата стягується як щомісячна абонентська плата або на основі обсягу транзакцій;
- Підтримка застосунку входить до складу оплати;
- Модернізація застосунку відбувається плавно і прозоро для клієнтів;
- Постачальник сервісу SaaS забезпечує безпеку та цілісність даних.

Платформа як послуга (англ. Platform as a service, PaaS) — це модель обслуговування, в межах якої споживачу надається можливість розгортання на базі хмарної інфраструктури створених ним або набутих прикладних програм, які розроблені з використанням мов програмування, бібліотек, сервісів та інструментів наданих хмарним провайдером. Споживач не має змоги керувати та контролювати базову інфраструктуру хмари (до її складу входять комунікаційні мережі, сервери, операційні системи, засоби збереження), проте він контролює розгорнуті прикладні програми та, можливо, налаштування середовища, в якому вони розміщені.

Інфраструктура як послуга (англ. Infrastructure as a service, IaaS) — це модель обслуговування, в межах якої споживачу надається можливість керувати засобами обробки та збереження, комунікаційними мережами, та іншими фундаментальними обчислювальними ресурсами, на базі яких споживач може розгортати та виконувати довільне програмне забезпечення, до складу якого можуть входити операційні

системи та прикладні програми. Споживач не керує фізичною та віртуальною інфраструктурою, що лежить в основі хмари, проте він контролює операційні системи, системи збереження, встановлені програми та, можливо, має обмежений контроль над деякими мережевими компонентами (наприклад, мережевими екранами вузлів).

IaaS складається з трьох основних компонентів:

- Апаратні засоби (сервери, системи зберігання даних, клієнтські системи, мережеве обладнання);
- Операційні системи та системне ПЗ (засоби віртуалізації, автоматизації, основні засоби управління ресурсами);
- Зв'язуюче ПЗ (наприклад, для управління системами).

Більш наглядно різницю хмарної моделі в порівняні з звичайною та різницю між трьома наведеними вище моделями розгортання можна побачити на схемі що наведена у додатку А.

1.3 Моделі розгортання

Приватна хмара (англ. private cloud) – використовується для надання сервісів всередині однієї компанії, яка є одночасно і замовником і компанією, що надає послуги. Це варіант реалізації "хмарної концепції", коли компанія створює її для себе, в рамках організації. В першу чергу такі впровадження спрямовані на забезпечення приватності. Створення приватних хмар знімає одне із важливих питань, яке неодмінно виникає у замовників при ознайомленні з цією концепцією – питання про захист даних з точки зору інформаційної безпеки. Так, як "приватна хмара" обмежена рамками самої компанії, це питання вирішується стандартними існуючими методами. Для private cloud характерне зниження вартості обладнання за рахунок можливості визначити потенційну кількість користувачів в зв’язку з тим, що перелік необхідних сервісів і потенційна кількість користувачів відома на етапі проектування.

Публічна хмара (англ. public cloud) – це хмарна інфраструктура, яка призначена для вільного використання широким загалом. Публічна хмара може перебувати у власності, керуванні та експлуатації комерційних, академічних (освітніх та наукових) або державних організацій (чи будь-якої їх комбінації). Публічна хмара перебуває в юрисдикції постачальника хмарних послуг.

Гібридна хмара (англ. hybrid cloud) – це хмарна інфраструктура, що складається з двох або більше різних хмарних інфраструктур (приватних, громадських або публічних), які залишаються унікальними сутностями, але з’єднанні між собою стандартизованими або приватними технологіями, що уможливлюють переносимість даних та прикладних програм (наприклад, використання ресурсів публічної хмари для балансування навантаження між хмарами).[6]

1.4 Віртуалізація

Віртуалізація – це процес абстрагування обчислювальних ресурсів від фізичної основи на якій вони побудовані

Віртуалізація дозволяє розподіл апаратних ресурсів фізичної машини між різними копіями віртуальних машин, на кожній з яких може бути встановлена своя операційна система. Пласт програмного забезпечення, що виконує віртуалізацію та керує навантаженняс на окремі екземпляри віртуальних машин, називається гіпервізором. Гіпервізор також може (але не зобов'язаний) надавати працюючим під його управлінням ОС засоби зв'язку і взаємодії між собою (наприклад, через обмін файлами або мережеві з'єднання) так, ніби ці ОС виконувалися на різних фізичних комп'ютерах та моделювати не існуюче на хост-машині апаратне забезпечення.[11]

Гіпервізор сам по собі в деякому роді є мінімальною операційною системою (мікроядром або наноядром). Він надає запущеним під його управлінням операційних систем сервіс віртуальних машин, віртуалізуючи або емулюючи апаратне забезпечення (в тому числі процесор), і керує цими віртуальними

машинами. Гіпервізор дозволяє незалежне «включення», «перезавантаження», «вимкнення» кожної з віртуальних машин з тією чи іншою ОС. При цьому операційна система, що працює у віртуальній машині під управлінням гіпервізору, може, але не зобов'язана «знати», що вона виконується у віртуальній машині, а не на реальному апаратному забезпеченні.

Гіпервізори поділяються на 3 типи:

Автономний гіпервізор (Тип 1) – має свої вбудовані драйвери пристроїв, моделі драйверів і планувальник і тому не залежить від базової ОС. Оскільки автономний гіпервізор працює безпосередньо на обладнанні, то він продуктивніший.

На основі базової ОС (Тип 2, V) – це компонент, який працює в одному кільці з ядром основної ОС (кільце 0). Гостьовий код може виконуватися прямо на фізичному процесорі, але доступ до пристроїв вводу-виводу комп'ютера з гостьової ОС здійснюється через другий компонент, звичайний процес основної ОС – монітор рівня користувача.

Гібридний (Тип 1+) – гібридний гіпервізор складається з двох частин: з тонкого гіпервізора, що контролює процесор і пам'ять, а також працюючої під його управлінням спеціальної сервісної ОС в кільці зниженого рівня. Через сервісну ОС гостьові ОС отримують доступ до фізичного устаткування.

1.5 Симетричне шифрування

При створенні програми був використаний алгоритм шифрування Rijndael.

Rijndael — симетричний алгоритм блочного шифрування, фіналіст конкурсу AES і прийнятий в якості американского стандарту шифрування урядом США. Вибір припав на Rijndael з розрахуванням на широке використання і активний аналіз алгоритму, як це було із його попередником, DES. Державний інститут стандартів і технологій (англ. National Institute of Standards and Technology, NIST) США опублікував попередню специфікацію AES Rijndael 26 жовтня 2001 року, після

п'ятилітньої підготовки. 26 травня 2002 року AES на основі Rijndael оголошено стандартом шифрування.

Алгоритм Рейндол підтримує широкий діапазон розміру блоку та ключа (розмірність блоку та ключа може змінюватись із кроком 32 біта у діапазоні від 128 до 256 біт) та використовує операції з полями Галуа $GF(2^8)$. Для ключа 128 біт алгоритм має 10 раундів у яких послідовно виконуються операції

- subBytes()
- shiftRows()
- mixcolumns()
- xorRoundKey()

Станом на 2006 рік Rijndael є одним із найпоширеніших алгоритмів симетричного шифрування

1.6 Огляд систем – аналогів

BoxCryptor – шифрує файли поміщенні в однойменну директорію за допомогою алгоритму AES з 256 бітовим ключем. Оскільки BoxCryptor орієнтований на роботу з сервісом DropBox, після інсталяції він автоматично виконує пошук директорії Dropbox та пропонує створити свою папку в ній. Разом з нею створюється і віртуальний диск. В програмі є можливість копіювати файли як в папку BoxCryptor, так і на цей диск. Їх вміст дублюється. При збереженні файлів в захищене сховище вони шифруються на льоту. При спробі відкрити файл, програма в реальному часі дешифрує їх. Таким чином ніяких недоліків при повсякденній роботі з файлами користувач не відчуває, однак на сервери DropBox вони передається у зашифрованому вигляді, що виключає можливість доступу до них третьої сторони. Також зашифрованими файли залишаються на комп’ютері користувача, що можна віднести до недоліків даної системи. На відмінність від інших рішень по шифруванню файлів CryptoBox шифрує окремі файли а не їх

контейнери, завдяки чому не виникає проблем з синхронізацією окремих файлів, а також з зміною розміру віртуального сховища.

Програма BoxCryptor знаходиться на стадії бета тестування, після якої розробники планують зробити програму платною. Проте файли розміром менше ніж 2 Гб можна буде шифрувати безкоштовно.

EncFS – це віртуальна криптографічна файлова система. Папка з зашифрованими файлами, яку потрібно буде зберігати в DropBox, монтується на в будь-яку іншу папку на комп'ютері користувача, де всі дані представляються у незашифрованому вигляді. Тобто для кінцевого користувача робота з файлами залишається абсолютно прозорою. Шифрування відбувається за допомогою ключів AES або Blowfish, при цьому сам ключ також зберігається в захищеній директорії, а імена файлів і папок в ній при бажанні перетворюються у випадковий набір символів. Також до переваг EncFS необхідно віднести можливість практично безмежного розширення папки як у меншу так і у більшу сторону. Проте недоліком є необхідність встановленої клієнтської програми DropBox.

TrueCrypt – це вільне програмне забезпечення з відкритим первинним кодом, що використовується для прямо поточного шифрування для 32 та 64-розрядних операційних систем Microsoft Windows 2000/XP/2003/Vista та Linux (консольна версія). Програма дозволяє створювати віртуальний зашифрований диск (том TrueCrypt) у вигляді файлу та підключати його як справжній логічний диск жорстокого диску. TrueCrypt може використовувати для зберігання зашифрованої інформації повністю весь існуючий логічний диск або якийсь носій інформації, наприклад, флопі-диск чи USB флеш-пам'ять. Всі збережені дані на диску TrueCrypt шифруються, включаючи імена файлів та директорій. Том TrueCrypt подібний до фізичного жорсткого диску, тому, наприклад, відновлювати файлову систему на шифрованому диску можна з допомогою звичних утиліт (напр.CHKDSK), проводити дефрагментацію шифрування: AES (256-bit key), Blowfish (448-bit key), CAST5 (128-bit key), Serpent і т.п. Доступні алгоритми (256-bit key), Triple DES, Twofish (256-bit key). Для використання шифрування у сервісі DropBox необхідно встановити офіційний клієнт.

2 АНАЛІЗ ПЕРЕВАГ ТА НЕДОЛІКІВ ВИКОРИСТАННЯ ХМАРНИХ СЕРВІСІВ

2.1 Аналіз переваг хмарних сервісів

На сьогоднішній день хмарний сервіс включає три основних характеристики, які відрізняють його від звичайного сервісу:

- режимність "ресурси по запиту";
- еластичність;
- незалежність від елементів управління інфраструктурою.

Таким чином, ці технології при сумісному використанні дозволяють користувачам хмарних обрахунків використовувати обчислювальні потужності і масиви даних, які за рахунок відповідних технологій віртуалізації і високого рівня абстракції надаються їм як послуги, а основними перевагами порівняно з звичайними серверами є:

- життєстійкість – можливість резервування критичних ресурсів;
- висока швидкість обробки даних;
- масштабованість: збільшення доступних обчислювальних потужностей, яка практично обмежена лише розміром "хмари";
- централізованість управління і обновлення програм;
- зниження витрат на побудову і супроводження;
- зниження затрат на апаратне і програмне забезпечення, на обслуговування і електроенергію;
- простота сумісної роботи групи користувачів.

При використанні хмарних обчислень, споживачі інформаційних технологій можуть істотно знизити капітальні витрати - на побудову центрів обробки даних, закупівлю серверного та мережевого обладнання, апаратних і програмних рішень щодо забезпечення безперервності і працездатності - так як ці витрати поглинаються провайдером хмарних послуг. Крім того, тривалий час побудови та введення в експлуатацію великих об'єктів інфраструктури інформаційних технологій та висока

їх початкова вартість обмежують можливість гнучко реагувати на потреби ринку, тоді як хмарні технології забезпечують можливість практично миттєво реагувати на збільшення попиту на обчислювальні потужності.

При використанні хмарних обчислень, витрати споживача зміщуються в бік операційних - таким чином компенсуються витрати на оплату послуг хмарних провайдерів.

2.2 Аналіз загроз безпеки

2.2.1 Неправомірне використання хмарних технологій

IaaS провайдери пропонують ілюзію нескінченних ресурсів, надання яких в користування проходить дуже просто і швидко. Часто це суміщено з реєстрацією нового користувача, коли зареєструватися може будь-яка людина, у якої є кредитна картка. Використовуючи простоту реєстрації спамери, автори неправомірного коду та інші зловмисники можуть використовувати хмарний сервіс у своїх злочинних цілях. Раніше тільки PaaS провайдери страждали від такого роду атак, однак, останні дослідження[3] демонструють, що хакери почали використовувати IaaS сервіс для організації взлому паролей, DDOS атак, розміщення зловмисний код та програмне забезпечення, створення ботнет мереж та іншого.

Наприклад IaaS сервіс був використаний для створення найбільшої ботнет мережі ZeuS втрати від якої тільки в США склали 70 мільйоні доларів [10], збереження троянського коня InfoStealer та розміщення різних вразливостей в Microsoft Office та Adobe PDF. Крім того великою проблемою є те що ботнет мережі використовували IaaS сервіси для керування своїми пірами та розповсюдження спаму. При цьому деякі IaaS сервіси потрапляли в чорні списки, а їх користувачі повністю ігнорувались почтовими серверами.

2.2.2 Незахищені програмні інтерфейси

Провайдери хмарної інфраструктури надають користувачам набір програмних інтерфейсів для керування ресурсами, віртуальними машинами чи сервісами. Безпечність всієї системи залежить від захищеності цих інтерфейсів. Починаючи від процесів аутентифікації та авторизації, і закінчуючи шифруванням програмні інтерфейси повинні забезпечувати максимальний рівень захисту від різного сорту атак зловмисників.

Анонімний доступ чи можливість використовувати фактор аутентифікації (пароль чи токен), а також передача реєстраційних даних відкритим текстом є основними ознаками небезпечних програмних інтерфейсів. Обмеженість можливостей моніторингу використання API, відсутність систем ведення журналу, а також невідомі взаємозв’язки між різними сервісами тільки збільшують ризики взлому.

2.2.3 Внутрішні порушники

Проблема неправомірного доступу до інформації зсередини надзвичайно небезпечна. Часто, на стороні провайдеру не введена система моніторингу активності співробітників, і це означає, що зловмисник може отримати доступ до інформації користувача, використовуючи своє службове положення. Так як провайдери не розкривають своєї політики набору співробітників, загроза може надходити, як від хакера любителя так і від організованої злочинної структури, що проникнула в ряди співробітників провайдера.

Прикладом цього є обвинувачення проти співробітника Google який користуючись своїм службовим положенням адміністратора, добував конфіденційну інформацію про логи чату, логи дзвінків Google Voice, та почту користувачів.[34]

2.2.4 Вразливість в хмарних технологіях

Як було казано вище провайдери IaaS сервісів використовують абстракцію апаратних ресурсів за допомогою систем віртуалізації. Однак апаратні ресурси можуть бути спроектовані без врахування роботи з розподіленими ресурсами. Для того щоб, мінімізувати вплив даного фактору, гіпервізор керує доступом віртуальної машини до апаратних ресурсів, однак навіть в гіпервізорах можуть існувати вразливості, використання яких може призвести до підвищення привілегій певного користувача або навіть до отримання неправомірного доступу до фізичного обладнання хмарного сервера.

2.2.5 Інші вразливості

Вірогідність втрата інформації, що може виникнути з великої кількості причин, таких як втрата ключа шифрування яка приводить до неможливості відновлення даних або вихід із ладу накопичувача інформації, в складних хмарних сервісах зростає через тісну взаємодію між великою кількістю елементів хмари.
IaaS провайдери надають ілюзію безграничних ресурсів, виділення яких у користування відбувається швидко і легко. Використовуючи простоту регістрації нового користувача, спамери, автори небезпечного коду і інші злочинні особи можуть використовувати хмарний сервіс у своїх злочинних цілях. І якщо раніше тільки PaaS провайдери страждали від такого сорту атак то тепер хакери стали використовувати IaaS сервіси для організації взлому паролів, DDOS атак, створення ботнет мереж та іншого.

Тому основними недоліками хмарних сервісів можна виділити наступні:

- залежність від "хмарного" провайдера;
- залежність від каналів зв’язку, які в більшості регіонів країни характеризуються відсутністю SLA-якості сервісу, що надається (QoS);

- залежність збереження даних користувача від компаній, що надають послугу хмарних обчислень.

2.3 Висновок по розділу

В даному розділі було розглянуто основні переваги хмарної моделі, такі як зниження витрат на апаратну інфраструктуру, безкоштовне обслуговування, можливість нарощування, швидкий доступ до даних, більш короткий час розгортання, а також її недоліки що зводяться в основному до необхідності у постійному доступі до мережі, збільшенням витрат при зростанню потужностей та проблемам забезпечення безпеки інформації.

Не дивлячись не те, що кількість плюсів переважає мінуси, в кожній конкретній ситуації кожен з зазначених факторів може бути вирішальним при прийнятті рішення, щодо використання сервісів хмарних обчислень. Зокрема таким фактором є безпека збереження інформації користувача. Для захисту інформації при використанні хмарних технологіях комп’ютерних обчислень вирішено використати шифрування на клієнтській стороні що забезпечує неможливість використання даних користувача третьою стороною при втраті облікового запису, неправомірного доступу з боку провайдеру, незахищеного арі хмарного сервісу. Недоліком такого рішення є повна втрата інформації при втраті ключа шифрування користувача.

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Аналіз функцій системи

Призначення даної системи синхронізація однієї з папок комп’ютеру користувача з його аккаунтом у хмарному сервісі DropBox. Відмінністю даної програми від клієнтської програми DropBox повинно бути шифрування інформації користувача перед її збереженням у онлайн сховище. Шифрування відбувається за допомогою симетричного алгоритму Рейндол, що забезпечить прямопоточне шифрування файлів за допомогою ключа шифрування, що вводиться користувачем у відповідне поле програми.

Синхронізація інформації відбувається шляхом циклічного порівняння вмісту локальної папки з її хмарним аналогом. При виявленню файлу, що відсутній на DropBox файл у потоці шифрується а потім передається на DropBox. Якщо ж файл відсутній у папці, і користувач правильно ввів ключ шифрування – файл буде завантажено з DropBox, розшифровано, та збережено у папку користувача.

Таким чином на комп'ютері користувача постійно зберігаються розшифровані файли, тоді як на сервісі DropBox їх зашифровані версії.

Файл, що був видалений з папки користувача під час синхронізації буде видалено з хмарного сховища.

Також є можливість зберігати на сервісі незашифровані файли – при синхронізації вони будуть завантажені незміненими. При завантаженні на сервіс файлів з різними паролями на комп’ютер користувача будуть завантажені усі файли, проте частина з них що завантажувалась під іншим паролем залишиться незміненою.

Програма має зручний інтерфейс, що виконаний з використанням принципів Microsoft design language (Metro UI).

3.2 Вибір мови програмування

Мовою програмування при розробці було обрано мову C# та платформу .NET Framework версії 4.0.

Common Language Runtime, скорочено CLR — «загальне середовище виконання мов» — це компонент пакету Microsoft .NET Framework, віртуальна машина, на якій виконуються всі мови платформи .NET Framework.

CLR транслюється початковий код в байт-код на мові IL, реалізація компіляції якого компанією Microsoft називається MSIL, а також надає MSIL-програмам (а отже, і програмам, написаним на мовах високого рівня, що підтримують .NET Framework) доступ до бібліотеки класів .NET Framework, або так званою .NET FCL (англ. Framework Class Library).

Середовище CLR є реалізацією специфікації CLI (англ. Common Language Infrastructure), специфікації загальномовної інфраструктури, компанією Microsoft.

Віртуальна машина CLR дозволяє програмістам забути про багато деталей про конкретний процесор, на якому виконуватиметься програма. CLR також забезпечує такі важливі служби як:

- управління пам'яттю;
- управління потоками;
- обробка винятків;
- збірка сміття;
- безпека виконання.

C# розроблявся як мова програмування прикладного рівня для CLR і, як такий, залежить, перш за все, від можливостей самої CLR. Це стосується, перш за все, системи типів C#. Присутність або відсутність тих або інших виразних особливостей мови диктується тим, чи може конкретна мовна особливість бути трансльована у відповідні конструкції CLR. Так, з розвитком CLR від версії 1.1 до 2.0 значно збагатився і сам C#; подібної взаємодії слід чекати і надалі. (Проте ця закономірність буде порушена з виходом C# 3.0, що є розширеннями мови, що не

спираються на розширення платформи .NET.) CLR надає C#, як і всім іншим .NET-орієнтованим мовам, багато можливостей, яких позбавлені «класичні» мови програмування. Наприклад, збірка сміття не реалізована в самому C#, а проводиться CLR для програм, написаних на C# точно так, як і це робиться для програм на VB.NET, J# тощо.

Таким чином технологія .NET Framework та мова програмування C# забезпечують простоту у керуванні пам'яттю та потоками, а також надзвичайну швидкість розробки програм завдяки об’єктно-орієнтованому підходу та таким інструментам як Visual Studio та Expression Blend.

3.3 Вибір архітектурної моделі програмного забезпечення

Архітектурна модель даної системи – три-рівнева архітектура, яка передбачає наявність наступних компонентів програми:

Клієнтський рівень – частина системи, з якою взаємодіє користувач (User Interface).

Рівень логіки – тут розміщена основна логіка програми. Рівень логіки даної програми складається з двох менших рівнів – ядра системи (його використовує клієнтський рівень для доступу до рівня підсистем) та підсистеми що реалізують різноманітні функції програми (такі як шифрування файлів) та мають доступ до рівня роботи з хмарою.

Рівень роботи з хмарою – частина системи, яка забезпечує безпосередню роботу з хмарною платформою, аудентифікацію користувача, запис файлів та зчитування їх з серверу.

Дана архітектурна модель надає такі переваги:

- масштабованість;
- ізольованість рівнів один від одного дозволяє швидко і простими засобами переконфігурувати систему при виникненні збоїв або при плановому обслуговуванні на одному з рівнів;

- висока безпека;
- висока надійність;

Недоліками є такі особливості:

- більш висока складність створення додатків;
- складніше в розгортанні і адмініструванні;
- високі вимоги до продуктивності серверів додатків і сервера бази даних, а, значить, і висока вартість серверного обладнання;
- високі вимоги до швидкості каналу (мережі)

Дані недоліки не є критичними для задачі, яку потрібно виконати.

В додатку Б показана архітектурна модель програмного забезпечення комп'ютерної системи. Дана діаграма ілюструє на які функціональні підсистеми розділена програма.

3.4 Розробка структурної моделі системи

Структурна модель (рис 3.1) визначає основні структурні елементи системи. Вона будується на основі діаграми варіантів використання і визначення основних функцій.

Дана програма складається з таких модулів:

- програмний модуль роботи з інтерфейсом;
- програмний модуль головної логіки;
- програмний модуль доступу до хмари.

Для роботи з інтерфейсом використовуються такі програмні модулі:

- програмний модуль відображення контролів;
- програмний модуль для відображення та додавання файлів;

В комплекс програм головної логіки входять:

- програмний модуль обробки запитів клієнта;
- програмний модуль шифрування та дешифрування файлів.

В комплекс програм роботи з хмарою:

- програмний модуль реалізації функції Login;
- програмний модуль для передачі та отримання файлів.

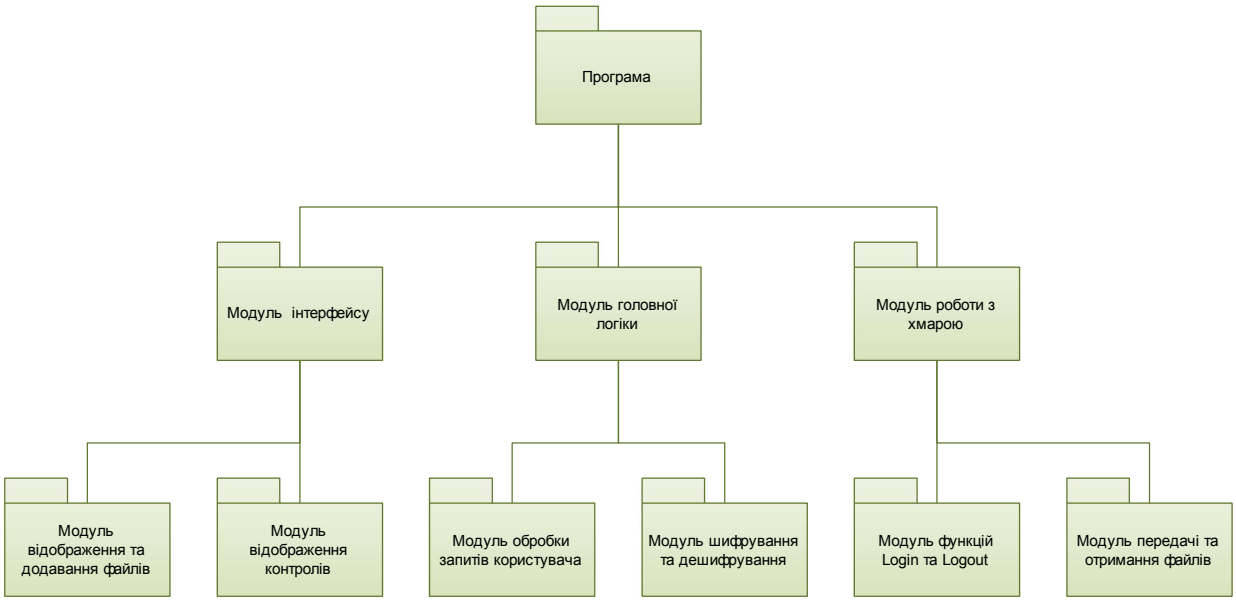


Рисунок 3.1 – Структурна модель системи

3.5 Розробка UML – діаграми використання

Діаграми використання (англ. Use Case) — це UML діаграми, які служать для документування вимог до системи, тобто, що має робити система. Діаграма використання є графом, що складається з множини акторів, прецедентів (варіантів використання) обмежених границею системи, асоціацій між акторами та прецедентами, відношень серед прецедентів, та відношень узагальнення між акторами [7].

Діаграми прецедентів відображають елементи моделі варіантів використання. Основними поняттями, пов'язаними з прецедентами є актори, прецеденти, та суб'єкт. Суб'єкт — це система, що розглядається і до якої відносяться прецеденти. Поведінка суб'єкта описується одним або більше прецедентами, що визначаються відповідно до потреб акторів.

Діаграма використання дозволяє здійснити аналіз функцій системи. Кожен окремий варіант використання описує поведінку системи відносно зовнішніх об’єктів [7, 8].

В UML діаграмі використання, що показана в додатку Г, як актори зазначені:

- Користувач (оператор).
- Інтерфейс системи.
- Система.
- Об’єкт управління.
- Пристрій управління.
- Інформаційно-вимірювальна система – забезпечує конвертацію аналогових сигналів в цифровий сигнал який записується на диску у вигляді бінарного файлу.

UML діаграма використання знаходиться у додатку В, і демонструє основні дії між акторами що наведені вище.

3.6 Розробка програмного забезпечення

Під час своєї роботи програма кожну секунду порівнює файли у обраній користувачем папці з файлами на сервісі DropBox. Лістинг класу який проводить синхронізацію файлів подано у додатку Г. Користувацький інтерфейс програми створено за допомогою Windows Presentation Foundation – графічної (презентаційна) підсистема в складі .NET Framework 3.0, та XAML (скорочення від Extensible Application Markup Language — розширювана мова розмітки застосунків), що є мовою розмітки, яку використовують для створення екземплярів об'єктів .NET. Хоча мова XAML — це технологія, що може бути застосовна до багатьох різних предметних областей, її головне призначення — конструювання інтерфейсів користувачів WPF. Інакше кажучи, документи XAML визначають розташування панелей, кнопок та інших елементів керування, що становлять вікна в застосунку WPF. Головне вікно програми показано на рисунку 3.2.

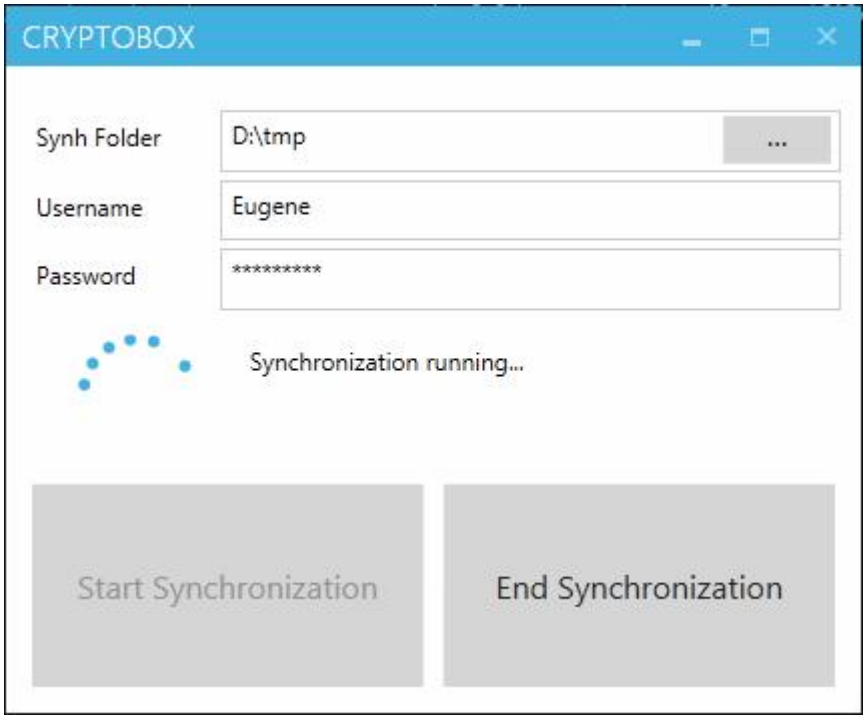


Рисунок 3.2 – Головне вікно програми

Лістинг вікна можна побачити у додатку Д. При натисненні кнопки початку синхронізації вперше програма відкриє у вбудованому браузері запит на дозвіл прив’язки програми до облікового запису DropBox (рис. 3.4).

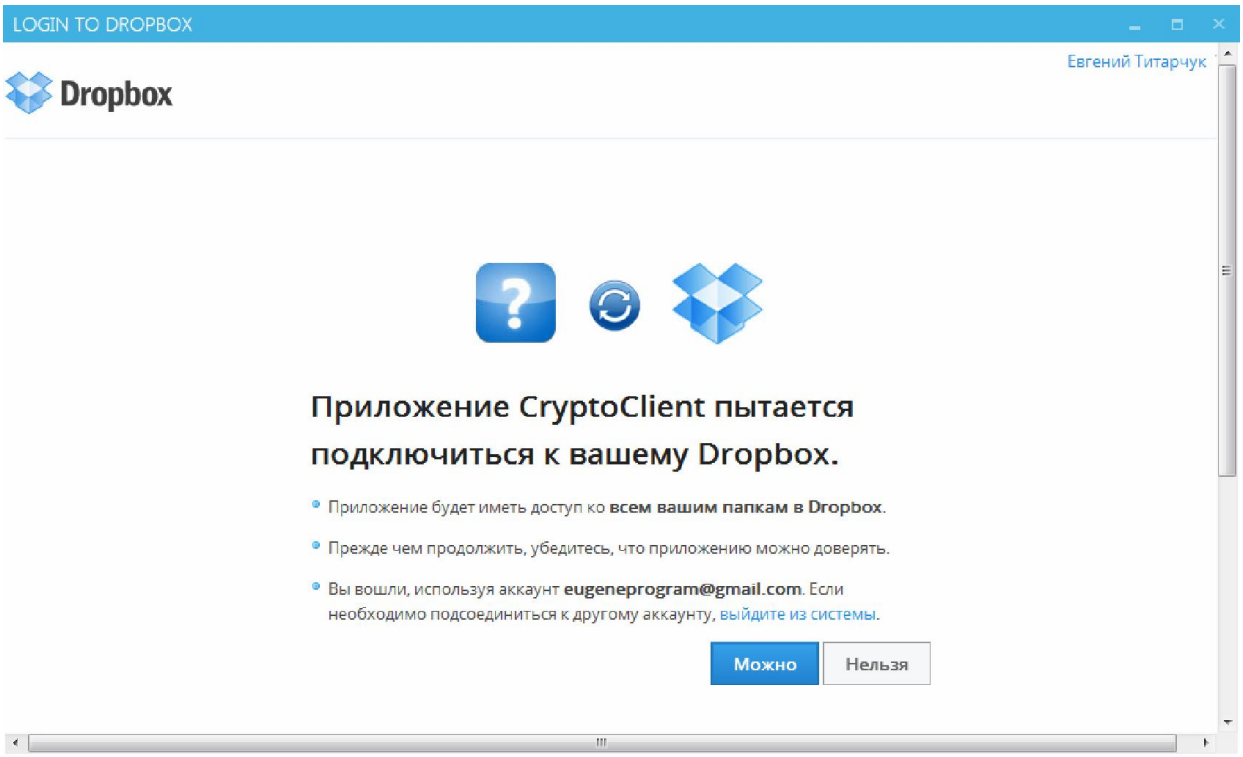


Рисунок 3.3 – Вікно прив’язки до облікового запису DropBox

Під час синхронізації файли розшифровуються та зашифровуються на клієнтському комп’ютері, таким чином на сервісі зберігаються тільки зашифровані версії файлів користувача. Код класу шифрування знаходиться у додатку Е.

4 ОХОРОНА ПРАЦІ

4.1 Аналіз небезпечних та шкідливих виробничих факторів у виробничому приміщенні

При роботі в даному приміщенні, виникає низка небезпечних і шкідливих виробничих факторів, які класифікуються за [25].

Небезпечний виробничий фактор – небажане явище, яке супроводжує виробничий процес і дія якого за певних умов може призвести до травми або іншого раптового погіршення здоров'я працівника (гострого отруєння, гострого захворювання) і навіть до раптової смерті.

Шкідливий виробничий фактор – небажане явище, яке супроводжує виробничий процес і вплив якого на працюючого може призвести до погіршення самопочуття, зниження працездатності, захворювання, виробничо зумовленого чи професійного, і навіть смерті, як результату захворювання.

Як правило, всі несприятливі виробничі чинники об'єднують в єдине поняття – небезпечний та шкідливий виробничий фактор.

Визначальними ознаками небезпечних та шкідливих виробничих факторів є: можливість безпосереднього негативного впливу на організм працівника; ускладнення нормального функціонування органів працівника; можливість порушення нормального стану елементів виробничого процесу, в результаті чого можуть виникати аварії, вибухи, пожежі, травматизм.

Вказане приміщення характеризується небезпечними та шкідливими виробничими факторами фізичної, хімічної, біологічної та психофізіологічної груп [25], які розподіляються таким чином:

1) Фізичні небезпечні і шкідливі виробничі фактори:

- підвищений рівень інфразвуку, шуму, ультразвуку та вібрації;
- підвищений рівень електромагнітних випромінювань;
- високе значення напруги в електричній мережі;

- понижена або підвищена температура, вологість і рухливість повітря робочої зони;
- підвищена інтенсивність теплового випромінювання;
- недостатність або відсутність природного освітлення;
- недостатня освітленість робочої зони;
- пряма або відбита блискучість.

2) Хімічні небезпечні і шкідливі фактори – шкідливі хімічні речовини.

3) Біологічні небезпечні і шкідливі виробничі фактори – немає.

4) Психофізіологічні небезпечні і шкідливі виробничі фактори:

- фізичні перевантаження – немає.
- нервово-психічні перевантаження:

Наведемо імовірні причини виникнення вказаних факторів і стисло опишемо їхній вплив на організм людини.

Підвищений рівень шуму і вібрації робочої зони може бути спричинений роботою таких елементів комп'ютерів, як жорсткий диск, вентилятори блоку живлення, охолодження мікропроцесора, швидкісні CD-ROM (DVD-ROM), механічні сканери, пересувні механічні частини принтера, що може спровокувати швидку стомленість працюючого, погіршення слуху, нервові розлади.

Підвищений рівень інфразвуку може бути спричинений вентиляторами та іншими рухомими елементами обладнання з частотою рухів менше 20 Гц або 1200 об/хв, що може викликати нездужання типу морської хвороби, нервову втому.

Підвищений рівень ультразвуку може бути викликаний обладнанням, у якому генеруються ультразвукові коливання для виконання технологічних операцій, а також обладнання, при експлуатації якого ультразвук виникає як побічний фактор, що може спровокувати зсуви у стані нервової, серцево-судинної, дихальної, ендокринної системах організму, у обміні речовин та терморегуляції працівника.

Підвищений рівень електромагнітних випромінювань радіочастотного діапазону може бути спричинений лабораторними та вимірювальними приладами різного призначення, персональними комп'ютерами та інше, що може спровокувати катаракту, при якій помутніння розвивається на мембрані кришталика, підвищену

частоту захворювання глаукомою, а під дією підвищеної концентрації пилу поблизу екрана дисплея підвищується імовірність виникнення дерматитів обличчя (прищі, екземи, свербіж шкіри).

Підвищений рівень електромагнітних випромінювань промислової частоти може бути спричинений струмоведучими частинами працюючих електроустановок, що може спровокувати злоякісні пухлини. Найбільш сильна дія цих полів виявляється на відстані до 30 см від екрана. Не меншої інтенсивності досягають ці поля із задньої сторони дисплея (джерело рядковий трансформатор) – їхній шкідливий вплив поширюється на відстань до 0,7-1 м.

Високе значення напруги в електричній мережі може бути спричинене наявністю електрообладнання, що може спровокувати больовий шок, знепритомніння, одержання опіків при щільному контакті зі струмоведучими частинами.

Понижена або підвищена температура повітря робочої зони може бути спричинена різкою зміною температури повітря навколишнього середовища, наявністю або відсутністю опалення робочого приміщення тощо. Це може спровокувати перегрів або переохолодження організму працівника.

Понижена або підвищена відносна вологість повітря робочої зони може бути спричинена різною кількістю води, що випаровується у приміщенні, метрологічними умовами поза приміщенням, що може спровокувати зменшення або збільшення тепловіддачі організмом людини, що сприяє його перегріванню або переохолодженню. Понижена або підвищена рухливість повітря робочої зони може бути спричинена нераціональними параметрами системи вентиляції або її відсутністю, що може спровокувати порушення реакції терморегуляції організму працівника.

Підвищена інтенсивність теплового випромінювання може бути спричинена теплом, що потрапляє у виробниче приміщення від обладнання, опалювальних приладів, людей тощо, що може спровокувати підвищення температури повітря в приміщенні вище допустимих меж.

Недостатність або відсутність природного освітлення може бути спричинена відсутністю або недостатніми розмірами віконних пройм, а також наявністю конфронтуючих будинків та споруд. Відсутність або недостатність природного освітлення приводить до напруження зору, послабляє увагу, приводить до настання передчасної стомленості.

Недостатня освітленість робочої зони може бути спричинена відсутністю або недостатністю природного освітлення, нераціональним розташуванням світильників та ламп штучного освітлення тощо. Недостатня освітленість може стати причиною низької продуктивності праці, оскільки очі працівника перенапружуються, при цьому стає складно відрізнити оброблювані предмети.

Пряма або відбита блискучість може бути спричинена дзеркальною відбиваючою і неплоскою зовнішньою поверхнею екрану монітора, що може призвести до виникнення астенопічних явищ та функціональних змін ока.

Шкідливі хімічні речовини в повітрі робочої зони можуть бути спричинені виділенням пилу, озону, оксидів азоту й аероіонізації під час роботи за комп'ютером. В приміщеннях із ПК оператори піддаються впливу пилу, що притягається до працівника і сильно наелектризованого обладнання. Головними джерелами озону на комп'ютеризованих місцях є електронно-променеві трубки відеотерміналів та лазерні принтери. При роботі ПК виникає іонізація середовища, що приводить до фізико-хімічних змін у структурі речовин. У деяких людей вплив сильної запиленості приміщення може викликати алергію. Оксиди азоту чинять подразливу дію на органи дихання, викликаючи кашель, блювоту, іноді головний біль. Озон є дуже сильним окисником і при концентрації вище ніж 0,1 мг/м3 шкідливо впливає на здоров’я людини. Надзвичайна небезпека озону для здоров'я людини пов'язана з тим, що він належить до хімічних сполук, що викликають в живих організмах зміни, схожі з тими, які ви никають-після дії іонізуючого випромінювання. Тому озон вважається не лише подразнюючою, а й канцерогенною речовиною. Іонізація повітря може викликати невелике підвищення температури тіла під час роботи за комп'ютером.

Перенапруження аналізаторів може бути спричинене інтенсивною роботою за ЕОМ, що призводить до швидкої втоми органів зору працюючого і навіть до погіршення зору.

Монотонність праці може бути спричинена одноманітністю роботи працюючого і призводить до швидкого розвитку втоми в зв'язку з локалізацією м'язових і нервових навантажень, незадоволення роботою і зниження творчої активності працівника. Монотонність викликає також у працівника гіподинамію, підвищену плинність кадрів, розвиток неврозів.

4.2 Карта умов праці

Карта умов праці потрібна з метою здійснення атестації робочого місця. В таблиці 4.2 наведено оцінку факторів виробничого і трудового процесів.

1. Гігієнічна оцінка умов праці:
   - підвищена концентрація шкідливої хім. речовини 4-го класу небезпеки – 1 ст.
   - підвищений рівень шуму – 1 ст.
   - підвищена швидкість руху повітря в теплий період року – 1 ст.
   - підвищена інтенсивність теплового випромінювання – 1 ст.
   - недостатня освітленість робочої зони штучним освітленням – 2 ст.
2. Оцінка технічного й організаційного рівня:
   - технічний рівень робочого місця не відповідає нормативним вимогам.
3. Атестація робочого місця:
   - робоче місце атестовано за другим ступенем шкідливості.

Таблиця 4.2 – Оцінка факторів виробничого і трудового процесів

Номер		Фактори виробничого середовища і трудового процесу	Нормативне значення (ГДР, ГДК)		Фактичне значення	3-й клас: шкідливі умови і характер праці		
			ни-жнє	вер-хнє		I ступінь	II ступінь	III ступінь
1		Шкідливі хімічні речовини:						
		1-й клас небезпеки		–	–			
		2-й клас небезпеки		–	–			
		3-й, 4-й класи небезпеки		0,1	0,182	+		
2		Вібрація		33	26			
3		Шум		50	51	+		
4		Інфразвук		110	46			
5		Ультразвук		110	29			
6		Неіонізуючі випромінювання:						
		− радіочастотний діапазон		3	2			
		− діапазон промислової частоти		5	2,17			
		− оптичний діапазон		0	0			
7		Мікроклімат у приміщенні:						
		− температура повітря, °С	18	27	16			
		− швидкість руху повітря, м/с	0,2	0,4	1,1	+		
		− відносна вологість повітря, %		65	33			
		− інтенсивність теплового випромінювання, Вт/м²		140	226	+		
8		Виробниче освітлення:						
		− розряд зорових робіт	4		4			
		− КПО для природного освітлення, %	1,5		2,4			
		− освітленість для штучного освітлення, лк	200		57		+	
		Кількість факторів	–	–	–	4	1	0

4.3 Рекомендації щодо покращення умов праці

З метою забезпечення чистоти повітря робочої зони потрібно доповнити природну вентиляцію механічною.

Для забезпечення допустимих параметрів шуму в приміщенні потрібно проводити постійне змащування підшипників вентиляторів системи вентиляції.

З метою забезпечення допустимих параметрів швидкості руху повітря в приміщенні доцільно зменшити продуктивність вентиляції.

Для забезпечення нормованих параметрів інтенсивності теплового випромінювання в приміщенні потрібно використати кондиціювання повітря.

Для забезпечення нормованих параметрів освітленості для штучного освітлення в приміщенні потрібно збільшити потужність/кількість ламп в освітлювальних установках.

4.4 Розрахунок місцевого штучного освітлення люмінесцентними лампами

Вихідні дані: довжина та висота розміщення світильника d = 0,5 м і h = 0,6 м.

З погляду задач зорової роботи, відповідно до [31] знаходимо, що вони відносяться до IV розряду зорових робіт. Вибираємо контраст об’єкта з фоном – середній, а характеристику фону – середню, яким відповідає підрозряд в.

Нормовані значення коефіцієнта природного освітлення (КПО) та мінімальні значення освітленості при штучному освітленні наведені в таблиці Х.2.

Відповідно до рекомендацій [31] розрахунок місцевого освітлення проводимо точковим методом. В цьому методі початково приймається, що світовий потік в кожному світильнику рівний 1000 лк.

Світловий потік ламп світильника місцевого освітлення визначається за формулою 4.1.

(4.1)

де Е – нормована освітленість при місцевому освітленні, лк;

$K_з$ – коефіцієнт запасу (для виробничих приміщень КЗ = 1,3...1,5);

μ – коефіцієнт, що враховує відбиту складову освітленості;

$\sum e$ – сумарна освітленість від світильників в контрольній точці, лк.

Таблиця 4.2 – Нормовані значення коефіцієнта природного освітлення та мінімальні освітленості при штучному освітленні

Характеристика зорової роботи	Найменший розмір об'єкта розрізнення,	Розряд зорової роботи	Підрозряд зорової роботи	Контраст об'єкта розрізнення з фоном	Характеристика фону	Освітленість при штучному освітленні, лк			КПО для бокового освітлення, %	
						комбіноване		загальне	Природного	Суміщеного
						всього	у т. ч. від загального			
Середньої точності	0,5-1	IV	в	середній	середній	400	200	200	1,5	0,9

Нормовану освітленість при місцевому штучному освітленні визначимо як різницю нормованих значень освітленості при комбінованому та зальному штучному освітленні за формулою:

(4.2)

$$E = E_k - E_з \text{ [лк]},$$

де Ек – нормоване значення освітленості при комбінованому штучному освітленні, лк;

Ез – нормоване значення освітленості при зальному штучному освітленні, лк.

Згідно таблиці 4.2 Ек = 400 лк; Ез = 200 лк.

Таким чином, знайдемо нормовану освітленість при місцевому штучному освітленні за формулою 4.1:

$$E = 400 - 200 = 200 \text{ (лк)}.$$

Приймаємо коефіцієнт запасу К3 = 1,3...1,5 = 1,4.

Для місцевого штучного освітлення приймаємо μ = 1.

Сумарна освітленість від найближчих світильників в контрольній точці $\sum e$ визначається за допомогою просторових ізолюксів умовної горизонтальної освітленості в залежності від геометричних розмірів d та h (рисунок 4.1).

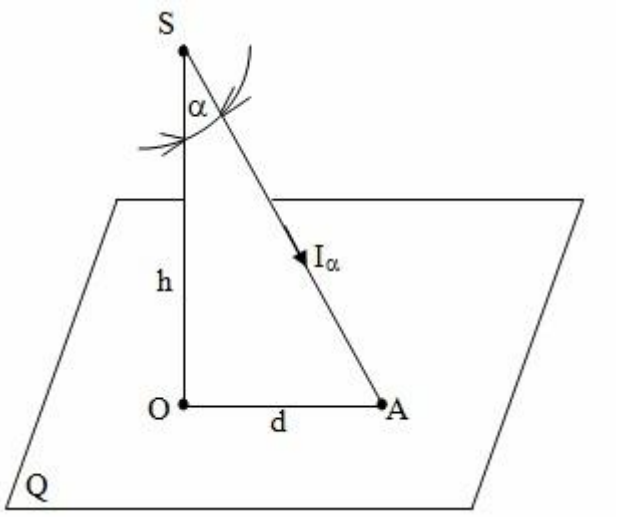


Рисунок 4.1 – Освітленість точки А, що належить горизонтальній площині Q, точковим джерелом світла S

Для d = 0,5 м і h = 0,6 м $\sum e$ = 29,5 лк.

Знайдемо значення світлового потік ламп світильника місцевого освітлення:

Для світильника місцевого освітлення вибираємо лампи ЛБ-80, для яких

світловий потік $Ф$ = 5400 лм.

Кількість ламп в світильнику місцевого освітлення розраховується за формулою:

(4.3)

Тоді за формулою 4.3:

$$n = \frac{9492}{5400} = 1{,}76\,(\text{шт})$$

Заокруглюємо число ламп у світильнику до найближчого цілого числа $n \approx 2$.

Знайдемо фактичне значення світлового потоку ламп світильника місцевого освітлення за формулою:

(4.4)

Визначимо відносне відхилення фактичного значення світлового потоку ламп світильника місцевого освітлення від розрахункового за формулою 4.5.

(4.5)

Знайдемо сумарну електричну потужність всіх ламп світильника за формулою 4.6.

$$\sum P_{СВ} = nP_{Л}\ [\text{Вт}], \tag{4.6}$$

де $P_{Л}$ = 80 Вт – потужність однієї лампи ЛБ-80.

Підставляючи відомі значення у формули (4.4, …, 4.6) отримаємо:

$$\Delta = \frac{10800 - 9492}{9492} 100 = 13{,}78\,(\%)$$

$$\sum P_{СВ} = 80 \cdot 2 = 160\,(\text{Вт})$$

Відносне відхилення Δ знаходиться в межах допустимих значень від –10% до +20%, що свідчить про правильність проведених розрахунків.

ВИСНОВКИ

В данній роботі було розглянуто основні загрози безпеки при використанні хмарних сервісів та споріднені теми. Дана характеристика основним параметрам хмарних сервісів, їх моделі обслуговування та розгортання, дано визначення віртуалізації. Було розглянуто існуючі переваги та недоліки хмарних сервісів, а також загрози приватній інформації в них та методи їх уникнення.

Після аналізу загроз безпеці приватної інформації зроблено висновок що шифрування приватної інформації користувача при збереженні її в хмарному сервісі зробить її недоступною для третьої стороні в більшості з перерахованих сценаріїв. На основі цього розроблено програму CryptoBox що шифрує дані на стороні клієнта, перед відправленням на хмарний сервіс, за допомогою стійкого симетричного алгоритму шифрування, також розглянуто існуючі аналоги створеної системи.

Проведено аналіз основних функції що повинна виконувати система, обрано мову програмування для створення системи, розроблено архітектурну та структурну моделі програми, діаграму використання.

З використанням нормативної літератури було здійснено аналіз шкідливих та небезпечних виробничих факторів у виробничому приміщенні (описання і класифікація потенційно шкідливих та небезпечних факторів, визначення можливих причини виникнення цих чинників і короткий опис їхньої дії на організм працівника); було заповнено карту умов праці; вказано рекомендації щодо покращення умов праці, а також здійснено розрахунок та вибір способу захисту від домінуючого шкідливого/небезпечного виробничого фактору.

ЛІТЕРАТУРА

1. Peter Mell, Timothy Grance. The NIST Definition of Cloud Computing / National Institute of Standards and Technology / Rebecca M. Blank. – Gaithersburg: NIST, 2011. – 286 с.
2. Климентьев И. П. Введение в Облачные вычисления / И. Климентьев, В. Устинов. – М.: УГУ, 2009. – 233 с.
3. Дж. Риз. Облачные вычисления / Дж. Риз. П.: БХВ-Петербург, 2011. — 288 с.
4. Питер Фингар. Dot.Cloud: облачные вычисления - бизнес-платформа XXI века / Питер Фингар. М.: Аквамариновая Книга, 2011. – 256 с.
5. Є. Гребнева, П. Єгоріхін. Хмарні сервіси / Є. Гребнева. – М.: CNews, 2011. – 282 с.
6. Хмарні обчислення [Електронний ресурс]: Хмарні обчислення // Wikipedia – Wikipedia. – Режим доступу: http://uk.wikipedia.org/wiki/Хмарні_обчислення. – Назва з екрану.
7. Крэг Ларман. Применение UML и шаблон проектирования. 2-е издание / Крэг Ларман. — М.: Вільямс, 2004. – 624 с.
8. Джим Арлоу. UML 2 и Унифицированный процесс. Практический объектно-ориентированный анализ и проектирование / Джим Арлоу — М.: Вильямс, 2007. – 617 с.
9. Cloud Services Forecast: 2009-2013 [Електронний ресурс]: Cloud Services Forecast: 2009-2013 // New IT – New IT. – Режим доступу: http://blogs.idc.com/ie/?p=543. – Назва з екрану.
10. ZeuS (троянская программа) [Електронний ресурс]: ZeuS (троянская программа) // Wikipedia – Wikipedia. – Режим доступу: http://ru.wikipedia.org/wiki/ZeuS_(троянская_программа). – Назва з екрану.

11. Віртуальна машина [Електронний ресурс]: Віртуальна машина // Wikipedia – Wikipedia. – Режим доступу: http://uk.wikipedia.org/wiki/Віртуальна_машина. – Назва з екрану
12. Облачные стратегии [Електронний ресурс]: Облачные стратегии // Бюро Соломатина – Бюро Соломатина. – Режим доступу: http://www.bureausolomatina.ru/ru/themes_in_progress/clouds. – Назва з екрану.
13. Облачные вычисления: Введение в программное обеспечение как сервис [Електронний ресурс]: Облачные вычисления: Введение в программное обеспечение как сервис // IBM – IBM. – Режим доступу: http://www.ibm.com/developerworks/ru/training/kp/cl-kp-cloudsaas. – Назва з екрану.
14. Інформаційні технології як сервіс [Електронний ресурс]: Інформаційні технології як сервіс // Microsoft – Microsoft. – Режим доступу: http://www.microsoft.com/ukraine/cloud/. – Назва з екрану.
15. Н. Макалистер. Виртуализация серверов / Н. Макалистер. — М.: Вільямс, 2007. – 264 с.
16. В. Романченко. Облачные вычисления на каждый день / В. Романченко. — М.: Вільямс, 2009. – 521 с.
17. Dropbox Core API [Електронний ресурс]: Dropbox Core API // Dropbox – Dropbox. – Режим доступу: https://www.dropbox.com/developers/core/. – Назва з екрану.
18. Getting started with Core API [Електронний ресурс]: Getting started with Core API // Dropbox – Dropbox. – Режим доступу: https://www.dropbox.com/developers/core/start/android – Назва з екрану.
19. Rest API [Електронний ресурс]: Rest API // Dropbox – Dropbox. – Режим доступу: https://www.dropbox.com/developers/core/docs – Назва з екрану.
20. SDK сторонних разработчиков [Електронний ресурс]: SDK сторонних разработчиков // Dropbox – Dropbox. – Режим доступу: https://www.dropbox.com/developers/core/sdk – Назва з екрану.
21. SharpBox – Store everything [Електронний ресурс]: SharpBox – Store everything // Dropbox – Dropbox. – Режим доступу: http://sharpbox.codeplex.com – Назва з екрану.

22. Нильс Фергюсон, Брюс Шнайер. Практическая криптографія / Нильс Фергюсон, Брюс Шнайер. – М.: Вильямс, 2005 – 424 с.

23. Шифрование данных – Симметричное шифрование [Електронний ресурс]: Шифрование даннях – Симметричное шифрование // Microsoft – MSDN. – Режим доступу: http://msdn.microsoft.com/ru-ru/library/as0w18af.aspx – Назва з екрану.

24. 10 способов улучшить Dropbox [Електронний ресурс]: 10 способов улучшить Dropbox // 3D News – 3D News. – Режим доступу: http://www.3dnews.ru/software/611599 – Назва з екрану

25. ГОСТ 12.0.003-74. ССБТ. Опасные и вредные производственные факторы. Классификация.

26. ПДК 4617-88. Общесоюзные санитарно-гигиенические и санитарно-противоэпидемические правила и нормы "Предельно допустимые концентрации (ПДК) вредных веществ в воздухе рабочей зоны"

27. ДСН 3.3.6.039-99. Державні санітарні норми виробничої та загальної вібрацій

28. ДСН 3.3.6-037-99. Санітарні норми виробничого шуму, ультразвуку та інфразвуку

29. ДСН 239-96. Санітарні норми і правила захисту населення від впливу електромагнітних випромінювань

30. ДСН 3.3.6.042-99. Державні санітарні норми мікроклімату виробничих приміщень

31. ДБН В.2.5-28-2006. Природне і штучне освітлення

32. О.В. Березюк. Методичні вказівки до виконання лабораторної роботи "Атестація робочих місць за умовами праці" з дисципліни "Охорона праці в галузі" для студентів усіх спеціальностей / Уклад. О.В. Березюк, М.С. Лемешев. – Вінниця: ВНТУ, 2010. – 21 с.

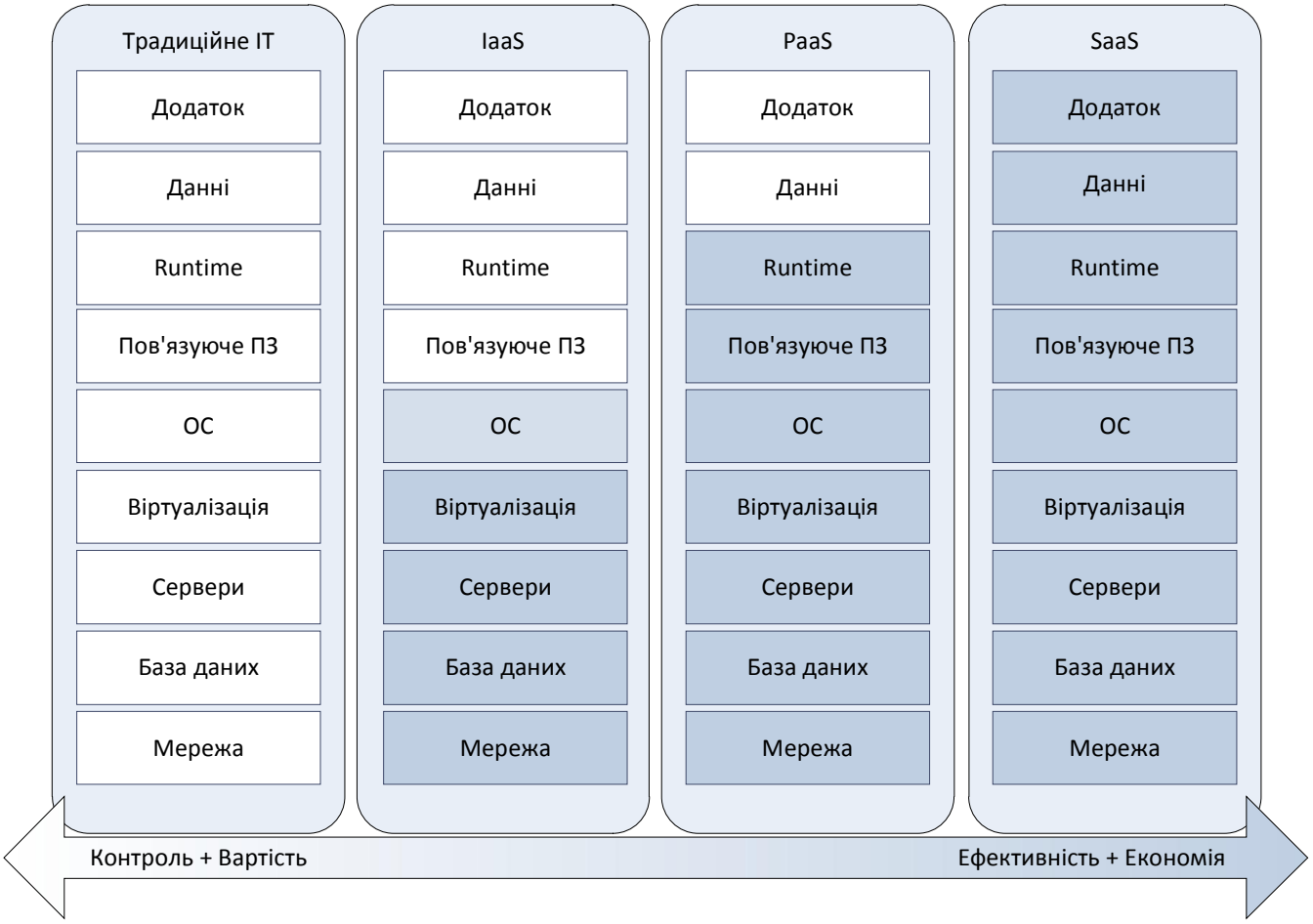
33. Service cloud [Електронний ресурс]: Service cloud // 3d news – 3d news. – Режим доступу: http://www.3dnews.ru/news/614523. – Назва з екрану.

34. Сотрудника Google обвиняют в слежке за пользователями [Електронний ресурс]: Сотрудника Google обвиняют в слежке за пользователями // SecurityLab – SecurityLab. – Режим доступу: http://www.securitylab.ru/news/397773.php. – Назва з екрану.

ДОДАТКИ

Додаток А.

Моделі обслуговування

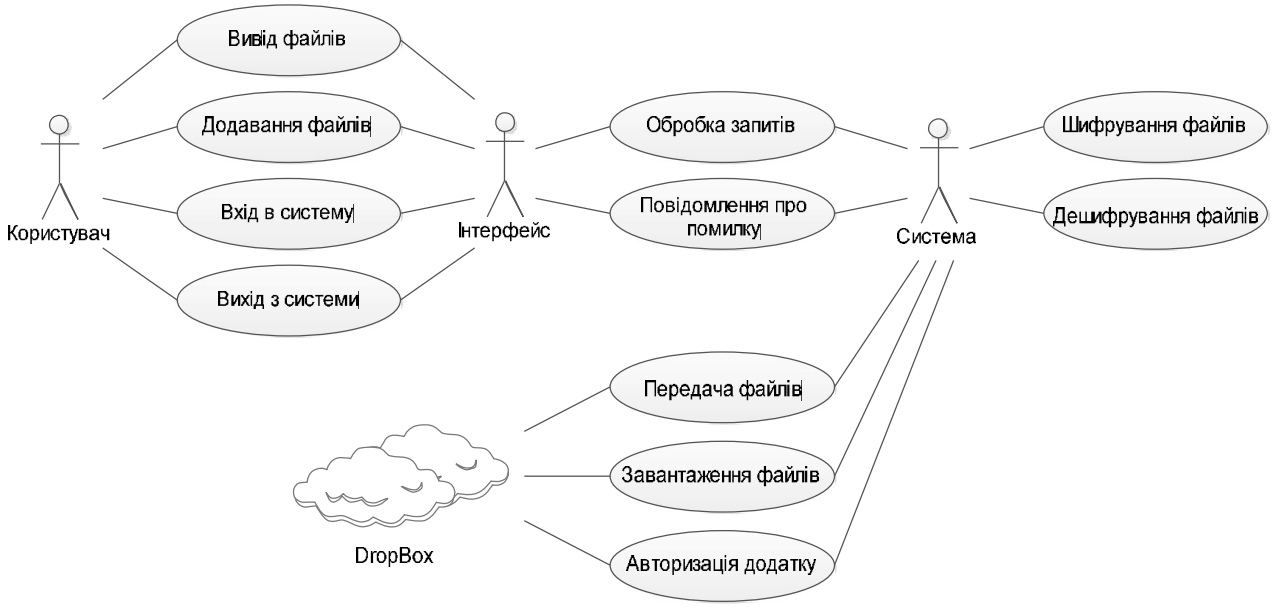


Додаток Б.

Архітектурна модель програми

Додаток В.

Діаграма використання



Додаток Г.

Лістинг класів синхронізації файлів

```
public class UploadManager : IDisposable
{
	private Thread _thread;
	private string[] _oldFiles;
	private string[] _nowFiles;
	private List<string> _deletedFiles;

	/// <summary>
	/// Rise when has been finded file what dosn't exist on DropBox
	/// </summary>
	public event FileSystemEventHandler FileMissedInDropBox;

	/// <summary>
	/// Rise when has been finded file what dosn't exist on PC
	/// </summary>
	public event FileSystemEventHandler FileMissedInComputer;

	/// <summary>
	/// Rise when file has been deleted
	/// </summary>
	public event FileSystemEventHandler FileDeletedInComputer;

	public UploadManager()
	{
	   _deletedFiles = new List<string>();
	   _thread = new Thread(CheckDifference);
	   _thread.Start();
	}

	private void CheckDifference()
	{
	   while (true)
	   {
	      CheckFilesForDelete();
	      CheckSystemFiles();
	      CheckCloudFiles();

	      Thread.Sleep(100);
	   }
	}

	/// <summary>
	/// Compare files on PC with DropBox files. If file
	/// exist on DropBox but not exist on PC
	/// FileMissedInComputer event will be called
	/// </summary>
	private void CheckSystemFiles()
	{
	   var folder = FileService.ConfigManager.FolderPath;
	   var filesInDropBox = DropBoxManager.Instanse.GetFiles();
	   var filesInComputer = Directory.GetFiles(folder, "*.*");
	   var fileNames = filesInComputer.Select(Path.GetFileName).ToList();

	   foreach (var file in filesInDropBox)
```

```
            {
                if (!fileNames.Contains(file) && !_deletedFiles.Contains(file))
                    OnFileMissedInComputer(new FileSystemEventArgs(WatcherChangeTypes.Created, folder, file));
            }
        }

        /// <summary>
        /// Compare files on PC with DropBox files. If file
        /// exist on PC but not exist on DropBox
        /// FileMissedInDropBox event will be called
        /// </summary>
        private void CheckCloudFiles()
        {
            var folder = FileService.ConfigManager.FolderPath;
            var filesInDropBox = DropBoxManager.Instanse.GetFiles();
            var filesInFolder = Directory.GetFiles(folder, "*.*");

            foreach (var file in filesInFolder)
            {
                if (!filesInDropBox.Contains(Path.GetFileName(file)) && !_deletedFiles.Contains(Path.GetFileName(file)))
                    OnFileMissedInDropBox(new FileSystemEventArgs(WatcherChangeTypes.Created, folder,
Path.GetFileName(file)));
            }
        }

        /// <summary>
        /// Compare files on PC with more early comparetion.
        /// If files not exist now but exist before it was
        /// deleted and FileDeletedInComputer event will be called
        /// </summary>
        private void CheckFilesForDelete()
        {
            _deletedFiles = new List<string>();
            var folder = FileService.ConfigManager.FolderPath;
            _nowFiles = Directory.GetFiles(folder, "*.*");
            if (_oldFiles == null)
            {
                _oldFiles = _nowFiles;
                return;
            }

            foreach (var file in _oldFiles)
            {
                if (!_nowFiles.Contains(file))
                {
                    _deletedFiles.Add(Path.GetFileName(file));
                    OnFileDeletedInComputer(new FileSystemEventArgs(WatcherChangeTypes.Deleted, folder,
Path.GetFileName(file)));
                }
            }

            _oldFiles = _nowFiles;
        }

        private void OnFileMissedInComputer(FileSystemEventArgs args)
        {
            var handler = FileMissedInComputer;
            if (handler != null)
                handler.Invoke(this, args);
        }

        private void OnFileMissedInDropBox(FileSystemEventArgs args)
```

```
        {
            var handler = FileMissedInDropBox;
            if (handler != null)
                handler.Invoke(this, args);
        }

        private void OnFileDeletedInComputer(FileSystemEventArgs args)
        {
            var handler = FileDeletedInComputer;
            if (handler != null)
                handler.Invoke(this, args);
        }

        public void Dispose()
        {
            _thread.Abort();
        }
    }


public class ConsoleService
    {
        private UploadManager _uploadManager;

        public void Start()
        {
            _uploadManager = new UploadManager();

            _uploadManager.FileMissedInComputer += DownloadFileFromDropBox;
            _uploadManager.FileMissedInDropBox += UploadFileToDropBox;
            _uploadManager.FileDeletedInComputer += DeleteFileInDropBox;
        }

        private void UploadFileToDropBox(object sender, FileSystemEventArgs e)
        {
            var stream = new FileStream(e.FullPath, FileMode.Open);
            DropBoxManager.Instanse.UploadFile(e.Name, EncryptManager.Encrypt(stream));
        }

        private void DownloadFileFromDropBox(object sender, FileSystemEventArgs e)
        {
            var decryptedBytes = EncryptManager.Decrypt(DropBoxManager.Instanse.DownloadFile(e.Name));
            var fileName = string.Format("{0}/{1}", FileService.ConfigManager.FolderPath, e.Name);
            using (var fileStream = new FileStream(fileName, FileMode.Create))
                fileStream.Write(decryptedBytes, 0, decryptedBytes.Length);
        }
        private void DeleteFileInDropBox(object sender, FileSystemEventArgs e)
        {
            DropBoxManager.Instanse.DeleteFile(e.Name);
        }
        public void Stop()
        {
            _uploadManager.Dispose();
        }
    }
```

Додаток Д.

Лістинг класу інтерфейсу

```
<Controls:MetroWindow x:Class="CryptoBox.GuiManager.MainWindow"
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:Controls="clr-namespace:MahApps.Metro.Controls;assembly=MahApps.Metro"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      Title="CRYPTOBOX" Height="353" Width="427">
   <Window.Resources>
      <ResourceDictionary>
         <ResourceDictionary.MergedDictionaries>
            <ResourceDictionary Source="pack://application:,,,/MahApps.Metro;component/Styles/Colours.xaml" />
            <ResourceDictionary Source="pack://application:,,,/MahApps.Metro;component/Styles/Fonts.xaml" />
            <ResourceDictionary Source="pack://application:,,,/MahApps.Metro;component/Styles/Controls.xaml" />
            <ResourceDictionary Source="pack://application:,,,/MahApps.Metro;component/Styles/Accents/Blue.xaml" />
            <ResourceDictionary Source="pack://application:,,,/MahApps.Metro;component/Styles/Accents/BaseLight.xaml" />
            <ResourceDictionary Source="pack://application:,,,/MahApps.Metro;component/Styles/FlatButton.xaml" />
         </ResourceDictionary.MergedDictionaries>
      </ResourceDictionary>
   </Window.Resources>
   <Grid>
      <Grid.RowDefinitions>
         <RowDefinition Height="20*"/><RowDefinition Height="80*"/><RowDefinition Height="120*"/>
      </Grid.RowDefinitions>
      <Grid.ColumnDefinitions>
         <ColumnDefinition Width="*"/><ColumnDefinition Width="*"/>
         <ColumnDefinition Width="*"/><ColumnDefinition Width="*"/>
      </Grid.ColumnDefinitions>
      <Button x:Name="_buttonStart" Content="Start Synchronization" Margin="10,2,2,10" Grid.Row="5" Grid.Column="0"
Grid.ColumnSpan="2" Click="ButtonStartClick"/>
      <Button x:Name="_buttonStop" Content="End Synchronization"  Margin="2,2,10,10" Grid.Row="5" Grid.Column="2"
Grid.ColumnSpan="2" Click="ButtonStopClick"/>
      <TextBox x:Name="_textBoxFolder" Margin="0,2,10,2" TextWrapping="Wrap" Grid.ColumnSpan="3" Grid.Row="1"
Grid.Column="1"/>
      <TextBox x:Name="_textBoxUserName" Margin="0,2,10,2" TextWrapping="Wrap" Text="TextBox"
Grid.ColumnSpan="3" Grid.Row="2" Grid.Column="1"/>
      <TextBox x:Name="_textBoxUserPassword" Margin="0,2,10,2" TextWrapping="Wrap" Text="" Grid.ColumnSpan="3"
Grid.Row="3" Grid.Column="1" GotFocus="TextBoxUserPasswordGotFocus"
LostFocus="TextBoxUserPasswordLostFocus" TextChanged="PasswordChanged"/>
      <Label Content="Synh Folder" Margin="10,2,5,2" Grid.Row="1" Grid.Column="0"/>
      <Label Content="Username" Margin="10,2,5,2" Grid.Row="2" Grid.Column="0"/>
      <Label Content="Password" Margin="10,2,5,2" Grid.Row="3" Grid.Column="0"/>
      <Controls:ProgressRing x:Name="_progressRing" IsActive="True" Margin="35,10,10,11" Grid.Row="4" Height="60"
Width="60"/>
      <Label x:Name="_labalLogs" Content="Get DropBox information..." Grid.Column="1" HorizontalAlignment="Left"
Margin="10,10,0,0" Grid.Row="4" VerticalAlignment="Top" Grid.ColumnSpan="2" Width="189" Height="26"/>
      <Button x:Name="_buttonBrowse" Content="..." Grid.Column="3" Margin="35,2,12,3" Grid.Row="1" Padding="10,-
6,10,0" Click="BrowseClick"/></Grid></Controls:MetroWindow>
```

Додаток Е.

Лістинг класу шифрування

```
public static class EncryptManager
{
    public static string Password { get; set; }
    public static byte[] Encrypt(Stream stream)
    {
        var pdb = new PasswordDeriveBytes(Password, new byte[]
          {0x49, 0x76, 0x61, 0x6e, 0x20, 0x4d, 0x65, 0x64, 0x76, 0x65, 0x64, 0x65, 0x76});
        var clearData = new byte[stream.Length];
        stream.Read(clearData, 0, clearData.Length);
        var ms = new MemoryStream();
        Rijndael alg = Rijndael.Create();
        alg.Key = pdb.GetBytes(32);
        alg.IV = pdb.GetBytes(16);
        using (var cryptoStream = new CryptoStream(ms, alg.CreateEncryptor(), CryptoStreamMode.Write))
            cryptoStream.Write(clearData, 0, clearData.Length);
        return ms.ToArray();
    }
    public static byte[] Decrypt(byte[] cipherData)
    {
        var pdb = new PasswordDeriveBytes(Password,
            new byte[] {0x49, 0x76, 0x61, 0x6e, 0x20, 0x4d, 0x65, 0x64, 0x76, 0x65, 0x64, 0x65, 0x76});
        var ms = new MemoryStream();
        Rijndael alg = Rijndael.Create();
        alg.Key = pdb.GetBytes(32);
        alg.IV = pdb.GetBytes(16);
        try
        {
            using (var cryptoStream = new CryptoStream(ms, alg.CreateDecryptor(), CryptoStreamMode.Write))
                cryptoStream.Write(cipherData, 0, cipherData.Length);
        }
        catch (Exception)
        {
            return cipherData;
         }
        return ms.ToArray();
    }
}
```

Додаток Ж.

Розрахункова таблиця з охорони праці

Фактори виробничого середовища	Норматив	Фактичне значення
Призначення приміщення		розробка
Склад повітря робочої зони	ПДК 4617-88	
Шкідлива речовина		озон
Концентрація шкідливої речовини в повітрі робочої зони, мг/м^3		0,182
Віброакустичні коливання		
Вид вібрації	ДСН 3.3.6 .039-99	загальна
Еквівалентний рівень віброприскорення, дБ		26
Еквівалентний рівень шуму, дБА	ДСН 3.3.6-037-99	51
Рівень інфразвуку, дБ		46
Рівень ультразвуку, дБ		29
Неіонізуючі випромінювання	ДСН 239-96	
Напруженість електричного поля радіочастотного діапазону, В/м		2
Напруженість електричного поля промислової частоти, кВ/м		2,17
Довжина хвилі випромінювання оптичного діапазону, нм	СанПиН 5804-91	0
Тривалість впливу випромінювання оптичного діапазону, с		0
Доза опромінення оптичного діапазону, Дж		0
Мікроклімат	ДСН 3.3 .6.042-99	
Енерговитрати, Вт		191

Додаток Ж (продовження).

Розрахункова таблиця з охорони праці

Фактори виробничого середовища	Норматив	Фактичне значення
Період року		теплий
Температура повітря для постійних робочих місць, °С		16
Відносна вологість повітря, %		33
Швидкість руху повітря, м/с		1,1
Інтенсивність теплового випромінювання, Вт/м²		226
Виробниче освітлення	ДБН В.2.5-28-2006	
Найменший розмір об’єкта розрізнення, мм		0,88
Контраст об’єкта розрізнення з фоном		середній
Характеристика фону		темний
КПО для бокового, природного освітлення, %		2,4
Освітленість для загального штучного освітлення, лк		57
Варіант розрахункового завдання		6